Sponsors

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Valeo

CONFERENCE PROCEEDINGS

Published by the Irish Pattern Recognition & Classification Society

iprcs.org

ISBN 978-0-9934207-4-0

Welcome

The 2020 Irish Machine Vision and Image Processing Conference (IMVIP 2020) is hosted online this year by IT Sligo.
The IMVIP Conference is Ireland's primary meeting for those researching in the fields of machine vision and image processing. The conference has been running since 1997 and provides a forum for the exchange of ideas and the presentation of research conducted both in Ireland and worldwide. IMVIP is a single-track conference consisting of high quality previously unpublished contributed papers focusing on both theoretical research and practical experiences in all areas.
This is the first time that IT Sligo has hosted the conference and the first year that is has run online. This, was of course, not by choice. We were well advanced with plans for the physical conference and would have loved to have you here with us in Sligo and hope that we will at some time in the future when the Covid-19 crisis ends. We had great plans for a banquet at the Glasshouse Hotel and a welcome reception in Hargadon's Bar with some great local craft beers from a local Sligo brewery called the White Hag. As academics we were able to relatively easily pivot to an online solution but these local hospitality businesses can not. So if you do visit Sligo in the future please consider giving your patronage to these businesses.
We are delighted to have three high-profile keynote speakers this year. Davide Scaramuza, Professor and Director of the Robotics and Perception Group University of Zurich and ETH Zurich, Henning Müller, Professor with responsibility for the eHealth unit HES-SO Valais and University of Geneva and Benazouz Bradai, Autonomous Driving Innovation Platform Manager & Senior Expert Valeo Driving Assistance Research - Paris.
We would like to extend our thanks to all those who have helped facilitate the conference and the pivot to an online conference, for their time, resources and assistance and a warm welcome to all speakers and delegates of the 22nd Irish Machine Vision & Image Processing Conference.

Seán Mullery
Organiser/Editor
Sligo
Ireland
August 2020

Programme Chair

Seán Mullery, IT Sligo

Organising Committee

Eva Murphy, IT Sligo
Shane Gilroy, IT Sligo
Vincent Andrearczyk, HES-SO

Programme Committee

Martin Alain, Trinity College Dublin
Vincent Andrearczyk, HES-SO The University of Applied Sciences and Arts of Western Switzerland
Donald Bailey, Massey University, New Zealand
Francesco Bianconi, Università degli Studi di Perugia, Italy
Kathy Clawson, University of Sunderland, UK
Sonya Coleman, Ulster University
Joan Condell, Ulster University
David Corrigan, Huawei Technologies
Jane Courtney, Technological University, Dublin
Kathleen Curran, University College Dublin
Rozenn Dahyot, Trinity College Dublin
Kenneth Dawson-Howe, Trinity College Dublin
Catherine Deegan, Technological University, Dublin
Cem Direkoglu, Middle East Technical University, Cyprus
Antonio Fernández, Universidad de Vigo, Spain
Bob Fisher, University of Edinburgh, UK
Guillaume Gales, Foundry
Bryan Gardiner, Ulster University
Jonathan Horgan, Valeo Vision Systems
Ciaran Eising, Valeo Vision Systems, UL
Dermot Kerr, Ulster University
Yasuyo Kita, National Institute of Advanced Industrial Science and Technology (AIST), Japan
Vladimir Krylov, Trinity College Dublin
Mikael Le Pendu, Trinity College Dublin
Suzanne Little, Dublin City University
Charles Markham, National University of Ireland, Maynooth
Sally Mcclean, Ulster University
John McDonald, National University of Ireland, Maynooth
Paul McKevitt, Ulster University
Derek Molloy, Dublin City University
Sean Mullery, Institute of Technology Sligo
Omar Nibouche, Ulster University
Robert Sadleir, Dublin City University
Bryan Scotney, Ulster University
Shane Gilroy, IT Sligo
David Vernon, Carnegie Mellon University Africa, Rwanda
Rudi Villing, National University of Ireland, Maynooth
Paul Whelan, VSG DCU
Reyer Zwiggelaar, Aberystwyth University, Wales

Keynote Speaker: Davide Scaramuzza

Davide Scaramuzza (Italian) is Professor of Robotics and Perception at both departments of Informatics (University of Zurich) and Neuroinformatics (University of Zurich and ETH Zurich), where he does research at the intersection of robotics, computer vision, and neuroscience. Specifically he investigates the use of standard and neuromorphic cameras to enable autonomous, agile, navigation of micro drones in search-and-rescue scenarios. He did his PhD in robotics and computer vision at ETH Zurich (with Roland Siegwart) and a postdoc at the University of Pennsylvania (with Vijay Kumar and Kostas Daniilidis). From 2009 to 2012, he led the European project sFly, which introduced the PX4 autopilot and pioneered visual-SLAM-based autonomous navigation of micro drones. For his research contributions in vision-based navigation with standard and neuromorphic cameras, he was awarded the IEEE Robotics and Automation Society Early Career Award, the SNSF-ERC Starting Grant, a Google Research Award, KUKA, Qualcomm, and Intel awards, the European Young Research Award, the Misha Mahowald Neuromorphic Engineering Award, and several conference paper awards. He coauthored the book "Introduction to Autonomous Mobile Robots" (published by MIT Press; 10,000 copies sold) and more than 100 papers on robotics and perception published in top-ranked journals (TRO, PAMI, IJCV, IJRR) and conferences (RSS, ICRA, CVPR, ICCV). In 2015, he cofounded a venture, called Zurich-Eye, dedicated to the commercialization of visual-inertial navigation solutions for mobile robots, which later became Facebook-Oculus Switzerland and Oculus' European research hub. He was also the strategic advisor of Dacuda, an ETH spinoff dedicated to inside-out VR solutions, which later became Magic Leap Zurich. Many aspects of his research have been prominently featured in wider media, such as The New York Times, BBC News, Discovery Channel, La Repubblica, Neue Zürcher Zeitung and in technology-focused media, such as IEEE Spectrum, MIT Technology Review, Tech Crunch, Wired, The Verge.

Keynote Speaker: Henning Müller

Henning Müller studied medical informatics at the University of Heidelberg, Germany, then worked at Daimler-Benz research in Portland, OR, USA. From 1998-2002 he worked on his PhD degree in computer vision at the University of Geneva, Switzerland with a research stay at Monash University, Melbourne, Australia. Since 2002, Henning has been working for the medical informatics service at the University Hospital of Geneva. Since 2007, he has been a full professor at the HES-SO Valais and since 2011 he is responsible for the eHealth unit of the school. Since 2014, he is also professor at the medical faculty of the University of Geneva. In 2015/2016 he was on sabbatical at the Martinos Center, part of Harvard Medical School in Boston, MA, USA to focus on research activities. Henning is coordinator of the ExaMode EU project, was coordinator of the Khresmoi EU project, scientific coordinator of the VISCERAL EU project and is initiator of the ImageCLEF benchmark that has run medical tasks every year since 2004. He has authored over 600 scientific papers with more than 16,000 citations and is in the editorial board of several journals and since 2020 is a member of the Swiss National Research Council.

Keynote Speaker: Benazouz Bradai

Benazouz Bradaï received his PhD degree in multi-sensor fusion in 2007 from Haute Alsace University in France. From 2007 to 2011, he was Algorithm Engineer and Expert for ADAS functions including Lighting Automation, Traffic Signs/lights recognition by camera and multi sensor fusion. Since 2011, he has been working on Automated Driving Innovation developments as Senior Expert and Innovation Platform Manager. His main research interest is Automated Driving with several scientific contributions in multi-sensors fusion, precise localization and mapping and Automated Driving Functional/SW architecture. He is a member of various professional associations including ADASIS Forum, SAE and SIA in France .

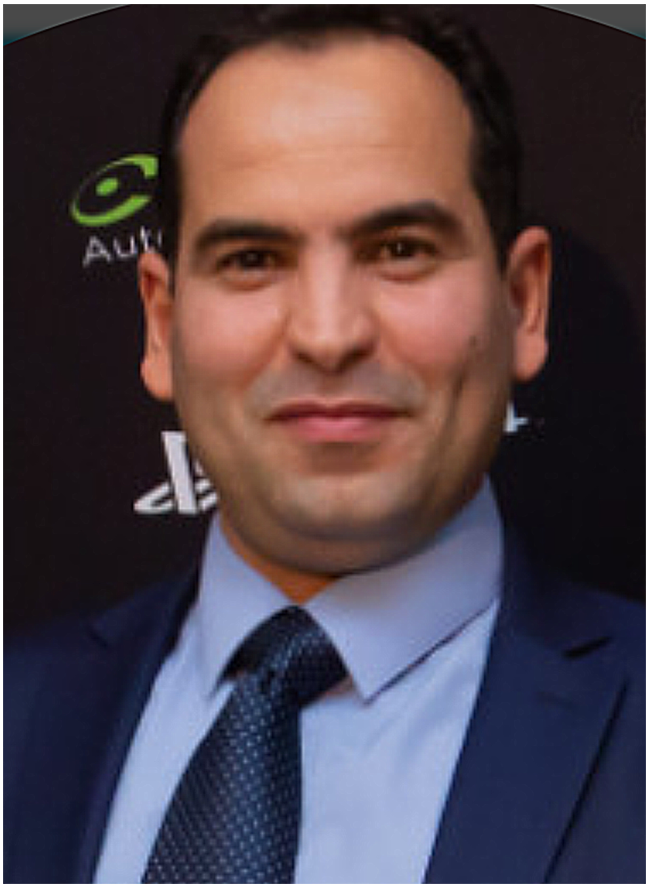


Table of Contents

 ISBN 978-0-9934207-5-7

A Study on Visual Perception of Light Field Content

Ailbhe Gill, Emin Zerman, Cagri Ozcinar, Aljosa Smolic*

V-SENSE, School of Computer Science, Trinity College Dublin, Dublin, Ireland

Abstract

The effective design of visual computing systems depends heavily on the anticipation of visual attention, or saliency. While visual attention is well investigated for conventional 2D images and video, it is nevertheless a very active research area for emerging immersive media. In particular, visual attention of light fields (light rays of a scene captured by a grid of cameras or micro lenses) has only recently become a focus of research. As they may be rendered and consumed in various ways, a primary challenge that arises is the definition of what visual perception of light field content should be. In this work, we present a visual attention study on light field content. We conducted perception experiments displaying them to users in various ways and collected corresponding visual attention data. Our analysis highlights characteristics of user behaviour in light field imaging applications. The light field data set and attention data are provided with this paper.

Keywords: Light fields, rendering, visual perception, visual attention, saliency

1 Introduction

New developments in capture [Broxton et al., 2019] and display technologies [Lee et al., 2016, Overbeck et al., 2018] introduced a novel way of visual media representation - the light field. In contrast to traditional imaging systems, which capture a 3D scene by projecting it onto a 2D surface, light fields [Levoy and Hanrahan, 1996,Gortler et al., 1996] encode all angular, directional and intensity information of light rays travelling within a 3D-space. Light fields can be displayed in 3D on 360-degree displays [Jones et al., 2007] or specialised light field displays [Lanman et al., 2011]. They can also be viewed in 2D, as sheared perspective views or as focal stacks - where images with differing focal planes of the light field appear sharp or "*in focus*". Focal stacks are computed using digital refocusing [Ng et al., 2005, Le Pendu et al., 2019].

Light fields hold more information than a regular image and can be used in various applications [Matysiak et al., 2020] including refocusing [Le Pendu et al., 2019] and streaming [Alain et al., 2019]. It's plausible that visual attention (where people look when they view a scene) varies according to media type. As a type distinct from 2D-image captures of scenes, light fields' relationship with visual attention may differ from conventional images. To our knowledge, saliency of varied renderings of light fields has not been previously investigated.

Hence, in this study, we built a light field visual attention database by bringing light fields from different sources together and collecting eye tracking data for different rendering scenarios of them. Our goal was to obtain ground truth visual attention data for light fields and analyse if it differs from attention in 2D images not generated from light fields. Light field refocusing was our chosen method to render the light fields which was representative of their 3D nature, but for a 2D display. We subsequently examined how changes in focus affected participants' visual attention, treating focus as a cue characteristic of light fields.

The rest of this paper is structured as follows. Section 2 discusses related work. Section 3 introduces the selected light fields and rendering scenarios considered in this study. Details related to the collection of eye-tracking data and user study are given in Section 4. Section 5 provides an analysis and presents the results. We outline our conclusions in Section 6.

*This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/RP/27760.

2 Related Work

Visual saliency is the subjective term describing perceived pertinent regions or elements of a scene which stand out in the scene context. It has been studied for images [Judd et al., 2009] and videos [Itti, 2005] on planar surfaces. Humans have been found to fixate on regions with greater edge density and local contrast [Reinagel and Zador, 1999]. Low level features such as intensity, orientation and color contrast have been found to guide visual attention [Itti et al., 1998] as well as high level features like faces [Buswell, 1935, Yarbus, 2013]. Viewers in visual attention experiments tend to move their gaze to targets near where they are currently looking. There is also a tendency to look at centrally located targets in their field of view. [Parkhurst et al., 2002].

Previous work in light field saliency has focused on object-based methods of visual saliency prediction [Li et al., 2016, Zhang et al., 2015, Sheng et al., 2016, Zhang et al., 2017, Wang et al., 2019, Zhang et al., 2020]. These works represent saliency ground truth as binary maps. These maps are obtained by manually segmenting objects that stand out in all-in-focus renderings of the light fields and human-labelling those segments as 1 and all other regions as 0. They focus on the localisation of instances of dominant objects, not taking into account tracked human gaze. Only one type of light field rendering is considered, effectively ignoring the 3D nature of light fields.

Eye fixation data collected using eye-tracking devices provides a more meaningful form of ground truth for visual attention compared to binary maps, since it represents the statistical distribution of fixation data. Saliency maps [Koch and Ullman, 1987], which are continuous density maps that represent the probability of fixation at every point in an image, can be computed from this data and can be analysed to make more accurate models for predicting visual attention in light fields. Our work addresses the visual saliency of all spatial locations that attract visual attention, be they regions, objects or points of interest, which is in contrast to previous work. We investigated how attention is affected by changes in focus. Our results show that characteristics specific to light fields influence visual attention, as refocusing is a distinctive feature of light field technology.

3 Database

The data was selected from four main light field datasets: Stanford (New) Light Field Archive [Vaish and Adams, 2008], EPFL Light Field Image Dataset [Rerabek and Ebrahimi, 2016], Disney High Spatio-Angular Resolution Light Fields [Kim et al., 2013] and HCI Heidelberg 4D Light Field Dataset [Honauer et al., 2016]. We believe that the selection of these four datasets makes the collected data representative as the light fields were acquired using a camera array [Vaish and Adams, 2008], a single camera with microlens array [Rerabek and Ebrahimi, 2016], a camera on a gantry [Kim et al., 2013], and computer generated imagery [Honauer et al., 2016] respectively.

We selected 20 light fields from these datasets according to the following criteria: they contained multiple regions or objects with high colour contrast between each other and contained regions with great edge density and local contrast at varied depths and spatial locations.

Slices of a light field focused at a sequence of depths form what is known as a focal stack. We generated these focal stacks for each of our light fields using the Fourier Disparity Layers method [Le Pendu et al., 2019] and used them to simulate traversing a 3D scene on a 2D display. We considered three different scenarios for light field rendering and rendered each light field in five ways as follows:

1. **all-in-focus:** all the points in the rendered image are in focus.

2. **region-in-focus:** one slice/image of the focal stack is rendered so only objects at that slice's specific depth of focus appear sharp. We rendered two regions *region-1* and *region-2* which have objects in opposite positions of the frame eg. left/right, top/bottom, foreground/background.

3. **focal-sweep:** all the images of the focal stack are rendered in sequence. We rendered two focal sweeps *front-to-back* with region of focus moving from foreground to background and *back-to-front* with region of focus moving from background to foreground.

 ISBN 978-0-9934207-5-7

This created database is made publicly available on our project webpage[1], in order to support further scientific studies in this field. On this webpage, we will also share some additional results which we could not add to this paper due to page limitations.

4 Eye Tracking Data Collection

4.1 Apparatus & Setup

We used a desktop mounted eye-tracker, the Eyelink 1000 plus [SR Research, 2016] which records eye movements with a sampling rate of 1000Hz. The visual stimuli were presented on a 23.8 inch Dell P2415Q monitor (height × width: 29.6 × 52.7 cm; native resolution: 4K/UHD/2160p; refresh rate: 60Hz). The monitor was placed at 67cm from the users eye which kept the visual angle of the stimuli between 39° and 24°. The resolution of the monitor was set to be 1920 × 1080 pixels (16:9 aspect ratio).

The experiment was held in a quiet, well lit room with white walls. We used the standard Eyelink 1000 chin rest to minimise head movement. We specified the width and height of the monitor as well as the resolution and our measured eye to screen distance in the eye-tracker configuration files. The test script was written in Matlab (R2019b) using the EyeLink Toolbox within Psychtoolbox 3 [Kleiner et al., 2007].

4.2 Participants & Methodology

We conducted the experiment, following ethics approval, on 21 participants (16 male and 5 female), aged between 18 and 37 with a mean age of 25.3. A department wide email was sent to students and staff for recruitment. All participants had normal vision or corrected-to-normal vision. The experiment lasted between 25 and 35 minutes for each participant. A brief oral overview of the experiment as well as an information sheet and consent form were provided to participants. They were instructed to view the stimuli freely and naturally while keeping their heads as still as possible. The chin rest position was fixed but the participants could raise or lower their chair until they were comfortable. The distance from the eye to eyetracker was kept at 53cm.

Eye movements of the left eye only were recorded. We used the Eyelink default monocular nine-point calibration and validation procedure. We showed the participants the light fields rendered all-in-focus for 4 seconds each to acquaint them with the data. They were then shown the five renderings of each light field, for 10 seconds each (120 frames with 12 fps), in randomised order, with a 2 seconds interval between. The interval screen was to ensure fixation was re-centered. Randomisation was used to avoid carryover [Greenwald, 1976].

The eye-tracker records eye events, *saccades*, *fixations* and blinks. Fixations are periods in which an area of a visual scene is kept on the fovea. Saccades are rapid movements of the eyes whose function is to change the point of fixation by directing the fovea towards an area of visual interest [Yarbus, 2013]. In the subsequent analysis section, we use fixations recorded by the EyeLink Core System in an EyeLink data file (EDF).

5 Results

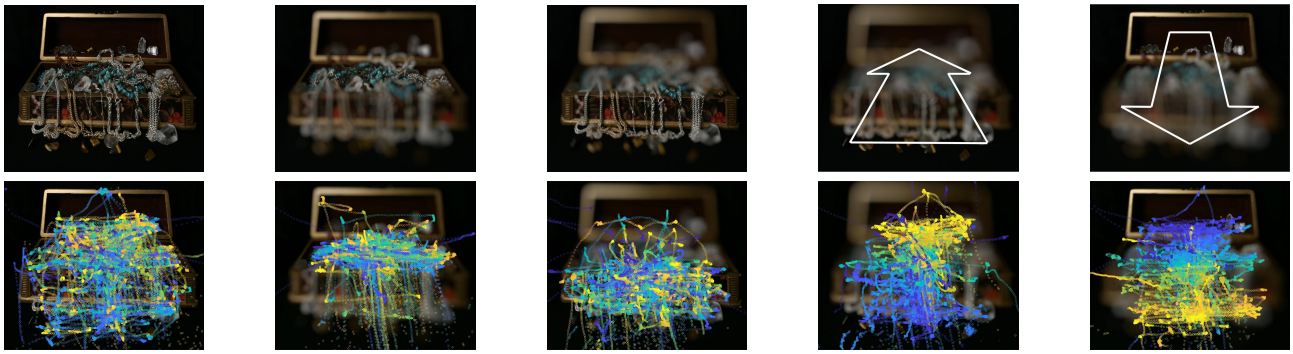
5.1 Qualitative Analysis

Fig. 1 shows the scanpaths of the raw eye tracking data for each rendering of two sample light fields. Clusters where the colours of the scanpaths are the same reveal a clear path of fixation. This can be observed in the scanpaths of the videos where the focal plane varies over time, as shown in the front-to-back and back-to-front renderings in Fig. 1. This is in contrast to the scanpaths of the data collected with non-varying focal planes i.e, all-in-focus, region-1 and region-2, which suggest that each participant views regions of the scene in a different order. This phenomenon is also seen in images rendered from the other light fields in this data set.

We generated saliency maps from the fixation points recorded by our eye-tracker to further analyse the visual saliency patterns in our data. We applied a Gaussian filter to our fixation data to obtain these maps. As

[1] https://v-sense.scss.tcd.ie/research/light-fields/visual-attention-for-light-fields/

 ISBN 978-0-9934207-5-7

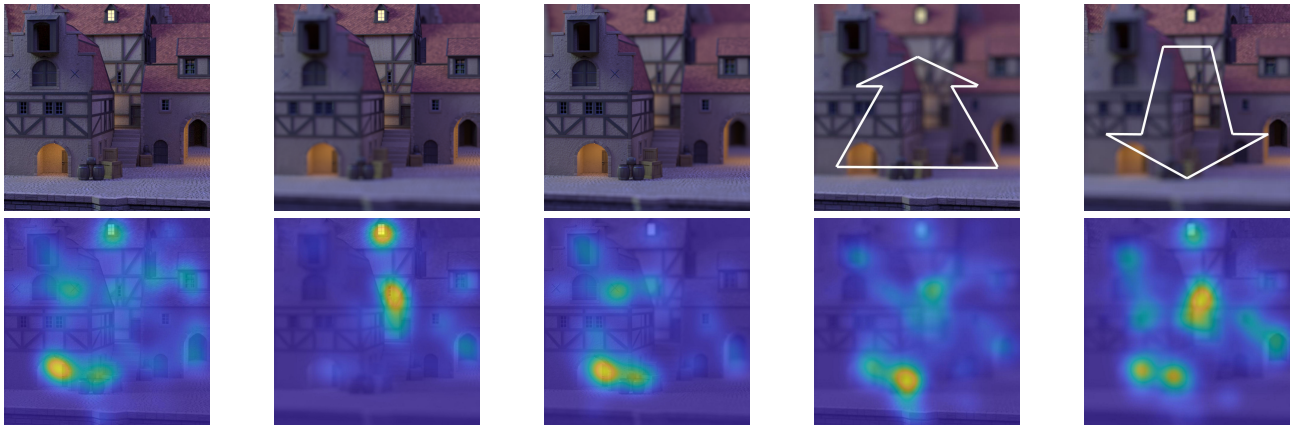


(a) All-in-focus (b) Region-1 (c) Region-2 (d) Front-to-back (e) Back-to-front

Figure 1: Scanpaths of Treasure light field renderings. Each continuous chain represents a participant, and colour mapping shows the passage of time which starts with blue. Yellow is the most recent time instant.

our largest image width was 47.11cm and height was 29.60cm, we calculated the visual angle to be 24.91° to 38.74° respectively. We then found that 1° visual angle corresponds to 47.66 pixels horizontally and 42.67 pixels vertically and used these values as our standard deviations σ_x and σ_y [Le Meur and Baccino, 2013]. We used the duration of the fixations as a weight when computing the Gaussian.

We created saliency maps for each light field and corresponding rendering in two ways. The first method involved computing maps using the fixations of all participants for the full 10 second video. The second split each video over time into 5 segments (of 2 seconds each). For each of these, we generated a saliency map per segment using the fixations of all participants. This allowed us to see changes in visual attention over time.



(a) All-in-focus (b) Region-1 (c) Region-2 (d) Front-to-back (e) Back-to-front

Figure 2: Medieval light field rendered three ways all-in-focus, front-to-back and back-to-front overlayed with a heatmap, generated from all participants and averaged over entire video.

We studied the heatmaps averaged over the entire 10-second video. We found that some had very similar saliency maps for the all-in-focus and focal-sweeps. However, there were also many saliency maps where focal-sweeps revealed other salient regions. For example, in the Medieval light field in Fig. 2, the centre building is salient in the back-to-front rendering whereas it is not in the all-in-focus rendering. This shows that a static rendering of a scene may not reveal all the salient regions present in the 3-dimensional light field data.

Moreover, to understand what causes participants to fixate on regions of focus and whether or not this is always the case, we compared the segments over time of static data (all-in-focus and region-in-focus renderings) to those of focally-varying data (focal-sweeps) for each light field. We found that gaze is held on objects that are in focus when they are also salient in the all-in-focus rendering. For example, the basketball and centre shoe in the Sideboard light field shown in Fig. 3 (a) and (b).

 ISBN 978-0-9934207-5-7

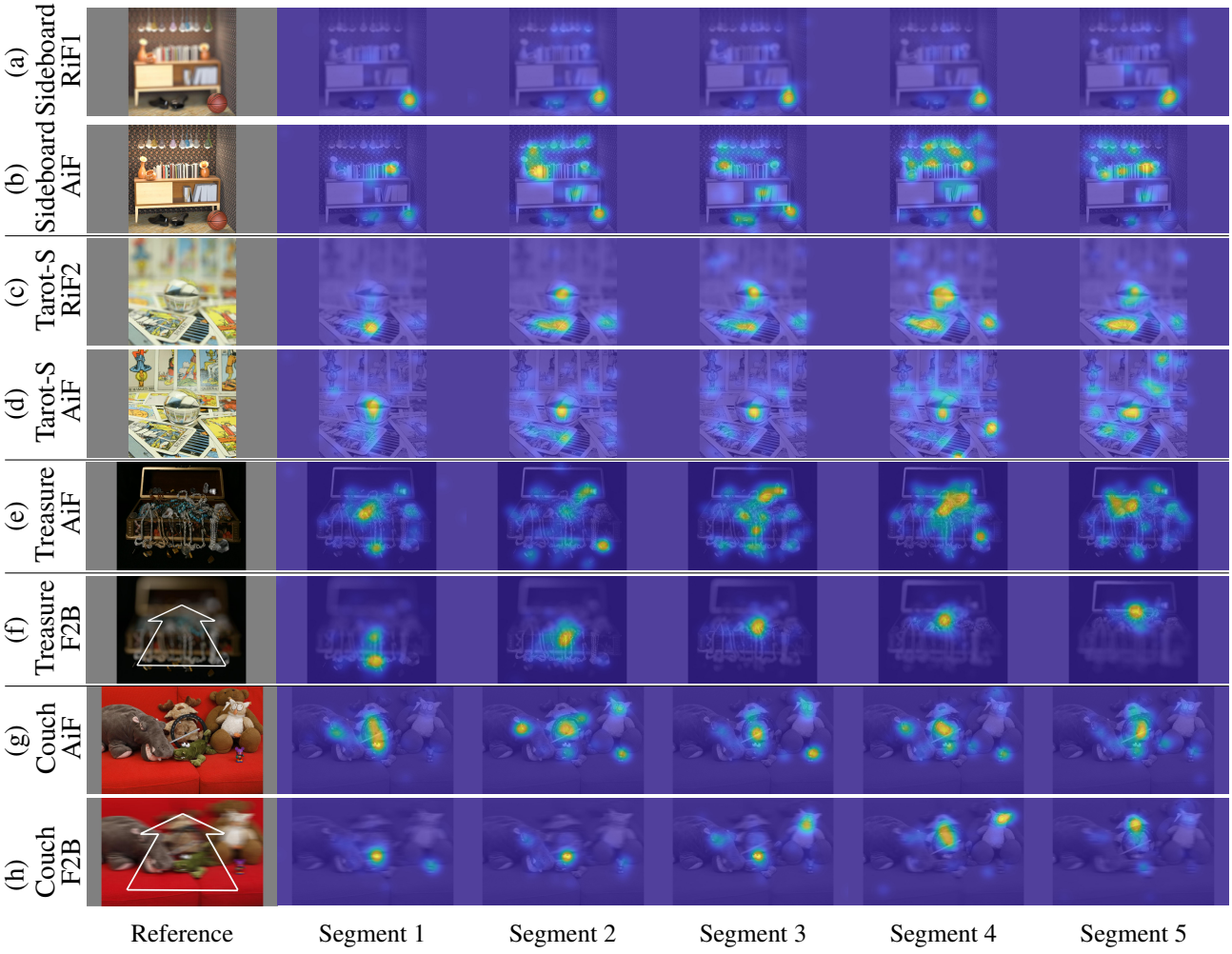


Figure 3: Light Field renderings overlayed with a heatmap generated from all participants and split over time into five 2-second segments. All-in-focus, region-in-focus and front-to-back renderings labelled *AiF*, *RiF*, *F2B*

There are other cases where there are objects in a scene that have a higher level of saliency in the all-in-focus rendering and they pull the viewers attention away from the region of focus in other renderings. For example, observe the region-in-focus rendering of the Tarot-S light field in Fig. 3 (c). Although the focus is in the foreground, the saliency map is also concentrated on the centre ball. As the centre ball has a high saliency in the all-in-focus rendering Fig. 3 (d), we can deduce that it is salient independent of its level of focus.

Some scenes do not depict a specific object/ exhibit a region of high saliency. These tend to produce highly dispersed saliency maps. This is demonstrated in the Treasure light field in Fig. 3 (e). The saliency dispersion in the all-in-focus rendering suggests that the viewer is likely to follow the region of focus almost exclusively in other renderings as seen in Fig. 3 (f).

This trend of following gaze is also evident when there are a few objects with similar levels of saliency in the all-in-focus rendering. This is seen in the animal heads and the central object in the Couch light field Fig. 3 (g). In the focal-sweep Fig. 3 (h), we can see the viewers gaze following the path of focus as above but not as smoothly, rather jumping between the salient objects that are in focus in each segment.

5.2 Quantitative Analysis

In this subsection, we provide a quantitative analysis to understand the relationship between human visual perception and light field rendering.

 ISBN 978-0-9934207-5-7

We infer from our qualitative analysis that viewers are likely to fixate on and follow regions in focus and that fixations of all-in-focus renderings were more dispersed. To further examine these observations, we used the calculated entropy of fixations recorded by the eye tracker to determine if participants' fixations varied more or less for each rendering per light field.

To calculate entropy, we first created a *fixation map* with the same spatial resolution as the stimuli (i.e., 1920×1080), which was populated with the fixations from all the users for a specific case, using Eqn. 1:

$$F_C(i,j) = \begin{cases} 1 & \text{if there is a fixation at } I(i,j) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where F is the fixation map, i and j are the row and column pixels, I is the image stimulus, and $C \in \{\text{AiF}, \text{RiF1}, \text{RiF2}, \text{B2F}, \text{F2B}\}$ denotes the rendering. This yields a fixation map which is sparse. The entropy values were generated for each of the five cases and reported in Table 1 using Matlab's `entropy(`F_C`)` function, computing the probability of 1s occuring in F_C.

The results show that all-in-focus renderings have higher entropy values on average compared to region-in-focus and focal-sweep ones, which suggests participants were more focused on average in the *focal-sweep* and *region-in-focus* cases, compared to the *all-in-focus* case. A two-tailed t-test confirmed that *all-in-focus* was significantly different than others ($\alpha = 0.05$). The differences among the other cases were not statistically significant.

Table 1: Entropy analysis results. The five cases are given in the columns: All-in-focus (*AiF*), region-in-focus #1 (*RiF1*), region-in-focus #2 (*RiF2*), back-to-front focal-sweep (*B2F*), and front-to-back focal-sweep (*F2B*).

Light fields	**AiF**	**RiF1**	**RiF2**	**B2F**	**F2B**
Boardgames	4.06	3.87	3.81	3.96	3.82
Church	4.12	3.46	3.64	3.48	3.41
Couch	4.27	4.04	3.82	3.64	3.52
Dino	3.92	3.82	3.61	3.82	3.67
Dishes	4.19	4.06	3.26	3.48	3.50
Friends	4.55	4.49	4.45	4.25	4.45
LegoKnights	4.02	3.45	3.62	3.27	3.34
Mansion	4.12	4.04	3.67	3.48	3.24
Medieval	4.02	3.71	3.83	3.65	3.76
Pens	3.91	3.68	3.79	3.84	3.80
Platonic	3.79	3.40	3.14	3.53	3.55
Sideboard	3.79	3.75	3.81	3.98	3.90
Table	3.85	3.47	3.85	3.76	3.65
Tarot-L	4.92	3.46	4.39	3.44	3.73
Tarot-S	4.62	3.85	3.80	3.91	4.16
Tower	3.77	3.72	3.84	3.63	3.42
Town	4.02	4.03	4.06	4.02	4.02
Treasure	4.01	3.47	3.66	3.54	3.46
Vespa	3.81	3.84	3.78	3.45	3.82
Vinyl	4.18	3.60	3.87	4.16	3.77
Average	4.10	3.76	3.78	3.71	3.70

6 Conclusion

In this paper, we outline our investigation into how attention is affected by changes in focus to verify whether characteristics specific to light fields influence visual attention. From analysis of the scanpaths of light fields, we conclude that there is a difference in visual attention of static renderings of light fields when compared to focally-varying renderings. This was reinforced by examining the saliency maps of the different rendering types. We found that visual attention was often guided by focus and objects/regions at the focal plane by observing saliency maps computed on segments of the renderings over time.

The ground truth data of previous light field saliency works are segmented objects of all-in-focus renderings. We found that these do not fully capture the visual attention of light fields. Salient information not present in the all-in-focus planar rendering is often revealed by saliency maps of focal-sweep renderings. It is also apparent in saliency maps that viewers did not only fixate on objects. Furthermore, the saliency maps of segmented all-in-focus renderings were more dispersed than those of other renderings. This observation was supported by analysis of the entropy of the fixation data for each rendering where we found that all-in-focus data had highest entropy which suggested greater randomness in the data. This variation in the visual attention of different renderings shows the limitations in the use of a saliency map of only one rendering type as ground truth.

We plan to use our saliency maps as ground truth data in future work for eye fixation prediction for light fields. As they depict the likelihood of eye fixation at every point of a light field capture they have applications in this field among others such as light field rendering and compression.

 ISBN 978-0-9934207-5-7

References

[Alain et al., 2019] Alain, M., Ozcinar, C., and Smolic, A. (2019). A study of light field streaming for an interactive refocusing application. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3761–3765.

[Broxton et al., 2019] Broxton, M., Busch, J., Dourgarian, J., DuVall, M., Erickson, D., Evangelakos, D., Flynn, J., Overbeck, R., Whalen, M., and Debevec, P. (2019). A low cost multi-camera array for panoramic light field video capture. In *SIGGRAPH Asia 2019 Posters*, SA '19, New York, NY, USA. Association for Computing Machinery.

[Buswell, 1935] Buswell, G. T. (1935). *How people look at pictures: a study of the psychology and perception in art.* Univ. Chicago Press.

[Gortler et al., 1996] Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. (1996). The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 43–54, New York, NY, USA. Association for Computing Machinery.

[Greenwald, 1976] Greenwald, A. G. (1976). Within-subjects designs: To use or not to use? *Psychological Bulletin*, 83(2):314.

[Honauer et al., 2016] Honauer, K., Johannsen, O., Kondermann, D., and Goldluecke, B. (2016). A dataset and evaluation methodology for depth estimation on 4D light fields. In *Asian Conference on Computer Vision*, pages 19–34. Springer.

[Itti, 2005] Itti, L. (2005). Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes. *Visual Cognition*, 12(6):1093–1123.

[Itti et al., 1998] Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1254–1259.

[Jones et al., 2007] Jones, A., McDowall, I., Yamada, H., Bolas, M., and Debevec, P. (2007). Rendering for an interactive 360 light field display. *ACM Transactions on Graphics (TOG)*, 26(3):40.

[Judd et al., 2009] Judd, T., Ehinger, K., Durand, F., and Torralba, A. (2009). Learning to predict where humans look. In *IEEE International Conference on Computer Vision (ICCV)*.

[Kim et al., 2013] Kim, C., Zimmer, H., Pritch, Y., Sorkine-Hornung, A., and Gross, M. H. (2013). Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73–1.

[Kleiner et al., 2007] Kleiner, M., Brainard, D., Pelli, D., Ingling, A., Murray, R., Broussard, C., et al. (2007). What's new in Psychtoolbox-3. *Perception*, 36(S).

[Koch and Ullman, 1987] Koch, C. and Ullman, S. (1987). Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer.

[Lanman et al., 2011] Lanman, D., Wetzstein, G., Hirsch, M., Heidrich, W., and Raskar, R. (2011). Polarization fields: Dynamic light field display using multi-layer lcds. *ACM Trans. Graph.*, 30(6):1–10.

[Le Meur and Baccino, 2013] Le Meur, O. and Baccino, T. (2013). Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266.

[Le Pendu et al., 2019] Le Pendu, M., Guillemot, C., and Smolic, A. (2019). A fourier disparity layer representation for light fields. *IEEE Transactions on Image Processing*, 28(11):5740–5753.

[Lee et al., 2016] Lee, S., Jang, C., Moon, S., Cho, J., and Lee, B. (2016). Additive light field displays: Realization of augmented reality with holographic optical elements. *ACM Trans. Graph.*, 35(4).

[Levoy and Hanrahan, 1996] Levoy, M. and Hanrahan, P. (1996). Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA. ACM.

[Li et al., 2016] Li, N., Ye, J., Ji, Y., Ling, H., and Yu, J. (2016). Saliency detection on light field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1605–1616.

[Matysiak et al., 2020] Matysiak, P., Grogan, M., Le Pendu, M., Alain, M., Zerman, E., and Smolic, A. (2020). High quality light field extraction and post-processing for raw plenoptic data. *IEEE Transactions on Image Processing*.

[Ng et al., 2005] Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P., et al. (2005). Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11):1–11.

[Overbeck et al., 2018] Overbeck, R. S., Erickson, D., Evangelakos, D., and Debevec, P. (2018). Welcome to light fields. In *ACM SIGGRAPH 2018 Virtual, Augmented, and Mixed Reality*, SIGGRAPH '18, New York, NY, USA. Association for Computing Machinery.

[Parkhurst et al., 2002] Parkhurst, D., Law, K., and Niebur, E. (2002). Modeling the role of salience in the allocation of overt visual attention. *Vision research*, 42(1):107–123.

[Reinagel and Zador, 1999] Reinagel, P. and Zador, A. M. (1999). Natural scene statistics at the centre of gaze. *Network: Computation in Neural Systems*, 10(4):341–350.

[Rerabek and Ebrahimi, 2016] Rerabek, M. and Ebrahimi, T. (2016). New light field image dataset. In *8th International Conference on Quality of Multimedia Experience (QoMEX)*.

[Sheng et al., 2016] Sheng, H., Zhang, S., Liu, X., and Xiong, Z. (2016). Relative location for light field saliency detection. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1631–1635. IEEE.

[SR Research, 2016] SR Research (2016). Eyelink 1000. `http://www.sr-research.com/eyelink1000.html`. [online].

[Vaish and Adams, 2008] Vaish, V. and Adams, A. (2008). The (new) stanford light field archive. `http://lightfield.stanford.edu`. [online].

[Wang et al., 2019] Wang, T., Piao, Y., Li, X., Zhang, L., and Lu, H. (2019). Deep learning for light field saliency detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8838–8848.

[Yarbus, 2013] Yarbus, A. L. (2013). *Eye movements and vision*. Springer.

[Zhang et al., 2020] Zhang, J., Liu, Y., Zhang, S., Poppe, R., and Wang, M. (2020). Light field saliency detection with deep convolutional networks. *IEEE Transactions on Image Processing*, 29:4421–4434.

[Zhang et al., 2015] Zhang, J., Wang, M., Gao, J., Wang, Y., Zhang, X., and Wu, X. (2015). Saliency detection with a deeper investigation of light field. In *IJCAI*, pages 2212–2218.

[Zhang et al., 2017] Zhang, J., Wang, M., Lin, L., Yang, X., Gao, J., and Rui, Y. (2017). Saliency detection on light field: A multi-cue approach. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 13(3):1–22.

A Study of Efficient Light Field Subsampling and Reconstruction Strategies

Yang Chen, Martin Alain, and Aljosa Smolic

V-SENSE project
Graphics Vision and Visualisation group (GV2)
Trinity College Dublin *

Abstract

Limited angular resolution is one of the main obstacles for practical applications of light fields. Although numerous approaches have been proposed to enhance angular resolution, view selection strategies have not been well explored in this area. In this paper, we study subsampling and reconstruction strategies for light fields. First, different subsampling strategies are studied with a fixed sampling ratio, such as row-wise sampling, column-wise sampling, or their combinations. Second, several strategies are explored to reconstruct intermediate views from four regularly sampled input views. The influence of the angular density of the input is also evaluated. We evaluate these strategies on both real-world and synthetic datasets, and optimal selection strategies are devised from our results. These can be applied in future light field research such as compression, angular super-resolution, and design of camera systems.

Keywords: Light Fields, Angular Super-resolution, Reconstruction and Subsampling, View Synthesis

1 Introduction

A light field (LF) is described as all light rays passing through a given 3D volume in the pioneer work of Levoy et al. [Levoy and Hanrahan, 1996]. Different from traditional 2D imaging systems, a 4D LF imaging system captures not only two spatial but also two additional angular dimensions. The two plane representation of light fields is adopted in this paper, which is represented as a collection of views taken from several view points parallel to a common plane, as shown in Figure 1. Another common representation of light fields are Epipolar Plane Images (EPI), which are 2D slices of the 4D light field obtained by fixing one spatial and one angular dimension.

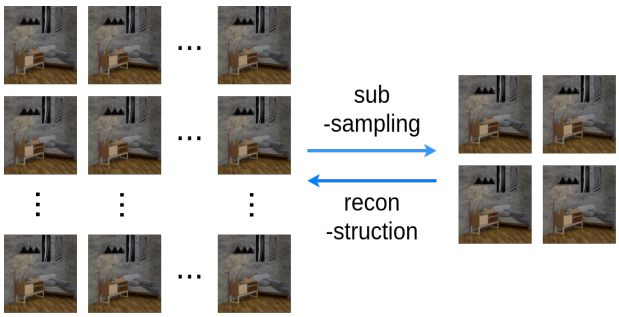
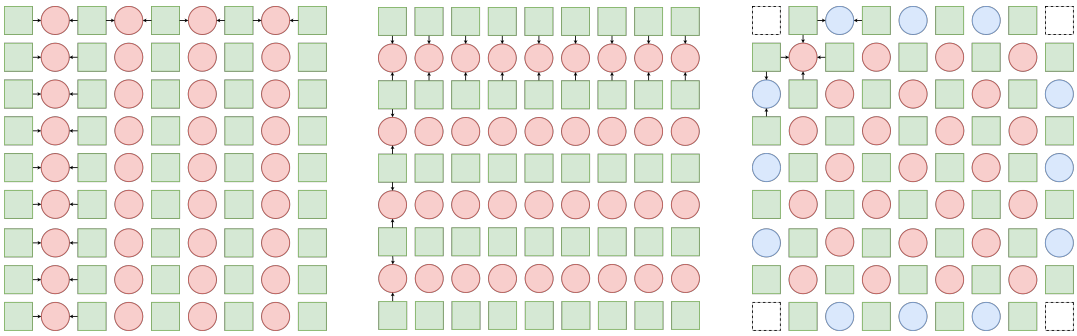


Figure 1: Subsampling and reconstruction of a two-plane representation light field.

Dense light fields are advantageous for various applications including virtual reality, augmented reality and image-based rendering [Yu, 2017, Wu et al., 2017a]. However, dense light fields are usually difficult to capture in practice and a trade-off has to be found between spatial and angular resolution. For example, camera arrays usually have good spatial resolution but sparse angular sampling, while the opposite is usually the case for plenoptic cameras.

*This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/RP/2776. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

 ISBN 978-0-9934207-5-7



(a) row-wise (b) column-wise (c) checkerboard

Figure 2: Three basic subsampling strategies. The squares are sampled views, the circles are reconstructed views and the dashed squares are unused views. Blue circles in (c) are reconstructed by row-wise or column-wise from two adjacent views to complete the LF.

Spatial super-resolution of light fields is a widely studied problem and many methods have been proposed with impressive results [Alain and Smolic, 2018, Rossi and Frossard, 2017, Yoon et al., 2015]. As for the angular resolution enhancement in light fields, one common solution is to apply view synthesis methods on LFs [Yoon et al., 2015, Kalantari et al., 2016, Wang et al., 2018, Vagharshakyan et al., 2017]. However, only limited attention has been paid to view selection strategies and previous work usually arbitrarily selects one fixed input pattern, such as along one angular row or within a $N \times N$ square matrix of views.

In this paper, we investigate light field subsampling and reconstruction strategies and evaluate their performance. We identify three issues to face: First, a benchmark method is required to compare these strategies. State-of-the-art (SOTA) light field subsampling and reconstruction methods are reviewed and evaluated experimentally, and the best performing approach is chosen as the benchmark for following experiments. Second, since LFs contain huge amounts of data, which poses challenges for storage and transmission, efficient subsampling methods are needed for LFs. Given a dense LF as input, we investigate which subsampling strategy can produce the best reconstruction results while keeping a satisfying sampling ratio. Third, since dense LFs are expensive to capture, reconstructing a dense LF from sparse input becomes an important and challenging topic. Here we investigate different strategies and different levels of sparsity for the task of reconstructing a LF from corner views. Eventually, we derive optimal strategies for each problem based on our results.

This paper is organized as follows. In Section 2, we review existing methods for light field angular super-resolution and more general techniques for view synthesis and video frame interpolation. In Section 3, a benchmark method is selected and the experiments for evaluation of subsampling and reconstruction strategies are described in detail. Then, these designs are evaluated and compared in Section 4 after applying the selected benchmark method. Finally, we present our conclusions in Section 5.

2 Related Work

Angular Super-resolution for Light Fields Wanner and Goldluecke proposed a variational framework to generate novel views from sparse input views, of which ghost artifacts can still be observed on the synthesized results [Wanner and Goldluecke, 2013]. Densely sampled light fields have small disparity between adjacent views, which makes them very suitable for frequency domain analysis. Therefore, Shi et al. proposed to reconstruct the dense light field by optimizing frequency coefficients in the continuous 2D Fourier domain [Shi et al., 2014]. A more advanced shearlet transform is adapted in [Vagharshakyan et al., 2017] on Epipolar Plane Images. However, wide EPI input is required by these Fourier methods, which is not feasible for some applications.

The strength of deep learning based approaches has been demonstrated in sevaral LF super-resolution methods. A first example designed a multi-stream network for light field spatial and angular super-resolution [Yoon et al., 2015]. Kalantari et al. proposed a learning based framework for light field view synthesis, which requires depth estimation as an intermediate step [Kalantari et al., 2016]. A sparse input consisting of the cor-

Table 1: Comparison between SOTA LF interpolation methods

row-wise	PSNR/SSIM		
	Mean	HCI	Stanford
Linear	32.37/0.9464	31.82/0.9501	33.47/0.9390
Shearlet [Vagharshakyan et al., 2017]	34.92/0.9592	35.78/0.9750	33.21/0.9277
LFEPI [Wu et al., 2017b]	37.39/0.9451	37.51/0.9420	37.15/0.9515
SepConv [Niklaus et al., 2017]	**38.94/0.9910**	**39.38/0.9945**	**38.07/0.9839**

Table 2: Comparison of pretrained, fine-tuned and retrained SepConv for row-wise interpolation

row-wise	pretrained	Fine-tuned	Retrained
PSNR(dB)	38.08	**39.74**	39.41
SSIM	0.9893	**0.9925**	0.9923

ner views of a dense light field is fed into a convolutional neural network to synthesize the original intermediate views. Wu et al. re-modeled light field angular super-resolution as a detail restoration problem in 2D EPI space (LFEPI) [Wu et al., 2017b]. A blur-restoration-deblur framework is built to process EPIs of a sparsely sampled LF and to recover the angular detail with a convolutional neural network. To utilize inherent consistency of the LF, Wang et al. introduced a pseudo 4D convolution by combining a 2D convolution on EPIs and a sequential 3D convolution [Wang et al., 2018]. While these methods presented innovative ideas and good results, fixed subsampling and reconstruction strategies are applied without investigating and optimising alternatives.

View Synthesis and Video Frame Interpolation Existing depth image-based rendering and video frame interpolation methods can be directly applied to the LF angular super-resolution problem. A deep architecture was introduced by Flynn et al. to synthetize novel views from wide disparity real-world inputs [Flynn et al., 2016]. Niklaus et al. proposed pairs of spatially individual 1D kernels, which are estimated from a trained convolutional neural network, to estimate motion and color interpolation within one stage (SepConv) [Niklaus et al., 2017].

3 Study of efficient subsampling and reconstruction strategies

In this section, we investigate different strategies for light field subsampling and reconstruction. Firstly, a benchmark LF view interpolation method has to be selected from SOTA approaches to evaluate all strategies. Next, given a fixed sampling ratio, three light field subsampling strategies are studied to reconstruct full size LFs from each sampled LF (Figure 2). Finally, six different reconstruction strategies are explored to generate a dense light field from inputs of varying sparsity (Figures 3 & 4).

3.1 Benchmark method selection

As our goal is to investigate sampling and reconstruction strategies for LFs, we first select a benchmark method to be applied in our experiments. The benchmark has to be flexible to work in different configurations but should also provide best possible interpolation quality. We therefore evaluate SepConv [Niklaus et al., 2017], Shearlet [Vagharshakyan et al., 2017] and LFEPI [Wu et al., 2017b] in an initial study. LFEPI is a representative learning based LF view synthesis method, and Shearlet is an efficient non-learning based reconstruction method in the Fourier domain. SepConv [Niklaus et al., 2017] was initially designed for video frame interpolation and employs a neural network based kernel estimator to interpolate views between adjacent input views. As such it is very flexible and can be also be used in various ways of LF view interpolation.

These SOTA methods are evaluated by using the same input pattern shown in Figure 2a resulting from row-wise sampling, in which sampled input views are represented as green squares and reconstructed output views are represented as red circles. According to Table 1, SepConv numerically outperforms all other methods significantly. Shearlet achieves better performance than linear interpolation. LFEPI scores well on PSNR, while Shearlet performs better in terms of average SSIM. However, both these two methods require an EPI as input. Thus SepConv is not only the best performing approach, but also the only one that can easily be adapted to

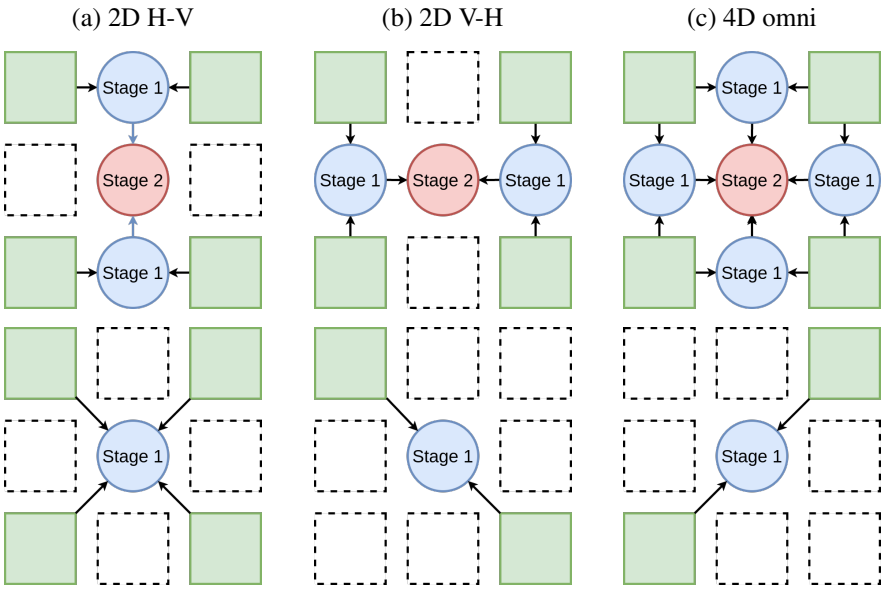


Figure 3: Six reconstruction strategies to interpolate 3x3 views from 4 input corner views (green), dashed square views are not used for interpolation, different colors of circles identify output views from different stages

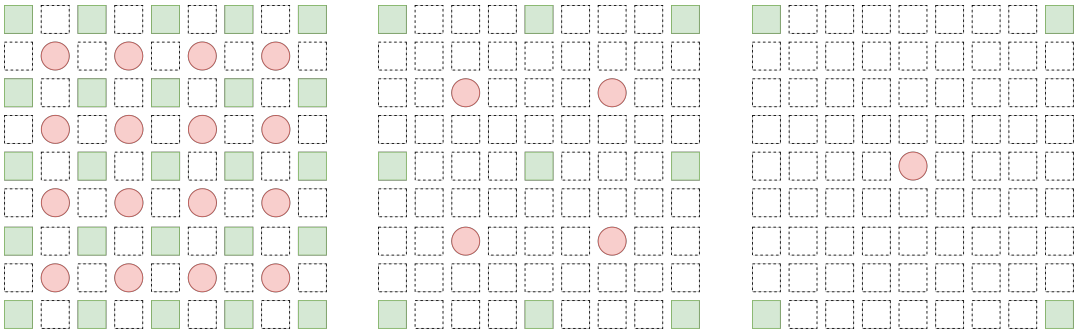


Figure 4: Three levels of angular density for LF reconstruction. The IRs of these levels are 30.9%, 11.1% and 4.9% respectively.

different configurations, as it only needs a pair of RGB images as input. We therefore continue to use SepConv in our further experiments.

Since SepConv was originally trained for video frame interpolation, we additionally fine-tuned the pre-trained model on our light field training dataset in order to further improve the performance. Table 2 shows improvements we can achieve by fine-tuning and retraining the initial network. For some of our experiments detailed below, we have to retrain SepConv appropriately in order to work with more than 2 input views, e.g. left, right, top and bottom neighbors.

3.2 Study of basic light field subsampling strategies

In this set of experiments, we compare basic subsampling strategies as illustrated in Figure 2, with the goal to identify the most efficient basic subsampling strategy among row-wise, column-wise and checkerboard. To evaluate these strategies, the sampled LFs are reconstructed using view interpolation. After reconstruction, the quality of the syntesized views can be compared to the corresponding ground truth. All these strategies have the same ratio of sampled views to total views, which is an important measure of the sparsity and defined as the $InputRatio$ (IR) in equation (1):

 ISBN 978-0-9934207-5-7

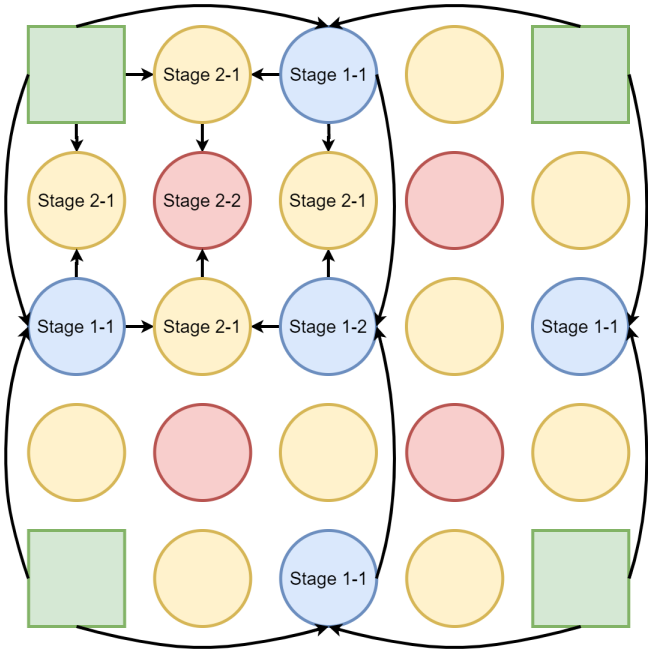


Figure 5: Stages to complete a quarter of the full LF for an input angular density of level 2 with $IR = 11.1\%$ (see Fig. 4)

$$IR = \frac{N_{InputViews}}{N_{InputViews} + N_{OutputViews}} \tag{1}$$

where the numbers of input views and output views of the interpolation method are $N_{InputViews}$ and $N_{OutputViews}$, respectively. The total number of views of the completed light field can be represented as the sum of $N_{InputViews}$ and $N_{OutputViews}$. The three basic subsampling strategies as shown in Figure 2 all have $IR \approx 55\%$.

We applied our fine-tuned SepConv as explained in Section 3.1 to interpolate the necessary views for row-wise and column-wise strategies from neighbouring views. For the checkerboard pattern we had to modify the original SepConv network in order to accept 4 views as input, top, bottom, left and right. The outer views depicted as blue circles in Figure 2c were synthesized by row-wise or column-wise interpolation from 2 neighbouring views.

3.3 Study of sparse light field reconstruction strategies

In this set of experiments, we compare different reconstruction strategies for sparse light fields. Taking a 3x3 matrix of views as example, six progressive reconstruction strategies can be applied as presented in Figure 3. Using four corner images as input, we can reconstruct the side images using the same row-wise and column-wise interpolation as before. Thus, the question becomes which is the best way to reconstruct the central view. Three of these strategies, including 2D horizontal-vertical (2D H-V), 2D vertical-horizontal (2D V-H) and 4D omni, are two stage cases involving generating side views as intermediate stage. The other three, including 4D diagonal, 2D left diagonal and 2D right diagonal, require only one stage to generate the central view.

To further study the influence of the angular density of the input views, we investigate three levels of density as shown in Figure 4. The distance between two input views is 1 view in level 1, and there are 3 views and 7 views distance in level 2 and 3, respectively. To compare fully reconstructed light fields, we reconstruct all missing views using appropriate strategies as follows. Level 1 is completed by row-wise and column-wise interpolation of the side views, using the 2D H-V method unless otherwise specified. The choice of 2D H-V as the default method is justified by its better performance demonstrated in Section 4. For level 2 and 3, we recursively fill using lower level methods to complete the full scale LF. The reconstruction of a quarter of the

 ISBN 978-0-9934207-5-7

Table 3: Evaluation of three basic subsampling strategies from Figure 2

	row-wise		column-wise		checkerboard	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
HCI	40.49	0.9956	40.33	0.9958	39.24	0.9935
Lytro	39.32	0.9932	39.07	0.9925	38.67	0.9921
Stanford	40.22	0.9936	39.37	0.9924	39.47	0.9928
Mean	**39.74**	**0.9925**	39.20	0.9915	39.33	0.9918

Table 4: Evaluation of six reconstruction strategies from Figure 3

	PSNR	SSIM
2D H-V	**37.42**	**0.9884**
2D V-H	37.21	0.9881
2D left diagonal	35.77	0.9846
2D right diagonal	35.86	0.9842
4D omni	36.51	0.9863
4D diagonal	36.86	0.9838

LF is shown in Figure 5, which is applied iteratively to each quarter one by one. Please find more detailed completion stages in the supplementary material [1].

4 Results

In this section, we summarize the results of our experiments, which were performed on an Intel Core i7-6700k 4.0GHz CPU, while neural network refining was performed on Nvidia Titan Xp GPUs.

The performance of strategies is evaluated on both real-world and synthetic LF datasets to validate their robustness. We used 27 real-world light fields captured by Lytro Illum cameras provided by EPFL [Rerabek and Ebrahimi, 2016] and INRIA [INR,], and 11 light fields from the Stanford dataset taken by a camera gantry [Sta,]. As for the synthetic LF dataset, all 28 light fields from the HCI benchmark [Honauer et al., 2016] were used. 10 light fields in total were selected from these datasets as the test set and the rest as training set. Additionally, 160 light fields from LF intrinsic [Alperovich et al., 2018] were added for retraining of SepConv to avoid the overfitting. All views were cropped to equal 512x512 resolution to accelerate the computation. In this study, we use the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) on RGB images to evaluate the algorithms numerically. For each light field, unless emphasized specifically, mean numerical results are computed over all views of the full light fields.

The results of the evaluation of basic subsampling strategies are shown in Table 3. Row-wise interpolation achieves the best scores on most light fields over all datasets. Please view our website for comprehensive results of each light field [1]. The difference to column-wise is most prominent for the Stanford data which was captured by a gantry, resulting in unequal vertical displacement. For the checkerboard pattern, the retraining of SepConv to accept 4 views may be the reason of its worse performance, as this way it could not benefit from the pretrained model of SepConv.

The results of sparse LF reconstruction strategies are shown in Table 4. Again, 4D strategies use a retrained model while the others use a fine-tuned model. The central view is used to evaluate these results as it is the only one that gets always reconstructed using any of the six strategies when filling a 3x3 block of views. 2D H-V performs best, as it accumulates less error than other strategies with 2 stages (row-wise first gives best reference for stage 2), while it has smaller distance between input views compared to direct diagonal strategies. Visual results of reconstruction strategies are shown in Figure 6. Occlusion artifacts around the tip of sword can be observed in diagonal strategies.

Finally, the effect of different levels of angular density is studied in Table 5. Since 2D H-V is the optimal strategy according to the previous conclusion, it is ultilized to complete the full scale LFs recursively as explained before. All synthesized views are averaged in this evaluation (different from only central views in Table 4). From Table 5 we get the expected decrease of interpolation quality with sparsity.

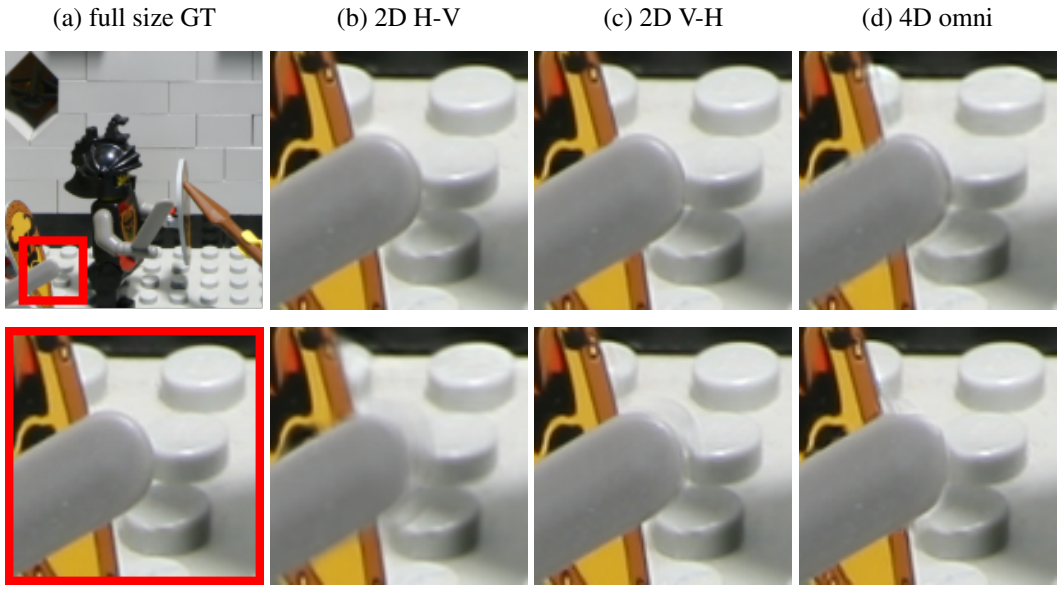
These insights about sparsity vs. quality and the best subsampling and reconstruction strategies can be beneficial for the design of LF coding approaches (maximum quality that can be achieved when omitting views) or camera systems (maximum camera distance for a desired quality and density).

[1] https://v-sense.scss.tcd.ie/?p=4450

 ISBN 978-0-9934207-5-7

Table 5: Comparison between level 1 & 2 & 3 from Figure 4 using 2D H-V

2D H-V	Level 1	Level 2	Level 3
PSNR(dB)	37.34	35.12	31.79
SSIM	0.9886	0.9825	0.9613



(a) full size GT (b) 2D H-V (c) 2D V-H (d) 4D omni

(e) GT (f) 4D diagonal (g) left diag (h) right diag

Figure 6: Visual results of six reconstruction strategies from Figure 3

5 Conclusion

In this paper, we presented a comprehensive study comparing different strategies for efficient light field subsampling and reconstruction. For this purpose we selected an existing view synthesis method among the best performing state-of-the-art techniques. Using this benchmark method, we first evaluate the best subsampling approach among a row-wise, a column-wise, and a checkerboard pattern with a fixed interpolation ratio, which concludes that the row-wise approach offers the best performances. Second, we investigate corner-based central view generation and compare the performance of six possible reconstruction strategies. In addition, we evaluate a multi-stage approach to reconstruct dense light fields from subsampled input with different levels of angular density. We found that using a row-wise followed by a col-wise reconstruction yields best performance. We hope these findings will help inspire researches related to light field subsampling and reconstruction, such as compression and camera array design. Further, a more explicit analysis regarding the relation of the disparity range to the strategy selection could be performed.

References

[INR,] Inria Lytro Illum dataset. `http://www.irisa.fr/temics/demos/lightField/CLIM/DataSoftware.html`. accessed: 26-01-2018.

[Sta,] The stanford light field archive. `http://lightfield.stanford.edu/lfs.html`. accessed: 05-03-2019.

[Alain and Smolic, 2018] Alain, M. and Smolic, A. (2018). Light field super-resolution via LFBM5D sparse coding. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2501–2505. IEEE.

[Alperovich et al., 2018] Alperovich, A., Johannsen, O., Strecke, M., and Goldluecke, B. (2018). Light field intrinsics with a deep encoder-decoder network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9145–9154.

[Flynn et al., 2016] Flynn, J., Neulander, I., Philbin, J., and Snavely, N. (2016). DeepStereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524.

[Honauer et al., 2016] Honauer, K., Johannsen, O., Kondermann, D., and Goldluecke, B. (2016). A dataset and evaluation methodology for depth estimation on 4D light fields. In *Asian Conference on Computer Vision*. Springer.

[Kalantari et al., 2016] Kalantari, N. K., Wang, T.-C., and Ramamoorthi, R. (2016). Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6):1–10.

[Levoy and Hanrahan, 1996] Levoy, M. and Hanrahan, P. (1996). Light field rendering. In *Proc. SIGGRAPH*, pages 31–42.

[Niklaus et al., 2017] Niklaus, S., Mai, L., and Liu, F. (2017). Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270.

[Rerabek and Ebrahimi, 2016] Rerabek, M. and Ebrahimi, T. (2016). New light field image dataset. In *Proceedings of the International Conference on Quality of Multimedia Experience*.

[Rossi and Frossard, 2017] Rossi, M. and Frossard, P. (2017). Graph-based light field super-resolution. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE.

[Shi et al., 2014] Shi, L., Hassanieh, H., Davis, A., Katabi, D., and Durand, F. (2014). Light field reconstruction using sparsity in the continuous fourier domain. *ACM Transactions on Graphics (TOG)*, 34(1):1–13.

[Vagharshakyan et al., 2017] Vagharshakyan, S., Bregovic, R., and Gotchev, A. (2017). Light field reconstruction using shearlet transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):133–147.

[Wang et al., 2018] Wang, Y., Liu, F., Wang, Z., Hou, G., Sun, Z., and Tan, T. (2018). End-to-end view synthesis for light field imaging with pseudo 4DCNN. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 333–348.

[Wanner and Goldluecke, 2013] Wanner, S. and Goldluecke, B. (2013). Variational light field analysis for disparity estimation and super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):606–619.

[Wu et al., 2017a] Wu, G., Masia, B., Jarabo, A., Zhang, Y., Wang, L., Dai, Q., Chai, T., and Liu, Y. (2017a). Light field image processing: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):926–954.

[Wu et al., 2017b] Wu, G., Zhao, M., Wang, L., Dai, Q., Chai, T., and Liu, Y. (2017b). Light field reconstruction using deep convolutional network on EPI. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6319–6327.

[Yoon et al., 2015] Yoon, Y., Jeon, H.-G., Yoo, D., Lee, J.-Y., and So Kweon, I. (2015). Learning a deep convolutional network for light-field image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 24–32.

[Yu, 2017] Yu, J. (2017). A light-field journey to virtual reality. *IEEE MultiMedia*, 24(2):104–112.

Detecting predation interaction using pretrained CNNs

Gabriel Rodrigues Palma[†], Charles Markham[⋆], and Rafael de Andrade Moral[⋆]

[†] *"Luiz de Queiroz" College of Agriculture, University of São Paulo, Brazil*
[⋆] *Maynooth University, Maynooth, Ireland*

Abstract

Identifying ecological interactions is an important task for understanding the complexity of animal communities. Many methods have been used to analyse these types of interactions, which include molecular, chemical and statistical methods. Recent studies have used an interdisciplinary approach into predation, also known as the "Feeding Interaction". This interaction is important for understanding wildlife, because variation in predation contributes substantially to species coexistence in the context of community ecology. Lately, ecologists have started to use large amounts of image data to study different types of animal behaviour. This paper shows that Deep learning (DL) models can act as an important tool for this context. Here we describe the use of a pretrained VGG16 Convolutional Neural Network architecture to detect the predation interaction recorded on images obtained from internet using web-scraping. The Gradient-weighted Class Activation Mapping (Grad-CAM) was applied to provide a method of visualisation of predation events in the image sequences.

Keywords: Deep Learning, Food Webs, Ecological Imaging, VGG16 Architecture.

1 Introduction

In ecology, to understand community complexity and species interactions, a multidisciplinary approach is fundamental [Barbour et al., 2016]. The diversity of methods includes statistical analysis of the species abundance, mathematical modeling of animal interaction, inference about interaction strength, food web analysis, and other tools ([Lajtha and Michener, 1995]; [Zar, 1999];[Baskerville, 2011]). These techniques are commonly used for measuring predation [Lajtha and Michener, 1995]. Predation is extremely important for life existence because variation in predation substantially contributes to species coexistence in the context of community ecology [Ricklefs, 2016]. Several methods can be used to identify predation. For example, biochemical methods use isotopic elements to identify the animal's diet; genetic methods use DNA sequence extraction to identify predators and prey ([Lajtha and Michener, 1995]; [Barbour et al., 2016]). These methods involve expensive analyses including DNA sequencing and biochemical analyses.

Deep Learning (DL) applications in ecology have been showing remarkable results, such as good accuracy rates on classification and prediction tasks [Christin et al., 2018]. They can also be applied to detect animal interactions as well ([Carval et al., 2019]; [Tresson et al., 2019]). To date, the main studies involving deep learning used to detect ecological interactions are based on invertebrates, mainly arthropods. Examples include insect-plant interactions [Pichler et al., 2019] and interactions between different invertebrates such as ants, spiders, cockroaches, slugs and weevils ([Carval et al., 2019]; [Tresson et al., 2019]). To obtain better results relating to the study of community complexity more animal taxa have to be analysed in this context.

This paper has two main objectives: to implement an algorithm able to detect predation in different animal taxa (Carnivorous mammals) and to visualise these predation events. Here, we use a Deep Learning (DL) method for detecting predation, more specifically, a pretrained Convolutional Neural Network (CNN) based on the VGG16 architecture considering the fact that this model was used on previous works for animal detection

 ISBN 978-0-9934207-5-7

[[Simonyan and Zisserman, 2014]; [Chollet, 2018]; [Wani, 2020]]. The Gradient-weighted Class Activation Mapping (Grad-CAM) [[Selvaraju et al., 2019]; [Chollet, 2018]] was used to visualise the features extracted from the images in which predation was detected.

2 State of the Art

Deep learning has been applied in the area of ecology in the past [Christin et al., 2018]. This includes species identification, classification, disease detection, and diversity studies [Christin et al., 2018]. The most used algorithms in this context are CNNs and Recurrent Neural Networks (RNNs) [Christin et al., 2018]. Recently, at the end of 2019, studies showed good DL performances on interaction identification using mainly the YOLOv3 CNN architecture ([Carval et al., 2019]; [Tresson et al., 2019]), Deep Neural Networks (DNNs) and other CNN techniques [Pichler et al., 2019]. Finally, the ecological interactions analysed involve mainly arthropod species such as plant-pollinator interactions [Pichler et al., 2019], competition, cooperation, predation, and scavenging ([Carval et al., 2019]; [Tresson et al., 2019]).

3 Methods

For this paper we compiled a data set containing 170 images (Figure 1) from vertebrates, mainly big carnivorous mammals including foxes, polarbears, tigers, lions and wolves which are classified as *predation* and *non-predation*. To obtain this data set, we used a web-scraping algorithm written in Python to look for images using the Google search engine, with the keywords: "Lion+predation", "Lion", "Wolves+predation", "Polar+bear+predation", "Polar+bear", "Fox+predation", "Fox+animal", "Tiger+predation", "Tiger" and "Predation". These images are of size 150×150, and were used for feature extraction with a pretrained VGG16 CNN architecture ([Simonyan and Zisserman, 2014]; [Chollet, 2018]; [Wani, 2020]). The feature extraction was based on convolution and pooling operations, and the parameters were optimised using the ImageNet data set, given the fact that its collection of animal images provides a generic model for the features involved in this work [Chollet, 2018]. Then, using the feature maps with size 4×4 provided by the pretrained model, we trained two additional densely connected (DC) layers with a dropout rate of 0.5, in an attempt to identify predation interactions correctly (see Figure 2 for an illustration of these operations).

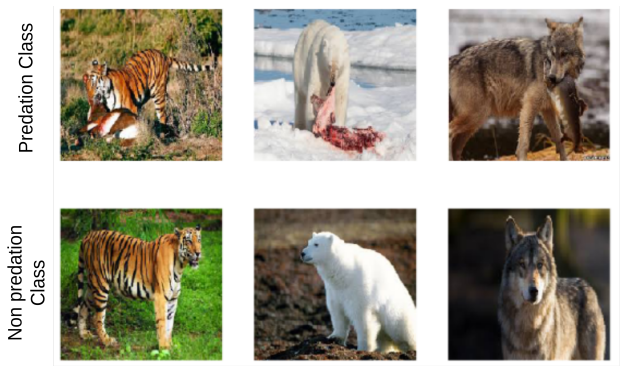


Figure 1: Examples of training data images.

To visualise the feature extraction provided by the VGG16 CNN architecture we used the Gradient-weighted Class Activation Mapping (Grad-CAM) [[Selvaraju et al., 2019]; [Chollet, 2018]], which briefly consists of using the gradient information of the last convolution layer of the standard VGG16 model to obtain important values for each CNN neuron in a particular decision. Here, we used an image of size 150×150 representing the predation interaction and we applied the Grad-CAM to visualise the extracted features. We used the last convolution layer and generated a heatmap containing this gradient information. Finally, we trained the model with 70 images, of which 35 were classified as *predation* and 35 as *non-predation*. For the test set, we used 60 images, of which 30 were classified as *predation* and 30 as *non-predation*. For the validation set, we used 40 images, of which 20 were classified as *predation* and 20 as *non-predation*. To compute the accuracy, recall, precision and F1 of the model, we trained it 30 times using 30 epochs, and we computed the average for each of the parameters values encountered in the 30^{th} epoch for the training and validation set to see the variation provided by the dropout layer. The code of the current methods used in this paper is available at `https://github.com/GabrielRPalma/predation-detection-CNN`.

 ISBN 978-0-9934207-5-7

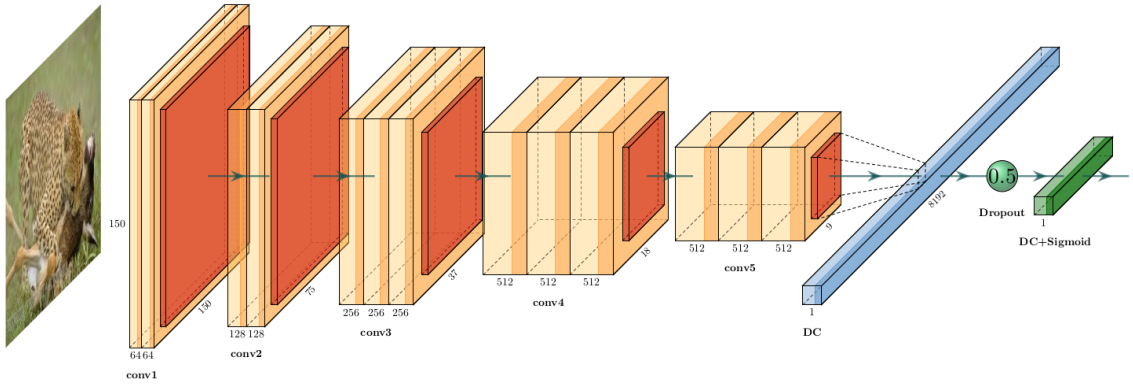


Figure 2: Scheme representing the pretrained VGG16 CNN architecture and the additional layers used to train the model. The convolution, max-pooling and dropout operations are represented in orange, red and green, respectively. Finally, densely connected (DC) layers are added with two activation functions: ReLU before the dropout and sigmoid afterwards.

4 Results and Discussion

Once the VGG16 was trained with the ImageNet data set, it could then be used to apply the Grad-CAM to visualise the most important features extracted by the CNN architecture. We observed that the ImageNet data set provided good features, which allow for detecting predation because in the predator-prey interaction the proximity between these animals is important to classify the image as *predation* (see Figure 3). This feature map visualisation technique is important for ecological studies. It takes into account the capabilities of this method in extracting patterns, which can be used for the understanding of more complex interactions in wildlife. Thus, with the pattern information provided by the chosen architecture, the training section using the added DC layers will contain spatial information that will be fundamental for the classification.

Figure 3: Visualisation of the feature extraction based on the Gradient-weighted Class Activation Mapping (Grad-CAM). The heatmap represents the weighted channels at the feature map by the gradient provided by the class *predation*.

After training the model (Figure 2), our results showed an average training accuracy of 100.0%, we also obtain 100.0% for precision, recall and F1. For the validation data we obtained an average accuracy of 60.0%, precision of 62.8%, recall of 60.0% and F1 of 61.4%. This means that the optimisation process used during the training stage is not converging properly. This is supported by the observation that the validation loss (a measure of the performance of the classifier created using validation data outside the training set) is not reducing at the same rate as the training loss (a measure of classifiers performance using only training data) given the number of epochs. The current number of training samples is not general enough to characterise the problem of interest and this fact is related to the parameter space provided by the training section which requires more data to reach this expected generalization. Therefore, given the small number of images identified on the internet, and that these images are mainly of vertebrate taxa, the work has demonstrated that it would be a challenge to create a general predation detector using this approach. Future collaborative work may provide access to larger datasets. However, this is a pilot study, and as a first work involving predation within different taxa, such as large mammals, our results show an optimistic scenario for improved DL studies of this kind.

5 Conclusions

We presented a different type of application for DL techniques, especially CNNs and the Grad-CAM, in the context of ecological interactions, more specifically, predation. Our results demonstrated the feasibility of detecting this interaction using a pretrained VGG16 CNN architecture. Further work includes attempting to enhance the generalisation of the proposed methods by finding more data for the training set and making changes to the CNN architecture to obtain more accurate results. This can be used by ecologists working with, e.g. camera trap data, and could be expanded to the analysis of video data.

References

[Barbour et al., 2016] Barbour, M. A., Fortuna, M. A., Bascompte, J., Nicholson, J. R., Julkunen-Tiitto, R., Jules, E. S., and Crutsinger, G. M. (2016). Genetic specificity of a plant–insect food web: Implications for linking genetic variation to network complexity. *Proceedings of the National Academy of Sciences*, 113(8):2128–2133.

[Baskerville, 2011] Baskerville (2011). Spatial guilds in the serengeti food web revealed by a bayesian group model citation metadata. *Plos computational biology*, 7(12):1–12.

[Carval et al., 2019] Carval, D., Tixier, P., Tresson, P., Puech, W., PagÃl's, C., Bagny Beilhe, L., and Roudine, S. (2019). Corigan: Assessing multiple species and interactions within images. *Methods in Ecology and Evolution*.

[Chollet, 2018] Chollet, F. (2018). *Deep Learning with Python*.

[Christin et al., 2018] Christin, S., Hervet, E., and Lecomte, N. (2018). Applications for deep learning in ecology.

[Lajtha and Michener, 1995] Lajtha, K. and Michener, R. (1995). Stable isotopes in ecology and environmental science, second edition. *The Journal of Animal Ecology*, 64.

[Pichler et al., 2019] Pichler, M., Boreux, V., Klein, A., Schleuning, M., and Hartig, F. (2019). Machine learning algorithms to infer trait matching and predict species interactions in ecological networks.

[Ricklefs, 2016] Ricklefs (2016). *The Economy of Nature*.

[Selvaraju et al., 2019] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2019). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336âĂŞ359.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*.

[Tresson et al., 2019] Tresson, P., Tixier, P., Puech, W., and Carval, D. (2019). Insect interaction analysis based on object detection and cnn. pages 1–6.

[Wani, 2020] Wani (2020). *Advances in Deep learning*.

[Zar, 1999] Zar, J. (1999). *Biostatistical analysis*.

Extracting Pasture Phenotype and Biomass Percentages using Weakly Supervised Multi-target Deep Learning on a Small Dataset

Badri Narayanan[1], Mohamed Saadeldin[1], Paul Albert[2], Kevin McGuinness[2], and Brian Mac Namee[1]

[1]*School of Computer Science, University College Dublin.*
[2]*Insight Centre for Data Analytics, Dublin City University.*

07-07-2020

Abstract

The dairy industry uses clover and grass as fodder for cows. Accurate estimation of grass and clover biomass yield enables smart decisions in optimizing fertilization and seeding density, resulting in increased productivity and positive environmental impact. Grass and clover are usually planted together, since clover is a nitrogen-fixing plant that brings nutrients to the soil. Adjusting the right percentages of clover and grass in a field reduces the need for external fertilization. Existing approaches for estimating the grass-clover composition of a field are expensive and time consuming—random samples of the pasture are clipped and then the components are physically separated to weigh and calculate percentages of dry grass, clover and weeds in each sample. There is growing interest in developing novel deep learning based approaches to non-destructively extract pasture phenotype indicators and biomass yield predictions of different plant species from agricultural imagery collected from the field. Providing these indicators and predictions from images alone remains a significant challenge. Heavy occlusions in the dense mixture of grass, clover and weeds make it difficult to estimate each component accurately. Moreover, although supervised deep learning models perform well with large datasets, it is tedious to acquire large and diverse collections of field images with precise ground truth for different biomass yields. In this paper, we demonstrate that applying data augmentation and transfer learning is effective in predicting multi-target biomass percentages of different plant species, even with a small training dataset. The scheme proposed in this paper used a training set of only 261 images and provided predictions of biomass percentages of grass, clover, white clover, red clover, and weeds with mean absolute error (MAE) of 6.77%, 6.92%, 6.21%, 6.89%, and 4.80% respectively. Evaluation and testing were performed on a publicly available dataset provided by the Biomass Prediction Challenge [Skovsen et al., 2019]. These results lay the foundation for our next set of experiments with semi-supervised learning to improve the benchmarks and will further the quest to identify phenotype characteristics from imagery in a non-destructive way.

Keywords: Computer vision, deep learning, transfer learning, smart agriculture, data augmentation, weak supervision

1 Introduction

The dairy industry uses clover and grass as fodder for cows. Grass and clover are grown together in fields to improve the consistency of high-quality biomass yield and to reduce the need for external fertilizers. Accurate estimation of the dry biomass percentages of grass and clover species in fields is very important for determining optimal seeding density, fertilizer application and elimination of weeds. Conventionally, this has been done manually by clipping random sample areas in the field and sending it to the lab to visually identify, separate and weigh different species. This manual approach is laborious and time consuming, and gives rise to the need for a more efficient and non-destructive approach for biomass estimation. Machine learning approaches that

directly estimate dry biomass percentages from images have huge potential for addressing this need. The task, however, is challenging. Two types of clover, red and white, are often grown together. Red clover has a shorter life cycle than white clover, but the two species are visually similar and difficult to distinguish [Skovsen et al., 2018]. Moreover, in addition to grass and clover, there is usually presence of undesirable weeds too. The choice of an appropriate machine learning solution should focus on accurately discriminating between the two clover species while accurately predicting the grass-clover-weeds mix ratio.

Deep learning models based on Convolutional Neural Networks (CNNs) have achieved impressive performance across various computer vision applications. This paper proposes a deep learning model that predicts biomass percentages of grass, white clover, red clover, and weeds from canopy view images collected from farms. The Grass Clover Image dataset from the Biomass Prediction Challenge [Skovsen et al., 2019] is used to develop this approach. This dataset poses unique challenges in the form of a multi-target prediction problem (a distribution of biomass yield across species types must be prdicted) and partial labelling with missing values of dry white and red clover in many examples. The training data also consists of just 261 labeled images, which is very small for training a deep model. CNNs can generalize efficiently when trained on large datasets, whereas small datasets pose an optimization problem, thus needing numerous iterations that invariably result in overfitting.

In this paper, we demonstrate an improvement in the state-of-art techniques for biomass prediction by providing an end-to-end approach from pixels to biomass directly from real images. This is in contrast to prevalent multi-step approaches that train on a large amount of synthetic images and a small number of labelled real images. We adapt a VGG-16 model pretrained on Imagenet to the multi-target regression problem of predicting biomass percentages. This adaptation, in combination with data imputation for missing values and weak supervision through differential sample weights for weak labels, does a good job of extracting features. We avoided overfitting through model regularisation and data augmentation to increase the number of training samples, that resulted in improved prediction of biomass for the small dataset.

The remainder of the paper proceeds as follows. Section 2 provides a review of current methods for pasture phenotype estimation, specifically those that apply deep learning techniques. Previous work in multi-target learning for regression problems, transfer learning and weak supervision is also highlighted. In Section 3, we explain the biomass percentages estimation process with the proposed adaptation of transfer learning and weak supervision. The section also covers a sequence of data pre-processing steps required to address the limitations of the dataset. Section 4 describes an evaluation experiment and discusses its results. Finally, in Section 5 we present our conclusions and indicators to our future research.

2 Related work

In this section, we review applications of deep learning in smart agriculture, as well as existing work on multi-target regression, transfer learning, and weak supervision.

2.1 Deep learning for biomass estimation

Kamilaris and Prenafeta-Boldú [2018] present an extensive survey of applications of deep-learning-based computer vision in smart agriculture. These applications include weed identification, land cover classification, plant recognition, fruit counting, and crop type classification, as well as biomass estimation. Larsen et al. [2018] and Skovsen et al. [2018] used aFully Convolutional Network (FCN) architecture [Long et al., 2015] for semantic segmentation to identify plant species in images at a pixel level. Larsen et al. use images captured by unmanned aerial vehicles from two different farms in Denmark to provide pixel coverage of species in an image. Whereas Skovsen et al. provide a two step approach of using 2 FCNs to classify pixels as grass, clover, weeds and soil in a large set of synthetic images, and further use these models on real images with biomass ground truth to predict biomass from the pixel percentages of the individual components through a regression model.

 ISBN 978-0-9934207-5-7

2.2 Multi-target regression

In a survey of multi-output learning, Xu et al. [2019] describe multi-target regression as a type of multi-output learning that predicts multiple real-valued outputs from a set of input features for an example instance. By simultaneously predicting all the targets, the algorithm captures the inherent relationships between the targets themselves in addition to their individual relationships with the input features. According to Borchani et al. [2015], capturing these relationships in the predictions will appropriately reflect real-world problems.

2.3 Transfer learning and weak supervision

Transfer learning between task domains reduces the effort to label training data for the target domain by transferring knowledge from a pre-trained model on a large dataset from a different domain. Pan and Yang [2010] explain the different situations where the source and target domains and tasks are same or different and accordingly describe the three categories of transfer learning: *inductive transfer learning, transductive transfer learning, and unsupervised transfer learning*. Additionally, they describe "what to transfer" through 4 different approaches: *instance transfer, feature representation transfer, parameter transfer, and relational knowledge transfer.* The *inductive transfer* setting where the source and target tasks are different, irrespective of the difference in the domains, resembles our problem. In the problem addressed by this paper, the visual recognition (classification) task of VGG-16 [Simonyan and Zisserman, 2014] approach using ImageNet (source domain) is different to the biomass prediction (regression) task with farm images (target domain). Our approach focused on learning latent features in the farm images by transferring feature representation knowledge from VGG-16, followed by a non-linear regression to predict the biomass from the learnt features.

Small datasets that have missing values can benefit from weakly supervised learning [Zhou, 2018]. Weak supervision can be of three types: *incomplete, inexact, and inaccurate*. Incomplete supervision refers to generating weak labels for a large subset of data that have missing labels because only a small subset can be hand-annotated by human experts. Inexact supervision arises in a scenario where fine-grained labels are difficult to annotate and the dataset has coarse-grained labels only. Inaccurate supervision is the case where the ground truth is largely expected to be imperfect because of various reasons like human error, crowdsourcing to obtain labels, or difficulties in recognizing and categorizing.

In the next section we describe an approach to estimating biomass percentages from images of grass (a multi-target regression problem) using a CNN. To train this CNN we utilise transfer learning and weak supervison as well as data augmentation techniques.

3 Estimating biomass percentages

This paper addresses the problem of estimating biomass yield of grass, clover (red and white) and weeds at different seasons of crop growth directly from real farm imagery. In the process, we tackle some unique challenges. Firstly, we are predicting multiple targets with an overall percentage distribution and a further percentage distribution of sub-targets. Secondly, we deal with a small dataset of real farm images that has missing values. We evaluated and tested our proposed approach the publicly available Grass Clover Image Dataset for the Biomass Prediction Challenge [Skovsen et al., 2019].

Skovsen et al. [2019] describe a baseline 2-step approach to this challenge. The first step trains 2 FCN semantic segmentation models initialized with pretrained weights from VGG16. The FCN models are trained using 1720 synthetic images. The first FCN classifies the pixels in the images into grass, clover, weeds, and soil; while the second FCN identifies red and white clover in them. Using 261 real high-resolution farm images with biomass labels a linear regression model is trained to predict the grass-white clover-red clover-weed dry biomass percentages from the the percentages of pixels of each type identified in the semantic segmentation performed by the two FCNs. The Biomass Prediction Challenge stipulates two metrics for each predicted category, the root mean square error (RMSE) and mean absolute error (MAE) to evaluate the model's performance.

3.1 Dataset

The Biomass Prediction Dataset has 261 training images captured from 3 different dairy farms, with their corresponding biomass compositions, expressed in actual and percentage terms. 174 unlabelled test images are provided along with with their harvest season for the contest evaluation. Each image is taken over a square frame of vegetation measuring $0.5m \times 0.5m$, at ground sampling distances (GSD) of 4-8 px mm^{-1}. The image sizes range approximately within 1800×1800 px to 3000×3000 px.

The metadata associated with each image in the dataset is categorised as basic, semi-advanced or advanced, and the biomass weights are measured over different harvest seasons. The semi-advanced category corresponds to data collected in the 1st seasonal harvest, and the advanced category refers to data collected in the other seasonal harvests. The dry biomass percentages of grass, clover and weeds comprise 100% of the dry biomass for all the 261 training examples. The semi-advanced and advanced categories, totalling 157 examples, additionally reflect the break up for the subspecies of white and red clovers. In this study we did not use seasonal harvest as a distinguishing feature, and so we have grouped semi-advanced and advanced labels together under a single category (advanced), as reflected in Table 1.

Seasonal Harvest No	Basic	Advanced	Grand Total
1	23	37	60
2	33	56	89
3	25	35	60
4	23	29	52
Total	104	157	261

Table 1: Data distribution and labeling. Semi-advanced and Advanced label types have been grouped together under Advanced

3.2 Data imputation for missing label values

Supervised learning requires datasets with complete labels for all data examples in the training set. The dataset, as explained above, is missing label values for the subspecies of white and red clover for all the 104 examples in the basic category, although the total clover biomass percentage is provided. This is a unique problem in a supervised learning setting where sub parts of the ground truth is unavailable and needs to be approximated in order to allow all data examples to be used for training.

Gelman and Hill [2006] describe different imputation techniques for estimating missing values using other available values. Three alternative methods, specifically multiple regression, mean and median value imputations were evaluated in this work for white and red clover label imputation. Models were trained using labels generated from each of the three alternative imputation methods for comparison of test results.

The deterministic regression imputation technique ignores the error term and predicts exact values for the missing cells from the observed data for the corresponding variables. Since there are multiple variables with missing values, regression cannot be directly applied. To work around this, we initially applied random imputation for the missing cells, followed by deterministic regression imputation multiple times. We included the categorical variables related to the harvest season as predictors for the regression.

In the alternative option of imputation using mean values, dry white clover and dry red clover were initially expressed as fractions of dry clover from Advanced categories. The mean values of these 2 fractions were then applied on the dry clover biomass of the basic category examples to calculate the respective proportions. The resulting proportions were then expressed as percentages of total biomass to get the imputed label values for white and red clover percentages respectively. We also explored imputing median values instead of mean as the third alternative.

 ISBN 978-0-9934207-5-7

3.3 Data augmentation

Training deep neural networks on small datasets cause optimization issues and overfitting. The network does not encounter sufficient example images to learn enough features for generalisation. In such cases, Krizhevsky et al. [2012] suggest that image augmentation can enhance the network's generalisation performance by artificially increasing the number of examples in the dataset through label-preserving transformations. In an earlier work, describing some good practices that can be applied to visual document analysis, Simard et al. [2003] explain that when working with datasets containing a small number of images, if transformation invariance properties are inherent in the image parameters, the generalisation performance of a network will improve when we feed the network with additional transformed data. The larger number of images enable the network to learn these invariances better.

For each image in our dataset, we fed the network with 10 modified training examples that are run-time random transformations of the original example image. Some sample transformations are shown in Figure 1. All images were reduced to a standard size of 500×500 px, as against a higher resolution of 1200×1200 px used in the baseline benchmark in the competition. We employed a combination of operations for random transformations that includes rotation range of 15 degrees, zoom range of 15%, 20% width / height shift, shear range of 15%, horizontal flips, and a channel shift range of 50. To prevent loss in the image from rotation and shifts, we used the wrap function for the fill_mode parameter.

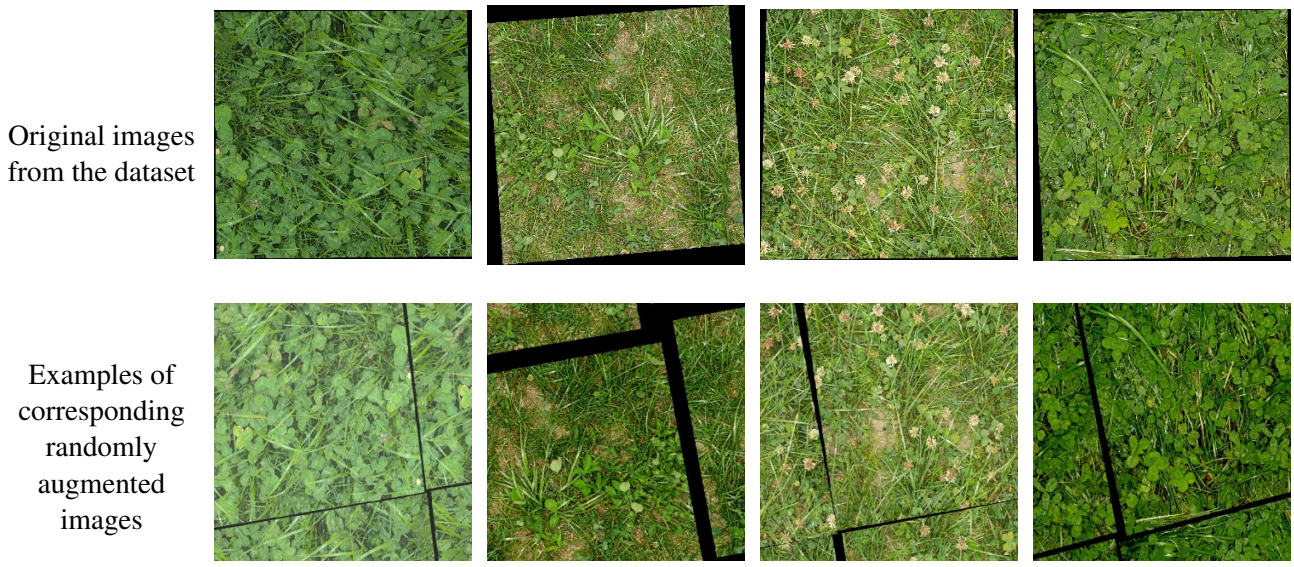


Figure 1: Image augmentation samples. The augmented images are resized to 500×500 px.

A cursory inspection of random images revealed that the grass-clover distribution is not homogeneous within an image. Since the ground truth data points cannot be proportionally divided, we avoided splitting the images into multiple smaller images. A total of 2,090 run-time augmented images per epoch constituted the final training dataset, increasing the training size by 10x.

3.4 Weak supervision

Approximations in the training labels through data imputation to account for scarcity, introduce noise in the ground truth and implies weak labels for certain examples. Such examples are given lesser importance in training by lowering their weights in the calculation of the loss function. This *Incomplete* form of weak supervision [Zhou, 2018] was adopted when training the model with imputed weak labels since they have a higher degree of uncertainty and are less reliable in training. We iterated with different options for the differential sample

weights and determined that 1:1.5 for basic vs advanced gave the best results.

3.5 Model architecture

Transfer learning was employed for feature extraction by initializing the CNN with pre-trained weights from the the well known VGG-16 architecture trained on the ImageNet dataset. We retained the convolutional layers and dropped the last two dense layers that were originally designed for a classification task. The weights of the convolutional layers were made non-trainable. This was to facilitate detection of feature representations by the original architecture in the target dataset. The convolutional layers were followed by 2 dense layers of 4,096 and 256 neurons each, with ReLU activations and 'random_uniform' kernel initialization. Each dense layer was accompanied by a batch normalization. The architecture was completed with an output layer of 4 neurons and a softmax activation to predict the percentages for grass, white clover, red clover and weeds.

The 261 images in the dataset were divided into a training set comprising of 209 images and a validation set with 52 images, which were kept standard across the different models for performance comparison. The 209 images in the training set were subjected to 10 random data augmentations with size 500×500 px at run-time, generating 2,090 different training images per epoch with a mini-batch size of 8. The network was thus never exposed to the original images. The network was trained for 100 epochs using Adam optimizer with a learning rate 10^{-3} and weight decay 10^{-3} / 200. The best weights based on the least validation loss across epochs were stored and used for predictions on the 174 unlabelled images in the test set for competition submissions. Root Mean Squared Error (RMSE) was used as the loss function.

4 Results

The model was trained three times with different sets of imputed labels obtained through mean, median and regression imputations. The results of the validation metrics RMSE and MAE based on each model's best weights for the least validation loss across epochs are compared for the three imputation techniques in Table 2. While it is clear that the median imputation technique yields the best overall result with the lowest MAE of 5.55, we find that the mean imputation technique performs the best in differentiating the white and red clovers with MAE scores of 5.99 and 5.63 respectively, and the overall MAE of 5.64 is comparable with the median technique. Ability to better discriminate the two subspecies of clover is a key objective.

	Grass		Clover		White Clover		Red Clover		Weeds		Overall	
Data Imputation technique	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Regression	8.14	6.84	8.55	6.92	8.24	6.44	9.33	6.80	5.51	3.95	8.06	6.19
Mean	8.00	6.21	7.69	6.16	7.44	5.99	7.33	5.63	5.68	4.20	7.27	5.64
Median	7.45	5.95	7.26	6.04	8.05	6.2	8.4	6.12	4.57	3.42	7.27	5.55

Table 2: Comparison of validation metrics RMSE and MAE for each biomass component as well as the overall performance across different imputation techniques.

The model weights that provided the lowest loss error on the validation set during training were saved and used to generate output predictions from the model on the test-set. Test results obtained from the model for each of the three different runs were submitted to the challenge website for evaluation against ground truth. The challenge organisers evaluate the submitted results independently. RMSE results for (grass, clover, white clover, red clover, and weeds) obtained on the test set using the proposed model with different label imputation schemes are summarized in Table 3. Similarly, MAE results are provided in Table 4.

In general the mean imputation scheme for labels provide better results compared to other schemes. Comparing the proposed model results to the baseline the model obtains lower error for the percentages of grass, clover, and white clover. The proposed model error results for weeds percentages are slightly higher than the baseline. Regarding red clover higher error compared to baseline it is very tricky to distinguish between White and red clover since they look quite similar. Further tuning or even model structure modifications might be required to improve the differentiation between white and red clover.

 ISBN 978-0-9934207-5-7

Table 3: RMSE results on the test dataset for different imputation schemes compared to baseline

Data Imputation	Grass	Clover	White Clover	Red Clover	Weeds
Baseline	9.05	9.91	9.51	**6.68**	**6.49**
Regression	8.98	10.03	8.78	10.46	6.86
Mean	**8.64**	**8.73**	8.16	10.11	6.95
Median	8.67	9.93	**8.09**	9.87	7.73

Table 4: MAE results on test data for different imputation schemes compared to baseline

Data Imputation	Grass	Clover	White Clover	Red Clover	Weeds
Baseline	6.85	7.82	7.61	**4.84**	**4.61**
Regression	6.96	7.94	6.84	7.70	4.80
Mean	**6.77**	**6.92**	**6.21**	7.74	5.02
Median	6.89	8.06	6.23	6.89	5.95

5 Conclusion

This paper describes a deep learning approach to estimating biomass percentages of different plant types grown together in the field. Traditional approaches for estimating biomass percentages are destructive, expensive and time consuming, since they require random samples of the field to be clipped and then manually separated and weighed. In this paper, a prediction approach based on transfer learning utilizing a VGG-16 model pre-trained on Imagenet, while adapting the final dense layers to perform multi-target regression is presented. The proposed scheme benefits from data augmentation, label imputation, and weak supervision to achieve best prediction outputs. The proposed model and training scheme was tested and evaluated on the the Grass Clover Dataset provided by the Biomass Prediction Challenge Skovsen et al. [2019]. Though the training set consists of only 261 images, the proposed model achieved predictions of biomass percentages for grass, clover, white clover, red clover, and weeds with Mean Absolute Error (MAE) of 6.77%, 6.92%, 6.21%, 6.89%, and 4.80% respectively. The proposed model predictions outperform the baseline provided in Biomass Prediction Challenge Skovsen et al. [2019] as it provide lower error for grass, clover, and white clover biomass percentages. The obtained prediction error for red clover and weeds is higher than baseline with a small margin for weeds. The initial choice of VGG16 for transfer learning is arbitrary and provides an early proof of concept. In the future, we plan to experiment with other networks such as ResNet, wResNet, or deeper VGG architectures. We will also investigate more sophisticated weak supervision and semi-supervised learning schemes that take advantage of unlabelled data.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and the Department of Agriculture, Food and Marine on behalf of the Government of Ireland under Grant Number [16/RC/3835] - VistaMilk.

References

Borchani, H., Varando, G., Bielza, C., and Larrañaga, P. (2015). A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233.

Gelman, A. and Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*, chapter 25, pages 529–544. Cambridge university press.

Kamilaris, A. and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147:70–90.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Larsen, D., Steen, K. A., Grooters, K., Green, O., Nyholm, R., et al. (2018). Autonomous mapping of grass-clover ratio based on unmanned aerial vehicles and convolutional neural networks. In *International Conference on Precision Agriculture*. International Society of Precision Agriculture.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Simard, P. Y., Steinkraus, D., Platt, J. C., et al. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Skovsen, S., Dyrmann, M., Eriksen, J., Gislum, R., Karstoft, H., and Jørgensen, R. N. (2018). Predicting dry matter composition of grass clover leys using data simulation and camera-based segmentation of field canopies into white clover, red clover, grass and weeds. In *International Conference on Precision Agriculture*. International Society of Precision Agriculture.

Skovsen, S., Dyrmann, M., Mortensen, A. K., Laursen, M. S., Gislum, R., Eriksen, J., Farkhani, S., Karstoft, H., and Jorgensen, R. N. (2019). The grassclover image dataset for semantic and hierarchical species understanding in agriculture. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Xu, D., Shi, Y., Tsang, I. W., Ong, Y.-S., Gong, C., and Shen, X. (2019). Survey on multi-output learning. *IEEE transactions on neural networks and learning systems*.

Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53.

Sub-Pixel Back-Projection Network For Lightweight Single Image Super-Resolution

Supratik Banerjee[1,2], Cagri Ozcinar[1], Aakanksha Rana[1], Aljosa Smolic[1], and Michael Manzke[1]

[1]*School of Computer Science and Statistics, Trinity College Dublin, Ireland.*
[2]*Rawky Tech LLP, Mumbai, India.*

Abstract

Convolutional neural network (CNN)-based methods have achieved great success for single-image super-resolution (SISR). However, most models attempt to improve reconstruction accuracy while increasing the requirement of number of model parameters. To tackle this problem, in this paper, we study reducing the number of parameters and computational cost of CNN-based SISR methods while maintaining the accuracy of super-resolution reconstruction performance. To this end, we introduce a novel network architecture for SISR, which strikes a good trade-off between reconstruction quality and low computational complexity. Specifically, we propose an iterative back-projection architecture using sub-pixel convolution instead of deconvolution layers. We evaluate the performance of computational and reconstruction accuracy for our proposed model with extensive quantitative and qualitative evaluations. Experimental results reveal that our proposed method uses fewer parameters and reduces the computational cost while maintaining reconstruction accuracy against state-of-the-art SISR methods over well-known four SR benchmark datasets. [1]

Keywords: super-resolution, convolutional neural network, sub-pixel convolution, iterative back-projection

1 Introduction

Single image super-resolution (SISR) is the process of recovering the high-resolution (HR) image from a given low-resolution (LR) image [1]. With the success in signal processing and machine learning, many learning-based SISR methods have been proposed in the literature, demonstrating promising results. Nowadays, these methods can be used in different applications [2, 3] such as medical imaging, surveillance, face recognition, and virtual reality [4].

Given the advances in SISR, it remains a challenge to deploy the most existing SISR models in real-time applications, demanding compact deep neural network architectures. In particular, some emerging applications require faster SISR methods to boost the imaging performance. For example, modern graphic cards can raise a game's frame rates using SISR algorithm [5]. In fact, most of the recent SISR algorithms are based upon very deep neural networks, requiring high number of parameters and computational cost for graphically-intensive workloads [6].

In this paper, we propose a new convolutional neural networks (CNNs)-based SISR method with an objective of factoring minimal reduction in perceptual quality while maintaining computational complexity. We use the previously developed SISR method in [7], and reduce its network parameters by simplifying the back-projection network architecture. For this, we replace the densely connected up- and down-projection units which comprise of several deconvolution and convolution layers by our proposed sub-pixel back-projection (SPBP) block. Experimental results validate the effectiveness of our proposed method in reconstructing accurate SR images. The proposed model requires a small number of parameters and low computational cost against

[1]This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/RP/2776.

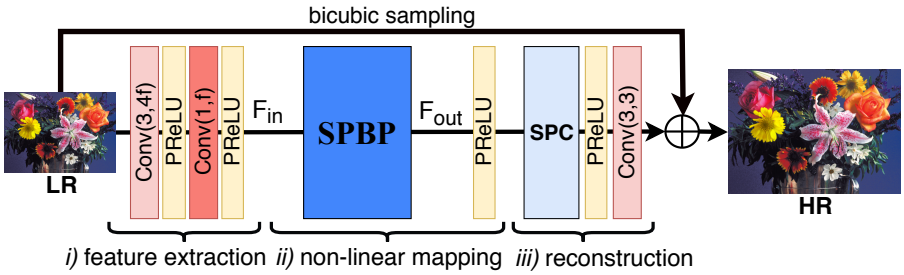


Figure 1: Proposed network architecture for SISR.

several state-of-the-art SISR methods over four well-known SR test datasets. In addition, we demonstrate two smaller variations of our network, SPBP-S (small) and SPBP-M (medium), which use even fewer parameters and has significantly lower computational cost.

The rest of the paper is organized as follows: Section 2 discusses the related CNN-based SISR works. Section 3 explains our proposed SISR model. Experimental results are presented in Section 4. Finally, Section 5 concludes the paper.

2 Related Work

Inspired by the performance improvements obtained by CNNs on computer vision tasks such as image-to-image translation [8], image captioning [9], Dong *et al.* proposed an SRCNN method [10]. This work proposed a three-layer network to learn the mapping between the desired HR image and its bicubic up-sampled LR image. Motivated by SRCNN, many CNN-based research works have been shown to use deeper networks to increase representation power further. For instance, Kim *et al.* [11] presented a very deep SR (VDSR) architecture to significantly improve the SR image reconstruction accuracy with the use of a 20 layer VGG network [12] along with global residual learning. Recently, Haris *et al.* [7] proposed a deep back-projection network (DBPN), which was based on the idea of iterative up- and down- sampling. However, their proposed network uses large filter sizes which increases the number of parameters, leading to higher computational complexity. Ahn *et al.* [13] designed a cascading mechanism on residual networks, which effectively boost the performance with multi-level representation and multiple short-cut connections for learning residuals in LR feature space. Li *et al.* [14] proposed (SRFBN) to improve reconstruction performance while having low parameters to reduce chances of over-fitting using a feedback mechanism, but it increases the computational cost of the network.

Computational efficiency of the neural networks designed for SISR is important. Dong *et al.* [15], for instance, designed an efficient network structure for fast SISR, called fast SR CNN (FSRCNN). With a similar aim, Shi *et al.* [16] proposed an efficient sub-pixel CNN (ESPCN). In their work, pixel shuffle network was used to upscale the image at the final step of the SR process. Even though their network demonstrates real-time performance, it lacks high reconstruction quality due to its architectural simplicity. Recently, a few SR networks [13, 17, 18] have been proposed to have low parameters and low computational complexity, while maintaining state-of-the-art reconstruction performance.

3 Proposed Model

As shown in Fig. 1, our proposed network architecture consists of three main blocks, namely, *i*) feature extraction (FE), *ii*) non-linear mapping (NLM), and *iii*) reconstruction. At the first block, we extract shallow features from the LR image. The second block extracts deeper features using an iterative back-projection technique. The third block up-samples and refines the final SR image. In the following, we present details of each block where convolutions are denoted as $Conv(k, n)$ with k being the filter size and n being the number of filters.

 ISBN 978-0-9934207-5-7

3.1 Feature extraction

The FE block consists of two convolution layers with PReLU as activation layers, similar to the architectures proposed in [7, 15]. The FE block is defined as:

$$F_{in}^0 = C_0^{FE}(I_{LR}), \quad \text{and} \quad F_{in}^1 = C_1^{FE}(F_{in}^0), \tag{1}$$

where $C_0^{FE} = Conv(3, 4f)$ with $C_1^{FE} = Conv(1, f)$, and f is the base number of filters. The low-level representation, F_{in}^0, is obtained from the LR image, I_{LR}, and the refined feature F_{in}^1 is obtained by F_{in}^0.

3.2 Non-linear mapping

Next, we present details about our proposed NLM block, called SPBP. Here, we reduce the computational cost for SISR, building upon and simplifying the back-projection block developed in [7]. This method, called DBPN, proposes the use of densely connected up and down projection units. These units make use of multiple convolution and deconvolution (Dconv) layers to back-project the feature maps, which makes the network computationally expensive.

To reduce the model complexity of DBPN, we propose to replace these up- and down-projection units and their error feedback mechanism with up- and down-sampling layers as SPC and convolution layers. Our inspiration for this new approach of using SPC over Dconv is based on the work of Shi *et al.* [16], where it is described that the SPC layer is $\log_2 r^2$ times faster than Dconv layer in the forward pass. Since SPC operates in LR space on a feature map of size $\left(n, \frac{W}{s}, \frac{H}{s}\right)$ and Dconv layer operates in HR space on a feature map of size $\left(\frac{n}{s^2}, W, H\right)$, where W and H are the dimensions of the input. We can represent the information contained in its feature maps as: $SPC = LR\left(n \times \frac{W}{s} \times \frac{H}{s}\right)$ and $Dconv = HR\left(\frac{n}{s^2} \times W \times H\right)$. The complexity of the layers with a filter size of $k \times k$ and scaling factor s will then be:

$$SPC = O\left(n \times n \times k \times k \times \frac{W}{s} \times \frac{H}{s}\right) \tag{2}$$

$$Dconv = O\left(\frac{n}{s^2} \times \frac{n}{s^2} \times sk \times sk \times W \times H\right) \tag{3}$$

Thus, the number of parameters are:

$$SPC = LR\,(n \times n \times k \times k) \tag{4}$$

$$Dconv = HR\left(\frac{n}{s^2} \times \frac{n}{s^2} \times sk \times sk\right) \tag{5}$$

For the same information retention and computational complexity, as shown in Eqs. (4) and (5). SPC contains larger number of parameters compared to Dconv, and therefore, upholds a higher representation power without adding computational complexity. For this reason, we propose to use SPC in order to reduces network parameters by simplifying the back-projection network architecture. This approach provides higher representation power and achieves an efficient feature mapping.

Figure 2 shows the design of SPBP, which comprises of an exterior and interior unit. The exterior unit is defined as:

$$H_0 = PS(C_{0,0}^{NLM}(F_{in}^1)\uparrow_s, \text{ and } L_0 = C_{0,1}^{NLM}(H_0)\downarrow_s, \tag{6}$$

where $\uparrow_s$, $\downarrow_s$ represent up-sample and down-sample operations respectively with a scale factor s. Also, $C_{0,0}^{NLM}$ represents $Conv(3, fs^2)$, where PS is the pixel-shuffle layer, which defines SPC. The SPBP block takes F_{in}^1, which is the first LR feature map in this block as input and produces an HR feature map, H_0. This is back-projected to a LR feature map L_0 using $C_{0,1}^{NLM}$, which represents $Conv(3, f)$. This is a single group of the proposed SPBP block.

The use of DenseNet [19] has demonstrated the alleviation of vanishing gradient problem. Also, the use of dense skip connections help to generate powerful high-level representations and encourages feature reuse.

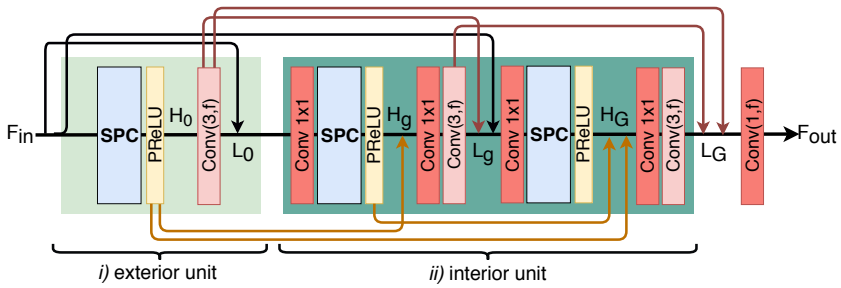


Figure 2: Sub-Pixel Back-Projection Block

Inspired by this, we introduce the use of dense connections in SPBP block, as similar to [7], which forms the interior unit of the block. Thus the interior unit of G groups is formulated as:

$$H_g = PS(C_{g,0}^{NLM}([F_{in}^1, L_0, \ldots, L_{g-1}])) \uparrow_s, \tag{7}$$

$$L_g = C_{g,1}^{NLM}([H_0, H_1, \ldots, H_g]) \downarrow_s, \tag{8}$$

$$F_{out} = C_{out}([L_1, L_2, \ldots, L_G]). \tag{9}$$

where $[F_{in}^1, L_0, \ldots, L_{g-1}]$ refers to the concatenation of F_{in}^1, LR feature maps $0, ..., g-1$ and H_g and is the HR feature map produced by the up-projection layer in the g^{th} group. Similarly, $[H_0, H_1, \ldots, H_g]$ refers to the concatenation of HR feature maps $0, ..., g$ and L_g is the LR feature map produced by the down-projection layer in the g^{th} group. C_{out} is a compression unit representing $Conv(1, f)$ to generate the output F_{out} by fusing LR features from the previous levels $1, ..., G$ of the SPBP block.

3.3 Reconstruction

This block uses a SPC layer which up-scales the LR feature map obtained from the SPBP block. This is followed a convolution layer which refines the up-sampled feature map. The reconstruction layer is defined as:

$$I_0^{Res} = PS(C_0^R(F_{out})) \uparrow_s, \text{ and } I_1^{Res} = C_1^R(I_0^{Res}), \tag{10}$$

$$I_{SR} = I_1^{Res} + f_{UP}(I_{LR}), \tag{11}$$

where I_0^{Res} is the residual upscale of $PS(C_0^R(F_{out}))$ with input F_{out}. I_1^{Res} is the refined residual HR feature map derived from $C_1^R(.)$, which is a $Conv(3, f_{out})$ where, $f_{out} = 3$ is the output feature map "RGB". Inspired by [14, 20] the super-resolved image is constructed by adding the refined HR feature map with $f_{UP}(.)$, which is bicubic up-sample of the LR image. Since the LR image contains abundant low-frequency information [21], this allows the network to bypass the LR information and focus only on the residual component from the HR image.

4 Results

In this section, we first describe our training details, and then we evaluate our proposed SISR method with state-of-the-art SISR methods using quantitative and qualitative experiments.

4.1 Training Details

All experimentation was carried out on ×2 scaling factor between LR and HR. The LR images were obtained by down-sampling HR images from the training set of *DIV2K* [22] dataset with bicubic interpolation. For training, the LR image-crop size was set as 48×48 with 40 random crops per image. The mini-batch size was set to 40 for all network configurations. Each proposed model was trained using the ADAM optimizer with L1 loss

for 1000 epochs, with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The learning rate was initialized as 10^{-4} and decayed by a factor of 2 in every 200 epochs. Image augmentation was was used for training by randomly flipping horizontally or vertically and rotating the training images like [14, 23]. Three different settings for the proposed SISR model, SPBP-S (small), SPBP-M (medium) and SPBP-L (large) use ($f = 16, G = 1$), ($f = 16, G = 10$), and ($f = 32, G = 10$) configurations, respectively. The proposed models have been implemented using the PyTorch library [24]. The training was performed using NVIDIA Titan-Xp GPU with 12 GB memory on Intel core i7-7700 machine.

4.2 Evaluation

To validate our proposed SPBP method, we performed a thorough experimental analysis using nine CNN-based state-of-the-art SISR algorithms: SRCNN [10], FSRCNN [15], ESPCN [16], VDSR [11], DBPN-SS [7], CARN [13], IDN [17], SRFBNs [14], FLSR [18]. As our focus is to develop a lightweight network for SISR, for simplicity, we do not show results for the published networks which are known to have a more complex model than CARN [13]. Each model was tested with four datasets, namely, *Set5* [25], *Set14* [26], *BSDS100* [27], and *Urban100* [28].

In the following, we compare the performance between our proposed methods (SPBP-S, SPBP-M, SPBP-L, SPBP-L+), and state-of-the-art SISR methods using quantitative and qualitative analysis. Similar to other SISR methods [14, 18, 23], we applied the self-ensemble strategy during testing on SPBP-L to further improve the reconstruction performance, we denote this method as SPBP-L+.

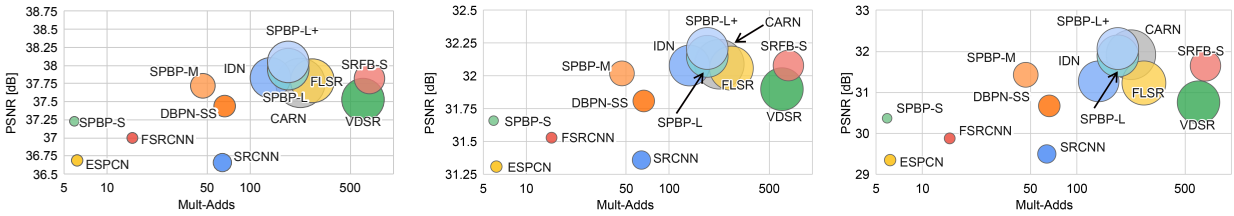
Table 1: Quantitative Results on four datasets. The highest reconstruction accuracy is indicated in **red** and second highest reconstruction accuracy in *blue*. [×2 upscaling]

Methods	**# of parameters**	**Multi-Adds**	**Datasets**							
			Set5		*Set14*		*BSDS100*		*Urban100*	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRCNN [10]	69K	63.8G	36.66	0.9542	32.42	0.9063	31.36	0.8879	29.50	0.8946
FSRCNN [15]	25K	15G	37.00	0.9558	32.63	0.9088	31.53	0.8920	29.88	0.9020
ESPCN [16]	26K	6.17G	36.69	0.9547	32.50	0.9076	31.31	0.8882	29.35	0.8937
VDSR [11]	665K	612.6G	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140
DBPN-SS [7]	109K	66.2G	37.44	0.9589	33.03	0.9127	31.81	0.8951	30.67	0.9128
CARN [13]	960K	223.7G	37.76	0.9590	33.52	0.9166	32.09	0.8978	*31.92*	0.9256
IDN [17]	591K	138.3G	37.83	0.9600	33.30	0.9148	32.08	0.8985	31.27	0.9196
SRFBNs [14]	282K	679.7G	37.82	0.9598	33.38	0.9155	32.08	0.8983	31.65	0.9232
FLSR [18]	717K	271.4G	37.79	0.9595	33.16	0.9143	32.06	0.8983	31.723	0.9183-
SPBP-S	24K	5.9G	37.23	0.9577	32.85	0.9109	31.66	0.8930	30.37	0.9091
SPBP-M	159K	46.6G	37.72	0.9593	33.33	0.9151	32.02	0.8975	31.43	0.9211
SPBP-L	629K	184G	*37.95*	*0.9603*	*33.54*	*0.9171*	*32.15*	*0.8994*	31.89	*0.9262*
SPBP-L+	629K	184G	**38.05**	**0.9606**	**33.62**	**0.9178**	**32.21**	**0.9001**	**32.07**	**0.9277**

4.2.1 Quantitative

We measured the performance of each method for its reconstructed accuracy of the SR image using PSNR and SSIM. Here, similar to previous works [11, 23], we cropped 2 pixels near image boundary and estimated quality scores using only the luminance channel (Y) of images. Also, we measure the computational complexity in terms of the number of operations with Mult-Adds, which is the number of composite multiply-accumulate operations. Table 1 compares the performance of the proposed SPBP-S, SPBP-M, and SPBP-L models with state-of-the-art methods in terms of # of parameters, computational complexity, and objective quality metrics.

We also examined the computational complexity of our model in comparison to other state-of-the-art methods concerning PSNR over the datasets. Fig. 3 shows trade-off between reconstruction accuracy (in terms of



(a) *Set5* [25] (b) *BSDS100* [27] (c) *Urban100* [28]

Figure 3: Trade-off between reconstruction accuracy versus number of operations and parameters on three datasets. The x-axis and the y-axis denote the Multi-Adds and PSNR [dB], and the size of the circle represents the number of parameters. The Mult-Adds is computed for HR image of size 720p.

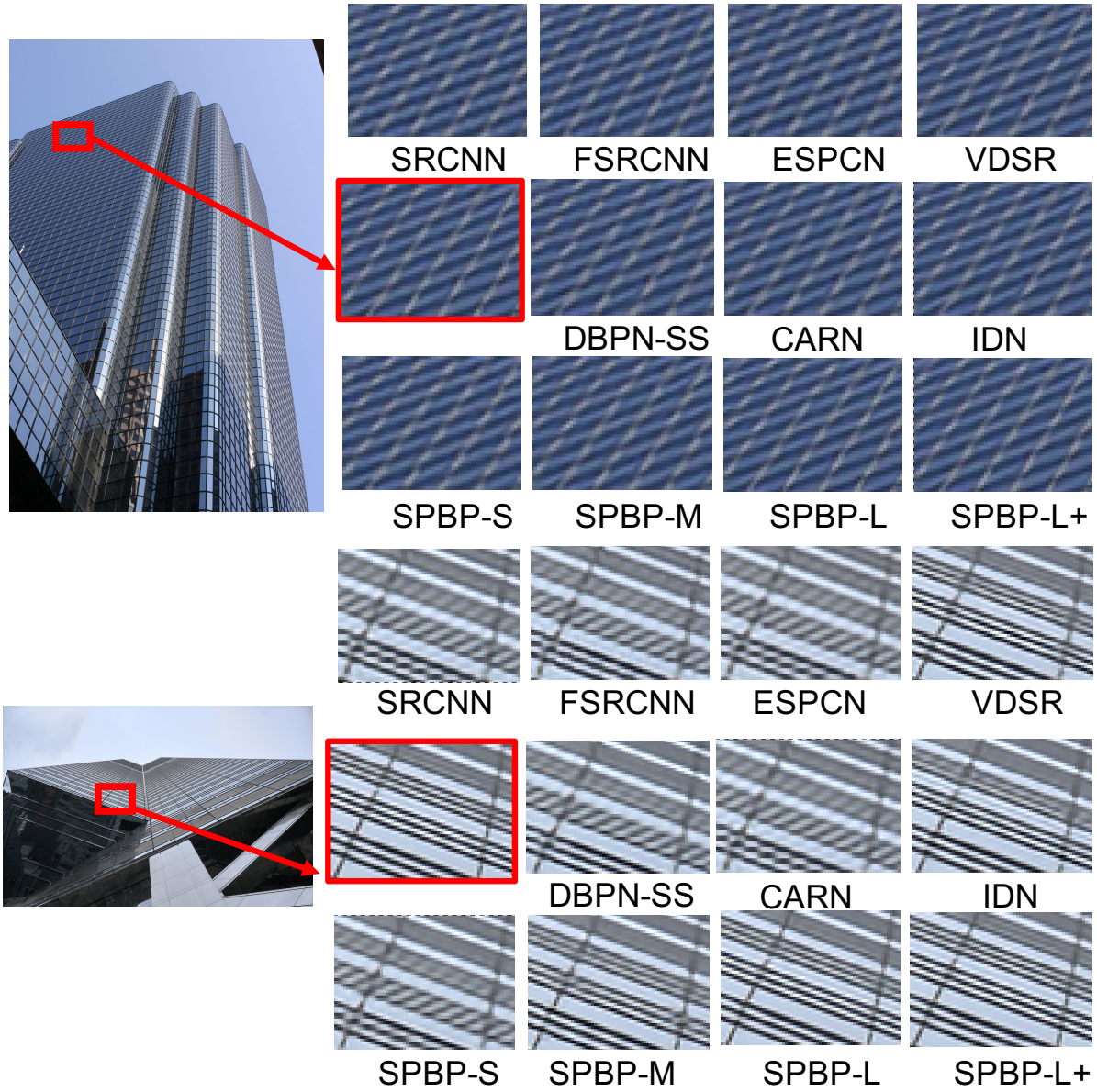


Figure 4: Qualitative comparison of our SPBP models with other works on “img_074" and “img_059" example images from the *Urban100*.

 ISBN 978-0-9934207-5-7

PSNR) versus number of operations and parameters over three datasets: (a)-*Set5*, *BSDS100*, and *Urban100*. In the experiment, the calculations were performed for HR image of size $720p$ (1280×720). Looking at the results, we see that our proposed models (SPBP-S, SPBP-M, SPBP-L, and SPBP-L+) outperform state-of-the-art methods in terms of PSNR for comparable parameter size and has a much lower computational cost.

Overall, our SPBP-L+ model, which has nearly 629K parameters, shows the best reconstruction accuracy performance in most of the benchmark datasets in terms of objective quality scores. Further, we observe that SPBP-M which has only 159K parameters performs very close in most of the benchmark datasets to FLSR, SRFBN, IDN and CARN, all of which have about double or more parameters. Comparing models with less than 100K parameters, we can clearly see SPBP-S outperforms all existing models (SRCNN, FSRCNN, ESPCN). These results prove that our developed models handle the image feature better than the other state-of-the-art methods with fewer parameters and lower computational complexity.

4.2.2 Qualitative

To provide qualitative visual comparison between methods, Fig. 4 shows some examples of reconstructed images from the *Urban100* dataset. We see that the proposed model can construct HR images with higher quality, compared to most of the state-of-the-art methods. Also, we observe that the proposed models have visually similar or better results compared to other state-of-the-art networks, such as CARN, IDN, VDSR, but with lower parameters and computational expense. Especially, the proposed SPBP models construct high frequency patterns with subjectively closer to the original HR.

5 Conclusion

In this paper, we proposed a novel sub-pixel convolution-based dense iterative back-projection network architecture for single-image super-resolution tasks. We showed the reconstruction accuracy and computational efficiency of employing our proposed models (SPBP-S, SPBP-M, SPBP-L) in terms of model parameters, quantitative quality measures (in terms of PSNR, SSIM), and qualitative evaluations. We also compared our proposed model with nine state-of-the-art SISR methods over well-known SR datasets and demonstrated that our proposed approach provides lower computational complexity while maintaining high reconstruction performance. This can be very well observed with SPBP-S which stands out to be the best performing network under 100K parameters.

References

[1] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine (SPM)*, vol. 20, no. 3, pp. 21–36, May 2003.

[2] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang, "Single-image super-resolution: A benchmark," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 372–386.

[3] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," *arXiv cs.CV 1902.06068*, Feb 2019.

[4] C. Ozcinar, A. Rana, and A. Smolic, "Super-resolution of omnidirectional images using adversarial learning," in *IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, 2019.

[5] Andrew Edelsten, "Nvidia dlss: Control and beyond," `https://www.nvidia.com/en-us/geforce/news/dlss-control-and-beyond/`, February 2020.

[6] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia (TMM)*, 2019.

[7] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 1664–1673.

[8] A. Rana, P. Singh, G. Valenzise, F. Dufaux, N. Komodakis, and A. Smolic, "Deep tone mapping operator for high dynamic range images," *IEEE Transactions on Image Processing*, vol. 29, pp. 1285–1298, 2020.

[9] Koustav Ghosal, Aakanksha Rana, and Aljosa Smolic, "Aesthetic image captioning from weakly-labelled photographs," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[10] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 38, no. 2, pp. 295–307, Feb 2016.

[11] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[13] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, Eds., Cham, 2018, pp. 256–272, Springer International Publishing.

[14] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[15] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham, 2016, pp. 391–407, Springer International Publishing.

[16] W. Shi, J. Caballero, F. HuszÃąr, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1874–1883.

[17] Zheng Hui, Xiumei Wang, and Xinbo Gao, "Fast and accurate single image super-resolution via information distillation network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] Y. Shi, S. Li, W. Li, and A. Liu, "Fast and lightweight image super-resolution based on dense residuals two-channel network," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 2826–2830.

[19] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2261–2269.

[20] D. Korobchenko and M. Foco, "Single image super-resolution using deep learning," `https://gwmt.nvidia.com/super-res/about`, June 2017.

[21] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *The European Conference on Computer Vision (ECCV)*, Cham, 2018, pp. 294–310, Springer.

[22] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 1122–1131.

[23] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 1132–1140.

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Workshop Autodiff Submission in Conference on Neural Information Processing Systems (NIPS-W)*, 2017.

[25] Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma, "Image super-resolution via sparse representation," *Trans. Img. Proc.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[26] Roman Zeyde, Michael Elad, and Matan Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*, Jean-Daniel Boissonnat, Patrick Chenin, Albert Cohen, Christian Gout, Tom Lyche, Marie-Laurence Mazure, and Larry Schumaker, Eds., Berlin, Heidelberg, 2012, pp. 711–730, Springer Berlin Heidelberg.

[27] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, vol. 33, no. 5, pp. 898–916, 2010.

[28] J. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5197–5206.

Tagged-ICP: An Iterative Closest Point Algorithm with Metadata Knowledge for Improved Matching of 3D Protein Structures

Peter Ankomah, Peter Vangorp, Ardhendu Behera, and Yonghuai Liu

Edge Hill University

Abstract

Three-dimensional shapes are important in representing physical objects in digital form. This digital representation is useful in applications in numerous fields including chemistry, biology, and engineering. The benefits in analyzing and processing such 3D digital data have given rise to vast amounts of available 3D data and related applications.

In recent times, techniques for determining the functional and structural relationships amongst proteins consider the whole 3D structure (spatial coordinates), proceeding from earlier techniques that were based on sequence information. However, techniques considering the 3D structure of all atomic positions of the protein are too demanding for fast similarity searches especially when the protein is made up of very large numbers of atoms.

Iterative Closest Point (ICP) is the standard algorithm for performing 3D shape matching tasks. ICP has several issues that can affect the process of 3D shape matching. These include the need for an initial transformation to ensure an optimal match, inability of algorithm to converge when rotations are large, and computational cost and complexity of the distance calculation.

We propose an improvement of the ICP algorithm, Tagged-ICP for matching 3D protein structures that takes into consideration known feature descriptions of the points. The search for correspondence in our algorithm matches atoms based on their meta data (atom types), making this approach more meaningful. Our algorithm also reduces the number of distance calculations by a factor depending on the partition. The neighbourhood information also increases the partitions and reduces the size of the search space even further.

Our experimental results based on the publicly accessible Protein Data Bank show that matching becomes inherently meaningful and the complexity of the distance calculation is reduced. Our results also demonstrate improvements in speed, accuracy and convergence on larger rotations over the standard ICP algorithm.

Keywords: Iterative Closest Point (ICP), 3D point cloud, 3D shape matching, 3D protein structures, Atom types

1 Introduction

Background and motivation Comparing shapes is basic to many core problems including matching MRI scans in medical imaging [Nabavi et al., 2000], developing complete objects from partial scans [Levoy et al., 2000] and studying similarity, molecular design, and protein docking in chemistry [Axenopoulos et al., 2016]. Devices and applications allowing 3D scanning have become ubiquitous [Yuan et al., 2016] and the benefits (e.g., high precision) in analysing 3D data have resulted in the emergence of new applications in different fields, processing these vast amounts of available 3D data. In fact, the number of 3D protein structures in the Protein Data Bank (PDB) [Berman et al., 2000] doubles every 18 months, requiring more advanced methods for organising the data [Holm and Sander, 1998].

In the field of molecular biology, comparing structures is used to investigate the functions and interactions between different molecules such as proteins and nucleic acids [Angaran et al., 2009]. Comparing structures of proteins is also fueled by the accepted principle that the 3D structure of a protein is linked with its function [Kinoshita and Nakamura, 2005]. Molecular biology projects have produced a vast number of 3D structures that have enabled the discovery of functions of proteins [Berman et al., 2000; Ellingson and Zhang, 2012].

 ISBN 978-0-9934207-5-7

Several approaches have been used to detect structural and functional relationships amongst proteins. The earliest algorithms were based on sequence information [Schmitt et al., 2002] such as comparing pairs of amino acids (molecules) in a protein structure [Needleman and Wunsch, 1970]. 3D protein comparisons require more sophisticated algorithms to capture, visualise and match the structures based on spatial coordinates. 3D Proteins can be represented as rigid objects and a transformation found to align them, however, more reliable matches would be attained if the coordinates are associated with some predefined properties [Schmitt et al., 2002] such as the atom types.

An improved shape matching algorithm can benefit this problem. A key computer vision problem is to best align two shapes by minimizing the distance between the source and target shapes in order to determine the extent of similarity or dissimilarity. This technique is used in tasks such as facial and fingerprint recognition, machine vision, assistive and automotive technologies [Burlacu et al., 2016].

The Iterative Closest Point algorithm (ICP) [Besl and McKay, 1992; Chen and Medioni, 1992] is a well-known and dominant shape matching algorithm [Attia and Slama, 2017] because of its simplicity and straight forwardness. The algorithm is very effective at registering (aligning) point clouds because of its speed and accuracy [Attia and Slama, 2017]. ICP iteratively registers a data point cloud to a model point cloud with the aim of best aligning them, whether or not they partially or fully overlap. The registration method of the algorithm finds a transformation that reduces the misalignment (distance) between the point clouds until a defined error threshold or the maximum number of iterations is met [Donoso et al., 2017a].

Several studies have used ICP to compare structures of proteins by representing them as point clouds of atoms [Weskamp et al., 2004; Shulman-Peleg et al., 2008].

Limitations of existing solutions ICP requires a good initial transformation to ensure that point clouds converge at an acceptable minimal. The algorithm may also not converge at all from a particular transformation [Besl and McKay, 1992]. In fact, larger transformations have actually been found to decrease the efficiency of the algorithm [Attia and Slama, 2017]. The presence of outliers (non-uniform points) can also affect alignment with ICP [Chen and Belaton, 2014]. Another notable problem is that ICP performs well in some data sets and context and worse in the others [Donoso et al., 2017b].

Overview This paper discusses the standard Iterative Closest Point algorithm and evaluates its performance against our variant on matching 3D protein data sets. Related works are discussed in Section 2. Section 3 details the analysis and implementation of our novel algorithm, Tagged-ICP that improves the matching of 3D proteins by considering metadata knowledge of the atoms. The section details how our algorithm also reduces the search space by partitioning the space according to atom types. Section 4 details the experimental setup and design as well as the results, analysis, and implications. Section 5 concludes this paper with the reiteration of the problems with ICP, the solution our algorithm provides, and our ongoing and future work.

2 Related Work

The standard ICP algorithm [Besl and McKay, 1992; Chen and Medioni, 1991] repeatedly computes the transformation that aligns a data point cloud and a model point cloud and applies this transformation to the data point cloud until it reaches a set error threshold or the maximum number of iterations.

Chen and Medioni [1991] and Besl and McKay [1992] independently published similar methods for creating a complete 3D model from a physical object using point clouds captured using 3D scanning from different angles of the object. The distance from each point in the data point cloud to each point in the model point cloud is computed to establish corresponding points. A set of transformations are then computed to register (align) these surfaces in an iterative way. The shapes are registered when the root mean square error (standard deviation of the distances between the two clouds) is acceptably small.

The ICP algorithm is designed to always converge. It also does not require the use of any extracted features or meta-data about the points in the cloud.

2.1 Algorithmic Improvements

Several variants of the algorithm have been developed to improve one or more aspects of the algorithm [Phillips et al., 2007]. Some of the improvements include reduction in overall computational cost, smaller mean square error, faster convergence speed and optimal selection of points for overall algorithm efficiency. The rise in the development of these variants has been fuelled by the simplicity and effectiveness of the ICP algorithm itself. Mora et al. [2016] provide detailed surveys of many variants and their improvements to the ICP algorithm.

Algorithm 1 Iterative Closest Point algorithm as formulated by Besl and McKay [1992].

function ITERATIVE CLOSEST POINT(P, X)
  Input: data and model point cloud P, X
  Initialisation: $P_0 \leftarrow P$
  for iteration $k := 0$ to $k_{\max}$ **do**
    closest points Y_k $\leftarrow$ CLOSEST POINT SEARCH(P_k, X)
    transformation M_k, MSE d_k $\leftarrow$ REGISTRATION(P_0, Y_k)
    $P_{k+1} \leftarrow$ TRANSFORM(M_k, P_0)
    if change in MSE $d_{k-1} - d_k <$ threshold **then**
      terminate
  return P_{k+1}, M_k, d_k

2.2 Use of Additional Meta-data

Some improvements to the ICP algorithm enable the use of known meta-data in the form of features and constraints (e.g., colors and labels) to make the search for correspondence between points more meaningful. Some features such as global orientation information can change when transformations are applied to them, as such, there is always the need to factor them when calculating the mean square error. On the other hand, invariant features such as colors, labels and local point positions for rigid shapes are not affected by transformations. These invariant features can also be used to improve the registration [Combes and Prima, 2009] by reducing the search space for correspondence [Sharp et al., 2002].

Schutz et al. [1998] proposed a multi-feature ICP variant that includes the color and global orientation information in the distance computation. the research showed that convergence has been considerably improved with the addition of the feature information. Similarly, Thirion [1996] demonstrated an ICP variant using geometric invariant feature points. The research showed that, for a rigid object, the relative positions of the points (local orientation) can serve as invariant feature points because their positions relative to the rigid shape do not change when transformations are applied.

The development of machine learning algorithms has seen their application to the registration problem [Aoki et al., 2019]. Wang and Solomon [2019] proposed a learning-based registration method based on the idea of deep learning. The method takes two point clouds and is able to predict the rigid transformation to align them. Having trained the machine learning model with ModelNet (a custom-built large-scale 3D CAD model dataset) [Wu et al., 2015], the algorithm was found to outperform ICP in terms of efficiency. This is due to the algorithm predicting a rigid transformation in a single pass compared to the iterative classical ICP.

2.3 Applications

ICP is appropriate for aligning protein surfaces which are often not similar [Axenopoulos et al., 2013]. This research focused on extracting features of the shape that were rotation invariant to aid in matching the protein surfaces for protein docking. ICP has also been extensively used for matching biological structures such as measuring the spatial structural similarity of biological data such as proteins. Lu et al. [2016] used ICP for alignment refinement after applying a data reduction method to reduce the missing residues when matching two proteins that are not identical. This two-stage process ensured that the matching was less sensitive to noise and ICP was able to be used effectively. Similarly, ICP was used in the comparison of binding sites on proteins for drug discovery [Bertolazzi et al., 2010]. The research overcame the problem of ICP needing a good initial transformation by formulating a continuous global optimization algorithm that iteratively updated random points in a cluster based on the worst matched point of the cluster. Using sample protein data taken from PDB [Berman et al., 2000], Bertolazzi et al. [2010] proposed a method to detect similarities in protein binding sites when such similarities actually exists. Ellingson and Zhang [2012] developed an ICP based algorithm for superimposing

and comparing protein binding-sites by representing protein atoms as point clouds, with atoms having descriptor labels of their properties to aid in the matching.

Other geometric information was also used to improve the precision of the algorithm including multiple initial local alignments derived from adapting 3D Delaunay triangulation [Cignoni et al., 1998]. Zhou et al. [2014] adapted ICP to group a 3D representation of the activities of bio-molecules. The research used ICP for computing the structural distances and another alignment method to compute the chemical distances. These distance measures were then combined and clustered.

3 The Tagged ICP Algorithm

Our algorithm is based on the basic ICP algorithm (see Algorithm 1). To establish correspondences, ICP has to compute the distances between each point of the model point cloud and each point of the data point cloud. This is an expensive and critical step that becomes a performance bottleneck as the number of points increases. Besl and McKay [1992] proposed using binary search data structures such as *k*d-tree to reduce the complexity of the distance calculation process.

In order to improve this expensive process, our algorithm uses known metadata to optimize the search for correspondences and to make it more meaningful. This optimisation reduces the search space because it only computes the distance between two points with the same metadata information which we call the *tag*. The tag consists of the point's own atom type (carbon, hydrogen, etc.) as well as the atom types of its k nearest neighbours, unordered. In the pre-processing step, the model cloud is partitioned, grouping atoms with identical tags. During the ICP algorithm, the closest point search only searches the partition with the same tag as the atom in the data shape. All results in this paper use $k = 3$. Note that this improvement can be combined with other optimisations of the inner **for** loop such as the *k*d-tree search suggested by Besl and McKay [1992].

Algorithm 2 Tagged Iterative Closest Point algorithm. The metadata of the protein is used to speed up the search for correspondence and improve the registration accuracy: The model point cloud X has been partitioned by tag and only the partition corresponding to the tag of p is searched.

function CLOSEST POINT SEARCH(P_k, X)
  Input: data and model point cloud P_k, X
  Initialisation: closest points $Y_k \leftarrow$ empty list
  for all points p in P_k **do**
    closest point distance $d_{\min} \leftarrow \infty$
    closest point $y \leftarrow$ null
    for all points x in $X_{p.\text{tag}}$ **do**
      distance $d \leftarrow$ DISTANCE(p, x)
      if $d < d_{\min}$ **then**
        closest point $y \leftarrow x$
        closest point distance $d_{\min} \leftarrow d$
    append closest point y to closest points Y_k
  return Y_k

3.1 Computational Complexity

Given a data point cloud P with N_P points and a model point cloud X with N_X points, the computational complexity of a naive Closest Point Search algorithm is $O(N_P N_X)$. The Tagged Closest Point Search has the same computational complexity but it reduces the number of distance calculations by a factor depending on the partitioning. Given fractional partition sizes $f_{P,t}$ and $f_{X,t}$ for each point cloud and each tag t such that $\sum_t f_{P,t} = 1$ and $\sum_t f_{X,t} = 1$, the number of distance calculations is reduced by a factor $F = \sum_t f_{P,t} f_{X,t} \leq 1$. The factor F would be smallest for a larger number of equal-size partitions. However, the fractional partition sizes in our dataset are given by the abundances of the atoms and their neighbourhoods. If the tag consisted only of the point's own atom type and omitted the neighbourhood information, the computation time would only be reduced by a factor of $F \approx 0.45$ in practice. The neighbourhood information increases the number of partitions and reduces their size to lower F further.

4 Experiments and Metrics

Our algorithm was implemented in C#. We also made use of Unity3D [Unity Technologies, 2019] as the IDE to develop the algorithm and visualise and inspect the shape matching results (Fig. 1).

 ISBN 978-0-9934207-5-7

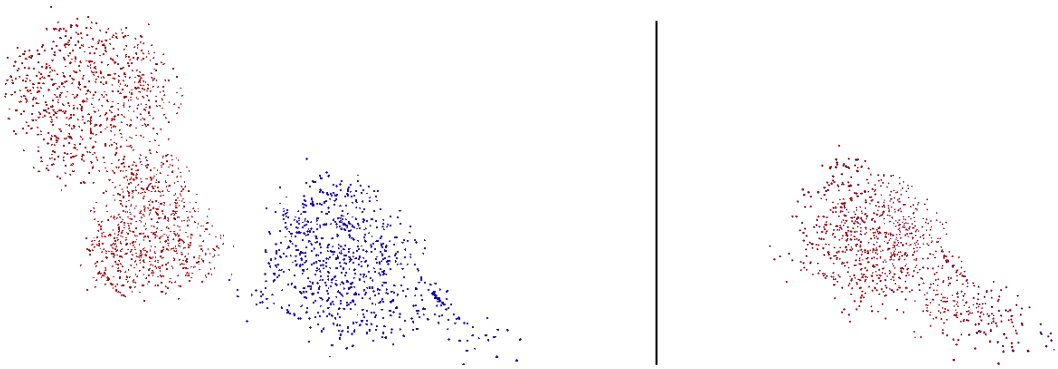


Figure 1: Same shape PDB ID: 2HAX with data shape at angle 30 degrees before match, and after Tagged-ICP match

The experiment setup allows a PDB (Protein Data Bank) [Berman et al., 2000] file to be uploaded using Cell Unity [Gehrer, 2015]. Cell Unity allows spatial information of a PDB file to be extracted and represented as a structure similar to the ball-and-stick model. For each protein dataset, the PDB ID was entered into Cell Unity and the application retrieved the PDB flat file and automatically generated a 3D rigid object representation of the protein based on the spatial coordinates of the atoms in the file. The atoms are represented by spheres and the spheres are able to show the 3D position of the atoms in the protein molecule. We then set our model and reference objects as intended for the generated shapes in our Unity application. The experiment was designed to match sets of the same and different protein structures. Three protein structures with PDB identifiers 2JKF (shapes with mutations), 2JKG (shapes with noise), and 2HAX (matching same shapes) were used in the experiment. For each test, the shapes were aligned along their centers of mass. Data shapes were then rotated at different angles (30, 60, 90, 120, 150 degrees) along the x-axis and the algorithms were run with 100 iterations. The aim of the experiment was to compare the performance in terms of the convergence and match quality of the Tagged-ICP to the original ICP algorithm over different initial rotation angles, noise levels, and when attempting to match two mutations of the same protein. Large initial rotation angles are known to cause difficulties for ICP algorithms. Noise added to the positions of the atoms simulates inaccuracies in protein measurement methods. Mutations are a common use case for matching the identical functional parts and highlighting the differences caused by the mutation. The standard ICP algorithm was chosen because we wanted to measure the effect of only our improvements. The success of this improvements means they can be incorporated into other variants for further experiments on 3D proteins. Using other variants of ICP meant that their improvements will also affect the convergence and match quality and we would not have measured our improvements accurately. We also did not place much emphasis on the translation vector because we aligned the shapes along their centers of mass, meaning they become reasonably close. In order to compare and analyse the performance of the matching of standard ICP with Tagged-ICP, we show two types of graphs.

The convergence graphs represent the root mean squared error of the alignment of the two shapes at different particular iterations. This data (iteration, mean square error) is generated in real time after each iteration. This allows us to understand which algorithm converged faster (iterated less to converge) in terms of the iteration count at convergence and not the time in seconds. The match quality graphs represent the cumulative percentage of points with their mean squared error when the shape matching is complete. This data (point, alignment error) allows us to understand the effective match quality based on the error of the cumulative sum of points. A higher cumulative percentage of points with a lower mean square error implies a better quality of match than a higher percentage of points with a higher mean square error. These graphs provide insights into the behaviours of different techniques: how fast and smoothly they evolve from one iteration to another. Each algorithm uses its own implementation of the closest point search. However, for a fair comparison of match quality, ICP finally runs Tagged-ICP's closest point search method, implying that the match quality graphs for ICP are determined by its registration process only. This is done to identify distortions between ICP's convergence and the quality of match for the different implementations. For instance, from these graphs we can understand that a bad or good convergence graph for ICP shows whether the match quality graph was improved because of Tagged-ICP's

 ISBN 978-0-9934207-5-7

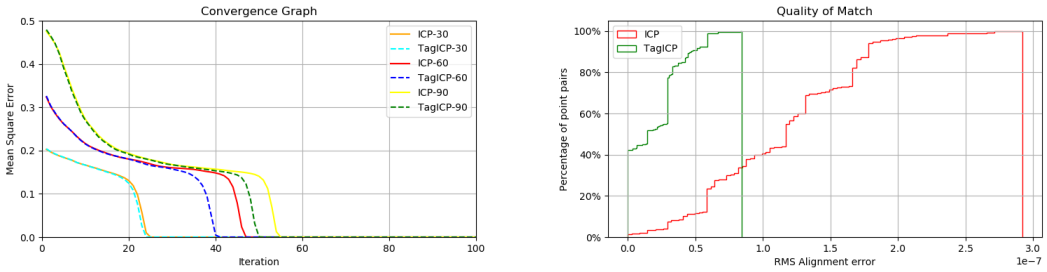


Figure 2: Same shape PDB ID: 2HAX. Convergence at angles 30, 60 and 90 degrees rotation about the x axis, and final match quality at 90 degrees

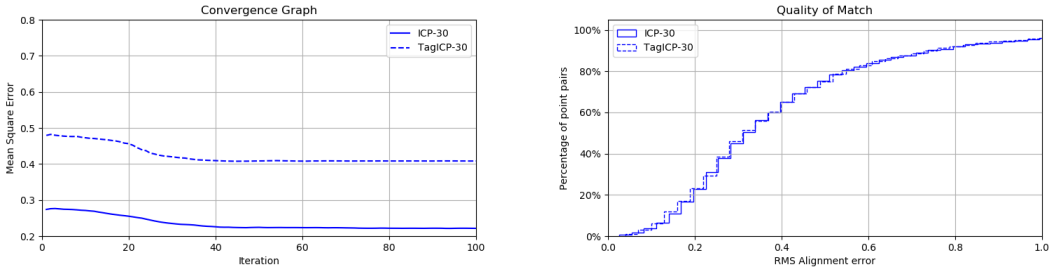


Figure 3: Mutation PDB ID: 2JKF and 2JKG. Convergence and final match quality

closest point search being used. Generating the data at these different phases allows us to study the progress of the convergence as well as the final alignment error for each correspondence.

5 Performance

Fig. 2 shows that for identical shapes, increasing angles result in a higher number of iterations required before convergence, as expected. We also see that Tagged-ICP performed better by convergence with fewer iterations than Standard ICP. Fig. 3 shows that for the mutations protein dataset (Same protein with the mutant one having single to many changes in the molecule in terms of the numbers of atoms and their positions), ICP had a better convergence quality. We can also deduce from the noise experiments (Fig. 4) that the final mean squared errors are related to the noise levels because, for each algorithm, the errors at convergence are directly proportional to the noise level so increasing noise levels shows convergence at higher mean squared error, as expected. On the other hand, the final converged errors do not correspond to the match quality graphs for ICP. That is because the convergence graphs use the algorithm's own closest points search method which gives ICP an unfair advantage . The match quality graphs show the fair comparison using the same, more meaningful closest point search method. This is because Tagged-ICP's closest points search method was used in a final iteration of the ICP which makes use of known feature descriptions to get a meaningful closest point search. Tagged-ICP's closest point search method was used in the final iteration for Standard ICP's match quality in order to compare the differences in the estimated transformations.

Similarly, the graphs show that the convergence speed is related to angle and not to the noise. We can see from the convergence graphs that the higher angles converge at much higher iteration. On the other hand, this pattern is not seen in the noise graphs. This is because the large angles of rotation make the data shape far away from the reference and thus make them differ significantly whilst the noise level is multiplied by a random distance factor that is within the diameter of the data shape.

From the graphs, we can deduce that increasing the levels of noise reduces the quality of the resulting match by increasing the final mean squared error, as expected.

6 Conclusion and Future Work

The standard ICP algorithm may not converge when rotation angles are high. The search for correspondence is also computationally expensive. We have presented an improvement to the ICP algorithm for matching 3D

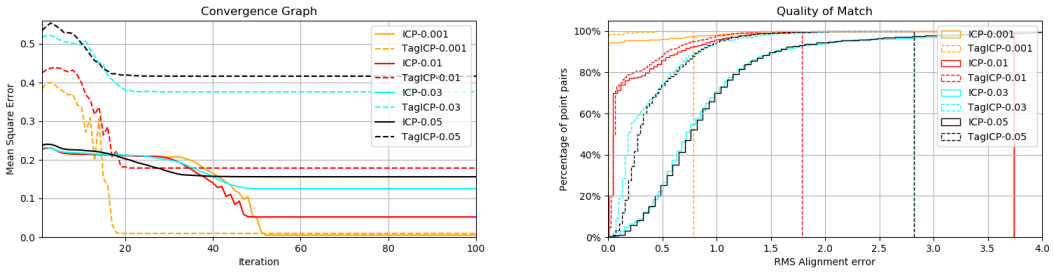


Figure 4: Noise PDB ID: 2JKF. Convergence and final match quality at varying noise levels, 0.001 to 0.05

protein structures that shows improvements over the standard ICP algorithm. Our improvement to the closest point search also demonstrates a smaller search space for searching each correspondence thus reducing the overall time and resulting complexity of the closest point search method. Some future work on Tagged-ICP include caching the partitioning of points based on tags which might further improve on speeds.

References

Angaran, S., Bock, M. E., Garutti, C., and Guerra, C. (2009). MolLoc: a web tool for the local structural alignment of molecular surfaces. *Nucleic Acids Res*, 37(suppl_2):W565–W570.

Aoki, Y., Goforth, H., Srivatsan, R. A., and Lucey, S. (2019). PointNetLK: Robust & Efficient Point Cloud Registration using PointNet. *arXiv:1903.05711 [cs]*.

Attia, M. and Slama, Y. (2017). Efficient Initial Guess Determination Based on 3D Point Cloud Projection for ICP Algorithms. In *International Conference on High Performance Computing Simulation*, pages 807–814.

Axenopoulos, A., Daras, P., Papadopoulos, G. E., and Houstis, E. N. (2013). SP-Dock: Protein-Protein Docking Using Shape and Physicochemical Complementarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(1):135–150.

Axenopoulos, A., Rafailidis, D., Papadopoulos, G., Houstis, E. N., and Daras, P. (2016). Similarity Search of Flexible 3D Molecules Combining Local and Global Shape Descriptors. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(5):954–970.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Res*, 28(1):235–242.

Bertolazzi, P., Guerra, C., and Liuzzi, G. (2010). A global optimization algorithm for protein surface alignment. *BMC Bioinformatics*, 11(1):488.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Burlacu, A., Cohal, A., Caraiman, S., and Condurache, D. (2016). Iterative closest point problem: A tensorial approach to finding the initial guess. In *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 508–513.

Chen, J. and Belaton, B. (2014). An Improved Iterative Closest Point Algorithm for Rigid Point Registration. In Wang, X., Pedrycz, W., Chan, P., and He, Q., editors, *Machine Learning and Cybernetics*, volume 481, pages 255–263. Springer Berlin Heidelberg, Berlin, Heidelberg.

Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. In *1991 IEEE International Conference on Robotics and Automation Proceedings*, pages 2724–2729 vol.3.

Chen, Y. and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155.

Cignoni, P., Montani, C., and Scopigno, R. (1998). DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed. *Computer-Aided Design*, 30(5):333–341.

Combes, B. and Prima, S. (2009). Prior affinity measures on matches for ICP-like nonlinear registration of free-form surfaces. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 370–373.

Donoso, F., Austin, K., and McAree, P. (2017a). Three new Iterative Closest Point variant-methods that improve scan matching for surface mining terrain. *Robotics and Autonomous Systems*, 95:117–128.

Donoso, F. A., Austin, K. J., and McAree, P. R. (2017b). How do ICP variants perform when used for scan matching terrain point clouds? *Robotics and Autonomous Systems*, 87:147–161.

Ellingson, L. and Zhang, J. (2012). Protein Surface Matching by Combining Local and Global Geometric Information. *PLoS ONE*, 7(7):e40540.

Gehrer, D. (2015). Cellunity - an interactive tool for illustrative visualization of molecular reactions. In *19th Central European Seminar on Computer Graphics (CESCG)*, page 7.

Holm, L. and Sander, C. (1998). Touring protein fold space with Dali/FSSP. *Nucleic Acids Res*, 26(1):316–319.

Kinoshita, K. and Nakamura, H. (2005). Identification of the ligand binding sites on the molecular surface of proteins. *Protein Sci*, 14(3):711–718.

Levoy, M., Rusinkiewicz, S., Ginzton, M., Ginsberg, J., Pulli, K., Koller, D., Anderson, S., Shade, J., Curless, B., Pereira, L., Davis, J., and Fulk, D. (2000). The digital michelangelo project 3d scanning of large statues. page 14.

Lu, J., Xu, G., Zhang, S., and Lu, B. (2016). An effective sequence-alignment-free superpositioning of pairwise or multiple structures with missing data. *Algorithms Mol Biol*, 11(1):18.

Mora, H., Mora-Pascual, J. M., García-García, A., and Martínez-González, P. (2016). Computational Analysis of Distance Operators for the Iterative Closest Point Algorithm. *PLOS ONE*, 11(10):e0164694.

Nabavi, A., Mamisch, C. T., Gering, D. T., Kacher, D. F., Pergolizzi, R. S., Wells, W. M., Kikinis, R., Black, P. M., and Jolesz, F. A. (2000). Image-guided therapy and intraoperative MRI in neurosurgery. *Minimally Invasive Therapy & Allied Technologies*, 9(3-4):277–286.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453. ZSCC: 0013205.

Phillips, J. M., Liu, R., and Tomasi, C. (2007). Outlier Robust ICP for Minimizing Fractional RMSD. In *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*, pages 427–434.

Schmitt, S., Kuhn, D., and Klebe, G. (2002). A New Method to Detect Related Function Among Proteins Independent of Sequence and Fold Homology. *Journal of Molecular Biology*, 323(2):387–406.

Schutz, C., Jost, T., and Hugli, H. (1998). Multi-feature matching algorithm for free-form 3D surface registration. In *Fourteenth International Conference on Pattern Recognition*, volume 2, pages 982–984 vol.2.

Sharp, G., Lee, S., and Wehe, D. (2002). ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):90–102.

Shulman-Peleg, A., Shatsky, M., Nussinov, R., and Wolfson, H. J. (2008). MultiBind and MAPPIS: webservers for multiple alignment of protein 3D-binding sites and their interactions. *Nucleic Acids Res*, 36(suppl_2):W260–W264.

Thirion, J.-P. (1996). New feature points based on geometric invariants for 3D image registration. *Int J Comput Vision*, 18(2):121–137.

Unity Technologies (2019). *Unity*.

Wang, Y. and Solomon, J. M. (2019). Deep Closest Point: Learning Representations for Point Cloud Registration. *arXiv:1905.03304 [cs]*.

Weskamp, N., Kuhn, D., Hüllermeier, E., and Klebe, G. (2004). Efficient similarity search in protein structure databases by k-clique hashing. *Bioinformatics*, 20(10):1522–1526.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets a deep representation for volumetric shapes. pages 1912–1920.

Yuan, C., Yu, X., and Luo, Z. (2016). 3D point cloud matching based on principal component analysis and iterative closest point algorithm. In *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, pages 404–408.

Zhou, L., Griffith, R., and Gaeta, B. (2014). Combining spatial and chemical information for clustering pharmacophores. *BMC Bioinformatics*, 15(16):S5.

A Performance Analysis of a State-of-the-Art CNN versus a Capsule Network for Cell Image Classification

Liam Murphy
Liam.Murphy2@dell.com

Dell Technologies

Ruairí O'Reilly
Ruairi.OReilly@cit.ie

Cork Institute of Technology

July 7, 2020

Abstract

Despite state of the art performance on object recognition and image classification problems, CNNs are considered to have two significant weaknesses. Firstly, their inability to cater for changes in object orientation, position or lighting. Secondly, their inability to deal with part-whole relationships between objects. Capsule Networks are an enhancement to CNNs to more closely model the viewpoint invariance capability of human vision. The application of Capsule Networks to well known datasets, such as MNIST and NORB, has achieved state of the art performance, while application to other datasets has had mixed results. The application of Capsule Networks to domains such as medical based imaging problems is of significant interest as they have been shown to train accurately on some datasets with limited training data. The contribution of this research is to compare the performance of a Capsule Network to a highly accurate CNN specifically developed for classification of malaria infected and uninfected cell images. It looks at how the accuracy of each model is affected by the volume of available training data, and at how robust each model is to classifying test images subjected to transformation such as rotation, shear and lighting change.

Keywords: Image Classification, Artificial Neural Networks, Convolutional Neural Networks, Capsule Networks

1 Introduction

While CNNs have been very successful in image classification tasks, Capsule Networks are designed to overcome their perceived shortcomings. [Sabour et al., 2017] describe how CNNs, while having the in-built ability to deal with object translational equivariance, do not have any such ability when it comes to other changes in object viewpoint such as rotation, and cannot generalise to new viewpoints. This problem has mainly been circumvented by using more data during model training, with as many viewpoints included. This limitation of CNNs is seen as a weakness, and viewed by [Hinton, 2017] as something that needs to be addressed.

A second problem with CNNs is what is known as the 'Picasso effect'. This effect is a consequence of the pooling and fully connected layers losing the spatial relationship information between features, which in turn means CNNs are not capable of establishing true part-whole composition [Hinton, 2017]. Capsule Networks address these problems in three ways:

1. In a Capsule Network, capsule neurons output a vector [Sabour et al., 2017] or a matrix [Sabour et al., 2018]. This non-scalar quantity allows capsule neurons to encode a more powerful representation of features such as 'pose' [Sabour et al., 2017] [Sabour et al., 2018].

2. Pooling is replaced with a more complex concept referred to as 'routing by agreement' [Sabour et al., 2017] [Sabour et al., 2018]. This mechanism is used between adjacent capsule layers. The purpose of 'routing by agreement' is for a higher and lower capsule layer to agree on which capsules in the higher layer will receive input from which capsules in the lower layer. The intent of this routing is to enable capsules to build part-whole relationships.

3. The fully connected final layers of CNNs are removed. In a CNN, the fully connected layers provide a crude form of part-whole feature relationships, as well as assignment of class probability. In a Capsule Network, the final layer contains class capsules, and the 'strength' of the capsules is used directly for the classification decision. This is possible because the 'routing by agreement', and the part-whole relationships that capsules form, means that the individual class capsules provide a high level 'viewpoint invariant' representation of a distinct class.

The motivation for this research is to establish what significance these two capabilities of a Capsule Network have when working with a dataset that has less distinct class separation than MNIST or NORB, such as cell images, particularly if the volume of training data is limited.

The remainder of the paper is organised as follows: Section 2 provides an overview of similar research into the application of Capsule Networks to medical image classification tasks. Section 3 summarises the experimental objectives and methodology used for this research. Section 4 presents the main results and findings, and Section 5 summarises the main conclusions from this work.

2 Related Work

While the work on Capsules by [Sabour et al., 2017] [Sabour et al., 2018] shows promising results for vector and matrix Capsules, the number and type of datasets to which they have been applied is limited. While research of Capsule Networks is relatively immature compared to CNNs, they are now attracting considerable interest. In the medical field, where dataset sizes can be limited, there have been some successful applications of Capsule Networks to classification problems.

[Iesmantas and Alzbutas, 2018] developed a Capsule Network model based on [Sabour et al., 2017] for use in the problem of breast cancer classification of histology images. Their Capsule Network achieved an average cross-validation accuracy of 87% across four types of biopsy images, more accurate than a CNN previously designed for the task. [Anupama et al., 2019] also developed a Capsule Network model based on [Sabour et al., 2017] for the same problem and dataset, and also found that it performed better than existing models, including CNNs.

Motivated by the potential of Capsule Networks to cater for affine transformations in images, as well as their ability to train on smaller datasets, [Afshar et al., 2018] applied a Capsule Network model to the classification of brain tumour images. They found that Capsule Networks outperformed a conventional CNN model previously designed for use on this problem. [Mobiny and Van Nguyen, 2018] applied a Capsule Network to the problem of lung cancer screening images and found that the Capsule Network outperformed a conventional CNN. They also concluded that it was a more appropriate model for use in cases where training data is limited.

The datasets used in the above research were all limited in size. That used by [Iesmantas and Alzbutas, 2018] and [Anupama et al., 2019] consisted of 249 training images for example. The dataset chosen for our evaluation was a publicly available dataset of malaria-infected and uninfected cells [Kaggle, 2018]. There are several motivations for selecting this dataset.

- The nature of the dataset images (Figure 1) suggests that there is no preferred orientation concerning the features that need to be learned. Handling variations in such orientation and pose is what a Capsule Network should be good at compared to a CNN, especially if training data is limited.

 ISBN 978-0-9934207-5-7

- The malaria dataset images are segmented and are devoid of a noisy background. Research by [Afshar et al., 2018] suggests the performance of Capsule Network models is improved when used on segmented images.
- While this research focuses on applying a Capsule Network to classifying segmented images of malaria-infected and uninfected cells, the results could have implications for the suitability of Capsule Networks versus CNNs to the automated or assisted classification of segmented images of other diseases, particularly if training data is limited.
- The number of images in the malaria dataset (over 26,000), facilitates training and evaluation of model performance across a range of training set sizes.

In addition to the above, a highly accurate sixteen layer CNN by [Liang et al., 2016] designed explicitly for this dataset, is available as a baseline against which to compare. The model by [Liang et al., 2016] achieved average accuracy of 97.3% on the malaria dataset.

3 Methodology

The first step undertaken was to pre-process a copy of the malaria cell image dataset [Kaggle, 2018] and standardise image dimensions. The dataset [Kaggle, 2018] consists of 13,779 images per class label (Uninfected, Parasitised). The images are not of uniform shape or colour and range in size from 55x40 pixels to 364x340. Each image in the dataset was re-sized to 44x44 pixels. These are the same image dimensions as used by [Liang et al., 2016] in their research. Figure 1 depicts a selection of processed cell images taken from the dataset. The class label "0" is used to denote healthy cells while label "1" indicates parasitised (infected) cells.

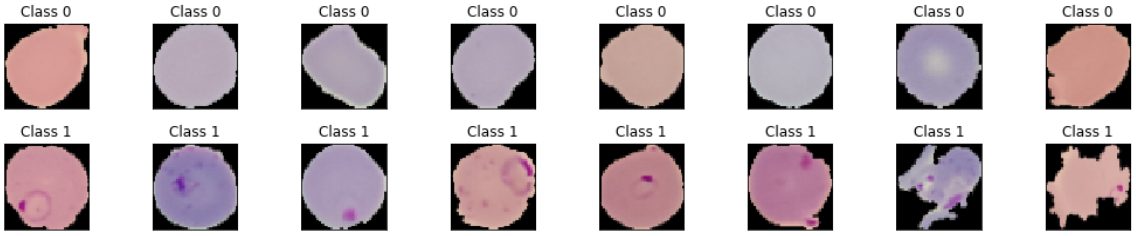


Figure 1: Sample of Cell Images from the Dataset ([Kaggle, 2018])

The second step was to implement the CNN described in [Liang et al., 2016] for baseline comparison. For this research, the model was implemented in Keras using the same filter sizes and other model configuration as detailed in [Liang et al., 2016]. For initial test, the dataset was split into a 10% test set of 2756 images. The remaining 24802 images were further split 90%/10% (22321/2481) into training and validation sets. When trained for 60 epochs and with a batch size of 32, the CNN achieved an accuracy of 95.6% on the validation test data. On the hold-out test set, the model scored an accuracy of 96.7%. Having developed the model, subsequent models were then built and evaluated using 10-fold Cross-Validation with different fractions of the available training data (as described in Section 4 and denoted in Table 1).[1]

The third step was to implement a Capsule Network based on [Sabour et al., 2017] and tune it for accurate performance on the malaria dataset. The Capsule Network implementation was based on the Keras model of [Guo, 2017]. Initial testing of the model using the same configuration as [Sabour et al., 2017], indicated that the model had too much capacity to train well on the malaria dataset. Only by significantly reducing the number of filters in the input convolution layer, and the number of capsules in the primary capsule layer was it possible to obtain models that would train to high accuracy and with reduced over-fit.

Figure 2 depicts the configuration of the [Sabour et al., 2017] model used for evaluation with the malaria dataset. The values of 5 and 11 were selected as the kernel sizes for the convolutional and PrimaryCaps layer, respectively. The number of filters in the input convolutional layer was 4 and only one capsule was used in the PrimaryCaps layer. A best model score of over 94% was achieved when testing this combination.

[1]The dataset and model implementations are available from `https://gitlab.com/liam.m.murphy/msccode.git`

 ISBN 978-0-9934207-5-7

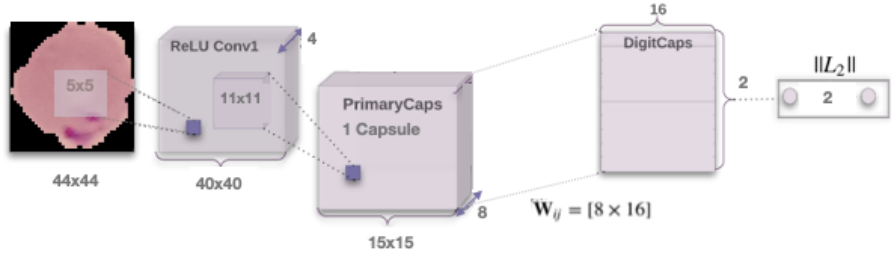


Figure 2: Capsule Network model configuration selected for evaluation.

Using more filters in the convolutional or more capsules in the PrimaryCaps layer made no noticeable difference to accuracy. Increasing capsules only served to increase model training time as it also requires more computation work by the 'Routing by Agreement' algorithm.

Figure 3 depicts a modified version of the Capsule Network that was also evaluated. A second convolutional layer was added in front of the PrimaryCaps layer. Using a stride of 2, this layer reduces the volume of data presented to the PrimaryCaps layer. This modified model proved to be more accurate than that of Figure 2.

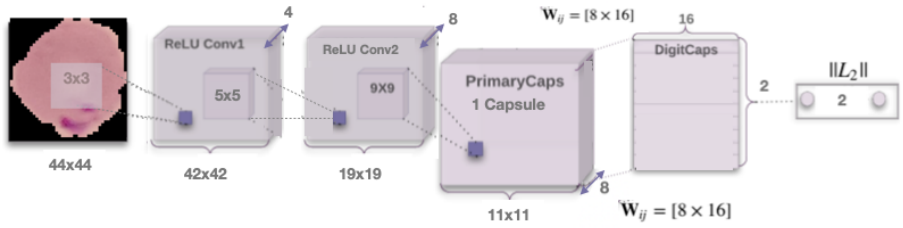


Figure 3: Modified Capsule Network to improve accuracy.

Having settled on Capsule Network configurations, models were then built and evaluated using 10-fold Cross-Validation with different fractions of the available training data. The prior assumption was that the Capsule Network might have an advantage over the CNN as training data was reduced.

The final step was to evaluate how each model type performed with test data that had been subjected to a transformation. The transformations included flips, rotation, shear and light intensity changes. For this test, the CNN model and Capsule Network models were trained for 30 epochs using 100% of the available training data (Table 1). Each model was then evaluated against the hold-out test set and the four augmented versions of the test set. Model training and inference time were also measured.

4 Experimental Setup and Results

4.1 Model Accuracy with 10-Fold Cross-Validation using different fractions of available data

For testing purposes, the dataset was split into a 10% hold out test set of 2756 images, with the remaining 24802 images being the maximum available for training and validation sets.

To evaluate model performance against the quantity of available training data, each model type was trained and cross-validation tested using six different volumes of training data. Table 1 denotes the allocation of images between training and validation for the six 10-fold cross-validation tests. Class balance was maintained in all training, validation and test splits. All models were trained using a batch size of 32. The number of epochs for each test was set to ensure that each model had an equivalent number of mini-batches over which to make weight adjustments during training. Epochs were calculated using:

$$epochs = (6000)/frac$$

Model check-pointing was also used in all test cases to save the model with minimum validation loss, and this was then scored against the test set.

Training Fraction	Images (10-folds)	Validation (1-fold)	Training (9-folds)	Test set	batch	epochs
100%	24802	2481	22321	2756	32	60
80%	19840	1984	17856	2756	32	75
60%	14880	1488	13392	2756	32	100
40%	9920	992	8928	2756	32	155
20%	4960	496	4464	2756	32	300
10%	2480	248	2232	2756	32	600

Table 1: Training, Validation and Test set sizes for the 10-fold Cross Validation tests

Fraction %	CNN Model	CapsNet	Modified CapsNet
100	0.9537 (+/- 0.0027)	0.9325 (+/- 0.0079)	0.9465 (+/- 0.0032)
80	0.9504 (+/- 0.0030)	0.9270 (+/- 0.0112)	0.9377 (+/- 0.0062)
60	0.9494 (+/- 0.0020)	0.9208 (+/- 0.0053)	0.9366 (+/- 0.0070)
40	0.9585 (+/- 0.0044)	0.9254 (+/- 0.0086)	0.9297 (+/- 0.0118)
20	0.9461 (+/- 0.0077)	0.8530 (+/- 0.0290)	0.9292 (+/- 0.0087)
10	0.9342 (+/- 0.0149)	0.8224 (+/- 0.0165)	0.8869 (+/- 0.0350)

Table 2: 10-fold cross validation accuracy with different fractions of data.

Table 2 denotes the 10-fold cross-validation results for the implemented model types. The results indicate that the CNN is more accurate in all cases. The CNN can get high accuracy even when trained on 10% of available training data. In comparison, the Capsule Network models decreases in accuracy when trained on the smaller training set volumes. Performance differences are even more pronounced when the training set size is further reduced. Figure 4 depicts learning curves for the models for training set sizes at 10% and less, and clearly highlights the problem of the Capsule Networks rapidly over-fitting to the training data. While data augmentation can be used to improve accuracy and fit on these lower training set sizes, the significance of the results, for this and similar datasets, is that this is more important for the Capsule Network.

4.2 Model Training and Inference Times

Table 3 denotes typical training and inference time measured on Google Colab. Note that this is representative only, as the type of GPU allocated by Colab is not fixed. To train the CNN model for 30 epochs using 100% of available training data (22321 images) typically takes just over 200 seconds, 3 times as fast as the Capsule Network Model. Training time for the Modified Capsule Network is only marginally better. Inference time for the CNN is significantly faster than either Capsule Network model by a factor of 4.

Metric	CNN	CapsNet	Mod CapsNet	Test details
Training	204.6 s	623.5 s	597.5 s	Training/Validation 22321/2481 Images 30 Epochs, Batch size 32. Single model
Inference	0.107 ms	0.476 ms	0.475 ms	Averaged over 27560 samples

Table 3: Model Training and Inference times. Measured on Colab with an Nvidia Tesla P4 GPU

While the Modified Capsule Network gets close to the CNN model accuracy, it should be noted that it does so with significantly fewer core model parameters. The bulk of the parameters in the Capsule Network lie in the reconstruction decoder (over 6.48 Million). In terms of the core layers, the modified model has only 37,400 parameters. By contrast, the CNN model has 977,000. The training time of the Capsule Networks can be mitigated somewhat by reducing the number of neurons in the reconstruction decoder. A test of the training

time for a Capsule Network with a reduced reconstruction decoder (800,000 parameters) still took twice as long to train compared to the CNN indicating that 'routing by agreement' is an expensive computation.

4.3 Model Performance with Augmented Test Data

For this experiment, the various models were trained for 30 epochs on all of the available training data. Each model was then tested against the hold-out test set along with four augmented versions of that set. The performance of each model on the unmodified (baseline) and augmented test sets are shown in Table 4. For the augmented tests, the results were averaged over 20 runs, with each run using 4096 images per test. The standard deviation is also shown.

Model	Baseline Accuracy	Flips	Rotation (90°)	Intensity (0.6, 1.4)	Shear (30)
Liang CNN	95.2	95.15 (+/-0.14)	94.05 (+/-0.18)	53.88 (+/-0.24)	93.17 (+/-0.22)
Capsule Net	93.3	93.31 (+/-0.17)	91.14 (+/-0.39)	50.01 (+/-0.01)	91.21 (+/-0.21)
Modified Caps. Net	94.9	94.86 (+/-0.12)	92.82 (+/-0.33)	50.00 (+/-0.00)	92.63 (+/-0.41)

Table 4: CNN and Capsule Network model performance on augmented test data

The results indicate that all model types are immune to flips and show similar impact to accuracy from rotational and shear transformation. This result is somewhat surprising as all transformations are expected to suit the Capsule Network model. The explanation for this is that the nature of the dataset most likely eliminates any advantage the Capsule Network might have. In the case of the malaria dataset, parasitised cells appear to be visually distinguishable from uninfected cells by the presence of blemishes within the cell image. These blemishes appear as reddish spots and in varying numbers, size and shape (see Figure 1). As such, they have no particular pattern, and the overall effect of this is to act as natural data augmentation with regard to shear, flips or rotation. In addition, the cell images are two dimensional, and variation in perspective has effectively been removed as a variable. In terms of classification, a model only needs to be sensitive to the presence or absence of the blemishes that are indicative of infection.

5 Conclusions

This research successfully demonstrated that a Capsule Network based on [Sabour et al., 2017] (Figure 2) can achieve very accurate results on the malaria dataset. A modified version (Figure 3), containing an extra convolution layer, was also evaluated and proved to be more accurate. Both Capsule Network models, however, were less accurate than a CNN implementation based on [Liang et al., 2016]. The difference in accuracy between the CNN and Capsule Network models becomes more pronounced as training data is limited (Table 2 and Figure 4).

Both Capsule Network models rapidly over-fit when the volume of training data is limited. The Modified Capsule Network model, with its extra convolution layer, can be seen to be more robust than the basic Capsule Network model. This robustness is due to the extra layer providing some mitigation against over-fit through reducing the volume of data presented to the PrimaryCaps layer. This research concludes that the underlying cause of over-fit is that the Capsule Network architecture does not offer sufficient regularisation to prevent it. While the loss function of [Sabour et al., 2017] provides some regularisation through the reconstruction decoder, this may not work well when classes have very similar shape, such as cell images. Research into addressing the regularisation problem is desirable, particularly methods that are not based on the reconstruction decoder, as the size of this decoder is also a disadvantage.

This research also found that no model type had any advantage over the other when classifying test data subjected to transformations such as shear or rotation. It was expected that this test would benefit the Capsule Network, particularly when training data is limited, as it is designed for viewpoint invariance. However, this

advantage is obviated by the nature of the dataset as feature orientation appears irrelevant. This is likely true for classification problems with similar datasets. Where Capsule Networks may offer an advantage is with datasets containing features that have a more significant spatial relationship, where viewpoints are more varied, or where more complex part-whole relationships may exist. A similar study, using a dataset where such 'pose' estimation is more relevant, would be beneficial.

Although the CNN has proved superior in the context of this dataset, Capsule Networks remain appealing by virtue of their theoretical ability to cater for viewpoint invariance. This enables them to realise less complex models compared to CNNs. They are not, however, a panacea for image classification problems. As in the case in the malaria dataset, they may be beaten by a specialised CNN, and depending on the relevance of 'pose' estimation for a dataset, may not be able to offer advantage over a CNN even when training data is limited. Training time also remains a significant barrier to more widespread use, so research into more computationally efficient algorithms with more parallelism is also necessary.

References

[Afshar et al., 2018] Afshar, P., Mohammadi, A., and Plataniotis, K. N. (2018). Brain tumor type classification via capsule networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3129–3133. IEEE.

[Anupama et al., 2019] Anupama, M., Sowmya, V., and Soman, K. (2019). Breast cancer classification using capsule network with preprocessed histology images. In *2019 International Conference on Communication and Signal Processing (ICCSP)*, pages 0143–0147. IEEE.

[Guo, 2017] Guo, X. (2017). Github - xifengguo/capsnet-keras: A keras implementation of capsnet in nips2017 paper "dynamic routing between capsules". `https://github.com/XifengGuo/CapsNet-Keras`. (Accessed on 02/24/2020).

[Hinton, 2017] Hinton, Geoffry, E. (2017). Geoffrey hinton talk "what is wrong with convolutional neural nets ?" - youtube. `https://www.youtube.com/watch?v=rTawFwUvnLE&feature=youtu.be`. (Accessed on 02/17/2020).

[Iesmantas and Alzbutas, 2018] Iesmantas, T. and Alzbutas, R. (2018). Convolutional capsule network for classification of breast cancer histology images. In *International Conference Image Analysis and Recognition*, pages 853–860. Springer.

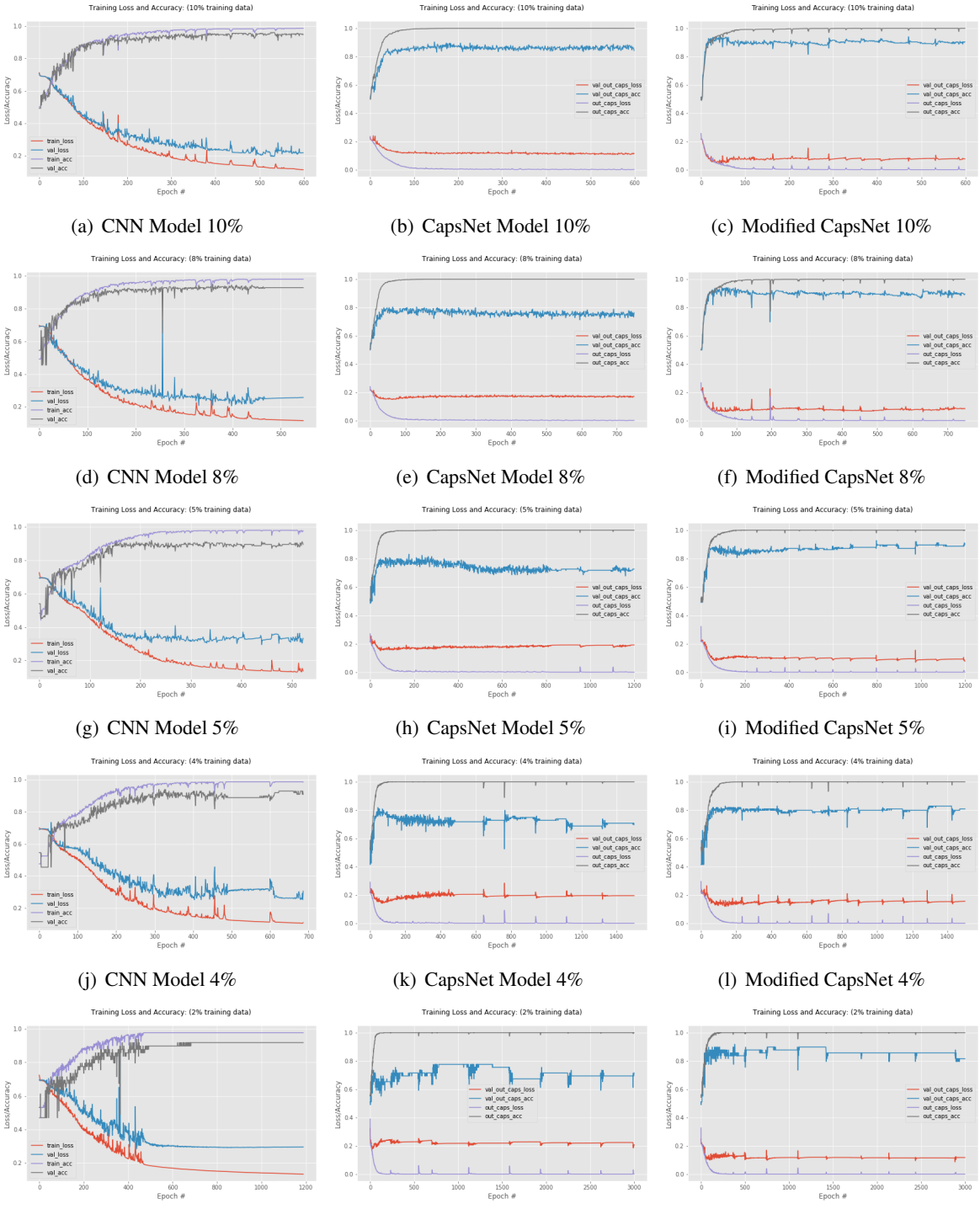
[Kaggle, 2018] Kaggle (2018). Malaria cell images dataset. `https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria`. (Accessed on 05/21/2020).

[Liang et al., 2016] Liang, Z., Powell, A., Ersoy, I., Poostchi, M., Silamut, K., Palaniappan, K., Guo, P., Hossain, M. A., Sameer, A., Maude, R. J., et al. (2016). Cnn-based image analysis for malaria diagnosis. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 493–496. IEEE.

[Mobiny and Van Nguyen, 2018] Mobiny, A. and Van Nguyen, H. (2018). Fast capsnet for lung cancer screening. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 741–749. Springer.

[Sabour et al., 2017] Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866.

[Sabour et al., 2018] Sabour, S., Frosst, N., and Hinton, G. E. (2018). Matrix capsules with em routing. In *6th International Conference on Learning Representations, ICLR*.



(a) CNN Model 10% (b) CapsNet Model 10% (c) Modified CapsNet 10%

(d) CNN Model 8% (e) CapsNet Model 8% (f) Modified CapsNet 8%

(g) CNN Model 5% (h) CapsNet Model 5% (i) Modified CapsNet 5%

(j) CNN Model 4% (k) CapsNet Model 4% (l) Modified CapsNet 4%

(m) CNN Model 2% (n) CapsNet Model 2% (o) Modified CapsNet 2%

Figure 4: Model learning curves for fractions of available training data

 ISBN 978-0-9934207-5-7

Consistency of Scale Equivariance in Internal Representations of CNNs

Vincent Andrearczyk[1], Mara Graziani[1,2], Henning Müller[1,2], and Adrien Depeursinge[1,3]

[1]*Institute of Information Systems, University of Applied Sciences Western Switzerland (HES-SO), Sierre, Switzerland*
[2]*Hopitaux Universitaires de Genève (HUG), Geneva, Switzerland*
[3]*Centre Hospitalier Universitaire Vaudois (CHUV), Lausanne, Switzerland*

August 24, 2020

Abstract

Despite the approximate invariance to scale learned in deep Convolutional Neural Networks (CNNs) trained on natural images, intermediate layers have been shown to contain information of scale while the invariance is only obtained in the final layers. In this paper, we experimentally analyze how this scale information is encoded in the hidden layers. Linear regression of scale is used to (i) evaluate whether scale information can be encoded, at a given layer, by individual response maps or a combination of many of them is necessary; (ii) evaluate whether the encoding of scale is shared among classes. If we can find a direction representative of scale variations in the hidden space, is this consistent across the data manifold? Or is it rather encoded locally within class-specific neighborhoods? We observe that scale information is encoded as a combination of a few response maps (around 3%) and that the encoding is relatively consistent across classes, with some amount of class-specific encoding.

1 Introduction

Convolutional Neural Networks (CNNs) can implicitly learn an approximate invariance to transformations and variations in the training data [1]. Excellent generalization to unseen images has been obtained in natural images (ImageNet [2]), despite their large intra-class variability. In particular, CNNs must learn invariance to scale to be able to recognize textures, objects and scenes regardless of the point of view. Scale plays a crucial role in computer vision and medical imaging [3]. Besides, transfer learning has been extensively used in many domains. Various empirical studies, e.g in medical imaging, compared vanilla CNNs with pre-trained models with little attempts in understanding why one may work better than another in different scenarios besides the speed of convergence. Studying scale equivariance in pre-trained CNNs informs us on the adaptability to domains distant from natural images, in which scale often carries crucial information. Motivated by the transfer learning of CNNs trained on natural images to other domains in which the scale and object size may be informative (e.g. fixed viewpoint in medical imaging), we studied the presence of scale information in intermediate activations in [4, 5]. It was shown that, for a given class, the scale can be linearly regressed within intermediate activations and that the invariance is learned in the final layers. In [5], this finding was used to improve the regression of magnification in histopathology images using pretrained CNNs.

In this paper, we build on top of the analyses in [4, 5] to better understand how the scale information is encoded in the internal activations of commonly used state of the art CNNs [1]. We evaluate empirically two key questions related to the learning process and data processing of CNNs. Is scale encoded by a single or by multiple response maps? Is scale encoded differently for different classes? To address the first question, we evaluate if a minimal combination of response maps suffices to encode the scale information, starting from

 ISBN 978-0-9934207-5-7

a single response only. While individualized neuron responses to a specific representation of a concept (e.g. scale, lighting or texture) were analyzed in [6, 7] by visualizing activations for exemplar input images, this paper provides a quantitative analysis. Second, the initial evaluation of scale regression in [4] considered only individual classes, leaving opened the question of whether scale is learned individually for each class or as a general feature that comprises multiple classes. We evaluate the consistency of the direction of increasing values of scale in the deep hidden space. This helps to understand whether scale encoded in the same way across the data manifold or if local regression and manifold learning should be used to capture the scale encoding.

As described in Section 3, the analysis is based on linear regression of scale at intermediate layers of CNNs trained on ImageNet. The scale is measured as a ratio based on manually annotated bounding boxes from the Pascal-VOC annotations. Section 4 is dedicated to the alignment of the scale regression along individual response maps. The consistency of the scale encoding across the data manifold is studied in Section 5.

2 Related Work

Several researchers have studied the invariance and equivariance to scale learned in CNNs by evaluating internal activations for images at different scales [6, 8]. More generally, the equivariance to geometric image transformations such as flips and rescaling in the intermediate feature space was studied in [9], concluding that scale invariance is implicitly learned in CNNs as prediction results are not improved by reversing the scaling transformations in the feature space. Contrasting these results, the vulnerability of standard CNNs to adversarial attacks with transformations including scaling was studied in [10], supporting the need for built-in invariance. A supervised training method was proposed in [11] to disentangle the transformations including rotations and scales, providing built-in equivariance properties.

Particularly related to our work, post-hoc interpretability methods, as defined in [12], interpret trained models without modifying their optimization. Among these methods, linear models have been used as probes to explain intermediate network layers, in line with our strategy to analyze scale equivariance. Linear classifier probes [13] were proposed to analyze class-separability at intermediate network layers in terms of the classification of the class labels by a linear model. Testing with Concept Activation Vectors (TCAV) [14] and Regression Concept Vectors (RCVs) [15, 16] were developed to analyze the presence of concepts (binary and continuous measures) in intermediate layers of a deep network by linear classification and regression, respectively. Linear probes, and particularly concept-based ones, have shown relevant results in different analyses and applications [17, 18]. This approach is well suited for our task of investigating scale equivariance in deep representation by modeling a linear probe for scale variations.

3 Methods

This section describes the notations and methods used in the experiments. We use an InceptionV3 [1] CNN trained on ImageNet [2][1]. We analyze this model at an intermediate layer by looking at the activations for different images of varying scale. In this paper, we evaluate the *mixed8* layer in InceptionV3 as it is a rather deep layer with large receptive fields and was particularly shown to encode scale in [4, 5].

We consider a scaling transformation of an input image I, $g_\sigma(I)$, parameterized by a scaling factor σ. We search a predictable linear transformation $g'_\sigma(\phi(I))$ in the d-dimensional feature space $\phi(\cdot)$ of the scaling $g_\sigma(\cdot)$ in the input space. If such a property is found, the representation $\phi(\cdot)$ is linearly equivariant to scale. To find a linear equivariance, one can search a regression vector $\mathbf{v}$ in the feature space to predict the scaling factor σ as a linear combination of the features $\phi_i(g_\sigma(I))$:[2]

$$\sigma = \sum_{i=1}^{d} v_i \phi_i(g_\sigma(I)) = \mathbf{v} \cdot \phi(g_\sigma(I)). \tag{1}$$

[1] Similar results were observed with ResNet50.

[2] For simplicity, we omit the intercept. In Eq. (1), the intercept would be v_0 with $\phi_0(g_\sigma(I)) = 1$

 ISBN 978-0-9934207-5-7

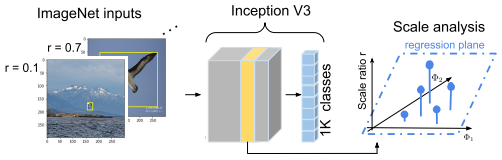


Figure 1: Overview of the scale analysis. The scale equivariance is analyzed at a given CNN layer (in yellow). Each data point I projected into this hidden space $\phi(I)$ is associated with its scale ratio r. We represent a regression plane corresponding to the vector $\mathbf{v}$ in two dimensions.

In this scenario, we can represent $g'_\sigma(\cdot)$ as a translation matrix in $\mathbb{R}^d$ by σ along $\mathbf{v}$, so that $g'_\sigma(\phi(I)) = \phi(I) + \mathbf{v}\cdot\sigma$.

To approximate $g'_\sigma(\cdot)$ and generalize to multiple input images, we consider images of objects that appear at various scales as shown in Fig. 2, and learn the regression of their corresponding scale ratios as in (1). The correlation of hidden features with the scale can be evaluated for rescaled versions of an image as in [8]. The rescaling, however, induces artifacts that can be highly correlated with response maps, as shown in [4, 19].

The approach used to analyze the scale information at a given layer is summarized in Fig. 1 and detailed in the following. We extract the activations for a set of inputs and spatially average the d response maps to obtain a d-dimensional feature vector $\phi(I)$ for each input image I. The averaging step is needed to remove locality and to reduce dimensionality as discussed in [16]. When dealing with flattened final layers, the spatial averaging does not apply. We also normalize the $\phi(I)$'s to have zero mean and unit variance on each dimension using the set of regression training data (not to be mistaken with the CNN training data). We then learn the regression vector $\mathbf{v}$ in (1). To this end, we regress a scale ratio which we define as $r = \sqrt{\frac{h_b \times w_b}{h_i \times w_i}}$, where h_b, w_b, h_i and w_i are the height and width of the object bounding box and of the image. Additionally to the standard residual sum of squares regression, we use a Lasso (L_1 norm minimization) regularization. It is used as a feature selection to evaluate the regression with a few features. The Lasso optimization function of the linear regression can be written as the residual sum of squares with L_1 penalization as follows.

$$\sum_{j=1}^{n} (y_j - \hat{y}_j)^2 + \alpha \sum_{i=0}^{d} |v_i|, \tag{2}$$

where n is the number of training images, y_j and $\hat{y}_j$ are the ground truth and predicted scale ratio r, α is the weight given to the Lasso regularization. Once optimized for a set of training images, the regression is evaluated on held-out images either from the same class or from another class using the R^2 coefficient of determination. $R^2 > 0$ means that the model is doing better than predicting the mean of the test set, while $R^2 = 1$ represents a perfect prediction. In the experiments, 220 training images from the *albatross* ImageNet class (ID: n02058221) are randomly drawn to regress the scale. Similarly, 220 test images are used as test set either from the same or from a different class as specified in the experiments.

Examples of images used in the analysis are illustrated in Fig.2. These three classes were selected from the ImageNet so that images contain a single object and these objects either share similarities across classes (two types of birds), or are fundamentally different (birds and racing cars).

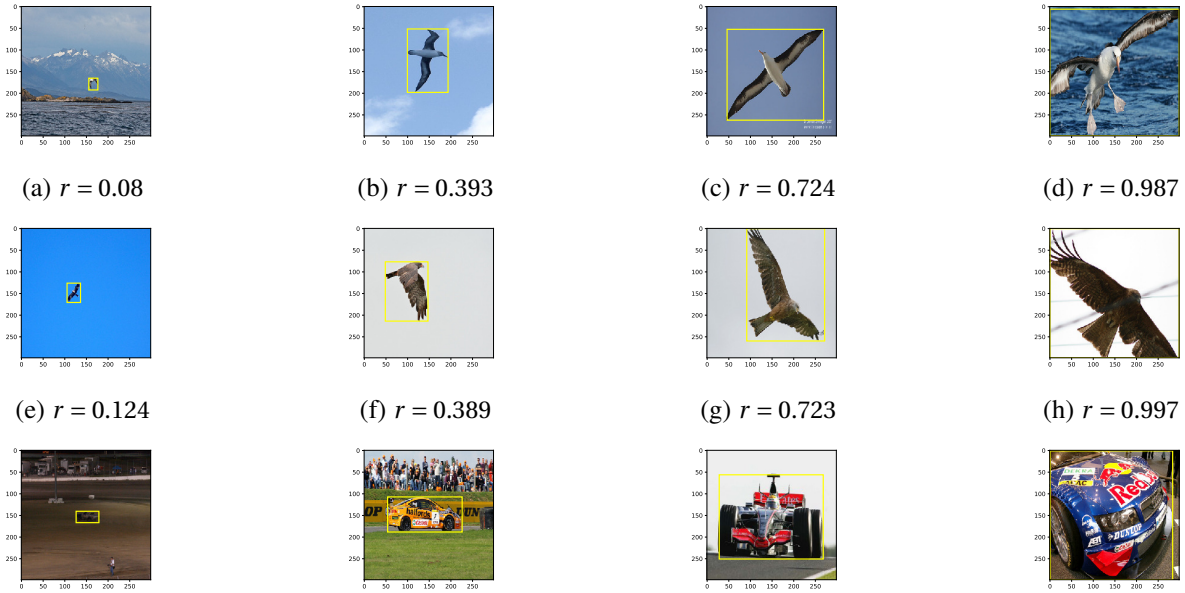
4 Alignment of Scale Information Along Individual Dimensions

In this section, we consider training and test images of the scale regression of a single class (*albatross*) as in [4].

4.1 Measuring Alignment

In this section, we evaluate whether the scale information is encoded by individual feature maps or as complex combinations of several feature maps. The alignment thus refers to scale values varying along a single feature map. After training the regression model, the regression vector is normalized: $\hat{\mathbf{v}} = \frac{\mathbf{v}}{|\mathbf{v}|}$. We then compute the

 ISBN 978-0-9934207-5-7



(a) $r = 0.08$ (b) $r = 0.393$ (c) $r = 0.724$ (d) $r = 0.987$

(e) $r = 0.124$ (f) $r = 0.389$ (g) $r = 0.723$ (h) $r = 0.997$

(i) $r = 0.124$ (j) $r = 0.395$ (k) $r = 0.723$ (l) $r = 0.970$

Figure 2: Examples of images and their respective scale ratio $r = \sqrt{\frac{h_b \times w_b}{h_i \times w_i}}$. Top row: *albatross* class; middle row *kite bird* class and bottom row: *racing car* class.

norm of the vector when retaining only a portion d' of its d dimensions sorted by their value $\hat{v}_i$. In this paper, we evaluate the layer *mixed8* of InceptionV3 which has $d = 1280$ feature maps. $|\hat{\mathbf{v}}_{1\ldots d'}| = 1$ means that all the scale information captured by the regression model is contained in the corresponding d' feature maps. This measure reflects how many feature maps (dimensions) are necessary to regress the scale. The first value, for a single coefficient, is $max(\hat{\mathbf{v}})$, which represents the largest alignment with an individual dimension. In Fig. 3 ('Non-reg' orange line), we report this contribution of dimensions in the regression of scale. We remind here that a dimension represents the spatial aggregation (average) of a deep response map.

With the Lasso regression (Fig. 3 'Lasso reg.' blue line) the regression vector aligns with few dimensions (40 out of 1280) while obtaining good generalization to new data. This means that the scale information can be represented by only a small portion of the response maps.

To complete this analysis of alignment along individual dimensions, we report the prediction performance of the Lasso scale regression for different values of α (see Eq. (2)), together with the number of non-zero coefficients in Fig. 4. The coefficient of determination R^2 is averaged across 10 runs (for each run, different splits are randomly drawn for the training and test sets). The results show that a good prediction of scale is obtained with a few dimensions (feature maps) retained by the Lasso regression. Even when drastically reducing the number of dimensions up to 1% of the original 1280, a good scale prediction is maintained with $R^2 > 0.6$. This compares well to the maximum value of 0.874 when using all dimensions. With 70 dimensions, no significant performance drop is observed as compared to this value. These results suggest that a combination of 1% of the features linearly encodes the scale information.

Yosinski et al. [6] showed by examples of activations that important features are learned in hidden layers and are encoded by individual neurons. It is generally referred to as local, as opposed to a representation distributed across multiple neurons. Our results are in line with previous work that showed the encoding of complex concepts distributed across response maps [7, 20].

 ISBN 978-0-9934207-5-7

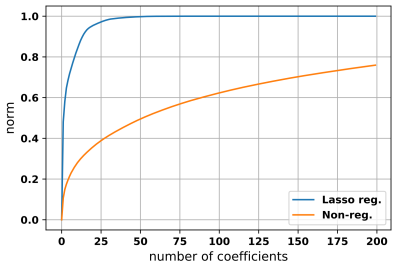


Figure 3: Analysis of the contribution of response maps for linear regression of scale. The norm $|\hat{\mathbf{v}}_{1\cdots d'}|$ of the normalized regression vector is reported for an increasing number d' of dimensions (sorted by decreasing contribution $\hat{v}_i$). We limit this analysis to the 200 largest values of $\hat{v}_i$. The non-regularized regression predicts the scale of the test data with a coefficient of determination $R^2 = 0.874$. The regularized Lasso regression is set to $\alpha = 0.001$. Only 40 coefficients out of 1280 are non-zero (~3%) and the scale is predicted on the test data with $R^2 = 0.839$.

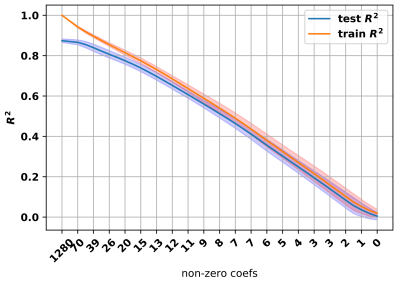


Figure 4: Evaluation of the scale prediction (R^2) of Lasso regression with different values of α. The model is trained and evaluated on images of the *albatross* class. The numbers of non-zero coefficients are reported on the x-axis instead of the corresponding α values (ranging from 0 to 0.01). The results are averaged across 10 runs and the 95% confidence intervals are reported. For comparison, we also report the results on the training data.

5 Directional Consistency of Scale Regression Across the Data Manifold

The data (ImageNet images here) lie on a complex manifold both at the input and hidden activations levels. The directions found with the scale regression lie in this space of (pooled) activations. Images from different classes can be far apart in this space and the question arises of whether learning a scale regression in one neighborhood (of a given class for instance) will generalize to the rest of the manifold. In other words, can we find a direction that is constantly equivariant with the scale information across the manifold? To empirically address this question, we adopt two complementary approaches. First, the generalization of a regression model trained on images from a given class will be evaluated on images from a different class. Second, the regression coefficients obtained from different classes will be compared (angle and cosine similarity between vectors).

A point that we have not mentioned yet is that the scale information does not come only from the averaging operation on response maps (see Section 3). Spatially large objects (i.e. large ratio r) cover a large region of the response maps thus resulting in larger averaged values that could be regressed. Note that this is one of the reasons why we use deep features (*mixed8* layer) with large effective receptive fields [21]. If the scale information, however, was solely contained in the size of the region covered by the object in the response maps, the regression would not generalize to new classes as different neurons activate for different objects or parts. We will show that the scale regression is, to some extent, similarly encoded for very different classes, suggesting that some neurons specifically encode scale information regardless of the object type. We can also analyze activations at the center of the object instead of a spatial average to get rid of the information contained in the region covered by the object in the response maps. In practice, we found this difficult to implement as the center of the object can vary substantially within the bounding box. Besides this, the effective receptive field, depending on the depth and input image, can be larger or smaller than the object of interest.

5.1 Predicting Scale in Unseen Classes

In this section, we evaluate the generalizability of a scale regression to images from unseen classes (unseen in the regression training phase). The regression model trained on images of a single class is evaluated on unseen images coming from a different class using the R^2. These results are compared with those obtained on images of the same class (results in Fig. 4) as reported in Table 1. We report the results for $\alpha = 0.001$ (for a

Table 1: Comparison of scale prediction within and across classes. The scale regression is trained with Lasso ($\alpha = 0.001$) on the training class and evaluated on held-out data of the same or a different class. Average accuracy across 10 runs and standard deviation are reported.

Training class	testing class	R^2 non-reg.	R^2 Lasso reg.
albatross	albatross	$0.843_{\pm 0.012\%}$	$0.839_{\pm 0.023\%}$
albatross	kite	$0.672_{\pm 0.076\%}$	$0.745_{\pm 0.042\%}$
albatross	car	$0.086_{\pm 0.240\%}$	$0.560_{\pm 0.069\%}$

(a) test class: *kite bird*

(b) test class: *racing car*

Figure 5: Evaluation of the scale prediction (R^2) of Lasso regression with different values of α. The model is trained on the *albatross* class and evaluated on (a) *kite bird* class and (b) *racing car* class. The numbers of non-zero coefficients are reported on the x-axis instead of the corresponding α values (ranging from 0 to 0.01). The results are averaged across 10 runs and the 95% confidence intervals are reported. For comparison, we also report the results on the training data.

good compromise between dimensionality reduction and performance), averaged across 10 runs. As mentioned previously, we train on the *albatross* class with 220 images and test on one class relatively similar (*kite birds*, ID: n01608432, 220 images) and one radically different (*racing cars*, ID: n04037443 220 images).

In Fig. 5, we report the scale prediction performance in different classes for varying values of α in the Lasso regression (see Eq. (2)). Unlike the prediction of images of the same class (Fig. 4), the best results are obtained with Lasso regression when it retains approximately 3% of the regression coefficients (39 out of 1280). The regression without regularization overfits the training class and does not generalize to new types of objects.

Table 1 and Fig. 5 show that the prediction of scale for images of a different type than the regression training data is relatively accurate, yet lower than on the same class ($R^2 = 0.745$ and 0.560 vs $R^2 = 0.839$). The prediction is also better for similar classes (two types of birds) than radically different classes (bird and car). It suggests some encoding of scale information relatively consistent across the data manifold, with some encoding more specific to each specific class or group of classes.

5.2 Comparing Regression Coefficients

In this section, we compare regression coefficients between multiple regression models to understand whether the scale encoding is the same for very different images (i.e. of different classes). For comparison of the regression vectors, we use the angle and cosine similarity between pairs of vectors. Both measures evaluate the alignment of two vectors, although the cosine similarity takes into account their magnitude while the angle does not. The angle, expressed in degrees, is bounded in $[0, 90]$. The cosine similarity is bounded in $[0, 1]$ and is 1 for an angle of $0°$. To obtain a baseline, we first train and compare two regression models on images from the same class. To evaluate the generalization, we then train and compare models on images from different classes. In Table 2, we report the angle between the coefficient vectors of different models as well as the cosine similarity, i.e. for two vectors $\mathbf{v_1}$ and $\mathbf{v_2}$, $\frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\| \|\mathbf{v_2}\|}$. The angle and cosine similarity are approximately $90°$ and 0 respectively when we regress randomly shuffled ratios. In a high-dimensional space, vectors are likely

 ISBN 978-0-9934207-5-7

Table 2: Comparison of regression models trained on several classes using angle and cosine similarity between vectors in the 1280-d space.

		non-reg.		Lasso	
Train	test	angle	cos.	angle	cos.
albatross	albatross	67.7°	0.410	46.2°	0.66
albatross	kite	72.4°	0.315	64.1°	0.414
albatross	car	86.7°	0.058	76.9°	0.227

to be orthogonal (angle 90° and cosine similarity of 0). The number of "almost-orthogonal" vectors grows exponentially with the dimension of the space (here $d = 1280$). Therefore, even an angle slightly below 90° and a cosine similarity slightly larger than zero can still reflect similarities in these high-dimensional vectors. These results further support the hypothesis of scale encoding being common across the data manifold, with some local encoding specific to different types of objects.

6 Conclusion

This paper proposed an experimental evaluation of scale equivariance inside CNNs. We built on top of our previous research using bounding-box-to-image ratios to train and evaluate regression models in deep activations. As an extension, we first showed that the scale information is encoded as a combination of a few response maps. Indeed, a good scale prediction was obtained when retaining less than 3% of the feature maps. A single response map, however, is not sufficient to encode scale information. We showed that the concept of scale is distributed across multiple response maps. As a second main result, we showed that scale information is encoded in a relatively consistent way across the data manifold. By learning a scale regression on a set of images from a given class, we can infer the scale of images from a completely different class. This result explains the generalization to scale regression in histopathology images from ImageNet pre-trained models in [5]. A limitation of our analysis is that we have only considered a linear regression. Besides, we considered only three classes for this exploratory work with a limited number of images.

With a similar approach, this analysis can be performed for other transformations (e.g. rotation) as well as binary or continuous measures (e.g. presence of a specific object part, first or second-order statistics) of the input images to understand how CNNs learn to detect and encode various types of information. Understanding the internal behavior of CNNs, "opening the black-box", is important to build intuition both for researchers designing or applying new models and end-users who lack understanding of the networks' behaviors. The benefits of interpreting deep representations are multiple. Understanding the encoding of scale can help, for instance, to debug deep models, to modify how scale is compressed and to compare its representation across different architectures. The analysis equivariance to transformations is important to ensure that the network preserves important information (not discarding scale if relevant), while evaluating the redundancy of the representation can be used, for instance, for pruning or compression purposes and ensuring generalization to new data.

Acknowledgment

Work supported by the SNSF grant 205320_179069 and the Horizon 2020 project PROCESS grant 777533.

References

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[3] A. Depeursinge, V. Andrearczyk, P. Whybra, J. van Griethuysen, H. Müller, R. Schaer, M. Vallières, and A. Zwanenburg, "Standardised convolutional filtering for radiomics," *preprint arXiv:2006.05470*, 2020.

[4] T. Lompech, M. Graziani, A. Depeursinge, and V. Andrearczyk, "On the scale invariance in state of the art CNNs trained on ImageNet," in *(submitted)*, 2020.

[5] M. Graziani, T. Lompech, A. Depeursinge, and V. Andrearczyk, "Interpretable CNN pruning for preserving scale-covariant features in medical imaging," in *iMIMIC at MICCAI*, 2020.

[6] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *Deep Learning workshop at ICML*, 2015.

[7] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Computer Vision and Pattern Recognition*, 2017, pp. 6541–6549.

[8] M. Aubry and B. C. Russell, "Understanding deep features with computer-generated imagery," in *International Conference on Computer Vision*, 2015, pp. 2875–2883.

[9] K. Lenc and A. Vedaldi, "Understanding image representations by measuring their equivariance and equivalence," in *Conference on Computer Vision and Pattern Recognition*, 2015, pp. 991–999.

[10] A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?" *Journal of Machine Learning Research*, vol. 20, no. 184, pp. 1–25, 2019.

[11] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Interpretable transformations with encoder-decoder networks," in *IEEE International Conference on Computer Vision*, 2017.

[12] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 30:31–30:57, 2018.

[13] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," in *International Conference on Learning Representations 2017 Workshop*, 2016.

[14] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV)," in *ICLR*, 2018.

[15] M. Graziani, V. Andrearczyk, and H. Müller, "Regression concept vectors for bidirectional explanations in histopathology," in *iMIMIC at MICCAI*, 2018, pp. 124–132.

[16] M. Graziani, V. Andrearczyk, S. Marchand-Maillet, and H. Müller, "Concept attribution: Explaining cnn decisions to physicians," *Computers in Biology and Medicine*, p. 103865, 2020.

[17] C. J. Cai, E. Reif, N. Hegde, J. Hipp, B. Kim, D. Smilkov, M. Wattenberg, F. Viegas, G. S. Corrado, M. C. Stumpe *et al.*, "Human-centered tools for coping with imperfect algorithms during medical decision-making," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[18] M. Graziani, H. Müller, and V. Andrearczyk, "Interpreting intentionally flawed models with linear probes," in *SDL-CV workshop at the IEEE International Conference on Computer Vision*, 2019.

[19] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou, "Fixing the train-test resolution discrepancy," in *Advances in Neural Information Processing Systems*, 2019.

[20] B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable basis decomposition for visual explanation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 119–134.

[21] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Advances in neural information processing systems*, 2016, pp. 4898–4906.

A 2.5D Vehicle Odometry Estimation for Vision Applications

Paul Moran *[1], Leroy-Francisco Periera[2], Anbuchezhiyan Selvaraju[2], Tejash Prakash[2], Pantelis Ermilios[1], John McDonald[1], Jonathan Horgan[1], and Ciarán Eising[1,3]

[1]*Valeo Vision Systems, Dunmore Road, Tuam, Co. Galway, Ireland*
[2]*Valeo India, Rajiv Gandhi Salai, Navalur, Chennai - 600 130, Tamil Nadu, India*
[3]*Department of Electronic and Computer Engineering, University of Limerick, Ireland*

Abstract

This paper proposes a method to estimate the pose of a sensor mounted on a vehicle as the vehicle moves through the world, an important topic for autonomous driving systems. Based on a set of commonly deployed vehicular odometric sensors, with outputs available on automotive communication buses (e.g. CAN or FlexRay), we describe a set of steps to combine a planar odometry based on wheel sensors with a suspension model based on linear suspension sensors. The aim is to determine a more accurate estimate of the camera pose. We outline its usage for applications in both visualisation and computer vision.

Keywords: Camera, Odometry, Calibration, Navigation, Localization

1 Introduction

A real-time estimate of a vehicle's pose in a world coordinate system is important for Advanced Driver-Assistance Systems (ADAS) and autonomous vehicles. Accurately measuring the pose of sensors attached to the vehicle is also vital for perception. Using accurate 3D odometry, the task of finding the sensor pose with known rigid body extrinsics within the world coordinate system is trivial through a simple coordinate system change. However, limited to only 2D odometry the pose of the sensor may be inaccurate due to suspension changes which are unaccounted for in 2D (planar) odometry. For instance, if there is a heavy loading in the vehicle, causing a suspension change from the nominal, then the system will have an inaccurate estimate of the camera extrinsics. This will cause an issue for any vehicular mapping system as without an accurate sensor ego pose the system cannot accurately localise perceived objects from that sensor relative to the vehicle.

Vehicle odometry can be estimated from various sensor types. Laser scanners can be used to estimate 2D odometry (Jaimez et al., 2016). Despite their accurate odometry estimates, due to their expense they are not universally deployed in vehicles. Visual odometry remains a significant area of research, and though it can give very high accuracy, the issue of scale resolution is still an unsolved topic (Liu et al., 2018). High-grade Global Navigation Satellite Systems (GNSS) and Inertial Navigation Systems (INS) can offer greater accuracy than wheel-based odometries (Aqel et al., 2016), but again are expensive and as such fail with ubiquitous deployment on vehicles (Gonzalez and Dabove, 2019). Visual-Inertial Odometry (Scaramuzza and Zhang, 2019) is a method that combines visual and inertial sources of odometry to overcome limitations of both sensor types. However, it suffers from the same problems of universal deployment as INS. Hence, wheel-based odometry remains popular, and continues to be an area of research in robotics and autonomous vehicles (Brunker et al., 2018). For further information, the reader is referred to Mohamed et al. (2019), who give a very complete overview of odometry in autonomous systems.

Wheel encoders are commonly deployed on vehicles (Brossard and Bonnabel, 2019) where the sensor information is broadcast on the vehicle's system bus (CAN, FlexRay, Ethernet). Typically, these encoders

*paul.moran@valeo.com

 ISBN 978-0-9934207-5-7

utilise Hall effect sensors (Popovic, 2003) though research continues into potentially better alternatives (Shah et al., 2019). To detect changes in heading, two common sensor types are deployed in vehicles: steering angle sensors (e.g. rotary potentiometer (Todd, 1975)) and/or yaw rate sensor (e.g. gyroscope (Passaro et al., 2017)).

Traditionally, wheel-based odometry was only used to provide a planar motion estimate of the vehicle; calculating an odometry estimate with only three degrees of freedom. Here we propose to augment the planar 2D wheel odometry by using sensors that measure the current level of the suspension of the vehicle using linear potentiometers (Todd, 1975), giving a much better estimate of the true extrinsic position of the camera for a given moment in time. These advanced sensors are becoming commonplace on vehicles with adjustable suspension for altering ride height. The technique described does not give a full 3D odometry estimate, but could metaphorically be referred to as a 2.5D estimate of odometry of the vehicle. In this paper, we use the yaw rate sensor, as this enables us to avoid using a specific model of vehicle steering (e.g. Ackermann), which may have inaccuracies.

This paper is organised as follows. In the following section, we discuss the motion of the vehicle on the ground plane, and the motion of the sensors due to changing suspension, and how both can be combined. In Section 3, we provide some results, examining the accuracy of the planar odometry, the behaviour of the sensors, and some results in application for human visualisation and computer vision.

2 Proposed Method

We define the coordinate system of the vehicle to have the origin at the rear axle, X^v-axis pointing forward in the direction of the vehicle, Z^v-axis pointing upward, roughly orthogonal to the ground plane, and Y^v-axis in the direction of left hand turning (X^v-axis is shown in Figure 1). We track the position of the vehicle in a world coordinate system with axes X^w, Y^w and Z^w. For vectors, we use the super-scripts v, w and c to indicate the coordinate system in which the vector is defined: the vehicle, world and sensor (typically camera) coordinate system. We use $\theta(t)$ to denote a continuous function in t and $\theta'(t)$ to denote it's derivative with respect to t. When using sampled data, we use numbered subscripts instead of parentheses, e.g. θ'_1, θ_1, t_1, etc. In this paper, we don't discuss the uncertainty of the model. However, uncertainty of wheel odometry models is described in some detail by Ben-Ari and Mondada (2017), and is applicable here.

2.1 Heading angle

The heading angle $\theta(t)$ (with radians as a unit, for example) at any point in time t (with seconds as a unit, for example) is given by integrating the continuous yaw rate function:

$$\theta(t_1) = \int_0^{t_1} \theta'(t)\,dt \tag{1}$$

Ignoring the constant of integration, we get absolute heading in the coordinate system of the position of the vehicle at time zero (i.e. the power on of the vehicle, or the start of running of the piece of implemented software). In the general case, we do not have the underlying yaw rate function, but rather we only have samples from the yaw rate sensor between the two times, t_1 and t_2. Thus, we can accumulate iteratively:

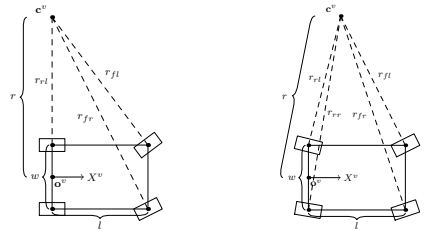
$$\theta(t_2) = \left(\int_{t_1}^{t_2} \theta'(t)\,dt\right) + \theta(t_1) \tag{2}$$

As the sensors are sampled (i.e. the continuous function is not available in reality), this is approximated as

$$\Delta\theta = \frac{\theta'_1 + \theta'_2}{2}(t_2 - t_1), \qquad \theta(t_2) \approx \theta_2 = \Delta\theta + \theta_1 \tag{3}$$

where θ'_1 and θ'_2 are the yaw rate samples (for example, with radians per second as a unit) at the times t_1 and t_2 respectively, and are sampled approximations of the continuous function $\theta'(t)$. Estimating $\Delta\theta$ is done by taking the average of the two yaw rate samples (in rad/s) and multiplying by the equivalent time difference to get the

 ISBN 978-0-9934207-5-7



(a) (b)

Figure 1: Odometry models: a) Fixed rear steering, and b) adaptive rear steering. All points on the vehicle turn through a fixed instantaneous turning centre ($\mathbf{c}^v$). w is the distance between wheel pairs, and l is the wheelbase. heading angle (in rad). This is iterative, as a new sample arrives θ_2 is assigned to the previous sample, θ_1, and θ_2 takes the value of the new sample. Thus from the yaw rate sensor, we can extract the absolute heading angle θ_t at any sample time t, and the delta heading angle from the previous sample $\Delta\theta$.

2.2 Planar Displacement

Planar odometry has an instantaneous centre of rotation (Jazar, 2008) as shown in Figure 1. The integration time is short enough (~10-20 ms on a system bus) to consider the curvature to be constant between two samples. The vehicle, in the two dimensions of the plane, can be considered to be rigidly moving. The relative positions of the points of contact of the tyres with the surface of the road remain constant. Hence, if the vehicle moves between two points in time, t_1 and t_2, and the angle, $\Delta\theta$, then the distance moved of any point is

$$d = r\Delta\theta \tag{4}$$

where r is the distance of the point on the vehicle to the instantaneous centre of rotation. $\Delta\theta$ is given by (3). Given a set of four samples of d for the four wheels of the vehicle $\{d_{rl}, d_{rr}, d_{fl}, d_{fr}\}$, the estimate of the distance from the wheel position to the turning centre (Figure 1) is given by

$$r_i = \frac{d_i}{\Delta\theta}, \qquad i \in \{rl, rr, fl, fr\} \tag{5}$$

For the case with fixed rear steering (Figure 1(a)), we can then get four estimates of the distance of the vehicle datum to the turning centre, with the average being our final estimate.

$$r_1 = r_{rl} - w/2, \qquad r_2 = r_{rr} + w/2, \qquad r_3 = \sqrt[+]{r_{fl}^2 - l^2} - w/2, \qquad r_4 = \sqrt[+]{r_{fr}^2 - l^2} + w/2 \tag{6}$$

$$r = \frac{r_1 + r_2 + r_3 + r_4}{4} \tag{7}$$

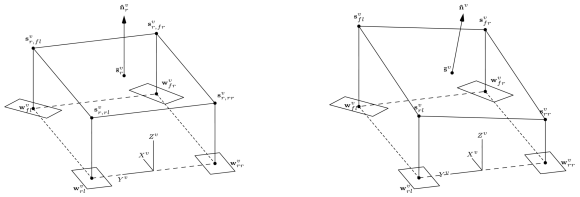
The distance from the centre of motion to the datum r is estimated using the average of the four extracted radii. w is the distance between the wheels, and l is the length from the front wheel pair to the rear wheel pair (i.e. wheelbase). The yaw rate is signed to give the "left" or "right" motion of the vehicle, and the wheel distances $\{d_{rl}, d_{rr}, d_{fl}, d_{fr}\}$ are signed to give the "forward" or "backward" motion of the vehicle. The instantaneous centre of rotation is therefore $\mathbf{c}^v = [0, r, 0]^\top$. In the case of adaptive rear steering (Figure 1(b)), we have two free parameters for $\mathbf{c}^v = [c_x^v, c_y^v, 0]^\top$. We solve this using least squares. The error function is given by

$$E(\mathbf{c}^v) = \sum_i \left|\mathbf{w}_i^v - \mathbf{c}^v\right|_2^2 - r_i^2, \qquad i \in \{rl, rr, fl, fr\} \tag{8}$$

and solving the partial differential equations $\delta E(\mathbf{c}^v)/\delta c_x^v = 0$ and $\delta E(\mathbf{c}^v)/\delta c_y^v = 0$ to obtain the estimate for $\mathbf{c}^v$. $\mathbf{w}_i^v$ indicates the position, in the vehicle coordinate system, of each of the wheels of the vehicle, given by appropriate combinations of w and l. r_i is from (5). Given the estimate of $\mathbf{c}^v$, the datum distance is simply

$$r = |\mathbf{c}^v|_2 \tag{9}$$

 ISBN 978-0-9934207-5-7



(a) (b)

Figure 2: The suspension plane models. a) settled state, and b) with loading

The motion vector, in vehicle coordinates, is given by

$$\Delta\mathbf{p}^v = [r\sin\Delta\theta, r\cos\Delta\theta, 0]^\top \tag{10}$$

where r is estimated from (7) or (9) as appropriate, and $\Delta\theta$ is estimated from (3). Given the heading angle, θ_1, at time t_1, the overall position of the vehicle at a given time t_2 is given by the accumulation

$$\mathbf{p}_2^w = {}^v\mathbf{R}_1^w \Delta\mathbf{p}^v + \mathbf{p}_1^w \tag{11}$$

where ${}^v\mathbf{R}_1^w$ is the 3×3 rotation matrix equivalent of the heading angle, θ_1, the rotation about the Z^w-axis. This is accumulative, so in the next iteration of the odometry calculation, $\mathbf{p}_2^w$ is assigned to $\mathbf{p}_1^w$.

2.3 Suspension model

A sensor (e.g. a camera) located on a vehicle has a particular set of extrinsic calibration parameters (rotation and translation) in the vehicle coordinate system. Note, calibration is usually done against the rigid coordinate system of the vehicle body, which doesn't take into account the pitching, rolling and settling of the vehicle suspension. $\mathbf{s}_i^v$ is the suspension point in the settled state. $\mathbf{s}_i^v$ is obtained by taking the wheel positions $\mathbf{w}_i^v$, and setting the z component to the height obtained from the calibrated linear potentiometers (Figure 2(a)). That is, if we set h_i as the set of heights from the sensors, then $\mathbf{s}_i^v = [w_{i,x}^v, w_{i,y}^v, h_i]^\top$. With no load on the vehicle, or no acceleration, the suspension will be in a settled state. The points form a plane in the vehicle coordinate system, defined by a normal vector $\hat{\mathbf{n}}_r^v$ and a reference point $\bar{\mathbf{s}}_r^v = \frac{1}{4}\sum \mathbf{s}_i^v$, which can be obtained using ordinary least squares. In all cases above, $i \in \{rl, rr, fl, fr\}$. In live operation, the suspension will change (Figure 2(b)). We can use the exact same procedure to extract a live description of the suspension plane model with the normal vector $\hat{\mathbf{n}}^v$ and reference point $\bar{\mathbf{s}}^v$. Only the z component of $\bar{\mathbf{s}}^v$ will be different compared to $\bar{\mathbf{s}}_r^v$, as the positions of the wheels $\mathbf{w}_i^v$ do not change with suspension changes. In order to combine the suspension changes with the odometry, we wish to represent it as a rotation matrix ${}^v\mathbf{R}_s^v$ and a translation vector $\mathbf{t}_s^v$. The translation is straightforwardly

$$\mathbf{t}_s^v = \bar{\mathbf{s}}_r^v - \bar{\mathbf{s}}^v \tag{12}$$

The rotation matrix is given by the axis-angle formula (recalling that $\hat{\mathbf{n}}^v$ and $\hat{\mathbf{n}}_r^v$ are both unit vectors):

$$\mathbf{a} = [a_x, a_y, a_z]^\top = \hat{\mathbf{n}}^v \times \hat{\mathbf{n}}_r^v, \qquad s = |\mathbf{a}|, \qquad c = \hat{\mathbf{n}}^v \cdot \hat{\mathbf{n}}_r^v$$

where s and c are the sine and cosine of the angle between $\hat{\mathbf{n}}^v$ and $\hat{\mathbf{n}}_r^v$. Then

$${}^v\mathbf{R}_s^v = \begin{bmatrix} c + a_x^2(1-c) & a_x a_y(1-c) - a_z s & a_x a_z(1-c) + a_y s \\ a_x a_y(1-c) + a_z s & c + a_y^2(1-c) & a_y a_z(1-c) - a_x s \\ a_x a_z(1-c) + a_y s & a_y a_z(1-c) + a_x s & c + a_z^2(1-c) \end{bmatrix}$$

Some notes on the assumptions of this model. Firstly, points on the vehicle body that are planar will remain planar under different suspension configurations. While there can be some flex in vehicle body, for the most part it can be considered rigid, and thus this assumption is valid. Secondly, the different suspension configurations cause our reference points to move vertically. Actually, this is not the case, as a changing suspension will cause a rotation of the vehicle body. However, vertical motion will dominate over lateral motion, and thus we can ignore the lateral motion of the reference points.

2.4 Sensor calibration

Sensors on the vehicle have an extrinsic calibration represented as a rotation matrix ${}^{v}\mathbf{R}_e^c$ and a position vector $\mathbf{c}_e^v$, in vehicle coordinates (Choi et al., 2018). Typically, the calibration procedure extracts the ${}^{v}\mathbf{R}_e^c$ and $\mathbf{c}_e^v$ of the cameras against an external reference, such as a local road plane defined approximately by the points of contact of the 4 wheels with the ground, or ground markings on such a surface, taking into account only a nominal reference suspension. This leads to a definition of the sensor calibration against the road plane rather than against the vehicle body. Thus, if the calibration runs when there is a heavy loading in the vehicle, causing a suspension change from the nominal, then the system will not calibrate for a "true" extrinsic position. To solve this, one must account for the suspension during the calibration procedure, and this gives us the extrinsic camera parameters considering the "nominal" or reference suspension. As described previously, we can get the rotation and translation due to suspension changes from the nominal (${}^{v}\mathbf{R}_s^v$, $\mathbf{t}_s^v$). During the calibration procedure, we get a calibrated rotation and translation (${}^{v}\mathbf{R}_{cal}^c$, t_{cal}). However, these include the offsets due to the suspension, as the algorithm runs when the suspension is different from nominal. To get the true extrinsic camera positions:

$$ {}^{v}\mathbf{R}_e^c = {}^{v}\mathbf{R}_s^{v\top}\, {}^{v}\mathbf{R}_{cal}^c \tag{13} $$

and similarly for the calibrated camera position $\mathbf{c}_e^v$. This is done for each camera, and then gives the calibration against the nominal or reference suspension setting for each camera.

2.5 Combining motions

The overall pose of the camera in the vehicle coordinate system is therefore given by the composition of the suspension model and calibration rotations

$$ {}^{v}\mathbf{R}_p^c = {}^{v}\mathbf{R}_e^c\, {}^{v}\mathbf{R}_s^v \tag{14} $$

The position of the camera in vehicle coordinates can be given by (note $\mathbf{t}_s^v = [0, 0, h]^\top$):

$$ \mathbf{c}_p^v = {}^{v}\mathbf{R}_s^v(\mathbf{c}_e^v + \mathbf{t}_s^v) \tag{15} $$

The position of the sensor in the world coordinate system can then be given by

$$ {}^{w}\mathbf{R}_p^c = {}^{v}\mathbf{R}_p^c\, {}^{w}\mathbf{R}^v, \qquad \mathbf{c}_p^w = {}^{v}\mathbf{R}^w\mathbf{c}_p^v + \mathbf{p}^w \tag{16} $$

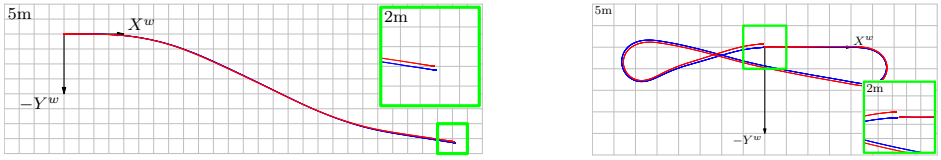
with ${}^{w}\mathbf{R}^v$ obtained from the odometry heading angle (3), and $\mathbf{p}^w$ is the vehicle position from equation (11).

3 Results

Ground truth isn't available for 2.5D odometry, or for suspension in general, as DGPS is the only ground truth sensor available in our system. With that in mind, we compare the planar 2D odometry to the ground truth DGPS, and then subjectively examine the performance of 2.5D odometry in the context of two vision applications; visualisation (top view) and computer vision (motion segmentation).

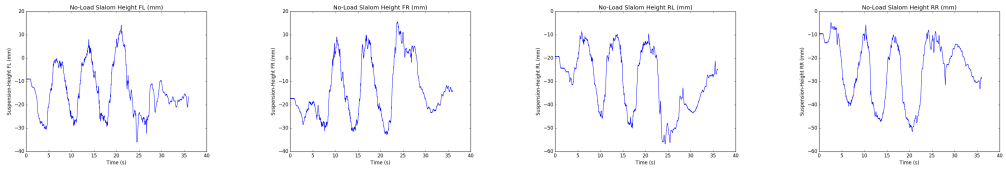
3.1 Planar Odometry

Figure 3 show the trajectory from two vehicle manoeuvres. The error for the simpler manoeuvre at the end of the trajectory is 0.46 m (Figure 3(a)), whereas the more complex manoeuvre has an overall drift of only 0.82 m (Figure 3(b)). Hence, this shows that 2D planar odometry is a sufficiently accurate input to our model.



(a) (b)

Figure 3: Calculated vehicle position (red) versus ground truth (DGPS) (blue): "standard" manoeuvre (a) and complex manoeuvre (b).



(a) front left (b) front right (c) rear left (d) rear right

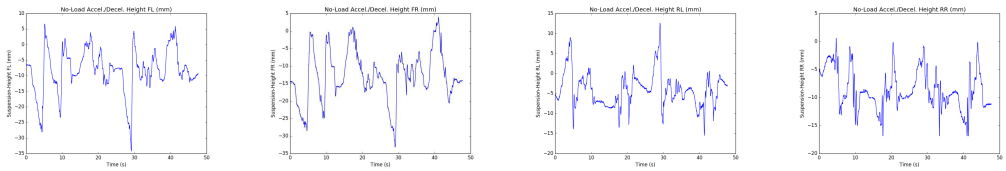
Figure 4: Plots of the suspension sensor height as a function of time for slalom motion.

3.2 Suspension Sensor Behaviour

Experiments were performed to test the accuracy of the algorithm that compensates calibration based on the suspension. The first experiment involved checking the stability and accuracy of the input ego-vehicle suspension data. The input data comes from sensors mounted at the arches of the four wheels: Front-left (fl), Front-Right (fr), Rear-Left (rl), and Rear-Right (rr). These sensors measure changes in their vertical height from the ground plane. Two particular cases were studied: slalom motion (driving in arcs or zig-zags), and acceleration and deceleration. The data was recorded from the CAN bus of a test vehicle. Figures 4 & 5 show the heights (mm) of each of the wheel arch sensors plotted as a function of time. For the slalom motion it can be seen that the peaks and troughs of the plots of the suspension sensors mounted on the left and right side of the ego-vehicle were out of phase (Figure 4) i.e. the peaks in the left wheel pair occur at the same moment in time as the troughs in the right wheel pair. This agrees with the physics of the use case, namely centripetal force. During the slalom the weight of the ego-vehicle is transferred directly to one side. Hence, the ego-vehicle becomes unbalanced with one side raised and the other side lowered. Similarly, for acceleration and deceleration (Figure 5) it was seen that the plots of the front and rear side are out of phase. Again, in agreement with the physics of the transfer of loading of the vehicle. During acceleration from rest the weight pushes down at the rear of the ego-vehicle and the front pitches upwards. Whereas during braking the opposite effect occurs.

3.3 Visualisation

We generate a top-view of the vehicle's surroundings using four fisheye cameras on the vehicle: front, rear, and the two wing-mirrors. The aim is to analyse the visual impact of utilising the suspension-corrected extrinsic parameters compared to the nominal extrinsic parameters. Table 1 shows the nominal and suspension-



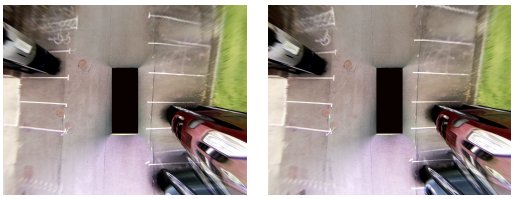
(a) front left (b) front right (c) rear left (d) rear right

Figure 5: Plots of the suspension sensor height as a function of time for acceleration and deceleration.

 ISBN 978-0-9934207-5-7

	Nominal				Suspension-Compensated			
	FV	RV	MVL	MVR	FV	RV	MVL	MVR
Height (mm)	603.23	880.29	950.97	966.74	592.12	887.23	944.20	964.89
Rot. X ()	91.08	64.24	61.67	62.11	92.48	62.66	61.50	62.16
Rot. Z1 ()	89.96	-90.71	167.94	2.87	91.23	-89.58	168.03	3.96
Rot. Z2 ()	-0.53	0.34	3.51	-6.42	-0.40	0.35	3.32	-6.53

Table 1: Nominal and suspension-compensated extrinsic camera calibration values.



(a) (b)

Figure 6: Top-view images generated using calibrated extrinsics (a), and suspension-corrected extrinsics (b).

compensated extrinsic calibration values for the scene shown in Figure 6. The discontinuities between the parts of the images created by different cameras is quite evident in the top-view without suspension corrected extrinsic parameters.

3.4 Computer Vision

Mariotti and Hughes (2019) describe a geometric means of motion segmentation, and mention explicitly that the results in that paper are generated from a three degrees of freedom odometry, giving the position of the sensor in a world coordinate system. Here we briefly show some results of just using the planar odometry (Figure 7(b)) versus the planar odometry incorporating suspension sensors (Figure 7(c)). Figure 7(a) shows the original frame. In Row I, the vehicle is turning with rolling of the vehicle on the suspension. In Row II, the vehicle is accelerating heavily, showing significant pitching. In both cases, it can be seen that the error in the motion segmentation map is significantly lower when suspension is taken into account.

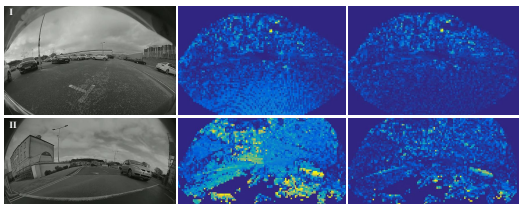


Figure 7: Motion segmentation maps: planar odometry (b), and planar odometry incorporating suspension (c). Motion likelihood increases from dark blue to red. The planar odometry results show that static objects are incorrectly predicting with much larger motion likelihoods since motion due to suspension is not compensated.

4 Conclusion

We have presented an odometry estimation algorithm using a set of sensors (yaw rate, wheel speed and suspension) commonly available on some modern, commercially available vehicles. It is computationally inexpensive, as the amount of data to process is minimal, but still provides significant improvement compared to just considering a planar odometry. This could be considered a 2.5D odometry, as it does not give a full 3D odometry (like from visual odometry) but it offers more than just the case of planar (2D) odometry. The results presented demonstrate that the integration error of the planar odometry is low. For visualisation applications, such as top-view, the use of the suspension sensors reduces stitching artefacts in the overlap regions between multiple cameras. The improved sensor extrinsic measure, relevant to all on board sensors, is key for perception and thus building precise environmental maps for automated and autonomous driving systems. For computer vision, the

2.5D odometry offers an advantage in the suppression of false positives, in the case that the computer vision requires an odometry input. Future work will consist of more rigorous experiments to determine the accuracy of the algorithm, and to test its use as an input to other computer vision applications. Visual-Inertial Odometry is an interesting area of development in robotics, in particular. Some further future work may be in integrating the 2.5D odometry with visual odometry, in the same way that low cost inertial sensors are integrated with visual odometry in Visual-Inertial Odometry. This would integrate the work presented in this paper entirely into a Visual SLAM environment.

References

Aqel, M. O. A., Marhaban, M. H., Saripan, M. I., and Ismail, N. B. (2016). Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, 5.

Ben-Ari, M. and Mondada, F. (2017). Robotic motion and odometry. In *Elements of Robotics*. Springer.

Brossard, M. and Bonnabel, S. (2019). Learning Wheel Odometry and IMU Errors for Localization. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Montreal, Canada.

Brunker, A., Wohlgemuth, T., Frey, M., and Gauterin, F. (2018). Odometry 2.0: A slip-adaptive eif-based four-wheel-odometry model for parking. *IEEE Transactions on Intelligent Vehicles*, 4(1):114–126.

Choi, K., Jung, H. G., and Suhr, J. K. (2018). Automatic calibration of an around view monitor system exploiting lane markings. *Sensors*, 18.

Gonzalez, R. and Dabove, P. (2019). Performance assessment of an ultra low-cost inertial measurement unit for ground vehicle navigation. *Sensors*, 19:3865.

Jaimez, M., Monroy, J. G., and Gonzalez-Jimenez, J. (2016). Planar odometry from a radial laser scanner. a range flow-based approach. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Jazar, R. N. (2008). *Vehicle Dynamics: Theory and Application*. Berlin: Springer.

Liu, L., Li, H., Dai, Y., and Pan, Q. (2018). Robust and efficient relative pose with a multi-camera system for autonomous driving in highly dynamic environments. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2432–2444.

Mariotti, L. and Hughes, C. (2019). Spherical formulation of moving object geometric constraints for monocular fisheye cameras. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*.

Mohamed, S. A. S., Haghbayan, M.-H., Westerlund, T., Heikkonen, J., Tenhunen, H., and Plosila, J. (2019). A survey on odometry for autonomous navigation systems. *IEEE Access*, 7.

Passaro, V. M. N., Cuccovillo, A., Vaiani, L., Carlo, M., and Campanella, C. E. (2017). Gyroscope technology and applications: A review in the industrial perspective. *Sensors*.

Popovic, R. S. (2003). *Hall Effect Devices, 2 edition*. CRC Press.

Scaramuzza, D. and Zhang, Z. (2019). Visual-inertial odometry of aerial robots. In *Encyclopedia of Robotics*. Springer.

Shah, H., Haldar, S., Ner, R., Jha, S., and Chakravarty, D. (2019). Ground vehicle odometry using a non-intrusive inertial speed sensor. In *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, pages 120–125.

Todd, C. D. (1975). *The Potentiometer Handbook: Users' Guide to Cost-Effective Applications*. McGraw-Hill.

CNN based Color and Thermal Image Fusion for Object Detection in Automated Driving

Ravi Yadav[1], Ahmed Samir[2], Hazem Rashed[2], Senthil Yogamani[3] and Rozenn Dahyot[1]

[1]*Trinity College Dublin, Ireland* [2]*Valeo R&D Cairo, Egypt* [3]*Valeo Vision Systems, Ireland*

Abstract

Visual spectrum camera is a primary sensor in an automated driving system. It provides a high information density at a low cost. Visual perception is extensively studied in the literature and it is a mature component deployed in existing commercial vehicles. Its main disadvantage is the performance degradation in low light scenarios. Thermal cameras are increasingly being used to complement cameras for dark conditions like night time or driving through a tunnel. In this paper, we explore CNN based fusion architecture for object detection. We explore two automotive datasets which provide data for both these sensors namely KAIST multispectral pedestrian dataset and FLIR thermal object detection dataset. We train baseline Faster-RCNN models for color only and thermal only models on KAIST dataset. Color model outperforms Thermal in day conditions and Thermal model outperforms color in night conditions illustrating their complementary nature. We construct a simple mid-level CNN fusion architecture which performs significantly better than the baseline models. We observe an improvement of 0.62% in miss rate compared to existing methods. We also explored the more recent FLIR dataset. Because of the vastly different resolution, aspect ratio and field of view of the color and thermal images provided, our simple fusion architecture did not perform well pointing out the need for further research in this area.

Keywords: Automated Driving, Visual Perception, Object Detection, Sensor Fusion, Deep Learning

1 Introduction

Automated driving is a rapidly progressing field where recent advances in deep learning are getting productized. Multiple sensors including camera, lidar, radar, ultrasonic, etc are commonly used to improve reliability and safety. Cameras are the primary sensor because of the dense semantic and geometric information it provides. Cameras provide a wide variety of visual perception tasks such as semantic segmentation [Briot et al., 2018], moving object detection [Yahiaoui et al., 2019], depth estimation [Kumar et al., 2018], re-localisation [Milz et al., 2018], soiling detection [Uřičář et al., 2019], etc. However, all these perception algorithms degrade severely during low light scenarios. In particular, night time scenarios with illuminance levels less than 1 lux have poor performance.

Thermal sensors complement color sensors for night time driving scenarios. They have already been deployed in several existing cars mainly for providing night vision on a dashboard for the driver and more recently supporting object detection. Thermal sensors can also aid in sun glare scenarios which is a common problem. In particular, thermal signatures of pedestrians and animals are unique and aid their detection. In this work, we explore the fusion of thermal images with color images for object detection. It is a relatively less explored area compared to the well explored Lidar and camera fusion [El Madawi et al., 2019, Rashed et al., 2019b, Ravi Kiran et al., 2018]. The contributions of this work include:

(1) Construction of CNN based fusion architectures and unimodal baselines.

(2) Ablation study of the three networks for day and night time scenes separately.

(3) Experimentation on two automotive datasets namely KAIST and FLIR. State of the art results on KAIST.

The rest of the paper is organized as follows: Section 2 reviews the related work in usage of thermal imagers in automated driving and fusion architectures using thermal images. Section 3 details the construction of proposed fusion architecture and its associated unimodal baselines. Section 4 discusses the experimental results in KAIST and FLIR datasets. Finally, section 5 provides concluding remarks.

2 Related Work

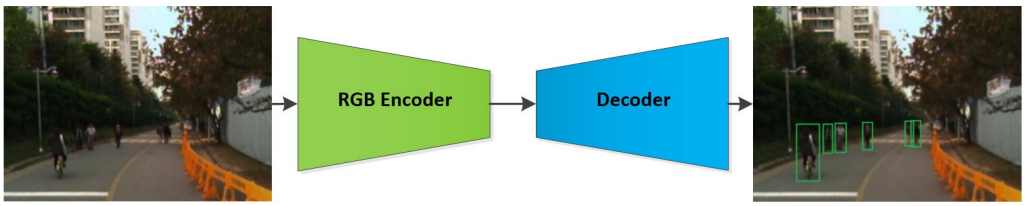
Thermal sensors in Automated Driving Systems: The infrared spectrum is divided into several specific areas namely Near-infrared (NIR), Short-wavelength infrared (SWIR), Mid-wavelength infrared (MWIR), Long-wavelength infrared (LWIR) and Far infrared (FIR). MWIR and LWIR are often called as Thermal Infrared (TIR) because they passively detect radiation emitted by certain objects like pedestrians, animals and vehicles [Miethig et al., 2019]. The amount of radiation corresponds to temperature and thus the term thermal. NIR and SWIR on the other hand are used as reflected infrared where an active source illuminates the scene and the reflection is measured off the objects. Both these technologies have been deployed in automotive systems primarily to improve night vision. Toyota Land-Cruiser had a night view system deployed in 2002 using active illumination. Mercedes-Benz had a night view assist system released in 2005. These systems simply displayed the infrared video feed on the dashboard for the driver to visualize the scene better. More recently, thermal infrared are becoming more popular because they do not need active illumination and are therefore cheaper and simpler system design. BMW 7 series had a night vision feature introduced in 2005 and animal detection was added in 2013. Audi A8 introduced night vision assistant feature in 2010. Both these systems were based on thermal sensors.

In this paper, we focus on thermal images. Thermal images represent the quantities of infrared (IR) energy that the object emits, transmits and reflects [Correa et al., 2012]. Carbon dioxide is responsible for majority of absorption of infrared but it is less pronounced for thermal images which eliminates the need for active illumination [Airouche et al., 2012]. However, thermal images still suffer from limited range of perception relative to cameras. Thermal images are particularly suited for identifying pedestrians and animals as they have a higher intensity. It can also be useful for identifying vehicles. The detection is particularly useful for dimly lit scenes where cameras don't perform well. Additionally, it also provides scene illumination invariance in brightly lit scenes. For example, it does not get affected by sun glare which is a common problem in automated driving. It is also invariant to shadows. However, it could be affected by external temperature changes. For example, intensity difference is lower in a hot desert environment or in the presence of an object emitting heat.

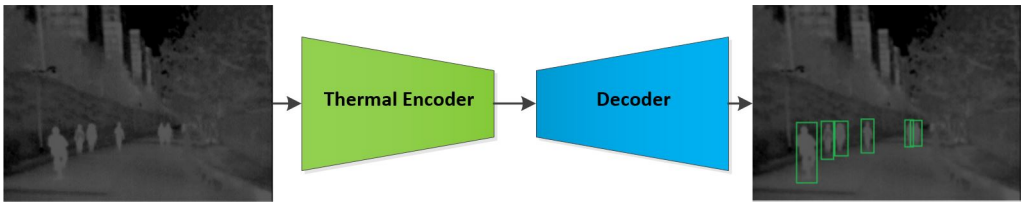
Fusion based Object Detection: Multimodal fusion is common in automated driving to provide higher reliability and hence a safer system. Typically they are used to combine heterogeneous sensors like radar or lidar with camera images and a late fusion scheme is used. In case of thermal and camera fusion, both produce images and it is a simpler form of fusion. As both sensors produce images, they can be aligned relatively easily for early or mid level fusion. Visual and thermal images are among the primary forms of data used for animal detection and tracking [Chen et al., 2008][Gade and Moeslund, 2014]. Because of the large diversity in appearance of animals, it is harder to build a good detector based on RGB cameras. The thermal cameras make this task easier because of the higher intensity which animals produce. Thermal cameras are also commonly used to detect pedestrians in night time scenes in surveillance systems [Gade and Moeslund, 2014] and in automotive systems [Binelli et al., 2005].

Bertozzi et al. [Bertozzi et al., 2003] developed a pedestrian detection system based on SVM using morphological features. Choi et al. [Choi et al., 2016] used independent algorithms for each image using edge-box object proposals and fused the output by concatenation which was further classified by support vector regression. Wagner et al. [Wagner et al., 2016] used RCNN for fusion of output proposals using late fusion and early fusion. Wang et al. [Wang et al., 2018] designed a specific features based on thermal image intensity maps of pedestrians and used a classifier. Konig et al [Konig et al., 2017] used Faster RCNN and CNN based fusion architecture for pedestrian detection from thermal and camera images. Li et al [Li and Wu, 2018] used a complex fusion architecture using CNN registration of multimodal features using dense CNN blocks.

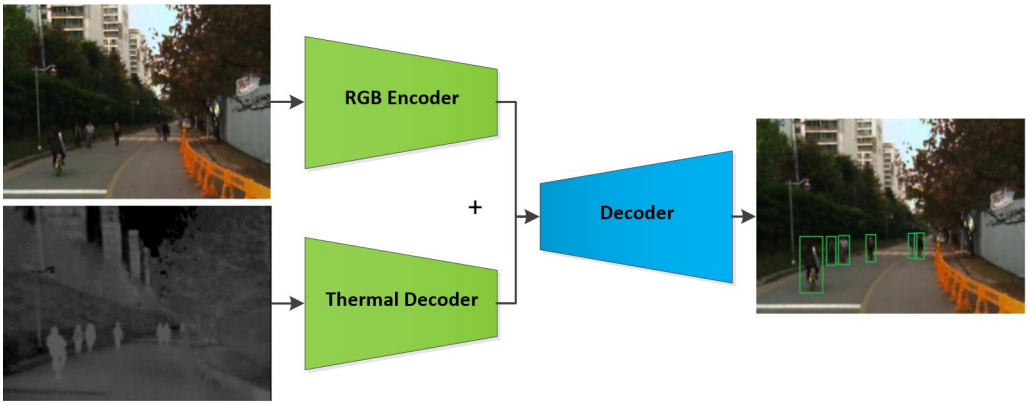
 ISBN 978-0-9934207-5-7



(a) Color image baseline network



(b) Thermal image baseline network



(c) Two Stream RGB and Thermal encoder fused network

Figure 1: Three types of architectures constructed and tested in the paper. (a) and (b) are baselines using RGB and Thermal only. (c) is proposed fusion architecture.

3 Proposed Fusion Architecture

In this section, we discuss the details of the different object detection networks used in this paper. We construct two unimodal baseline networks which work on color and thermal only. Color network (Figure 1 (a)) is based on a standard Faster RCNN object detection network. Thermal variant (Figure 1 (b)) was adapted to take it one intensity plane instead of the three color planes. We then construct a two-stream RGB+Thermal network (Figure 1 (c)). In this case, the two unimodal encoded feature tensor are combined using a summation junction. This leads to the fused feature tensor of same dimension which can be fed to the same dimensional decoder as per the baselines.

Unimodal Baselines: The unimodal baselines are used to evaluate the contribution of each input signal separately on object detection. We make use of the FasterRCNN network [Yang et al., 2017], a state-of-the-art architecture for object detection. As we had to train the thermal baseline from scratch, we choose to use a simpler VGG16 encoder for practical reasons instead of complex encoder like Resnet101. Input signal is fed into a VGG16 encoder where the outputs features are used to generate object proposals through a Region Proposal Network (RPN). Regions of Interest (ROIs) generated from RPN are fed into ROI pooling and classification stage to provide the final output. We train the same network on both inputs separately to serve as baseline for comparison with fusion method.

Two stream (RGB+Thermal) fusion network: Inspired from [Rashed et al., 2019c] [Rashed et al., 2019a] [Siam et al., 2018], we construct a two stream network which utilizes two parallel VGG16 encoders to extract appearance and thermal features separately. Our main design goal is to augment thermal image data into an

existing visual object detection system as an upgrade in higher cost vehicles. Thus we designed an encoder fusion network which preserves the encoder feature tensor of object detection network to enable multiple tasks to be implemented on camera side [Sistu et al., 2019] [Chennupati et al., 2019]. Fusion between feature maps is done through a summation junction for the last convolution layer outputs from both encoders providing an output volume of the same shape. The same decoder implemented in unimodel baselines is used to generate the final predictions. Significant improvement has been observed in results as demonstrated in Table 1. However, the two stream network is computationally more complex with more parameters compared to unimodal baselines due to the increased number of parameters in the encoder part. On the other hand, the proposed approach has the advantage of making use of the visual encoder for other tasks.

4 Experiments

In this section, we present the datasets used, experimental setup and results.

4.1 Datasets

KAIST [Hwang et al., 2015] is a multi-spectral pedestrian dataset that provides 95k images that are split into 50k for training and 45k for testing images. The dataset is captured in Korea during day and night times. It is designed particularly for fusion systems and thus they provide perfectly aligned thermal and visual images. However, the annotations are derived from continuous video sequence, hence it has redundant information. We removed every second image of the training dataset to speed-up the training and there was no observable degradation of performance. We have also eliminated objects which have a height of less than 50 pixels as suggested in [Li et al., 2019b]. These object samples are less informative and increase false positives .

FLIR [FLIR, 2018] is a dataset created by the thermal sensor manufacturer FLIR. The dataset comprised of 60% day and 40% night-time images taken while driving in California. 10k frames were provided with bounding box annotation for five classes namely person, car, bicycle, dog and other vehicles. Although synchronized camera images were provided, different resolution cameras were used at multiple instances ranging from 0.3MP to 3.1MP. The camera had a vastly different field of view and resolution and it was not possible to align by a simple edit like crop or resize.. The thermal intensity image had a bit-depth of 14-bits and 8-bit resolution. All these differences made the dataset very challenging for designing a fusion algorithm.

4.2 Experimental Results

Experimental Setup: For all experiments, Adam optimizer is used with a learning rate of $1e^{-5}$. L2 regularization is used to penalize loss function with a factor of value of $1e^{-4}$ to avoid over-fitting the data. The encoder is initialized with VGG pre-trained weights on ImageNet. Dropout with probability 0.5 is utilized for 1x1 convolutional layers. Log average miss rate is used as the performance metric as demonstrated in [Xu et al., 2017] for comparing performance of all the models.

Table 1 compares quantitative results of the proposed fusion algorithm with other algorithms reported on the KAIST dataset. We achieved an improvement of 0.62% in log average miss rate over the state of the art. [Guan et al., 2019] produces better results by augmenting semantic segmentation task and thus we leave it to maintain a fair comparison for purely object detection task methods. Furthermore, our architecture is very simple and can be improved significantly by incorporating a larger encoder and more sophisticated fusion schemes. Our algorithm could have benefited by the recent advances in training algorithms and hyperparameter tuning compared to previous algorithms published at least a year ago.

We also performed an ablation study to understand the variance in day and night scenarios comparing unimodal and fusion networks. As seen in Table 2, the thermal network achieved around 39% improvement over color network confirming the weakness of cameras in low light. For day time, color network achieved an improvement of 24% because of the richer texture it captures relative to thermal sensor. These results illustrate

 ISBN 978-0-9934207-5-7

Fusion Algorithm	**Log average miss rate**
ACF+T+THOG [Hwang et al., 2015]	54.40%
CMT-CNN [Xu et al., 2017]	49.55%
LateFusion CNN [Wagner et al., 2016]	43.80%
Halfway Fusion [Liu et al., 2016]	36.22%
Illumination aware Faster-RCNN [Li et al., 2019a]	29.99%
FusionRPN + BDT [Konig et al., 2017]	29.83%
IATDNN [Guan et al., 2019]	29.62%
Proposed Fusion	**29%**
IATDNN + Semantic Segmentation [Guan et al., 2019]	26.37%

Table 1: Quantitative comparison of proposed algorithm with other methods on KAIST dataset.

Test Set	**RGB**	**Thermal**	**Fusion**
Day	31%	55%	**26%**
Night	64%	25%	**32%**
All	47.7%	38%	**29%**

Table 2: Object Detection Results (log average miss rate) over KAIST Dataset.

the complementary nature of these two sensors for day and night. Proposed fusion network consistently outperforms both the unimodel networks demonstrating an efficient fusion scheme. Qualitative results are provided in Figure 2 illustrating detection results on RGB only, Thermal only and fusion algorithm. The first two rows illustrates night scene detections where thermal results are better. In the second row, pedestrians are not even visible in the color image, thus making it impossible for the network to detect. Second two rows correspond to day scenes where color network performs better. This corresponds to the quantitative results observed. Qualitative results for the test sequences are shared in `https://streamable.com/rlkdor`.

We evaluated the thermal and fusion architectures on FLIR dataset. Thermal baseline produced a miss rate of 52% which is significantly higher than KAIST miss rate of 38%. This could be attributed to five classes instead of one in KAIST. Additionally, there is a larger proportion of day scenes in FLIR. The 14-bit input resolution also caused to fully leverage weights of pre-trained 8-bit network initialization. Because of the misalignment of resolution/aspect ratio (640x512 for thermal and 1600x1800 for color), spatial displacement of sensors, different field of views and bit depth the proposed fusion algorithm did not perform well and could not improve upon thermal baseline. This suggests that a more sophisticated fusion scheme is necessary to align the encoded features better. This is quite challenging without leveraging the prior information about field of views and sensor extrinsic calibrations and currently an unsolved problem.

5 Conclusion

In this paper, we explored the problem of fusion based object detection using color and thermal images. We developed unimodal baseline architectures for color and thermal individually using Faster-RCNN. Then we constructed a fusion network which outperforms the individual baselines. We also obtain state of the art results reported on the KAIST dataset. We demonstrated that a simple end to end CNN architecture is able to effectively fuse when the input data resolution is aligned. However, this model does not perform well in FLIR dataset where the color and thermal images have vastly different resolutions and field of views. We hope that this study encourages further research in construction of more generic fusion networks.

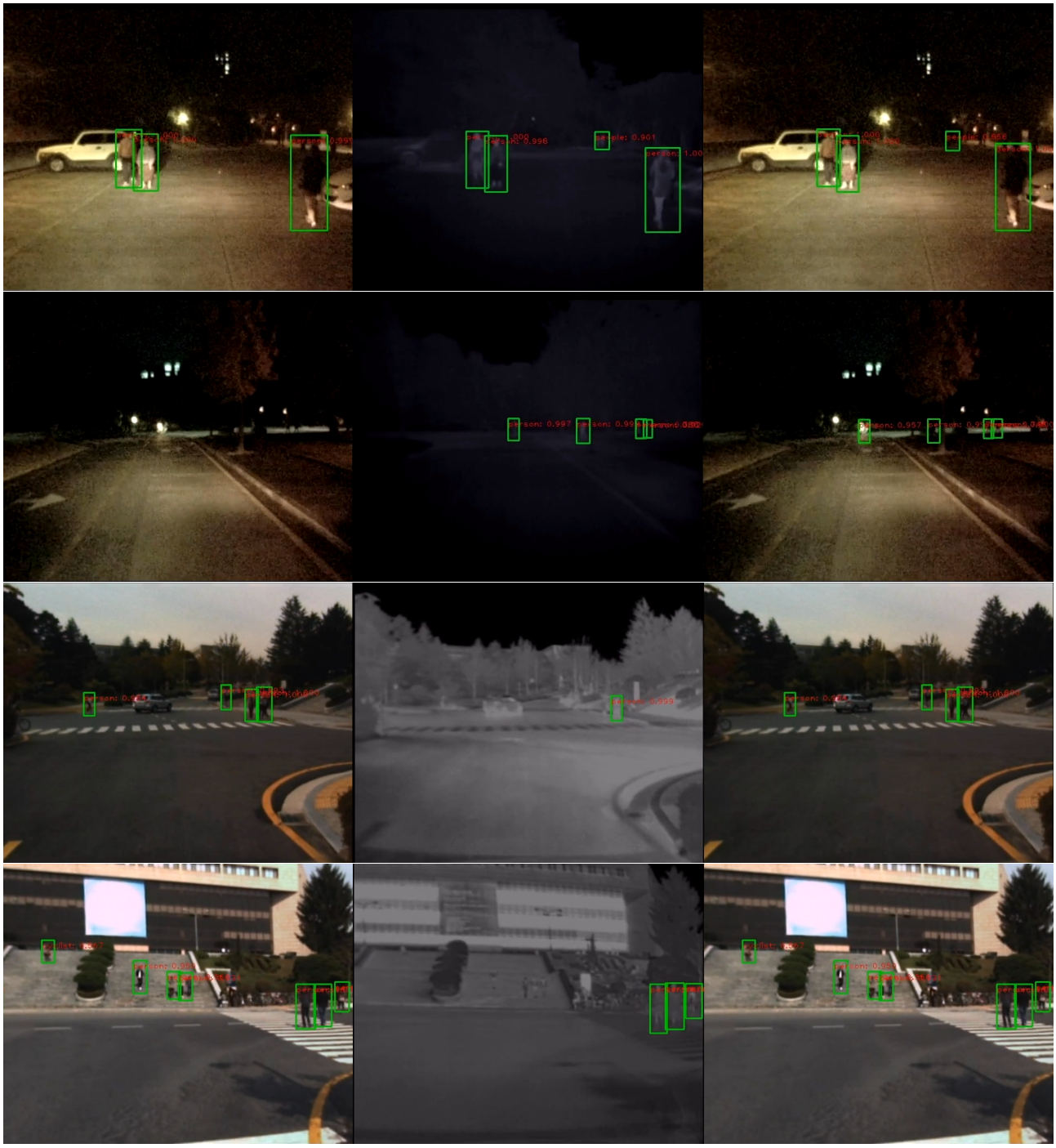


Figure 2: Qualitative comparison of object detection outputs from RGB, Thermal image baselines and proposed fusion algorithm from left to right.

References

[Airouche et al., 2012] Airouche, M., Bentabet, L., Zelmat, M., and Gao, G. (2012). Pedestrian tracking using color, thermal and location cue measurements: a dsmt-based framework. *Machine Vision and Applications*.

[Bertozzi et al., 2003] Bertozzi, M., Broggi, A., Grisleri, P., Graf, T., and Meinecke, M. (2003). Pedestrian detection in infrared images. In *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings*, pages 662–667.

[Binelli et al., 2005] Binelli, E., Broggi, A., Fascioli, A., Ghidoni, S., Grisleri, P., Graf, T., and Meinecke, M. (2005). A modular tracking system for far infrared pedestrian recognition. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 759–764. IEEE.

[Briot et al., 2018] Briot, A., Viswanath, P., and Yogamani, S. (2018). Analysis of efficient cnn design techniques for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 663–672.

[Chen et al., 2008] Chen, M.-Q., Wang, J., Zhang, M.-X., Chen, M.-G., Zhu, X.-F., Min, F.-F., and Tan, Z.-C. (2008). Catalytic effects of eight inorganic additives on pyrolysis of pine wood sawdust by microwave heating. *Journal of Analytical and Applied Pyrolysis*, 82(1):145–150.

[Chennupati et al., 2019] Chennupati, S., Sistu, G., Yogamani, S., and A Rawashdeh, S. (2019). Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0.

[Choi et al., 2016] Choi, H., Kim, S., Park, K., and Sohn, K. (2016). Multi-spectral pedestrian detection based on accumulated object proposal with fully convolutional networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 621–626. IEEE.

[Correa et al., 2012] Correa, M., Hermosilla, G., Verschae, R., and Ruiz-del Solar, J. (2012). Human detection and identification by robots using thermal and visual information in domestic environments. *Journal of Intelligent & Robotic Systems*, 66(1-2):223–243.

[El Madawi et al., 2019] El Madawi, K., Rashed, H., El Sallab, A., Nasr, O., Kamel, H., and Yogamani, S. (2019). Rgb and lidar fusion based 3d semantic segmentation for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 7–12. IEEE.

[FLIR, 2018] FLIR (2018). FLIR Thermal Dataset for ADAS. `https://www.flir.com/oem/adas/adas-dataset-form/`.

[Gade and Moeslund, 2014] Gade, R. and Moeslund, T. B. (2014). Thermal cameras and applications: a survey. *Machine vision and applications*, 25(1):245–262.

[Guan et al., 2019] Guan, D., Cao, Y., Yang, J., Cao, Y., and Yang, M. Y. (2019). Fusion of multispectral data through illumination-aware deep neural networks for pedestrian detection. *Information Fusion*, 50:148–157.

[Hwang et al., 2015] Hwang, S., Park, J., Kim, N., Choi, Y., and So Kweon, I. (2015). Multispectral pedestrian detection: Benchmark dataset and baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1037–1045.

[Konig et al., 2017] Konig, D., Adam, M., Jarvers, C., Layher, G., Neumann, H., and Teutsch, M. (2017). Fully convolutional region proposal networks for multispectral person detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 49–56.

[Kumar et al., 2018] Kumar, V. R., Milz, S., Witt, C., Simon, M., Amende, K., Petzold, J., Yogamani, S., and Pech, T. (2018). Monocular fisheye camera depth estimation using sparse lidar supervision. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2853–2858. IEEE.

[Li et al., 2019a] Li, C., Song, D., Tong, R., and Tang, M. (2019a). Illumination-aware faster r-cnn for robust multispectral pedestrian detection. *Pattern Recognition*, 85:161–171.

[Li and Wu, 2018] Li, H. and Wu, X.-J. (2018). Densefuse: A fusion approach to infrared and visible images. *IEEE Transactions on Image Processing*, 28(5):2614–2623.

[Li et al., 2019b] Li, S., Araujo, I. B., Ren, W., Wang, Z., Tokuda, E. K., Junior, R. H., Cesar-Junior, R., Zhang, J., Guo, X., and Cao, X. (2019b). Single image deraining: A comprehensive benchmark analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Liu et al., 2016] Liu, J., Zhang, S., Wang, S., and Metaxas, D. N. (2016). Multispectral deep neural networks for pedestrian detection. *arXiv preprint arXiv:1611.02644*.

[Miethig et al., 2019] Miethig, B., Liu, A., Habibi, S., and Mohrenschildt, M. v. (2019). Leveraging thermal imaging for autonomous driving. In *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*.

[Milz et al., 2018] Milz, S., Arbeiter, G., Witt, C., Abdallah, B., and Yogamani, S. (2018). Visual slam for automated driving: Exploring the applications of deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 247–257.

[Rashed et al., 2019a] Rashed, H., El Sallab, A., Yogamani, S., and ElHelw, M. (2019a). Motion and depth augmented semantic segmentation for autonomous navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 364–370.

[Rashed et al., 2019b] Rashed, H., Ramzy, M., Vaquero, V., El Sallab, A., Sistu, G., and Yogamani, S. (2019b). Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*.

[Rashed et al., 2019c] Rashed, H., Yogamani, S., El-Sallab, A., Krízek, P., and El-Helw, M. (2019c). Optical flow augmented semantic segmentation networks for automated driving. *arXiv preprint arXiv:1901.07355*.

[Ravi Kiran et al., 2018] Ravi Kiran, B., Roldao, L., Irastorza, B., Verastegui, R., Suss, S., Yogamani, S., Talpaert, V., Lepoutre, A., and Trehard, G. (2018). Real-time dynamic object detection for autonomous driving using prior 3d-maps. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Siam et al., 2018] Siam, M., Mahgoub, H., Zahran, M., Yogamani, S., Jagersand, M., and El-Sallab, A. (2018). Modnet: Motion and appearance based moving object detection network for autonomous driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2859–2864.

[Sistu et al., 2019] Sistu, G., Leang, I., Chennupati, S., Yogamani, S., Hughes, C., Milz, S., and Rawashdeh, S. (2019). Neurall: Towards a unified visual perception model for automated driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 796–803. IEEE.

[Uřičář et al., 2019] Uřičář, M., Křížek, P., Sistu, G., and Yogamani, S. (2019). Soilingnet: Soiling detection on automotive surround-view cameras. In *2019 22nd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE.

[Wagner et al., 2016] Wagner, J., Fischer, V., Herman, M., and Behnke, S. (2016). Multispectral pedestrian detection using deep fusion convolutional neural networks. In *ESANN*.

[Wang et al., 2018] Wang, Z., Lin, L., and Li, Y. (2018). Multi-feature fusion based region of interest generation method for far-infrared pedestrian detection system. In *2018 IEEE Intelligent Vehicles Symposium*.

[Xu et al., 2017] Xu, D., Ouyang, W., Ricci, E., Wang, X., and Sebe, N. (2017). Learning cross-modal deep representations for robust pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5363–5371.

[Yahiaoui et al., 2019] Yahiaoui, M., Rashed, H., Mariotti, L., Sistu, G., Clancy, I., Yahiaoui, L., Kumar, V. R., and Yogamani, S. (2019). Fisheyemodnet: Moving object detection on surround-view cameras for autonomous driving. *arXiv preprint arXiv:1908.11789*.

[Yang et al., 2017] Yang, J., Lu, J., Batra, D., and Parikh, D. (2017). A faster pytorch implementation of faster r-cnn. *https://github.com/jwyang/faster-rcnn.pytorch*.

Object Polygonization in Traffic Scenes using Small Eigenvalue Analysis

Naresh Y G, Venkatesh G M, Noel E. O'Connor and Suzanne Little

Insight SFI Research Centre for Data Analytics
Dublin City University
Dublin, Ireland
Email: naresh.yg@gmail.com; venkatesh.gurrammmunirathnam2@mail.dcu.ie;
noel.oconnor@dcu.ie; suzanne.little@dcu.ie

Abstract

Shape polygonization is an effective and convenient method to compress the storage requirements of a shape curve. Polygonal approximation offers an invariant representation of local properties even after digitization of a shape curve. In this paper, we propose to use universal threshold for polygonal approximation of any two-dimensional object boundary by exploiting the strength of small eigenvalues. We also propose to adapt the Jaccard Index as a metric to measure the effectiveness of shape polygonization. In the context of this paper, we have conducted extensive experiments on the semantically segmented images from Cityscapes dataset to polygonize the objects in the traffic scenes. Further, to corroborate the efficacy of the proposed method, experiments on the MPEG-7 shape database are conducted. Results obtained by the proposed technique are encouraging and can enable greater compression of annotation documents. This is particularly critical in the domain of instrumented vehicles where large volumes of high quality video must be exhaustively annotated without loss of accuracy and least man-hours.

Keywords: Dominant Point, Shape Representation, Shape polygonization, Small Eigenvalue

1 Introduction

Polygonal approximation is a convenient way to compress the digital representation of a closed shape curve. The polygonal approximation is achieved through detecting dominant points on the two-dimensional shape curve. Dominant points are some times termed significant points on the shape curve that are capable of representing the curvatures in the shape curve or contours. The local properties of a closed shape curve are preserved even after compression due to polygonal approximation. An ideal polygonal approximation should not be affected by linear transformations such as rotation, scaling and translation.

Polygon approximation methods can be applied to shape analysis, pattern classification, image understanding, 3D reconstruction, cartography and computer simulation applications. One example for computer simulation is emulation of traffic utilising the polygonal approximation of objects in the scene. Furthermore, for training machine learning systems in autonomous driving scenarios, labelled polygonized objects in traffic scenes play a major role in detection and recognition activities. An instrumented vehicle can collect up to 50TB of data per 8 hours of operation using 4-8 cameras. This video data then needs to be semantically annotated to be used. The number of objects per video frame will vary considerably but using a sample from a standard dataset [Cordts et al., 2016]. We observe an average of 7 and 11.8 instances per image of humans and vehicles that may require huge man-hours of annotation with fully described polygon boundaries. High quality fully annotated video with 30 frames per second will therefore generate significant quantities of annotation data.

 ISBN 978-0-9934207-5-7

Effective polygonal approximation that doesn't reduce the accuracy of modelling will greatly improve the efficiency of these processes. With this motivation, we address the polygonal approximation of objects in traffic scenes.

The rest of this paper is as follows: Section 2 gives an overview of prior work on polygonal approximate of shape curves. Section 3 describes the polygonal approximation technique and the need for post-processing to reduce duplication of vertices. Section 4 presents detailed experiments based on the proposed technique. Section 5 gives a conclusion on the proposed work.

2 Related Work

Polygonal approximation methods can be classified as sequential, merge based (heuristic) and split and merge methods [Morgera, 2012]. Alternatively, these algorithms can be classified [Madrid-Cuevas et al., 2016] on two criteria polygon approximation [Perez and Vidal, 1994] having number of dominant points to be fixed or specifying maximum allowable distortion after polygonal approximation. A brief review of current methods according to this classification is presented in this section.

One of the first heuristic polygonal approximation techniques [Ramer, 1972, Douglas and Peucker, 1973] proposed using an iterative method to divide a closed shape curve into a polygon with minimal number of vertices or dominant points. [Masood and Haq, 2007] proposed an heuristic method by elimination of break points (vertices of polygonised boundary) iteratively for polygonization of digital shape curve. Similarly, by eliminating one break point based on associated error value is proposed in [Masood, 2008]. [Carmona-Poyato et al., 2011] used an iterative method that optimises an objective function known as reference approximation for polygonal approximation of digital shape curve using the method proposed in [Perez and Vidal, 1994]. [Marji and Siy, 2003] proposed a method based on region-of-support of every boundary point capable of preserving the symmetry of the shape after polygonal approximation. Similarly, [Wu, 2003] proposed a method of detecting dominant points using region of support at every point of the boundary instead of considering a fixed length of support region and [Bhowmick and Bhattacharya, 2007] proposed geometric constraints for straightness properties of digital curve to detect the dominant points. Polygon approximation by specifying the maximum allowable distortion is derived based on sum of squares deviation criterion [Salotti, 2002] to detect the dominant point in a shape curve.

There are some methods that do not use any user defined parameters as criterion for finding optimal dominant points. The methods [Ramer, 1972, Douglas and Peucker, 1973, Carmona-Poyato et al., 2011] are modified using a non-parametric approach [Prasad et al., 2012] that utilises the theoretical bound of the deviation of the pixels obtained by the digitisation of a line segment. Another method [Madrid-Cuevas et al., 2016], which is also a non-parametric approach, that exploits the split/merge technique and a quality measure known as Figure-of-Merit. In spite of many algorithms for polygonal approximation, still there is scope for improvement in addressing near real-time polygonization of shape curves or boundaries of objects. The methods in [Bhowmick and Bhattacharya, 2007, Dinesh et al., 2005] exploit the straightness of the line segment to detect dominant points in the shape curve. In [Dinesh et al., 2005] employed a split based method that uses small eigenvalue for polygonization of a shape curve. It recursively splits a closed boundary curve considering a threshold for small eigenvalue. The value of the threshold and its computation is not specified. In split based techniques, duplication of vertices arises due to recursive splitting of boundary based on a decision criteria. This disadvantage shall be overcome using a merge based technique. Furthermore, suppression of collinear points after polygonal approximation assist in avoiding duplicated dominant points in a straight line segment.

3 Proposed Method

The proposed method utilizes a merging technique to detect dominant points of an object boundary in two dimensional space. In this technique, a decision is made at every boundary point to choose to merge the point in the set of dominant points to form a vertex of the approximating polygon. Theoretically, the eigenvalue

associated with a straight line is equal to zero [Guru et al., 2004, Dinesh et al., 2005]. However, the eigenvalue associated with a digital line will be slightly more than zero. The proposed method uses small eigenvalue [Tsai et al., 1999, Guru et al., 2004] of the co-variance matrix of boundary points of an object as a decision criterion.

3.1 Small Eigenvalue Computation

The small eigenvalue is computed by adopting the methodology proposed in [Tsai et al., 1999]. The choice of parameters is carried out similarly as specified in [Guru et al., 2004]. Let $B = \{b_k(x_i, y_i) | i = 1, 2, ..., N_k\}$ be N number of boundary pixels belonging to k_{th} boundary of an object, the small eigenvalue λ of the covariance matrix of B is computed as following:

$$\lambda = \frac{1}{2}\left[C_{11} + C_{22} - \sqrt{(C_{11} + C_{22})^2 - 4C_{12}^2}\right] \tag{1}$$

where

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

is a covariance matrix of B and the coefficients in the matrix are computed as follows:

$$C_{11} = \frac{1}{N}\sum_{i=1}^{N}(x_i^2 - c_x^2) \tag{2}$$

$$C_{12} = C_{21} = \frac{1}{N}\sum_{i=1}^{N}(x_i.y_i - c_x.c_y) \tag{3}$$

$$C_{22} = \frac{1}{N}\sum_{i=1}^{N}(y_i^2 - c_y^2) \tag{4}$$

c_x and c_y are the mean values of the x and y co-ordinates respectively.

3.2 Polygon Approximation Algorithm

The input for the proposed technique is a set of boundary/contour pixels b_k of an object in the traffic scene and the output is a set of dominant points P resulting in polygonal approximation of the object. The algorithm for the proposed polygonal approximation of a given boundary is as follows:

Algorithm 1 Polygonal Approximation

Require: $B = \{b_k(x_i, y_i) | i = 1, 2, ..., N_k\}$
Ensure: $P = P_{k1} \dots P_{kM}$
1: $StartPoint \leftarrow 0$
2: $Index \leftarrow 0$
3: **for** $i \leftarrow 1$ to N_k **do**
4: $Points \leftarrow b_k[StartPoint : i+1, :]$
5: Compute λ for $Points$ using Eqn.(1)
6: **if** $\lambda > Threshold$
7: $StartPoint \leftarrow i$
8: $P[Index] \leftarrow b_k(x_i, y_i)$
9: $Index++$
10: **end if**
11: **end for**

 ISBN 978-0-9934207-5-7

3.3 Post-processing of vertices

The vertices obtained after applying the polygonal approximation on the boundary pixels will have closer vicinity at curvature extreme portions of the boundary pixels as shown in Figure. 1(a). The points shown in the highlighted portion of Figure. 1(c) are nearly collinear. It is necessary to reduce such points by retaining certain points that are not collinear. This can be achieved by re-applying the same polygonal approximation method mentioned in 3.1 with the smallest threshold as a parameter for small eigenvalue as mentioned in [Guru et al., 2004]. This results in effective suppression of collinear points in the curvature extreme region as shown in Figure. 1(d).

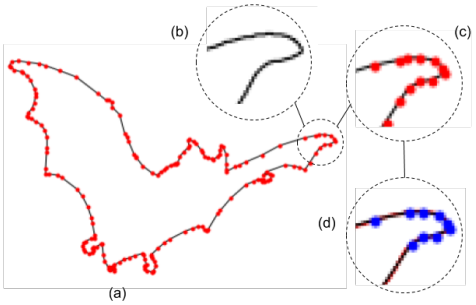


Figure 1: Illustration of collinear points suppression after polygonal approximation

4 Experiments

To validate the efficacy of the proposed polygon approximation technique, we have conducted experiments on the standard MPEG-7 shape database and on semantic segmentation images from the validation set of the CityScapes dataset [Cordts et al., 2016].

According to our literature study, only certain contours like hammer, plane, chromosome, leaf, semicircles and infinity are used. In our experiments, we have also considered more complex shapes and the result on a subset of data is shown in Figure. 2. The polygon approximation technique is evaluated in terms of Compression Ratio (CR) and Integral Square Error (ISE). Compression Ratio (Eqn. 5) is defined as the ratio between the number of points in the contour (N) and the number of points(P) obtained after polygonal approximation.

$$CR = \frac{N}{P} \tag{5}$$

The Integral Square Error ISE is the sum of square error and is defined as follows:

$$ISE = \sum_{i=1}^{N} e_i^2 \tag{6}$$

where e_i is the distance between the original contour point and the approximated line segment.

In assessment of semantic segmentation, the Jaccard index, also known as Intersection over Union (IoU), is popularly used. This is a statistical measure used to define the proximity of two sample sets that helps in understanding the approximation or prediction quantitatively. According to our literature review, Jaccard index is not found as a metric to evaluate the proximity between the original shape curve and its respective polygonal approximation. However, in semantic segmentation [Badrinarayanan et al., 2015], the Jaccard index effectively explains the proximity of actual regions and their respective predicted regions by the segmentation techniques. Therefore, in our experiments, we use Jaccard index to find the ratio of the symmetric difference between the original shape curve and its respective polygonal approximation to their union. Hence, Jaccard index explains the empirical proximity between a shape curve and its polygonal approximation. A shape curve, S, is represented by a set of boundary points and its respective polygonal approximation, P, yields a set of dominant points that effectively preserves the curvature information. The Jaccard index, $J(SR, PR)$, is given by:

$$J(SR, PR) = \frac{SR \cap PR}{SR \cup PR} \tag{7}$$

 ISBN 978-0-9934207-5-7

where SR, is the region of original shape contour (S) and PR is the region of polygonal shape approximation (P) using the proposed method.

4.1 Experiments on MPEG-7 Shape Database

Several methods [Prasad et al., 2012, Carmona-Poyato et al., 2011] used the Compression Ratio (CR) and Integral Square Error (ISE) as the quality metrics to present the efficacy of their methods on a small set of contours rather than the whole dataset. In our experimentation, we have considered all the objects mentioned in Table. 2 in the dataset for evaluation of the performance of the proposed technique. Figure 2 shows the dominant point extraction on some of the shapes – spring, bat, elephant and guitar. The above mentioned shapes have curvature extremes in their shape. Hence, these shapes are considered for visualisation. It can be observed from figure 2 that the polygonal approximation of all the shapes is effective in preserving the curvature extreme of original shape curve. Table. 1 shows the number of contour points (CountourPts), approximated dominant points (M), compression ratio (CR) and integral square error (ISE) of the polygonal approximation and its post-processing for the shapes shown in figure 2. Table. 2 shows overall mean of Jaccard Index (JD), Compression Ratio (CR) and Integrated square error (ISE) of polygonal approximation and its post-processing for every class of shape in the MPEG-7 dataset. It can be observed that the performance of the proposed technique is efficient in terms of CR and JD on most of the object classes in the MPEG-7 dataset. However, for the object class $Glass$, performance of the proposed technique decreases in terms of the Jaccard Index after post-processing. Figure. 4(b) shows the drawback of the proposed post-processing step where one of the dominant points is removed resulting in deformation of the shape curve after polygonal approximation.

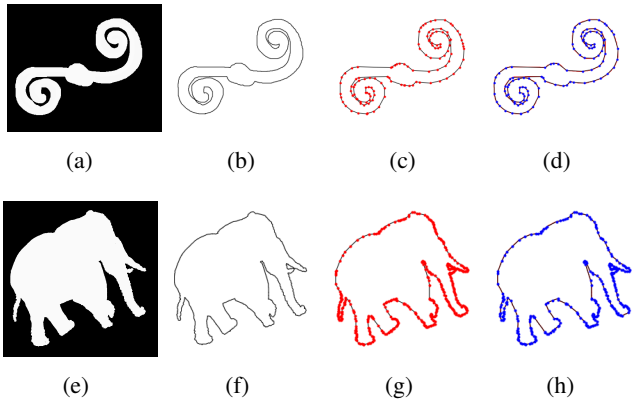


Figure 2: Dominant point extraction on some of the shapes in MPEG-7 dataset possessing curvature extreme. 1^{st} column: Input images, 2^{nd} column: contour extraction output, 3^{rd} column: dominant points and 4^{th} column: dominant points after post processing.

Input	ContourPts	M	JD	CR	ISE	M^P	JD^P	CR^P	ISE^P
Spring-1	1948	122	0.956	15.967	1.926	61	0.864	31.934	5.493
Elephant-1	3317	386	0.99	8.593	1.81	182	0.982	18.225	6.093

Table 1: Metric on shapes considered in figure 2 from MPEG-7 dataset. $*^P$ represents post-processed values

4.2 Polygonization of Semantically Segmented Objects

In order to show the applicability of the proposed technique in real-time visual analysis and automatic annotation tasks, we have conducted experiments on semantic segmentation results on the CityScapes Dataset. We present the results of our polygonal approximation on object shapes yielded by semantic segmentation [Badrinarayanan et al., 2015]. Table. 3 shows the Jaccard Index and compression ratio of polygonal approximation and after its post-processing. Figure. 3 illustrates polygonization of objects in a traffic scene. The performance

Input	JD	CR	ISE	JD^P	CR^P	ISE^P
Apple	0.98	16.44	5.51	0.98	17.75	5.64
Bat	0.98	10.31	2.17	0.98	15.23	3.08
Beetle	0.95	7.83	2.34	0.94	11.47	3.04
Bell	0.98	12.21	1.48	0.97	15.88	1.86
Bird	0.98	12.72	3.49	0.98	15.92	4.79
Bone	0.95	56.42	226.98	0.92	60.18	252.98
Bottle	0.96	20.8	0.65	0.95	26.24	0.82
Brick	0.97	16.28	1.88	0.97	19.72	2.57
Butterfly	0.98	8.99	2.05	0.98	13.22	2.64
Camel	0.97	11.48	1.36	0.97	13.95	1.77
Car	0.97	12.28	1.21	0.97	13.91	1.21
Carriage	0.95	11.18	1.27	0.94	12.31	1.33
Cellular Phone	0.98	31.55	0.88	0.97	39	0.86
Chicken	0.97	8.76	2.08	0.97	11.36	2.57
Children	0.96	10.31	0.57	0.96	12.96	0.65
Chopper	0.96	12.85	1.78	0.96	14.56	2.03
Classic	0.99	15.56	1.72	0.98	18.8	1.92
Comma	0.98	49.99	15.25	0.98	55.16	17.71
Crown	0.94	8.2	2.35	0.94	9.7	2.69
Cup	0.98	20.53	1.84	0.98	23.72	2.31
Deer	0.93	5.57	2.28	0.92	8.33	3.53
Device	0.98	23.17	29.95	0.96	40.04	28.72
Dog	0.97	10.25	1.47	0.97	13.34	1.85
Elephant	0.96	9.76	2.34	0.96	13.84	3.74
Face	0.99	15.19	3.44	0.99	18.31	4.04
Fish	0.96	11.42	2.01	0.96	14.24	2.41
Flatfish	0.99	15.27	5.65	0.99	20.81	10.67
Fly	0.94	9.83	3.44	0.93	12.85	4.78
Fork	0.94	26.61	2.89	0.93	32.43	4.14
Fountain	0.97	14.98	1.73	0.97	17	1.98

Input	JD	CR	ISE	JD^P	CR^P	ISE^P
Frog	0.98	12.57	3.19	0.98	14.93	3.98
Glass	0.96	83.62	30.33	0.85	88.95	31.8
Guitar	0.97	15.9	5.73	0.97	18.95	7.81
Hammer	0.91	30.51	4.51	0.87	34.46	4.95
Hat	0.97	18.65	1.37	0.97	21.68	1.48
Hcircle	0.98	38.48	11.29	0.98	40.86	12.06
Heart	0.99	29.11	39.3	0.99	30.55	36.39
Horse	0.98	11.19	1.91	0.98	16.17	2.7
Horseshoe	0.93	13.87	1.45	0.92	16.71	2.21
Jar	0.97	13.04	3.18	0.97	14.95	3.81
Key	0.98	19.55	2.21	0.98	22.68	2.46
Lizzard	0.97	9.4	2.35	0.97	13.76	3.43
Lmfish	0.97	10.3	2.15	0.97	14.27	3.22
Misk	0.99	20.1	6.41	0.99	22.2	8.85
Octopus	0.96	13.25	2	0.95	14.91	2.19
Pencil	0.95	39.63	12.77	0.95	48.3	16.54
Personal Car	0.99	18.14	3.87	0.98	20.72	4.49
Rat	0.95	12.74	1.66	0.95	14.37	1.83
Ray	0.99	11.36	5.39	0.98	16.29	8.73
Sea Snake	0.95	13.53	4.3	0.95	18.43	7.4
Shoe	0.99	19.79	1.93	0.99	23.02	2.37
Spoon	0.96	17.35	9.96	0.95	20.98	23.81
Spring	0.92	13.13	1.52	0.91	15.75	1.99
Stef	0.9	10.44	1.73	0.9	12.15	2.19
Teddy	0.98	11.86	1.88	0.98	13.55	2.26
Tree	0.97	14.38	3.38	0.96	17.89	4.84
Truck	0.95	9.27	1.31	0.95	11.25	1.33
Turtle	0.98	8.51	1.66	0.98	12.42	2.3
Watch	0.97	20.09	1.04	0.97	27.41	1.44
–	–	–	–	–	–	–

Table 2: Metric extracted on MPEG-7 dataset

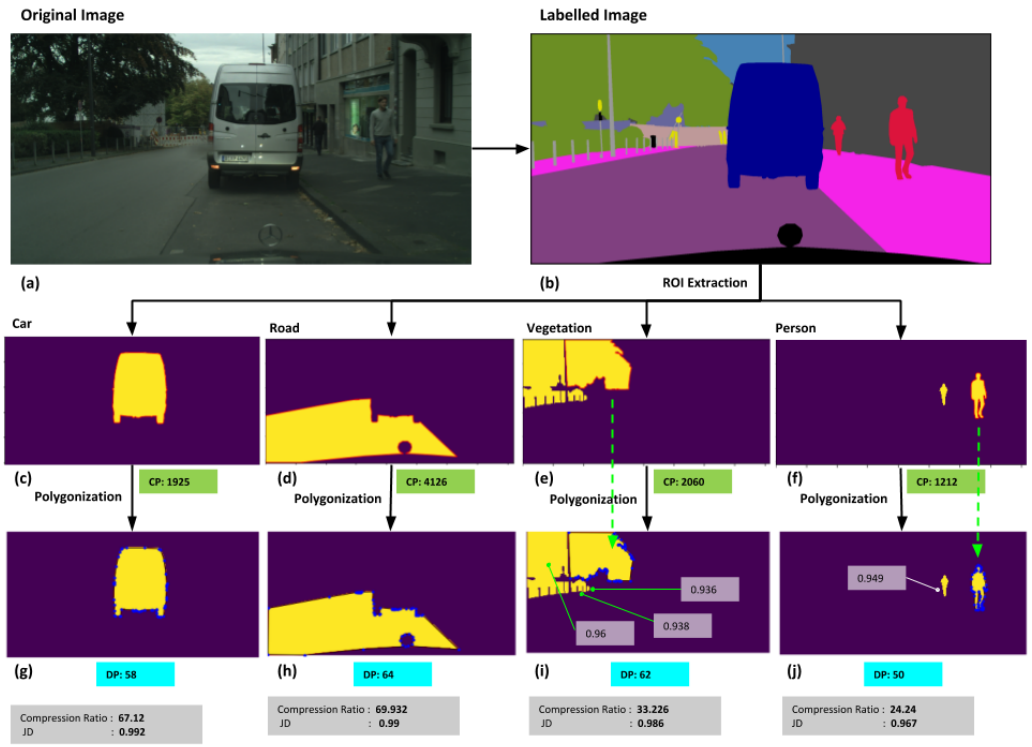
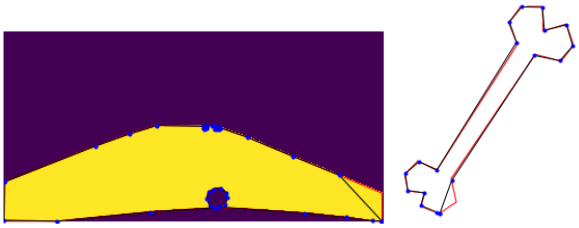


Figure 3: Polygonal Approximation of objects in a traffic scene

of the proposed algorithm for most of the object classes in the CityScapes dataset is considerably more effective in compressing and retaining the shape of the digital curve. From Table. 3, it can be observed that the proposed technique is effective on most of the traffic object classes however on the object class *Pole* and *Sidewalk* are slightly low. This is due to post-processing step which results in extra compression of near rectangular shapes and shapes of small objects which possess minimal boundary points in the boundary curvature. The application of a post-processing step on certain shapes, which are near regular shape, leads to filtering out (removing) dominant points that are important to preserving the curvature extreme of the shape curve. Figure. 4 shows

two examples that highlight the drawback of this post-processing method. It can be clearly observed that one of the dominant points is filtered out affecting the shape of the curve which results in clamping of the shape structure. Figure. 4(a) shows the clamping effect on the object class $Road$. One of the dominant points which preserves the curvature extreme has been filtered out and resulted in deformation of the shape after polygonal approximation.



(a) (b)

Figure 4: Illustration of a drawback of the proposed post-processing method

5 Conclusion

A simple yet effective technique that exploits the merge based method using small eigenvalue of a straight line for polygonal approximation has been proposed. This technique helps in retaining the curvature extreme of the shape contour with minimum points. We have shown that Jaccard Index is an effective metric to assess polygonal approximation. The proposed technique has been applied on semantic segmentation results on CityScapes dataset for representation of object shape in the scene which enables effective shape recovery of the objects during visual analysis and annotation tasks. Hence, the proposed technique plays a key role in automatic annotation tasks involved in ADAS (Advanced driver-assistance systems) and autonomy applications.

Acknowledgments

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement number 688099) project $Cloud-LSVA$ and from Science Foundation Ireland under grant number $SFI/16/SP/3804$. The Insight SFI Research Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number $SFI/12/RC/2289_P2$.

Object Class	JD	CR	JD^P	CR^P
Bicycle	0.923	13.215	0.91	14.775
Building	0.921	23.853	0.894	25.704
Bus	0.943	23.495	0.933	25.518
Car	0.949	19.083	0.942	21.028
Fence	0.917	32.323	0.884	34.157
Motorcycle	0.922	12.662	0.914	14.016
Person	0.924	13.008	0.913	14.863
Pole	0.779	51.683	0.771	52.317
Rider	0.92	12.113	0.909	13.947
Road	0.947	46.548	0.933	49.814

Object Class	JD	CR	JD^P	CR^P
Sidewalk	0.915	32.875	0.872	35.567
Sky	0.931	22.503	0.907	24.62
Terrain	0.909	30.098	0.875	32.461
Traffic Light	0.918	16.512	0.885	17.686
Traffic Sign	0.92	26.681	0.91	27.385
Train	0.93	26.54	0.924	28.243
Truck	0.936	22.195	0.932	24.339
Vegetation	0.925	21.259	0.902	23.111
Wall	0.916	35.176	0.886	36.722
–	–	–	–	–

Table 3: Metric extracted on Object Classes in CityScapes Dataset

References

[Badrinarayanan et al., 2015] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*.

[Bhowmick and Bhattacharya, 2007] Bhowmick, P. and Bhattacharya, B. B. (2007). Fast polygonal approximation of digital curves using relaxed straightness properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9).

[Carmona-Poyato et al., 2011] Carmona-Poyato, A., Medina-Carnicer, R., Madrid-Cuevas, F. J., Muñoz-Salinas, R., and Fernández-García, N. (2011). A new measurement for assessing polygonal approximation of curves. *Pattern Recognition*, 44(1):45–54.

[Cordts et al., 2016] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223.

[Dinesh et al., 2005] Dinesh, R., Damle, S. S., and Guru, D. (2005). A split-based method for polygonal approximation of shape curves. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 382–387. Springer.

[Douglas and Peucker, 1973] Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122.

[Guru et al., 2004] Guru, D., Shekar, B., and Nagabhushan, P. (2004). A simple and robust line detection algorithm based on small eigenvalue analysis. *Pattern Recognition Letters*, 25(1):1–13.

[Madrid-Cuevas et al., 2016] Madrid-Cuevas, F. J., Aguilera-Aguilera, E. J., Carmona-Poyato, A., Muñoz-Salinas, R., Medina-Carnicer, R., and Fernández-García, N. (2016). An efficient unsupervised method for obtaining polygonal approximations of closed digital planar curves. *Journal of Visual Communication and Image Representation*, 39:152–163.

[Marji and Siy, 2003] Marji, M. and Siy, P. (2003). A new algorithm for dominant points detection and polygonization of digital curves. *Pattern recognition*, 36(10):2239–2251.

[Masood, 2008] Masood, A. (2008). Dominant point detection by reverse polygonization of digital curves. *Image and Vision Computing*, 26(5):702–715.

[Masood and Haq, 2007] Masood, A. and Haq, S. A. (2007). A novel approach to polygonal approximation of digital curves. *Journal of Visual Communication and Image Representation*, 18(3):264–274.

[Morgera, 2012] Morgera, A. (2012). *Dominant points detection for shape analysis*. PhD thesis, Universita'degli Studi di Cagliari.

[Perez and Vidal, 1994] Perez, J.-C. and Vidal, E. (1994). Optimum polygonal approximation of digitized curves. *Pattern recognition letters*, 15(8):743–750.

[Prasad et al., 2012] Prasad, D. K., Leung, M. K., Quek, C., and Cho, S.-Y. (2012). A novel framework for making dominant point detection methods non-parametric. *Image and Vision Computing*, 30(11):843–859.

[Ramer, 1972] Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3):244–256.

[Salotti, 2002] Salotti, M. (2002). Optimal polygonal approximation of digitized curves using the sum of square deviations criterion. *Pattern Recognition*, 35(2):435–443.

[Tsai et al., 1999] Tsai, D.-M., Hou, H.-T., and Su, H.-J. (1999). Boundary-based corner detection using eigenvalues of covariance matrices. *Pattern Recognition Letters*, 20(1):31–40.

[Wu, 2003] Wu, W.-Y. (2003). An adaptive method for detecting dominant points. *Pattern Recognition*, 36(10):2231–2237.

Patch based Colour Transfer using SIFT Flow

Hana Alghamdi & Rozenn Dahyot

School of Computer Science & Statistics
Trinity College Dublin, Ireland
alghamdh@tcd.ie, rozenn.dahyot@tcd.ie

Abstract

We propose a new colour transfer method with Optimal Transport (OT) to transfer the colour of a source image to match the colour of a target image of the same scene that may exhibit large motion changes between images. By definition OT does not take into account any available information about correspondences when computing the optimal solution. To tackle this problem we propose to encode overlapping neighborhoods of pixels using both their colour and spatial correspondences estimated using motion estimation. We solve the high dimensional problem in 1D space using an iterative projection approach. We further introduce smoothing as part of the iterative algorithms for solving optimal transport namely Iterative Distribution Transport (IDT) and its variant the Sliced Wasserstein Distance (SWD). Experiments show quantitative and qualitative improvements over previous state of the art colour transfer methods.

Keywords: Optimal Transport, Nadaraya-Watson estimator, Iterative Distribution Transfer, Sliced Wasserstein Distance, Colour Transfer

1 Introduction

Colour variations between photographs often happen due to illumination changes, using different cameras, different in-camera settings or due to tonal adjustments of the users. Colour transfer methods have been developed to transform a source colour image into a specified target colour image to match colour statistics or eliminate colour variations between different photographs. Colour transfer has many applications in image processing problems, ranging from generating colour consistent image mosaicing and stitching [1] to colour enhancement and style manipulation [2].

When computing the transfer function, considering colour information only does not take into account the fact that coherent colours should be transferred to neighboring pixels, which can create results with blocky artifacts emphasizing JPEG compression blocks, or increase noise. To tackle this problem, Alghamdi et al. [3] proposed the Patch based Colour Transfer (`PCT_OT`) approach that encodes overlapping neighborhoods of pixels, taking into account both their colour and pixel positions. The `PCT_OT` algorithm not only shows improvements over the state of the art methods but also shows limitations by creating shadow artifacts when there are large changes between target and source images. In this paper we propose to improve `PCT_OT` by first improving the data preparation step for defining patches thanks to SIFT flow [4]. We estimate motions between images using the SIFT flow approach and incorporate the spatial correspondence information in the encoded overlapping neighborhoods of pixels. This formulation makes OT *implicitly* take into account correspondences when computing the optimal solution. Our second contribution is to introduce smoothing as part of the iterative algorithms for solving optimal transport namely Iterative Distribution Transport (IDT) and its variant the Sliced Wasserstein Distance (SWD).

2 PCT_OT with SIFT Flow

2.1 Combine colour and spatial information

The spatial information for the target image is calculated using the SIFT flow method which estimates dense spatial correspondences by robustly aligning complex scene pairs containing significant spatial differences

 ISBN 978-0-9934207-5-7

[4], while in PCT_OT [3] the original pixel positions in the grid coordinate of the image are used. Using correspondences will allow colour transfer between images that contain moving objects and overcome the limitations in PCT_OT. More specifically, let y^p be the 2D pixel position of the target image to be computed, and let $\mathbf{p} = (a, b)$ be the 2D grid coordinate of the target image and $\mathbf{w}(\mathbf{p}) = (u(\mathbf{p}), v(\mathbf{p}))$ be the flow vector at $\mathbf{p}$ computed using the SIFT flow method, then $y^p = \mathbf{p} + \mathbf{w}(\mathbf{p}) = (a + u(\mathbf{p}), b + v(\mathbf{p}))$ is the new pixel position in the target image that match a pixel position in the source image. The pixel's colour y^c and its pixel position y^p are concatenated into a vector $y = (y^c, y^p)^T$ such that $\dim(y) = \dim(y^c) + \dim(y^p)$. The source image keeps the grid coordinate of the image as pixel positions, i.e $x^p = \mathbf{p}$ and similarly to the target image the pixel's colour x^c and its pixel position x^p are concatenated into a vector $x = (x^c, x^p)^T$ such that $\dim(x) = \dim(x^c) + \dim(x^p)$.

2.2 Data normalisation

Since the colours have integer values from 0 to 255, and the spatial values can be anything depending on the size of the image, we normalize all the colour and position coordinates to lie between 0 and 255 to create a hypercube in $\mathbb{R}^d$ in order to produce consistent results regardless of the size of the image and better control parameters. We then stretch that space in the direction of the spatial coordinates by a factor w to make it harder to move the pixels in the spatial domain than in the colour domain, because since we are focusing on transferring colour between images of a same scene, we know that the scenes are overlapped and hence the more overlapped areas we have the higher w value we can set.

2.3 Create patch vectors

In a similar way to PCT_OT [3] we encode overlapping neighborhoods of pixels to preserve local topology information. Starting from the origin of the coordinate system of the images (upper left corner), we use a sliding window operation of window size $k \times k$ to extract overlapping patches. From each individual patch we create a high dimensional vector in $\mathbb{R}^{d \times k \times k}$. We apply this process to the source and target images to create patch vector sets $\{x_i\}$ and $\{y_j\}$ for each respectively.

3 Smoothed solution for 1D Optimal Transport

The OT problem consists of estimating the minimum cost (referred to as the Wasserstein Distance [5] or as the Earth Mover's Distance [6]) of transfering a source distribution to a target distribution. As a byproduct of OT distance estimation, the mapping ϕ itself between the two distributions is also provided. Monge's formulation of OT [5] defines the deterministic coupling $y = \phi(x)$ between random vectors $x \sim f(x)$ and $y \sim g(y)$ that capture the colour information of the source and target images respectively, and its solution minimizes the total transportation cost:

$$\underset{\phi}{\operatorname{argmin}} \int \|x - \phi(x)\|^2 \, f(x) \, dx \quad \text{such that}: \quad f(x) = g(\phi(x)) \, |\det \nabla \phi(x)| \tag{1}$$

where f is the probability density function (pdf) of x and g is the pdf of y. The solution for ϕ can be found using existing algorithms such as linear programming, and the Hungarian and Auction algorithms [7]. However, in practice it is difficult to find a solution for colour images when $\dim(x) = \dim(y) = d > 1$ as the computational complexity of these solvers increases in multidimensional spaces [8]. But for $d = 1$, with $x, y \in \mathbb{R}$, a solution for ϕ is straightforward and can be defined using the increasing rearrangement [5]:

$$\phi^{OT} = G^{-1} \circ F \tag{2}$$

where F and G are the cumulative distributions of the colour values in the source and target images respectively.

3.1 Iterative Distribution Transfer (IDT)

The 1D solution ϕ^{OT} Eq. (2) has been used to tackle problems in multidimensional colour spaces and of particular interest is the Iterative Distribution Transfer (IDT) algorithm for colour transfer proposed by Pitié et al. [9]. They proposed to iteratively project colour values $\{x_i\}_{i=1}^n$ and $\{y_j\}_{j=1}^m$ originally in $\mathbb{R}^d$ to a 1D subspace

 ISBN 978-0-9934207-5-7

and solve the OT using ϕ^{OT} Eq. (2) in this 1D subspace and then propagate the solution back to $\mathbb{R}^d$ space. This operation is repeated with different directions in 1D space until convergence. This strategy was inspired by the idea of the Radon Transform [9] which states the following proposition: if the target and source colour points are aligned in all possible 1D projective spaces, then matching is also achieved in $\mathbb{R}^d$ space. Note that the implementation of IDT approximates F and G using cumulative histograms which can be considered as a form of quantile matching but with irregular quantile increments derived from the cumulative histograms of the source and target images - as source and target quantiles do not match exactly, interpolation can be used to compute solution [9].

3.2 Sliced-Wasserstein Distance (SWD)

The Sliced Wasserstein Distance (SWD) algorithm follows from the iterative projection approach of IDT but computes the 1D solution ϕ^{OT} with quantile matching instead of cumulative histogram matching [10, 11]. More specifically, SWD sorts the n 1D projections of the source and target images respectively to define quantiles with regular increments of size $\frac{1}{n}$ between 0 and 1 for both source and target distributions. The SWD algorithm can be computed in $\mathcal{O}(n\log(n))$ operations using a fast sorting algorithm [10]. When a small number of observations are available, using SWD is best but with a large number of observations, histogram matching with IDT is more efficient.

3.3 Smoothing ϕ^{OT} with Nadaraya Watson Estimator

Giving the correspondences $\{(x_i, y_i)\}_{i=1,\cdots,n}$, the Nadaraya Watson (NW) estimator is defined as follows:

$$\mathbb{E}[y|x] = \int y\ p(y|x)\ dy = \int y\ \frac{p(y,x)}{p(x)}\ dy \simeq \frac{n^{-1}\sum_{i=1}^{n} y_i\ K_h(x-x_i)}{n^{-1}\sum_{i=1}^{n} K_h(x-x_i)} = \phi_h^{NW}(x) \tag{3}$$

With this form NW can be seen as locally weighted average of $\{y_i\}_{i=1,\cdots,n}$, using a kernel as a weighting function where the bandwidth h is the hyperparameter or scale parameter of the kernel, the larger the value of h the more ϕ_h^{NW} gets smoothed. We propose to smooth ϕ^{OT} computed in IDT or SWD by using non-parametric Nadaraya Watson estimator. At each iteration t, following the step of calculating the optimal map ϕ^{OT}, we feed the OT estimated correspondences $\{(x_i, \phi^{OT}(x_i))\}_{i=1}^{n}$ to the NW estimator to compute a smoother OT solution, denoted as ϕ_h^{OT}, defined as follows:

$$\phi_h^{OT}(x) = \frac{\sum_{i=1}^{n} \phi^{OT}(x_i)\ K_h(x-x_i)}{\sum_{i=1}^{n} K_h(x-x_i)} \tag{4}$$

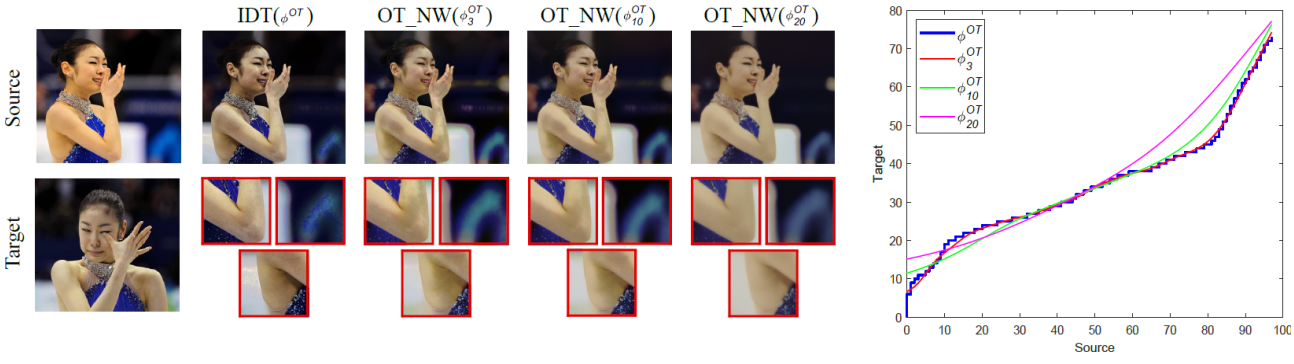


Figure 1: Results shows the smoothed Optimal Transport solution using non-parametric Nadaraya-Watson (ϕ_h^{OT}) with different bandwidth values $h = \{3, 10, 20\}$. Nadaraya-Watson significantly reduces the grainy artifacts produced by the original Optimal Transport function (ϕ^{OT}), mapping the source patch projections to the target patch projections, the bigger h value the more smoothed mapping. The results processed without post processing step. Note that the graph is a zoom in on 0-255 range pixel value.

Figure 1 illustrates the effect of computing smoother OT solutions using NW with different bandwidth values on colour transfer compared with the original OT solution computed using IDT algorithm [9]. Optimal

 ISBN 978-0-9934207-5-7

Transport solutions are suitable in situations where the function that we need to estimate must satisfy important side conditions, such as being strictly increasing, and the non-parametric NW estimator on top of the OT solution can provide the smoothness required in the estimated function. In addition, one of the important characteristics of using OT and NW estimators is that they do not assume explicit expression controlled by parameters on the regression function which makes them directly employable. In the following sections we are applying OT and NW smoothing in the relevant context of colour transfer where the function that we need to estimate must satisfy the condition of being an increasing function.

4 Experimental Assessment

We provide here quantitative and qualitative evaluations of our approach noted `OT_NW` with comparisons to different state of the art colour transfer methods noted `IDT` [9], `PMLS` [2], `GPS/LCP` and `FGPS/LCP` [12], `L2` [13] and `PCT_OT` [3]. In these evaluations we use image pairs with similar content from an existing dataset provided by Hwang et al [2]. The dataset includes registered pairs of images (source and target) taken with different cameras and settings, and different illuminations and recolouring styles.

4.1 Colour space and parameters settings

We use the RGB colour space where each pixel is represented by its 3D RGB colour values and its 2D spatial position. Our patches with combined colour and spatial features create a vector in 125 dimensions ($5 \times 5 \times 5$) for the RGB colours (3D) and position component (2D). We found that a patch size of 5×5 captures enough of a pixel's neighbourhood. We stretch the hypercube space in $\mathbb{R}^d$ in the direction of the spatial coordinates by a factor $w = 10$ to make it harder to move the pixels in the spatial domain than in the colour domain. We experimented with different bandwidth values and we found a fixed value of $h = 10$ gives best results.

4.2 Evaluation metrics

To quantitatively assess the recolouring results, four metrics are used: peak signal to noise ratio (PSNR) [14], structural similarity index (SSIM) [15], colour image difference (CID) [16] and feature similarity index (FSIMc) [17]. These metrics are often used when considering source and target images of the same content [18, 19, 2, 12]. Note that the results using `PMLS` were provided by the authors [2]. It has already been shown in [13] that `PMLS` performs better than two other more recent techniques using correspondences [20, 21], so `PMLS` is the one reported here with [3, 13] as algorithms that account for correspondences.

4.3 Experimental Results

Figures 2–5 show detailed tables of quantitative results for each metric along with box plots carrying a lot of statistical details. The purpose of the box plots is to visualize differences among methods and to show how close our method is to the state of the art algorithms. Figure 2 (b) and Figure 5 (b) show PSNR and FSIMc metrics results respectively, by examining the box plots in both figures we see that the four methods `PMLS`, `L2`, `PCT_OT` and `OT_NW` are greatly overlapped with each other, the median and mean values (the mean shown as red dots in the plots) are the highest among all algorithms and are very close in value and the whiskers length almost similar indicating similar data variation and consistency. Figure 3 (b) shows the SSIM box plot where we can see that `OT_NW` performs similarly to `PMLS` and `L2`'s highest scoring values while here the median line of `PCT_OT` box lies outside the three top algorithms scoring the lowest value among them. With CID metric in Figure 4, `OT_NW` performs similarly to `PMLS`, `L2` and `PCT_OT`. In conclusion, the quantitative metrics show that our algorithm with Nadaraya Watson `OT_NW` performs similarly with top methods `PMLS`, `L2` and `PCT_OT` and outperforms the rest of the state of the art algorithms.

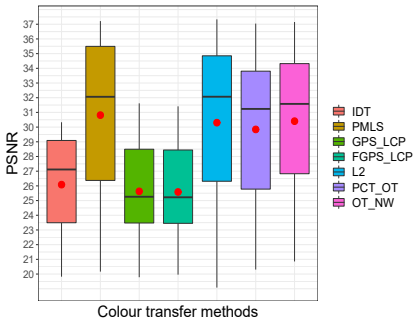
Figure 7 provides qualitative results. For clarity, the results are presented in image mosaics, created by switching between the target image and the transformed source image column wise (Figure 7, top row). If the colour transfer is accurate, the resulting mosaic should look like a single image (ignoring the small motion displacement between source and target images), otherwise column differences appear. As can be noted, our approach `OT_NW` with Nadaraya Watson step is visually the best at removing the column differences.

While `PMLS` and `PCT_OT` provide equivalent results to our method in terms of metrics measures, `PMLS` introduces visual artifacts if the input images are not registered correctly (Figure 6), while our method is robust

 ISBN 978-0-9934207-5-7

	PSNR ↑						
	IDT 2007	PMLS 2014	GPS/LCP 2018	FGPS/LCP 2018	L2 2019	PCT_OT 2019	OT_NW 5x5
Gangnam1	25.354	35.725	24.048	23.936	35.358	31.479	33.565
Gangnam2	27.116	36.553	25.952	25.944	35.524	35.502	33.627
Gangnam3	22.372	35.007	21.908	21.913	33.284	26.393	28.217
Illum	19.822	20.167	19.785	19.960	19.079	20.306	20.858
Building	20.554	22.634	22.736	22.769	20.499	25.019	24.039
Playground	27.184	27.835	25.501	25.436	27.647	28.482	28.491
Flower1	24.238	26.981	23.765	23.706	26.857	25.186	27.158
Flower2	25.417	25.760	25.259	25.223	25.772	26.373	26.497
Tonal1	30.082	37.215	31.617	31.413	37.332	37.044	37.151
Tonal2	27.992	31.508	25.062	25.087	31.356	32.049	31.579
Tonal3	29.575	36.246	28.136	28.065	36.644	33.793	35.014
Tonal4	28.605	34.521	28.852	28.848	34.344	33.819	35.320
Tonal5	30.330	35.260	29.580	29.448	34.303	36.437	36.616
Mart	22.747	24.742	23.183	23.196	24.450	24.509	25.189
Sculpture	29.884	32.062	29.037	28.820	32.067	31.237	32.735
Mean	26.085	30.814	25.628	25.584	30.301	29.842	30.404
SE	0.905	1.459	0.841	0.821	1.518	1.306	1.291

(a)

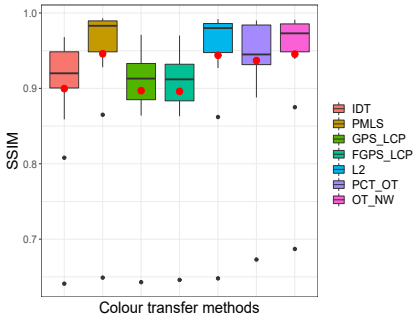


(b)

Figure 2: Metric comparison, using PSNR [14]. (a) Red, blue, and green indicate 1^{st}, 2^{nd}, and 3^{rd} best performance respectively in the table (higher values are better), (b) visualized in box plot (best viewed in colour and zoomed in).

	SSIM ↑						
	IDT 2007	PMLS 2014	GPS/LCP 2018	FGPS/LCP 2018	L2 2019	PCT_OT 2019	OT_NW 5x5
Gangnam1	0.900	0.992	0.892	0.891	0.990	0.964	0.973
Gangnam2	0.920	0.993	0.909	0.909	0.986	0.980	0.976
Gangnam3	0.859	0.991	0.873	0.864	0.980	0.930	0.959
Illum	0.641	0.649	0.643	0.646	0.648	0.673	0.687
Building	0.808	0.865	0.864	0.863	0.862	0.888	0.875
Playground	0.920	0.940	0.878	0.876	0.939	0.939	0.943
Flower1	0.909	0.967	0.913	0.912	0.966	0.926	0.959
Flower2	0.901	0.928	0.894	0.894	0.927	0.933	0.939
Tonal1	0.953	0.988	0.971	0.970	0.987	0.988	0.991
Tonal2	0.968	0.987	0.926	0.926	0.986	0.988	0.986
Tonal3	0.962	0.992	0.947	0.946	0.992	0.987	0.990
Tonal4	0.944	0.983	0.932	0.932	0.983	0.981	0.985
Tonal5	0.965	0.986	0.953	0.954	0.985	0.990	0.991
Mart	0.904	0.957	0.925	0.925	0.956	0.941	0.954
Sculpture	0.942	0.971	0.934	0.932	0.972	0.945	0.974
Mean	0.900	0.946	0.897	0.896	0.944	0.937	0.946
SE	0.022	0.023	0.020	0.020	0.023	0.020	0.020

(a)



(b)

Figure 3: Metric comparison, using SSIM [15]. (a) Red, blue, and green indicate 1^{st}, 2^{nd}, and 3^{rd} best performance respectively in the table (higher values are better), (b) visualized in box plot (best viewed in colour and zoomed in).

to registration errors. Note that although the accuracy of the PSNR, SSIM, CID and FSIMc metrics relies on the fact that the input images are registered correctly; if this is not the case, these metrics may not accurately capture all artifacts (Figures 7 and 6). In addition, due to the Nadaraya Watson smoothing step in our algorithm, our approach allows us to create a smoother colour transfer result, and can also alleviate JPEG compression artifacts and noise (cf. Figure 6 for comparison). `PCT_OT` can also create shadow artifacts when there are large changes between target and source images (Figure 6, in example 'building'), while our method `OT_NW` can correctly transfer colours between images that contain significant spatial differences and alleviates the shadow artifacts, as can be seen in Figure 6 with examples 'illum', 'mart' and 'building'.

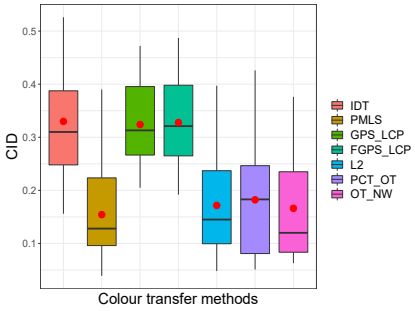
5 Conclusion

Several contributions to colour transfer with OT have been made in this paper, showing quantitative and qualitative improvements over state of the art methods. In particular, first, correspondences information as well as colour content of pixels are both encoded in the high dimensional feature vectors, and second, we introduced smoothing as part of the iterative algorithms for solving optimal transport namely Iterative Distribution Transport (IDT) and its variant the Sliced Wasserstein Distance (SWD). The algorithm allows denoising, artifact removal as well as smooth colour transfer between images that may contain large motion changes.

 ISBN 978-0-9934207-5-7

	CID ↓						
	IDT 2007	PMLS 2014	GPS/LCP 2018	FGPS/LCP 2018	L2 2019	PCT_OT 2019	OT_NW 5x5
Gangnam1	0.252	0.040	0.226	0.222	0.048	0.085	0.088
Gangnam2	0.268	0.039	0.291	0.292	0.089	0.068	0.109
Gangnam3	0.496	0.108	0.472	0.487	0.193	0.261	0.267
Illum	0.386	0.390	0.395	0.396	0.397	0.377	0.376
Building	0.374	0.228	0.313	0.321	0.249	0.183	0.275
Playground	0.440	0.238	0.443	0.471	0.254	0.209	0.221
Flower1	0.389	0.163	0.396	0.400	0.174	0.285	0.194
Flower2	0.337	0.245	0.322	0.323	0.266	0.218	0.201
Tonal1	0.310	0.101	0.285	0.308	0.111	0.097	0.063
Tonal2	0.288	0.128	0.351	0.347	0.145	0.099	0.118
Tonal3	0.244	0.079	0.294	0.294	0.081	0.077	0.079
Tonal4	0.240	0.108	0.248	0.238	0.107	0.065	0.065
Tonal5	0.156	0.091	0.205	0.192	0.092	0.051	0.067
Mart	0.526	0.219	0.405	0.402	0.225	0.426	0.249
Sculpture	0.242	0.137	0.213	0.224	0.143	0.232	0.120
Mean	0.330	0.154	0.324	0.328	0.172	0.182	0.166
SE	0.027	0.024	0.022	0.023	0.024	0.031	0.025

(a)

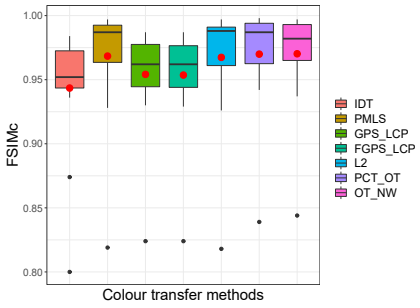


(b)

Figure 4: Metric comparison, using CID [16]. (a) Red, blue, and green indicate 1^{st}, 2^{nd}, and 3^{rd} best performance respectively in the table (lower values are better), (b) visualized in box plot (best viewed in colour and zoomed in).

	FSIMc ↑						
	IDT 2007	PMLS 2014	GPS/LCP 2018	FGPS/LCP 2018	L2 2019	PCT_OT 2019	OT_NW 5x5
Gangnam1	0.936	0.986	0.944	0.943	0.985	0.972	0.979
Gangnam2	0.952	0.992	0.962	0.962	0.988	0.990	0.986
Gangnam3	0.946	0.992	0.962	0.961	0.990	0.987	0.982
Illum	0.800	0.819	0.824	0.824	0.818	0.839	0.844
Building	0.874	0.928	0.930	0.929	0.926	0.942	0.937
Playground	0.950	0.958	0.933	0.932	0.955	0.956	0.960
Flower1	0.954	0.975	0.968	0.967	0.976	0.971	0.977
Flower2	0.941	0.950	0.945	0.945	0.949	0.954	0.956
Tonal1	0.964	0.997	0.986	0.986	0.997	0.998	0.997
Tonal2	0.984	0.993	0.973	0.973	0.992	0.993	0.992
Tonal3	0.979	0.997	0.984	0.983	0.997	0.997	0.995
Tonal4	0.966	0.989	0.972	0.973	0.990	0.995	0.994
Tonal5	0.980	0.994	0.987	0.987	0.993	0.998	0.997
Mart	0.946	0.969	0.960	0.959	0.967	0.969	0.970
Sculpture	0.980	0.987	0.982	0.980	0.988	0.988	0.987
Mean	0.943	0.968	0.954	0.954	0.967	0.970	0.970
SE	0.012	0.012	0.010	0.010	0.012	0.010	0.010

(a)



(b)

Figure 5: Metric comparison, using FSIMc [17]. (a) Red, blue, and green indicate 1^{st}, 2^{nd}, and 3^{rd} best performance respectively in the table (higher values are better), (b) visualized in box plot (best viewed in colour and zoomed in).

Figure 6: A close up look at some of the results generated using the `PMLS` [2], `L2` [13], `PCT_OT` [3] and our algorithm `OT_NW` (best viewed in colour and zoomed in).

 ISBN 978-0-9934207-5-7

Figure 7: A close up look at some of the results generated using the `IDT` [9], `PMLS` [2], `GPS/LCP` and `FGPS/LCP` [12], `L2` [13], `PCT_OT` [3] and our algorithm `OT_NW`. The results are presented in image mosaics, created by switching between the source (or the result i.e the transformed source) and the target image column wise, if the colour transfer is accurate, the resulting mosaic should look like a single image (best viewed in colour and zoomed in).

 ISBN 978-0-9934207-5-7

Acknowledgments

This work is partly funded by a scholarship from Umm Al-Qura University, Saudi Arabia, and in part by a research grant from Science Foundation Ireland (SFI) under the Grant Number 15/RP/2776, and the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) that is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

[1] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, Aug 2007.

[2] Y. Hwang, J. Lee, I. S. Kweon, and S. J. Kim. Color transfer using probabilistic moving least squares. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3342–3349, June 2014.

[3] H. Alghamdi, M. Grogan, and R. Dahyot. Patch-based colour transfer with optimal transport. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, Sep. 2019. `https://github.com/leshep/PCT_OT`.

[4] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, May 2011. `https://people.csail.mit.edu/celiu/SIFTflow/`.

[5] C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[6] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, Nov 2000.

[7] F. Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, pages 99–102, 2015.

[8] C. Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

[9] F. Pitié, A. C. Kokaram, and R. Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1):123 – 137, 2007. `https://github.com/frcs/colour-transfer`.

[10] J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer Berlin Heidelberg, 2012.

[11] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, Jan 2015.

[12] F. Bellavia and C. Colombo. Dissecting and reassembling color correction algorithms for image stitching. *IEEE Trans. Image Process.*, 27(2):735–748, Feb 2018.

[13] M. Grogan and R. Dahyot. L2 divergence for robust colour transfer. *Computer Vision and Image Understanding*, 181:39–49, 2019. `https://github.com/groganma/gmm-colour-transfer`.

[14] D. Salomon. *Data compression: the complete reference*. Springer Science & Business Media, 2004.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, April 2004.

[16] J. Preiss, F. Fernandes, and P. Urban. Color-image quality assessment: From prediction to optimization. *IEEE Trans. Image Process.*, 23(3):1366–1378, March 2014.

[17] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Trans. Image Process.*, 20(8):2378–2386, Aug 2011.

[18] I. Lissner, J. Preiss, P. Urban, M. S. Lichtenauer, and P. Zolliker. Image-difference prediction: From grayscale to color. *IEEE Trans. Image Process.*, 22(2):435–446, Feb 2013.

[19] M. Oliveira, A. D. Sappa, and V. Santos. A probabilistic approach for color correction in image mosaicking applications. *IEEE Trans. Image Process.*, 24(2):508–523, Feb 2015.

[20] J. Park, Y. Tai, S. N. Sinha, and I. S. Kweon. Efficient and robust color consistency for community photo collections. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 430–438, June 2016.

[21] M. Xia, J. Y. Renping, X. M. Zhang, and J. Xiao. Color consistency correction based on remapping optimization for image stitching. In *IEEE Int. Conf. on Computer Vision Workshops (ICCVW)*, pages 2977–2984, Oct 2017.

Defect Exclusive Custom Vocabulary for Classification

Terence Sweeney, Sonya Coleman, Dermot Kerr

School of Computing, Engineering and Intelligent Systems,
Ulster University
Londonderry, Northern Ireland

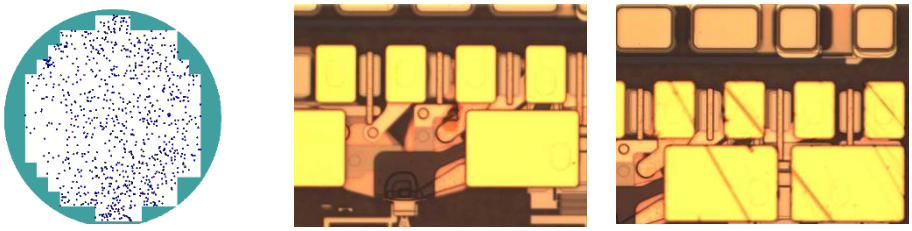
Abstract

Automated inspection has become a vital part of quality control in many industries, including during semiconductor wafer production. Current processes often focus on finding defects by comparing images with a 'golden' image pixel to pixel or, more recently, using shallow or deep learning based approaches. We present an alternative approach which uses the Bag of Visual Words technique to determine local features that correspond to specific defects within a wafer image, known as a custom vocabulary. Using this custom vocabulary combined with machine learning, we can characterise and accurately classify defects found on wafer images.

Keywords: Defect Detection, Local Features, Classification, Bag of Visual words, Machine Vision

1 Introduction

Semiconductor wafers are a component used in most electronic devices, including phones and hard drive media. During the manufacture of these wafers' inspection is vital to detect defects and ensure high quality. There are a multitude of methods that have been proposed for detecting these defects, with many techniques focusing on defects present across the whole wafer. In this case, when defects are detected they are marked on a wafer bin map (Figure 1(a)) to identify the total amount of defects present. This approach has been tested extensively [Ooi 2013, Mital, 1991] and is very good when looking for widespread defects across a production line and removing an irreparable product early in the process. However, it is sometimes desirable to not just determine the location of a defect but also to classify the defect type as some defects can be repaired with a cleaning phase, increasing overall wafer yield on the production line and reducing waste, which is critical in today's competitive world. Classifying types of defects is very difficult using the wafer bin map, however high-resolution images of individual defects on single dies are often taken across the production line. An example of a high-resolution image is illustrated in Figure 1 (b) and a high resolution defect image in Figure 1(c).



(a) **(b)** **(c)**

Figure 1: (a) Wafer bin map with detected defects coloured blue, (b) Single high-resolution die (Golden) image and (c) a scratch defect image

When considering the use of die images, most previous work is focussed on the use of global image features. Tobin's [Tobin, 2001] content based image retrieval golden image comparison is an example of a method commonly employed by most of the prolific Automated Defect Classification (ADC) machines [Tarasemi, 2019, Chou 1997] with it commonly being known as the 'Golden Image' approach. The images used in the experiments presented in this paper are captured by an ADC machine known as the Rudolph NSX105. The Rudolph NSX105 [Tarasemi, 2019] is a commonly used industry standard inspection device which uses the golden image approach. An initial set of golden images are manually selected and added to the inspection system which then uses its initial stage camera to strobe over the wafer, comparing captured images with the corresponding set of golden images. Hence, every single time a wafer product is updated, or a new wafer product developed, a new set of golden images must be created, and the system updated. Another critical problem with the NSX105's inspection process is that while it can

 ISBN 978-0-9934207-5-7

determine a defect at a specific location, it cannot determine the type of defect that has been found on the die ; hence the severity of the defect is unknown. This may result in more serious defect types, such as corrosion damage on critical parts, going unnoticed until later in the production process or products being removed from the production line with defects that are not critical which can be costly. Therefore, in this paper, we propose defect exclusive visual words for defect classification by using local image features. The combination of these custom visual word vocabularies along with machine learning, enables accurate defect classification which is a promising step towards an automated inspection algorithm.

2 Methodology

2. 1 Bag of Visual Words

The proposed ADC system is based on the use of the bag-of-visual-words method. This is an extension of the bag-of-words (BoW) text retrieval method making it suitable for use with image data. When using the BoW technique on a text document, a normalised histogram of word counts is computed as well as a sparse-term vector, where each bin corresponds to a term in the vocabulary. In the context of image data, this technique [Csurka, 2004] enables the generalisation of local image feature descriptors which are similar.

2.2 Support Vector Machines

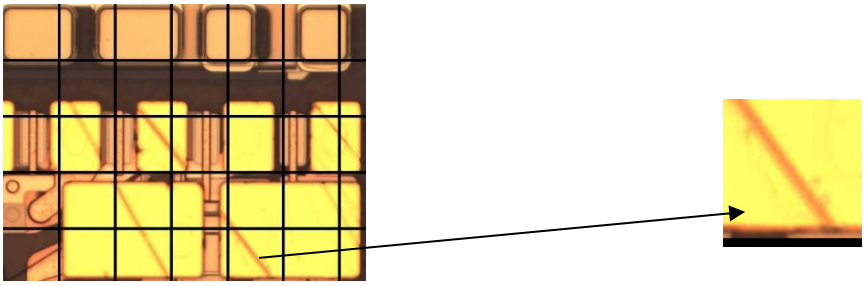
Support vector machines are a machine learning classifier that find the most efficient hyperplane to separate data into a number of classes. It can do this utilising three different types of mathematical kernel function that take the data and transform it into a useful classification metric. The kernel types are: linear, polynomial and radial basis function (RBF). Linear is often useful for binary classification tasks. The polynomial kernel is a popular approach often used in image processing. The RBF kernel is often seen as a general purpose kernel. In addition, a penalty parameter, known as C, is used to adjust how the SVM avoids misclassification of each training example. This parameter is useful when working with datasets where features are homogenous, such as the images used in the experiments here, as we can optimise the classification response. Support Vector Machines have been used in conjunction with Bag of Visual words in previous research [Henschel, 2014] with good results.

2.3 Custom Vocabulary

A custom vocabulary is an augmentation of bag-of-visual-words, where the visual codebook that is created is augmented or pruned to focus on the features of most interest in the image. Examples of this approach include using two codebooks [Devi, 2017], where two vocabularies are created using different training set classes before being tested in order to observe which vocabulary returns the highest accuracy for each testing class. Therefore, selecting the features which are important results in a stronger final codebook, which can be seen as comparable to a boosting classifier.

The proposed custom vocabulary is based on the use of the SURF feature detector and descriptor as previous work found that it outperforms SIFT for this purpose [Sweeney 2019]. Although the results from this experiment were promising we observed that when using full images, several visual words focussed on background features, such as corners and edges of parts, rather than defect features that we were looking for. Hence, we decided to produce a defect-only image dataset by removing the background and focusing on only the defect. To do this an automated cropping system was developed. The original 600*600 image is segmented into 35 images of 100*100 pixels in size by cropping using a sliding window. We then create a defect only dataset by using a subset of these images. An example of a scratch defect image with cropping locations is illustrated in Figure 2(a) and a selected defect-only image is illustrated in Figure 2(b).

 ISBN 978-0-9934207-5-7



(A) (B)

Figure 2–Example of the grid cropping system (A) and the resultant image (B)

In the proposed methodology the SURF interest point detector is used to obtain key-points k_n and corresponding SURF descriptors d_n where $i = 1 \ldots n$ such that a keypoint is represented as:

$$k_i = (x_i, y_i, d_i),$$

where x and y are the coordinates of a point in an image. The SURF keypoint descriptors are of 64 dimensions. An image feature set S can be represented by the set of local keypoint descriptors such that

$$S_I = \{k_1, k_2, \ldots, k_n\}.$$

where $I = 1 \ldots m$ and m is the number of images in the image set. The BoVW algorithm B is considered to quantize the descriptor $d \in R^I$

$$B: R^I \rightarrow [1, K] d \rightarrow B(d).$$

The B assigns descriptor $d \in R^I$ to the appropriate cluster K, where each cluster represents a visual word and the set of visual words is the initial defect vocabulary. Using the cropped images, we use all the words as they are all focussed on the defect only.

With the use of this automated cropping approach the total number of detected key-points k_n is restricted to be between 10-80 due to the size of the defect-only images (100x100px), thus the number of clusters k becomes more specialised and the number of images in the image set m exponentially grows from 70 images to 316 in this case.

3 Classification results

The methodology is applied to two classes, 70 scratch defect images and 100 control non-defect images. For testing the system, we reserve 20 images from each class and the remainder is used to train the system. The scratch defect images are subsequently cropped, resulting in 316 defect-only images. The BoVW methodology is applied to obtain the defect exclusive custom vocabulary. Next, we apply the custom vocabulary to the complete (uncropped) images for both the control non-defect and scratch defect classes and use the resulting BoVW histograms as input to a Support Vector Machine to train a binary classifier as scratch or no defect. System performance is then tested using the 20 scratch defect and 20 control non-defect training images. The overall classification results are recorded in Table 1.

SVM		
	Full Images	Cropped Defects
Linear C=1	50%	50%
Linear C=10	50%	50%
Linear C=100	75%	95%
Poly C=1	50%	50%
Poly C=10	62%	87%
Poly C=100	50%	**95%**
RBF C=1	75%	80%
RBF C=10	**80%**	**95%**
RBF C=100	**85%**	**97%**

Table 1 – Accuracy Results for SVM

 ISBN 978-0-9934207-5-7

Table 1 shows the results of the SVM classification for this experimentation where 3 kernel types of Linear, Polynomial and Radial Basis Function were used along with 3 different variations of the penalty parameter C to observe which of these combinations returned the most promising results. The results demonstrate that an improvement in performance is obtained when a defect exclusive custom vocabulary is utilised, compared with a general vocabulary for a full image method which contains background information. As seen from Table 1, the accuracy gained from this method is higher than that of the full image method, resulting in scores of 95% and 97% from the Polynomial and Radial Basis Function kernels whereas it only increased to a max of 85% in the full image experiment.

4 Conclusion and Further work

This paper presents an efficient approach for the development of a custom vocabulary for defects in semiconductor wafers by utilizing the Bag-of-Visual Words approach. The *custom vocabulary* is combined with a number of support vector machine algorithms to determine defects in the semiconductor wafers. The results for this approach demonstrate high accuracy for one defect type and therefore further work with focus on making this a multiclass problem with further defect types being included.

Acknowledgement

This work was funded by a DfE CAST scholarship in collaboration with Seagate Technology. We would also like to thank Seagate Technology for providing the image dataset used in the research.

References

[Tarasemi, 2019] Tarasemi, "Rudolph August NSX 105 Automated Defect Inspection," 2019. [Online]. Available:https://www.tarasemi.com/product/rudolph-august-nsx-105-automated-defect-inspection/. [Accessed 2 June 2019].

[Csurka, 2004] G. e. a. Csurka, "Visual categorization with bags of keypoints," in Workshop on Statistical learning in computer vision, 2004.

[Hentschel, 2014] C. Hentschel and H. Sack, "Does one size really fit all? Evaluating classifiers in Bag-of-Visual-Words classification," in 14th International Conference on Knowledge Technologies and Data-driven Business., Graz, Austria, 2014.

[Mital and Tobin, 1991] D. P. Mital and E. K. Teoh, "Computer based wafer inspection system," in Proceeding of international conference on industrial electronics, control and instrumentation, Kobe, 1991.

[Ooi, 2013] M. P.-l. e. a. Ooi, "Defect cluster recognition system for fabricated semiconductor wafers," Engineering Applications of Artificial Intelligence, pp. 1029-1043, 2013.

[Sweeney, 2019] T. Sweeney, S. Coleman and D. Kerr, "A Machine Learning Approach to Wafer Defect Classification using Bag of Visual Words," in Irish Machine Vision and Image Processing Conference. 2019, Dublin, 2019.

[Devi et al., 2017] S. Varsha Devi and e. al, "Better object recognition using bag of visual word model with compact vocabulary," in 13th International Conference on Emerging Technologies (ICET), Islamabad, 2017.

[Chou, 1997] P. Chou and e. al, "Automatic defect classification for semiconductor manufacturing," Machine Vision and Applications, vol. 9, no. 4, pp. 201-214, 1997.

[Tobin et al., 2001] K. Tobin, T. Karnowski and F. Lakhani, "Integrated applications of inspection data in the semiconductor manufacturing environment," Metrology-based Control for Micro-Manufacturing, pp. 31-41, 2001.

Keypoint Autoencoders: Learning Interest Points of Semantics

Ruoxi Shi, Zhengrong Xue, and Xinyang Li

Shanghai Jiao Tong University

Abstract

Understanding point clouds is of great importance. Many previous methods focus on detecting salient keypoints to identity structures of point clouds. However, existing methods neglect the semantics of points selected, leading to poor performance on downstream tasks. In this paper, we propose Keypoint Autoencoder, an unsupervised learning method for detecting keypoints. We encourage selecting sparse semantic keypoints by enforcing the reconstruction from keypoints to the original point cloud. To make sparse keypoint selection differentiable, Soft Keypoint Proposal is adopted by calculating weighted averages among input points. A downstream task of classifying shape with sparse keypoints is conducted to demonstrate the distinctiveness of our selected keypoints. Semantic Accuracy and Semantic Richness are proposed and our method gives competitive or even better performance than state of the arts on these two metrics.

Keywords: 3D Keypoint Detecting, Machine Vision, Deep Learning

1. Introduction

Point cloud is considered to be an irregular data format [Qi et al., 2017]. Due to the large number of data points, processing a point cloud is often challenging. One way to handle this problem is to represent a large-scale dense point cloud with a relatively small-scale sparse keypoints. Traditional geometric based keypoint detectors like [Harris and Stephens, 1988] fail to extract semantic-rich information from an arbitrary object and their success heavily depends on their manual parameter selections.

Figure 1: **Example of Detected Keypoints.**

In the trend of deep learning, Qi et al. proposed PointNet [Qi et al., 2017], a novel neural network capable of supervised learning tasks on point clouds. Unfortunately, as the understanding towards which keypoints are of semantics may vary in person, it is virtually impossible to obtain ground-truth keypoint labels. As a result, few supervised data-driven methods prove to be successful in selecting keypoints from a point cloud. Recently, some unsupervised methods [Li and Lee, 2019][Yew and Lee, 2018] are proposed. However, they only stress high repeatability of keypoints and neglect semantics, thus behaving poorly for downstream tasks especially under small number of keypoints.

In view of the challenges above, we propose Keypoint Autoencoder (KAE). Autoencoders have long been utilized to reduce redundant information in signals. Similarly, we utilize them to reduce 'redundant' points in the point cloud to detect keypoints. We slightly modified the traditional autoencoder so that a set of keypoints is used as the latent representation instead of an arbitrary latent vector. To this end, the encoder outputs a set of probability distributions on input points, and a proposal module is invoked on the distributions to get the final keypoints. The decoder tries to reconstruct the original point cloud from the keypoints. To make the hard keypoint selection process differentiable, Soft Keypoint Proposal is proposed, the main idea of which is to compute an average among input points weighted by the probability distributions. Furthermore, on the basis of KAE, Auxiliary Classifier Keypoint Autoencoder (AC-KAE) is built where class information of the point cloud is added to assist downstream tasks and class-wise feature learning. In our framework, semantic information in detected keypoints is greatly encouraged since it is impossible to reconstruct each object with only shape information and no

 ISBN 978-0-9934207-5-7

semantics, especially under low quantities of keypoints demanded.

We verify the performance of our method with two different metrics: distinctiveness in terms of downstream classifier accuracy and semantic information quality. We propose **Semantic Accuracy** and **Semantic Richness** to comprehensively decide the quality of semantic information expressed by the keypoints. A Mean Opinion Score test is conducted using these two indexes. An example of keypoints detected with our method is shown in figure 1. It can be seen that points with semantics (head and tail, roots of engines, tip of wings) are detected as keypoints.

2. Related Work

PointNet [Qi et al., 2017] and its variant PointNet++ [Qi et al., 2017] exploit max pooling, a symmetric function, to process unordered, varying-length point clouds into a regular, fixed-length global feature vector, which enables deep learning networks to consume point clouds. They and many other PointNet-based networks are proposed to deal with various 3D vision tasks such as classification, segmentation and pose estimation. However, there are still few unsupervised methods designed for extracting keypoints from a point cloud. So far, USIP [Li and Lee, 2019] is considered to be the state-of-the-art method. USIP takes advantage of probabilistic chamfer loss to obtain highly repeatable keypoints. However, keypoints generated from USIP don't necessarily contain much semantic information, which makes USIP less helpful for actual downstream tasks. Meanwhile, S-NET [Dovrat et al., 2019] is aware of downstream tasks, which certainly avoids the demerits of USIP. However, the nature that it is tailored to specific tasks disqualifies S-NET to be a keypoint detector, as the point set it selects changes w.r.t. different downstream tasks. In comparison, our method takes semantics into consideration, which is helpful for downstream tasks. Furthermore, the point set we select only depends on the raw point cloud, and thus it is invariant to specific tasks.

3 Method

The Pipeline. There are four major modules in our design: an encoder, a decoder, a proposal module to obtain keypoints from the encoder, and an optional auxiliary classifier. The architecture is shown in figure 2.

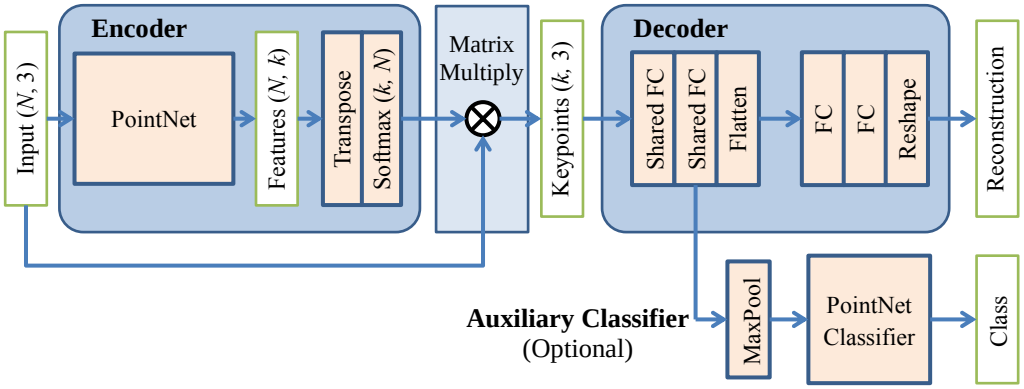


Figure 2: **Architecture of Keypoint Autoencoder.**

Encoder. Suppose we want the system to extract k keypoints. Then the encoder inputs a point cloud (shape $[N, 3]$), which goes through a PointNet [Qi et al., 2017] to obtain point-wise features (shape $[N, k]$). After that it is transposed and a softmax activation is applied to obtain k probability distributions (shape $[k, N]$).

Soft Keypoint Proposal. This small module is invoked on the distributions to get k keypoints (shape $[k, 3]$). These points act as the input to decoder. However, no gradient would be propagated into the encoder if keypoints were directly sampled from input. Upon this we propose the Soft Keypoint Proposal: Input points are weighted by the distributions obtained from the encoder, the weighted average of which is passed into the decoder. Formally, if we denote $\boldsymbol{K_s}(i)$ as the soft keypoints, $\boldsymbol{X_j}$ as the j-th input point (each a 3-D vector), and $D_{k\times N}$ is the outputs of the encoder (after applying softmax), then

 ISBN 978-0-9934207-5-7

$$\boldsymbol{K_s}(i) = \sum_{j=1}^{N} D_{ij}\boldsymbol{X}_j\,, i = 1, 2, 3, \ldots, k$$

Note that it is equivalent to a matrix multiplication as shown in figure 2.

In the detection process, an ordinary selection operation is performed according to the probability distributions obtained from the encoder. Points with highest probabilities are selected with NMS (Non Maximal Suppression) until the number of points selected reaches the requested value. These points act as the final keypoints detected.

Decoder. The decoder is composed of fully connected (FC) layers. The first block shares weights among points to get point-wise features (shape $[k, 64]$). The features are reconstructed into a point cloud of same size as the input point cloud (shape $[N, 3]$) by the second block. Chamfer loss [Barrow and Tenenbaum, 1977] is applied between the reconstructed point cloud and the original one:

$$L_c = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$

Where $S_1, S_2 \subset \boldsymbol{R}^3$ are original and reconstructed point clouds, respectively.

Auxiliary Classification. In addition to the point cloud itself, meta information such as the class of the object that yields the point cloud is usually available in the dataset, which may help improve the distinctiveness of selected keypoints. Some common features shared by a whole class of point clouds such as symmetries and object structures can be learned to improve semantics of the keypoints. Thus we propose Auxiliary Classifier Keypoint Autoencoder (AC-KAE). A PointNet classifier branches from the keypoint feature layer and contributes gradient from classification results to the network. The branch is marked as optional in the figure.

4. Experiment

In this section, we demonstrate that our method shows strong performance detecting keypoints with semantic information of high distinctiveness or is even better than state of the art.

4.1. Distinctiveness: Downstream Classifier Accuracy

Distinctiveness is an important metric of the keypoint detector. A set of distinctive keypoints should describe an object uniquely and one should be able to clearly classify an object based on these sparse keypoints. Therefore, we come up with a downstream task described below.

For each point cloud in the ModelNet40 [Wu et al., 2015] dataset, different detectors are applied to obtain 8 keypoints. With only these keypoints as input, a same PointNet classifier is trained. The overall accuracy is evaluated after 50 epochs. For the KAE based models we proposed, the soft keypoints are adopted.

	FPS	S-NET	USIP	KAE (ours)	AC-KAE (ours)
Accuracy	68.3%	83.6%	56.8%	83.7%	**85.7%**

Table 1: **Accuracy of Downstream PointNet Classifier on the ModelNet40 Classification Problem.**

Our methods achieve better results than previous models, which means that our autoencoder based detectors are better in distinctiveness. The USIP detector trains against repeatability as [Dovrat et al., 2019] stated in the paper, and it can be concluded that the USIP detector loses distinctiveness with a small quantity of requested keypoints.

4.2. Semantic Information

Semantic information is a vital property of interest points. We propose two quantities, namely **Semantic Accuracy** and **Semantic Richness** to describe the accuracy and completeness of semantic information expressed with the keypoints selected.

It is hard to formally define these metrics. Basically, high **Semantic Accuracy** is achieved when the points selected are those of semantics in the point cloud. High **Semantic Richness** is achieved when most of the

 ISBN 978-0-9934207-5-7

semantics are covered. However, the relation between computable metrics and these two quantities is unclear and may depend on specific point cloud structure. As a result the metrics can only be subjectively compared among different keypoint detectors. Here some examples are given in figure 3 together with a basic analysis on the semantic information of selected keypoints from the three models.

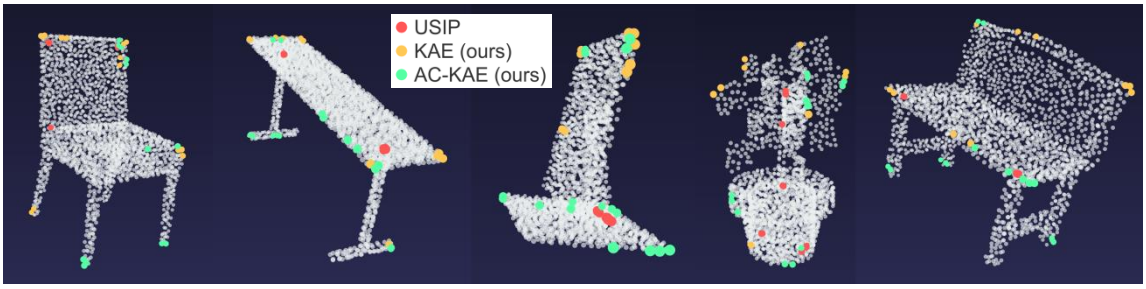


Figure 3: **Examples of Keypoints Detected by USIP, KAE and AC-KAE Detectors.** It can be seen that the keypoints from USIP mainly fall on joints near the central area, making their semantic information accurate but not rich. The keypoints from our methods are mainly at corners of the cloud, but also appear on some major joints. It also demonstrates AC-KAE model's interesting ability to make use of the symmetries from classes of the point cloud, thus sparing more keypoints to represent extra semantic information.

Mean Opinion Score (MOS) Test. To further evaluate the semantic information contained in the keypoints generated by our proposed models, we conducted an MOS test among 28 undergraduate students and engineers majoring in relative fields. Detectors are used to extract 16 keypoints from 10 randomly chosen point clouds from the ModelNet40 [Wu et al., 2015] dataset. After viewing the keypoints together with the original point cloud, the subjects are asked to grade the semantic accuracy and richness of keypoints in a 10-point scale score. To ensure the effectiveness of the test, the subjects are all blind to the labels.

	USIP	KAE (ours)	AC-KAE (ours)
Semantic Accuracy	4.571	**6.049**	5.546
Semantic Richness	4.157	**5.814**	5.556

Table 2: **Subjective 10-scale Mean Opinion Scores.**

It can be seen from the table that our KAE method achieves highest scores, and is significantly better than USIP. However, the AC-KAE model performs worse than the KAE model. A possible cause is that our AC-KAE method makes use of the symmetries and avoids points with similar semantics, so some parts of point clouds are neglected, leading to a drop in subjective opinion scores.

References

[Qi et al., 2017] Qi, C.R., Su, H., Mo, K., & Guibas, L.J. (2017). *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 77-85.

[Harris and Stephens, 1988] Harris, C.G., & Stephens, M. (1988). *A Combined Corner and Edge Detector*. Alvey Vision Conference.

[Li and Lee, 2019] Li, J., & Lee, G.H. (2019). *USIP: Unsupervised Stable Interest Point Detection From 3D Point Clouds*. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 361-370.

[Yew and Lee, 2018] Yew, Z.J., & Lee, G.H. (2018). *3DFeat-Net: Weakly Supervised Local 3D Features for Point Cloud Registration*. ArXiv, abs/1807.09413.

[Qi et al., 2017] Qi, C.R., Yi, L., Su, H., & Guibas, L.J. (2017). *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. NIPS.

[Dovrat et al., 2019] Dovrat, O., Lang, I., & Avidan, S. (2019). *Learning to Sample*. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2755-2764.

[Barrow and Tenenbaum, 1977] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf. *Parametric correspondence and chamfer matching: Two new techniques for image matching*. In Int'l Joint Conf. of Artif. Intel., pages 659–663, 1977.

[Wu et al., 2015] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang and J. Xiao. *3D ShapeNets: A Deep Representation for Volumetric Shapes*. Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)

Projective Texture Mapping on Reconstructed Scenes

William Clifford* and Charles Markham

Department of Computer Science, Maynooth University, Maynooth, Co. Kildare, Ireland.

Abstract

This paper describes a method to integrate projective texture mapping into the existing frame work of a typical 3D mapping system. The approach assumes the geometry of the 3D model provides ground truth. The pose of the sensor measurement is adjusted in order to optimize registration of projected texture with the reconstructed surface. In addition the following technical challenges were addressed in this paper: reverse projection, re-projection error, and depth culling with respect to the texture.

Keywords: 3d reconstruction, texturing, point clouds

1 Introduction

Building 3D point clouds and models based on real world measurements has been an active area of research in the vision, robotics, and photogrammetry communities for many years [Triggs et al., 1999, Klein and Murray, 2007]. Typically, these algorithms produce a set of poses of the sensors and a set of locations for landmarks in a 3D map (e.g. feature correspondences). Active camera's, such as the Kinect, can produce high density point clouds and have lead to significant advances in the area of 3D reconstruction [Izadi et al., 2011, Whelan et al., 2015]. These algorithms provide an opportunity to create 3D virtual representations of the real world. Improvements to methods of texturing could enhance the experience for users. The work preseneted here will examine a method of enhancing visual fidelity of these 3D scenes.

2 State of the Art

A popular technique for computing colour on surfaces is to fuse them, this is done by projecting many views into the scene and creating a weighted average of RGB values for each vertex [Curless and Levoy, 1996]. This approach produces lower resolution or smoothed surface intensities as a consequence of averaging. This is the approach employed in most state of the art algorithms [Izadi et al., 2011, Whelan et al., 2015]. Some techniques reduce over smoothing by optimizing local camera pose photometrically by matching surface colours to image colours [Zhou and Koltun, 2014]. These optimizations have since been expanded upon in [Bi et al., 2017, Huang et al., 2017]. These are static solutions which do not change their appearance depending on the view.

Projective texture mapping is a method of projecting an image onto a surface, a similar method is used in computer graphics to produce shadow maps [Segal et al., 1992]. It has seen attention in the area of texturing 3D models but the emphasis to date focuses on selecting the view to project and noise reduction caused by projection error [Debevec et al., 1998, Velho and Sossai Jr, 2007, Waechter et al., 2014].

*William.Clifford@mu.ie

 ISBN 978-0-9934207-5-7

3 Preliminary

Elastic fusion was used to reconstruct a 3D model of a scene. Its inputs were the camera intrinsics of the sensor used, and a set of RGB-D frames from a depth camera. The outputs were a 3D map and a set of odometry measurements corresponding to the RGB-D frames. Consider a map composed of vertices generated from some mapping system, in our case from elastic fusion [Whelan et al., 2015], call this $\mathcal{M}$. This map is composed of a set of vertices where each vertex has a position $\mathbf{p} \in \mathbb{R}^3$, and colour $\mathbf{c} \in \mathbb{N}^3$.

$$\mathbf{P_t} = \begin{bmatrix} \mathbf{R_t} & \mathbf{t_t} \\ 0 & 1 \end{bmatrix} \in \mathbb{SE}_3 \quad (1)$$

The image space domain is defined as $\Omega \subset \mathbb{N}^2$, where an RGB-D frame was used. Depth maps $\mathcal{D}$ of depth pixels $d : \Omega \to \mathbb{R}$ and the colour image C of colour pixels $\mathbf{c} : \Omega \to \mathbb{N}^3$. The system also produces odometry for the camera which represents the camera's pose following construction of these vertices. Let these locations be represented by a rotation matrix $\mathbf{R} \in \mathbb{SO}_3$, and a translation $\mathbf{t} \in \mathbb{R}^3$. These are all considered in the global reference frame. These can be joined together into a transformation matrix as follows, where the subscript t is the pose at a given time, see Equation 1.

$$\mathbf{u} = \pi(\mathbf{K}\mathbf{P_t}\mathbf{p}) \quad (2)$$

Each camera also has a projection matrix associated with it. Let each of these be represented by a matrix $\mathbf{K}$ which functions as a mapping of points $f(x) : \mathbb{P}^3 \to \mathbb{P}^2$. This is also known as the camera's intrinsic parameters. To compute these projections the points used must be in homogeneous form. These matrices will allow for transformations between image space and world space, and vice versa. Vertices in the map at position $\mathbf{p} = [x, y, z]^\intercal$, can be mapped to an image point $\mathbf{u} \in \Omega$ for a given camera, Equation 2. Where $\pi(\mathbf{p}) = (x/z, y/z)^\intercal$ is the dehomogenisation of the point.

4 Method

4.1 Synthetic data capture

Synthetic data from ICL-NUIM were used for RGB-D measurements of a 3D model [Handa et al., 2014]. The sequence was the Living Room 'lr kt0' trajectory. Frame-to-frame tracking was enabled as it produced lowest root mean squared error for predicted poses when compared with the ground truth poses. The final map was outputted along with predicted odometry. The surfel map was then processed by a Poisson Mesh reconstruction using Meshlab, reducing the point count from 622367 to 253088 points [Kazhdan et al., 2006, Cignoni et al., 2008].

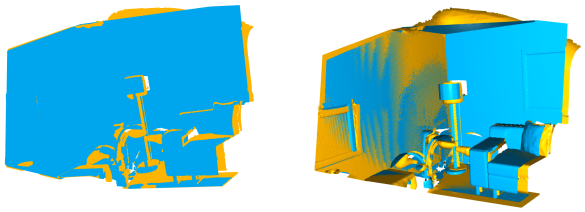


Figure 1: Reprojections of the depth map of the model at the estimated pose, $\mathcal{P}_t$, in yellow and reprojection of the depth sensor measurement for that pose, $\mathcal{D}_t$, in blue. Left image is the initial disparity between the point clouds. Right image is the reprojections after the poses have been optimized with ICP. Images were created using Open3d [Zhou et al., 2018]

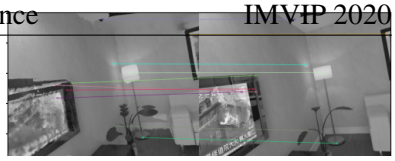
4.2 Improving Alignments

Elastic fusion utilized surface loop closure optimizations using a sparse deformation graph embedded in the surface rather than a probabilistic pose graph for transforming keyframes. Given that the model was complete, focus could be brought back into aligning poses with the model. The model, $\mathcal{M}$, was treated as ground truth and the transformation for the pose, $\mathbf{P}_t$, was optimized relative to the model. The depth map $\mathcal{P}_t$ of the model $\mathcal{M}$ was acquired, given the proposed pose from elastic fusion $\mathbf{P}_t$, $\forall\, t$. Each of these depth maps were compared against the sensor measurement depth map associated for these poses $\mathcal{D}_t$. The two depth maps were reprojected as though their poses were both at $\mathbf{P}_t$. Poses were then re-aligned using iterative closest point (ICP) algorithm to solve for a better transformation between the model and the predicted pose, $\mathbf{P}_t + \Delta\dot{\mathbf{P}}_t$. The most effective method was the point to plane implementation of ICP, given the geometry, see Figure 1, [Chen and Medioni, 1992].

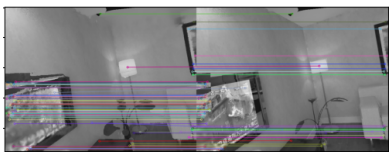
 ISBN 978-0-9934207-5-7

4.3 Texture mapping

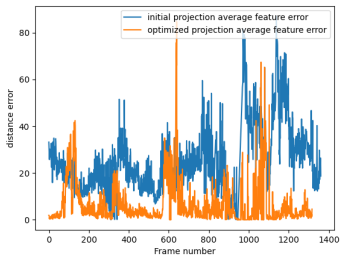
Assuming good odometry (small reprojection error between capture localization and the model), the colour values of the image C should re-project onto the surfaces of $\mathcal{M}$, and produce fewer visual artefacts. It is helpful to consider the odometry as a list of poses in the order of the frames captured of the form Equation (1). The calibration matrix, $\mathbf{K}$, is the same for all poses $\mathbf{P}_t$. Every previously captured pose could be expressed between all points $p \in \mathbb{R}^3$ in the map $\mathcal{M}$ with the frame corresponding to that pose. Similar to Equation (2), where $\mathbf{u}$ is the texel to be sampled for the surface. In projective texture mapping there is an issue of points behind the projector being textured, referred to as reverse projection [Everitt, 2001]. This is addressed by considering the points in eye-space and checking the condition: $\mathbf{P}_t p[z] > 0, \forall p$, to check the points are in front of the projector. Before allowing references to the frame to be made, following projective mapping, points were clipped to ensure that only real points are mapped onto the image C. The points were clipped if they lie outside of the clipping points (normalised device coordinates) for $p \in \mathbb{R}$ in the range $[-1, 1]$.



(a) Initial Projection.



(b) Optimized Projection.



(c) Error Measure.

Figure 2: (a) and (b) show single view feature matches. (c) the feature match error for every frame in the sequence.

5 Results

Using standard reconstruction, subsampling, and Poisson mesh reconstruction resulted in smoothed geometry and colours, left image of Figure 2a. Using projective texture mapping, the surface geometry can remain at a lower resolution without having to degrade the texture, right image of Figure 2a. It was attempted to rectify misalignment issues by using ICP Figure 2b. The measure of misalignment was acquired by computing SIFT features and matching them across the vertex colours image and the resulting projections Figures 2a and 2b, [Lowe, 1999]. The disparity between the features was measured as euclidean distance on the image planes of each image and was averaged. This measure was reported for the entire Living Room 'lr kt0' trajectory from ICL-NUIM dataset Figure 2c. For Figure 2a the average error was a distance of 20 ± 4.33 pixels (standard deviation: 14) with a fitness measure (number of features matched to projection versus total number of features in vertex colours image) of 0.018. For Figure 2b the average error was 0.68 ± 0.33 pixels (standard deviation: 3) with a fitness of 0.15.

6 Conclusion

On average the optimization used produces lower disparity of keypoint features while compared to the initial pose, Figure 2c. Disparity grows larger for the optimized projection on parts of the sequence with low geometric features, which is to be expected with ICP. In these cases it would be wiser to optimize based on colour features, if at all [Zhou and Koltun, 2014].

This paper has demonstrated a method for projective texture mapping onto models created in 3D mapping systems. It allows for subsampling of the geometry to reduce geometric complexity on dense point clouds without having to sacrfice texture quality of the surfaces. The model is treated as ground truth so that the sensor measurements can be fitted to that model in order to reduce visual artefacts. Improving the alignment of frames to model may enhance the results of view dependent algorithms [Debevec et al., 1998, Velho and Sossai Jr, 2007].

References

[Bi et al., 2017] Bi, S., Kalantari, N. K., and Ramamoorthi, R. (2017). Patch-based optimization for image-based texture mapping. *ACM Trans. Graph.*, 36(4):106–1.

[Chen and Medioni, 1992] Chen, Y. and Medioni, G. G. (1992). Object modeling by registration of multiple range images. *Image Vis. Comput.*, 10(3):145–155.

[Cignoni et al., 2008] Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G. (2008). Meshlab: an open-source mesh processing tool. In *Eurographics Italian chapter conference*, pages 129–136.

[Curless and Levoy, 1996] Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312.

[Debevec et al., 1998] Debevec, P., Yu, Y., and Borshukov, G. (1998). Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Workshop on Rendering Techniques*, pages 105–116. Springer.

[Everitt, 2001] Everitt, C. (2001). Projective texture mapping. *White paper, NVidia Corporation*, 4(3).

[Handa et al., 2014] Handa, A., Whelan, T., McDonald, J., and Davison, A. (2014). A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China.

[Huang et al., 2017] Huang, J., Dai, A., Guibas, L. J., and Nießner, M. (2017). 3dlite: towards commodity 3d scanning for content creation. *ACM Trans. Graph.*, 36(6):203–1.

[Izadi et al., 2011] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. (2011). Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *UIST '11 Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM.

[Kazhdan et al., 2006] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7.

[Klein and Murray, 2007] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee.

[Segal et al., 1992] Segal, M., Korobkin, C., Van Widenfelt, R., Foran, J., and Haeberli, P. (1992). Fast shadows and lighting effects using texture mapping. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 249–252.

[Triggs et al., 1999] Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (1999). Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer.

[Velho and Sossai Jr, 2007] Velho, L. and Sossai Jr, J. (2007). Projective texture atlas construction for 3d photography. *The Visual Computer*, 23(9-11):621–629.

[Waechter et al., 2014] Waechter, M., Moehrle, N., and Goesele, M. (2014). Let there be color! large-scale texturing of 3d reconstructions. In *European conference on computer vision*, pages 836–850. Springer.

[Whelan et al., 2015] Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., and Davison, A. (2015). Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems.

[Zhou and Koltun, 2014] Zhou, Q.-Y. and Koltun, V. (2014). Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Transactions on Graphics (TOG)*, 33(4):1–10.

[Zhou et al., 2018] Zhou, Q.-Y., Park, J., and Koltun, V. (2018). Open3d: A modern library for 3d data processing. cite arxiv:1801.09847Comment: http://www.open3d.org.

***Rb-PaSta*Net**: A Few-Shot Human-Object Interaction Detection Based on Rules and Part States

Shenyu Zhang, Zichen Zhu, Qingquan Bao

Shanghai Jiao Tong University

Abstract

Existing Human-Object Interaction (HOI) Detection approaches have achieved great progress on non-rare classes while rare HOI classes are still not well-detected. In this paper, we intend to apply human prior knowledge into the existing work. So we add human-labeled rules to ***PaSta*Net** and propose ***Rb-PaSta*Net** aimed at improving rare HOI classes detection. Our results show a certain improvement of the rare classes, while the non-rare classes and the overall improvement is more considerable.

Keywords: Human-Object Interaction, Body Part State, Rule-based Network

1 Introduction

When building an intelligent system, understanding human activities from still images plays a critical role. As a sub-task of visual relationship comprehension [10], Human-Object Interaction (HOI) infers types of interactions through retrieving human and object locations. Related to human and object understanding, HOI will boost activity understanding [6], imitation learning [1], etc.

Generally, this high-level cognition task is addressed in one-stage [3], i.e. directly mapping pixels to activity concepts. Closer to our work, Li *et al.* [8] takes advantages of *part-level semantics* and builds a human activity knowledge engine to infer interactiveness. However, nearly all state-of-the-art methods encounter few-shot classes' performance bottleneck due to rare training datasets.

In light of this, we propose a ***Rb-PaSta*Net** (Rule-based Part State Net) to augment the existing work *PaSta*Net [8]. With the introduction of human prior knowledge as rules, we believed ***Rb-PaSta*Net** would have more information about the physical world besides training data. Specifically, we manually label the weights of each human body part in 162 less-than-ten-shot HOI classes. Two versions are introduced: one consists of weights in Decimal numbers derived from the average of three authors' labels, the other Boolean numbers derived from the former version. These weights are added to part attentions, which means the importance of certain body parts in an HOI class, to strengthen the learning of few-shot HOI classes.

Finally, our method achieves some insignificant improvement on few-shot HOI classes, e.g. the Decimal version makes 0.13 mAP improvement of the rare classes on HICO-DET [2]. Meanwhile, we gladly found that the Boolean version has improved the non-rare classes and the overall mAP by 0.22 and 0.2. So after we point out the deficiencies of current human-based rules, we propose some viable approaches to enlarge the improvement.

2 Related Work

Our work is in the field of Human-Object Interaction (HOI) where image-based, instance-based and body-part-based patterns are mainly used.

 ISBN 978-0-9934207-5-7

Human-object Interaction Most of the daily human activities involve HOI [2, 7] . Thanks to Deep Neural Networks (DNNs), many great improvements have been made in the detection of such events [2, 5, 12]. Chao *et al.* [2] combined visual features and spatial locations to construct a multi-stream model. Qi *et al.* [12] proposed Graph Parsing Neural Network (GPNN) incorporating DNN and graphical model to iteratively update states and classify pairs. Gao *et al.* [5] developed an instance centric attention module to increase the information from the region of interest and improve the HOI classification. Li *et al.* [9] explored interactiveness knowledge learned from various HOI datasets, implicitly increasing the training data for rare HOI classes. While these works have contributed to the improvement of HOI detection, the progress of the few-shot HOI classes is insufficient [8].

Part States Lu *et al.* [11] proposed a discrete set of part states through tokenizing the semantic space and bases a sort of basic descriptors on segmentation [4]. Furthermore, Li *et al.* [8] utilized states of 10 human body natural parts to represent activities and reasons out the activities with *part-level sementics*. *PaSta*Net makes great improvements and reaches the-state-of-the-art in both full and few-shot tasks. Despite that, the mAP of one-shot set in the *PaSta*Net is still below 0.3. To improve that, in this paper, we mainly focus on few-shot problems in HOI.

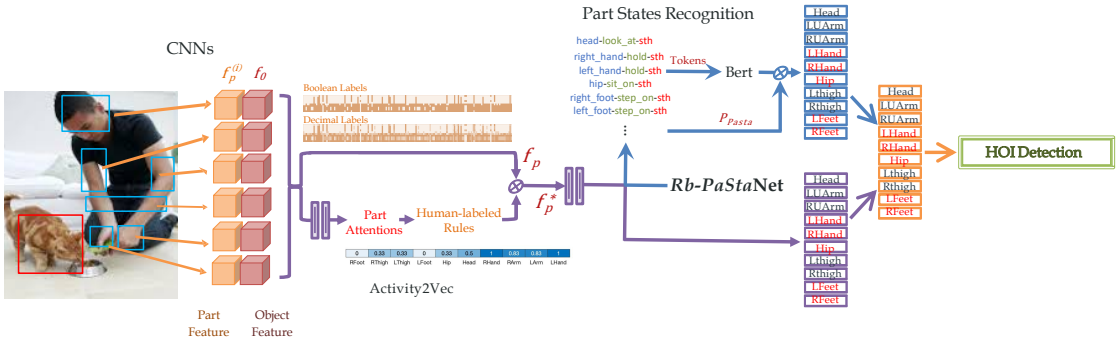


Figure 1: Overview of ***Rb-PaSta*Net**

3 Approach

In this section we introduce the construction of **(*Rb-PaSta*Net)** in Figure1 to tackle the few-shot problems. Why current DNN-based models do not perform well in few-shot HOI classes may lie in insufficient information offered by training datasets. Therefore, our method introduces human prior knowledge to reinforce learning in rare HOI classes. Considering *PaSta*Net [8] using body parts' action as a medium to infer HOI, adding prior rules into each body parts action weights can be viable. In the following paragraphs, the choice of the dataset, the method to label and the process of rules construction are specified.

Data We conduct model training based on the *PaSta*Net database, with more than 200k training examples. The existing network divides object-action into 600 classes(including the no-interaction ones). A total of 162 classes only appear in the training set less than ten times. Our experiments are targeted at optimizing existing networks in terms of these rare classes.

Label As for the rare classes, three authors perform manual annotations respectively based on the body parts' involvement in the corresponding object-action according to our prior knowledge. In our annotation, label 1 indicates a strong correlation; 0.5 indicates a certain degree of correlation; 0 indicates irrelevance. As for common classes, all parts are labelled 1. Then we average three annotations as one type of label. The average value is distributed between 0 and 1. Besides, to find a better rule, we map the resulting Decimal label to the Boolean label (True if the decimal label is no less than 0.5, or False otherwise). The two groups of labels and the All-True control group (original label) were trained separately. Table1 shows an example of labelled weights in an HOI "feed a cat" and all weights are displayed in Figure2 where the upper one represents the Boolean version and the lower one represents the Decimal version.

 ISBN 978-0-9934207-5-7

Implementation In *PaSta*Net [8], all features will be initially input to a **Part Relevance Predictor** telling a body part's importance in an action. Formally, a certain attention is

$$a_i = \mathscr{P}_{pa}(f_p^{(i)}, f_o) \tag{1}$$

where $\mathscr{P}_{pa}(\cdot)$ is the part attention predictor and $f_p^{(i)}$, f_o indicate features of a part and an interacted object respectively. In ***Rb-PaSta*Net**, we introduce rules :

$$a_i^{Rb} = a_i \cdot a_{rules} \tag{2}$$

where a_i^{Rb} represents attentions added with rules and a_{rules} indicates weights we have labeled in the last paragraph. Then we compute scores and cross-entropy loss like *PaSta*Net with a_i^{Rb} instead of a_i.

	Body Parts										
Method	RFoot	RThigh	LThigh	LFoot	Hip	Head	RHand	RArm	LArm	LHand	Average
Original	1	1	1	1	1	1	1	1	1	1	1
Decimal	0	0.33	0.33	0	0.33	0.5	1	0.83	0.83	1	0.52
Bool	0	0	0	0	0	1	1	1	1	1	0.5

Table 1: The weights of "feed a cat"

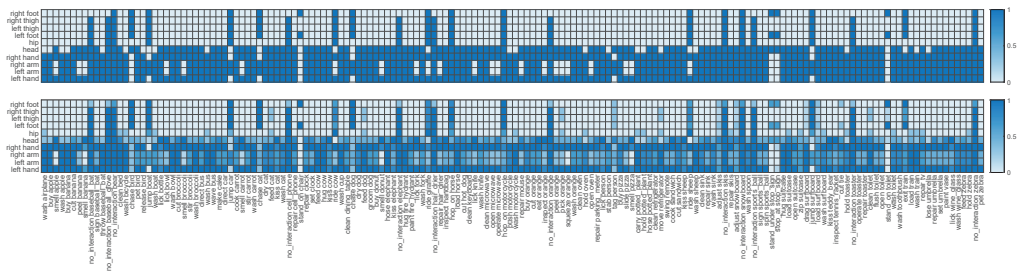


Figure 2: the Boolean and Decimal label matrix

4 Experiment

Settings We adopt one HOI datasets HICO-DET [2] with 600 HOI categories on 80 objects categories and 117 verbs. We first use best pre-trained Activity2Vec with *instance-level Pasta* labels [8] and then fine-tune ***Rb-PaSta*Net** on HICO-DET. All testing data are separated from pre-training and fine-tuning. We follow the metrics of [2] for results and PaSta detection. The fine-tuning takes 2M iterations and the learning rate is 1e-3 with 1:4 of positive and negative samples. A late fusion strategy is adopted.

Method	Full(def)	Rare(def)	Non-rare(def)	Full(ko)	Rare(ko)	Non-rare(ko)
PaStaNet	21.92	20.44	22.37	23.86	22.31	24.33
Rb-PaStaNet(Boolean)	22.12	20.54	22.59	24.04	22.43	24.52
Rb-PaStaNet(Decimal)	21.94	20.57	22.35	23.86	22.46	24.28

Table 2: Results on HICO-DET. We follow the evaluation metrics in [2]: def means *Default* setting where the full test set is detected, while ko means *Known Object* setting where the target object category is given.

Results As Table2 shows, Boolean and Decimal versions of ***Rb-PaSta*Net** achieve 0.1 and 0.13 mAP improvements on rare HOI classes. Although the human labels are not very precise concerning they are based on few people's intuition and comprehension, the result has proved that by applying human prior knowledge the mAP can be improved. Meanwhile, the result shows that the Boolean version has improved the non-rare classes and the overall mAP by 0.22 and 0.2. After analysing the scores of each classes, we find out that the rules are also influencing those non-rare classes(just think those classes share the same label-[1,1,1,1,1,1,1,1,1,1]). So we have proposed a few possible approaches that may increase the competitiveness of ***Rb-PaSta*Net**:

 ISBN 978-0-9934207-5-7

- label all the 600 HOI classes more comprehensively and rigorously
- label few-shot pictures instead of considering the rare classes as a whole

5 Conclusion

In this article, based on the existing *PaSta*Net, we propose ***Rb-PaSta*Net** instead. Our goal is to improve on rare classes by adjusting the training weights of different parts. The experiment result has proved our method is feasible, but it also leaves much room for improvement. The probable reason is that our rule itself is not accurate enough because it is generated by three authors. In the future, we hope to get a more accurate and detailed version with the help of volunteers. We believe a better rule can make more improvement.

Acknowledgments

We would like to express our very great appreciation to Cewu Lu and Yong-Lu Li for their constructive suggestions and guidance throughout this research work. We would also like to extend our thanks to Liang Xu for his assistance in terms of codes. Their patience and carefulness have been very much appreciated.

References

[1] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robot. Auton. Syst.*, 57(5):469483, May 2009.

[2] Y. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng. Learning to detect human-object interactions. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 381–389, 2018.

[3] Vincent Delaitre, Josef Sivic, and Ivan Laptev. Learning person-object interactions for action recognition in still images. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1503–1511. Curran Associates, Inc., 2011.

[4] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting, 2019.

[5] Chen Gao, Yuliang Zou, and Jia-Bin Huang. ican: Instance-centric attention network for human-object interaction detection. *CoRR*, abs/1808.10437, 2018.

[6] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015.

[7] Yong-Lu Li, Xinpeng Liu, Han Lu, Shiyi Wang, Junqi Liu, Jiefeng Li, and Cewu Lu. Detailed 2d-3d joint representation for human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10166–10175, 2020.

[8] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Shiyi Wang, Hao-Shu Fang, Ze Ma, Mingyang Chen, and Cewu Lu. Pastanet: Toward human activity knowledge engine. In *CVPR*, 2020.

[9] Yong-Lu Li, Siyuan Zhou, Xijie Huang, Liang Xu, Ze Ma, Hao-Shu Fang, Yanfeng Wang, and Cewu Lu. Transferable interactiveness knowledge for human-object interaction detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3585–3594, 2019.

[10] Cewu Lu, Ranjay Krishna, Michael S. Bernstein, and Fei-Fei Li. Visual relationship detection with language priors. *CoRR*, abs/1608.00187, 2016.

[11] Cewu Lu, Hao Su, Yonglu Li, Yongyi Lu, Li Yi, Chi-Keung Tang, and Leonidas J Guibas. Beyond holistic object recognition: Enriching image understanding with part states. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6955–6963, 2018.

[12] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *Computer Vision – ECCV 2018*, pages 407–423, Cham, 2018. Springer International Publishing.

Oropharynx Detection in PET-CT for Tumor Segmentation

Vincent Andrearczyk[1], Valentin Oreiller[1,2], and Adrien Depeursinge[1,2]

[1]*Institute of Information Systems, University of Applied Sciences Western Switzerland (HES-SO), Sierre, Switzerland*
[2]*Centre Hospitalier Universitaire Vaudois (CHUV), Lausanne, Switzerland*

Abstract

We propose an automatic detection of the oropharyngeal area in PET-CT images. This detection can be used to preprocess images for efficient segmentation of Head and Neck (H&N) tumors in the cropped regions by a Convolutional Neural Network (CNN) for treatment planning and large-scale radiomics studies (e.g. prognosis prediction). The developed method is based on simple image processing steps to segment the brain on the PET image and retrieve a fixed size bounding box of the extended oropharyngeal region. We evaluate the results by measuring whether the primary Gross Tumor Volume (GTV) is fully contained in the bounding box. 194 out of 201 regions (96.5%) are correctly detected. The code is available on our GitHub repository.[1].

Keywords: Medical imaging, oropharynx, detection, preprocessing

1 Introduction

Head and Neck (H&N) cancers are among the most common cancers worldwide (5^{th} leading cancer by incidence) [Parkin et al., 2005]. In particular, the oropharynx is located in the middle part of the throat (pharynx) and is a common site for the development of H&N tumors.

In [Andrearczyk et al., 2020], we developed an automatic H&N primary tumor and lymph nodes segmentation from Positron Emission Tomography-Computed Tomography (PET-CT) images using a V-Net Convolutional Neural Network (CNN). To analyze the oropharyngeal region, a bounding box was used to crop the input volume to input to the CNN. This bounding box was defined by centering a fixed size volume around a minimal bounding box containing the tumor. While this approach allowed us to evaluate the CNN performance, an automatic region detection is necessary in clinical practice to obtain a fully automatic pipeline. To this end, we propose a simple method based on morphological operations in the PET image to automatically locate the extended oropharyngeal region with a bounding box. To evaluate the approach in Section 3, we consider a detection to be correct if the primary tumor is fully contained within the bounding box.

2 Methods

In this section, we describe the automatic detection of a $144 \times 144 \times 144$[2] bounding box representing the extended oropharyngeal region.

The region detection is based on brain detection in the PET volume. We first apply a Gaussian filter with a standard deviation $\sigma = 3$ to remove potential noise and high frequencies that could impair the following

[1] `github.com/voreille/hecktor`, as of August 2020.

[2] This size is in mm^3. It is used because it covers the oropharyngeal region and is a typical input size for standards 3D CNNs that will be used with these data. Naturally, the results presented in Section 3 depend on this size since a larger bounding box would more likely contain the tumor.

 ISBN 978-0-9934207-5-7

threshold. A threshold is then applied to the Standardized Uptake Values (SUV) in the PET image. A fixed threshold value of three is used as it allows a good separation of the brain and the rest of the head. We then simply find the brain as the largest connected component. Note that we compute this detection only in the top third (z-axis) of the PET volume because high SUV values can be found in the bladder and in the injection point for some cases.

Once the brain is segmented, we define the extended oropharyngeal region as a bounding box with a fixed physical size ($144 \times 144 \times 144$) using predefined shifts from the brain volume. In the following description, we consider the axes in the patient reference; i.e. x goes from right to left, y from anterior to posterior and z from inferior to superior. On the z-axis, we find the lowest voxel of the brain and set the center of the box three centimeters below. On the x-axis, the center is set as the midpoint between the most-left and most-right brain voxels. Finally, on the y-axis, the center is set as the midpoint between the most-anterior and most-posterior brain voxels, shifted by three centimeters forward. The shift values are manually set based on a rough visual approximation of the head dimensions and (distances between brain and throat, mouth, tumor) and tilt. Although head sizes and poses vary, the algorithm is robust to these variations and to variations in the shift values since the bounding boxes are larger than the oropharyngeal region. Alternatively, these shift values could be computed from a training set if ground truth annotations of the oropharyngeal regions were available. The code for this automatic detection is available on our github repository[3].

3 Results and Discussions

We evaluate the proposed method on the 201 training cases used in the HEad and neCK TumOR (HECKTOR) tumor segmentation challenge at MICCAI 2020[4]. Examples of bounding boxes overlayed on top of 2D slices of the original volumes are illustrated in Fig. 1 together with the brain segmentations. A total of 194 out of 201 regions (96.5%) are correctly detected by our method, i.e. the primary tumors are fully contained within the bounding boxes. The failure cases, i.e. primary tumor not fully contained in the bounding box or entirely missed, are illustrated in Fig. 2 and detailed in the following. Four misdetections are due to abnormal SUV values in the PET image (e.g Fig. 2.a). These can be due to incorrect information such as patient weight or time between injection and acquisition. To prepare the challenge data, we simply correct the threshold value for these cases to detect correct bounding boxes. For the training of segmentation algorithms after our region detection, the SUVs can be standardized as in [Mortazi et al., 2020] to account for these possibly incorrect SUV values. The other misdetections are due to the brain being outside the image for one (see Fig. 2.b) and a high tumor on the z-axis together with the head tilted (e.g. Fig. 2.c) for the other two.

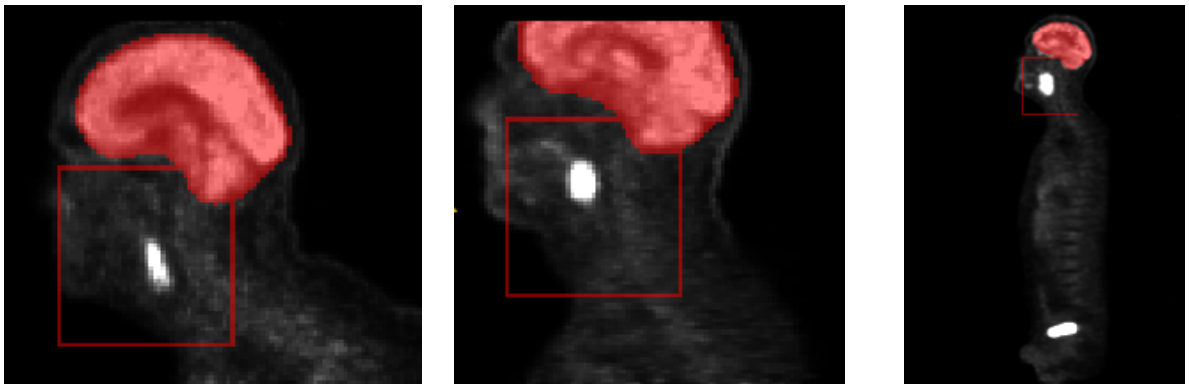
4 Conclusions and Future Work

In this short paper, we showed that simple image processing techniques can be used to accurately detect the extended oropharyngeal region containing primary H&N tumors. The automatic detection was successful for 96.5% of the cases, while the failure instances were due to extreme cases (see Fig. 2) and were easily corrected. Another approach could be considered by registering the CT volumes to a reference volume with known anatomical regions. However, it is a difficult task due to the variation in body parts that are covered by the different scans (e.g. full body, head only or cropped brain).

This method was used to pre-define bounding boxes for an automatic tumor segmentation task in the development of the HECKTOR challenge at MICCAI 2020 which will allow participants to evaluate and compare their segmentation algorithms in PET-CT images. In turn, it will provide a fully automatic pipeline including region detection and tumor segmentation with high potential in clinical practice as well as radiomics studies [Zwanenburg et al., 2020].

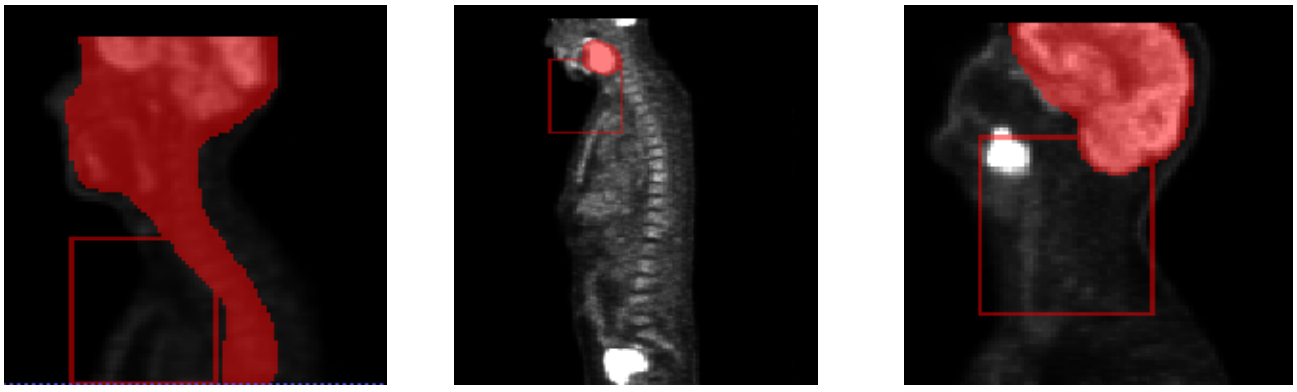
[3] github.com/voreille/hecktor, as of August 2020.

[4] `https://www.aicrowd.com/challenges/hecktor`, as of July 2020.



(a) (b) (c)

Figure 1: Examples of 2D slices of automatically generated bounding boxes of the extended oropharyngeal region. We illustrate the bounding box and the brain segmentation overlayed on top of the PET image. The primary gross tumor volume is characterized by a high PET activation.



(a) (b) (c)

Figure 2: Examples of 2D slices of failed extended oropharyngeal region detection. (a) The SUVs are low due to erroneous information in the DICOM metadata resulting in an incorrect segmentation of the brain, thus of the oropharynx, (b) The brain is outside the image resulting in a wrong segmentation and (c) The head is tilted and and the tumor is high on the z-axis, resulting in a bounding box too low.

 ISBN 978-0-9934207-5-7

Acknowledgments

This work was supported by the Swiss National Science Foundation (SNSF, grant 205320_179069) and the Swiss Personalized Health Network (SPHN via the IMAGINE and QA4IQI projects).

References

[Andrearczyk et al., 2020] Andrearczyk, V., Oreiller, V., Vallières, M., Castelli, J., Elhalawani, H., Boughdad, S., Jreige, M., Prior, J. O., and Depeursinge, A. (2020). Automatic segmentation of head and neck tumors and nodal metastases in PET-CT scans. In *Medical Imaging with Deep Learning (MIDL 2020)*.

[Mortazi et al., 2020] Mortazi, A., Udupa, J. K., Tong, Y., and Torigian, D. A. (2020). A post-acquisition standardization method for positron emission tomography images. In *Medical Imaging 2020: Computer-Aided Diagnosis*, volume 11314, page 113143U. International Society for Optics and Photonics.

[Parkin et al., 2005] Parkin, D. M., Bray, F., Ferlay, J., and Pisani, P. (2005). Global cancer statistics, 2002. *CA: a cancer journal for clinicians*, 55(2):74–108.

[Zwanenburg et al., 2020] Zwanenburg, A., Vallières, M., Abdalah, M. A., Aerts, H. J., Andrearczyk, V., Apte, A., Ashrafinia, S., Bakas, S., Beukinga, R. J., Boellaard, R., et al. (2020). The image biomarker standardization initiative: standardized quantitative radiomics for high throughput image-based phenotyping. *Radiology*.

An Ensemble-based Approach to the Detection of COVID-19 Induced Pneumonia using X-Ray Imagery

Chand Sheikh, Farshad Ghassemi Toosi, and Ruairí O'Reilly

Cork Institute of Technology

July 7, 2020

Keywords: Image classification, ensemble learning, transfer learning, COVID-19

Abstract

The rapid emergence and spread of COVID-19 resulted in a surge in demand for laboratory-based testing globally. Currently, the gold standard diagnostic approach is large-scale molecular testing of biological samples which detect the SARS-CoV-2 virus RNA. Infrastructure limitations and supply shortages are limiting testing capacity with a growing demand for COVID-19 diagnostics across the EU.

X-ray imagery is essential in establishing the severity of a multitude of diseases and monitoring responses of patients in the hospital setting. X-ray imagery should not be used to screen for or as a first-line test to detect COVID-19. However, X-ray presents an ideal opportunity to integrate additional screening measures into a pre-existing workflow. This paper investigates the utilisation of machine learning in automating the detection of COVID-19 induced pneumonia from X-Ray imagery.

The approach will assist radiologists in the monitoring and differentiation of pneumonia caused by COVID-19 from other viral causes. A classification for the presence, or absence, of pneumonia caused by COVID-19 and other viral causes is derived. The paper contributes an initial investigation into an ensemble-learning based approach using transfer learning models *VGG16*, *Inception* and *ResNet*. The results of this work indicate an improved performance using ensemble-based learning as compared to an individual transfer learning model.

1 Introduction

Pneumonia is a respiratory illness that occurs due to inflammation of tissues in one or both lungs caused by a bacterial or viral infection. In late 2019 an unknown cause for pneumonia was diagnosed in Wuhan and later it was found to be a new virus – severe acute respiratory syndrome coronavirus 2, or Sars-CoV-2. The strain of coronavirus that causes COVID-19.

Chest X-ray images are one of the most commonly used techniques to diagnose pneumonia from normal cases [Ticinesi et al., 2016]. However, the diagnosis of pneumonia from X-ray imagery is challenging as characteristic features within the image can be vague and these characteristics overlap with other diseases adding to the difficulty faced by radiologists in arriving at a diagnosis. A potential solution to address this concern is to seek a second opinion from another radiologist. While a diagnosis from a second radiologist is a reliable source, limited, or a lack of, availability may introduce time delays to the process resulting in a sub-optimal patient outcome. Thus, a decision support system capable of providing an automated "second opinion" would be of benefit and contributes to the motivation for carrying out the proposed work.

Image classification as an automated computerised decision-maker has been long used to detect a disease from imagery [Ker et al., 2017]. Detecting malaria and detecting pneumonia are two examples of the usage of image classification in the field of biomedical imagery [Liang et al., 2016] [Ayan and Ünver, 2019]. Image classification is the sub-domain of machine learning that deals with classifying images based on some existing

 ISBN 978-0-9934207-5-7

visual features in the image. There are several different image classification techniques, each which is suitable for a particular task [Lu and Weng, 2007]. Image classification problems are typically classified as supervised learning problems with a prominent approach in recent years being *Convolutional Neural Networks* (CNN). CNNs are deep neural networks with a large number of hidden layers in the network. The main objective of any machine learning model is to generalise to some unseen data based on the data used for training. Therefore, the more data used in training a model, the more unseen data can be analysed accurately.

Transfer Learning (TL) is a neural network that has been trained on a large quantity of data, so the task of learning may be started from a more advanced starting point as opposed to a random initialisation. TL is a methodology that improves learning and addresses the greediness of neural networks for data to some extent [Torrey and Shavlik, 2010]. *Inception*, *Xception*, *ResNet* and *VGG* are examples of such networks. In general, TL improves the baseline performance and time to train (TTT) of the learning process [Sarkar et al., 2018]. Along with the TL's advantages, there are a few limitations with TL models. In TL, each network is trained on a specific dataset with a specific architecture. For instance, *VGG16* in [Simonyan and Zisserman, 2014] is trained on the *ImageNet* dataset, the specificity of the TL network can potentially limit the general use of these models on real-world problems.Ideally, the training data used for pre-training should be similar to the test data.

Ensemble techniques [Zhou, 2009] are an approach that enables the utilisation of multiple pre-trained models to improve performance in classification tasks. There are multiple works [Torrey and Shavlik, 2010] [Dietterich et al., 2002], [Brown, 2010] that have used ensemble techniques for the task of image classification. Each model is initially trained on a particular dataset which might not optimally generalise the model to unseen data. Therefore, to increase the confidence of the decision made by a model, ensemble techniques may be used to make an aggregated decision from a group of models instead of one single model. Ensemble learning uses a combination of different models' output to form a "committee of decision makers" [Brown, 2010].

In this work, an ensemble-based approach built on top of Transfer Learning models for differentiating the cause of pneumonia is explored. The ensemble learning technique used in this work is *Model averaging* [Brownlee, 2018] and it is a simple way to improve the performance of ML algorithms [Hinton et al., 2015]. Three distinct classes are considered: 1) Pneumonia COVID-19 caused 2) Pneumonia Viral (non-COVID) caused and 3) Normal. The Ensemble-based approach adopted is built on top of TL trained neural networks that take X-ray imagery as input and automatically classifies them. TL models such as *VGG16*, *Inception* and *ResNet* and a number of permutations of ensemble-based approaches, *VGG16 + Inception* or *Inception + Resnet*, are presented. The *F1*, *Recall*, *Precision* and *Accuracy* are reported for each model. The models are initially trained and validated on binary classification (i.e., Pneumonia vs Normal) which is a replication of AYAN's work [Ayan and Ünver, 2019]. Thereafter, the work is extended to incorporate multi-class classification (Pneumonia COVID, Pneumonia non-COVID and Normal).

2 Related Work

Machine learning has demonstrated notable successes in the diagnosis of respiratory-related diseases such as pneumonia [Ayan and Ünver, 2019, Cohen et al., 2020]. Several projects have advanced the tools and techniques for automated analysis in this area. The predominant approach used to diagnose respiratory diseases automatically are image-based (X-ray and CT). In [Ayan and Ünver, 2019] X-ray images were used to perform comparative analysis for the detection of pneumonia. The work used a CNN for a binary classification of **pneumonia** and **non-pneumonia** instances.

In [Pan and Yang, 2009] TL is used as a means for enabling generalised feature extraction to be transferred to alternate application domains, significantly increasing the performance of the learning by avoiding a number of prepossessing tasks. In [Ayan and Ünver, 2019] TL is used and a comparison between *Xception* and *VGG16* models is undertaken, the findings report an improved accuracy for *VGG16* against *Xception*, 0.87% vs 0.82%. The confusion matrices for the two models indicate a better recall on pneumonia (true positive) for *Xception* as compared to *VGG16*, 0.94% vs 0.89%. Additionally, *VGG16* requires a shorter TTT (17% less then *Xception*). The conclusion of the work indicates the specificity detection capability of each model, e.g., *VGG16* is stronger in detecting normal cases as opposed to *Xception*, which seems to be stronger in detecting pneumonia cases.

 ISBN 978-0-9934207-5-7

In [Xu, 2020], the use of TL was applied to the detection of COVID-19 using X-ray images. The TL model utilised is *VGG16*, and the dataset has four categories: 1) Healthy: 79 images, 2) Pneumonia (Viral): 79 images 3) Pneumonia (Bacterial): 79 images and 4) Pneumonia (COVID-19): 69 images. Several experiments were performed: The first was a binary classification to differentiate healthy lung X-rays from infected X-rays by $SARS-COV-2$ virus and an 81% validation accuracy was obtained over 20 epochs. The experiment was repeated with three classes: 1) Healthy, 2) Pneumonia (Bacterial cause) 3) Pneumonia (Virus cause), and the validation accuracy remained at approximately 80%. Finally, the experiment was repeated with four classes (as listed above), and the accuracy decreased to 60%. The rationale presented for the decrease in accuracy is the overlap between pneumonia caused by COVID-19 and pneumonia caused by alternate viral infections.

Although most research is based on TL, some works propose purpose built models for image classification. In [Mittal et al., 2020] a multi-layered capsule network known, as **CapsNet**, was developed to detect pneumonia using X-ray images. Different models using a combination of capsules and convolutions that outperformed the previously proposed models in the field are presented [Sabour et al., 2017].

In [Wang and Wong, 2020] a deep CNN called **COVID-Net** was developed to diagnose COVID-19 from X-ray images. Images, in this work, are divided into three classes: 1) Normal 2) Pneumonia from non-COVID causes 3) COVID-19. Along with this new CNN, an open-access benchmark dataset called **COVIDx** is introduced containing 13,975 X-ray images from 13,870 patients. The initial results indicate that **COVID-Net** has minimal false positives when detecting COVID-19 (third category). This is not surprising as the neural network is designed specifically for COVID-19 detection. In a recent work by Cohen [Cohen et al., 2020] the severity of pneumonia is predicted by making use of *DenseNet* [Huang et al., 2017] (Pneumonia detection) and linear regression to predict two scores (lung involvement and degree of opacity). This study used 94 X-ray images from a public COVID-19 image data collection [Cohen, 2020].

The closest work to that proposed is a recent project [Nishio et al., 2020] in which X-ray images were used for differentiation of pneumonia caused by COVID from pneumonia caused by non-COVID causes. *VGG16*, *ResNet*, *MobileNet*, *DenseNet* and *EfficientNet* are used as TL models while ensemble models were not exploited. *VGG16* produced the best accuracy amongst the models.

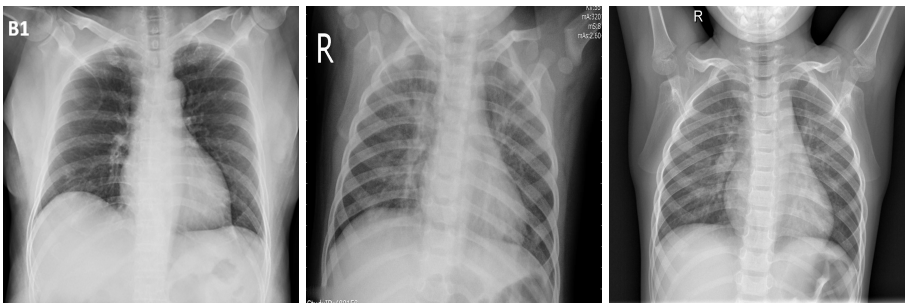
3 Methodology

In this work a quantitative approach to measure the effectiveness of TL models, and the effectiveness of ensemble learning, in classifying *Pneumonia non-COVID-19*, *Pneumonia COVID-19* and *non-Pneumonia* cases from X-ray imagery is carried out.

The data used for the experimental work comes from multiple sources. The *Pneumonia non-COVID-19* and *Normal* images are sourced from [Kermany et al., 2018] (this dataset was also used by [Ayan and Ünver, 2019]). Additional data containing *Pneumonia COVID-19* images was sourced from two resources, the *COVID-19 RADIOGRAPHY DATABASE* [Chowdhury et al., 2020] and the *Public Open Dataset* [Cohen, 2020]. Figures 1a, 1c and 1b depict three X-ray samples from [Chowdhury et al., 2020]. The data was pre-processed in order to be prepared for the classification task. Images were converted to 224×224 pixels to have a standardised image size and the class weight technique in *Keras* is used to address the unequal distribution of classes within the dataset. The distribution of of class instances for both experiments is denoted in Table 1a and Table 1b. 10-fold cross validation is used in training the TL models for both binary classification and multi-class classification. A *Hold-Out* test set was withheld from cross-fold to enable validation of ensemble based approaches on unseen data.

In this work, *VGG16*, *Inception* and *ResNet* as three TL models are used for image classification. Most of these models have been trained on a huge quantity of data and include a large set of classes. As there are multiple pre-trained TL models available, the first concern is the selection of the pre-trained models. The issue is explored in a number of works [Torrey and Shavlik, 2010], [Ayan and Ünver, 2019], [Chollet, 2017] and the *Inception* model typically performs better when compared to *VGG16* or *ResNet*. Although *Inception* performs slightly better, the rationale for this slight improvement is not satisfactory on its own as models are pre-trained on a different dataset from that used in the work proposed. A clear example is the work by [Nishio et al., 2020]

 ISBN 978-0-9934207-5-7



(a) **COVID-19** (b) **Viral-Pneumonia** (c) **Normal**

Figure 1: X-ray images sample

	Normal	**Pneumonia**
Training	1349	3883
Validation	234	390
Test	234	390
Total	1817	4663

(a) Binary classification.

	Normal	**Pneumonia non-COVID**	**Pneumonia COVID**
Training	1349	3883	236
Validation	234	390	58
Test	234	390	58
Total	1817	4663	352

(b) Multi-class classification.

Table 1: Data distribution of Imagery.

in which *VGG16* is the best performing amongst five different TL models. In order to address this concern, the application of Ensemble learning is proposed.

Two experiments are proposed: The first experiment, as a proof of concept, is a binary classification, Pneumonia non-COVID vs Normal and the second experiment is a multi class classification, Pneumonia non-COVID vs Pneumonia COVID vs Normal. For both experiments the TL models, and the permutations of TL models for the ensemble-based approaches, denoted in Table 2 are utilised. For each model, a set of metrics (**F1 Score**, **Precision**, **Recall**, **Accuracy** and **Loss**) are measured for cell to cell comparisons among different models. All experimental work uses the same set of data augmentation techniques: 1) Horizontal flip 2) Random crop 3) Gaussian blur 4) Linear contrast 5) Scale 6) Zoom. Table 4 and 5 denote the hyperparamters for each of the TL models.

Transfer Learning models (Base models)		
VGG16	Inception	ResNet
Ensemble models		
VGG16+Inception	VGG16+ResNet	Inception + ResNet
Inception + ResNet + VGG16s		

Table 2: Transfer Learning and Ensemble models used in both experiments (Binary and Multi-class classification.

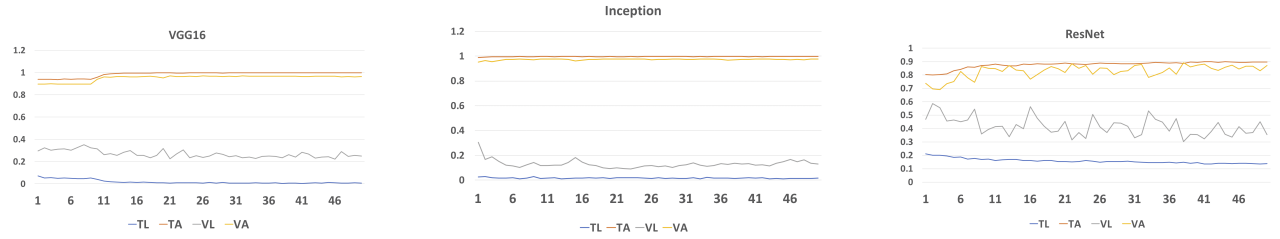
3.1 Binary Classification

To empirically validate our approach and act as a proof of concept, a re-implementation of the work in [Ayan and Ünver, 2019] is performed. The main difference between [Ayan and Ünver, 2019] work and this experiment is the use of *Inception* instead of *Xception*. The models used in both experiments are listed in Table 2. A number of common metrics such as recall, precision and F1-measure were tested and the individual result for each class (Normal vs Pneumonia) is presented in Table 4.

 ISBN 978-0-9934207-5-7

Hyper-parameter	VGG16	Inception (V3)	ResNet50
Activation	relu	relu	relu
Dropout	Yes (0.3 & 0.2) (FC)	Yes (0.3) (FC)	No
Activation (Final)	Softmax	Softmax	Softmax
Layers	25	316	3
Optimiser	RMSprop	RMSprop	RMSprop
Optimiser LR/Decay	0.0001/1e-6	0.0001/1e-6	0.0001/1e-6
Batch size	16	16	16
Batch normalisation	Yes (1st & 2nd FC[1])	Yes (2nd FC)	Yes (On FC)

Table 3: Hyper-parameter settings for each of the models. Values are mostly inspired by related works [Ayan and Ünver, 2019]. Since InceptionV3 uses multiple filters in each layer, the total number of layers in Inception would be larger than other models.

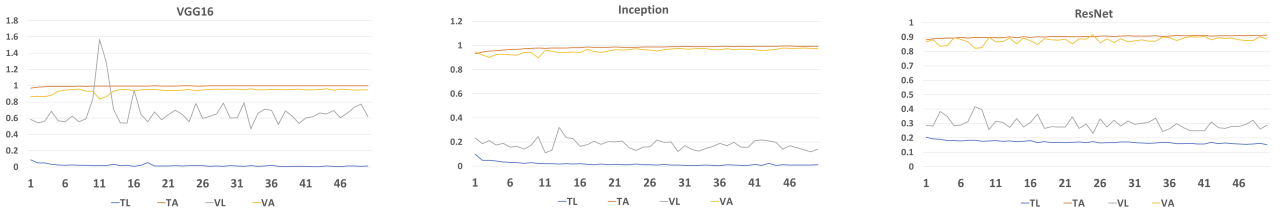


(a) **VGG16** (b) **Inception** (c) **ResNet**

Figure 2: Learning curves and training accuracy for the three base models for binary classification over 50 epochs. VL: Validation Loss, VA: Validation Accuracy, TL: Training Loss, TA: Training Accuracy

3.2 Multi-class Classification

We further extended the first experiment to try the aforementioned models (see Table 2) for multi-class classification where the third class is Pneumonia caused by COVID-19. Note that the second class in both experiments is images of confirmed Pneumonia caused by non-COVID-19 causes. In this experiment all the base models were tested using cross-validation and multiple metrics (Recall, Precision, Accuracy and F1-measure) are calculated and reported in Table 5. Since there are three classes, therefore, the said metrics are reported individually for all classes. The loss value for both train and validation sets along with accuracy for all models are also visualised and depicted in Figures 3a, 3b and 3c. To fully inspect the performance of models, different permutation of ensemble models are tested in order to make a fair conclusion at the end. The different ensemble combinations are listed in the lower half of the Table 5.



(a) **VGG16** (b) **Inception** (c) **ResNet**

Figure 3: Learning curves and training accuracy for the three base models for multiclass classification over 50 epochs.

 ISBN 978-0-9934207-5-7

Model	Accuracy	R-N	P-N	F-N	R-P	P-P	F-P
Base Models							
VGG16	0.827	0.455	1	0.62	0.972	0.826	0.89
Inception	0.867	0.663	0.977	0.788	0.99	0.831	0.90
ResNet	0.836	0.636	0.896	0.74	0.955	0.816	0.88
Ensemble Models							
ResNet + VGG16 + Inception	0.876	0.59	1	0.75	1	0.91	0.84
ResNet + VGG16	0.873	0.67	0.99	0.80	0.99	0.83	0.91
Inception + VGG16	0.889	0.59	1	0.75	1	0.85	0.92
Inception + ResNet	0.88	0.54	1	0.70	0.99	0.84	0.91

Table 4: First Experiment: Results for each of the models as evaluated against the "hold-out" (Test) Data. **R-N** Stands for Recall-Normal, **P-N** stands for Precision-Normal, **F-N** stands for F1 Score-Normal, **R-P** Stands for Recall-Pneumonia, **P-P** stands for Precision-Pneumonia, **F-P** stands for F1 Score-Pneumonia

Model	Accuracy	R-N	P-N	F-N	R-P	P-P	F-P	R-C	P-C	F-C
Base Models										
VGG16	0.83	0.594	0.916	0.718	0.989	0.805	0.888	0.694	0.822	0.75
Inception	0.886	0.52	0.992	0.68	0.973	0.925	0.948	0.768	0.888	0.783
ResNet	0.818	0.61	0.875	0.713	0.96	0.813	0.88	0.601	0.868	0.692
Ensemble Models										
ResNet + VGG16 + Inception	0.891	0.96	0.59	0.73	0.86	0.99	0.92	1	0.78	0.87
ResNet + VGG16	0.86	0.92	0.62	0.74	0.83	0.99	0.90	0.98	0.72	0.83
Inception + VGG16	0.894	0.97	0.57	0.72	0.88	0.99	0.93	0.96	0.76	0.85
Inception + ResNet	0.925	0.99	0.56	0.72	0.92	0.99	0.95	1	0.81	0.90

Table 5: Second Experiment: Results for each of the models as evaluated against the "hold-out" (Test) Data. **R-N** Stands for Recall-Normal, **P-N** stands for Precision-Normal, **F-N** stands for F1 Score-Normal, **R-P** Stands for Recall-Pneumonia, **P-P** stands for Precision-Pneumonia, **F-P** stands for F1 Score-Pneumonia, **R-C** Stands for Recall-COVID, **P-C** stands for Precision-COVID, **F-C** stands for F1 Score-COVID

3.3 Results

In this work, two different experiments are carried out: a binary classification and a multi-class classification on a set of X-ray images using three base TL models and permutations of ensembles constructed from the base TL models.

The binary classification had 7 different runs, three of which are based on individual TL models (VGG16, Inception and ResNet) and four of which are the different combination of TL models using ensemble modeling. As denoted in Table 4, *Inception* has the highest accuracy at approximately 87% and the highest *F1* score for Pneumonia detection. This is also apparent from the results depicted in Figure 2b as the accuracy is high from the beginning as opposed to *VGG16*.

The second highest accuracy is *ResNet* with approximately 84% followed by *VGG16* with approximately 83% accuracy. It should be noted that the results in Table 4 are from testing the models on *Hold-Out* or *Test* data that has not be seen whilst training the model.

Table 4 denotes the results of the ensemble models for binary classification. The highest accuracy is when *Inception* and *VGG16* are ensembled (88.9%) followed by the ensemble of *Inception* and *ResNet* (88%). The highest accuracy ensemble including *Inception* does not come as much of a surprise as *Inception* has a high accuracy as an individual model. Another expected result is the ensemble of *ResNet* and *VGG16* having the lowest accuracy (87.3%). This can be explained based on their individual performances as base models being less than that of *Inception*. Note that in all experiments, k-fold cross validation was employed.

The multi-class classification, saw the dataset divided into three classes (Normal, Pneumonia COVID, Pneumonia non-COVID). The base models produce similar results to that of the first experiment. As denoted in

Table 5, *Inception* outperformed the other two models with regards to accuracy. The F1-measure for *Inception* model is higher than the other two models for Pneumonia non-COVID and Pneumonia COVID classes and it is lower than the other two classes for the Normal class.

In the second part of Table 5, as expected, the ensemble models that include *Inception* results in a higher accuracy and the ensemble model that excludes *Inception* results in the lowest accuracy (*ResNet + VGG16*). Unlike the first experiment where the ensemble of *Inception* and *VGG16* has the highest accuracy, in the second experiment, the ensemble of *Inception* and *ResNet* has the highest accuracy. The same pattern emerged for F1-measure in Pneumonia COVID and Pneumonia non-COVID classes where the ensembled of *Inception* and *ResNet* has the highest F1-measure.

4 Conclusions and Future Work

This paper draws from a variety of existing research work and projects in the field of machine learning for detecting the causes of Pneumonia: COVID-19 vs other viral causes, using X-ray images. Transfer Learning (TL) and Ensemble Learning are the two methodologies that are exploited here. Two different experiments were performed in this work 1) **Binary classification** (Pneumonia vs Normal) 2) **Multi class classification** (Pneumonia COVID-19, Pneumonia non-COVID-19, Normal). Three different TL models (*VGG16*, *ResNet* and *Inception*) are employed in both experiments, and ensembles made from permutations of these models are tested.

The *Inception* model in both binary and multi-class classification outperformed all the other models with the average accuracy 87% and 89% respectively. The strength of the *Inception* model is also evident in ensemble modelling in both experiments. The ensemble models that do not include *Inception* do not perform as well as those that include *Inception* in both experiments. In [Ayan and Ünver, 2019] work, *VGG16* outperforms *Xception* while in our work, *Inception* outperforms *VGG16* which implies that *Inception* would outperform *Xception*, however *Xception* was not used in this work.

The contribution of this work is the use of ensemble learning for every possible combination of TL models. The research can be extended further to use 1) Other deep learning models such as Capsnet [Sabour et al., 2017] 2) Hyperparameter tuning and increased quantities of data 3) The application of the approach to CT imagery all of which will be investigated as part of future work.

References

[Ayan and Ünver, 2019] Ayan, E. and Ünver, H. M. (2019). Diagnosis of Pneumonia from Chest X-Ray Images Using Deep Learning. In *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, pages 1–5. IEEE.

[Brown, 2010] Brown, G. (2010). Ensemble Learning. *Encyclopedia of Machine Learning*, 312.

[Brownlee, 2018] Brownlee, J. (2018). How to Develop an Ensemble of Deep Learning Models in Keras.

[Chollet, 2017] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

[Chowdhury et al., 2020] Chowdhury, M. E., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., Islam, K. R., Khan, M. S., Iqbal, A., Al-Emadi, N., et al. (2020). Can ai help in screening viral and covid-19 pneumonia? *arXiv preprint arXiv:2003.13145*.

[Cohen, 2020] Cohen, J. P. (2020). ieee8023/covid-chestxray-dataset. original-date: 2020-02-14T23:22:23Z.

[Cohen et al., 2020] Cohen, J. P., Dao, L., Morrison, P., Roth, K., Bengio, Y., Shen, B., Abbasi, A., Hoshmand-Kochi, M., Ghassemi, M., Li, H., et al. (2020). Predicting COVID-19 Pneumonia Severity on Chest X-ray with Deep Learning. *arXiv preprint arXiv:2005.11856*.

 ISBN 978-0-9934207-5-7

[Dietterich et al., 2002] Dietterich, T. G. et al. (2002). Ensemble Learning. *The handbook of brain theory and neural networks*, 2:110–125.

[Hinton et al., 2015] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

[Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

[Ker et al., 2017] Ker, J., Wang, L., Rao, J., and Lim, T. (2017). Deep Learning Applications in Medical Image Analysis. *Ieee Access*, 6:9375–9389.

[Kermany et al., 2018] Kermany, D., Zhang, K., and Goldbaum, M. (2018). Large Dataset of Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images. 3.

[Liang et al., 2016] Liang, Z., Powell, A., Ersoy, I., Poostchi, M., Silamut, K., Palaniappan, K., Guo, P., Hossain, M. A., Sameer, A., Maude, R. J., et al. (2016). CNN-based Image Analysis for Malaria Diagnosis. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 493–496. IEEE.

[Lu and Weng, 2007] Lu, D. and Weng, Q. (2007). A Survey of Image Classification Methods and Techniques for Improving Classification Performance. *International journal of Remote sensing*, 28(5):823–870.

[Mittal et al., 2020] Mittal, A., Kumar, D., Mittal, M., Saba, T., Abunadi, I., Rehman, A., and Roy, S. (2020). Detecting pneumonia using convolutions and dynamic capsule routing for chest x-ray images. *Sensors*, 20(4):1068.

[Nishio et al., 2020] Nishio, M., Noguchi, S., Matsuo, H., and Murakami, T. (2020). Automatic classification between COVID-19 pneumonia, non-COVID-19 pneumonia, and the healthy on chest X-ray image: combination of data augmentation methods in a small dataset. *arXiv:2006.00730 [cs, eess]*. arXiv: 2006.00730.

[Pan and Yang, 2009] Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

[Sabour et al., 2017] Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866.

[Sarkar et al., 2018] Sarkar, D., Bali, R., and Ghosh, T. (2018). *Hands-On Transfer Learning with Python*. Packt Publishing.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Vgg-16. *arXiv Prepr*.

[Ticinesi et al., 2016] Ticinesi, A., Lauretani, F., Nouvenne, A., Mori, G., Chiussi, G., Maggio, M., and Meschi, T. (2016). Lung ultrasound and chest x-ray for detecting pneumonia in an acute geriatric ward. *Medicine*, 95(27).

[Torrey and Shavlik, 2010] Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI Global.

[Wang and Wong, 2020] Wang, L. and Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images. *arXiv preprint arXiv:2003.09871*.

[Xu, 2020] Xu, A. Y. (2020). Detecting COVID-19 induced Pneumonia from Chest X-rays with Transfer Learning: An implementation in Tensorflow and Keras.

[Zhou, 2009] Zhou, Z.-H. (2009). Ensemble learning. *Encyclopedia of biometrics*, 1:270–273.

Multi-Person Full Body Pose Estimation

Haoyi Zhu, Cheng Jie, Shaofei Jiang

Shanghai Jiao Tong University, China
{zhuhaoyi, jiec_tech, jiangshaofei}@sjtu.edu.cn

Abstract

Multi-person pose estimation plays an important role in many fields. Although previous works have researched a lot on different parts of human pose estimation, full body pose estimation for multi-person still needs further research. Our work has developed an integrated model through knowledge distillation which can estimate full body poses. Trained based on the AlphaPose system and MSCOCO2017 dataset, our model achieves 51.5 mAP on the validation dataset annotated manually by ourselves. Related resources are available at `https://esflfei.github.io/esflfei.gethub.io/website.html`.

Keywords: Multi-Person Pose Estimation, Full Body Pose, Knowledge Distillation

1 Introduction

Multi-person pose estimation has become increasingly popular in computer vision field in recent years. It has many applications such as human-computer interaction, augmented reality, and sports analytics. It can also improve the performance of re-targeting, tracking, and action recognition.

Previous works on this topic mainly focus on pose estimation of human body or different parts of human, such as head pose estimation and hand pose estimation. However, little research has been conducted on full body pose estimation. OpenPose [1] is currently the only system that can estimate multi-person full body pose, which is bottom-up and have to use mutiple networks, and [5] develops a single network on full body pose estimation based on it, which applies PAF network architecture and multi-task learning to get body part candidates and uses bipartite graph matching to reach the final full body pose.

Our work develops an integrated model to directly estimate multi-person full body pose through a single network based on AlphaPose [2], the state-of-the-art multi-person body pose estimation system. The insight of this paper is to train a multi-person full body pose estimation model through knowledge distillation. Our inspiration is from the teacher-student model. The body keypoints groundtruth can be obtained from the annotation of MSCOCO2017 dataset [4]. Based on them, we can get the predicted keypoints of face, hand and foot. We treat the predicted keypoints as pseudo labels and put all of them together with the body keypoints to get the full body pose label. Thus, we can use it to train a model that can estimate multi-person full body pose.

We train our model on the AlphaPose system and the MSCOCO2017 train dataset. We then annotate full body keypoints on MSCOCO2017 validation dataset by ourselves, where our model reaches a result of 51.5 mAP, 10.0 higher than the latest OpenPose model. Our model performs pretty well on foot and body. When faces or hands are too small or ocludded, the detection accuracy of them will decrease.

2 Related Work

There are four parts of research related to our work, including hand pose estimation, face keypoint detection, foot keypoint detection, and body pose estimation.

Hand Pose Estimation Due to high cost and challenges in manual annotation of hand keypoint, there does not exist any large hand keypoint dataset. To overcome this problem, Simon et al. [6] generated a labeled hand keypoint dataset by developing multiview bootstrapping and trained a single view hand keypoint detector.

 ISBN 978-0-9934207-5-7

Face Keypoint Detection There are mainly two kinds of approaches to achieve face keypoint detection: regression-based methods and template fitting. Regression Methods rely on Convolutional Neural Networks and often use convolutional heatmap regression, while template fitting usually employ a series of regression functions to fit the original image by creating face templates.

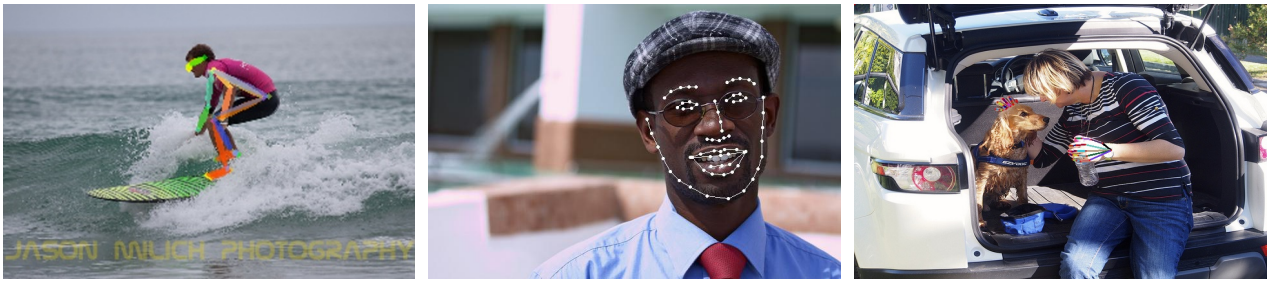
Foot Keypoint Detection Cao et al. [1] developed the first foot dataset based on the COCO dataset. The first detector combining body and foot keypoint was also trained.

Body Pose Estimation The early way to accomplish body pose estimation is to infer from both local observations and spatial dependencies of body parts. It is divided into two categories: tree-structured graphical-based models and non-tree models. With the development of CNN, the accuracy on body pose estimation grew rapidly and multi-person estimation became possible, mainly containing two ways: top-down and bottom-up.

3 Approach

Our aim is to acquire a pseudo label of 133 full body pose keypoints through knowledge distillation, including 17 body keypoints, 6 foot keypoints, 68 face keypoints and 42 hand keypoints (21 per hand), and use them as groundtruth when training. In this paper, we get the pseudo label based on the MSCOCO2017 train dataset. Since the 17 body keypoints have already been labeled in the dataset, we actually only have to obtain the rest and merge all of them together.

3.1 Data Annotation



(a) (b) (c)

Figure 1: The visualization results of our data annotation. (a)(b)(c) are examples of the body and foot keypoints, face keypoints and hand keypoints respectively.

Foot Keypoints Using the existing model based on AlphaPose which detects 17 body keypoints and 6 extended foot keypoints, we can directly obtain the foot keypoints and the visualized connection.

Face Keypoints Based on the regular architecture of PRNet, we first predict the position of face bounding box using the annotated body parts of nose, eyes and ears. The face bounding box is then cropped and fed into PRNet where the input 2D image is mapped to a corresponding colored UV texture map so that we can predict UV parameters by CNN to do face reconstruction [3]. Finally, we get the 68 detected face keypoints.

Hand Keypoints In this section, we extract the hand detection block in OpenPose independently for the output of 21 keypoints per hand. We use annotated body parts to predict the hand box proposals. The proposals then go through a hand estimation model based on Multiview Bootstrapped Training [6], which generates geometrically corresponding hand keypoints annotations under an external supervision source of multiple views and uses these annotations to further improve the detector. Finally, the detected hand keypoints are added to the end of keypoints list.

 ISBN 978-0-9934207-5-7

Figure 2: The final result of full body pose annotation.

Now a multi-person full body keypoints annotation containing body, face, hand and foot keypoints is constructed. The merged json can then be utilized as the ground truth for the subsequent training. It can be directly thrown into a single mature pose estimation network (AlphaPose in this paper) to train a full body pose estimation model.

3.2 System Framework

Our work is based on the AlphaPose system which follows the RMPE framework [2]. It is SPPE-based and follows the two-step framework which contains human detection and pose estimation. The whole framework mainly consists of three blocks: Symmetric Spatial Transformer Network (SSTN), Parametric Pose Non-Maximum-Suppression (NMS) and Pose-Guided Proposals Generator (PGPG). The SSTN attached to the SPPE is used for obtaining an accurate single person region from a rough bounding box. After detecting human proposals, SSTN transforms the proposals to make them more suitable for SPPE, and de-transforms them after SPPE. To improve this step, in the training process, there is an added parallel SPPE branch which back-propagates the center-located pose errors. A parametric pose NMS is then introduced to eliminate redundant pose estimations, which defines a novel pose distance metric with a data-driven optimizer. Finally, the PGPG is employed for data augmentation by learning the output distribution of human detection results.

3.3 Training

Our model is trained using seven Nvidia GeForce RTX 2080 Ti graphics cards. Our batch size is set to 8 and the Adam optimizer is used. The input resolution is 256×192 while the output heatmap resolution is set to 64×48. The initial learning rate is set to 0.001 with a learning factor of 0.1. Our model is trained and iterated through 328 epochs to improve performance.

4 Experiment

We have manually annotated the full body keypoints on validation dataset of MSCOCO2017 [4], which includes 5000 images, to evaluate our model. We choose YOLO as our human detector, due to its high efficiency and accuracy. We apply AlphaPose to train our model. Figure 3 shows the visualization result of our model. We then run the OpenPose system, using both the model_25 (the common model of OpenPose [1]) plus face and hand flag and model_135 (the model trained by [5]), and compare its results with ours, which is shown in **table 1**. We can see that our mAP reaches 51.5, 10.0 higher than [5] and 13.3 higher than [1]. Among the three methods, our AP and AR under any condition are all the best.

Methods	**mAP**	AP^{50}	AP^{75}	AP^{M}	AP^{L}	**mAR**	AR^{50}	AR^{75}	AR^{M}	AR^{L}
OpenPose [1]	38.2	51.3	32.5	28.8	46.6	44.8	58.7	41.2	29.0	55.6
single [5]	41.5	69.4	29.1	38.8	36.4	49.4	73.1	42.2	40.1	48.6
ours	**51.5**	**74.0**	**46.5**	**45.8**	**46.9**	**59.7**	**77.1**	**57.4**	**53.2**	**56.7**

Table 1: Results of OpenPose and our method on the validation dataset.

We have found that the foot and body part perform the best. Even if the human instances are small or crowded, they still works well. The face part performs well most of time, but when faces are too small or partly hidden, the detection accuracy will decrease. The hand detection, however, performs the worst and is easy to fail or make mistakes when hands are too small or occluded.

Figure 3: Two of the visulization results of our work.

The failures mainly result from the pseudo label, because the PRNet itself perform not well on small and occluded faces and the accuracy of hand detection in Openpose is limited. In other words, there originally exists errors in the ground truth we used for training, which is certain to influence the final result. But anyway, our method has been proved to be correct and effective. In general, our contribution is that we have proposed a novel, better method of full body pose estimation, obtained the full body pseudo labels and trained a full body pose estimation model better than previous ones.

References

[1] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[2] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017.

[3] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 534–551, 2018.

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[5] Gines Hidalgo Martinez, Yaadhav Raaj, Haroon Idrees, Donglai Xiang, Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Single-network whole-body pose estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6981–6990. IEEE, 2019.

[6] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017.

Candidate Subspace Screening for Linear Subspace Clustering with Energy Minimization

Katsuya Hotta[1], Haoran Xie[2], and Chao Zhang[1]

[1]*University of Fukui, Japan*

[2]*Japan Advanced Institute of Science and Technology, Japan*

Abstract

In this paper, we propose a method of linear subspace clustering by screening the least frequently used candidate subspaces (LFUCS) under the framework of energy minimization. A common issue in such problem is that when data points consist of inliers distributing within an inlier range, the performance of clustering methods tends to degrade. To alleviate this issue, our clustering algorithm removes the LFUCS to improve the quality of candidate subspaces, as well as the clustering performance. Quantitative experiments on both synthetic and real-world data demonstrate that our method outperforms competitive methods.

Keywords: Linear subspace clustering, Candidate subspace screening, Energy minimization

1 Introduction

Many computer vision tasks (e.g., face segmentation and motion segmentation) require clustering high-dimensional data. In the context of linear subspace clustering, data points are supposed to lay near/on a union of low-dimensional subspaces. To solve the clustering problem, subspaces and the corresponding belonging points should be jointly estimated. In the past decades, various algorithms of linear subspace clustering have been proposed. Among them, many state-of-the-art methods belong to either of the nearest neighbor (NN) based methods [Park et al., 2014] or the energy-based methods [Zhang, 2019]. As one of the common issues, the presence of inlier ranges of subspaces can directly degrade the clustering performance, as shown in Figure 1. To alleviate this issue, in the case of NN based methods, it is necessary to construct a neighborhood relationship that considers the inlier range, while in the case of energy based methods, a promising solution is to improve the quality of subspace proposals (candidates). In this paper, we propose a method under the framework of energy minimization by removing the least frequently used candidate subspaces (LFUCS) during the labeling procedure. Specifically, our method iteratively repeats the following two steps: (a) conduct labeling with respect to the current candidate subspaces by energy minimization, and (b) remove LFUCS and update the candidate subspaces, until the termination criterion is met. Our idea is based on a natural assumption that LFUCS has a low probability of being the underlying correct subspace.

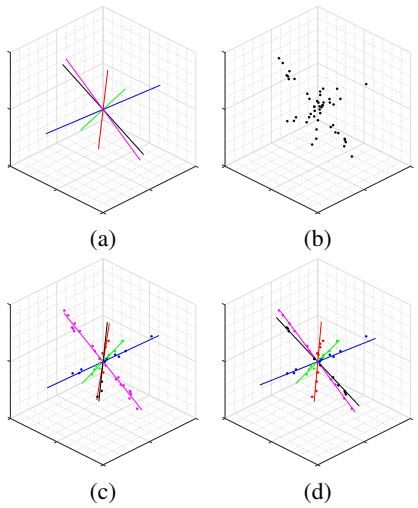


Figure 1: An example of clustering a union of linear subspaces. (a) Randomly generated ground truth subspaces. (b) Input data points are randomly generated from (a) within a inlier range (i.e., points do not lay exactly on the underlying lines). (c) A typical failure result of conventional methods. (d) Result of our method.

 ISBN 978-0-9934207-5-7

Algorithm 1 Overview of Proposed Algorithm

Input: Data point set $\mathcal{Y} = \{y_1, \ldots, y_N\}$, Adjacency matrix $W = \{0,1\}^{N \times N}$, Number of subspaces L, Subspace dimension d
Output: Estimated subspaces $\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_L\}$, Labeling results $\omega = \{\omega_1, \ldots, \omega_N\}$

1: $y_i = y_i / \|y_i\|_2, \forall i \in [N]$ ▷ Normalize magnitudes
2: $\mathcal{U}^* \leftarrow$ Estimate N candidate subspaces from the neighborhood relationship W of each point using standard PCA [Vidal, 2011]
3: $L^* = N$
4: **repeat**
5: $\omega \leftarrow$ Assigning labels to $y_i \in \mathcal{Y}$ by α-expansion ▷ Labels are given by the indices of L^* subspaces
6: L^* is updated to the number of labels appear in ω, $\mathcal{U}^*$ is updated to the subspaces (labels) appear in ω
7: **if** $L < L^*$ **then** ▷ Removing least frequently used candidate subspaces (LFUCS)
8: $k \leftarrow \operatorname{argmin}_{j \in [L^*]} \sum_{t=1}^{N} \mathbb{1}(\omega_t, j)$
9: $\mathcal{U}^* \leftarrow \mathcal{U}^* \setminus \mathcal{U}^*_k$
10: **end if**
11: $\mathcal{U}^* \leftarrow$ Update candidate subspaces from the labeling result ω using standard PCA [Vidal, 2011]
12: **until** the number of candidate subspaces L^* reaches L
13: **return** ω, $\mathcal{U} \leftarrow \mathcal{U}^*$

2 Our Approach

2.1 Problem Setting

We first define some notations to facilitate the explanation of our algorithm. A set of N data points in $\mathbb{R}^p$ is defined by $\mathcal{Y} = \{y_1, \ldots, y_N\}$. Each data point y_i is lying on/near L subspaces of d dimension which is smaller than p. Each data point is supposed to be assigned a label via the algorithm to indicate its nearest subspace, with its labeling result denoted by w_i. A set of N labeling results of the data points is defined by $\omega = \{\omega_1, \ldots, \omega_N\}$. A set of L subspaces is defined by $\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_L\}$. L is the number of subspaces, which is given by the user. L^* denotes the number of candidate subspaces. Labels are defined by the indices of L^* estimated subspaces. Besides, for simplicity, we denote the the set of N indices by $[N] = \{1, 2, \ldots, N\}$ and $\mathbb{1}(\cdot, \cdot)$ is an indicator function that returns true (i.e., 1) if the left input equals to the right input, and false (i.e., 0) otherwise.

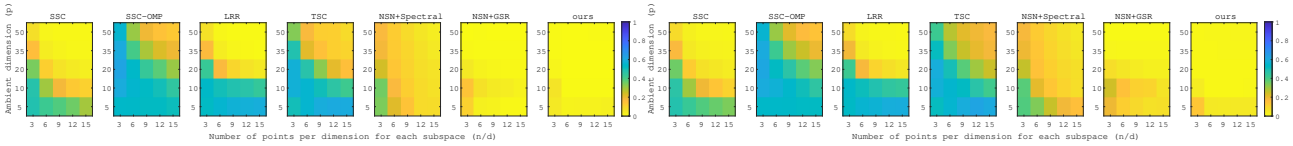
2.2 Algorithm

Our algorithm improves the performance of linear subspace clustering by removing LFUCS under the framework of energy minimization. An overview of our algorithm is presented in Algorithm 1. Specifically, as input, a data point set $\mathcal{Y}$ and an adjacency matrix W with knowledge about the neighborhood of each data point are given. W is used to initialize the candidate subspaces (line 2) and can be estimated using existing methods such as [Park et al., 2014, Liu et al., 2012]. In the following steps, by alternately updating labeling results (line 5) and removing LFUCS (line 7-10), the number of candidate subspaces L^* gradually approaches L, thus the subspaces are refined (line 11). α-expansion [Boykov et al., 2001] is used for labeling by minimizing the following energy function,

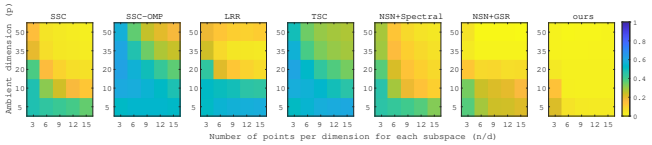
$$E(\omega) = \alpha \sum_{i \in [N]} \|y_i - \mathcal{U}_{\omega_i}(\mathcal{U}_{\omega_i}^\top y_i)\|_2 + \beta \sum_{(i,j) \in \mathcal{E}} \mathbb{1}(\omega_i, \omega_j), \tag{1}$$

where α and β are the coefficients for balancing between the data term and the smooth term. $\mathcal{U}_{\omega_i}$ is ω_i-th candidate subspace, and ω_i is the label assigned to y_i. $\mathcal{E}$ is the set of the edges connecting two data points in α-expansion, which is defined by the k-nearest neighbor graph [Von Luxburg, 2007]. The data term represents the geometric error over the data points with respect to candidate subspaces. The smooth term represents the smoothness of the graph with respect to data points and $\mathcal{E}$. Our algorithm is inspired by the observation that data points often do not exactly lay on the subspace. On the other hand, it is difficult to select candidate subspaces in the form of energy minimization alone because the quality of candidate subspaces can hardly be quantified. To produce better labeling results, the selection of candidate subspaces is performed by iteratively removing the least frequently used $\mathcal{U}^*_j$ (i.e., least number of data points are assigned to $\mathcal{U}^*$). The idea behind LFUCS is

 ISBN 978-0-9934207-5-7



(a) $\sigma = 0.001$ (b) $\sigma = 0.003$



(c) $\sigma = 0.01$

Figure 2: The results of average CE over 100 random trials with five d-dimensional subspaces and n random points per subspace with inlier range controlled by σ. d/p is fixed to 3/5. Lower CE values indicate that more data points are clustered correctly.

Table 1: Comparison of CE and computational time on face clustering.

	K-means	K-flats	SSC	LRR	SCC	SSC-OMP	TSC	NSN-Spectral	ours
Mean CE (%)	45.59	37.91	3.17	43.19	16.59	6.47	11.52	1.31	1.27
Median CE (%)	48.05	40.23	0.00	47.66	9.77	1.56	1.56	0.78	0.78
Avg. Time (sec)	0.08	2.74	8.68	0.96	41.68	0.07	0.08	0.22	0.70

natural and straightforward: more data points are supposed to lay on/near a "good" subspace. In other words, it is possible to bring the underlying subspaces to light by iteratively examining the number of data points that lay on/near candidate subspaces.

3 Experimental Results

In this section, we quantitatively evaluate the clustering performance of our method by comparing against competitive methods on both synthetic data and real-world data to show the usefulness of our method. The competitive methods are K-means, K-flats, SCC [Chen and Lerman, 2009], TSC [Heckel and Bölcskei, 2013], LRR [Liu et al., 2012], SSC [Elhamifar and Vidal, 2013], and SSC-OMP [Dyer et al., 2013], NSN+Spectral [Park et al., 2014], NSN+GSR [Park et al., 2014]. The number of replicates of K-means and K-flats is fixed to 10. Our method uses an adjacency matrix estimated by NSN. The performance is valued by the clustering error (CE) defined as,

$$CE = \min_{\pi \in \Pi_L} \frac{1}{N} \sum_{i=1}^{N} (1 - \mathbb{1}(\omega_i^*, \pi(\omega_i))), \tag{2}$$

where L is the number of subspaces, and Π_L is the permutation space of $[L]$. ω_i^* indicates the ground truth labeling result of y_i, and $\pi(\omega_i)$ is the labeling result with respect to permutation π. Since clustering is invariant to the permutation of label indices, the CE is equal to the minimum disagreement over Π_L. We first compare the performance on randomly generated synthetic data. In the ambient space, five ground truth subspaces are randomly generated, and n data points are generated within the inlier range. To confirm the statistical results, the dimension number of ambient space p and subspace d are variable with the constraint $d/p = 3/5$. Comparison on CE is shown in Figure 2. Our method clearly outperforms others. In particular, we can confirm that the clustering performance improves as the number of data points increases. We then conduct a comparison with the real-world task of face clustering. The data set is the extended Yale B data set [Georghiades et al., 2001] in which the frontal view faces of 38 subjects under different 64 illumination conditions are included. Table 1 summarizes the CE and computational time of two subjects selected randomly over 38 subjects for 50 trials. Figure 3 shows an visual result. We can see that our method outperforms competitive methods in mean CE.

 ISBN 978-0-9934207-5-7

Figure 3: An example of face clustering under three illumination conditions. Subject A and Subject B are represented by the red (middle) and blue (right) bounding boxes. A/B are labeling results of different methods.

4 Conclusion

In this paper, we improved the performance of linear subspace clustering under the framework of energy minimization by removing LFUCS. Our method generally outperforms competitive methods on both synthetic and real data. One limitation is that only a small inlier range can be tolerated at the current version. As future work, we would like to treat the inliers far from the corresponding subspace as outliers, which can be removed in advance.

References

[Boykov et al., 2001] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

[Chen and Lerman, 2009] Chen, G. and Lerman, G. (2009). Spectral curvature clustering. *International Journal of Computer Vision*, 81(3):317–330.

[Dyer et al., 2013] Dyer, E. L., Sankaranarayanan, A. C., and Baraniuk, R. G. (2013). Greedy feature selection for subspace clustering. *The Journal of Machine Learning Research*, 14(1):2487–2517.

[Elhamifar and Vidal, 2013] Elhamifar, E. and Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781.

[Georghiades et al., 2001] Georghiades, A. S., Belhumeur, P. N., and Kriegman, D. J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660.

[Heckel and Bölcskei, 2013] Heckel, R. and Bölcskei, H. (2013). Subspace clustering via thresholding and spectral clustering. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3263–3267. IEEE.

[Liu et al., 2012] Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., and Ma, Y. (2012). Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184.

[Park et al., 2014] Park, D., Caramanis, C., and Sanghavi, S. (2014). Greedy subspace clustering. In *Advances in Neural Information Processing Systems*, pages 2753–2761.

[Vidal, 2011] Vidal, R. (2011). Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68.

[Von Luxburg, 2007] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

[Zhang, 2019] Zhang, C. (2019). Energy minimization over m-branched enumeration for generalized linear subspace clustering. *IEICE TRANSACTIONS on Information and Systems*, 102(12):2485–2492.

Efficient Visual Place Retrieval System Using Google Street View

Reem Aljuaidi & Rozenn Dahyot

School of Computer Science and Statistics
Trinity College Dublin, Ireland
aljuaidr@tcd.ie, Rozenn.Dahyot@tcd.ie

August 14, 2020

Abstract

Visual place retrieval in a large geo-tagged image dataset at street level aims at giving a query image (with no GPS) a geo-tag using only its visual information. We apply aggregated binary descriptor (Oriented FAST and Rotated BRIEF (ORB)) with Mini Batch Vector of Locally Aggregated descriptor for feature extraction and aggregation phases. Our experimental results measured with search accuracy (mAP) and search time (s) show that our ORB-MBVLAD is significantly faster with good search accuracy (mAP) compared to the other state-of-the-art methods.

Keywords: Image retrieval, Feature extraction, Geo-tagging

1 Introduction

Large-scale image retrieval is an important task in computer vision. It helps many computer vision and robotics applications, e.g., object detection, visual place recognition, and navigation. In image retrieval systems, hand-crafted features and indexing algorithms have been used [3, 7]. More recently, methods based on convolutional neural networks (CNNs) for global descriptor learning have been shown to improve retrieval results [2, 8]. In this paper, we aim to make the retrieval system more efficient for retrieving the correct visual place from a large database. We apply ORB [9] with Mini Batch VLAD [1] instead of non-binary local features (SIFT [6]) for extraction feature. We also test Mini Batch VLAD [1] using ORB [9] and SIFT [6] for Google street view datasets and Oxford 5k datasets [7]. The result shows that using this aggregated binary descriptor is efficient for both speed and mAP.

2 Visual Place Retrieval

Traditionally, the visual place recognition problem has been cast as an image retrieval task. The location of the query image is estimated using the locations of the most similar images obtained by querying large geo-tagged database. In large datasets, the visual retrieval systems should efficiently retrieve similar images from these datasets. Each dataset image is usually represented using local invariant features such as SIFT [6] that are aggregated into a single vector representation for the entire image such as bag-of-visual-words [3], VLAD [4]. We first discuss image retrieval methods, before turning to a discussion of some visual place retrieval methods relate to our work.

 ISBN 978-0-9934207-5-7

2.1 Image Retrieval Systems and Related works

Image retrieval systems. Most of the established retrieval systems rely on local visual features e.g., SURF and SIFT coupled with approximate nearest neighbor search methods. They match local visual features descriptors between single images or look for similar descriptors. All descriptors need to be compared individually, which means that the methods are efficient only in small datasets. To reduce the cost of extracting and matching local visual descriptors, the researchers introduced the use of binary local descriptors such as (BRISK and ORB [9]). Binary descriptors are extracted much faster, and more compact than non-binary ones. To solve the problem of the quantization error in Bag-of-Words (BoW) representation, Vector of Locally Aggregated Descriptors (VLAD) [4] was introduced. VLAD approach computes the sum of the residuals between each visual word and the corresponding clustering center. However, VLAD is defined by non-binary features such as SIFT, the cost of extracting local descriptors and aggregating them is still high.

Related works. VLAD is one of the traditional methods widely used in place recognition and image retrieval [4]. The descriptor is a low-dimensional vector, and its goal is to serve as a featurization of the image as a whole. A visual vocabulary is built from a dataset, extracting D-dimensional descriptors from affine-invariant detections and clustered into k centers. In the task of querying an image with n local descriptors, the residual from each descriptor to each cluster center is calculated. Then, the residual is summed per each cluster and formed in k D-dimensional aggregate vectors. All vectors in the images are compared using the original Euclidean distance metric. Mini Batch-VLAD, a modification of the VLAD, was formalized by Aljuaidi et al. [1] to improve visual place retrieval. The key idea of MB-VLAD [1] is to aggregate features using a vector with a fixed size, and to learn the vocabulary word using a mini batch k-means clustering algorithm.[10]. In this way, computational load is reduced and it is still convenient to use standard distance measures to retrieve relevant images. The dimensions are reduced via principal component analysis (PCA). Our paper relates to this work, however we use a different method for the extraction phase. We use binary local features ORB [9] with Mini Batch VLAD [1] instead of non-binary local features to overcome the problem of improving the performance of the retrieval system in large datasets. Some works rely on regional searches to improve place recognition and image retrieval performance. Kim et al. [5] use maximally stable external regions (MSER) with a bundled VLAD. However, the problem with this method is computational cost and a lengthy time frame to compare local features within large databases. It needs to calculate VLAD vectors in each image patch. More recently, features learned from deep neural networks have used for place recognition and retrieval [2]. In NetVLAD, a set of local descriptors of a single image is learned by a convolutional neural network, and the compact form of the local descriptors is computed similarly to VLAD [4]. However, this method takes a long time during the feature-extraction phase which is not suitable for all real-time applications.

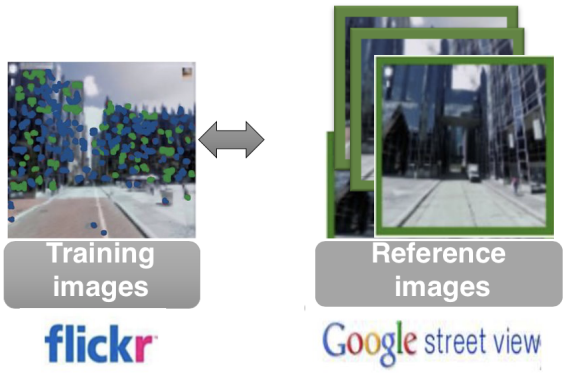


Figure 1: Our training dataset. For each training images that have GPS-tag (Flickr platform 918 images), we retrieve top n images from the reference set (Google Street View includes 27,520 images).

3 Aggregated binary local descriptor ORB-MB-VLAD

Oriented FAST and Rotated BRIEF (ORB). E.Rublee et al. [9] introduced Oriented FAST and Rotated BRIEF (ORB) in 2011. ORB builds on the well-known FAST keypoint detector and the BRIEF descriptor. Both of these techniques are attractive because of their good performance and low cost. ORB algorithm starts by finding the position of the key points by FAST and then selecting N best points. After that, it adds the direction of the points in intensity centroids. Finally, binary descriptors are extracted by BRIEF and low correlative

pixel blocks are found by the greedy algorithm.

Mini Batch Vector of Locally Aggregated Descriptor (MB-VLAD) For the aggregated phase, we use the extension of VLAD [1]. Vector of Locally Aggregated Descriptor represents an image by a single fixed-size vector using K-means clustering [4]. Aljuaidi et al. [1] propose learning VLAD by using mini batch k-means clustering to retrieve a place from geotagged dataset. Mini Batch VLAD approach [1] builds a codebook dictionary $C = \{c_1, c_2, \ldots, c_k\}$ from m k feature vectors in the reference dataset. To generate the dictionary, a mini batch k-means clustering algorithm is used. For an image having m descriptors I $\{x_1, \cdots, x_m\}$, the VLAD coefficient V_i is computed by accumulation over these descriptors in cluster c_i The final VLAD representation is a concatenation and followed by L_2 normalization. Thereafter, VLAD encodes features by computing the residuals. Then, the residuals are stacked together as vector v. In Mini Batch VLAD [1], they used the set of centroids inferred by mini batch k-means algorithm with cm_i as the cluster centers. Clustering input data are given binary descriptors before adding them into mini batches. Two normalizations are applied to compute the final VLAD when the dictionary is generated.

4 Experiments

Training and testing dataset. For each image in the training set, we retrieved 100 images from the reference set using image similarity metrics. For the reference image set, we use the reference Google Street View Dataset provided by [12] covering the Pittsburgh (U.S.) area. These images capture both street views and some parts of the city's taller buildings. The co-located GPS-tagged training image set was downloaded from Flickr platform. For each location in geo-location ground truth, we obtained up to 25 most relevant images. Figure 1 shows the training method. The test image set consisted of 145 internet collection images from [11], contains various street-level images and buildings in Pittsburgh (U.S.) area.
Results. We applied ORB with Mini Batch VLAD on image retrieval benchmark Oxford 5k dataset [7] and our trained dataset. We compare the results with the state-of-the-art methods VLAD [4], PBVLAD [5], NetVLAD [2] and MBVLAD [1]. The results are shown in table 1. Note that we repeat our experiment 6- times because we used k-means clustering which has the unsupervised nature of clustering. The average values of precision are applied. The performance evaluated by the mean Average Precision (mAP). The precision defines the relevant images retrieved numbers in response to a query image (number of relevant images retrieved / total number of images retrieved). For query images, we select images from the test dataset randomly. A visual vocabulary of 16 words and 128 vocabulary sizes is applied for all methods in both datasets. All methods are evaluated without dimensionality reduction. For searching time (s), the average response time (s) (retrieval time per query) in each method was calculated. The results in Table 1 show the proposed aggregated binary descriptor method achieved the fastest retrieval performance. The average retrieval time of our ORB-MBVLAD on the Oxford 5k [7] and Google Street View image datasets are 0.076 and 0.521 seconds, respectively. NetVLAD [2] achieves the best search accuracy for both datasets, however it is the slowest method time compared with others. Our proposed method came second in terms of search accuracy with mAP 0.49 on the Oxford 5k dataset.

5 Conclusion

This paper addresses the problem of efficient visual place retrieval using street view images. To extract features, we apply binary features with an aggregated descriptor. The result shows that proposed descriptor has a significant improvement in performance in terms of search time (s) with no longer effect of the search accuracy.

Acknowledgments

The first author is supported by Prince Sattam bin Abdulaziz University Scholarship funded by Saudi Arabian Government.

Descriptor	mAP	time(s)
VLAD [4]	0.27	0.944
PB-VLAD[5]	0.32	1.384
NetVLAD[2]	**0.58**	1.171
MB-VLAD [1]	0.41	0.722
Ours	0.49	**0.521**

Descriptor	mAP	time(s)
VLAD[4]	0.33	0.364
PB-VLAD[5]	0.36	1.173
NetVLAD[2]	**0.51**	1.255
MB-VLAD[1]	0.47	0.414
Ours	0.44	**0.076**

Table 1: Right: Retrieval performance of different methods using street view dataset on Pittsburgh city. Left: Retrieval performance of different methods using Oxford 5k dataset [7]. Best results with bold.

References

[1] Reem Aljuaidi, Jing Su, and Rozenn Dahyot. Mini-batch vlad for visual place retrieval. In *2019 30th Irish Signals and Systems Conference (ISSC)*, pages 1–6, 2019.

[2] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[3] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, page 304–317, Berlin, Heidelberg, 2008. Springer-Verlag.

[4] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR 2010 - 23rd IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311, San Francisco, United States, June 2010. IEEE Computer Society.

[5] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Predicting good features for image geo-localization using per-bundle vlad. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1170–1178, Washington, DC, USA, 2015. IEEE Computer Society.

[6] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[7] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*. IEEE Computer Society, 2007.

[8] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. *ECCV*, pages 3–20, 2016.

[9] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society.

[10] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1177–1178, New York, NY, USA, 2010. ACM.

[11] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 255–268, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[12] Amir Roshan Zamir and Mubarak Shah. Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1546–1558, 2014.

Utilising Domain Transformations in Multi-Camera Re-Identification Scenarios beyond Data Augmentation

Glen Brown, Jesus Martinez del Rincon, and Paul Miller

Centre for Secure Information Technologies, Queen's University Belfast

August 2020

Abstract

GANs and CycleGANs, such as CamStyle, have proved to be very effective when augmenting multi-camera datasets in re-identification scenarios. They achieve this by transforming the training images into the domain of each camera. However, this learned domain adaptation is not exploited at the re-identification stage. In this work we propose an extension to CamStyle where the domain transformation is not only used in training for data augmentation, but also further integrated in testing for improving the re-identification performance when different cameras in the scenario are in distinct domains.

Keywords: Facial Re-Identification; Domain Transformation; Data Augmentation; Deep Learning; GAN; CycleGAN

1 Methodology

1.1 CycleGANs for Data Augmentation

CamStyle[1] is a data augmentation technique used when training re-identification models on multi-camera scenarios. CamStyle can be broken down into three high-level stages:

1. Train CycleGANs[2] on every pair of cameras in the multi-camera scenario. As each pair of cameras have their own CycleGAN G, so an image from any camera can be transformed into the domain of another camera. This results in $\binom{C}{2}$ or $\frac{1}{2}(C-1)C$ trained CycleGANs where C is the number of cameras in the scenario.

2. Augment the dataset. Using the CycleGANs from Stage 1, every image from every camera in the original dataset is transformed into every other camera. This multiplies the size of the dataset D by the number of cameras in it, resulting in an augmented dataset of size $D' = D \times C$. The resulting dataset consists of $\frac{D'}{C}$ real samples and $\frac{C-1}{C}D'$ generated or "fake" samples.

3. Use the augmented dataset to train a Re-identification model. The original CamStyle[1] uses the training strategy laid out in [3]. This effectively trains the network in a classification scenario, where during training the inputs are the sample images and the output is Softmax probability distribution over the identities in the dataset. The only difference at this stage between the real and fake data is that the fake samples generated in Stage 2 have label smoothing regularization (LSR) [4] applied to their ground-truth identity distribution. After the model M has been trained on the classification scenario, the final layers of the model are removed, allowing feature vectors to be retrieved as output. The model is then evaluated in a typical re-identification scenario: matching images in a query set to images in a gallery set by producing feature vectors and ranking the corresponding Euclidean distances.

 ISBN 978-0-9934207-5-7

1.2 Extending CamStyle to the Re-identification Phase

Note that in Section 1.1, the trained CycleGANs are not used in any way or form during the actual re-identification stage despite having learned directly relevant information to perform domain adaptation across cameras/domains for a given scenario, and therefore improving the likelihood to better match identities in reidentification. Instead, CamStyle is only used to augment the training data.

We decided to therefore extend the original CamStyle[1] protocol to also perform domain transformations during the re-identification stage. The motivation for this is straightforward: if the gallery and query images are from different domains, then transforming them into the same domain before extracting feature vectors may lead to increased re-identification performance.

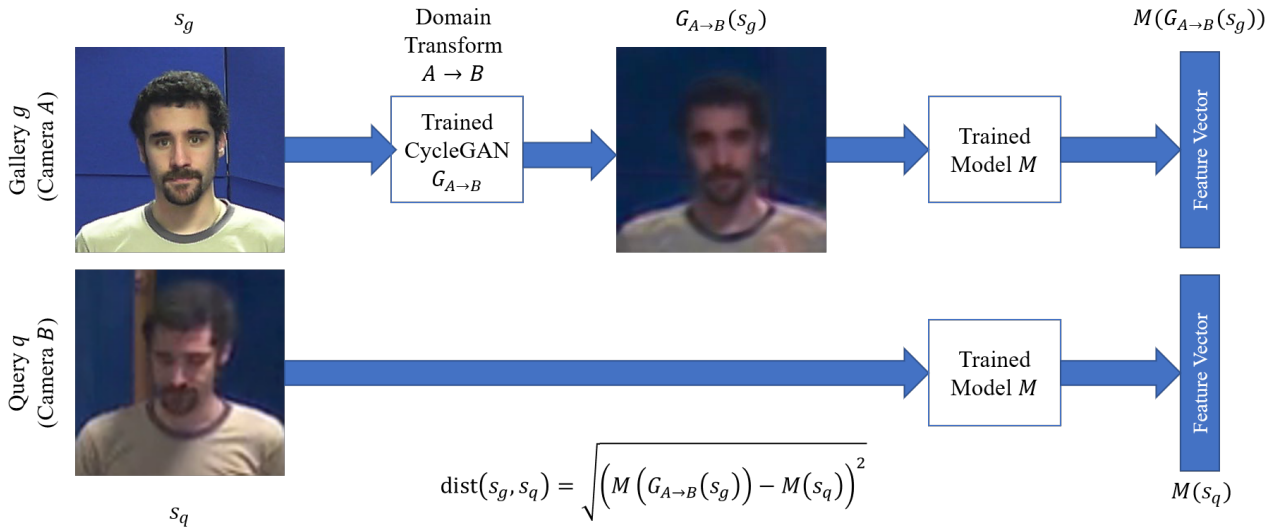


Figure 1: Gallery to Query Domain Transformation during Re-Identification. s is a sample, G is a trained generator from a CycleGAN, and M is the trained re-identification model.

Figure 1 shows how this extended CamStyle protocol works when transforming the gallery-image into the query-image's domain. This domain adaptation can also be performed in both directions, resulting in three possible domain adaptation protocols at testing time: *Gallery to Query* (eq. 2), *Query to Gallery* (eq. 1), and *Both* (eq. 3). *Both* entails calculating the distances between query and gallery sample using both *Query to Gallery* and *Gallery to Query*, and adding the resulting distances together.

$$\text{dist}(s_g, s_q) = \sqrt{(M(G_{A\rightarrow B}(s_g)) - M(s_q))^2} \quad (1) \qquad \text{dist}(s_q, s_g) = \sqrt{(M(G_{B\rightarrow A}(s_q)) - M(s_g))^2} \quad (2)$$

$$\text{dist}_{both}(s_q, s_g) = \text{dist}(s_g, s_q) + \text{dist}(s_q, s_g) \quad (3)$$

2 Experimental Design and Results

2.1 Experimental Setting

We kept the same domain adaptation models and training strategy as used in the original CamStyle[1], which is also equal to the original CycleGAN[2]. The generator of which contains 9 Residual Blocks[5], while the discriminator is a 70×70 PatchGAN[6]. As in [1], the Adam[7] optimizer, trained for 50 epochs with a batch size of 1 is used. The learning rate for the Generator is 0.0002 and 0.0001 for the Discriminator, which linearly decreases to 0 after epoch 30.

The re-id model also matches the original CamStyle[1] protocol. The backbone of the model consists of a ResNet50[5] model with a final Softmax layer appended to it. The number of units in the final

Softmax layer is equal to the number of identities in the training set the model is being trained on. During evaluation, this Softmax Layer is removed and feature vectors are output instead.

During training we also utilised Horizontal Random Flip and Dropout (both set to 50%). The model is trained using the Stochastic Gradient Descent optimizer for 50 epochs. The learning rate is initialised at 0.1 and decreases to 0.01 at epoch 40.

One difference in the model is that the original protocol normalises the input images by an ad-hoc pre-computed constant. As we are using datasets with very different properties (facial vs full-body, colour vs greyscale, etc) we instead used Batch Normalisation[8].

Due to the datasets consisting of different image sizes, we varied the size of the loaded images and the batch size per dataset to maximise computational efficiency. *MMF*[9] and *Market-1501*[10] were trained with a batch size of 256, with image sizes of $256 \times 256 \times 3$ and $256 \times 128 \times 3$ respectively. Both *Chokepoint*[11] and *COXFaceDB*[12] used a batch size of 512, with an image size of $128 \times 128 \times 3$.

2.2 Datasets

In order to prove the generality of our approach, 4 different reidentification datasets are used in this paper, each comprising different properties and applications.

MMF[9] is a facial dataset comprised of 77 subjects and 6 cameras: 3 high-quality cameras (A, B, C) and three low-quality cameras (1, 2, 3). During evaluation, images from the high quality cameras (A, B, C) were selected for use as the gallery with images from the low-quality cameras (1, 2, 3) being used as the query set. As this dataset was originally produced by extracting frames from video, many images are similar so we also limited the dataset to 10 randomly selected images per identity per camera.

COXFaceDB[12] comprises facial images with 1000 identities, with low-quality video footage taken with 3 cameras (amounting to 412415 images), and a single high-quality frontal still-image for each identity. We followed the V2S testing protocol defined in [12], with the exception that we only used a single fold and not all 10 due to time and computational capacity constraints. The high-quality frontal still images were chosen for use in the gallery, with random frames selected from the video footage used in the query set.

Market-1501[10] is a full-body person re-identification dataset. It consists of 6 cameras, all sharing the same general features and quality. There are a total of 1501 identities split into train (750 identities) and test (751 identities) sets. During testing, 3,368 images with hand-drawn bounding boxes are used as the query images, with the rest of the test set being used as the gallery (which also includes distractors).

Chokepoint[11] is another facial dataset, and consists of 2 subsets, one with 25 subjects and the other with 29 subjects. The dataset was constructed over a number of sessions, with each session having 3 cameras. Facial crops have been pre-extracted and saved in greyscale. We followed the general testing protocol stated in [11], using only the proscribed groups of images in $G1$ and $G2$: training on $G1$ and evaluating on $G2$.

2.3 Results

The results of our experiments are shown below in Table 1. In addition to the three transformations mentioned in Section 1.2, we also evaluated not using the CamStyle augmentation at all (No CamStyle), and not performing any transformation during the re-identification stage (CamStyle - No Transform).

 ISBN 978-0-9934207-5-7

Table 1: Extended CamStyle Protocol. All values are mean Average Precision (mAP)

Configuration	MMF[9]	COXFaceDB[12]	Market-1501[10]	Chokepoint[11]
No CamStyle	0.443	0.138	0.454	0.808
CamStyle - No Transform	0.678	**0.257**	0.560	0.930
CamStyle - Query to Gallery	0.622	0.210	0.517	0.927
CamStyle - Gallery to Query	**0.794**	0.159	0.549	**0.938**
CamStyle - Both	0.757	0.246	**0.563**	**0.938**

For all datasets, some implementation of CamStyle was better than not using it at all. For *Market-1501* and *Chokepoint*, utilizing domain transformations at the re-identification stage did give the best results by a small amount. This is likely due to all the cameras / domains in those datasets having similar properties, meaning it is fairly straightforward for the re-identification model itself to identify people across domains.

When the gallery and query sets have different properties though, such as in *MMF*, using the extended CamStyle protocol gave significant improvements. The caveat being that if a transformation is difficult (such as going from a low-quality to a high-quality domain), then care must be taken with which particular transformation is selected for use. This explains the better results of the *Gallery to Query* transformation than its counterpart. Due to the unbalanced nature of *COXFaceDB* (gallery-like images vs query-like images), the CycleGANs overfit during training so we cannot take advantage of the most useful transformations. While unable to transform individual images while sufficiently maintaining identity across domains, in aggregate the generated images still boost overall performance.

As a more general choice, using both the *Gallery to Query* and *Query to Gallery* transformations together seems to led to the most consistent results overall.

3 Conclusion

In this paper we proposed an extended CamStyle protocol, where the learned domain adaptation models learned for data augmentation are further exploited in the testing phase by transforming between domains prior to extracting reidentification feature vectors. Our approach has been evaluated in four distinctive reidentification datasets, and found competitive against the baseline. Notably, it leads to up to 12% improvement regarding the original protocol when domains are significantly different.

4 References

[1] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "CamStyle: A novel data augmentation method for person re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1176–1190, Mar. 2019.

[2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv:1703.10593 [cs]*, Mar. 2017.

[3] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv:1610.02984 [cs]*, Oct. 2016.

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 770–778.

[6] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," *arXiv:1604.04382 [cs]*, Apr. 2016.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980 [cs]*, Jan. 2017.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Feb. 2015.

[9] G. Brown, J. M. del Rincon, and P. Miller, "A comparative study of face re-identification systems under real-world conditions," in *Irish machine vision and image processing conference proceedings 2018: Proceedings*, 2018, pp. 137–144.

[10] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2015.

[11] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B. C. Lovell, "Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition," in *CVPR 2011 WORKSHOPS*, 2011, pp. 74–81.

[12] Z. Huang, S. Shan, R. Wang, H. Zhang, S. Lao, A. Kuerban, and X. Chen, "A benchmark and comparative study of video-based face recognition on COX face database," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5967–5981, Dec. 2015.

Automatic Recognition of Repetitive Hand Movements

Fiona Marshall, Shuai Zhang and Bryan Scotney

School of Computing, Ulster University

Abstract

Repetitive movements are associated with several neurological disorders including autism, dementia and Parkinson's disease. The automatic recognition of these movements can provide information that enables carers and clinicians to better monitor these conditions. Whilst a variety of approaches has been used to detect and measure repetitive body movements, the automatic recognition of repetitive hand movements has seen less attention. We introduce a novel video-based dataset of repetitive hand movements and propose a simple but effective method of recognition based on a small number of *discriminative poses*.

Keywords: Repetitive Hand Movements, Repetitive Behaviours, Activity Recognition, Skeletal Joints, Agitation

1 Introduction

Activity recognition is an active field of research. Despite this, the recognition of repetitive behaviours has not been widely explored. Applications include the detection and measurement of agitation in people with dementia, stereotypy in autism, and tremors in Parkinson's disease. The automatic recognition of repetitive movements can supplement existing care by providing accurate records, informing care choices, and preventing injury, whilst relieving pressure on carers and, through better understanding of the conditions, reliance on medication.

A range of sensors has been employed to detect repetitive movements, including accelerometers [Alam et al. 2018; Gilchrist et al. 2018], depth sensors [Chikhaoui et al., 2017] and video [Liu et al., 2019]. Video provides more detailed information than is available from accelerometers without the intrusion of wearing the sensor. To date, most video-based activity recognition of repetitive movements has focused on gross body movements whilst gesture recognition has centred around emblematic hand gestures for sign language interpretation and human-computer interaction. We combine methods for activity and gesture recognition to create a video-based approach that can recognise repetitive hand movements of different intensities that might indicate frustration, agitation, or anxiety.

This research explores the feasibility of using video to automatically classify repetitive hand movement. In addition, the effect of the movements being carried out at different speeds and by different subjects is examined.

2 Literature review

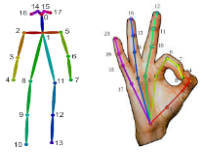


Figure 1: OpenPose extracts 18 body and 21 hand joints from video.

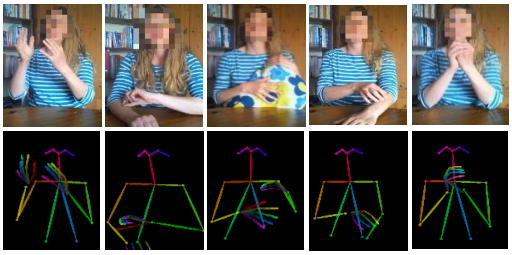
Video has been widely used for activity and gesture recognition; however, it usually requires lots of data and can be computationally expensive. Joint-based approaches can be more efficient [Herath et al., 2017]. Joint locations may be obtained from 3D sensors or video. Hand and body joints may be extracted from video using tools such as Openpose [Simon et al., 2017] as shown in Figure 1. This approach has been shown to be as effective for activity recognition as using specialist sensors [Marshall et al., 2019]. Additionally, video cameras are cheap, widely available, and can have a large field of view. Most research into the automatic detection of repetitive movement has focused on detecting agitation using statistical techniques [Alam et al., 2018]. However, recognition of the type of movement is vital for carers to better understand the person's needs. Dynamic Time Warping (DTW) [Rihawi et al., 2017] and ensemble tree approaches [Chikhaoui et al., 2017] have been used to recognise agitated body movements. Automatic measurement of the rate of repetitive movements has also been explored [Liu et al., 2019]. Recognising repetitive hand movement is challenging due to the number of joints and variety of possible movements of each hand.

 ISBN 978-0-9934207-5-7

3 Method

3.1 Data Collection

We are only aware of a small number of dynamic hand movement datasets. As these focus on gestures suitable for human-computer interaction rather than spontaneous or repetitive movements, we have created a novel dataset of five repetitive hand movements.



a b c d e

Figure 2: Five repetitive movements were recorded at 4 speeds for 30 seconds each: a) clap, b) pick, c) rub, d) scratch, e) wring hands. Joint position was extracted using Openpose.

Fifteen healthy participants recorded five hand movements with an RGB camera: *clapping*, *picking arm*, *scratching arm*, *hand wringing*, and *rubbing an object*, as shown in Figure 2. The dataset was recorded by participants in their own homes, creating a varied and demanding dataset representative of home or care settings. Variations included participant position (standing or seated), dominant hand and hand position. Figure 3 shows examples of inter-subject variation in *clapping* which included participants clapping at different heights, moving one or both hands, and extending arms different amounts between claps. Similarities between the types of hand movements and variations in the speed of movement add to the complexity of the dataset.

Each movement was performed four times for 6 seconds at speeds of 30, 50, 70 and 90 repetitions per minute, creating a dataset of 1200 sequences. Most videos were recorded at a speed of 30 frames per second; those that were not were up or down sampled to this rate; resulting in sequences of 180 frames. Three different testing protocols are used. To establish the overall effectiveness of our method, we employ 4-fold validation (Test1); the effect of different speeds of movement is investigated using 4-fold cross-speed validation (Test2); and to explore the effect of unseen subjects, 15-fold cross-subject validation is used (Test3). Test1 and Test2 are non-cross subject. The study was approved by the relevant Ulster University Faculty Research Ethics Filter Committee.

Figure 3: Inter-subject variation in clapping

3.2 Data Cleaning

Using Openpose, we represent each person as a stickman. Each frame is pre-processed following the approach used in previous research [Marshall et al., 2019] to create a scale-normalised skeletal pose invariant to participant size and distance from the camera. Each pose consists of 8 upper body and 42 hand joints, $P_f = \{j_{f,1}, \dots, j_{f,50}\}$, with joints centred around the neck where $j_{f,i} = (x_{f,i}, y_{f,i})$ are the co-ordinates of the i^{th} joint of frame f. The dominant hand is identified and, where necessary, poses are reflected so that the dominant hand is always the right hand. The dominant hand is defined as the hand with the greatest wrist movement, calculated by summing the maximum Euclidean distance between wrist locations over ½ second (15 frame) sliding windows for each wrist.

$$wrist\ \ movement = \sum_{w=1}^{166} \left(\max \left(\sqrt{\left(x_{a,wrist} - x_{b,wrist}\right)^2 + \left(y_{a,wrist} - y_{b,wrist}\right)^2}\ \forall\ a,b = w,..,w+14 \right) \right) \quad [1]$$

The purpose of using windows instead of individual frames to calculate wrist movement is to remove the effect of any hand tremors. Finally, the hand, wrist and elbow joints are re-centred around the centre of the wrists, resulting in the scale-normalised skeletal arm and hand pose $\hat{P}_f = \{\hat{j}_{f,1}, \dots, \hat{j}_{f,46}\}$ where

$$\hat{j}_{f,i} = \left(x_{f,i} - 0.5(x_{f,right\ wrist} + x_{f,left\ wrist}),\ y_{f,i} - 0.5(y_{f,right\ wrist} + y_{f,left\ wrist})\right) \quad [2]$$

3.3 Classification of Movement

The simplest approach to classify movement is to reduce each sequence to a single feature vector and apply a classifier such as a Support Vector Machine (SVM); however vital information may be lost in the data reduction process. Recurrent Neural Networks (RNN) networks are widely used for sequential classification but can require lots of training data. A less data intensive method is Naive Bayes Nearest Neighbours (NBNN) which measures the similarity between frames in sequences without considering temporal order. The NBNN algorithm [Yang and Tian, 2014]: Given an unknown sequence, S $=\{\hat{P}_1, \dots, \hat{P}_{180}\}$, each observation is matched with its nearest neighbour from

 ISBN 978-0-9934207-5-7

each class of the training data. The probability of the observation belonging to a class is estimated according to the sum of the distances between each unknown observation and its nearest neighbour for the whole sequence. The predicted class c^* is the class with minimum total distance between observations and nearest neighbours:

$$c^* = \arg min_c \sum_i^{180} \left\| \hat{P}_i - NN_c[\hat{P}_i] \right\| \quad [3]$$

where NN_c is the nearest neighbour of $\hat{P}_i$ in class c.

3.4 Selection of Discriminative Poses

Reducing a sequence to a single *mean pose* is a simple but effective way to classify sequences of repetitive movement. We define the *mean pose* as the mean position of each joint over the six-second (180 frame) sequence:

$$mean\ joint_i = \left(\frac{\sum_{t=1}^{180} x_{t,i}\ ,\ \sum_{t=1}^{180} y_{t,i}}{180} \right) \quad [4]$$

Experimental results of 15-fold cross-subject classification (Test3) of sequences represented by a single *mean pose* with SVM achieved an accuracy of 64.3%, suggesting that *mean poses* are an efficient and accurate way to represent repetitive movements. As reducing a sequence to a single pose removes all dynamic information, additional *discriminative poses* are included. The *maximum pose* is the pose where the centres of each hand are furthest apart; conversely the *minimum pose* is where hands are closest together. A sequence of repetitive hand movements can then be summarised by a *mean, maximum* and *minimum pose* as shown in Figure 4. To add rigour, each six-second sequence is divided into three 2 second (60 frames) sub-sequences from which *mean pose* and *maximum* and *minimum movement* are obtained; 60 frames are the smallest window that will always contain movement. Thus, each sequence is described by 9 *discriminative poses, D*. For NBNN classification we determine the class as:

$$c^* = \arg min_c \sum_i^9 \| D_i - NN_c[D_i] \| \quad [5]$$

When using *discriminative poses* with NBNN, nearest neighbours for *mean pose, maximum and minimum movement* are searched for from within their own type.

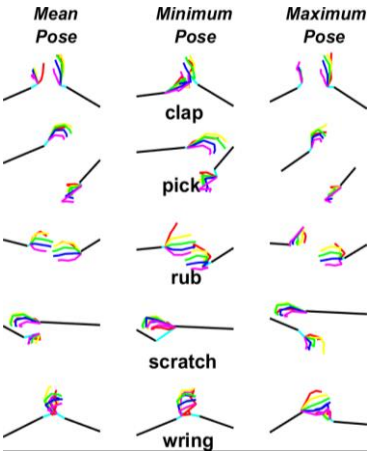


Figure 4: *Discriminative Poses*. Left to right: *mean pose, minimum pose* where hands are closest together, *maximum pose* where hands are furthest apart.

4 Results

Classification accuracies of the repetitive hand movements using NBNN with *discriminative poses* are reported in Figure 5. 4-fold validation accuracy of 98.7% (Test1) establishes the effectiveness of *discriminative poses* in recognising repetitive movements. A 4-fold cross-speed (Test2) accuracy of 98.2% demonstrates that *discriminative poses* can recognise the class of movement even when trained using repetitive movements of different speeds to the test data. Due to inter-subject variation, cross-subject (Test3) accuracy for the different folds varied between 51% and 91%, with an average accuracy of 73.8% and standard deviation of 12.2. *Scratch* and *pick* are the most similar movements and were the most often confused. Cross-subject testing was repeated excluding *scratch and pick*,

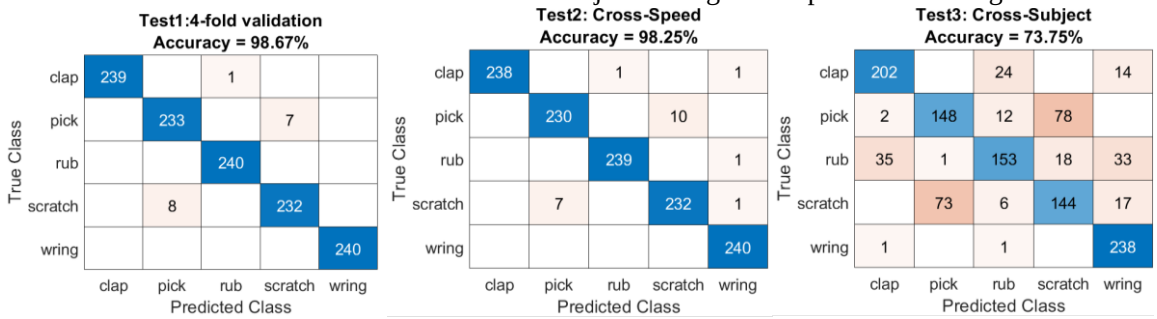


Figure 5: Confusion Matrix for different test protocols using NBNN with *discriminative poses*. Left: Test1, 4-fold validation; centre: Test2, 4-fold cross-speed validation and right: Test2: 15-fold cross subject validation. The method performs well when trained on subject specific data but does not individualise well. It can generalise between the different speeds.

 ISBN 978-0-9934207-5-7

resulting in 84.2% accuracy when *pick* is removed and 84.4% accuracy when *scratch* is removed, shown in Figure 6. This suggests that *discriminative poses* can distinguish between the repetitive movements of new subjects if the movements are sufficiently different.

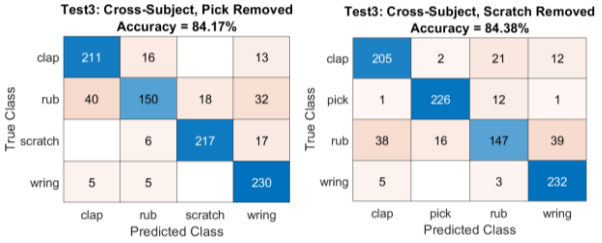


Figure 6: Confusion Matrix showing accuracy when pick or scratch is removed from the dataset.

We compare the Test3 *discriminate poses* accuracies with those of other approaches that use all frames. NBNN achieved the highest accuracy at 74.6%. However, as a lazy classifier, NBNN is very slow at classifying long sequences; each sequence took over 2 seconds to predict making it unsuitable for practical applications. Classification using a Long Short-Term Memory RNN reached 65.9% accuracy indicating that emphasising sequential information may confuse the recognition of repetitive movements. Adding additional features such as wrist angle and joint velocity led to small reductions in overall accuracy.

5 Discussion and Conclusion

Whilst the sequence of frames is important for activity recognition, it is less relevant for recognising repetitive movements. *Discriminative poses* consisting of *mean, minimum* and *maximum poses* provide an effective summary of repetitive movements by ignoring uninformative frames whilst capturing the key patterns of pose and motion. Variations in classification accuracy in Test3 suggests that collecting more data may be useful in improving cross-subject recognition. Our experience of the difficulty in differentiating between different types of repetitive hand movement in new subjects is consistent with other studies; for example, researchers were unable to distinguish between hand flapping and hitting hands together in autistic children [Gilchrist et al., 2018].

This is the first research to focus on the recognition of repetitive hand movements. We have demonstrated a simple and effective method using *discriminative poses* for accurately recognising repetitive hand movements for known subjects. In addition to recognising the type of movement, recognising the frequency and intensity of movements is also important to clinicians and caregivers and is the focus of our ongoing research.

References

[Alam et al. 2018] Alam, Ridwan, Martha Anderson, Azziza Bankole, and John Lach. 2018. "Inferring Physical Agitation in Dementia Using Smartwatch and Sequential Behavior Models." *2018 IEEE EMBS International Conference on Biomedical and Health Informatics, BHI 2018* 2018-January(March): 170–73.

[Chikhaoui et al., 2017] Chikhaoui, Belkacem, Bing Ye, and Alex Mihailidis. 2017. "Feature-Level Combination of Skeleton Joints and Body Parts for Accurate Aggressive and Agitated Behavior Recognition." *Journal of Ambient Intelligence and Humanized Computing* 8(6): 957–76.

[Gilchrist et al., 2018]Gilchrist, Kristin H et al. 2018. "Automated Detection of Repetitive Motor Behaviors as an Outcome Measurement in Intellectual and Developmental Disabilities." *Journal of Autism and Developmental Disorders* 48(5): 1458–1466.

[Herath et al., 2017] Herath, Samitha, Mehrtash Harandi, and Fatih Porikli. 2017. "Going Deeper into Action Recognition: A Survey." *Image and Vision Computing* 60: 4–21.

[Liu et al., 2019] Liu, Yu et al. 2019. "Vision-Based Method for Automatic Quantification of Parkinsonian Bradykinesia." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27(10): 1952–61.

[Marshall et al., 2019] Marshall, Fiona, Shuai Zhang, and Bryan Scotney. 2019. "Comparison of Activity Recognition Using 2D and 3D Skeletal Joint Data." 2019 *Irish Machine Vision and Image Processing Conference*

[Rihawi et al., 2017] Rihawi, Omar, Djamal Merad, and Jean Luc Damoiseaux. 2017. "3D-AD: 3D-Autism Dataset for Repetitive Behaviours with Kinect Sensor." *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017* (August).

[Simon et al., 2017] Simon, Tomas, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. "Hand Keypoint Detection in Single Images Using Multiview Bootstrapping." *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January: 4645–53.

[Yang and Tian, 2014] Yang, Xiaodong, and Yingli Tian. 2014. "Effective 3D Action Recognition Using EigenJoints." *Journal of Visual Communication and Image Representation* 25(1): 2–11.

Sensor tilt via conic sections

Brian O'Sullivan and Piotr Stec

FotoNation Ltd., an Xperi Company, Block 5, Parkmore Business Park East, Galway, Ireland.

Abstract

Distortions due to a misalignment between the lens elements and imaging sensor of a camera, are modelled in this article using conic sections. Light enters the camera lens at a *field angle* θ relative to the optical axis, and exits the lens pupil at the *lens distorted angle* β. For a given lens distortion angle, a cone of possible light paths exist which are intersected by a tilted imaging sensor creating an ellipse of *sensor tilt shifted points*. Conversely a sensor at right angles to the optical axis intersects this cone to create a circle of possible points. We derive the mapping between the circle and ellipse, as a function of the unit normal to the tilted sensor, and outline how this formalism for sensor tilt correction can be integrated into a generic camera calibration algorithm.

Keywords: ccd sensor tilt, tangential distortion, camera calibration, conic sections.

Camera calibration is the process of determining the geometric properties of a camera, and is a prerequisite for making accurate geometric measurements from image data [6]. A calibration algorithm requires both the forward and backward mapping of the camera model. In this way the expected 2D image of a 3D object can be easily generated, and given the 2D image the process is invertible as the 3D point coordinates of the imaged object can be calculated [4, 5, 7]. Calibration is generally performed by photographing a calibration target whose geometric properties are known. Typically the target is a one or three plane checkerboard, and a detector is used to extract the pixel coordinates of the checkerboard corners. Given the camera model, and a set of images with detected corners, the camera parameters are optimized until the errors between the reprojected points from the model, and the detected points in the image, are minimized.

In this article we outline a typical camera calibration algorithm with one modification - we model the distortions caused by misalignments between the lens and imaging sensor using conic sections and non-linear maps [9], in favour of the thin prism model [2] currently in use. The model is illustrated in figure 1.

1 Camera model

Following refs [1]-[11] a generic camera calibration algorithm can be represented by the sequence of mappings:

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens}} \hat{p} \xrightarrow{\text{sensor}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

World coordinates $\hat{X} \mapsto \hat{P}$: The matrix $\hat{X} = [\vec{X}_1\ \vec{X}_2\ \vec{X}_3\ \cdots\ \vec{X}_i\ \cdots\ ]$ are the 3D homogeneous coordinates of the test target corners $\vec{X}_i = (X_i\ Y_i\ Z_i\ 1)^\dagger$, expressed in the coordinates system of the target. $\hat{P}$ are the world coordinates of the target corners after rotation and translation by the extrinsic matrix, $\hat{P} = [\,\hat{R}\,|\,\vec{t}\,]\,\hat{X}$.

Lens distorted coordinates $\hat{P} \mapsto \hat{p}$: The world points $\hat{P}$ are projected through the lens according to the lens' radial distortion function

$$r(\theta_i) = \theta_i + \kappa_1\theta_i^3 + \kappa_2\theta_i^5 + \kappa_3\theta_i^7 + \cdots$$

 ISBN 978-0-9934207-5-7

where $\kappa_1, \kappa_2, \ldots$ are the Taylor series coefficients, and the field angle θ_i is the arctangent of the length of the world point $\vec{P}_i$ in the xy plane, divided by the z component. The azimuthal angle is the arctangent of the x and y coordinates. We drop the use of the point index i for brevity $\theta_i \to \theta$, $\phi_i \to \phi$, and $\vec{P}_i = \left(P_x, P_y, P_z\right)$.

$$\theta = \tan^{-1}\left[\sqrt{P_x^2 + P_y^2}/P_z\right] \qquad \phi = \tan^{-1}\left[P_y/P_x\right]$$

These parameters are then used to describe the lens distorted points in homogeneous camera coordinates:

$$\vec{p} = \begin{pmatrix} r(\theta)\cos(\phi) & r(\theta)\sin(\phi) & 1 \end{pmatrix}^\dagger \qquad \text{and} \qquad \hat{p} = \begin{bmatrix} \vec{p}_1 \, \vec{p}_2 \cdots \vec{p}_i \cdots \end{bmatrix}$$

The points $\hat{p}$ are projected onto the tilted imaging sensor as illustrated in figure $1(a)$.

The next mapping $\hat{p} \xmapsto{\text{sensor}} \hat{p}'$, takes account of distortions that arise due to the misalignment of the lens elements and imaging sensor. Typically these are referred to as tangential distortions and are accounted for using the Brown-Conrady model [1, 2]. At this point in the discussion we deviate from tradition and propose to model pixel distortions from a tilted ccd imaging sensor, using conic sections and non-linear maps.

Illustrated in figure $1(c)$ is a side view of the camera model. The world points are projected through the lens element onto the imaging sensor(s). The ccd sensor has 2 positions. In green is the ideal position - perpendicular to the optical axis, and in red is the tilted sensor. The perpendicular and tilted sensors are labelled S_0 and S_1, and have unit normals $-\vec{z}$, and $\vec{n}$, respectively. Photons emanating from the world point(s) $\vec{P}$ are projected through the lens elements to impact the perpendicular sensor S_0 at the point(s) $\vec{p}$.

$$\vec{z} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\dagger \qquad \vec{n} = \begin{pmatrix} n_x & n_y & n_z \end{pmatrix}^\dagger \qquad \vec{p} = \begin{pmatrix} p_x & p_y & 1 \end{pmatrix}^\dagger$$

Tilt shifted coordinates $\hat{p} \mapsto \hat{p}'$: The projected point forms a vector $\lambda\vec{p}$ touching the tilted plane S_1, and connecting with the vector $\vec{q}$ in the local coordinates of S_1, see figure $1(c)$. Taking the dot product of both sides of this vector equation, with the unit normal $\vec{n}$ to the tilted sensor S_1, we arrive at an expression for the scaling factor λ, since $\vec{n} \cdot \vec{q} = 0$.

$$\vec{q} = \lambda\vec{p} - \vec{z} \qquad \to \qquad \lambda = \vec{n} \cdot \vec{z}\left(\vec{n} \cdot \vec{p}\right)^{-1} \qquad \to \qquad \lambda = n_z\left(n_x p_x + n_y p_y + n_z\right)^{-1} \tag{1}$$

In order to express $\vec{q}$ in the local co-ordinate system of S_0, the S_1 plane is rotated to align with the (camera) coordinate system of S_0. As the lens is rotationally symmetric about the optical axis, the roll angle is ambiguous, and S_1 maps to S_0 as the unit normal $\vec{n}$ is rotated to $-\vec{z}$. The Rodrigues rotation matrix with rotation axis $\vec{k} = -\vec{n} \times \vec{z}$, and rotation angle $\vartheta = \cos^{-1}(-\vec{n} \cdot \vec{z})$, rotates the vector $\vec{n}$ to $-\vec{z}$:

$$\vec{k} = \begin{pmatrix} -n_y & n_x & 0 \end{pmatrix}^\dagger \qquad \cos(\vartheta) = -n_z$$

and the rotation matrix simplifies to

$$\hat{R} = \hat{\sigma}_1 + \hat{k} + \hat{k}^2 \frac{1}{1 + \cos(\vartheta)} = \begin{pmatrix} 1 + \frac{n_x^2}{n_z - 1} & \frac{n_x n_y}{n_z - 1} & n_x \\ \frac{n_x n_y}{n_z - 1} & 1 + \frac{n_y^2}{n_z - 1} & n_y \\ -n_x & -n_y & -n_z \end{pmatrix} \tag{2}$$

where $\hat{\sigma}_1$ is the identity matrix, and $\hat{k}$ is a skew symmetric matrix composed of the elements of $\vec{k}$.

To perform the rotation around the projective plane origin, the points are first shifted to the coordinate centre $\vec{o}$, then rotated, then shifted back to the original location. The tilt shifted points in the coordinate system of the S_0 sensor plane are defined:

$$\vec{p}\,' = \hat{R}\left(\lambda\vec{p} - \vec{z}\right) + \vec{z} \tag{3}$$

The tilt shifted map $\vec{p} \mapsto \vec{p}\,'$:

$$p_x{}' = \left(\left(n_x^2 + n_z(n_z - 1)\right)p_x + n_x n_y p_y\right)\left(n_x p_x + n_y p_y + n_z\right)^{-1}(n_z - 1)^{-1} \tag{4a}$$

$$p_y{}' = \left(\left(n_y^2 + n_z(n_z - 1)\right)p_y + n_x n_y p_x\right)\left(n_x p_x + n_y p_y + n_z\right)^{-1}(n_z - 1)^{-1} \tag{4b}$$

 ISBN 978-0-9934207-5-7

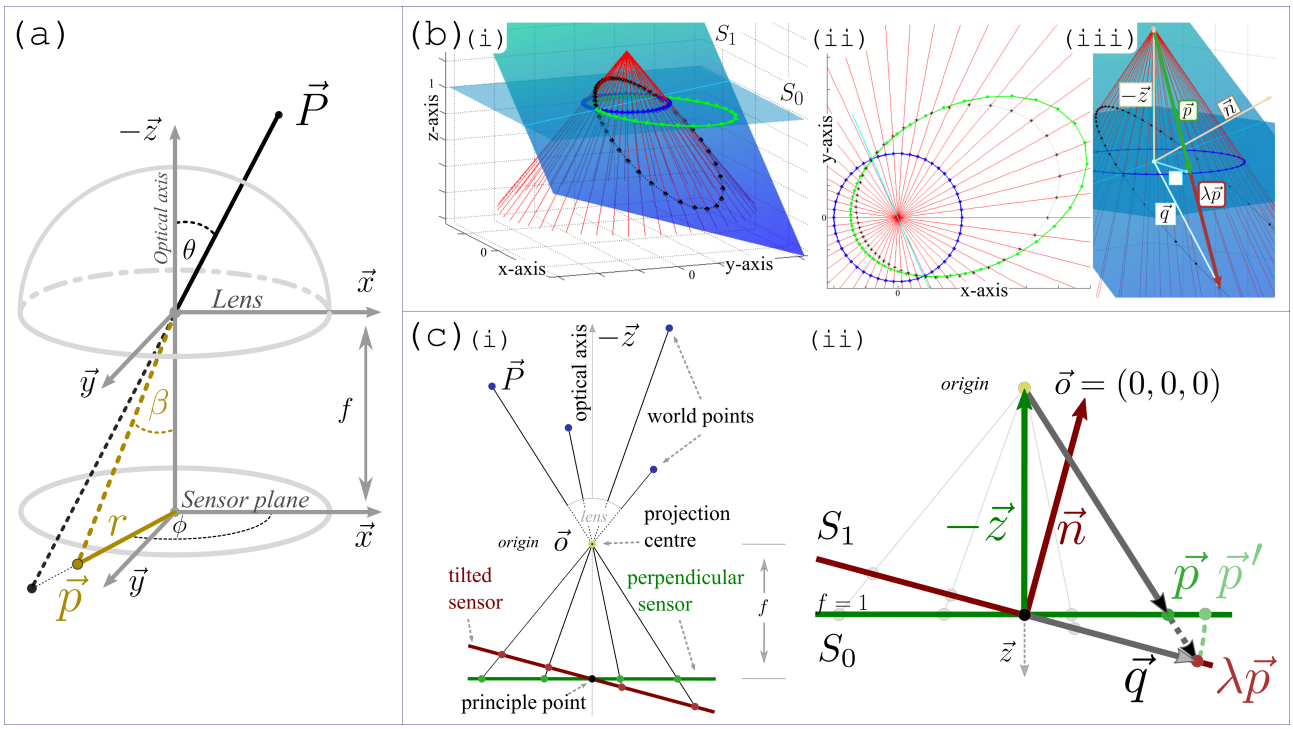


Figure 1: (*a*) Camera model. (*b*) Conic sections (*i*) intersection of the cone with the sensors S_0 and S_1, (*ii*) top view of the cone-plane intersection showing the circle to ellipse transformation, (*iii*) illustration of the vector map. (*c*) Sensor tilt (*i*) schematic, (*ii*) vector map.

Pixel coordinates $\hat{p}' \mapsto \hat{x}$: The mapping from homogeneous camera coordinates to pixel coordinates is the linear transformation $\hat{x} = \hat{K}\hat{p}'$, where $\hat{K}$ is the intrinsic camera matrix containing the xy-focal lengths (f_x, f_y) and the optical centre (x_c, y_c). This concludes our outline of the generic camera model.

2 Inverse camera model

For the purposes of camera calibration it is equally important that the inverse mappings of the camera model are known. The inverse camera model begins with the imaged 2D target corners $\hat{x}$, unfolding until the registered 3D target corners $\hat{X}$.

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor}} \hat{p} \xrightarrow{\text{lens}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

The inverse mappings are specifically used in the calculation of the Jacobian, and non-linear optimization of the camera parameters. As sensor tilt via conic sections is the only new consideration to the calibration protocol, here we confine our discussion to the inverted map of the tilt shifted points.

Tilt compensated coordinates $\hat{p}' \mapsto \hat{p}$: Given the tilt distorted points $\vec{p}'$ generated by a sensor of known tilt $\vec{n}$, the tilt compensated coordinates $\vec{p}$ are obtained by first shifting the points to $\vec{o}$, performing the inverse rotation $\hat{R}^{\dagger}$, shifting back, and then rescaling through division by the z component.

$$\vec{p} = \lambda' \left(\hat{R}^{\dagger} \left(\vec{p}' - \vec{z} \right) + \vec{z} \right) \tag{5}$$

where the scaling factor λ' is the pinhole projection,

$$\lambda' = \left(n_x p'_x + n_y p'_y + 1 \right)^{-1} \tag{6}$$

 ISBN 978-0-9934207-5-7

The tilt corrected map $\vec{p}\,' \mapsto \vec{p}$:

$$p_x = \left(\left(n_x^2 + n_z - 1\right) {p_x}' + n_x n_y {p_y}'\right)\left(n_x {p_x}' + n_y {p_y}' + 1\right)^{-1} (n_z - 1)^{-1} \tag{7a}$$

$$p_y = \left(\left(n_y^2 + n_z - 1\right) {p_y}' + n_x n_y {p_x}'\right)\left(n_x {p_x}' + n_y {p_y}' + 1\right)^{-1} (n_z - 1)^{-1} \tag{7b}$$

3 Summary and outlook

We have shown that conic sections and non-linear maps are a geometric transformation that describes pixel distortions due to a tilted ccd sensor, in a natural way. We believe that the conic sections model for sensor-lens misalignment is of great interest for camera calibration algorithms when and where precision is paramount.

References

[1] S. Beauchemin, R. Bajcsy and G. Givaty, *"Modelling and Removing Radial and Tangential Distortions in Spherical Lenses"* Multi-Image Analysis. Lecture Notes in Computer Science, **2032**: *pages* 1-21 (2001).

[2] D.C. Brown *"Decentering Distortion of Lenses"* Photometric Eng., **32**(3) *pp* 444-462, (1966). J.G. Fryer and D.C. Brown, *" Lens distortion for close-range photogrammetry"*, Photogrammetric Engineering and Remote Sensing, Vol. **52**(1): *pp* 51-58, (1986).

[3] Donald B. Gennery, *"Generalized Camera Calibration Including Fish-Eye Lenses"* International Journal of Computer Vision, **68**: *pages* 239-266 (2006).

[4] R. Hartley and A. Zisserman, *"Multiple View Geometry in computer vision"*. Cambridge University Press (2003). R.Hartley and S. B. Kang, *"Parameter free radial distortion correction with center of distortion estimation,"* IEEE Trans. on Pattern Analysis and Machine Intelligence, **29**(8): *pp* 1309-1321 (2008).

[5] J. Heikkila and O. Silven, *"A four-step camera calibration procedure with implicit image correction "*. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1997).

[6] J. Kannala and S. Brandt, *"A generic camera calibration method for fish-eye lenses."* ICPR *pp* 10-13 (2004). J. Kannala, J. Heikkilä and S. Brandt. *"Geometric camera calibration."* Wiley Encyc. of Comp. Sci. and Eng., (2008).

[7] Davide Scaramuzza, Agostino Martinelli and Roland Siegwart, *"A Toolbox for Easily Calibrating Omnidirectional Cameras"* IEEE conference on Intelligent Robots and Systems, *pages* 5695-5701 (2006). Davide Scaramuzza, Agostino Martinelli and Roland Siegwart. *"A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion"*, Proceedings of IEEE International Conference of Vision Systems, January (2006).

[8] S. Shah and J. Aggarwal *"Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation."* Patt. Rec. **29**(11): *pp* 1775-1788 (1996).

[9] Piotr Stec and Brian O'Sullivan, *"Method for compensating for the off axis tilting of a lens"* United states patent number: 10,356,346 B1, July (2019). github: mo-geometry/conic_sections

[10] R.Y. Tsai, *"An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision."* Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, *pages* 364-374 (1986). R. Y. Tsai *"A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses."* IEEE J. Robotics Automation, **3**(4): *pages* 323-344 (1987). B.K.P. Horn, *"Tsai's camera calibration method revisited."* MIT coursework (2000).

[11] Z. Zhang. *"A flexible new technique for camera calibration."* IEEE Transactions on Pattern Analysis and Machine Intelligence, **22**(11): *pages* 1330-1334 (2000).

Published by the Irish Pattern Recognition & Classification Society

iprcs.org

ISBN 978-0-9934207-4-0