IRISH MACHINE VISION & IMAGE PROCESSING

Conference proceedings 2015

IPRCS

Irish Pattern Recognition & Classification Society

Editors:
Rozenn Dahyot
Gerard Lacey
Kenneth Dawson-Howe
François Pitié
David Moloney

Published by the Irish Pattern Recognition & Classification Society

iprcs.org

ISBN 978-0-9934207-0-2

Introduction

The 2015 Irish Machine Vision and Image Processing Conference (IMVIP 2015) was hosted this year at Trinity College Dublin, the University of Dublin, under the organisation of the School of Computer Science and Statistics, and the Department of Electronic and Electrical Engineering.

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IMVIP is a single track conference consisting of high quality previously unpublished contributed papers focussing on both theoretical research and practical experiences in all areas. After a rigorous review process, 19 papers were selected for the conference. We wish to sincerely thank the members of the Programme Committee for generously giving their time, effort and expertise in reviewing the submissions.

Continuing the tradition of inviting high-profile speakers to IMVIP, we are delighted to have keynote presentations at IMVIP 2015 from Prof. Takeo Kanade (Carnegie Mellon University), Dr. Aljosa Smolic (Disney), Prof. Anil Kokaram (Youtube & Trinity College Dublin), Dr. John McDonald (Maynooth University) and Prof. Oscar Deniz Suarez (University of Castilla-La Mancha).

IMVIP is run in association with the Irish Pattern Recognition & Classification Society (iprcs.org), a member organisation of the International Association for Pattern Recognition (IAPR) and the International Federation of Classification Society (IFCS). In addition to IPRCS, we would like to thank industry sponsors to IMVIP 2015 Movidius, Surewash, Daqri and Fotonation, as well as Trinity College and the Science Gallery for facilitating this conference.

As a final note, many thanks to Seán Cronin, Mairéad Grogan, Seán Bruton and Abdullah Bulbul for their help organising IMVIP 2015 conference.

Rozenn Dahyot, Gerard Lacey, Kenneth Dawson-Howe, François Pitié & David Moloney
Trinity College Dublin
Ireland
August 2015

 ISBN 978-0-9934207-0-2

Keynote speaker: Takeo Kanade

Sponsored by Movidius

Title: *Smart Headlight: A new active augmented reality that improves how the reality appears to a human*

Abstract: A combination of computer vision and projector-based illumination opens the possibility for a new type of computer vision technologies. One of them is augmented reality: selectively illuminating the scene to improve or manipulate how the reality itself, rather than its display, appears to a human. One such example is the Smart Headlight being developed at Carnegie Mellon University's Robotics Institute. The project team has been working on a set of new capabilities for the headlight, such as making rain drops and snowflakes disappear, allowing for the high beams to always be on without glare, and enhancing the appearance of objects of interest. Using the Smart Headlight as an example, this talk will further discuss various ideas, concepts and possible applications of coaxial and non-coaxial projector-camera systems.

About the speaker: Professor Takeo Kanade is the U. A. and Helen Whitaker University Professor of Computer Science and Robotics at Carnegie Mellon University. He received his Doctoral degree in Electrical Engineering from Kyoto University, Japan, in 1974. After holding a faculty position there, he joined Carnegie Mellon University in 1980. He was the Director of the Robotics Institute from 1992 to 2001. He also founded and directed the Digital Human Research Center in Tokyo from 2001 to 2010, and the Quality of Life Technology Center at Carnegie Mellon from 2006 to 2012. Dr. Kanade works in multiple areas of robotics: computer vision, multi-media, manipulators, autonomous mobile robots, medical robotics and sensors, producing more than 400 technical papers and more than 20 patents. Dr. Kanade has been elected to the National Academy of Engineering and the American Academy of Arts and Sciences. Awards he received include the Franklin Institute Medal and Bower Prize, ACM/AAAI Newell Award, Okawa Award, NEC Computer and Communication Award, Joseph Engelberger Award, IEEE Robotics and Automation Society Pioneer Award, and IEEE PAMI Azriel Rosenfeld Lifetime Accomplishment Award.

 ISBN 978-0-9934207-0-2

Keynote speaker: Aljoša Smolić

Sponsored by surewash

Title: *Thinking in Video Volumes*

Abstract: Video is typically represented as a temporal sequence of arrays of values. These values contain strong interconnections in spatial and temporal dimensions, which can be exploited for efficient processing. However, computational complexity and memory restrictions limited exploitation of temporal interconnections in the past. Today's computing power enables development of a new class of algorithms that operate on video volumes. FeatureFlow provides efficient solutions for classical problems in visual computing such as optical flow, disparity estimation, and data propagation. DuctTake is a novel approach for spatio-temporal video compositing. Temporally consistent tone mapping of HDR video is another application scenario of this principle, which will be covered in this talk.

About the speaker: Dr. Aljoša Smolić joined Disney Research Zurich, Switzerland in 2009, where he is employed as Senior Research Scientist and Head of the *Video of the Future* group. He has been involved in several national and international research projects, where he conducted research in various fields of video processing, video coding, computer vision and computer graphics and published more than 90 referred papers in these fields. In current projects he is responsible for research in 2D video, 3D video and free viewpoint video processing and coding.

 ISBN 978-0-9934207-0-2

Keynote speaker: Anil Kokaram

Sponsored by FotoNation

Title: *Pushing Pixels at YouTube*

Abstract: The Video Infrastructure Division is concerned with the care, feeding and transport of pixels from ingested source material to the final display device. With more than 1 Billion users, 300 hours of video uploaded per minute, and thousands of different output devices as targets, the technological challenges are significant. This talk exposes some of the technology behind the massively distributed transcoding and broadcast center that is YouTube. We consider in particular the difficulties caused by scale and highlight the importance of high level, automated "black box" control for many of the video processing tools which are considered standard today.

About the speaker: Prof. Anil Kokaram is a Tech Lead in the Transcoding Group at YouTube/Google. He leads a small team responsible for video quality and develops video processing algorithms for quality improvement in various pipelines. He is also a Professor at Trinity College Dublin, Ireland and continues to supervise a small number of students at www.sigmedia.tv in the EE Dept there. His main expertise is in the broad areas of DSP for Video Processing. Bayesian Inference and motion estimation. He has published over 100 refereed papers in these areas. In 2007 he was awarded a Science and Engineering Academy Award for his work in video processing for post-production applications. He was founder of a company (GreenParrotPictures) producing video enhancement software that was acquired by Google in 2011. He is currently an Associate Editor of the IEEE Transactions on CCts and Systems for Video Technology.

 ISBN 978-0-9934207-0-2

Keynote speakers: John McDonald

Sponsored by

Title: *Visual SLAM: from sparse mapping to 3D perception*

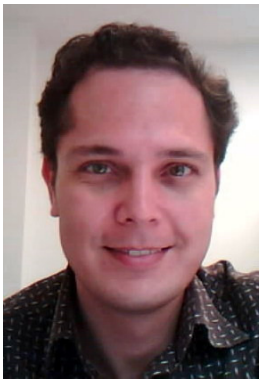
Abstract: From fully autonomous vehicles to markerless AR, gaming to household robotics, recent progress in visual SLAM is providing such systems with the foundations for higher level scene interpretation, visualisation, and interaction. This talk will provide an overview of the visual SLAM problem in the context of two systems developed jointly between Maynooth University and MIT. The first system employs a feature based approach for multi-session visual mapping where multiple separate mapping sessions can be combined into a single globally consistent model of the environment. The second system, known as Kintinuous, provides a real-time dense SLAM system that allows globally consistent mesh based mapping over extended scales. Results will be presented for both systems using a number of different datasets. Finally the talk will present the application of Kintinuous to a number of robotic tasks demonstrating the benefits of the resulting dense representations for 3D perception.

About the speaker: Dr. John McDonald has been a lecturer at the Department of Computer Science, NUI Maynooth since 1997 where he leads the Computer Vision Group. His primary research interest is computer vision, working in areas including visual simultaneous localisation and mapping (vSLAM), intelligent vehicle systems, vision based geotechnologies, face and gesture analysis, and digital holography. He has been a visiting scientist at the University of Connecticut, the National Centre for Geocomputation (NCG) at NUI Maynooth, and the Computer Science and Artificial Intelligence Lab (CSAIL) at the Massachussetts Institute of Technology. He is a member of the Callan Institute at NUI Maynooth.

 ISBN 978-0-9934207-0-2

Keynote speakers: Oscar Deniz Suarez

Sponsored by Movidius



Title: *Project Eyes of Things*

Abstract: Vision, our richest sensor, allows inferring big data from reality. Arguably, to be “smart everywhere” we will need to have “eyes everywhere”. Currently, computer vision is rapidly moving beyond academic research and factory automation. The possibilities are endless in terms of wearable applications, augmented reality, surveillance, ambient-assisted living, etc. Vision is, however, the most demanding sensor in terms of power consumption and required processing power, which can explain the shortage of development platforms with low-cost mobile processing and IoT features. Our objective in this EU-funded innovation project running from 2015 to 2017 is to build an optimized core vision platform that can work independently and also embedded into all types of artefacts.

About the speaker: Prof. Oscar Deniz Suarez is an Associate Professor at University of Castilla-La Mancha. He is the author of more than 50 refereed papers in journals and conferences. He has 2 patents and has also contributed to OpenCV, the well-known open source computer vision library. He is the author of 3 books on OpenCV and OpenCV programming for mobile devices. Oscar has also served as visiting researcher at Carnegie Mellon University (USA), Imperial College London (UK) and Leica Biosystems (Ireland). He is a Senior Member of IEEE and is affiliated with the AAAI, IAPR and The Computer Vision Foundation. He serves as an Academic Editor of Journal PLoS ONE and Associate Editor of IEEE Consumer Electronics. Currently, he is the Coordinator of EU Horizon 2020 "Eyes of Things" project and partner in FP7 AIDPATH Marie Curie Action. He serves as a reviewer/expert for EU programs such as Eurostars.

 ISBN 978-0-9934207-0-2

Conference Chairs

- Rozenn Dahyot, Trinity College Dublin
- Gerard Lacey, Trinity College Dublin
- Kenneth Dawson-Howe, Trinity College Dublin
- Francois Pitié, Trinity College Dublin

Industry Chair

- David Moloney, Movidius

Programme Committee

- Abdullah Bulbul, Trinity College Dublin
- Ahmed Bouridane, Northumbria University, Newcastle, UK
- Andy Shearer, National University of Ireland Galway
- Antonio Fernández, University of Vigo, Spain
- Bob Fisher, University of Edinburgh
- Bryan Gardiner, Ulster University
- Bryan W. Scotney, Ulster University
- Cem Direkoglu, Middle East Technical University, Cyprus
- Danny Crookes, Queen's University of Belfast
- David Vernon, University of Skövde, Sweden
- Derek Molloy, Dublin City University
- Dermot Kerr, Ulster University
- Donald Bailey, Massey University, New Zealand
- Fionn Murtagh, University of London, UK
- Francesco Bianconi, University of Perugia, Italy
- George Moore, Ulster University
- Hiroshi Sako, Hosei University, Japan
- Jane Courtney, Dublin Institute of Technology
- Joan Condell, Ulster University
- John Barron, The Univerity of Western Ontario, Canada
- John Mc Donald, National University of Ireland Maynooth
- John Winder, Ulster University
- Jonathan Ruttle, SureWash, Dublin
- Kathleen Curran, Univerity College Dublin
- Kevin McGuinness, Dublin City University
- Madonna Herron, Ulster University
- Nicholas Devaney, National University of Ireland, Galway
- Noel O'Connor, Dublin City University
- Paul Mc Kevitt, Ulster University
- Paul Miller, Queen's University of Belfast
- Paul Whelan, Dublin City University
- Philip Morrow, Ulster University
- Reyer Zwiggelaar, Aberystwyth University, UK
- Robert Sadleir, Dublin City University

 ISBN 978-0-9934207-0-2

- Sally McClean, Ulster University
- Sonya Coleman, Ulster University
- Sudeep Sarkar, University of South Florida, USA
- Tom Naughton, National University of Ireland Maynooth

Table of Contents

1 Dynamic Texture Classification using Combined Co-Occurrence Matrices of Optical Flow
V. Andrearczyk & P. F. Whelan **3**

2 Investigation into DCT Feature Selection for Visual Lip-Based Biometric Authentication
C. Wright, D. Stewart, P. Miller & F. Campbell-West **11**

3 3D Reconstruction of Reflective Spherical Surfaces from Multiple Images
A. Bulbul, M. Grogan & R. Dahyot **19**

4 Kernel Density Filtering for Noisy Point Clouds in One Step
M.A. Brophy, S.S. Beauchemin & J.L. Barron **27**

5 Multiscale "Squiral" (Square-Spiral) Image Processing
M. Jing, S.A. Coleman, B.W. Scotney & M. McGinnity **35**

6 Hand Hygiene Poses Recognition with RGB-D Videos
B. Xia, R. Dahyot, J. Ruttle, D. Caulfield & G. Lacey **43**

7 Architecture for Recognizing Stacked Box Objects for Automated Warehousing Robot System
T. Fuji, N.Kimura & K. Ito **51**

8 Symmetry and Repeating Structure Detection
M. Jilani, P. Corcoran & M. Bertolotto **59**

9 Bayer Interpolation with Skip Mode
D.G. Bailey, M. Contreras & G. Sen Gupta **68**

10 Gradient Magnitude Based Normalised Convolution
A. Al-Kabbany, S. Coleman & D. Kerr **76**

11 PatchCity: Procedural City Generation using Texture Synthesis
J.D. Bustard & L. P. de Valmency **83**

12 Simplifying Genetic Algorithm: A Bit Order Determined Sampling Method for Adaptive Template Matching
C. Zhang & T. Akashi **91**

13 Automatic Segmented Area Structured Lighting
K. Goyal, H. Baghsiahi & D.R. Selviah **97**

14 Machine Learning in Prediction of Prostate Brachytherapy Rectal Dose Classes at Day 30
P. Leydon, F. Sullivan, F. Jamaluddin, P. Woulfe, D. Greene & K. Curran **105**

15 Resolution enhancement of thermal imaging
C. Lynch, N. Devaney & A. Drimbarean **110**

16 Interpolating eigenvectors from second-stage PCA to find the pose angle in handshape recognition
M. Oliveira & A. Sutherland **114**

17 Range Image Feature Extraction using a Hexagonal Pixel-based Framework
B. Gardiner & S. Coleman **118**

18 Investigation of Face Tracking Accuracy by Obscuration Filters for Privacy Protection
J. Sato & T. Akashi **122**

19 Cone detection and blood vessel segmentation on AO retinal images
L. Mariotti & N. Devaney **126**

 ISBN 978-0-9934207-0-2

Dynamic Texture Classification using Combined Co-Occurrence Matrices of Optical Flow

V. Andrearczyk & Paul F. Whelan

Centre for Image Processing and Analysis (CIPA), Dublin City University, Dublin 9, Ireland

Abstract

This paper presents a new approach to Dynamic Texture (DT) classification based on the spatiotemporal analysis of the motion. The Grey Level Co-occurrence Matrix (GLCM) is modified to analyse the distribution of the magnitude and the orientation of the Optical Flow which describes the motion. Our method is therefore called Combined Co-occurrence Matrix of Optical Flow (CCMOF). The potential of a multiresolution analysis of the motion is revealed by experimentation. We also demonstrate the importance of the analysis of motion in the spatiotemporal domain. Finally, we demonstrate that adding a spatiotemporal motion analysis (CCMOF) to an appearance analysis (Local Binary Patterns on Three Orthogonal Planes (LBP-TOP)) significantly improves the classification results.

Keywords: Dynamic Texture, classification, Optical Flow, co-occurrence, spatiotemporal

1 Introduction

Dynamic Texture (DT) is an extension of static texture in the temporal domain, introducing temporal variations such as motion and deformation. Doretto et al. [6] describe a DT as a sequence of images of moving scenes that exhibits certain stationary properties in time. Examples of natural DTs include smoke, clouds, trees and waves. The analysis of DTs embraces several major problems including classification, segmentation, synthesis and indexing for retrieval. Such analysis is essential for a large range of applications including surveillance, medical image analysis and remote sensing. This paper focuses on the classification of DTs. However, the developed algorithm and ideas can be used in other DT problem domains. The aim in DT classification is to assign an unknown sequence to a set of DT classes.

The analysis of motion for DT classification is of particular interest due to both the logic of the approach and the positive results obtained in the literature. Motion, along with shapes and color, is a key element in the analysis of a scene by the human visual system [21]. Moreover, while watching a sequence of images, the human's brain interprets the succession of still frames as a dynamic moving scene [1]. Therefore, combining a motion and an appearance analysis seems to be a natural way of dealing with dynamics. However, the literature either focuses on the spatial distribution of the Optical Flow (OF) (*motion based*) or on the pixel intensities' distribution in the spatiotemporal domain (*statistical* and *transforms*). A significant amount of information is lost by neglecting the evolution of the motion over time. In this paper, we present two new methods for DT classification which explore the potential of the analysis of the motion in the spatiotemporal domain. We use a co-occurrence matrix approach applied on the OF field in thespatiotemporal domain. We investigate the positive impact of a temporal multiresolution analysis of the motion as well as the combination of motion and appearance analysis.

2 State of the Art

The main methods developed for DT analysis can be classified in four categories, namely *statistical*, *motion based*, *model based* and *spatiotemporal transforms*.

Statistical approaches mainly use standard texture analysis in a new manner to include the temporal dimension. The Grey Level Co-occurrence Matrix (GLCM) is used by Flores et al. [8] to extract a set of features on each frames of the DT sequence. This is the most basic adaptation of a static texture analysis method to DT analysis. Hu et al. [14] also use well-known spatial descriptors on each frame but combine them with GLCM features calculated in the temporal domain to capture the correlation of neighbouring pixels in time. Boujiha et al. describe in [2] a spatiotemporal co-occurrence matrix approach, extracting co-occurrence matrices on a 3D (x, y, t) neighbourhood. In a similar manner, Zhao and Pietikäinen extend the Local Binary Pattern (LBP) to DTs by creating the LBP Volume (LBP-V) [24] and the LBP on Three Orthogonal Planes (LBP-TOP) [23]. The latter extracts LBPs on three planes: XY which is the classic spatial texture LBP as well as XT and YT which consider temporal variations of the pixel intensities.

Realising the limitations of only analysing a DT as a 3D image, **motion based** methods were developed. Extracting the motion between consecutive frames of the DT sequence can be achieved, among other methods, by Normal Flow [16], OF [4, 7, 15], or Local Motion Pattern (LMP) [9]. Various statistical analyses have been developed to extract features describing the spatial distribution of the motion field such as GLCM, Fourier spectrum, difference statistics [16], histograms [4, 9, 15] and statistics on the derivatives of the OF [7]. These motion features are often logically combined with features based on spatial statistical analyses.

Introduced by Saisan et al. in [20], **model based** methods aim at estimating the parameters of a Linear Dynamical System (LDS) using a system identification theory. This approach is designed for a synthesis problem. However, the estimated parameters can be used for a classification task [6]. Positive results were obtained in the learning and synthesis of temporal stationary sequences such as waves and clouds [6]. However, the model based approach raises several difficulties such as the distance between models lying in a non-linear space and the non-invariance to rotation and scale. Finally, it is not suitable for segmentation since it assumes stochastic and segmented DTs. Ravichandran et al. [18] overcome the view-invariance problem using a Bag of dynamical Systems (BoS) similar to a Bag of Features with LDSs as feature descriptors, obtaining precise results.

In the same manner as statistical methods, several transform approaches used in texture analysis were extended to **spatiotemporal transform** for DT analysis. Derpanis and Wildes [5] use spacetime oriented filters extracting interesting features which describe intrinsic properties of the DTs such as unstructured, static motion and transparency. In [17], Qiao and Wang use 3D Dual Tree Complex Wavelet combining an appearance with a dynamic analysis. Finally, Gonçalves et al. use spatiotemporal Gabor filters in [11].

3 Classification using Extended Plots and multiresolution

In this section, we develop a DT classification method based on the analysis of the OF. The idea of extracting the history of motion on Extended Directional and Extended Magnitude Plots developed in [16] is further explored in particular with a multiresolution analysis, different features and a rotation invariant method.

Co-occurrence matrix background The Grey Level Co-occurrence Matrix (GLCM) aims at describing the relationships between neighbouring pixel intensities by analysing their joint probability function [12, 13]. The GLCM summarizes the occurrence of pairs of pixels on a texture image. The $(i, j)^{th}$ entry of the matrix represents the number of times a pixel with intensity value i is separated from another pixel with intensity value j at a distance d in the direction θ. This matrix contains meaningful information about the distribution of the pixel intensities as indicated by the 14 Haralick's features [12, 13].

 ISBN 978-0-9934207-0-2

Method description

OF extraction: The OF is calculated between every two consecutive frames of the sequence using Black's algorithm [22]. As suggested by Sun et al. [16], most magnitudes being in the range zero to four pixels, the flow vectors with magnitudes greater than four pixels are regarded as noise and labelled *stationary points*. The other points are called *moving points*.

Quantisation: Based on [16], the magnitude and the direction of the moving points are mapped into the Magnitude Plots and the Directional Plots. The magnitude is arbitrarily mapped into nine grey levels shown in Table 1. Regions of darker grey in the Magnitude Plots correspond to regions with larger normal flow magnitudes. The direction is mapped into nine grey levels as shown in Figure 1.

magnitude value	mapping
]3.555, 4]	0
]3.111 , 3.555]	31
]2.666 , 3.111]	63
]2.222 , 2.666]	95
]1.777 , 2.222]	127
]1.333 , 1.777]	159
]0.888 , 1.333]	191
]0.444 , 0.888]	223
[0 , 0.444]	255

Table 1: Magnitude mapping of moving points.

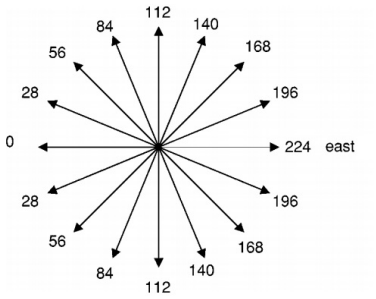


Figure 1: Quantisation of normal flow direction into 16 grey levels. (figure extracted from [16])

Extended Plots: The Extended Magnitude Plots (EMPs) and Extended Directional Plots (EDPs) [16] are calculated in order to trace the motion history. τ Magnitude Plots f_t are "superimposed" to construct an EMP, where $t \in [1, \tau]$ is the position of the plot in the temporal domain. That is to say, the EMP $F(i, j)$ at pixel location i and j takes the value of the last moving point in $f_t(i, j)$. It takes the value 255 if $f_1(i, j), f_2(i, j), \ldots, f_\tau(i, j)$ are only stationary points. The EDP $G(i, j)$ is similarly calculated from τ Directional Plots $g_t(i, j)$, resulting in 10 grey levels (9 directions and an extra level 255 for stationary points). Following the setup from [16], we choose $\tau = 5$.

Multiresolution: A basic multiresolution analysis is performed, with the purpose of demonstrating the potential of this approach in a DT classification problem. The original sequences are decomposed into several temporal resolutions by applying a Gaussian pyramid.

Dynamic Texture features: The co-occurrence matrices of the EMPs (9 by 9 matrices) and of the EDPs (10 by 10 matrices) are calculated and averaged over the whole sequence for each resolution. An angle $\theta = 0°$ and a distance $d = 1$ pixel between the pixel neighbours are chosen in this experiment as a proof of concept. The final features are composed of the raw mean co-occurrence matrices of each resolution vectorized and concatenated in one single feature vector.

Rotation Invariance: In order to further estimate the robustness of the algorithm, rotation invariance is developed in the feature extraction process. The rows and columns of the co-occurrence matrices are re-organised depending on the dominant orientation of the EDPs.This is similar to rotating the OF vectors so that the dominant direction always points leftwards. Hence, similar features are extracted, for instance, from sequences of cars moving in different directions.

Results and discussions The developed algorithm is tested on the Dyntex++ database [10]. It consists of 36 classes of 100 sequences of size 50*50*50. The experimental setup is similar to [10, 18, 19]. 50 sequences are randomly selected from each class as the training set, and the other 50 sequences are used in testing. This process is repeated 100 times to obtain the average classification rate. We use a linear Support Vector Machine (SVM) classifier [3]. The classification results are summarised in Table 2 and compared to the state

of the art in Table 3. The state of the art method which obtains the best results ([19]) uses PCA-cLBP (PCA on the concatenated LBP), PI-LBP (Patch-Independent), PD-LBP (Patch-Dependent) and a super histogram averaging the histograms across the sequence. Finally it uses a Radial Basis Function (RBF) SVM classifier. Our experiment does not aim to maximise the classification results, but rather to demonstrate the importance of the OF, the multiresolution analysis and the rotation invariance in a DT classification framework. It performs only 2.7% worse than the state of the art on the Dyntex++ database. The multiresolution analysis greatly improves the performance of our method from 74.8% to 89.7% with respectively one and four resolutions. Moreover, one could expect the classification success rate to dramatically drop with the rotation invariance since crucial information about the direction is lost. However, it is interesting to point out that we only measure a drop of 2.3% and 1.6% with respectively four and one resolutions.

number of resolutions	1	2	3	4
normal	74.8%	84.4%	87.4%	89.7%
rotation invariant	73.2%	82.5%	85.6%	87.4%

Table 2: Global classification results of our method on Dyntex ++ using one to four resolution images (starting from the original image and adding lower resolutions with the Gaussian pyramid).

Method	Classification rate
DL-PEGASOS [10]	63.7%
Extended Plots - 4 resolutions (our method)	89.7%
PCA-cLBP/PI-LBP/PD-LBP+super histogram + RBF SVM [19]	92.4%

Table 3: Classification results comparison with the state of the art on Dyntex++.

Finally, it should be noted that some sequences are misclassified into classes with very similar motion. For instance, eight sequences of the class 27 "leaves on branches swaying with wind" are misclassified into the class 33 ("branches swaying in wind (no leaves)". Adding a spatial analysis to this method would overcome this issue.

4 Combined Co-occurrence Matrix of Optical Flow (CCMOF)

Introduction As shown in section 3, the co-occurrence matrix approach developed in [12, 13] can be used to characterise the distribution of the OF. However, creating a co-occurrence matrix from a vector image is not as straightforward as the classic GLCM extracted from pixel intensities. Indeed, the occurrence of two values must be taken into account such as the x and y components of the flow vectors or their magnitudes and directions. This is why the EDPs and EMPs were used in [16]. However, analysing the two components separately gives rise to a loss of information carried by the joint distribution. Furthermore, the quantisation of the magnitude is not as simple as the quantisation of a grey-scale image or an orientation image. Grey-scale images and orientation images are respectively defined in the bounded intervals [0,255] and [-180,180], whereas the magnitude is only limited by the size of the image. In [16] and section 3, flow vectors with a magnitude larger than four pixels were considered as noise which is not the case in many DT sequences. Extending the quantisation to the maximum magnitude that a correct flow vector can have (maximum magnitude of all the sequences, up to 25 pixels length) is not a convenient solution either. In order to overcome these issues, a new framework combining the magnitude and the orientation of the flow is developed using individual quantisation levels of the magnitudes for each sequence.

Method description The framework of our CCMOF method is illustrated in Figure 2. The OF is calculated between every two consecutive frames of the DT sequence, using the algorithm from [22]. A sequence of 100 frames thus results in 99 motion vector images. In the second block, the magnitude and the orientation of the OF are calculated from the x and y values of the motion vectors. Subsequently, the magnitude and the orientation are jointly quantised. As shown in Figure 3, eight bins of magnitude and eight bins of orientation are used,

 ISBN 978-0-9934207-0-2

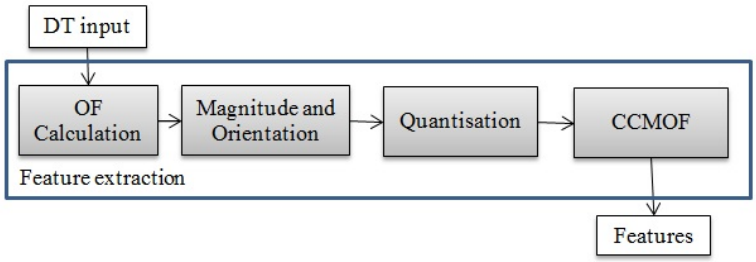


Figure 2: Feature extraction diagram of the CCMOF.

resulting in 64 bins spanning the OF vector values. For each sequence, the magnitude is linearly quantized in eight bins in the ranges $[0, M_s]$ ($[0, \frac{1}{8}M_s], [\frac{1}{8}M_s, \frac{2}{8}M_s], \ldots, [\frac{7}{8}M_s, M_s]$), where M_s is the largest magnitude of the sequence after removing large outliers which will not contribute to the CCMOFs, using Peirce's criterion. Finally, the co-occurrence of the neighbouring flow vectors is calculated, resulting in three 64 by 64 CCMOFs; two for the spatial, one for the temporal domain. The matrices extracted in the space domain summarise the occurrence of the pairs of neighbours on the x and y axes. This co-occurrence is calculated on every frame, then summed over the entire sequence and normalised. In the time domain, the same process is applied with neighbours on the temporal axis. These co-occurrence matrices differ from the classic GLCM as they combine two dimensions. The definition of new features is necessary in order to extract the distributions of the magnitude and of the orientation as well as their joint distribution. In total, 19 features are calculated based on Haralick's features [13]; 11 in the spatial domain and eight in the temporal domain: The *Energy*, *Contrast*, *Orientation Contrast*, *Magnitude Contrast*, *Correlation*, *Sum of Squares*, *Homogeneity* and *Entropy* are extracted from both the spatial and temporal CCMOFs. The *Dominant Orientation*, *Dominant Magnitude* and *Weighted Dominant Orientation* are calculated only in the spatial domain. We need to slightly modify the calculation of the Haralick's features [13] regarding the difference n between neighbours. In Haralick's features, the difference between neighbours n is defined as Equation (1) and represents the variation in the neighbours' intensities.

$$n = |i - j|, \quad n \in \{q_1, q_2, \ldots, q_N\}, \tag{1}$$

where $i, j \in \{q_1, q_2, \ldots, q_N\}$ represent the quantised values of the pair of neighbours and $q_1, q_2, \ldots, q_N$ are the N discrete grey levels of quantization. In our case, n is chosen as the Manhattan distance between neighbour vectors (Equation (2)).

$$n = |\hat{M}_i - \hat{M}_j| + |\hat{O}_i - \hat{O}_j| \mod 4, \quad n \in \{0;1;2;\ldots;11\}, \tag{2}$$

where $\hat{M}_i, \hat{M}_j$ are the quantised magnitudes of the neighbour vectors and $\hat{O}_i, \hat{O}_j$ their orientations. On Figure 3, the distance between the two vectors is $n = 2$. Finally, it was observed that the smallest magnitudes in a vector image are often due to noise on a static part of the DT. For instance, in a traffic DT, OF vectors are calculated on the supposedly static road with magnitudes close to zero. Their orientation is therefore not relevant and reduces the discriminative power of the features calculated from the CCMOFs. Therefore, the same 19 features are calculated from the CCMOFs in which the bin corresponding to the smallest magnitudes is removed (resulting in 7 by 8 matrices). A total of 38 features is thus extracted from each DT sequence.
DT recognition, as proposed in [19], should largely rely on appearance analysis as it contains the most discriminative information. Therefore, we combine our motion features with LBP-TOP features [23]. Thus, we cover the analysis of the motion (CCMOF) in the spatiotemporal domain as well as the pixel intensities' distribution (LBP-TOP) in the spatiotemporal domain.

Results and discussion A sequential feature selection is applied in order to determine those which discriminate the best between classes and to remove irrelevant and redundant features.We use a K-Nearest Neighbors (K-NN) classifier with $K = 1$. A distance $d = 1$ pixel between neighbours is chosen for the construction of the co-occurrence matrices. For the testing of this method, a dataset is created using sequences from the Dyntex database. The current benchmarks (Dyntex, Dyntex++, UCLA) suffer, in our point of view, several drawbacks.

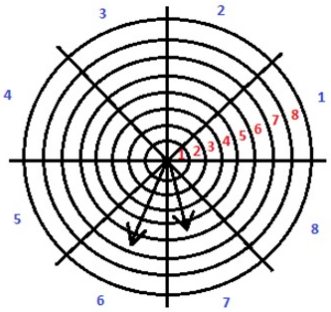


Figure 3: OF quantisation with two vector examples ($\hat{M} = 7$, $\hat{O} = 6$ (left); $\hat{M} = 6$, $\hat{O} = 7$ (right)).

We sought to rectify those by focusing on videos which exhibit only one DT and contain the same dynamic on the entire space-time domain. We also select sequences which depict different scenes, from various viewpoints. The resulting dataset contains 10 classes of DTs, each with 20 sequences of size 100*100*50. The experimental setup is the same as in section 3. 10 sequences are randomly selected from each class for the training set. The other 10 sequences are used for testing. The confusion matrix presenting the classification results using the CCMOF features in combination with LBP-TOP is shown in Figure 4 and the overall classification results are presented in Table 4. One can notice that the main misclassifications occur for very similar classes ("water"/"waves" and "foliage"/"trees") which share similar dynamics and spatial texture.

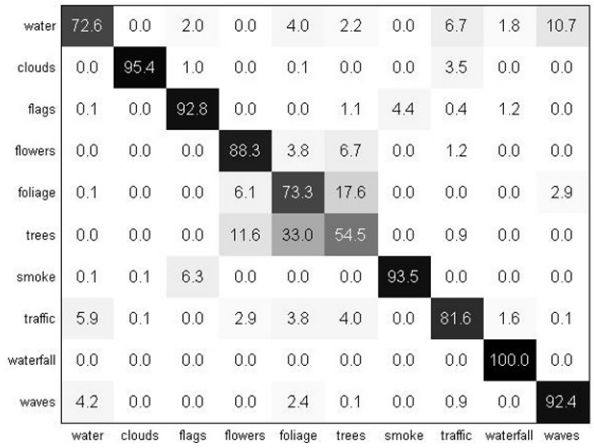


Figure 4: Confusion matrix of CCMOF + LBP-TOP on the developed database.

Method	Classification rate
CCMOF	68.0%
LBP-TOP [23]	81.7%
LBP-TOP + CCMOF	84.4%

Table 4: Classification results of CCMOF and LBP-TOP on the developed database.

The motion analysis using the CCMOF approach significantly improves the classification of our dataset when combined with the analysis of the spatiotemporal distribution of the pixel intensities (LBP-TOP). Solely analysing the motion is not sufficient to obtain a classification as accurate as the literature. It confirms that a DT recognition system mostly relies on the spatial texture analysis [19] and the motion analysis provides complementary information that can improve the performance.

5 Conclusion

In this paper, we investigated the analysis of the OF in the spatiotemporal domain and the combination with complementary spatiotemporal features based on pixel intensities' distribution. Section 3 showed a simple use of the OF for DT classification with co-occurrence matrices as well as the potential of a temporal multiresolution analysis of the motion. Section 4 presented good results obtained with our new method CCMOF. In particular it showed the importance of the analysis of the motion in the spatiotemporal domain, the combination of an appearance and motion analysis as well as the joint analysis of the magnitude and orientation of the motion. Our CCMOF approach greatly improves the state of the art on the developed database when combined with appearance analysis (LBP-TOP).

References

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *JOSA A*, 2:284–299, 1985.

[2] T. Boujiha, J.-G. Postaire, A. Sbihi, and A. Mouradi. New approach for dynamic textures discrimination. pages 1–4, 2010.

[3] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2:27, 2011.

[4] J. Chen, G. Zhao, M. Salo, E. Rahtu, and M. Pietikainen. Automatic dynamic texture segmentation using local descriptors and optical flow. *Image Processing, IEEE Transactions on*, 22:326–339, 2013.

[5] K. G. Derpanis and R. P. Wildes. Dynamic texture recognition based on distributions of spacetime oriented structure. pages 191–198, 2010.

[6] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51:91–109, 2003.

[7] S. Fazekas and D. Chetverikov. Dynamic texture recognition using optical flow features and temporal periodicity. pages 25–32, 2007.

[8] A. B. Flores, L. A. Robles, R. M. M. Tepalt, and J. D. C. Aragon. Identifying precursory cancer lesions using temporal texture analysis. pages 34–39, 2005.

[9] P. Gao and C. L. Xu. Extended statistical landscape features for dynamic texture recognition. volume 4, pages 548–551, 2008.

[10] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. pages 223–236. 2010.

[11] W. N. Gonçalves, B. B. Machado, and O. M. Bruno. Spatiotemporal gabor filters: a new method for dynamic texture recognition. *arXiv preprint arXiv:1201.3612*, 2012.

[12] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67:786–804, 1979.

[13] R. M. Haralick, K. Shanmugam, and I. H. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, pages 610–621, 1973.

[14] Y. Hu, J. Carmona, and R. F. Murphy. Application of temporal texture features to automated analysis of protein subcellular locations in time series fluorescence microscope images. pages 1028–1031, 2006.

[15] Z. Lu, W. Xie, J. Pei, and J. Huang. Dynamic texture recognition by spatio-temporal multiresolution histograms. volume 2, pages 241–246, 2005.

[16] C.-H. Peh and L.-F. Cheong. Synergizing spatial and temporal texture. *Image Processing, IEEE Transactions on*, 11:1179–1191, 2002.

[17] Y.-l. Qiao and F.-s. Wang. Dynamic texture classification based on dual-tree complex wavelet transform. pages 823–826, 2011.

[18] A. Ravichandran, R. Chaudhry, and R. Vidal. View-invariant dynamic texture recognition using a bag of dynamical systems. pages 1651–1657, 2009.

[19] J. Ren, X. Jiang, and J. Yuan. Dynamic texture recognition using enhanced lbp features. pages 2400–2404, 2013.

[20] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto. Dynamic texture recognition. volume 2, pages II–58, 2001.

[21] L. C. Sincich and J. C. Horton. The circuitry of v1 and v2: integration of color, form, and motion. *Annu. Rev. Neurosci.*, 28:303–326, 2005.

[22] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. pages 2432–2439, 2010.

[23] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29:915–928, 2007.

[24] G. Zhao and M. Pietikäinen. Dynamic texture recognition using volume local binary patterns. pages 165–177. 2007.

Investigation into DCT Feature Selection for Visual Lip-Based Biometric Authentication

C Wright, D Stewart, P Miller, F Campbell-West

Centre for Secure Information Technologies (CSIT)
Queen's University Belfast
{cmclarnon03, Dw.Stewart, p.miller, f.h.campbellwest} @qub.ac.uk

Abstract

This paper investigated using lip movements as a behavioural biometric for person authentication. The system was trained, evaluated and tested using the XM2VTS dataset, following the Lausanne Protocol configuration II. Features were selected from the DCT coefficients of the greyscale lip image. This paper investigated the number of DCT coefficients selected, the selection process, and static and dynamic feature combinations. Using a Gaussian Mixture Model - Universal Background Model framework an Equal Error Rate of 2.20% was achieved during evaluation and on an unseen test set a False Acceptance Rate of 1.7% and False Rejection Rate of 3.0% was achieved. This compares favourably with face authentication results on the same dataset whilst not being susceptible to spoofing attacks.

Keywords: Authentication, Biometrics, GMM-UBM, XM2VTS, DCT

1 Introduction

It is widely recognised that passwords are not enough to use as a sole means of authentication, which has been made apparent by many high profile hacking cases. This has resulted in biometric authentication becoming increasingly popular. Physiological-based biometric systems have been incorporated into the most common mobile platforms, i.e. Android's Face unlock and iPhones fingerprint scanner, and have been hacked using replay attacks and spoofing [Racoma, 2012], [Kleinman, 2014]. Behavioural biometrics are potentially more difficult to crack but are also more complex to capture, model and authenticate robustly.

In this area 'Speaker Verification' is acknowledged as the ability to authenticate a person's claimed identity from their voice [Campbell, 1997]. Gaussian Mixture Model–Universal Background (GMM-UBM) systems are commonly used with speaker verification systems, [Hautamäki et al., 2015]. The set up involves using creating a GMM to model each individual, and another large GMM that represents the whole population – the UBM. When authenticating a person's claimed identity, a likelihood is calculated with respect to their individual model and another with respect to the UBM. A resulting score can be calculated using these likelihoods.

[Cetingul et al., 2006] researched lip motion features for speaker and speech recognition using Hidden Markov Models (HMMs). Features evaluated include dense motion features within a bounding box around the lips, and features created from lip shape (contours) and motion. The MVGL-AVD database consisting of 50 individuals was used. The best recorded result for speaker recognition was found during the cross validation of the system using motion features with an Equal Error Rate (EER) of 5.2%.

[Faraj and Bigun, 2006] investigated a combination of audio and visual features from the lips for person authentication. Experiments used Gaussian Mixture Models (GMM), the XM2VTS database and the Lausanne Protocol, configuration I. Results reported an EER of 22% on visual features alone.

Whilst lip features have shown promise in previous published work there has been no accompanying comparative results for other modalities on the same dataset, e.g. face. This paper investigates using the GMM-UBM framework to model lip movements as a behavioural biometric. The XM2VTS dataset and the Lausanne

 ISBN 978-0-9934207-0-2

Protocol, configuration II was followed [Luettin and Maître, 1998] in an attempt to rigorously benchmark this system and allow for comparison. Discrete Cosine Transform (DCT) coefficients of greyscale lip images were used as visual lip features. As with existing speaker verification systems a likelihood value was calculated using both models of the claimed individual and the UBM.

2 DCT Features

The DCT coefficients were chosen as they can capture lip appearance in a compact form. Investigating gender recognition [Stewart et al., 2013] used DCT coefficients as a feature, the positive results demonstrate that they captured speaker specific appearance and dynamics. However there has been no rigorous investigation of DCT-based features for speaker authentication which is the aim of this work.

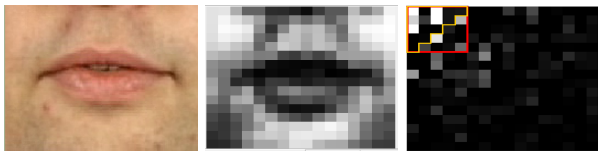


Figure 1: Individual 0, Session 1, video 1, frame 1. From left to right: Pre-processed cropped image, Grey scale resized image, DCT coefficients of the frame with both square and triangular mask

Although we are aware that DCT coefficients can be useful for modelling lip appearance, we do not know how many DCT coefficients are required to effectively model identities and we do not know the relative utility of static DCT coefficients compared to their derivative features. Furthermore when extracting DCT coefficients as features from the full DCT coefficient matrix, two common masks can be applied, square or triangular. It has been shown for lip-based speech recognition that a triangular mask offers better performance, [Stewart et al., 2013], and we will seek to establish if the same is true when modelling identities.

Figure 1 shows how the video data is processed during these experiments. The first image shows the cropped Region Of Interest (ROI). Next each frame is converted to grey scale. After histogram equalisation the frame is resized to a 16 by 16 pixel image, this is shown in the second image in figure 1. The third image shows a visual representation of the 2D DCT coefficients of the frame overlaid with both square and triangular masks.

The masks were used to extract the required number of DCT coefficients, k. The extracted DCT coefficients of the j^{th} frame are represented by $\mathbf{x}_j$, a $k \times 1$ column vector. The frames are stacked to make a $k \times n$ matrix, where n is the number of frames in a video, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. Normalisation was used to reduce the effects of inter-session variability using:

$$\overline{x}_j^i = \frac{(x_j^i - \mu^i)}{\sigma^i}, \tag{1}$$

where μ^i is the mean of the i^{th} DCT coefficient across all frames, similarly σ^i is the standard deviation. The resulting normalised feature for the entire video input is therefore: $\overline{\mathbf{X}} = \{\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, ..., \overline{\mathbf{x}}_n\}$. The same steps were used to prepare the videos for all steps in the training, evaluation and testing.

3 GMM Modelling

GMMs were used to represent both the UBM and the individual models. During an attempted login the system will test the input against the claimed individual model and against the UBM. Using $\overline{\mathbf{X}}$, we want to compute how alike it is to the features that created a model, λ. The likelihood $p(\overline{\mathbf{X}}|\lambda)$, is calculated using:

$$p(\overline{\mathbf{X}}|\lambda) = \prod_{j=1}^{n} \sum_{i=1}^{M} \omega_i \; p_i\left(\overline{\mathbf{x}}, \; j\right) \tag{2}$$

where M is the number of unimodal gaussians, n is the number of frames, $\overline{\mathbf{x}}_j$ is the normalised j^{th} frame, and the mixture weights, ω_i must satisfy the constraint $\sum_{i=1}^{M} \omega_i = 1$. During training the objective is to maximise $p(\overline{\mathbf{X}}|\lambda)$, where $p_i\left(\overline{\mathbf{x}}_j\right)$ is the likelihood of the j^{th} frame to the i^{th} unimodal gaussian, and the i^{th} unimodal gaussian is parameterised by a mean, $\boldsymbol{\mu}_i$, and a covariance matrix $\boldsymbol{\Sigma}_i$ as described in 3:

 ISBN 978-0-9934207-0-2

$$p_i\left(\overline{\mathbf{x}}_j\right) = \frac{1}{(2\pi)^{\frac{k}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}\left(\overline{\mathbf{x}}_j - \boldsymbol{\mu}_i\right)^T(\boldsymbol{\Sigma}_i)^{-1}\left(\overline{\mathbf{x}}_j - \boldsymbol{\mu}_i\right)\right\} \tag{3}$$

4 Classification

We want to compute the likelihood that the feature extracted from the video input, $\overline{\mathbf{X}}$, was generated by the claimed identity, and the likelihood that the feature was not generated by the claimed identity. If we denote the likelihood of $\overline{\mathbf{X}}$ being generated by the claimed identity as $p(\overline{\mathbf{X}}\,|\lambda_{hyp})$, where λ_{hyp} represents the mean vector and covariance matrix parameters of the hypothesised Gaussian model, and the likelihood of $\overline{\mathbf{X}}$ of being generated by anybody else as $p(\overline{\mathbf{X}}|\lambda_{UBM})$, where λ_{UBM} represents the mean vector and covariance matrix parameters of the UBM Gaussian model. Then a log-likelihood ratio can be calculated using equation 4:

$$\Lambda(\overline{\mathbf{X}}) = \log\ p(\overline{\mathbf{X}}\,|\lambda_{hyp}) - \log p(\overline{\mathbf{X}}|\lambda_{UBM}) \tag{4}$$

The log-likelihood generated from equation 4 can then be tested against the threshold and the identity accepted or rejected as shown in the modular diagram in figure 2.

5 Experiments

The aim of these experiments was to investigate the feature representation that produced the lowest EER when varying:

- The mask type used to select the number of DCT coefficients. The right most image in figure 1 shows both triangular and square masks.
- The number of DCT coefficients. Square masks were tested from a 3 by 3 mask producing a feature vector containing 9 DCT coefficients to a 7 by 7 mask producing 49 DCT coefficients. For the triangular masks the range went from a 4 by 4 mask producing 10 DCT coefficients to a 9 by 9 mask producing 45 DCT coefficients.
- The 'type' of feature, ie. Static / Dynamic. This work looked into the performance of static, dynamic and combinations of both to help find the optimum feature representation.

The dynamic features included the first and second order derivatives of the static DCT coefficients with respect to time, known as delta, Δ, and deltadelta, $\Delta\Delta$, features. After testing the features separately, all combinations of the 3 features were tested. The features are combined by concatenating the feature vectors. For example if 15 DCT coefficients had been selected, when combining static and Δ, the $\Delta\Delta$, DCT coefficients were concatenated after the static making a total of 30 DCT coefficients.

5.1 System Overview

Figure 2 shows a modular diagram showing how the system was used during testing. After a video was read in the features were extracted and normalised as described in section 2. The features are used to get a log-likelihood from the claimed GMM and the UBM as described in section 3, and a log-likelihood ratio was obtained using equation 4. The ratio log-likelihood will be either accepted or rejected based on the threshold set during the evaluation stage. This is how the system would be used in deployment.

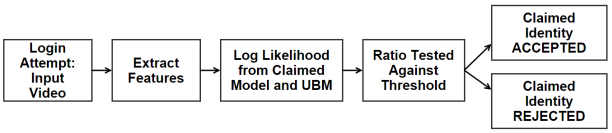


Figure 2: General Modular Diagram of the System

5.2 Database and Protocol

Experiments were carried out on the XM2VTS [Messer et al., 1999]. The XM2VTS is a large audio-visual database containing video recordings of 295 individuals during 4 sessions. Each session contains 2 videos per person and the sessions were recorded over 4 months. For these experiments only digit sequences spoken during recording sessions were used. Pre-processed video data was also used as the audio was removed and the video was cropped to only contain mouth region, for preprocessing steps see [Seymour et al., 2008].

For these experiments the Lausanne Protocol [Luettin and Maître, 1998], configuration II was strictly followed. The Lausanne Protocol is a closed-set verification protocol [Bourlai et al., 2005], because the population of clients does not change the system does not need to account for new users in the evaluation and testing. As shown in figure 3, the protocol divides the dataset into Training, Evaluation and Test data.

- The Training data consists of video from the first 2 sessions for 200 individuals.
- The Evaluation data consists of video data from the third session for the same 200 individuals. Plus all video data for a separate 25 individuals not used in training. See section 5.4 for more details on how these videos are used to represent returning clients and imposters.
- The Test data is made up of the 2 videos from the fourth session of the 200 individuals used for training, plus all video data available for a separate 70 individuals not used in the training.

Data from 3 individuals was removed from our tests as some vidos were found to be corrupt. The IDs of the individuals removed are 342, 272 and 313. Figure 3 shows the number of individuals used in these experiments after the corrupt videos were removed.

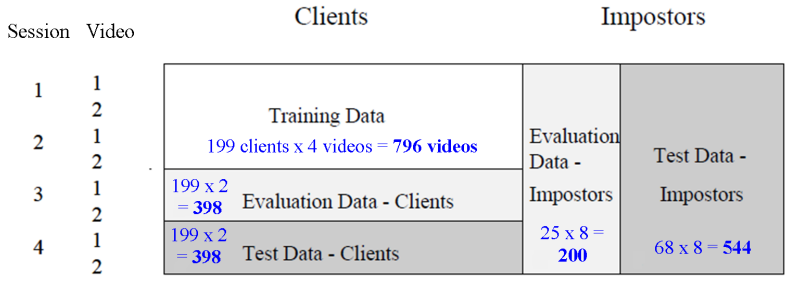


Figure 3: Partitioning of the XM2VTS database according to the protocol Configuration II

5.3 Training

The UBM was trained with all the designated Training data as specified in figure 3, 796 videos from 4 individuals. General guidelines for unconstrained speech suggest 512-2048 Mixtures for the UBM, where lower-order mixtures are more common with constrained speech such as digits and fixed vocabulary [Bimbot et al., 2004]. For the experiments in this work all UBMs were trained with 256 mixtures as all video data contains digits. Individual GMMs were created for each of the 199 individuals in the training data and each model was created using 32 mixtures, likewise 32 mixtures has been used in speaker recognition for individual models, [Stewart et al., 2013].

5.4 Evaluation

Evaluation was carried out in order to select a threshold before running the system on the unseen Test data, using the Evaluation data specified in figure 3. All 598 videos were tested against all 199 individual models, producing $598 \times 199 = 118,604$ attempted logins, with 398 registered user attempts and 118,206 imposter attempts.

Kevin Murphy's toolbox [Murphy, 2001] was used to create the GMM's and retrieve the log-likelihoods. System performance on the Evaluation data was measured by calculating the False Acceptance Rate (FAR- Imposters who can login as another) and the False Rejection Rate (FRR - individuals who cannot in as themselves) and using this to calculate the EER. The EER is the point when the FAR = FRR. The threshold for this system was set to the EER.

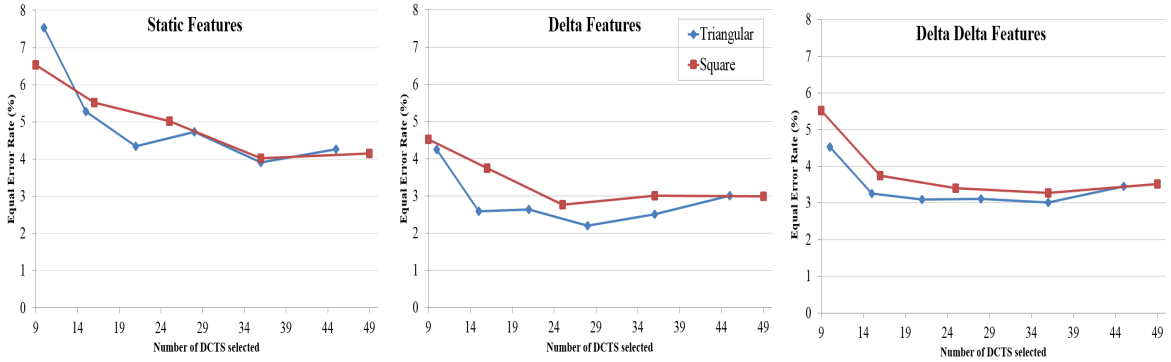


Figure 4: Evaluation results for Static, Δ, ΔΔ features.

The graphs in figure 4 show the EER against the number of DCT coefficients. The graphs show the static, Δ and ΔΔ features. It can be seen that the triangular feature selection outperforms the square feature selection. All 3 graphs show that as the number of DCT coefficients increases the EER is reduced, and it appears that no additional information is gained by using more than 28 DCT coefficients. It can also be seen that the Static features produce the highest EER, and the Δ features produced the lowest EER.

	No. DCT coefficients			
Features	15	21	28	36
Static	5.28	4.34	4.73	3.91
Δ	2.59	2.63	**2.20**	2.51
ΔΔ	3.25	3.09	3.14	3.02
Static & Δ	3.59	3.27	3.27	3.52
Static & ΔΔ	4.02	3.98	4.02	3.52
Δ & ΔΔ	3.10	3.52	2.94	3.27
Static, Δ & ΔΔ	3.02	3.69	3.52	3.94

Table 1: Equal Error Rate (%) on Evaluation set: Showing highest performing number of DCT coefficients, selected used a triangular mask.

Following this, experiments were then run to test combinations of static, Δ and ΔΔ features using triangular feature selection and 15, 21, 28 and 36 DCT coefficients. Combining the static and dynamic information did not appear to add additional information to the features as the Δ alone produced the lowest EER, 2.20%. Results can be seen in table 1.

5.5 Testing

From the evaluation, the set up producing the lowest EER was found to be triangular features using 28 Δ DCT coefficients. The optimum threshold for this system was then calculated based on the EER and applied for testing the unseen data. Before running the unseen Test data on the system using the threshold calculated, practice dictates that the system is retrained using both the Training and Evaluation data [Hastie et al., 2009]. These experiments investigated both this practice and running the unseen Test data on the system without retraining. In theory a system would be trained with all available data before deployment and a threshold calculated based on the data it was trained on. If we retrain the models with the Evaluation data it would be expected the threshold calculated in the evaluation would no longer be optimum therefore the unseen data would be expected to not perform as well.

The 942 videos were tested against all the 199 individual models. This produced $942 \times 199 = 187,458$ attempted logins, with 398 registered user attempted logins and 187,060 imposter attacks. The results for this set up can seen in the top row in table 2.

	FRR	FAR
Models Not Retrained	3.02%	1.68%
Models Retrained	1.76%	4.21%

Table 2: False Rejection Rate (FRR) & False Acceptance Rate (FAR) for unseen test data.

Note the performance (in terms of both FAR and FRR) from evaluation for these same models with the same operating threshold was 2.20%. As seen in table 2, a FRR of 3.02% and FAR of 1.67% was obtained.

On the bottom row of table 2 we can see the results for the Test data after evaluation, when the models have been retrained to contain all the Training and Evaluation data. The table shows a FRR of 1.76% and a FAR of 4.21%.

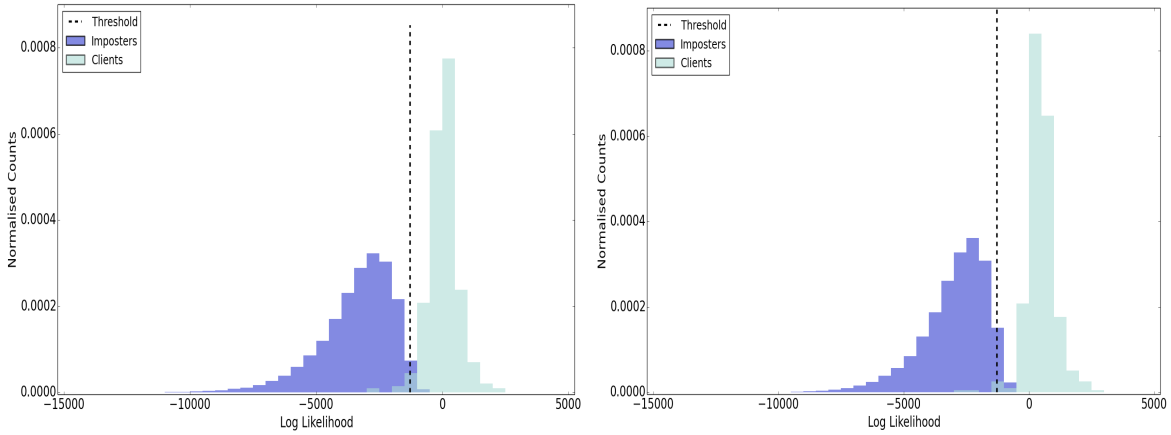


Figure 5: Histogram showing the Client & Imposter Log Likelihoods. Left to right: Not retrained, Retrained

The image on the left in figure 5 shows a histogram of the normalised log-likelihoods of the Test data and the threshold is marked on with a dashed line. There is small overlap in clients and imposters around the threshold. We can see that no matter what threshold was chosen in this experimental setup there will always be FAR or FRR, but the chosen threshold does appear to minimise both the FAR and FRR for unseen data. The image on the right in the figure shows the normalised log-likelihoods for the retrained setup. By comparing the histograms in figure 5 we can see how the threshold set for models trained with less data is no longer optimum with increased training data. This means that if the model is retrained with more data a new threshold should be calculated as in the evaluation stage of these experiments. The overlap of imposters and clients also appears to have reduced in this histogram, indicating that the system improves with more Training data.

As the threshold was set during evaluation, the top row in table 2 shows the true results on unseen data after training and evaluation. These results show even with limited Training data the system successfully authenticated 96.98% of registered clients and successfully prevented access to 98.32% of the imposters.

These results compare very favourably with previous lip based authentication results even though some were on much smaller datasets, [Cetingul et al., 2006]. On the Evaluation data we achieved an EER of 2.2%, producing a predicted performance of 97.8%. This is an improvement on [Cetingul et al., 2006] et al who recorded an error rate of 5.2%. Faraj et al [Faraj and Bigun, 2006] recorded a performance of 78% for the visual features on their own. The EER was used to calculate a threshold which was used to test unseen Test data, this gives a more accurate result on how the system would work in a deployment scenario. These tests produced a FAR of 1.7% and a FRR of 3.0%.

The DCT-based features compare well with results recorded in [Bhattacharjee and Sarmah, 2012], where a 4.55% EER was achieved with a GMM-UBM system and audio features for authentication.

The performance of the system using these features also compares very well with the face recognition system by [Brady et al., 2007] who recorded an error rate of 2.5% on the Evaluation data using the same database and protocol, and the facial recognition system presented by IDIAP in [Messer et al., 2003]. IDIAP used a GMM system and DCT features of the full face image and achieved an EER of 2.45% on the Evaluation data

and on the Test set a FAR and FRR of 1.35% and 0.75% respectively.

Upon further analysis of the specific test cases which caused the FRR errors to occur. The 3.02% of FRR errors equated to 12 attempted logins from only 9 individuals from the 199 registered individuals in the system. Of these, only 3 individuals failed to be authenticated as themselves on both of their test videos. Therefore only 1.5% of individuals could not be authenticated successfully if at least two attempts were considered.

Figure 6 illustrates the data for the 3 problematic individuals. Upon close inspection, the most obvious reason for individual 79 not being authenticated appears to be inconsistent registration of the lip ROI which led to slight rotation of the Test dataset frames compared to the Training frames. Similarly, it is inconsistent ROI extraction which appears to have caused the error for individual 264. In this case the individuals facial hair may have caused the poor lip tracking. For individual 191 the error appears to be caused by a significant change in facial hair prior to the test.

Figure 6: From top down, individual 79, 191, 264. From left to right: Frames 1-4 are from each video included in model, Frames 5-6 are from each video that failed to login

6 Conclusion

This work provided a rigorous investigation of the effectiveness of DCT-based features for modelling speaker's lips within a GMM-UBM verification framework. In particular, we investigated the performance of different numbers of DCT coefficients, different selection of masks and different DCT-based feature types. The types included the static DCT coefficients and their first and second order derivatives known as Δ and $\Delta\Delta$ features. The largest available dataset for such experiments was used, including 292 individuals, namely the XM2VTS database along with the robust Lausanne Protocol configuration II. We showed for the first time that:

- Δ features produced the best feature representation over static, $\Delta\Delta$ and multiple combinations
- 28 DCT coefficients were found to be optimal for the feature
- Triangular mask used in feature selection is better than a square mask

On the Evaluation set an EER of 2.2% was obtained, producing a predicted performance of 97.8%. The EER was used to calculate a threshold which was used to test unseen data, this gives amore accurate result on how the system would work in a deployment scenario. These tests produced a FAR of 1.7% and a FRR of 3.0%.

These results compare very favourably with previous works in verification and authentication using alternative features and models, and compared to facial recognition systems using the same database and protocol.

Our analysis of the errors indicates that the system performance can be affected by poor and inconsistent lip ROI tracking. We will be investigating this further in our future work.

References

[Bhattacharjee and Sarmah, 2012] Bhattacharjee, U. and Sarmah, K. (2012). Gmm-ubm based speaker verification in multilingual environments. *Internation Journal of Computer Science*.

[Bimbot et al., 2004] Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-García, J., Petrovska-Delacrétaz, D., and Reynolds, D. A. (2004). A tutorial on text-independent speaker verification. *EURASIP J. Appl. Signal Process.*, 2004:430–451.

[Bourlai et al., 2005] Bourlai, T., Messer, K., and Kittler, J. (2005). Scenario based performance optimisation in face verification using smart cards. In *Audio-and Video-Based Biometric Person Authentication*, pages 289–300. Springer.

[Brady et al., 2007] Brady, K., Brandstein, M., Quatieri, T., and Dunn, B. (2007). An evaluation of audio-visual person recognition on the xm2vts corpus using the lausanne protocols. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–237. IEEE.

[Campbell, 1997] Campbell, J.P., J. (1997). Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9):1437–1462.

[Cetingul et al., 2006] Cetingul, H. E., Yemez, Y., Erzin, E., and Tekalp, A. M. (2006). Discriminative analysis of lip motion features for speaker identification and speech-reading. *Trans. Img. Proc.*, 15(10):2879–2891.

[Faraj and Bigun, 2006] Faraj, M. and Bigun, J. (2006). Motion features from lip movement for person authentication. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1059–1062.

[Hastie et al., 2009] Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. (2009). *The elements of statistical learning : data mining, inference, and prediction*. Springer series in statistics. Springer, New York. Autres impressions : 2011 (corr.), 2013 (7e corr.).

[Hautamäki et al., 2015] Hautamäki, R. G., Kinnunen, T., Hautamäki, V., and Laukkanenn, A.-M. (2015). Automatic versus human speaker verification: The case of voice mimicry. *Speech Communication*.

[Kleinman, 2014] Kleinman, Z. (2014). Politician's fingerprint 'cloned from photos' by hacker. http://www.bbc.co.uk/news/technology-30623611.

[Luettin and Maître, 1998] Luettin, J. and Maître, G. (1998). Evaluation protocol for the extended M2VTS database (XM2VTSDB). Idiap-Com Idiap-Com-05-1998, IDIAP.

[Messer et al., 2003] Messer, K., Kittler, J., Sadeghi, M., Marcel, S., Marcel, C., Bengio, S., Cardinaux, F., Sanderson, C., Czyz, J., Vandendorpe, L., Srisuk, S., Petrou, M., Kurutach, W., Kadyrov, A., Paredes, R., Kadyrov, E., Kepenekci, B., Tek, F., Akar, G. B., Mavity, N., and Deravi, F. (2003). Face verification competition on the xm2vts database. In *In 4th Int. Conf. Audio and Video Based Biometric Person Authentication*, pages 964–974.

[Messer et al., 1999] Messer, K., Matas, J., Kittler, J., and Jonsson, K. (1999). Xm2vtsdb: The extended m2vts database. In *In Second International Conference on Audio and Video-based Biometric Person Authentication*, pages 72–77.

[Murphy, 2001] Murphy, K. P. (2001). The bayes net toolbox for matlab. In *Computing Science and Statistics*.

[Racoma, 2012] Racoma, J. A. (2012). Android jelly bean face unlock 'liveness' check easily hacked with photo editing. http://www.androidauthority.com/android-jelly-bean-face-unlock-blink-hacking-105556.

[Seymour et al., 2008] Seymour, R., Stewart, D., and Ming, J. (2008). Comparison of image transform-based features for visual speech recognition in clean and corrupted videos. *J. Image Video Process.*, 2008:14:1–14:9.

[Stewart et al., 2013] Stewart, D., Pass, A., and Zhang, J. (2013). Gender classification via lips: Static and dynamic features. *IET Biometrics*, 2(1):28–34.

3D Reconstruction of Reflective Spherical Surfaces from Multiple Images

Abdullah Bulbul, Mairead Grogan & Rozenn Dahyot

School of Computer Science and Statistics
Trinity College Dublin, Ireland
{bulbulm, mgrogan, Rozenn.Dahyot}@tcd.ie

Abstract

Despite the recent advances in 3D reconstruction from images, the state of the art methods fail to accurately reconstruct objects with reflective materials. The underlying reason for this inaccuracy is that the detected image features belong to the reflected scene instead of the reconstructed object and do not lie on the surface of the object. In this study, we propose a method to refine the 3D reconstruction of reflective convex surfaces. This method utilizes the geometrical distortion of the reflected scenes behind a spherical surface.

Keywords: 3D reconstruction, Shape from images, Hough Transform, Specular surface

1 Introduction

Creating 3D worlds in the form of meshes that can be efficiently manipulated by engines has found many applications from computer games and virtual reality, to virtual museums visits. To assist artists in the creation of 3D meshes of real existing objects, many software systems have been developed using RGB or RGB-D images and videos.

In this paper we have created a dataset of images captured around a sculpture made of specular material, which is composed of several spherical elements (cf. Fig. 1). Using this dataset we propose to infer a 3D model of the sculpture by using several image processing routines. First a 3D point cloud is computed using the set of images. Each element of this point cloud has vertex, colour RGB and normal information. From this information we propose a Hough Transform like process to infer the centers and radii of the spheres. Such compact information allows us to easily generate a 3D representation convenient to use in a virtual environment such as Metropolis [O'Sullivan, 2010].

Figure 1: Walton Sculpture (Trinity college Dublin Ireland).

2 State of the Art

2.1 3D reconstruction

Reconstructing an unknown 3D geometry from multiple images is fundamentally based on finding corresponding features in different images and solving the correspondence problem in 3D. This is what the human visual

 ISBN 978-0-9934207-0-2

system does to perceive the world in 3D using triangulation based depth cues, e.g, stereo vision, motion parallax. In computer graphics and vision literature this procedure is called Structure from Motion (SfM). An earlier complete model of 3D reconstruction from unordered photos is given by Pollefeys et al. [Pollefeys et al., 2004]. In a well known study, Snavely et al. [Snavely et al., 2006] proposes a sparse 3D reconstruction method and improves the methods proposed in [Brown and Lowe, 2005] and [Hartley and Zisserman, 2004]. Using the sparse reconstruction output, Furukawa and Ponce generate dense 3D point clouds [Furukawa and Ponce, 2010]. Another similar study utilizes sparse 3D reconstructions to generate separate depth maps for each input image [Goesele et al., 2007]. Then these depth maps are used for dense 3D reconstruction.

Recently SfM studies have advanced in terms of the employed parallelism [Agarwal et al., 2011], performance [Frahm et al., 2010, Wu, 2013], which enables using a high number of input images, and symmetry detection [Cohen et al., 2012, Ceylan et al., 2014] which improves the accuracy of the outputs. All of these studies rely on corresponding feature matches among multiple images to determine a point in the 3D environment. However, for reflective surfaces these correspondences do not reside exactly on the surface of an object, which causes incorrect 3D reconstructions around the reflective surfaces. In order to refine reconstruction of scenes including planar reflective surfaces, Wanner and Goldluecke [Wanner and Goldluecke, 2013] propose a method that uses images from a regular structure of viewpoints. Another group of studies that work on reconstruction of reflective objects uses specular flow information [Roth and Black, 2006, Sankaranarayanan et al., 2010, Balzer et al., 2011]. In these studies, reflected objects are known and the way they are distorted is utilized for estimating surface geometry. In our case, neither the viewpoints are distributed regularly nor the reflected objects are known; therefore, none of these studies are directly applicable.

2.2 Sphere detection in point clouds

The Hough Transform is a standard popular technique for finding parametric shapes. It is a one-to-many mapping approach where each observation votes for multiple models (e.g. lines). This corresponds to computing a histogram in the latent space of the parameters describing the shape, and the bins collecting the most votes are deemed as potential candidate occurrences of the shape of interest. This technique can be difficult to use when dealing with noisy data and for exploring high dimensional latent spaces. To find spheres in 3D images, Cao et al have proposed a hierarchical Hough transform based technique by first finding circles in 2D image slices [Cao et al., 2006]. A sphere is then inferred by correlating circle parameters.

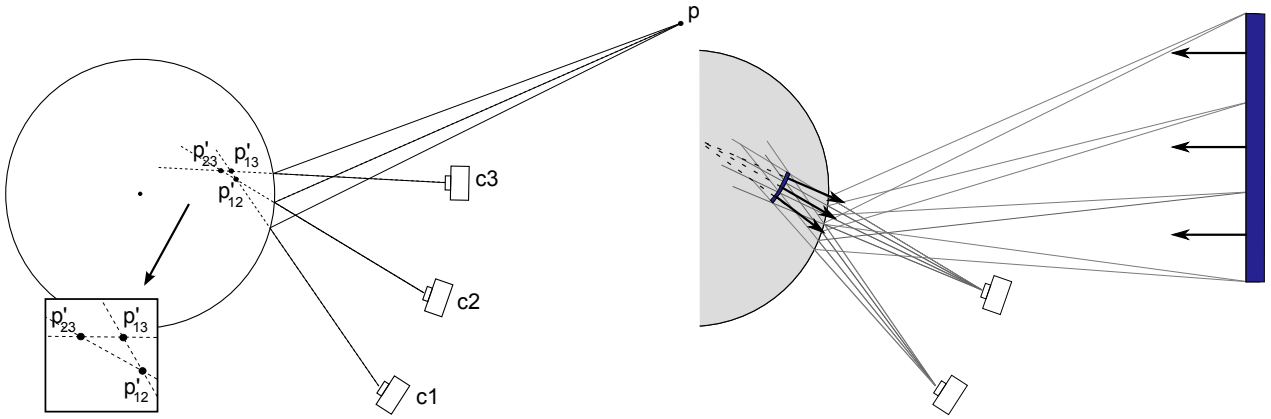
To alleviate the discrete histogram formulation of the Hough Transform, Dahyot proposed smooth kernel density estimates (KDE) to aggregate all votes for finding lines [Dahyot, 2009]. Moreover, when normal information extracted from gradient direction information of images is available, it can also easily be encapsulated in the KDE. However no extension of this KDE modeling to circle or sphere is available yet.

Schnabel et al. [Schnabel et al., 2007] proposed a many-to-one RANSAC based technique for finding multiple shapes (plane, sphere, cylinder, cone, torus) in 3D point clouds containing vertices augmented with normals. RANSAC is not designed to detect multiple occurrences of a shape and this limitation is overcome by using localised sampling. We propose here a global Hough based approach for finding multiple spheres occurring in noisy point clouds (vertices and normals) computed from images.

3 Geometry of Reflections in a Convex Surface

For a Lambertian surface, image features lie over the surface regardless of the viewer's position. Thus, it is possible to extract the 3D geometry of regions that show a Lambertian behaviour using multiple images taken from different viewpoints. This convenient property does not hold for reflective surfaces. A planar mirror is the simplest case of a reflective surface in which the reflected geometry is symmetrical over the surface. For curved surfaces, however, the reflected geometry does not stay at a static location for different viewpoints. Assuming a point p that is reflected over a convex surface, each pair of viewpoints indicates a different location for the reflected point p' (Figure 2-a).

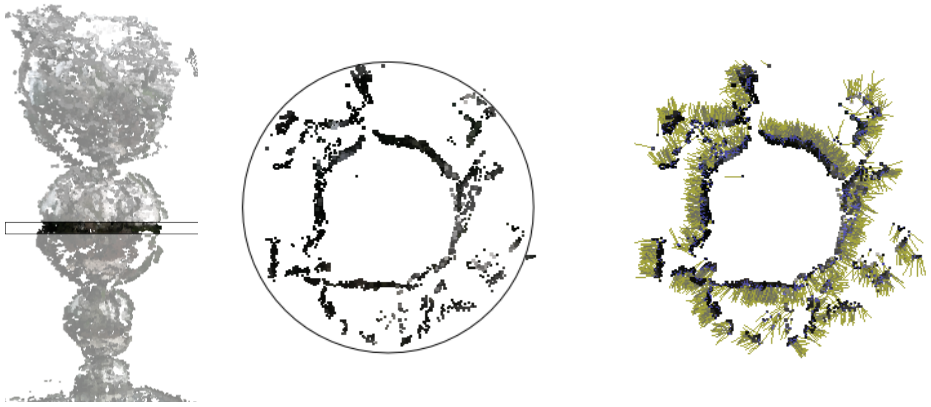
 ISBN 978-0-9934207-0-2



(a) (b)

Figure 2: (a) Pairwise matches $p'_{i,j}$ of a reflected point p among 3 different views. (b) Distortion of reflected shape and normals. Reflection of a straight feature is curved behind the spherical surface and the normal lines of the reflection converge.

Despite the lack of a stable reflection location, pairwise matches among viewpoints result in 3D reconstructions of the reflected geometry inside the sphere as shown in Figure 3. Having a pair of viewpoints, similarly with stereoscopic vision, reflections of a scene over a spherical surface appear behind the surface of a sphere and the geometry is curved (Figure 2-b). The generated vertices are distributed between the sphere surface and its focal distance ($r/2$) according to the actual distances of the reflected points.



(a) (b)

Figure 3: (a) A slice of the reconstructed point cloud belonging to the spherical surface shown on the left. (b) The same point cloud with per-vertex normals. Point clouds are generated by visualSFM [Wu, 2013] and the dense reconstruction method of [Furukawa and Ponce, 2010].

4 3D reconstruction from images

Our 3D reconstruction pipeline differs from a conventional one by including a sphere detection and refinement step (Figure 4), which is especially useful in the presence of reflective spherical surfaces. The pipeline starts with a sparse 3D reconstruction using the input images, including camera adjustments for each view. Then using the sparse reconstruction results and camera parameters the reconstruction is densified and per-vertex normals are estimated. Using the normal and position information, the resulting point cloud is refined by detecting and correcting spherical surfaces.

 ISBN 978-0-9934207-0-2

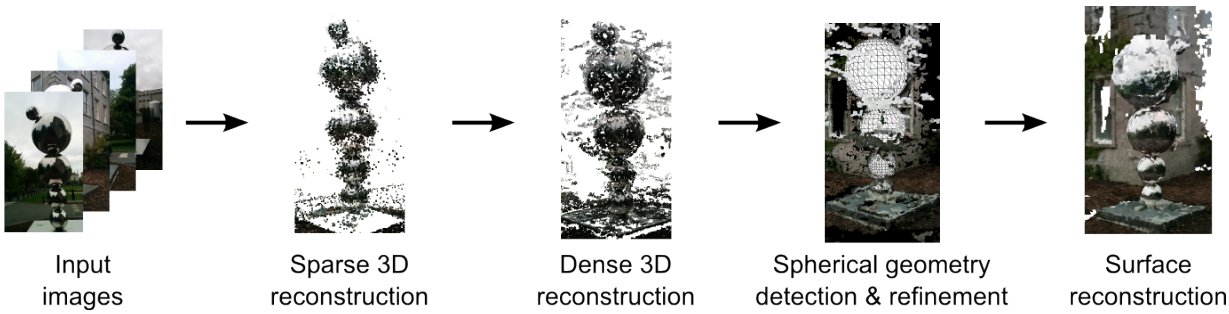
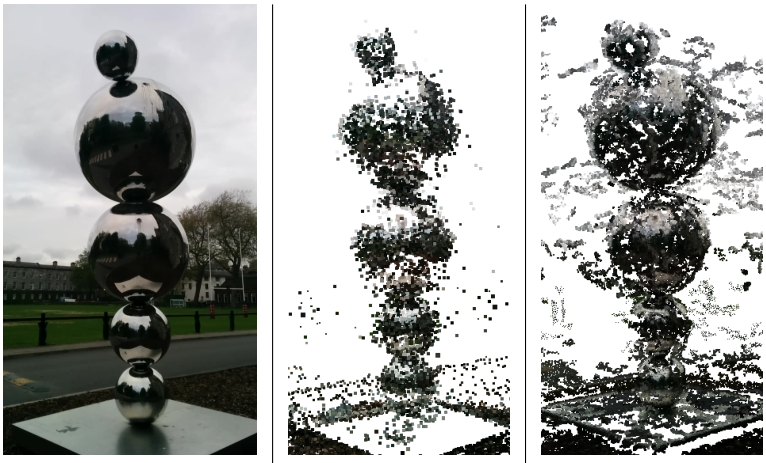


Figure 4: Overview of the proposed 3D reconstruction method.

4.1 Computation of the 3D point cloud

The proposed method works with an unordered set of images taken with a handheld camera or using the images from photo-sharing websites and social media. An SfM based method is used to generate a sparse 3D reconstruction using SIFT feature matches. There are several methods for sparse reconstruction and bundle adjustment such as the ones proposed by [Snavely et al., 2006, Lourakis and Argyros, 2009]. We have used the sparse reconstruction implementation in VisualSFM [Wu, 2013], which is a publicly available tool for 3D reconstruction.

Similarly, dense 3D reconstructions could be obtained by employing the methods of [Furukawa and Ponce, 2010] or [Goesele et al., 2007]. Both of them gave high quality dense reconstructions in our trials and we have used Furukawa and Ponce's method. The resulting sparse and dense reconstructions can be seen in Figure 5.



(a) (b) (c)

Figure 5: (a) One of the 122 input images, (b) sparse reconstruction, (c) dense reconstruction

4.2 Extracting Spheres Using Normal Correspondances

As explained in Section 3, per-vertex normals of a reconstructed spherical region with a reflective surface have a high probability of intersecting at locations inside the corresponding sphere. Therefore, we employ a special type of Hough Transform to determine these intersection points.

Finding candidates. Assuming $\mathcal{V}$ is the set of all vertices in the point cloud, each vertex $v_i \in \mathcal{V}$ and its corresponding normal n_i defines a line or ray in 3D space with the equation $v_i + t.n_i$ where t is a number in $\mathbb{R}$. Since it is unlikely to have lines which intersect exactly in the 3D space (See Figure 6), we expect the closest distance between a pair of lines to be smaller than a small threshold ϵ (set to 1/1024 of the diameter of the whole point cloud's bounding box), to be classified as intersecting. The distance between two

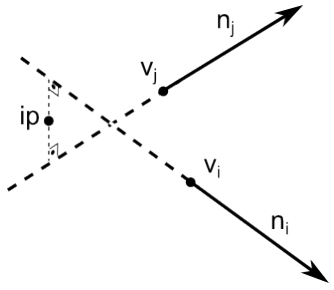


Figure 6: Intersection of two rays.

 ISBN 978-0-9934207-0-2

rays is calculated as follows:

$$d_{i,j} = \left| (v_i - v_j).\frac{n_i \times n_j}{\|n_i \times n_j\|} \right|. \tag{1}$$

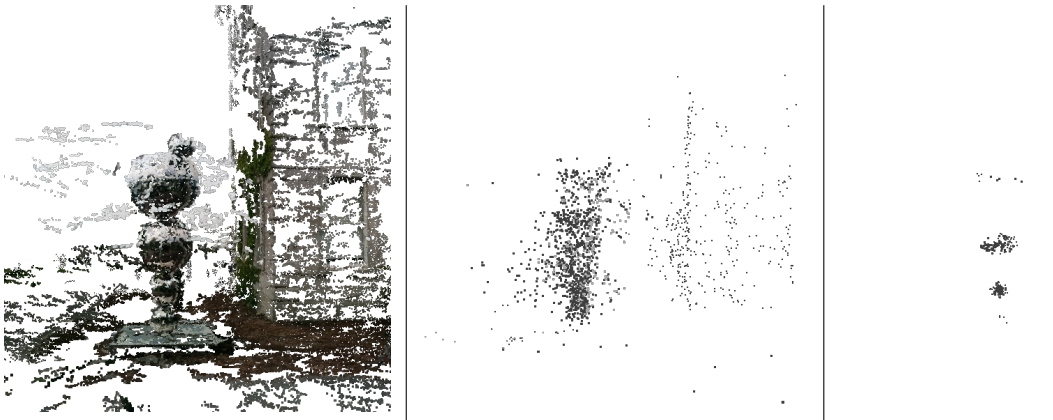
Another key property of a valid intersection point $\mathbf{i_p}$ is that the distances between $\mathbf{i_p}$ and the related vertices should be approximately the same, as indicated in Figure 2-b. Lastly, in order to avoid classifying very close and parallel vertices, which are very common in dense reconstruction of planar surfaces, as valid intersection points; a third rule is set. This rule enforces the valid line pairs to get closer to each other towards the intersection point. Simply, the following equation must hold:

$$\|v_i - v_j\| > k\,\epsilon, \tag{2}$$

where k is a parameter greater than 1 that adjusts the firmness of this rule. As we know that a valid intersection point has $d_{i,j} < \epsilon$, this rule ensures that we avoid parallel lines. A lower k value results in more candidate points and decreases performance, while a high value causes eliminating good candidates. We have empirically set k to 16. Combining these three rules, we define the set of all valid intersection points (IP) as follows:

$$IP = \left\{ (v_i, v_j) \in \mathcal{V}, \mathbf{i_p}_{i,j} \,|\, \left(d_{i,j} < \epsilon\right) \wedge \left(\|v_i - v_j\| > k\,\epsilon\right) \wedge \left(0.9 < \frac{\|v_i - \mathbf{i_p}_{i,j}\|}{\|v_j - \mathbf{i_p}_{i,j}\|} < 1.1\right) \right\}. \tag{3}$$

Voting procedure. After determining the candidate points, a voting procedure is performed with them. For that purpose, each line defined by v_i and n_i votes for a candidate point in IP if it passes through its ϵ proximity. The intersection points with the most votes are selected as the valid candidate centers. The median of the distances between an intersection point and all of its voters is assigned as the radius $r(\mathbf{i_p})$ of this intersection point $\mathbf{i_p}$ to be used in later steps. Figure 7 shows the set of intersection points and the selected intersections after the applied Hough Transform.



(a) (b) (c)

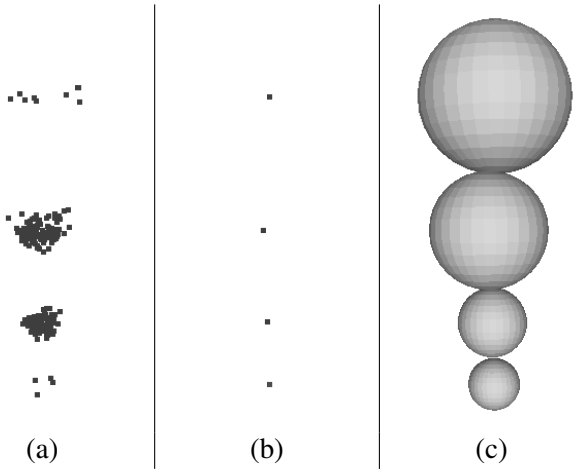
Figure 7: (a) Initial dense point cloud. (b) Candidate intersection points, IP. (c) Selected intersections after the voting procedure.

Merging close intersection points. As shown in Figures 2 and 3, there are multiple intersection points inside of a sphere due to the specular nature of the sphere's material. Normals belonging to different reflected objects intersect at different positions. Therefore, these intersection points are merged to get the final sphere locations. Two intersection points are merged if either of the intersection points lie inside the sphere formed by the other intersection point and its radius. When merging two intersection points $\mathbf{i_p}_i$ and $\mathbf{i_p}_j \in IP$, properties of the merged intersection point $\mathbf{i_p}_m$ are assigned as follows.

$$\begin{cases} \mathbf{i_p}_m & = w_i\ \mathbf{i_p}_i + w_j\ \mathbf{i_p}_j, \\ r(\mathbf{i_p}_m) & = w_i\ r(\mathbf{i_p}_i) + w_j\ r(\mathbf{i_p}_j) \\ \mathcal{V}(\mathbf{i_p}_m) & = \mathcal{V}(\mathbf{i_p}_i) \cup \mathcal{V}(\mathbf{i_p}_j) \end{cases} \tag{4}$$

 ISBN 978-0-9934207-0-2

where $\mathcal{V}(\mathbf{i_p})$ is the set of vertices which voted for $\mathbf{i_p}$ and w_i and w_j are the weights assigned proportional to the sizes of $\mathcal{V}(\mathbf{i_p}_i)$ and $\mathcal{V}(\mathbf{i_p}_j)$ respectively. All selected intersection points conforming to the merging criteria are successively merged resulting in the final set of spheres (See Figure 8-b).



(a) (b) (c)

Figure 8: (a) Selected intersections before merging, (b) Final intersection points, used as the sphere centers. (c) Spheres with estimated radii.

Estimating the radius. Figure 9 shows the distribution of distances between a sphere center and the vertices belonging to the corresponding spheres. Except for some outliers, all of the vertices are expected to reside inside the sphere or over its surface. Therefore, after eliminating the outliers, determining the largest of the distances between a sphere center and corresponding vertices gives a reliable radius estimation. As seen from Figure 9, the sorted distance curves change smoothly up to a point that indicates the presence of outliers. Assume that D is the descendingly sorted distances (outliers come first) and D' is its rate of change. The first distance $d \in D$ with $d' < 2 * \text{mean}(D')$ is selected as the radius. The resulting spheres are illustrated in Figure 8-c.

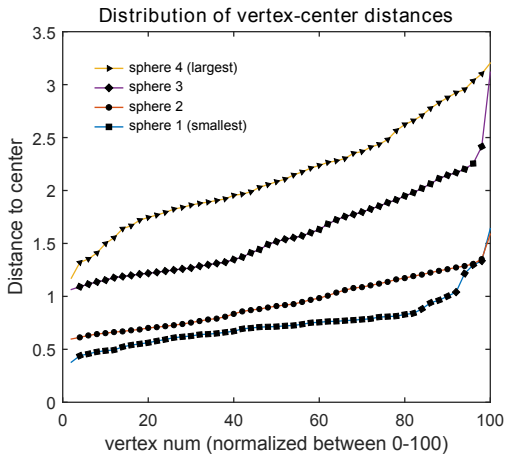


Figure 9: Each line corresponds to a different sphere.

Refining point cloud. After detecting the spheres, all vertices inside the spheres are projected onto the sphere surface using the following formula.

$$v' = v + \ r(\mathbf{i_p}) \frac{(v - \mathbf{i_p})}{\| v - \mathbf{i_p} \|}, \qquad (5)$$

where v' is the updated location of vertex v inside a sphere with center $\mathbf{i_p}$ and radius $r(\mathbf{i_p})$. The resulting positions could be kept as a point cloud or could be used for surface reconstruction. Figure 10 shows the reconstructed surface using our method and presents its effectiveness.

5 Conclusion

We have proposed a new modelling technique which allows for the reconstruction of 3D specular spherical surfaces using structure from motion software and a Hough Transform like procedure. Currently, the method is only applicable to spherical surfaces without any irregular structure such as concave indents and cracks over the surface. Future work will extend this method to enable the reconstruction of other reflective surfaces such as windows, which can also be semi transparent. Ultimately combining all of these techniques will allow us to

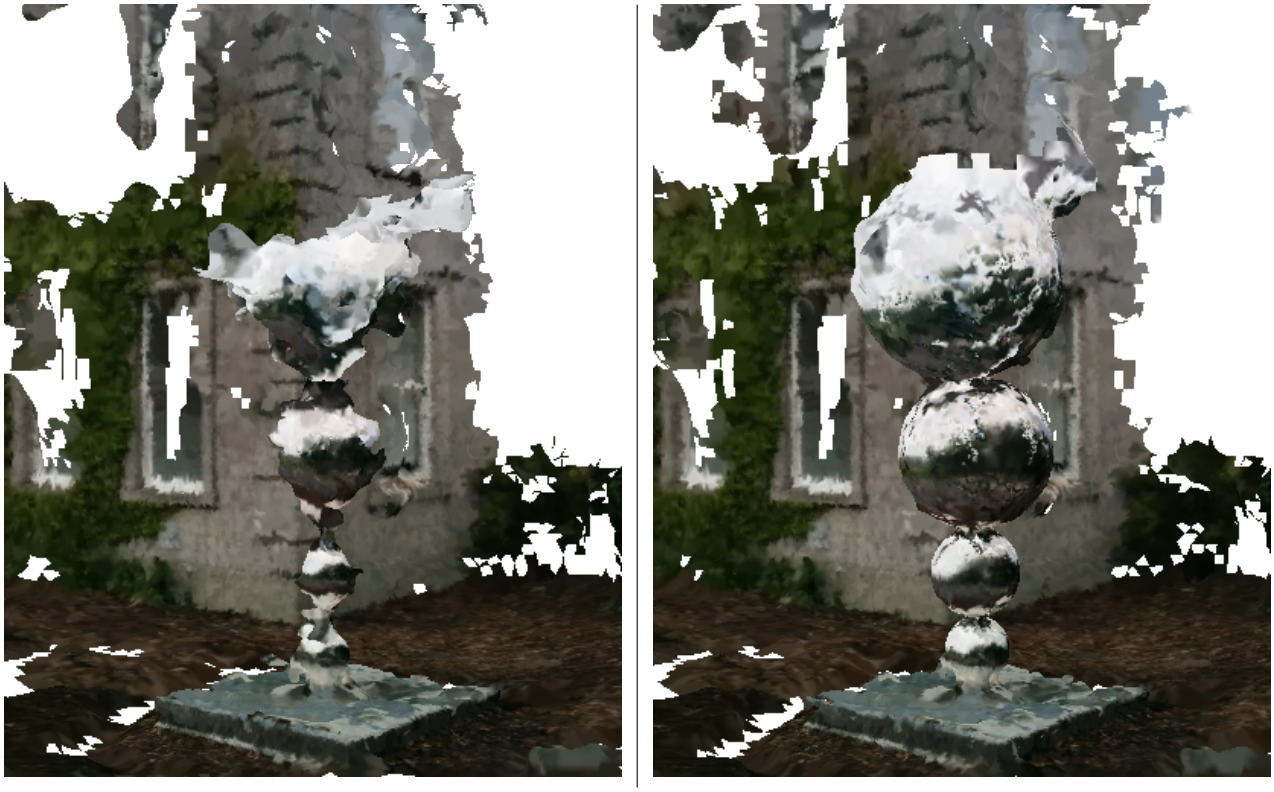


Figure 10: Comparison with the state of the art pipeline (left) using VisualSFM [Wu, 2013] for point cloud generation followed by Poisson Surface Reconstruction [Kazhdan et al., 2006]. On the right, the result of our reconstruction by adding our reflective spherical surface refinement step after point cloud generation.

recreate a more realistic 3D model using images captured from multiple sources (e.g. social media images or drone footage).

Acknowledgments

This work has been supported by EU FP7-PEOPLE-2013-IAPP GRAISearch grant (612334).

References

[Agarwal et al., 2011] Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. (2011). Building rome in a day. *Commun. ACM*, 54(10):105–112.

[Balzer et al., 2011] Balzer, J., Hofer, S., and Beyerer, J. (2011). Multiview specular stereo reconstruction of large mirror surfaces. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2537–2544.

[Brown and Lowe, 2005] Brown, M. and Lowe, D. (2005). Unsupervised 3d object recognition and reconstruction in unordered datasets. In *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pages 56–63.

[Cao et al., 2006] Cao, M., Ye, C., Doessel, O., and Liu, C. (2006). Spherical parameter detection based on hierarchical hough transform. *Pattern Recognition Letters*, 27(9):980 – 986.

[Ceylan et al., 2014] Ceylan, D., Mitra, N. J., Zheng, Y., and Pauly, M. (2014). Coupled structure-from-motion and 3d symmetry detection for urban facades. *ACM Trans. Graph.*, 33(1):2:1–2:15.

[Cohen et al., 2012] Cohen, A., Zach, C., Sinha, S., and Pollefeys, M. (2012). Discovering and exploiting 3d symmetries in structure from motion. In *CVPR*. Computer Vision and Patter Recognition.

[Dahyot, 2009] Dahyot, R. (2009). Statistical hough transform. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 31(8):1502–1509.

[Frahm et al., 2010] Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., and Pollefeys, M. (2010). Building rome on a cloudless day. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 368–381, Berlin, Heidelberg. Springer-Verlag.

[Furukawa and Ponce, 2010] Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376.

[Goesele et al., 2007] Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. (2007). Multi-view stereo for community photo collections. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8.

[Hartley and Zisserman, 2004] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

[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*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Lourakis and Argyros, 2009] Lourakis, M. A. and Argyros, A. (2009). SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30.

[O'Sullivan, 2010] O'Sullivan, C. (2010). Metropolis: Supercrowds for multisensory virtual environments,. ireland: Science foundation ireland funded project 2007-2010. `https://www.youtube.com/watch?v=fp6jvW3shTk`. accessed: 2015-05-28.

[Pollefeys et al., 2004] Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., and Koch, R. (2004). Visual modeling with a hand-held camera. *Int. J. Comput. Vision*, 59(3):207–232.

[Roth and Black, 2006] Roth, S. and Black, M. (2006). Specular flow and the recovery of surface structure. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1869–1876.

[Sankaranarayanan et al., 2010] Sankaranarayanan, A., Veeraraghavan, A., Tuzel, O., and Agrawal, A. (2010). Specular surface reconstruction from sparse reflection correspondences. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1245–1252.

[Schnabel et al., 2007] Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226.

[Snavely et al., 2006] Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3d. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 835–846, New York, NY, USA. ACM.

[Wanner and Goldluecke, 2013] Wanner, S. and Goldluecke, B. (2013). Reconstructing reflective and transparent surfaces from epipolar plane images. In Weickert, J., Hein, M., and Schiele, B., editors, *Pattern Recognition*, volume 8142 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin Heidelberg.

[Wu, 2013] Wu, C. (2013). Towards linear-time incremental structure from motion. In *3D Vision - 3DV 2013, 2013 International Conference on*, pages 127–134.

Kernel Density Filtering for Noisy Point Clouds in One Step

M.A. Brophy, S.S. Beauchemin, J.L. Barron

Department of Computer Science
University of Western Ontario, Canada
{mbrophy5,beau,barron}@csd.uwo.ca

Abstract

We present a method for filtering noisy point clouds, specifically those constructed from merged depth maps as obtained from a range scanner or multiple view stereo (MVS), applying techniques that have previously been used in finding outliers in clustered data, but not in MVS or range scanning. We estimate the probability density function (PDF) over the space of observed points via a technique called kernel density estimation. We utilize Mahalanobis distance and a variable bandwidth for weighting kernels accordingly, based on the nature of neighbouring points. Further, we incorporate a distance metric called the Reachability Distance that, as we show in our results, gives better discrimination than a classical Mahalanobis distance-based metric. With the addition of this nearest neighbour metric, we can produce results that are ready for meshing without any post-processing of the cloud. We mesh our filtered point clouds using a traditional surface fitting technique that is unequipped to deal with noise to demonstrate the efficacy of our method.

1 Introduction

Following a renaissance in energy-based MVS methods in the literature, there has been a return to methods that merge depth maps from multiple views to generate a representative point cloud. This change in methodology can be traced back to the work of [Goesele et al., 2006], whose simple method generated depth maps for each camera using adjacent views, and then merged those depth maps using third-party software designed to merge range images into a complex model. Prior to this, the MVS literature was dominated by methods that evaluated photo-consistency over a dense grid and then used an energy minimization technique to extract a representative surface, which is often the minimal surface.

A cursory glance at recent results on the Middlebury multi-view stereo data sets [Goesele et al., 2006] indicates incredible improvement of modern stereo matching algorithms over their predecessors. Such advancements are possible because of the improvements in disparity map generation like ordering constraints, bi-directional image matching, etc. That said, merging multiple depth maps and fitting a surface to the resultant point cloud remains a challenging endeavour, at least partially because of the presence of outliers and the general location of these outliers. These outliers can be very difficult to filter, as outlier clusters can occur both near to and far away from the true surface. Mahalanobis distance-based density estimation cannot correctly identify points that are close to the center of these outlier clusters as not being part of the surface.

The goal of this paper is to demonstrate that it is possible to fit a surface to a point cloud with very large quantity of outliers (a ratio of 5:1 outliers to inliers), by filtering using anisotropic kernel density estimation with variable bandwidth, and subsequently fitting a surface to this filtered cloud using standard surface meshing software [Cignoni et al., 2008]. We take clean, merged point clouds and populate them with noise. This way, we know the ground truth and can thus quantify this method's ability to discriminate inliers from outliers with the Receiver Operating Characteristic (ROC) curves [Provost and Fawcett, 2001].

Our method uses a process called kernel density estimation to construct a probability density function over the space of discrete data that we obtain from measured data. The target application is filtering point clouds obtained from MVS data. [Xi et al., 2009] and [Schall et al., 2005] utilize a similar process, but our method differs from theirs in that we use a variable bandwidth based on the nearest neighbour of each point that contributes to a density estimate, as inspired by [Latecki et al., 2007, Loftsgaarden and Quesenberry, 1965,

 ISBN 978-0-9934207-0-2

Terrell and Scott, 1992]. These works utilize variable bandwidths in finding outliers in clustered data. Our key observation is that these metrics are also very effective in filtering MVS and range scan data. Mahalanobis distance works well for discrimination of points near the true surface, in our experience better than Euclidean distance-based methods, but is prone to accept false positives near the center of a cluster of outliers. [Xi et al., 2009] note that, when necessary, they apply their filter repeatedly apply their filter until they get a result that is visually acceptable. Our method provides sufficient discriminatory ability such that it can be applied to a noisy point cloud once and the result can be meshed as is. We quantify our results and compare the discriminatory ability of our method with the Mahalanobis distance-based method used in [Xi et al., 2009].

2 Previous Work

[Lu et al., 2005] utilize tensor voting (TV) with a minimal surface-based fitting scheme to reconstruct surfaces from highly noisy (1:1 signal-to-noise ratio) point clouds. They use the level set formulation of [Zhao et al., 2000] to evolve an initial implicit surface to fit the points, but add an extra term to influence the motion in the direction of the tensor.

When working with noisy data, one way to deal with outliers is to average them out. This is the method utilized by [Goesele et al., 2006]. The software they use, *VRIP*, converts each depth map into a signed distance function (SDF), and then merges these SDFs using a weighted averaging based on the angle between the observed points and the sensor in each depth map. The area over which a point can be averaged with another point is referred to as its “ramp". Such a method results in smoothing of observations, and subsequent depth based methods that explicitly filter out a subset of the point cloud construct a surface from actual, unsmoothed observations.

[Li et al., 2010] identify “tracks", matched features that are found in at least three different views of an object, and then use bundle adjustment to recover the 3D point. If the reprojection error is above a certain threshold in one of the images, the point in question is discarded.

[Campbell et al., 2008] keep multiple hypotheses for prospective matches, and then use a spatial consistency measure in a Markov Random Field minimization scheme to recover better matches. If a point’s hypotheses are not spatially consistent with its neighbours, it is discarded. [Bradley et al., 2008] construct point clouds using multi-scale matching and then use an iterative filtering method for outlier detection on the resultant point cloud. They compute the projection of a point and its neighbours to a plane and then evaluate the fit using a density function.

[Xi et al., 2009] is the reference point for this work. They merge depth maps constructed from multiple views and use an anisotropic kernel density estimation method combined with a projected line search to obtain the maximum along each normal to find the maximum area of density on each normals path. We forego the use of reprojection error and “ramps" and instead use a density estimator to determine the quality of an observation. [Schall et al., 2005] also use an anisotropic kernel for filtering. Their method is similar to [Xi et al., 2009] in that they use an iterative method to move points along the normal direction to areas of maximal density. Further, they eliminate noise from high quality scans, and generate smooth surfaces from very high quality scans. Our focus is somewhat different, in that we study the circumstance where the quantity of outliers equals or exceeds the number of inliers, testing the ability to discern between inliers and the types of outliers that we see in photometric stereo, i.e. those that are both clustered near and far from the true surface. Further, we wish to simply filter the point cloud, as opposed to iteratively shifting the points to the area of highest density along the normal.

We assert that if a slightly better density estimator is used, one that uses both Mahalanobis distance, an adaptive bandwidth and the reachability distance [Breunig et al., 2000], we can mesh resultant point clouds without any further complication. We use a more advanced nearest neighbour metric, point clouds from range scans and MVS can be filtered without the projected line search while yielding an output cloud that can be meshed using an off-the-shelf method without any other pre-processing.

3 Obtaining a Probability Density Function from Measured Data

The easiest way to construct a probability density function from a set of points is a binning approach, similar to the construction of a histogram. Consider the problem in 1D: Say we have a set of measured data

 ISBN 978-0-9934207-0-2

$X = [x_1, x_2, \ldots, x_n]$ where each data point x_i is a scalar-valued observation. If we construct a set of k bins and simply count the number of items that fall within each bin, we can easily construct a histogram.

A number of questions arise:

- How many bins should we use?
- Where should our bins start and end?
- How does this strategy scale in higher dimensions, i.e., how do we determine the orientation of the bins in multiple dimensions?

The next section introduces a better method for constructing a PDF.

3.1 Kernel Density Functions

Kernel density functions propose to solve the problem of obtaining a probability density function in a different way. Instead of creating arbitrary bins of data, the density is instead evaluated at each point, using the distance to neighbouring points as input to a kernel function, the most commonly used of which is the Gaussian kernel [Xi et al., 2009]

$$K(x) = \frac{1}{(2\pi)^d} \exp\left(-\frac{x}{2}\right), \tag{1}$$

where d is the dimension. This is referred to as the Parzen Window technique [Parzen, 1962]. For each point $x_i \in X$, we rely on all points within a predefined radius to calculate the density of x_i using the kernel $K(x)$

$$f(x_i) = \frac{C_{k,d}}{n \cdot h} \sum_{j=1}^{n} K\left(\frac{||x_i - x_j||^2}{h}\right), \tag{2}$$

where the points x_j are the n neighbouring points of x_i within radius r, $C_{k,d}$ is a weight constant and h is the bandwidth.

Intuitively, $f(x_i)$ will be close to 1 if the sum of distances between x_i and its neighbouring points is small compared to the bandwidth. In other words, areas that contain a large number of points inside of their radius will yield a large density estimate and thus are more likely to be considered inliers than outliers.

3.2 Mahalanobis Distance

[Xi et al., 2009] utilize a more advanced method for filtering, based on the observation that the distribution of noise in point clouds tends to be anisotropic in nature. Thus, they evaluate an anisotropic kernel f of fixed radius r and shape at each point x to estimate its density utilizing its neighbouring points within distance r. Instead of using the L^2 distance between points within r, they find the distance to the center of mass by making use of

$$f(x) = \frac{C_{k,d}}{n} \sum_{i=1}^{n} K(d_\Sigma(x, x_i)), \tag{3}$$

where the kernel K is as defined previously. $d_\Sigma(\cdot,\cdot)$ is the Mahalanobis distance, which is defined as

$$d_\Sigma(x, x_i) = \left((x - x_i)^T H^{-1} (x - x_i)\right)^{1/2},$$

where the covariance matrix

$$H = DD^T,$$

can be constructed using

$$D = (x_1 - x, x_2 - x, \cdots, x_n - x).$$

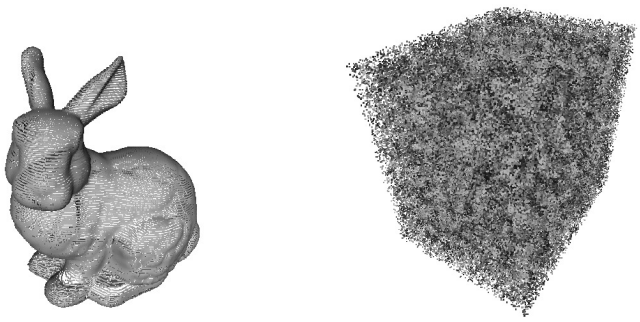
They find the location of highest density within the neighbourhood (r) and then use the distance to this point as the distance for the kernel to evaluate. This method discriminates between inliers and outliers when near the “true" surface much more robustly than the basic kernel density estimation method that we defined in Equation (2). We found that this method still had good discriminatory power when the signal-to-noise ratio was 1:10, i.e. we added 10 randomly generated outliers for each inlier in the point cloud’s bounding volume. Their method is prone to errors when there are clusters of outliers in a small area, a common occurrence.

This method isn’t new, it has been used in outlier estimation when dealing with clusters of data in the past. Likewise, our work is based on well-founded principles that are known in the literature. It hasn’t been applied to this domain, and its discriminatory ability is notable.

3.3 Local Bandwidth Estimation

The idea to use a local estimate of bandwidth $h(x_i)$ comes from [Latecki et al., 2007] who applied it to detecting outliers in clusters of data. It effectively unweights points whose n-th nearest neighbour is any larger than a very small distance from the point itself. Applying this to surface fitting makes a lot of sense, as points that lay near the true surface should have a neighbour in its very near vicinity.

4 Methodology



(a) Bunny. (b) Bunny, noise added.

Figure 1: The *Bunny* point cloud with 5:1 ratio of noise to inliers added.

We attempt to remove outliers via a method that differs from the previously described one in two ways. Based on the density of the nearest neighbour, we can weight each kernel accordingly. In other words, if a neighbouring point x_i itself has a low density, the bandwidth $h(x_i)$ will be lower and thus the contribution to the magnitude of $f(x)$ will be smaller than an equally distant point that exists in an area of higher density.

$$f(x_i) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{h(x_j)^d}K\left(\frac{d_\Sigma(x_j, x_i)}{h(x_j)}\right) \tag{4}$$

The bandwidth is the distance of the nearest neighbour to x_i, the dimension d is 3, $n = size(NN(x_i)$, where $NN(\cdot)$ is the set containing the nearest neighbours of x_i, the points within the radius r. The Gaussian kernel is as defined previously. The bandwidth,

$$h(x_j) = \min(d_\Sigma(x_j, x_k))$$

where $x_k \in NN(x_j)$, i.e. x_k is the nearest neighbour to x_j, when Mahalanobis distance is used to determine the “closeness" of two points. This method differs from the anisotropic kernel density method described in the previous section in that the density of a point *relies on the density of its neighbouring points*. In other words, we could have a point x_i and its nearest k points, and in the previous method, its kernel density estimate $q(x_i)$ would be the same irrespective of the points surrounding these neighbours.

4.1 Reachability Distance

We can extend this idea further by replacing the numerator of Equation (4) with a more robust metric called the *reachability distance*, where

$$\mathbf{rd}(x_i, x_j) = \max(d_\Sigma(x_j, x_i), d_\Sigma(x_k, x_j)). \tag{5}$$

This yields

$$f(x_i) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{h(x_j)^d}K\left(\frac{\mathbf{rd}(x_i, x_j)}{h(x_j)}\right), \tag{6}$$

and is thus composed of both the distance from x_i to its neighbours x_j, and distances of neighbouring points x_j to their nearest neighbour (x_k). If $rd(x_i, x_j) = d_\Sigma(x_i, x_j)$ then the inside of K is $-\frac{d_\Sigma(x_i, x_j)}{2d_\Sigma(x_j, x_k)}$ where the

 ISBN 978-0-9934207-0-2

numerator is greater than the denominator, and thus yields an increasingly smaller value as this difference increases when evaluated by the exponential. If $rd(x_i, x_j) = d_\Sigma(x_j, x_k)$ then the inside of the exponential is $-\frac{1}{2}$ and thus gives the minimal value.

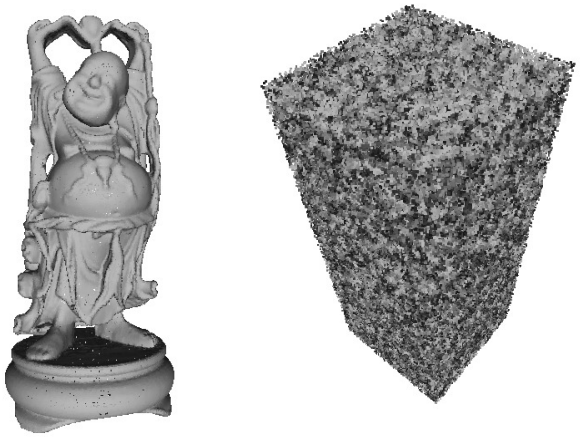
If this neighbour x_j's nearest neighbour is quite far away, i.e., x_k is a somewhat "isolated" point, it gives us very little information about the nature of point x_i, as being in the same neighbourhood as a likely outlier is not a great clue that x_i is an inlier. If, however, the distance to x_j is larger than the distance to x_j's nearest neighbour, we can say that having such a point in the neighbourhood is good evidence, and it is thus more likely that x_i is indeed an inlier!

5 Kernel Density Filtering on Data with Additive Noise

We approach it first using anisotropic kernel density estimation, where we evaluate the density of the data within a set radius of each data point. (3) evaluates the contribution of each point within this area, and takes into account the local density of each of these points as well.

We filter the signal by removing points whose density are below some threshold $\tau \in [0, 1]$. The method struggles with clusters of outliers though, and a substantial percentage of inliers are removed before the outliers fall below τ.

Interestingly, when we use the nearest-neighbour kernel density estimate, even in conjunction with the less discriminative L^2-norm, we still recover a much more accurate signal. That said, the performance of the anisotropic kernel density near the signal is better.



(a) Buddha statue. (b) Statue with noise added.

Figure 2: The *Buddha* point cloud with 5:1 ratio of noise to inliers added.

6 Results

We test our algorithm on the *Bunny* and *Buddha* data sets from Stanford and add random noise of varying quantities to determine the ability of our density-based method to discriminate between inliers and outliers. To quantify the filter's ability to discriminate between an inlier, i.e. a member of the normal class ("NC") and an outlier ("C"), we generate a ROC curve, which plots the detection rate (r_D) versus the false alarm rate (r_{FA})

$$r_D = \frac{TP}{TP+FN} \tag{7}$$

$$r_{FA} = \frac{FP}{FP+TN} \tag{8}$$

where *TP* is the number of true positives, *FN* is the number of false negatives, and *FP* is the number of false positives and *TN* is the number of true negatives. The nature of these terms is explained the confusion matrix seen in Table 1. A perfect ROC curve has an area of one beneath said curve.

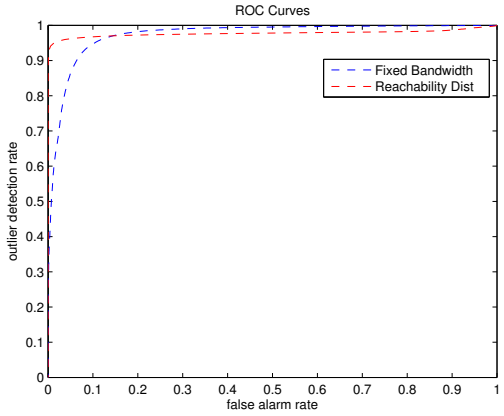


Figure 3: ROC curve for anisotropic filters on the *Bunny* point cloud with 5:1 noise. The reachability distance/adaptive method is in red.

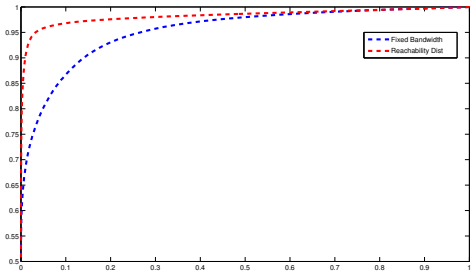


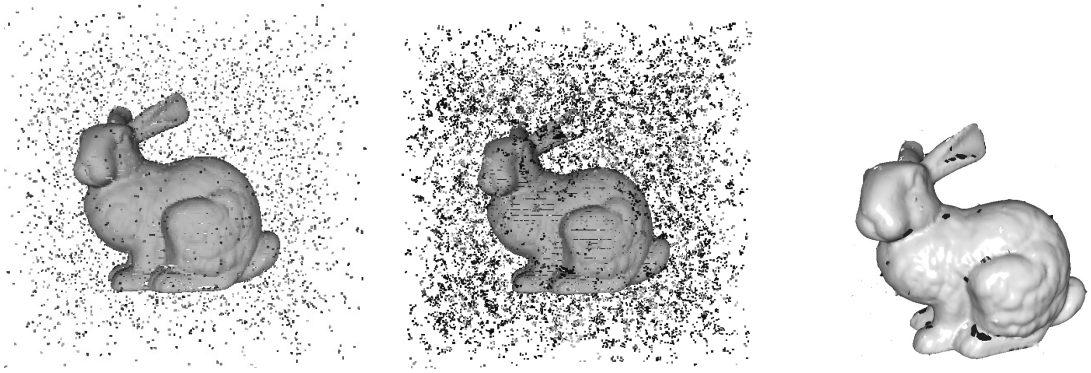
Figure 4: ROC curve for anisotropic filters on the *Buddha* point cloud with 5:1 noise. The reachability distance/adaptive method is in red.

As can be seen from the ROC curves in Figures 4 and 3, our method offers excellent discrimination between inliers and outliers in both circumstances. When one uses a strictly Mahalanobis-based density estimator on the *Bunny* data, a substantial portion of outliers remain. To make this more clear, we display the remaining points in the resultant cloud in Figure 5. There are far too many outliers in Figure 5(b) to allow for the fitting of a surface points, and true surface points are being thresholded as we remove more outliers. In Figure 5(a), a few outliers remain, but they occur in such low densities that they do not interfere with the subsequent surface fitting. We meshed the points from Figure 5(a) in Figure 5(c) with the ball pivoting algorithm [Bernardini et al., 1999].

Likewise, we see that our method works extremely well on the *Buddha* data set. Most impressive is its ability to handle the thin part of the statue above the head. As we see in Figure 6, despite the addition of 5:1 noise, we can still perform an accurate reconstruction of the surface with our method.

We also experimented with using the k nearest neighbours of x_i for estimation of the Mahalanobis distance in the above examples, but we found that the slight increase in discrimination was not worth the added time complexity.

Adjusting the area of support, r, has an effect on the nature of the filtering. If the algorithm is having trouble removing outliers near the surface, it may be useful to *decrease* the radius of the points that contribute to the density estimate. Increasing r will include more points with a larger distance to a point x_i if it is an outlier, but it will do the same for an inlier. The radius should be large enough to contain a sufficient number of points (for our purposes, ≥ 50), but small enough that the density estimates are excessively "smooth". Ideally, the estimation of f for any outlier near the surface will include a large sampling of inliers (i.e. actual surface points) to weight the center of mass correctly, yielding a small distance to points on the surface, and a large distance to outliers. Further, if there are thin areas on the surface, a small r can be useful to ensure that a density estimate at x is only influenced by its neighbours on the surface, not close by points



(a) Reachability distance. (b) Fixed Mahalanobis. (c), Bunny, meshed.

Figure 5: The *Bunny* point cloud with 5:1 noise, filtered. It was meshed with the ball pivoting method.

	Predicted Outlier Class (C)	Predicted Normal Class (NC)
Actual Outliers	True Positive	False Negative
Actual Normal Class	False Positive	True Negative

Table 1: Confusion matrix describing the different classifications of inliers and outliers.

that belong to a different part of the surface.

7 Conclusions and Future Work

We have demonstrated that our filtering method performs well on challenging data sets, even when the point cloud to which we apply our method is corrupted by large amount of noise. In reality, point clouds obtained from MVS or range scanning are not even near as noisy as our two corrupted point clouds. That said, the nature of the noise may be such that noise resides near the true surface, and it will thus be more difficult for the method to decipher whether a point is an inlier or an outlier.

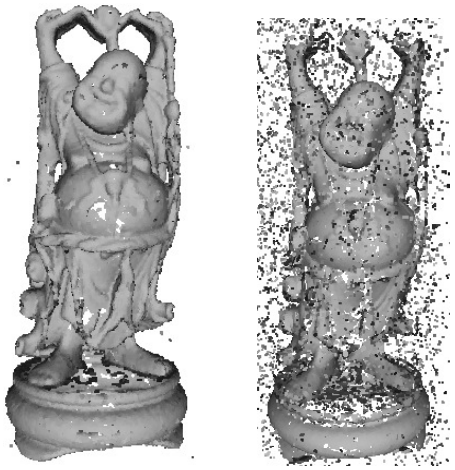
In in future, we would like to automate the process of fitting a surface to our filtered cloud, possibly by including our density estimate in a surface evolution scheme, similar to the level set-based method of [Zhao et al., 2000], with an extra term for density. It might be effective to include the confidence measure of each point in the point cloud from the stereo matching process. In the end, the goal is to obtain extremely accurate multi-view surface reconstructions of objects from multiple views, and a filtering method like the one we've presented is a step in that direction.

References

[Bernardini et al., 1999] Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4):349–359.

[Bradley et al., 2008] Bradley, D., Boubekeur, T., Berlin, T., and Heidrich, W. (2008). Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Breunig et al., 2000] Breunig, M., Kriegel, H., Ng, R., and Sander, J. (2000). Lof: Identifying density based local outliers. In *Proceedings of the ACM SIGMOD Conference*, Dallas, TX.



(a) Reachability distance. (b) Mahalanobis distance.

Figure 6: Buddha, filtered. The reachability distance-based filtering method leaves almost no outliers.

[Campbell et al., 2008] Campbell, N., Vogiatzis, G., Hernandez, C., and Cipolla, R. (2008). Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Proceedings of the European Conference on Computer Vision ECCV*, volume 1, pages 766–779.

[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 Scarano, V., Chiara, R. D., and Erra, U., editors, *Eurographics Italian Chapter Conference*. Eurographics.

[Goesele et al., 2006] Goesele, M., Curless, B., and Seitz, S. (2006). Multi-view stereo revisited. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2402–2409.

[Latecki et al., 2007] Latecki, L., Lazarevic, A., and Pokrajac, D. (2007). Outlier detection with kernel density functions. In *Proceedings of Machine Learning and Data Mining in Pattern Recognition*, pages 61–75.

[Li et al., 2010] Li, J., Li, E., Chen, Y., Xu, L., and Zhang, Y. (2010). Bundled depth-map merging for multi-view stereo. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:2769–2776.

[Loftsgaarden and Quesenberry, 1965] Loftsgaarden, D. O. and Quesenberry, C. P. (1965). A nonparametric estimate of a multi-variate density function. *Annals of Mathematical Statistics*, 36:1049–1051.

[Lu et al., 2005] Lu, H., Zhao, H., Jiang, M., Zhou, S., and Zhou, T. (2005). A surface reconstruction method for highly noisy point clouds. In *Proceedings of Variational, Geometric, and Level Set Methods in Computer Vision, LNCS*. Springer.

[Parzen, 1962] Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076.

[Provost and Fawcett, 2001] Provost, F. and Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231.

[Schall et al., 2005] Schall, O., Belyaev, A., and Seidel, H.-P. (2005). Robust filtering of noisy scattered point data. In Pauly, M. and Zwicker, M., editors, *IEEE/Eurographics Symposium on Point-Based Graphics*, pages 71–77, Stony Brook, New York, USA. Eurographics Association.

[Terrell and Scott, 1992] Terrell, G. R. and Scott, D. W. (1992). Variable kernel density estimation. *The Annals of Statistics*, 20(3):1236–1265.

[Xi et al., 2009] Xi, Y., Duan, Y., and Zhao, H. (2009). A nonparametric approach for noisy point data preprocessing. *International Journal of CAD/CAM*, 9(1):31–36.

[Zhao et al., 2000] Zhao, H., Osher, S., Merriman, B., and Kang, M. (2000). Implicit and non-parametric shape reconstruction from unorganized data using a variational level set method. *Computer Vision and Image Understanding*, 80:295–319.

Multiscale “Squiral" (Square-Spiral) Image Processing

Min Jing[1], Sonya. A. Coleman[1], Bryan. W. Scotney[2], Martin McGinnity[3]

[1]*Intelligent Systems Research Centre*
[2]*School of Computing and Information Engineering*
University of Ulster
[3] *School of Science and Technology, Nottingham Trent University*
United Kingdom

Abstract

In this paper, we present a multiscale “squiral" (square spiral) image processing (SIP) framework. An efficient spiral addressing scheme is deployed for standard pixel based square images to facilitate fast image processing. A SIP-based convolution technique is developed by simulating the “eye tremor" phenomenon of the human visual system. The multiscale SIP operators are constructed by converting existing square image operators according to the SIP addressing scheme. The results of edge detection based on three-layer SIP images and SIP operator at four different scales demonstrate the efficiency of the proposed framework by comparison with standard 2D convolution.

Keywords: spiral addressing scheme, multiscale operator, fast image processing, edge detection

1 Introduction

The spiral architecture [10] has been employed as an efficient addressing scheme in hexagonal image processing (HIP) [6], whereby the image pixel indices can be stored in a one-dimensional vector that enables fast image processing. Inspired by the spiral architecture for HIP, a spiral addressing scheme has been proposed for standard rectangular pixel based images, which is referred as “squiral" (square-spiral) image processing (SIP) [3]. The results of application of SIP based on Sobel edge detection demonstrate the efficiency of the proposed SIP framework. In this work, we aim to extend the SIP-based approach to multiscale operators.

It is well known that the edges of images are captured in the visual context of mammals at different resolution levels [11]. Therefore, the detection of image features in a multiscale sense is more appropriate than based on a single scale. A multiscale approach is desirable to achieve good results for detection and localisation. One of the most popular ways of applying edge detection operators at multiple scales is through the use of image pyramids [4], which applies the same operator (a smoothing function) to images being down-sampled to different scales. Alternatively, a multiscale representation of an image can be obtained by convolution of the image with a two-dimensional smoothing function defined at different scales. Studies on multiscale images have been carried out to enhance the performance of image processing [2, 5, 8, 12]. For example, since a multiscale Canny edge detection is equivalent to finding the local maxima of a wavelet transform, [5] studied the properties of multiscale edges through the wavelet theory. They show that the evolution of wavelet local maxima across scales characterise the local shape of irregular structures. In medical cell image analysis [12], differential wavelet transforms were deployed to facilitate the extraction of multiscale geometric features of chromosome images. In [8], an efficient scaling edge detector is developed by incorporating the image integral to achieve a fast image processing.

The objective of this study is to extend the SIP-based framework for use with multiscale operators. In this paper we first explain the SIP addressing scheme and conversion to SIP from rectangular images. We

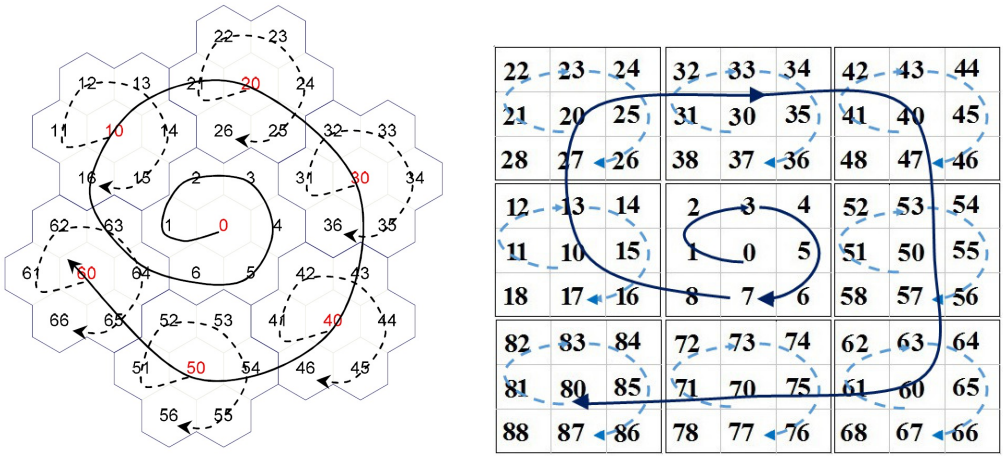
 ISBN 978-0-9934207-0-2

then give details of SIP-based convolution by simulating “eye tremor" in the human vision system, followed by development of multiscale SIP operators to be used in SIP based convolution. For illustration we use the application of edge detection. The preliminary results for performance evaluation based on three-layer SIP images and SIP operator at four different scales demonstrate the efficiency of the proposed approach.

2 Method

2.1 Spiral Addressing Scheme for Square Images

The spiral indexing schemes are illustrated in Fig. 1 for (a) hexagonal images and (b) square images. It can be seen that the hexagonal image can be considered as a layer-λ cluster comprising 7^λ pixels, and a square image consists of 9^λ pixels. In Fig. 1, only the first three layers are shown, corresponding to $\lambda = 0$, 1 and 2. The SIP structure facilitates the use of base 9 numbering to address each pixel within the image. For example, the pixels in layer-1 are labelled from 0 to 8, indexed in a clockwise direction. The base 9 indexing continues into each layer, e.g. layer-2 starts from 10, 11, 12, ..., and finishes at 88. Subsequent layers are structured recursively. The converted SIP image is stored in a one-dimensional vector according to the spiral addresses.



(a) (b)

Figure 1: Spiral addressing scheme for (a) hexagonal image and (b) square image (only the central part of the image is shown).

2.2 SIP Conversion

Due to existing hardware for capturing and displaying images being based on a rectangular architecture, HIP conversion requires preprocessing using a resampling scheme to match the location and value of points in an original rectangular pixel-based image to location and value in a hexagonal lattice [6]. Unlike HIP, SIP conversion directly uses the original lattice of a square image. For a rectangular image of size $M \times N$, the number of SIP layers λ can be found by $\lambda = (logM + logN)/log9$; then the length of the one-dimensional SIP image is 9^λ. Based on the spiral addressing scheme, a SIP address can be represented as:

$$a_n a_{n-1} ... a_1 = \sum_{i=1}^{n} a_i \times 10^{i-1} \tag{1}$$

where $0 \leq a_i < 9$ and $\sum$ denotes Spiral Addition and $\times$ indicates Spiral Multiplication [6]. To locate a SIP address corresponding to Cartesian coordinates (x,y), if the location of the centre of the image is defined as $L(0) = (0,0)$, then L(1) = (-1,0), L(2) = (-1,1), L(3) = (0,1), L(4) = (1,1), L(5) = (1,0), L(6) = (1,-1), L(7) = (0,-1) and L(8) = (-1,-1). Based on the SIP addressing scheme shown in Fig.1 (b), the points in a higher SIP layer can

be located by calculating the shift required from the centre point to the target point by

$$L(a_i \times 10^{i-1}) = 3^{i-1} \times L(a_i) \tag{2}$$

For example, a point at L(4536) can be located by $L(4536) = L(4000) + L(500) + L(30) + L(6) = 3^3 \times L(4) + 3^2 \times L(5) + 3 \times L(3) + L(6) = (37, 29)$. Hence the point L(4536) can be found by shifting the start point from (0,0) to (37,29). After conversion from a 2D image, the 1D SIP image is stored as a vector for further processing.

The SIP conversion starts from the centre of the image and the converted SIP is based on a square image that corresponds in size to layer-λ, which can be slightly smaller than the original image size. Hence a border region beyond that may not be processed. This is not a significant restriction, as in most image applications the elements of interest are unlikely to be at the periphery of the image. Alternatively, if it is required to cover all pixels, we may extend the image size to the next SIP layer by padding zeros to the boundary pixels.

2.3 Simulation of Eye Tremor

When using a standard 2D addressing scheme on an image, the addresses of a pixel's neighbours can be determined easily. However, determining a pixel's neighbours in a one-dimensional addressing scheme is not straightforward and requires significant computation. Inspired by [7,9], an eye tremor based framework for SIP is developed to help to determine a pixel's neighbours in a one-dimensional addressing scheme. An example of the pixel offsets for nine offset images in a layer-1 eye tremor are illustrated in Fig. 2.

I_2	I_3	I_4
I_1	I_0	I_5
I_8	I_7	I_6

Figure 2: Pixel offsets for the eight offset images used in layer-1 eye tremor.

Consider I_0 as a "base" SIP image; eight additional images $I_j, j = 0, 1, 2..., 8$ are obtained by shifting I_0 by one pixel in the image plane along the spiral addressing scheme. The "centre" of each image I_j is located at a pixel within the layer-1 neighbourhood centred at image I_0. Each image is stored as a vector after being converted from the 2D image structure. Similar to layer-1, a layer-2 eye tremor can be created based on the SIP addressing scheme containing eighty one eye tremor images, which is required for operators at higher scales.

2.4 SIP Convolution

For a given image I_0, convolution of a SIP operator (denoted as H_λ where λ denotes the SIP operator layer) across the entire image plane is achieved by applying the operator sparsely to each of the 9^λ eye tremor images I_j, and then combining the resultant outputs. For example, a SIP convolution using a layer-1 operator needs nine eye tremor images; for layer-2 operators, eighty one eye tremor images are needed.

Based on the eye tremor framework, for each image I_j, we apply the operator H_λ only when centred at those pixels with spiral address $0(mod 9^\lambda)$, hence achieving non-overlapping convolution. The convolution of the image I_j with an operator H_λ can be represented as,

$$G_\lambda^j(s_0) = \sum_{s \in N_\lambda(s_0)} H_\lambda(s) \times I_j(s), \tag{3}$$

where $\forall s_0 \in \{s | s = 0(mod 9^\lambda))\}$ and $N_\lambda(s_0)$ denotes the λ-neighbourhood centred on the pixel with spiral address s_0 in image I_j. Take a layer-1 operator H_1 as an example, the matrix implementation of convolution with I_0 in

 ISBN 978-0-9934207-0-2

Eq. (3) can be written as:

$$\begin{pmatrix} G_1^0(0) \\ G_1^0(10) \\ \vdots \\ G_1^0(k) \end{pmatrix} = \begin{pmatrix} I_0(0) & I_0(1) & ... & I_0(8) \\ I_0(10) & I_0(11) & ... & I_0(18) \\ \vdots & \vdots & \ddots & \vdots \\ I_0(k) & I_0(k+1) & ... & I_0(k+8) \end{pmatrix} \begin{pmatrix} H_1(0) \\ H_1(1) \\ \vdots \\ H_1(8) \end{pmatrix} \tag{4}$$

where $k = 0, 10, 20, 30, ...$ We can apply the same process to the remaining eight images $(I_1, ..., I_8)$; each is an image created by shifting the origin by one pixel from I_0. The process is illustrated in Fig. 3. The outcome can be obtained by assembling the values of G_1^j into a vector with the following arrangement:

$$\begin{pmatrix} G_1^0(0) & G_1^1(0) & \cdots & G_1^8(0) \\ G_1^0(10) & G_1^1(10) & \cdots & G_1^8(10) \\ \vdots & \vdots & \ddots & \vdots \\ G_1^0(k) & G_1^1(k) & \cdots & G_1^8(k) \end{pmatrix} \tag{5}$$

The final outcome in a SIP format is obtained by rearranging each row of the matrix in Eq.(5) into a vector,

$$[G_1^0(0)G_1^1(0)...G_1^8(0)G_1^0(10)G_1^1(10)...G_1^8(10)...G_1^0(k)G_1^1(k)...G_1^8(k)] \tag{6}$$

This assembly within the one-dimensional vector achieves the same outcome as standard 2D convolution. However, since all processing is executed in vector form, computation is significantly faster than standard convolution.

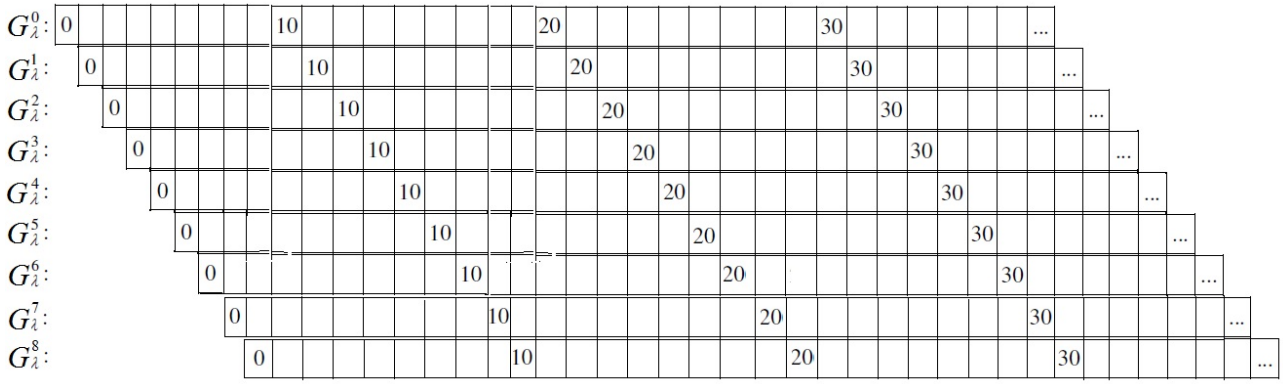


Figure 3: Assembly of the one-dimensional vectors for SIP-based convolution.

2.5 Multiscale SIP Operators

To obtain a multiscale representation of an image, a popular approach is the use of image pyramids [4], in which the same operator is applied to down sampled images at different scales. Here we present an alternative approach of applying a SIP operator at different scales.

Unlike HIP-based feature extraction, which requires x- and y-components of a hexagonal operator to be designed accordingly [1], the SIP-based approach supports direct convolution of standard image processing operators after converting the square operators into a SIP operator (a one dimensional vector). Fig. 4 presents the index addressing for the elements of multiscale SIP operators.

It can be seen that an operator of size 3×3 can be converted into a layer-1 SIP vector. For SIP based convolution Eq. (3), the length of the SIP operator H_λ matches the required number of eye tremor images. The operators of size 5×5 and 7×7 are converted into a layer-2 SIP vector by padding zeros to extend them to size 9×9. This process enables SIP convolution to be applied with operators at different scales.

 ISBN 978-0-9934207-0-2

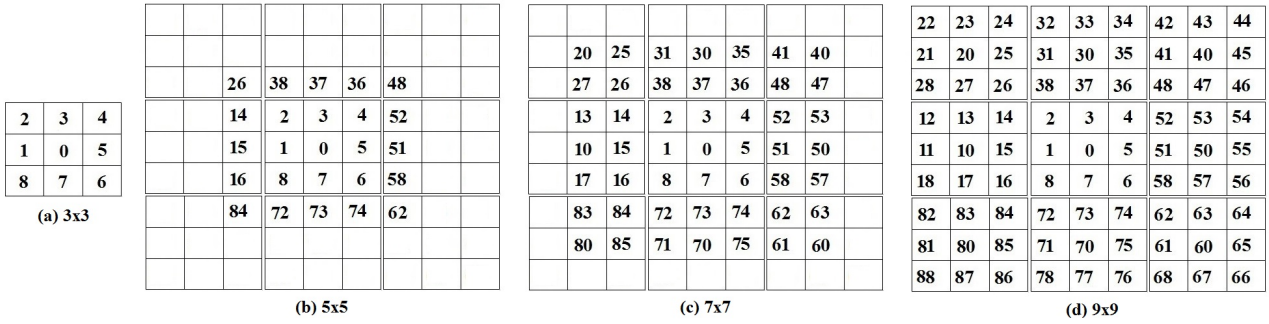


Figure 4: Index addresses for multiscale SIP operators: (a) 3×3; (b) 5×5; (c) 7×7 and (d) 9×9.

3 Experimental Results

3.1 Edge Detection

We first evaluated the performance of SIP edge detection for different image sizes using the Sobel edge detector (3×3) as a feature detector. In the experiment, we used three test images: lena, peppers and coins, with image sizes of 100×100, 384×520 and 738×900 respectively. Each image was converted into a SIP image, and in which the number of SIP layers for each are: layer-4 (81×81), layer-5 (243×243) and layer-6 (729×729), respectively. We compare the SIP approach to standard 2D convolution. Implementation of standard 2D convolution involves use of four for-loops that moves the kernel along each row and column of the input image and then computes the weighted sum over the neighbourhood.

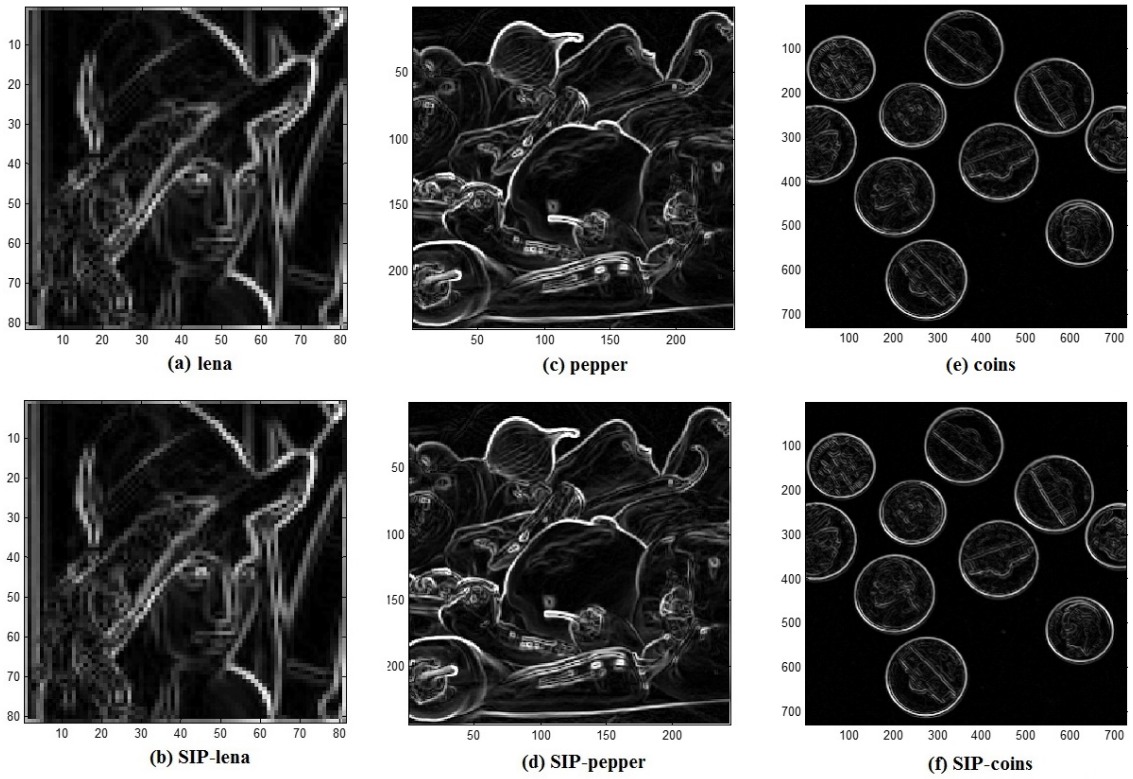


Figure 5: Comparison of edge results from standard 2D convolution and SIP-based convolution for layer-4 (lena), layer-5 (pepper) and layer-6 (coins) SIP images.

The results of edge maps are shown in Fig. 5. The top row shows the results from the standard 2D

 ISBN 978-0-9934207-0-2

convolution and the bottom row shows the results from the SIP approach. As SIP and standard convolution differ only in the approaches used for pixel indexing and image storage, the results of convolution are the same for both methods, which is clearly shown in Fig. 5.

3.2 Multiscale Operators

To implement the SIP multiscale operators for edge detection, we used the Canny edge detector to generate four operators with size 3×3, 5×5, 7×7 and 9×9, which were then converted to SIP operators as described in Section 2.5. The operators were generated based on first derivatives of 2D Gaussian and the standard deviation of the Gaussian functions was set as the square root of the filter size.

SIP convolution produces the same edge maps as the standard 2D convolution approach, and some examples are shown in Fig. 5. We show only the results from multiscale SIP operators here, where Fig. 6 presents gradient outputs and Fig. 7 presents the thresholded edge maps. In terms of quantitative analysis, the correlation coefficients between the edge maps from the SIP-based and standard 2D convolution approaches are all equal to 1 (as evaluated in [3]). These results demonstrate that the SIP-based approach to operator implementation can be used successfully at multiple scales.

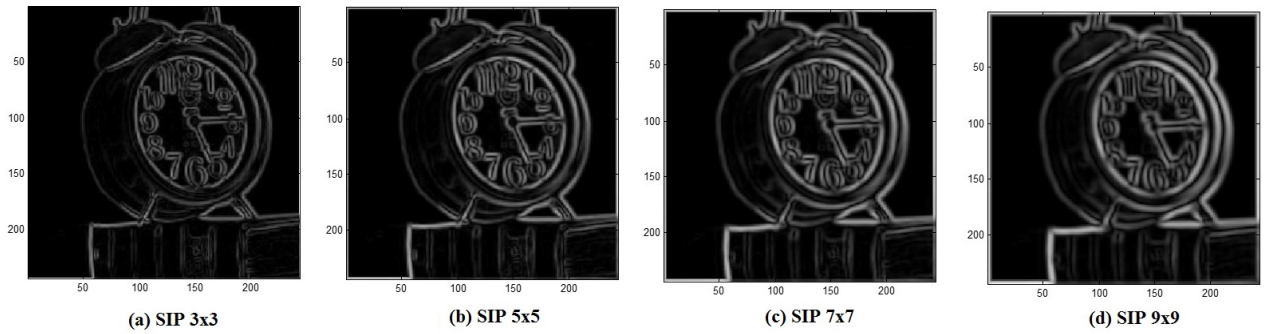


Figure 6: The edge maps from SIP-based method based on multiscale operators.

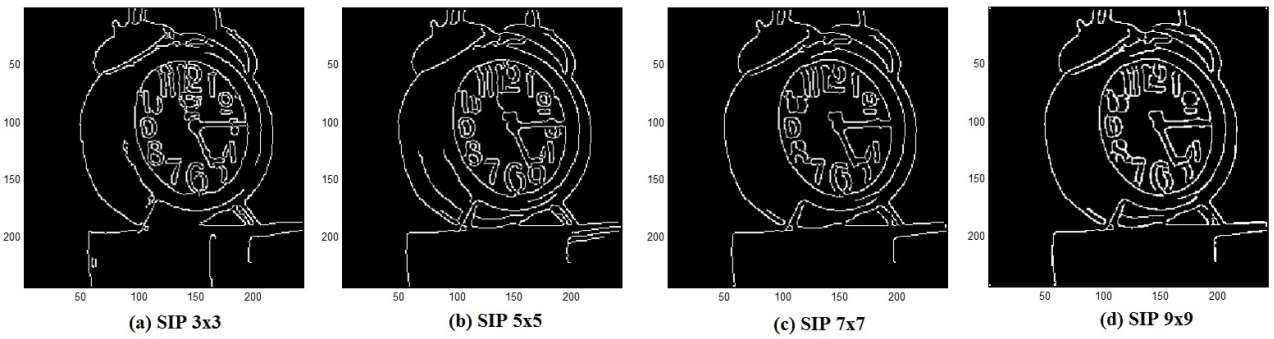


Figure 7: The edge results from SIP-based method based on multiscale operators after applying threshold.

3.3 Comparison of Run Time

To demonstrate the efficiency of the SIP based approach, the run-times are computed for the application of edge detection on the Lena, peppers and coins images using both standard convolution and the SIP based method. The same image size as the SIP image is used in each case. The results of run times based on averages over 100 runs are given in Fig. 8. From the results, we can see that for all cases, SIP is faster than standard 2D convolution. For the standard method, the computational cost increases as the sizes of images and operators increase. For the SIP-based approach, run time slightly increases with the image size, but time used for 5×5,

7×7 and 9×9 remains similar as they are all based on a layer-2 operator 9×9. The results demonstrate the efficiency of the proposed multiscale SIP framework.

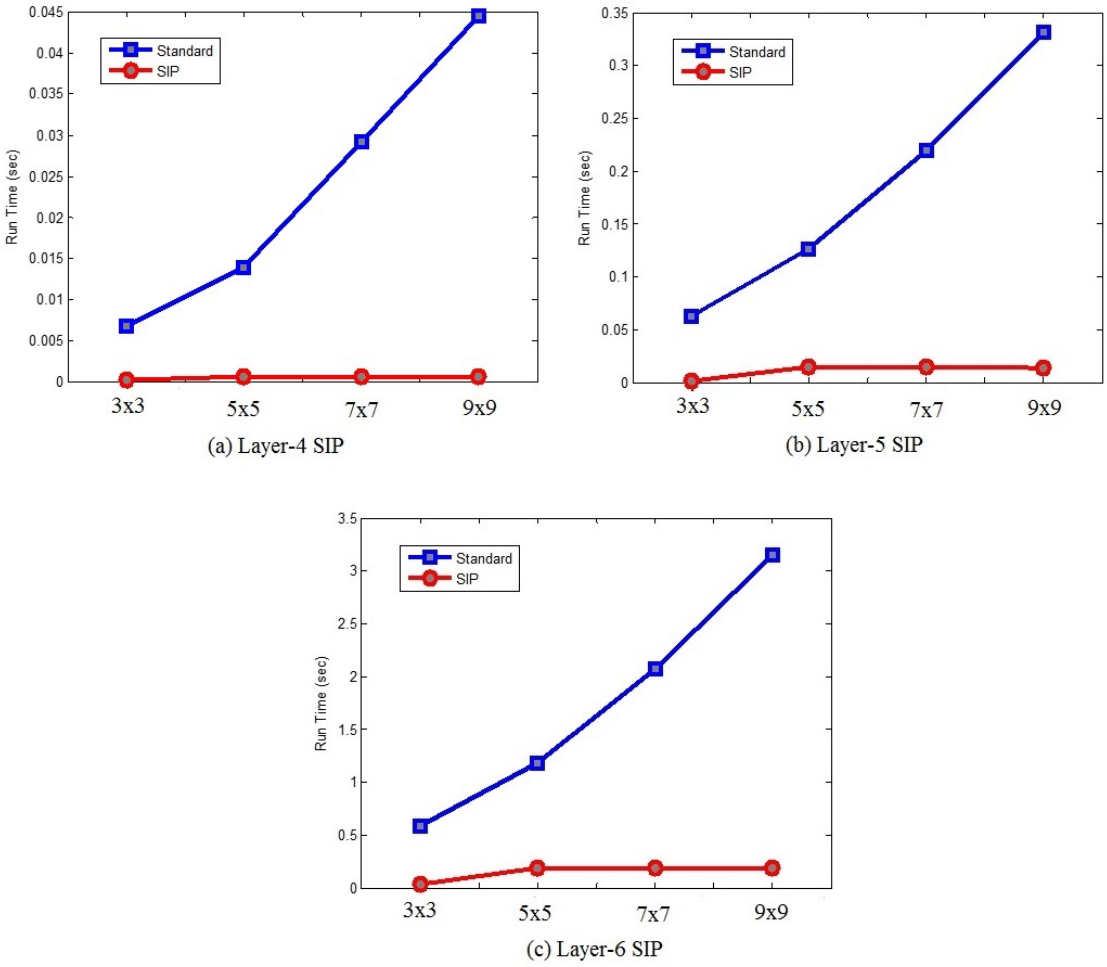


Figure 8: Comparison of run times for standard and SIP-based methods using multiscale operators.

4 Conclusion

In this paper, we present a novel framework developed for a multiscale SIP based convolution. The SIP-based approach enables fast feature detection by use of a spiral architecture in conjunction with eye tremor and non-overlapping convolution. The multiscale SIP operators can be formed by converting existing square image operators according to the SIP addressing scheme. Extending a layer-1 SIP operator to layer-2 level enables multiscale operators to be applied in the SIP framework. The results based on multiscale operators demonstrate the efficiency of the proposed method. The research outcomes facilitate the further development of the SIP based key-point detection, which will be conducted in the future work.

Acknowledgments

This work is supported by FP7 project Slandail (Security System for Language and Image Analysis. Project No: 607691). .

References

[1] SA. Coleman, BW. Scotney and B. Gardiner, “A Biologically Inspired Approach for Fast Image Processing," In IAPR Proc. Machine Vision Applications, pp.129-132, 2013.

[2] Z. Farbman, R. Fattal, D. Lischinski and R. Szeliski, “Edge-preserving decompositions for multi-scale tone and detail manipulation," ACM Transactions on Graphics (TOG), vol.27, no. 67, 2008.

[3] M. Jing, BW. Scotney, SA. Coleman and TM. McGinnity, “Biologically Inspired Spiral Image Processing for Square Images", In Proc. IAPR MVA, pp. 102-105, 2015.

[4] T. Lindeberg, “Feature Detection with Automatic Scale Selection," International Journal of Computer Vision, vol.30, pp. 79-116, 1998.

[5] S. Mallat, S. Zhong, “Characterization of signals from multiscale edges," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.14 , issue. 7, pp. 710-732, 1992.

[6] L. Middleton and J. Sivaswamy, “Hexagonal Image Processing; A Practical Approach," Springer 2005.

[7] A. Roka, et al., “Edge Detection Model Based on Involuntary Eye Movements of the Eye-Retina System," Acta Polytechnica Hungarica, 4(1), pp. 31-46, 2007.

[8] D. Kerr, SA. Coleman and BW. Scotney,“Efficiently Scaling Edge Detectors," In Proc. International Machine Vision and Image Processing Conference (IMVIP), 2010.

[9] BW. Scotney, SA. Coleman and B. Gardiner, “Biologically Motivated Feature Extraction Using the Spiral Architecture", In Proc. IEEE ICIP, pp. 221-224, 2011.

[10] P. Sheridan, T. Hintz and D. Alexander, “Pseudo-invariant Image Transformations on a Hexagonal Lattice," Image and Vision Computing, vol. 18, pp. 907-917, 2000.

[11] Yu-Ping Wang, “Image representations using multiscale differential operators," IEEE Transactions on Image Processing, Volume: 8, Issue: 12, pp. 1757- 1771, 1999.

[12] Y. P. Wang, Q. Wu, and K. R. Castleman and Z. Xiong, “Chromosome image enhancement using multiscale differential operators," IEEE Transactions on Medical Imaging, vol. 22, issue. 5, pp. 685-693, 2003.

Hand Hygiene Poses Recognition with RGB-D Videos

Baiqiang Xia[†], Rozenn Dahyot[†], Jonathan Ruttle[‡], Darren Caulfield[‡] and Gerard Lacey[†‡]

† : *School of Computer Science and Statistics, Trinity College Dublin, Ireland*
‡ : *Glanta ltd, Surewash*

Abstract

Hand hygiene is the most effective way in preventing the health care-associated infection. In this work, we propose to investigate the automatic recognition of the hand hygiene poses with RGB-D videos. Different classifiers are experimented with the Histogram of Oriented Gradient (HOG) features extracted from the hand regions. With a frame-level classification rate of more than 95%, and with 100% video-level classification rate, we demonstrate the effectiveness of our method for recognizing these hand hygiene poses. Also, we demonstrate that using the temporal information, and combining the color with depth information can improve the recognition accuracy.

Keywords: Hand Hygiene, Poses Recognition, RGB-D

1 Introduction

According to the World Health Organization (WHO), hands are the main pathways of germ transmission in health care-associated infections (HCAI), which causes thousands of people deaths and billions of money losses each year. Hand hygiene plays a crucial role in the prevention of HCAI, which is usually achieved by rubbing hands with an alcohol-based formulation. In the WHO Guideline book of hand hygiene [WHO, 2009], routine gestures for hand hygiene with alcohol-based hand-rub formulation or soap[1] have been suggested. More specifically, some hand gestures of interest in the hand hygiene procedure are illustrated in Figure 1.

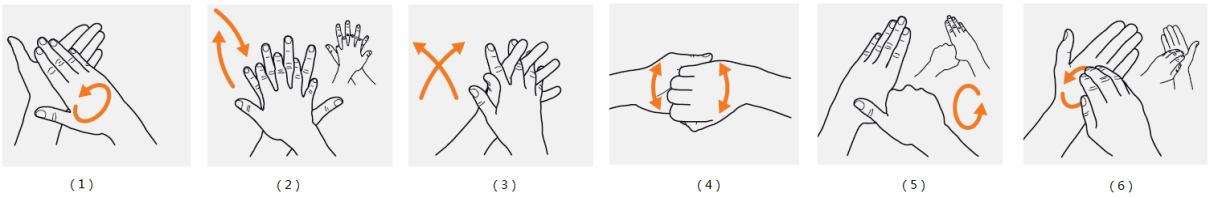


Figure 1: Hand hygiene Poses suggested by WHO. (1) : Rub hands palm to palm, (2) : Right palm over left dorsum with interlaced fingers and vice versa, (3) : Palm to palm with fingers interlaced, (4) : Backs of fingers to opposing palms with fingers interlocked, (5) : Rotational rubbing of left thumb clasped in right palm and vice versa, (6) : Rotational rubbing with clasped fingers of right hand in left palm and vice versa.

In the following, we first briefly review the vision based approaches for hand pose recognition with highlights on hand-washing poses recognition (section 2), and then present our own approach (section 3). Experimental results (section 4) show the effectiveness of the proposed approach in this hand hygiene recognition task, as well as the benefit of combining depth and color information, and using the temporal information. Finally, section 5 draws some conclusions of this work.

[1] Soaps are suggested when the hands are visibly soiled. If not, the alcohol-based hand-rub formulation is suggested to be used

 ISBN 978-0-9934207-0-2

2 State of the Art

Hand pose has attracted active research in computer vision domain, either for the hand pose estimation task which aims at the reconstruction of hand posture [Supancic III et al., 2015], or for the hand pose classification task which aims at recognizing a set of predefined hand gestures [Suarez and Murphy, 2012]. Concerning hand pose classification, it usually involves an early stage of hand localization in the video frames, followed by a machine learning stage which applies classifiers on extracted features to finally predict the pose [Suarez and Murphy, 2012, Sarkar et al., 2013]. For hand localization, various hand segmentation methods were proposed in consideration of the skin-color maps [Ibraheem et al., 2013]. These methods are claimed to suffer greatly from light changes, even with an illumination-invariant color schemes. In recent years, with the spread of commodity depth cameras, more and more works use simply a depth threshold to isolate the hands [Suarez and Murphy, 2012]. This type of method usually works with the presence of a single hand, which is also required to be the closest object to the camera. Ghobadi et al. [Ghobadi et al., 2007] explored the combination of the color and depth information for hand segmentation, in a pixel level clustering scheme. In the following machine learning stage, the hands are first represented by relevant features, such as color features [Llorca et al., 2011], shape features [Keskin et al., 2012, Suryanarayan et al., 2010], volume features [Suryanarayan et al., 2010], and temporal motion features [Elmezain et al., 2009]. The features are then fed to classifiers to predict the pose label, such as the Hidden Markov Model (HMM) [Kurakin et al., 2012], the Neural Networks [Hasan and Abdul-Kareem, 2014], the Support Vector Machine [Llorca et al., 2011], the Random Forest [Keskin et al., 2012], and the Linear Discriminant Analysis (LDA) [Wang and Zhang, 2013], etc.

Many applications have been derived from the related research, which mainly fall in the areas of human computer interaction (HCI) [Rautaray and Agrawal, 2015], robot control [Khan and Ibraheem, 2012] and human sign language recognition [Slama et al., 2014]. Concerning hand washing poses recognition, only a few works have been issued with videos from standard RGB cameras. In [Llorca et al., 2007, Llorca et al., 2011], researchers proposed to use a set of 21 binary SVM classifiers for recognizing the 6 hand washing gestures suggested in [RCN, 2005], together with another hand pose class defined as not belonging to any of the 6 poses. Prior statistics of skin/non-skin color, and motion information between frames are required in their method for hand segmentation. In both works, the videos were manually filtered and labeled by human experts on each frame. In [Hoey et al., 2010], a vision based system is proposed to assist people with difficulties in washing their hands. As far as we know, the depth information has not been investigated yet for this specific task.

3 Hand Hygiene Poses Recognition

We propose a computer vision system for recognizing these poses in RGB-D video streams of hands (cf. Figure 2). With a set of RGB-D hand hygiene videos, we first perform hand region segmentation on each frame with the Expectation Maximization technique working on the Gaussian Mixture Model. Then we extract the Histogram of Oriented Gradient (HOG) features on the hand regions, which are later processed with the Principal Component Analysis (PCA) for dimensionality reduction. The PCA transformed features are then used for training and testing, with different classifiers. Section 3.1 presents our strategy for hand segmentation. Features and classifiers are presented in section 3.2.

3.1 EM-based hands segmentation

To analyze the behavior of the hands in the videos, we are required to separate the hands from the background. To this end, for each pair of RGB and depth images, we first transformed the RGB image into the HSV color space, then constructed a feature vector (Fp) on each valid pixel using the Hue (h) and Saturation (s) information, together with the depth (d) information from the depth image. Formally, it means Fp={h,s,d}, with h, s, and d as integers in the scale of [0,255]. With these pixel-level features, we explored the Expectation Maximization (EM) technique with the Gaussian Mixture Models representation of the image features for hand segmentation. The Gaussian Mixture represents each segment of the features by a parametric Gaussian Distri-

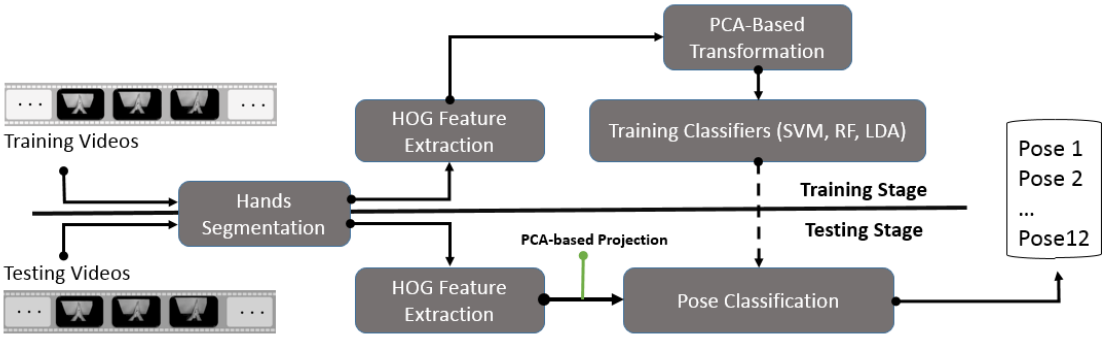


Figure 2: Overview of the proposed hand hygiene poses recognition pipeline.

bution. The Expectation Maximization (EM) technique then searches for the best parameters of the Gaussian models, which produce a maximum likelihood estimation with the given features. In the output of EM, each pixel is labeled with the Gaussian model with the highest probability to produce this pixel. It has been reported in [Ghobadi et al., 2007] that effective hand segmentation can be achieved using the EM technique and the Gaussian Mixture Models, based on the depth and color information. In our work, We find 4 Gaussian kernels provide the most accurate segmentation of hands in visual observation.

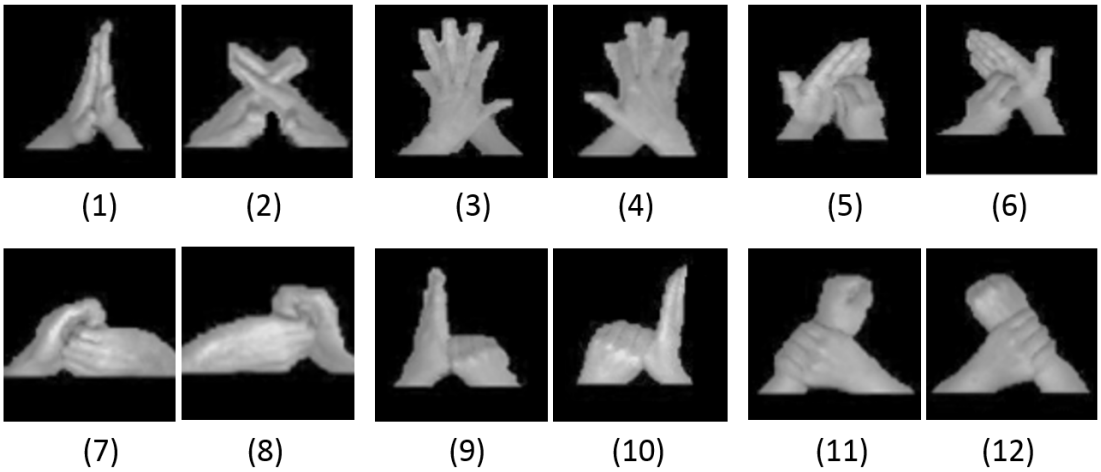


Figure 3: Illustration of segmentation output for 12 hand hygiene poses. (1) : Rub hands palm to palm; (2) : Rub interlaced fingers; (3) and (4) : Left palm over right dorsum with interlaced fingers and vice-versa; (5) and (6) : Rotational rubbing, backwards and forwards with clasped fingers of right hand in left palm and vice versa; (7) and (8) : Backs of right fingers to left palm with fingers interlocked and vice versa; (9) and (10) : Rotational rubbing of left thumb clasped in right palm and vice versa; (11) and (12) : Rotational rubbing left wrist clasped in right hand and vice versa.

3.2 Features Extraction and Machine Learning

With the segmented frames, we cut out the hand region, by removing the pixels below the row where the two arms are joined. The remaining part is then re-sized to a unified resolution of 80×80, for extraction of the Histogram of Oriented Gradient (HOG) features [Dalal and Triggs, 2005]. The HOG features are later explored in pose classification with different classifiers, namely the linear-kernel SVM classifier [Chang and Lin, 2011], the Linear Discriminant Analysis (LDA) classifier [Scholkopft and Mullert, 1999], and the Random Forest classifier [Breiman, 2001].

 ISBN 978-0-9934207-0-2

4 Experimental Results

4.1 Video Dataset and experimental design

We collected 72 RGB videos and 72 corresponding depth videos from 6 different subjects using the SoftKinetic DS325 camera. Each subject performs 12 different hand hygiene poses above a planar table, with the camera projecting vertically downwards. Each video lasts 2-4 seconds. With a frame rate of 25 per second, it results in 42-110 frames in each video. As there is misalignment between the images from RGB and depth sensors due to different sensor positions, we warped each color frame to the corresponding depth frame with the warping matrix provided by the camera system. In Figure 4, we illustrate an episode of a depth video and the corresponding RGB color video after warping.

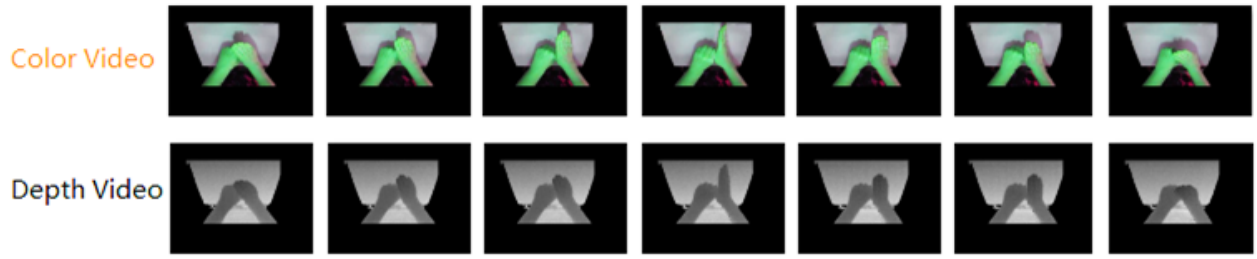


Figure 4: Example of an RGB Video and the corresponding Depth Video

We use the Leave-One-Person-Out (LOPO) subject-independent cross-validation protocol in evaluating the performance of the pose classification method, where each subject is used for the testing step once, with the remaining subjects used in the training step. This protocol enables the largest number of instances in training step, with the requirement that no subjects should enroll in both training and testing steps at the same round. Under this protocol, we explore the usage of the RGB channel (using the corresponding grayscale values), and the depth channel separately, in hand hygiene poses recognition. To address the high dimensionality of the HOG features (2916 dimensions), in each round of the cross-validation, the original HOG features are projected in lower dimensional subspace using the Principal Component Analysis (PCA). We note here that, in each round, the projection matrix of PCA is learned on the training set alone.

4.2 Frame-level Pose recognition

Figure 5 shows the pose classification results when recognizing on each frame of the videos. The x-axis shows the dimensionality of the PCA-based feature subspace, and the y-axis shows the average recognition rate over all the frames, for different classifiers. In Figure 5, for both the color and depth channels, the LDA classifier outperforms the Linear-SVM classifier, and further outperforms the Random Forest classifier. In Figure 5 (a), with 75 dimensions of the PCA-based features, the the LDA classifier achieves 94.80% pose recognition rate. The corresponding results for the Linear-SVM is 93.17%, and is 91.87% for Random Forest. With the depth information, as shown in Figure 5 (b), the LDA classifier achieves 92.35% classification rate using 100 dimensions of the PCA-based features. Correspondingly, the Linear-SVM classifier achieves 89.75%, and the Random Forest classifier reaches 87.97%. These results demonstrate the effectiveness of our hand hygiene pose recognition approach, as well as the stability of the PCA projections in different rounds of the cross-validation. Additionally, the color channel generally outperforms the Depth channel. We assume that, apart from the data modality differences, the low quality of the depth data provided by the camera also accounts for this. In Table 1, we show the average time consumption for classifying each frame, concerning the above results, generated by an Intel Core i7 CPU 3.70 GHZ with 16GB of RAM, in Matlab implementation. Again, the LDA classifier demonstrates significant merit over the Linear-SVM. Notably, the Random Forest classifier results in the smallest time consumption. Considering both the recognition rate and time consumption, **we choose the results from the LDA classifier for further analysis**.

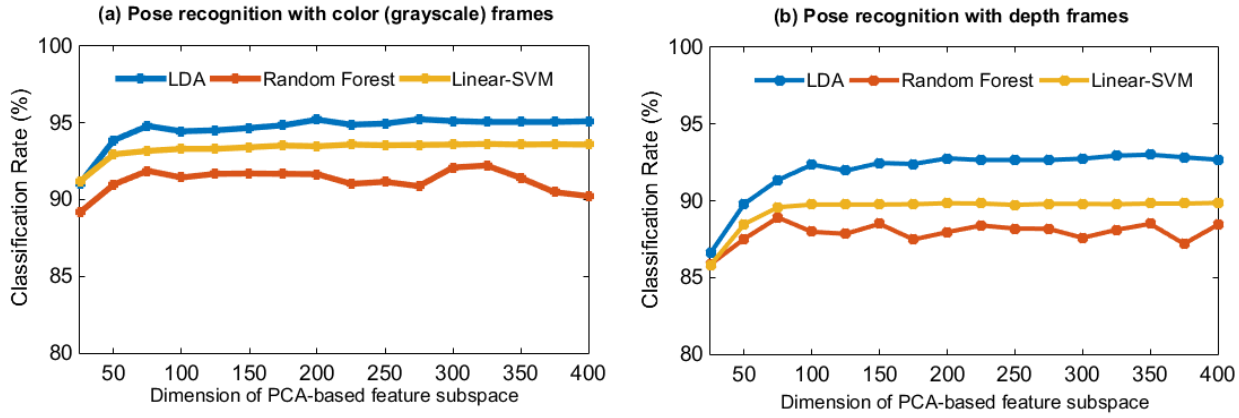


Figure 5: Frame by Frame Pose recognition Results

	LDA	Random Forest	Linear-SVM
Color	0.0100 ms	0.0077 ms	0.6774 ms
Depth	0.0139 ms	0.0086 ms	0.8372 ms

Table 1: Time Consumption for classifying each frame (millisecond).

Apart from the above, another perspective of the results is the confusion matrix, which shows the interaction of different classes during classification. Thus, we show in Table 2 and Table 3 the confusion matrices of the recognition results from the LDA classifier, with the color frames and with the depth frames, respectively. In the two tables, the ground truth goes with the row labels, and the pose estimation result goes with the column labels. In both confusion matrices, the diagonal parts have the dominant accumulation of the data. Very few instances are confused, as shown in the off-diagonal parts of the matrices (blanks mean 0). It means that our pose recognition approach is not biased to particular poses.

The closest work to ours in the state of the art is presented in [Llorca et al., 2011], in which the researchers performed the recognition of 7 types of hand poses with the RGB videos. With manually filtered and labeled frames in 6 videos for training and frames of another 2 videos used for testing, they achieved 82.29% correctness over the 7 pose classes, using 21 binary SVM classifiers. No cross-validation scheme was issued in their work. In comparison, as shown in Figure 5 (a) and Table 2, we have achieved a more promising classification rate of 94.80% over 12 hand pose classes in subject-independent cross-validation, with a single LDA classifier which also works much faster than SVM as shown in Table 1. In addition, Figure 5 (b) and Table 3 also make the first published results for hand hygiene gesture recognition using the depth information.

4.3 Video-level Pose recognition results

Although the frame-level pose recognition has demonstrated its effectiveness, the temporal evolution information of the video frames has been omitted. We are motivated to explore the usage of the temporal relationship of video frames. With this concern, we switch from recognizing the pose in the frames, to recognizing the pose in the episodes. By applying the sliding-window technique on the temporal frames, we formatted a set of video episodes. The poses in these episodes are recognized as the majority voting results of the results from the containing frames. As the window size controls the length of the episode (number of frames), it further influences the recognition accuracy. Thus, we show in Figure 6 the relationship between the window size and the recognition rate. To combine the contributions from both channels, we also propose a fusion scheme which performs majority voting on frames of both the color and the depth channels within each episode. For the

 ISBN 978-0-9934207-0-2

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
G1	315	1			2				1	4	6	
G2		308									1	
G3		8	377	8								
G4			1	395								
G5	2	5			337	2	4			12	22	
G6						406				8	6	
G7							434			31	1	
G8					4			349		14		
G9	10	1							382		31	
G10		2			7		2	1		369	9	1
G11	1	4				1	2		1		379	
G12		3									21	324

Table 2: Confusion matrix of pose recognition with color frames. G: Ground truth, E: Estimated label

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
G1	307				1						21	
G2		307									2	
G3		2	362	26								3
G4		1		394	1							
G5	10	2	2		316		7	12	2	20	4	9
G6						413	1			3	3	
G7	1						433			17	4	11
G8					19			300		47	1	
G9	13	4		2					385		20	
G10	1	1			5		8	9	3	356	4	4
G11	26	1				1	13				347	
G12										3	3	342

Table 3: Confusion matrix of pose recognition with depth frames. G: Ground truth, E: Estimated label

majority voting procedures, we use the same frame-level results as for the previous confusion matrices.

In Figure 6, we observe that, for both the color and the depth channels, the larger the sliding window size, the higher the pose recognition rate. With a window size of 10, we achieve 98.99% and 97.86% recognition rate for color and depth channels, respectively. With window size growing to 20, the corresponding recognition rates reach 99.38% and 98.31%. These results outperform significantly the corresponding frame-level results, which were 94.80% in the color channel and 92.35% for the depth channel. Apart from this, when we take each video as one episode (not shown in the Figure), both the color and depth channels achieve 100% classification rate. These results demonstrate that the temporal information gives enhancement to this pose recognition problem. Also, the fusion scheme obviously outperforms each single channel. With window size of 10, we achieve 99.17% pose classification rate. It demonstrates that the combination of the color and the depth channels also enhances the pose recognition performance.

5 Conclusion

In this work, we proposed a computer vision framework to recognize the hand hygiene poses from RGB-D videos. Using the LDA classifier on PCA-projected HOG features, we achieve >95% frame-level recognition

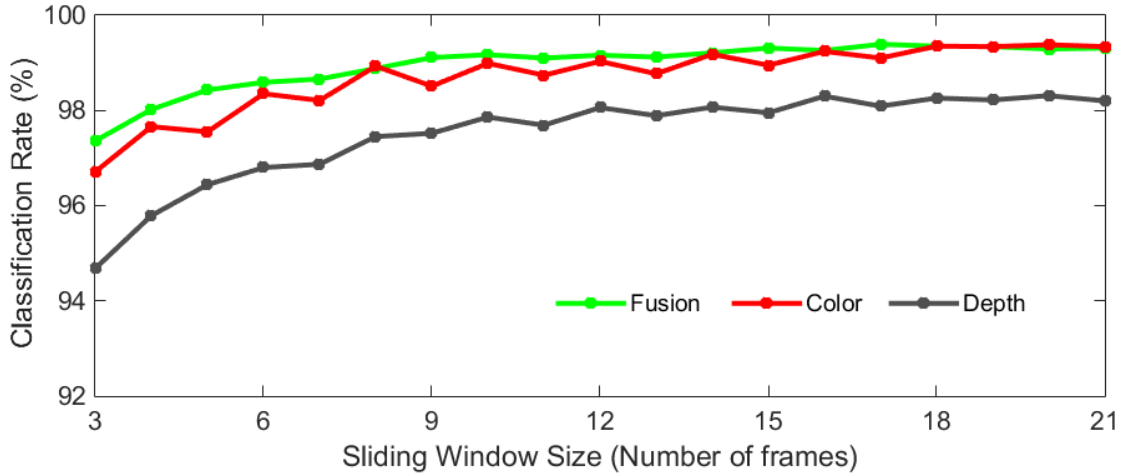


Figure 6: Pose recognition rates with different sliding window size

rate, and 100% video-level classification rate. It demonstrates the capability of the proposed method in discriminating the hand-hygiene poses. In addition, the temporal information further improves the performance of our system. We also show that the fusion of color and depth channels is beneficial to the recognition performance. The resulted system could be applied in hand hygiene monitoring, education, and also could be extended to more constrained scenarios, such as in surgery preparation where additional hand hygiene poses are required.

Acknowledgments

This work has been supported by the Innovation partnership project (IP-2014-0290) funded by Enterprise Ireland, the European Regional Development Fund, Movidius.com and SureWash.com.

References

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

[Elmezain et al., 2009] Elmezain, M., Al-Hamadi, A., Pathan, S. S., and Michaelis, B. (2009). Spatio-temporal feature extraction-based hand gesture recognition for isolated american sign language and arabic numbers. In *Image and Signal Processing and Analysis, 2009. ISPA 2009. Proceedings of 6th International Symposium on*, pages 254–259. IEEE.

[Ghobadi et al., 2007] Ghobadi, S., Loepprich, O., Hartmann, K., and Loffeld, O. (2007). Hand segmentation using 2d/3d images. In *IVCNZ 2007 Conference, Hamilton, New Zealand*, volume 5.

[Hasan and Abdul-Kareem, 2014] Hasan, H. and Abdul-Kareem, S. (2014). Static hand gesture recognition using neural networks. *Artificial Intelligence Review*, 41(2):147–181.

[Hoey et al., 2010] Hoey, J., Poupart, P., von Bertoldi, A., Craig, T., Boutilier, C., and Mihailidis, A. (2010). Automated handwashing assistance for persons with dementia using video and a partially observable markov decision process. *Computer Vision and Image Understanding*, 114(5):503–519.

[Ibraheem et al., 2013] Ibraheem, N. A., Khan, R. Z., and Hasan, M. M. (2013). Comparative study of skin color based segmentation techniques. *International Journal of Applied Information Systems (IJAIS)*.

[Keskin et al., 2012] Keskin, C., Kıraç, F., Kara, Y. E., and Akarun, L. (2012). Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Computer Vision–ECCV 2012*, pages 852–863. Springer.

[Khan and Ibraheem, 2012] Khan, R. Z. and Ibraheem, N. A. (2012). hand gesture recognition: a literature review. *International Journal of Artificial Intelligence & Applications (IJAIA)*, 3(4).

[Kurakin et al., 2012] Kurakin, A., Zhang, Z., and Liu, Z. (2012). A real time system for dynamic hand gesture recognition with a depth sensor. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1975–1979. IEEE.

[Llorca et al., 2007] Llorca, D., Vilarino, F., Zhou, Z., and Lacey, G. (2007). A multi-class svm classifier ensemble for automatic hand washing quality assessment. In *BMVC Proc. Brit Mach Vision Conference, Warwick, UK*, pages 213–223.

[Llorca et al., 2011] Llorca, D. F., Parra, I., Sotelo, M. Á., and Lacey, G. (2011). A vision-based system for automatic hand washing quality assessment. *Machine Vision and Applications*, 22(2):219–234.

[Rautaray and Agrawal, 2015] Rautaray, S. S. and Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54.

[RCN, 2005] RCN (2005). Methicillin resistant staphylococcus aureus (mrsa): guidance for nursing staff. *http://www.nhs.uk/conditions/mrsa/documents/*.

[Sarkar et al., 2013] Sarkar, A. R., Sanyal, G., and Majumder, S. (2013). Hand gesture recognition systems: a survey. *International Journal of Computer Applications (0975–8887)*, 71(15).

[Scholkopft and Mullert, 1999] Scholkopft, B. and Mullert, K.-R. (1999). Fisher discriminant analysis with kernels. In *Proceedings of the 1999 IEEE Signal Processing Society Workshop Neural Networks for Signal Processing IX, Madison, WI, USA*, pages 23–25.

[Slama et al., 2014] Slama, R., Wannous, H., and Daoudi, M. (2014). Grassmannian representation of motion depth for 3d human gesture and action recognition. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 3499–3504. IEEE.

[Suarez and Murphy, 2012] Suarez, J. and Murphy, R. (2012). Hand gesture recognition with depth images: A review. In *RO-MAN, 2012 IEEE*, pages 411–417.

[Supancic III et al., 2015] Supancic III, J. S., Rogez, G., Yang, Y., Shotton, J., and Ramanan, D. (2015). Depth-based hand pose estimation: methods, data, and challenges. *arXiv preprint arXiv:1504.06378*.

[Suryanarayan et al., 2010] Suryanarayan, P., Subramanian, A., and Mandalapu, D. (2010). Dynamic hand pose recognition using depth data. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3105–3108. IEEE.

[Wang and Zhang, 2013] Wang, Y. and Zhang, L. (2013). 3d hand gesture recognition based on polar rotation feature and linear discriminant analysis. In *Intelligent Control and Information Processing (ICICIP), 2013 Fourth International Conference on*, pages 215–219. IEEE.

[WHO, 2009] WHO (2009). WHO guidelines on hand hygiene in health care. *Accessed via http://www.who.int/gpsc/5may/tools/9789241597906/en/*.

Architecture for Recognizing Stacked Box Objects for Automated Warehousing Robot System

Taiki Fuji, Nobutaka Kimura, and Kiyoto Ito

Hitachi, Ltd., Japan
taiki.fuji.mn@hitachi.com

Abstract

We present an architecture for recognizing stacked box objects for a robotic picking system where robots pick up an object in a warehouse. To pick up an object, a robot should recognize the object's position and pose. Additionally, there is a need to identify whether each object is a target object or not. In a common warehouse, most of the products are box objects. However, it is difficult to recognize the objects when they are neatly arranged on a shelf. In that case, poor 3D surface data such as that of a flat surface are obtained. Thus, conventional 3D object shape matching methods such as iterative closest points (ICP) and 3D point feature histogram (PFH) cannot be applied. In addition to the problem, a segmentation problem in which two boxes are recognized as one box can occur. Moreover, the robot has to put aside other boxes on a target object when the target object is placed under other boxes. Therefore, all positions and poses of boxes should be recognized. To solve the above mentioned problems, the labels that are attached to most of the boxes in warehouse are focused on. As a first step, all labels are detected by using color classification. Then, the box front face regions are estimated on the basis of the label's positions with the object classification. As a second step, the object's position and pose are estimated by using both the color and depth data of the box's front face regions including the labels. To show the effectiveness of our approach, we show the results of recognizing stacked boxes on a shelf and demonstrate a robotic picking system to which our recognition architecture is applied.

Keywords: Robot Vision, Robot Picking, Object Recognition, Automated Warehousing System

1 Introduction

New automated warehousing robot systems that include picking and conveying items have been brought to public attention. These systems are different from traditional automated warehousing systems, which are designed for specific and high-volume manufacturing such as automated storage and retrieval systems. The new systems deal with a lot of various kinds of products. The purpose of the systems is productive efficiency and flexibility corresponding to the seesaw between supply and demand.

Kiva Systems, owned by the Amazon group [kiv, 2015], is famous for demonstrating a new type of automated warehousing robot system that uses multiple automatic guided vehicles (AGVs). Amazon aims to promote the efficiency of warehouse work and downsize surplus manpower. Indeed, each AGV carries a shelf to a worker in a picking station, so the system reduces personnel cost. However, this is not a sufficient goal in the pursuit of efficiency. Amazon also says automated item picking is an important skill for many robot applications including warehousing, manufacturing, and service robotics. The goal of robotic automation is an alternative robot system for human picking work in which workers pick only required products. The "Amazon picking challenge" [ama, 2015] is a competition aimed to garner methods for creating such a picking system.

A robot cannot pick up a target object without details on the object's position and pose. Additionally, identification of objects is also needed to confirm whether the picked object is the target object or not. A variety

 ISBN 978-0-9934207-0-2

of object recognition methods are proposed such as using texture (RGB or gray) image base, depth image base, or both.

In a texture image base, descriptors of feature points and edge features are used. Key point descriptor base methods are suitable for texture objects. Scale-invariant feature transform (SIFT) [Lowe, 1999] is the most popular descriptor for object detection. In the case of texture-less objects, edge feature is preferred for object boundaries. A detection and tracking method for 3D texture-less objects that uses a particle filtering framework in real time [Choi et al., 2013] is proposed.

Recently, low-cost RGB-D cameras such as Kinect and Xtion have been used for many robotic and vision applications. These devices can get the color and depth data of a scene. Compared with 2D images, 3D data are more invariant to geometric changes. The iterative closest point (ICP) algorithm [Besl et al., 1992] is often used to register 3D point clouds. Moreover, the fast point feature histogram (FPFH) [Rusu et al., 2009] and viewpoint feature histogram (VFH) [Rusu et al., 2010] are proposed to be extended for geometry-based shape descriptors. There have also been numerous papers on using RGB-D data [Choi et al., 2012, Bo et al., 2011, Wnag et al., 2014, Tang et al., 2012]. These methods are proposed for bulk item picking with distinctive surfaces and colors.

The above mentioned methods are effective with objects that have rich variations in surface normals. Moreover, color features are used for enhancing the accuracy of target recognition. In a warehouse, box-type items such as cardboard boxes account for a substantial fraction of the products. That is, it is hard to obtain rich 3D surface data, though the face data of boxes can be obtained. Therefore, methods that need to have rich variations in surface normals are not suited for the box-shape objects.

In this paper, an architecture for recognizing neatly stacked box objects for robot picking is presented. Many shelf board spaces are stacked with some boxes that have the same shape and size but that differ in type from each other. In such cases, a segmentation problem with object recognition occurs when using a depth camera. This is because the camera position of a picking robot is needed to be set in front of a shelf. A shoe box is a typical example. Even the same kinds of boxes have different colors and sizes with respect to each manufacturer. Therefore, if a target box is put in the space between other boxes, there is a need to remove the distractions for the robot picking. Therefore, all objects that are in the camera frame need to be recognized.

1.1 Related Work

Many studies have been made on recognition methods with depth data. A point pair feature (PPF) that uses two points on surfaces and their normals with depth data is proposed [Drost et al., 2010]. This approach has been recently enhanced by incorporating the visibility context [Kim et al., 2011] or considering better boundary information [Woodford et al., 2014]. The surface point pair feature is efficient for objects that have rich variations in surface normals. However, simple surfaces like planar objects are not suited to recognition because a lot of different point pairs fall into the same hash slot.

Most of the effective studies have focused on recognition in cluttered spaces with a RGB-D sensor. There have also been numerous papers on using RGB-D data. An object recognition method that does not rely on assumption [Choi et al. 2012]. It can estimate the pose of a target object in a heavily cluttered scene. Bo et al. [Bo et al., 2011] also presented novel five depth kernel descriptors to capture different object cues including size, shape, and edge. Wang et al. [Wang et al., 2014] presented a novel framework consisting of a global object descriptor viewpoint oriented color-shape histogram, which combines color and shape information for both object recognition and highly accurate object pose retrieval. Tang et al. [Tang et al., 2012] presented an end-to-end instance recognition approach for textured objects. These studies also assume that the camera is set at the upper right positions in a clutter scene.

1.2 Contributions

The contributions of this work are the following. First, a box object recognition architecture is proposed for robot picking in which box objects are stacked on a shelf and a camera position is set in front of the shelf. Second, we demonstrate robot picking applied with our object recognition architecture.

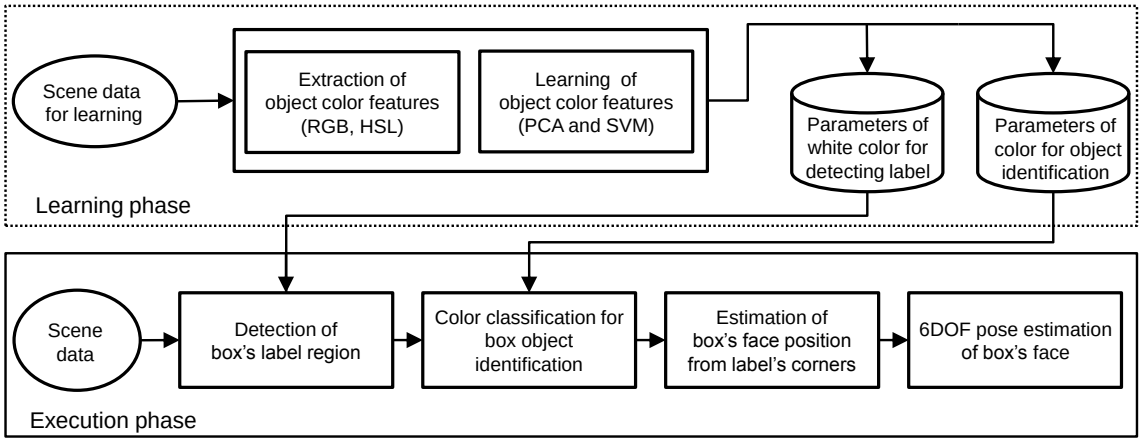


Figure 1: Proposed recognition architecture

2 Recognition Architecture for Stacked Box Objects

The purpose of the recognition is to identify whether detected objects are the target object or not and to estimate a box object's position and pose for robot picking. In a warehouse, most box-type items are labeled with merchandise information. Therefore, the labels are used as an effective way to recognize box objects placed on a shelf.

The proposed recognition architecture is illustrated in Figure 1. The proposed recognition architecture has a learning phase and execution phase. In the learning phase, label and box face color features are extracted and learned for obtaining the feature distribution and parameters of color classification for the object identification. In the execution phase, labels of box objects are first detected from a scene color frame. Next, the objects are identified and the box faces are estimated on the basis of the label positions. Finally, the object position and pose are estimated.

2.1 Learning Phase of Color Classification

In the learning phase, a principal component analysis (PCA) and support vector machine (SVM) are applied to the color classification. Previously, scene images of target objects are prepared. The scene images are obtained in different lighting conditions so that the color classification is robust to changes in lighting conditions. As color coordinate systems, RGB and HLS (hue, lightness, and saturation) are used for the learning of classification. HLS is the most common cylindrical-coordinate representation of points in an RGB color model. This coordinate system is more intuitive and perceptually relevant than an RGB cube. Therefore, the data dimension is six dimensional coordinates.

To reduce dimensionality and bring out a strong pattern of the data, the PCA is applied to the learning data represented in 6D color. The learning data include statistically redundant components. Therefore, dimensionality reduction means obtaining compact, accurate, and well represented data. PCA is a basis transformation for diagonalizing an estimate of the covariance matrix of the sample data $\boldsymbol{p}_{f,i}, i = 1,...,6, \boldsymbol{p}_{f,i} \in \Re^6, \sum_{i=1}^{6} \boldsymbol{p}_{f,i} = 0$ defined as

$$\boldsymbol{S} = \frac{1}{N} \sum_{i=1}^{6} \boldsymbol{p}_{f,i} \boldsymbol{p}_{f,i}^T. \tag{1}$$

An eigen problem has to be solved,

$$\boldsymbol{S}\boldsymbol{v} = \lambda \boldsymbol{v}, \tag{2}$$

where the eigenvalues $\lambda \geq 0$ and eigenvectors $\boldsymbol{v} \in \Re^6$. The number of dimensions is reduced from six to two-four by considering a cumulative contribution ratio of over 90%.

The object color is classified by using a soft-margin SVM [Bishop, 2006]. SVM is one of the major supervised learning algorithms for solving class separation problems. The basic form of an SVM classifier can be

 ISBN 978-0-9934207-0-2

expressed as

$$g(\boldsymbol{x}) = \boldsymbol{\omega}^T \phi(\boldsymbol{x}) + b, \tag{3}$$

where the input vector $\boldsymbol{x} \in \Re^N$ and $\boldsymbol{\omega}$ is a normal vector of a separating hyper-plane in the feature space produced from the mapping of a function $\phi(\boldsymbol{x})$. The sign of $g(\boldsymbol{x})$ indicates whether vector $\boldsymbol{x}$ is classified as label color or other, e.g., such as the color of the shelf board, or box colors. The training of SVM is defined as a constrained optimization problem:

$$\begin{aligned} \min \quad & Q(\boldsymbol{\omega}, b, \xi) = \frac{1}{2}||\boldsymbol{\omega}||^2 + C\sum_{i=1}^{N} \xi_i \\ s.t. \quad & y_i(\boldsymbol{\omega}^T \phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \forall i, \end{aligned} \tag{4}$$

where C is a trade-off parameter that controls the relative importance of minimizing the norm of $\boldsymbol{\omega}$ and satisfying the margin constraint for each data point. In this paper, radial basis function (RBF) kernel is used as the feature classifier. The C value is set to 45.0, and the gamma value of the kernel is set to 5.0 in empirical basis. Parameters obtained by the above processing are applied to the color classification in the execution phase.

2.2 Execution Phase of Identification and Pose Estimation

The process of the execution phase is the following steps (cf. Figure 2).

Step 1 Extract label candidate regions by judging labels' color (white) and region size
Step 2 Confirm whether the region is labeled or not and perform four direction judgment by template matching
Step 3 Classify the object's color around the labels by using PCA and SVM
Step 4 Estimate the box object position with corresponding front face corners on the basis of the label corners and depth data
Step 5 Calculate homography matrix $\boldsymbol{H}$ to estimate the object's x, y, and $roll$ angle
Step 6 Calculate the center of front face's depth data to estimate z (depth)
Step 7 Calculate the normal vector of the box front face to estimate the $pitch$ and yaw angle
Step 8 Overlay the object 3D shape model on the estimated position and pose
Step 9 Send the recognized data to the motion planning function for picking the target object

As a first step, label regions are detected with color classification and region size from a camera frame image. The image is masked by the depth data, assuming that the distance between a robot and the objects on a shelf is from 300 to 700 mm. The corner positions of a label candidate region are automatically numbered as Figure

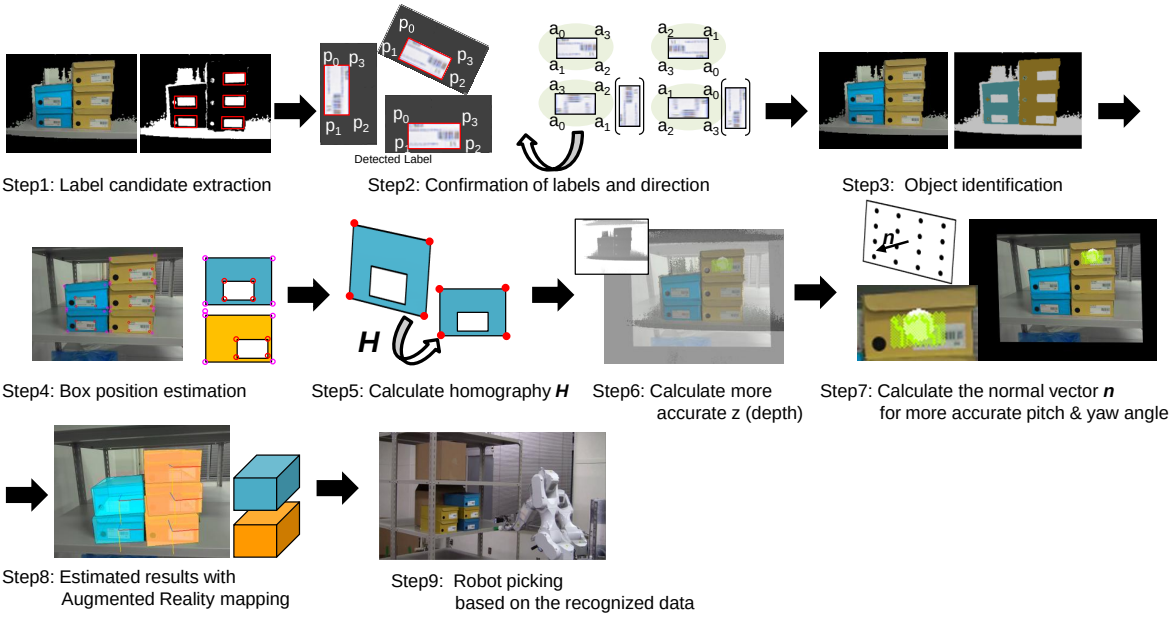


Figure 2: Identification and 6-DOF pose estimation of stacked box objects on a shelf

2 (Step 2, left). Therefore, the direction of a label candidate region is judged by template matching, that is, zero-mean normalized cross-correlation (ZNCC), for confirming the corresponding corners between p_i and $a_i(i = 0,..3)$. Next, the detected objects are identified with the color classification around the label region.

In pose estimation, RGB-D(depth) data are used. A box object's position and pose are estimated after obtaining the corner positions of the box's front face on the basis of the label's corner positions on a camera image, In this paper, the color data are used to estimate the object's position and pose with homography estimation. A perspective transformation between reference image points $\boldsymbol{M}$ and scene image points $\boldsymbol{m}$ on a target plane is related by a homography $\boldsymbol{H}$ with the following equation.

$$s\tilde{\boldsymbol{m}} = \boldsymbol{H}\tilde{\boldsymbol{M}} \tag{5}$$

$\boldsymbol{H}$ is calculated with RANSAC [Fischler and Bolles, 1981]. Then, $\boldsymbol{H}$ is decomposed to the rotation vector $\boldsymbol{r}$ and translation vector $\boldsymbol{t}$.

To enhance the accuracies of the z position, pitch, and yaw angles, the depth data are used for the estimation. The normal orientation is calculated with singular value decomposition (SVD). The best-fit plane minimizes the sum of the squared distances. Thus, the best-fit plane achieves

$$\min_{\|\boldsymbol{n}\|=1} \sum_{i=1}^{m} \|\boldsymbol{p}_i^T \boldsymbol{n} - d\mathbf{1}\|^2, \tag{6}$$

where $\boldsymbol{p}_i = (x_i, y_i, z_i)^T (i = 1, ..., m)$, $\boldsymbol{n} = (a, b, c)^T$, $|d|$ is the distance from the box object to the camera, and $\mathbf{1}$ is a vector of m ones. Finally, the recognition data are sent to the motion planning, and then, the robot picks up the items.

3 Robot Picking

In this section, the method for the one type of robot picking is indicated. As mentioned above, in a warehouse, a lot of boxes are neatly stored on shelves. If these boxes are tightly arranged, finger type grippers are unsuitable for the grasping because the fingers hit the other box objects. Therefore, adsorption-type grippers are suitable in this case. Moreover, dual-arm manipulation is effective at picking up an object.

Now, how does the robot grasp the boxes? One arm that has the absorption type gripper is moved following the vector perpendicular to the box's front plane as shown in Figure 3. Another arm is moved under the box for support to keep the balance of the box. Hence, object recognition of 6-DOF pose estimation is desired to get the center position of the box's front face and normal vectors for vertically adsorbing onto the face.

4 Experiments

To verify the effectiveness of the proposed recognition architecture applied to a robot system, the following experiments were conducted in a simulated warehouse. First, the recognition performance of the proposed architecture for stacked rectangular objects was verified to be applicable to a robotic picking demonstration. Second, the robustness of the proposed recognition architecture was confirmed in changing lighting conditions. Finally, by applying the recognition architecture to the picking robot, the effectiveness of the robot system, which includes the recognition and how the robot picking is performed, was verified.

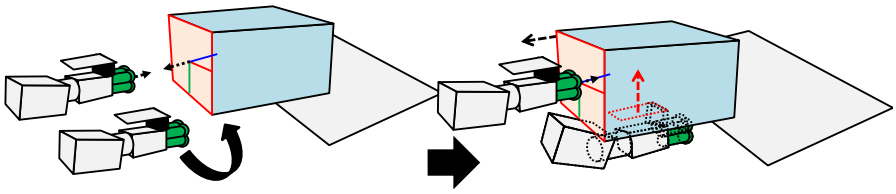


Figure 3: Method for robot picking with double manipulation

 ISBN 978-0-9934207-0-2

4.1 Experimental Conditions

The experiments were conducted with a robot picking system implemented with the proposed recognition architecture on a laptop PC with an Intel Core i7-4700MQ 2.4-GHz processor and 8.00 GB RAM. However, two CPU cores (four threads) were used in consideration of other robot functions. DepthSense325 from Softkinetic, which supports 25 fps, was used as a RGB-D camera. The color and depth resolutions were set at 640 × 480 and 320 × 240, respectively.

In object recognition experiments, stacked boxes on a shelf board were identified, and the position and pose were estimated. The RGB-D camera was attached to the robot manipulator, and the camera shooting position was set in front of the shelf. The distance between the camera and rectangular objects was around the 500 mm. Shoe boxes stored in up to three rows were used. There were two kinds of boxes.

4.2 Results and Discussion

Figure 4 shows examples of the recognition results with the AR technique. For reasons of internal company security, the colors of the boxes in this and the following figures were manually changed in the figures only. Two kinds of box objects were used, but the number of object kinds was able to be increased. As a result, the box objects placed on the shelf were detected even if the 3D shape data could not be obtained. Not only neatly stacked objects but also arbitrarily stacked objects whose pose was changed were able to be recognized. However, pitch angle errors sometimes varied widely. The cause of the errors is considered to be due to the lid of the shoe box protruding. The recognition misses for the box was due to the depth data being masked.

Table 1 indicates the recognition rate and time. The recognition rate was over 97[%]. It took 675 [msec]. on average to recognize the box objects. As for the position and pose estimation, the box objects were recognized in error range of less than ±10 [mm], ±5 [deg.], which is the acceptable error range when an adsorption gripper is used and the boxes are placed within ± 30 [deg.] rotation of yaw angle against to the camera direction. The position and pose error was calculated when the box objects were neatly stacked in front of the camera.

Next, the results of object recognition in changing illumination conditions are shown in Figure 5. In this experiment, six kinds of illumination changes were tried, considering any warehouse's lighting conditions. As a result, the proposed recognition architecture was able to recognize the box objects in the conditions.

Finally, the results of the robot picking demonstration with our recognition architecture applied to the robot are shown in Figure 6. Figure 6 shows the sequence of the robot picking. First, the robot took scene camera frame data. Then, the recognition process was conducted. After the recognition results were received, the box that was at the highest position in the vertical direction (y direction) was picked up by the double arms.

Figure 4: Recognition results of box objects

Table 1: Recognition rate and time

Recognition rate (Ave.)	Recognition time (Ave.)
97.4[%]	675 [msec.]

 ISBN 978-0-9934207-0-2

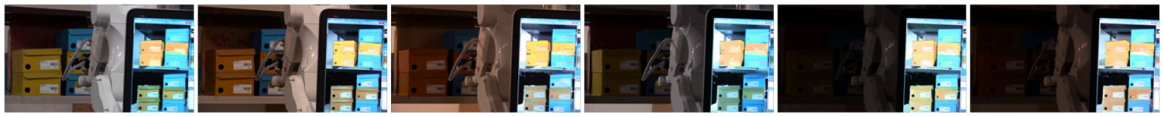


Figure 5: Recognition results in variety of lighting conditions

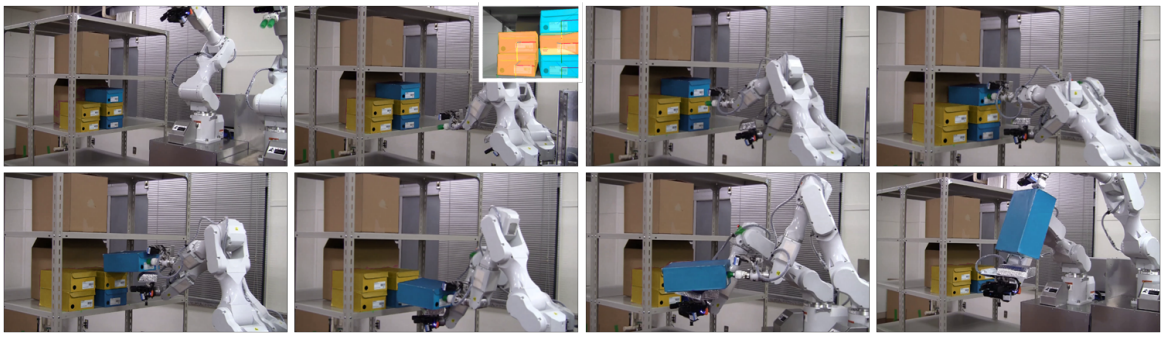


Figure 6: Results of robot picking sequence with object recognition architecture

The top of the absorption gripper, which was attached to one of the manipulators, was moved to the normal vector of the recognized box. Another manipulator, which had a board, was moved under the box. In this demonstration, the posture parameters (rotation of $pitch$, yaw, and $roll$) were set to zero for safe movement, but these recognition accuracies meet the performance required to be able to do the picking.

5 Conclusion

We presented an architecture for recognizing stacked box objects for automated warehousing robot systems with RGB-D camera data. Our approach learns each box object color and label color. In experiments, object identification and pose estimation were able to be conducted when box-type objects were neatly arranged on a shelf. Moreover, robot picking by double arm manipulation was demonstrated and successfully accomplished with our recognition approach. Hence, the effectiveness of the proposed recognition architecture was confirmed.

In future work, we will develop a method for recognizing multiple kinds of products in a warehouse. In addition to the recognition function, we will also develop a function for judging grasping positions depending on the application of the finger type grippers.

References

[ama, 2015] (2015). Amazon picking challenge. http://amazonpickingchallenge.org/.

[kiv, 2015] (2015). Kiva systems. http://www.kivasystems.com/.

[Bishop, 2006] Bishop, C. (2006). *Pattern recognition and machine learning*, volume 4. springer New York.

[Bo et al., 2011] Bo, L., Ren, X., and Fox, D. (2011). Depth kernel descriptors for object recognition. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 821–826. IEEE.

[Drost et al., 2010] Drost, B., Ulrich, M., Navab, N., and Ilic, S. (2010). Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 998–1005. IEEE.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on Computer vision*, volume 2, pages 1150–1157. IEEE.

[Rusu et al., 2009] Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *ICRA'09. IEEE International Conference on Robotics and Automation, 2009.*, pages 3212–3217. IEEE.

[Rusu et al., 2010] Rusu, R. B., Bradski, G., Thibaux, R., and Hsu, J. (2010). Fast 3d recognition and pose using the viewpoint feature histogram. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2155–2162. IEEE.

[Tang et al., 2012] Tang, J., Miller, S., Singh, A., and Abbeel, P. (2012). A textured object recognition pipeline for color and depth image data. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3467–3474. IEEE.

[Wang et al., 2014] Wang, W., Chen, L., Liu, Z., Kühnlenz, K., and Burschka, D. (2014). Textured/textureless object recognition and pose estimation using rgb-d image. *Journal of Real-Time Image Processing*, pages 1–16.

[Woodford et al., 2014] Woodford, O. J., Pham, M.-T., Maki, A., Perbet, F., and Stenger, B. (2014). Demisting the hough transform for 3d shape recognition and registration. *International Journal of Computer Vision*, 106(3):332–341.

Symmetry and Repeating Structure Detection

Musfira Jilani, Padraig Corcoran & Michela Bertolotto

School of Computer Science & Informatics
University College Dublin, Ireland

Abstract

Transformation voting is a general paradigm to symmetry detection based on the concept of accumulating evidence or votes to the existence of transformations between symmetrical elements. However implementing this paradigm involves a number of important design decisions which greatly influence performance. In this paper we articulate these design decisions and subsequently propose an implementation which makes a number of contributions. These include the first consideration and solution to the issues presented by transformation direction; the characterization of transformations in terms of their origin location which results in the detection of more meaningful symmetries; and, an effective measure of transformation similarity. Using symmetry detection as a platform, we also propose a method capable of detecting regular and irregular patterns of repeating structures where the problem is posed in terms of finding maximum cliques in a graph.

Keywords: Symmetry Detection, Transformation Voting, Graph Clustering.

1 Introduction

Symmetry is a ubiquitous phenomena exhibited in almost all man-made and natural environments. Consequently symmetry detection in geometrical models has become an important component in many applications including model completion, model manipulation, model synthesis and shape classification [Mitra et al., 2013]. An object may exhibit both within and between object symmetries. For example consider Figure 1 which contains the model of a cat in three different poses. Consider the pair of cats at the back of the figure. The cat on the right exhibits a within object symmetry between its left and right sides. That is, each side is related to the other by a reflection. The pairs of cats in this figure also exhibit a between object symmetry between their heads. That is, each head is related to the other by a rigid body transformation. Formally a symmetry contained in a model M may be defined as follows. Let the elements of $\{s_i, s_j\}$ be subsets of M. A symmetry exists between these two elements if $s_i = T_{ij}(s_j)$ for some transformation T_{ij}. A model may contain more than a single symmetry as is the case in the above example. A symmetry can be classified in terms of the group the corresponding transformation belongs to such as the Euclidean [Mitra et al., 2006], or reflection [Podolak et al., 2006] groups.

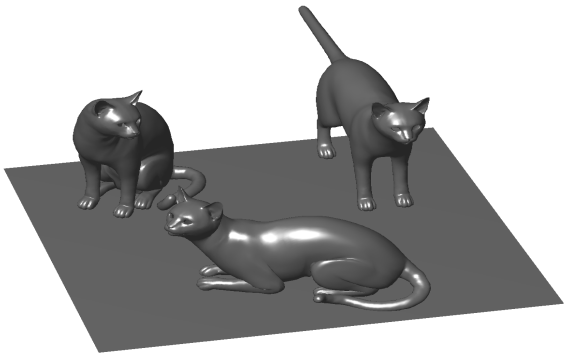


Figure 1: Symmetries and repeating geometrical structures are exhibited.

In this work we focus on detecting symmetries where the corresponding transformations belong to the Euclidean group generated by translations, rotations, and uniform scaling. A number of authors have proposed methods for detecting other types of symmetries. For example [Podolak et al., 2006] proposed a method for detecting reflection symmetries. For an overview of the different types of symmetries and corresponding methods for detecting them please see [Mitra et al., 2013]. Mitra et al. [Mitra et al., 2013] provides a detailed summary of existing methods for symmetry detection in 3D models and groups them into different general

 ISBN 978-0-9934207-0-2

paradigms. One of these paradigms, which Mitra et al. [Mitra et al., 2013] refers to as transformation space voting, attempts to accumulate evidence to the existence of transformations corresponding to symmetries. This paradigm represents a natural extension of the above standard definition of symmetry and, as such, there exist a number of corresponding implementations of it. However implementing this paradigm involves a number of important design decisions which greatly influence performance. In this paper we articulate these design decisions and subsequently propose an implementation which overcomes a number of limitations associated with existing implementations. The implementation in question detects symmetries where the corresponding transformations belong to the Euclidean group.

A super-class of symmetries are repeating structures. For example in Figure 1 the cat's head corresponds to a repeating structure which occurs three times. Similar to symmetry detection, repeating structure detection has many applications including shape classification and object discovery. Formally a repeating structure in a model M may be defined as follows. Let R be the set $\{r_1, r_2, \ldots, r_n\}$ such that $r_i \subset M$ for all i. R corresponds to a repeating structure if $r_i = T_{ij}(r_j)$ for all i, j where T_{ij} is a transformation. That is, a symmetry exists between all pairs of elements of R. In this work we also propose a method for detecting repeating structures which builds upon the proposed method for detecting symmetries.

The layout of this paper is as follows. Section 2 describes in detail our implementation of the transformation voting paradigm toward symmetry detection. Section 3 describes our proposed method for detecting repeating structures. In sections 4 and 5 we present results and draw conclusions respectively.

2 Symmetry Detection

The first implementation of the transformation space voting paradigm was proposed by Mitra et al. [Mitra et al., 2006].Pauly et al. [Pauly et al., 2008] subsequently used a slight variation of this method as a platform for detecting regular repeating structures. On a conceptual level methods belonging to this paradigm operate as follows. Keypoints are first detected and matched. Each such match generates a transformation between keypoints which in turn provides some evidence (referred to as a vote by Mitra et al. [Mitra et al., 2013]) to the existence of a corresponding symmetry. This evidence exists in the transformation space and it is necessary to perform some type grouping or clustering of this evidence to infer the existence of corresponding symmetries.

We will now describe the various steps involved in our implementation of the transformation voting paradigm towards symmetry detection. In particular we articulate the design decisions involved in implementing each of these steps and subsequently propose an implementation which overcomes a number of limitations associated with existing implementations.

2.1 Keypoint detection

Keypoints are defined as prominent points according to a particular definition of saliency and therefore are good candidates for repeated detection and subsequent matching [Tombari et al., 2013]. A number of authors have performed in-depth evaluations of existing keypoint detection methods for 3D meshes [Dutagaci et al., 2012, Tombari et al., 2013]. Also of note is the fact that many keypoint detection methods associate to each detected point a characteristic local scale which can be used in subsequent analysis such a feature description [Darom and Keller, 2012]. The symmetry detection methods of [Mitra et al., 2006] and [Pauly et al., 2008] do not perform keypoint detection. Instead Mitra et al. [Mitra et al., 2006] performs a random sampling of points while Pauly et al. [Pauly et al., 2008] performs a uniform sampling such that the average spacing between points is constant. The disadvantage of not having a keypoint detection stage is that points sampled may not be distinctive and, in turn, it may not be possible to subsequently reliably find correct matches.

In the proposed symmetry detection method we detect keypoints and a corresponding local scale using the method of Darom et al. [Darom and Keller, 2012] which detects keypoints as local maxima of the *Difference of Gaussians* (DoG) in scale and space. This method is an extension of the method by Lowe [Lowe, 2004], for detecting keypoints and a corresponding local scale in 2D images, to 3D meshes.

 ISBN 978-0-9934207-0-2

2.2 Keypoint description and matching

Following keypoint detection it is necessary to discover a set of matching keypoint pairs $K = \{(a_1, b_1), \ldots (a_m, b_m)\}$ such that for each pair the corresponding local neighborhoods have a similar shape. The matching of keypoints is made difficult by the fact that the local neighborhood of matching pairs may have slightly different shapes, be at different scales and orientated differently. To overcome these issues, for each keypoint, a descriptor is typically computed which is invariant to such changes, while still preserving discriminability between non-matching keypoints [Darom and Keller, 2012]. An indepth evaluation of existing point descriptor methods can be found in [Heider et al., 2012]. The symmetry detection methods of Mitra et al. [Mitra et al., 2006] and Pauly et al. [Pauly et al., 2008] both use point descriptors which are a function of principles curvatures approximated at the point in question. In the proposed symmetry detection method we use the keypoint descriptor proposed by Darom et al. [Darom and Keller, 2012] which is an extension of the 2D image SIFT descriptor [Lowe, 2004] to 3D meshes.

Once keypoints and their descriptors have been computed, the next step is to establish matches between keypoint pairs. Different *matching strategies* can be used to perform this task; an overview of these can be found in [Szeliski, 2011]. In the symmetry detection method of Mitra et al. [Mitra et al., 2006] the authors establish matches between all keypoint pairs within a given distance of each other defined using a suitable metric. Pauly et al. [Pauly et al., 2008] use a bag of words model and establishes matches between all keypoints pairs assigned to the same word. This approach is also used in the symmetry detection method proposed in this work. Specifically, codewords are generated using k-means clustering with k is set equal to 10% of the dataset size; the works of Knopp et al. [Knopp et al., 2010, Knopp et al., 2011] generate codewords in a similar manner.

2.3 Pose estimation and representation

A transformation T in the Euclidean group acting on a point x can be represented as a sequence of three transformations using Equation 1 where T_r represents a rotation transformation, T_s a scaling transformation and T_t a translation transformation.

$$T(x) = T_t(T_s(T_r(x))) \tag{1}$$

In the proposed symmetry detection method we compute and represent keypoint poses as follows. The position of each keypoint is computed to be the x, y and z coordinates of the point in question and this is represented by a three dimensional vector. The scale of each keypoint is computed using the method of Darom et al. [Darom and Keller, 2012] described in section 2.1 and is represented using a real number. The orientation is computed such that the z axis is assigned to the surface normal, the x axis is assigned to the cross product of the z axis and the eigenvector associated with the largest eigenvalue, and the y axis is assigned to the cross product of the x axis and the z axis. The eigenvalues and eigenvectors for each point are calculated in a neighborhood of size equal to the local scale estimated above. This method for assigning an orientation was shown by Petrelli et al. [Petrelli and Di Stefano, 2011] to perform well relative to other methods in terms of robustness (in their article Petrelli et al. [Petrelli and Di Stefano, 2011] refer to this method as EM). The orientation of each Keypoint is represented using a unit quaternion [Kuipers, 2002].

2.4 Transformation direction consideration

A single symmetry can be represented by two distinct transformations. For example consider Figure 2(a) which contains a symmetry between two triangles corresponding to a translation by n units in the x direction. This symmetry may be represented by a transformation from the left triangle to the right triangle (a translation vector of $(n, 0)$), or by a transformation from the right triangle to the left triangle (a translation vector of $(-n, 0)$). Therefore any pair of matching keypoints (a, b) resulting from an underlying symmetry corresponds to two distinct transformation directions. That is, a transformation from a to b and a transformation from b to a. In the context of implementing the transformation voting paradigm to symmetry detection this property presents some challenges which must be considered. On one extreme, since each symmetry is represented by

 ISBN 978-0-9934207-0-2

two transformations this evidence may be divided amongst two distinct locations in the transformation space and result in a missed detection. On the other extreme, a naive analysis of the evidence may result in a double detection of a single symmetry.

Our solution to this problem involves defining a method which for a given symmetry consistently chooses a single transformation direction, which we refer to as the dominant transformation direction, between all corresponding matching keypoints. For example consider Figure 2(b) where three pairs of keypoints, corresponding to triangle corners, have been matched. In this case the same transformation direction, corresponding to positive translation in the x direction, has been assigned to each match. Symmetries can subsequent be detected, while avoiding the issues above, by applying a standard clustering method to the transformation space. This solution is inspired by keypoint detection techniques which assign a dominant orientation to each keypoint [Lowe, 2004]. The symmetry detection method of Mitra et al. [Mitra et al., 2006] does not consider the issues transformation direction presents.

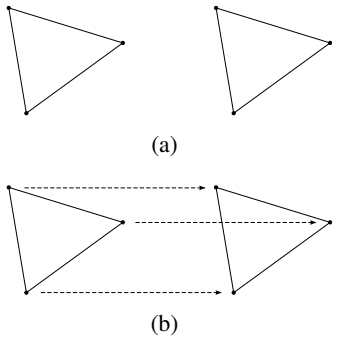


Figure 2: A symmetry corresponding to a translation in the x direction is depicted in (a). For each matching pair of keypoints in (b) a single equal transformation direction, represented by arrows, is assigned.

2.5 Transformation representation

Every transformation between matched keypoint pairs provides some evidence (referred to a as vote by [Mitra et al., 2013]) to the existence of a corresponding symmetry. Therefore the grouping or clustering of such transformations allows one to accumulate such evidence. However due to the nature of biological growth, manufacturing imprecision, stochastic process or sensor noise, symmetries are generally approximate as opposed to exact. Even if they were exact, slight inaccuracies in the keypoint pose estimation would result in a symmetry generating a set of similar but unequal transformations. Therefore it is important to represent transformations in a manner which subsequently allows an accurate measure of similarity to be defined. In the symmetry detection method of Mitra et al. [Mitra et al., 2006], the authors represent a given transformation by seven dimensional vector containing a scaling element, three translation elements and three rotation elements. The rotation elements in question correspond to Euler angles. The major drawback of this representation is that the values of the Euler angles depend on the order of rotations about the three principle axes. That is, it depends on a representation that is not unique [Huynh, 2009]. In the symmetry detection method of Pauly et al. [Pauly et al., 2008] the authors represent represent transformations in a similar manner to Mitra et al. [Mitra et al., 2006] except that rotations are represented using a rotation axis and angle. This representation is rarely used in practice; Dunn et al. [Dunn and Parberry, 2011] refers to it as a "conceptual tool" that "gets relatively little direct use compared to the other formats".

In this work we represent a transformation between a keypoint pair (a, b), where a and b are the origin and destination of the transformation respectively, as follows. The representation contains rotation, scaling, translation and origin components which we discuss in turn. Let q_a and q_b be the unit quaternion representing the orientations of the keypoints a and b respectively. We represent the rotation component of the transformation from a to b as a unit quaternion t_q which is computed using Equation 2 where a_b^{-1} is the inverse of quaternion a_b. As stated by Horn [Horn, 1986], unit quaternions provide a "representation such that the space of rotations allows a metric to be defined on it". As such, they provide an effective platform for defining the similarity between transformation rotations.

$$t_q = b_q a_b^{-1} \tag{2}$$

Let a_s and b_s be real numbers representing the local scale values for keypoints a and b respectively (see

 ISBN 978-0-9934207-0-2

section 2.1). We represent the scaling component of the transformation from a to b as a real number t_s which is computed using Equation 3.

$$t_s = \frac{a_s}{b_s} \tag{3}$$

Let a_l and b_l be three dimensional vectors representing the spatial locations of a and b respectively. We represent the translation component of the transformation from a to b as a vector t_t which is computed using Equation 4 where $t_q b_l t_q^{-1}$ represents the rotation of b_l by t_q [Kuipers, 2002].

$$t_t = \left(t_q b_l t_q^{-1}\right) t_s - a_l \tag{4}$$

We define an additional transformation component corresponding to the origin of the transformation from a to b as a vector t_o which is computed using Equation 5.

$$t_o = a_l \tag{5}$$

2.6 Transformation similarity measure

Towards clustering transformations it is necessary to define a measure of transformation similarity. In the symmetry detection method of Mitra et al. [Mitra et al., 2006], the authors represent a transformation using a seven dimensional vector. A norm is defined for this vector and used to measure the similarity of two transformations. Pauly et al. [Pauly et al., 2008] uses a decision tree approach to determining transformation similarity.

In the proposed work a transformation is represented by the four components t_q, t_s, t_t and t_o defined in section 2.5. In order to combine these components into a single distance value we define a metric over each and combine the corresponding individual distances using a probabilistic interpretation. Let t_q^x and t_q^y be quaternions corresponding to the rotation components of two transformation x and y as defined by Equation 2. The angular distance between these quaternions is represented by the term $d_q^{x,y}$ and is computed by Equation 6; this measure of distance has been proven to be a metric [Huynh, 2009].

$$d_q^{x,y} = 2\arccos\left(|x \cdot y|\right) \tag{6}$$

Similarly we define the metrics $d_s^{x,y}$, $d_t^{x,y}$ and $d_o^{x,y}$ on the transformation components t_s, t_t and t_o respectively to be the Euclidean distance. We combine the above metrics using Equation 7 where κ is the Gaussian kernel defined in Equation 8, and σ_q, σ_s, σ_t and σ_o are corresponding Gaussian scale parameters for each metric.

$$d^{x,y} = \kappa\left(d_q^{x,y}, \sigma_q\right)\kappa\left(d_s^{x,y}, \sigma_s\right)\kappa\left(d_t^{x,y}, \sigma_t\right)\kappa\left(d_o^{x,y}, \sigma_o\right) \tag{7}$$

$$\kappa\left(d, \sigma\right) = \exp\left(\frac{-d^2}{2\sigma^2}\right) \tag{8}$$

The kernel κ returns a value in the interval $[0, 1]$ and can be interpreted as the probability that the corresponding components of the transformations x and y are equal. Consequently the value $d^{x,y}$ lies in the interval $[0, 1]$ and can be interpreted as the probability that transformations x and y are equal.

2.7 Symmetry detection via clustering

In this section we describe the approach used to detect groups or clusters of similar transformations corresponding to symmetries. The symmetry detection methods of [Mitra et al., 2006] and [Pauly et al., 2008] perform clustering using mean-shift clustering. In this work we use a graph clustering, also known as community detection, method to perform clustering. The aim of graph clustering is to detect highly connected sub-graphs or clusters contained in a given graph. A detailed review of graph clustering can be found in [Fortunato, 2010]. We

perform graph clustering in two steps. Firstly we construct an undirected unweighted graph $G_s = (V_s, E_s)$. Here V_s is a set of vertices representing transformations between matched keypoint pairs (see section 2.5) and E_s is a set of edges between these vertices where the corresponding transformations are deemed similar (see section 2.6). Two transformations x and y are deemed similar if $d^{x,y}$, as defined in Equation 7, is above a threshold t_s. Clusters are detected in the graph G_s using the graph clustering method of [Lancichinetti et al., 2011]. This method detects clusters which are statistically significant with respect to a random null model, specifically the configuration model. This method is capable of detecting overlapping clusters. Although this method returns a hierarchy of clusters, in this work we only consider the lowest level of the hierarchy.

Each cluster detected by the above algorithm corresponds to a set of transformations and in turn a symmetry. We represent the ith detected symmetry as a two element set $\{p_1^i, p_2^i\}$ where one element, say p_1^i, is the corresponding set of transformation origin keypoints and the other element, say p_2^i, is the corresponding set of transformation destination keypoints. We define P to be the set representing each of the n detected clusters; that is $P = \{\{p_1^1, p_2^1\}, \ldots, \{p_1^n, p_2^n\}\}$. Mitra et al. [Mitra et al., 2013] describes a number of alternative ways to representing detected symmetries such as segmentations.

3 Repeating structure detection

The task of repeating structure detection is to detect a set $R = \{r_1, r_2, \ldots, r_n\}$ such that $r_i = T_{ij}(r_j)$ for all i, j where T_{ij} is a transformation. That is, a symmetry exists between all pairs of elements of R. For example consider Figure 3. For each pair of squares or triangles a symmetry exists and is represented by a set of arrows pointing in a single direction. This implies that the square and triangle are repeating structures. Repeating structures can be classified as regular or irregular. Regular repeating structures occur when the symmetries between elements of R exhibit a regular structure in the transformation space. Pauly et al. [Pauly et al., 2008] developed a method for detecting such regular repeating structures. Irregular repeating structures occur when the symmetries between elements of R do not exhibit a regular structure in the transformation space.

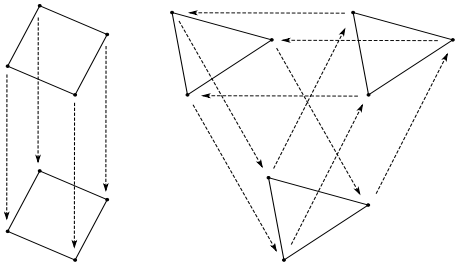


Figure 3: Symmetries and repeating geometrical structures are exhibited.

In this work we propose a general method capable of detecting both regular and irregular repeating structures. We pose the problem of detecting repeating structures as one of finding maximum cliques in a graph. Let P be the representation of symmetries defined in section 2.7 and b_j^i be the axis-aligned minimum bounding box of the keypoint locations p_j^i. We construct an unweighted undirected graph $G_r = (V_r, E_r)$ such that each element p_j^i of P is assigned to a single vertex $v \in V_r$. Two elements p_j^i and p_l^k are assigned to the same vertex if their corresponding bounding boxes intersect; that is, if $b_j^i \cap b_l^k \neq \emptyset$. For each $\{p_1^i, p_2^i\} \in P$ an edge $e \in E_r$ is constructed between the pair of vertices in G_r to which p_1^i and p_2^i are assigned. However attempting to discover exact cliques in G_r will result in a detection method which is not robust to failures in detecting all corresponding symmetries. We therefore extract approximate cliques as connected components in the graph G_r.

4 Results

In this section we demonstrate the effectiveness of the proposed symmetry and repeating structure detection methods. We applied the proposed methods to meshes generated using the *TOSCA high-resolution* dataset [Bronstein et al., 2008]. This dataset contains nine different articulated objects in a variety of poses. We chose this dataset because the keypoint detection and description techniques used within our methods were shown to perform well on it in the context of a number of applications including object retrieval. We created

test datasets by selecting a number of different objects in different poses and integrated by placing them on a level surface. Let us define an object part as a connected subset of an object which is rigid under changes in pose. Using this definition the proposed symmetry detection method will detect pairs of matching object parts and proposed repeating structure detection method will detect sets of matching object parts. Although Mitra et al. [Mitra et al., 2006] applied their symmetry detection method to an object, specifically a horse, in two different poses, the authors constrained keypoint matches to be between the different poses. That is, they reduced the complexity of the problem by exploiting a given object segmentation. In our work we do not assume any object segmentation and allow both within and between object matching keypoints. Suitable parameter values for our proposed symmetry and repeating structure detection methods were determined by trial and error where visual inspection of corresponding results was performed.

Figure 4(a) shows one test dataset used in our evaluation which contains the model of a cat in three different poses and the model of a wolf in two different poses. This mesh contains a total of 92,370 vertices. Applying the proposed symmetry detection method resulted in the detection of 28 symmetries. One such symmetry which exists between the cat's head is displayed in Figure 4(b) where keypoints belonging to each element of the symmetry are represented by a unique color. The remaining detected symmetries corresponded to symmetries between the wolf's head, the cat's head, the wolf's feet, the cats feet, the wolf's tail and the cat's tail. All detected symmetries were meaningful. Applying the proposed repeating structure detection method resulted in the detection of 9 repeating structures. Two such repeating structures which correspond to the cats's head and left feet are, where each occurrence of a reoccurring structure is represented by a unique color, are displayed in Figure 4(c) and Figure 4(d) respectively. The remaining detected repeating structures were all meaningful corresponded to the wolf's head, the cat's left right feet, the wolf's four individual feet and the wolf's tail. The proposed symmetry detection method does not detect symmetries corresponding to reflection transformations. The detection of two distinct repeating structures corresponding to the left and right cat's feet can be attributed to this fact. It is important to note that no symmetry or repeating structure corresponding to the cat's or wolf's torso was detected. This can be attributed to the fact that these regions are very smooth and not salient at a small scale.

5 Conclusions

This paper presents a state of the art implementation of the transformation voting paradigm to symmetry detection and a subsequent application of this to repeating structure detection. Relative to existing works the research presented in this paper makes the following four major contributions:

1. It represents the first work to consider and provide a solution to the issues presented by transformation direction.
2. It proposes a more suitable keypoint pose representation which allows transformations between keypoints to be more effectively represented and compared. It also characterizes transformations in terms their origin which allows for more meaningful symmetry detection.
3. It proposes a novel method for detecting regular and irregular repeating structures which poses the task as a problem of finding maximum cliques in a graph.

Despite these advances a number of challenges and possible future research directions remains. Some of the more pressing of these include the following. As discussed in the results section of this paper the proposed symmetry detection method fails to detect symmetries between regions which do not contain small scale salient features. This is a fundamental limitation of current implementations of the transformation voting paradigm which use matching keypoints to define transformations. To overcome this issue it is necessary to incorporate more global features such as segments or contours.

References

[Bronstein et al., 2008] Bronstein, A. M., Bronstein, M., Bronstein, M. M., and Kimmel, R. (2008). *Numerical geometry of non-rigid shapes*. Springer.

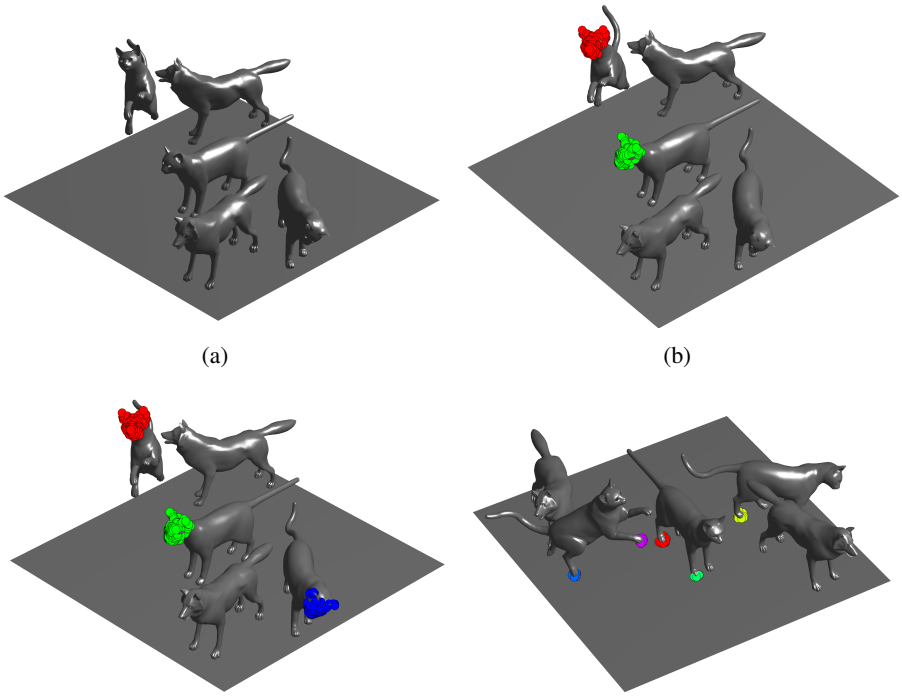


Figure 4: For the model in (a), (b) represents a detected symmetry while (c) and (d) represent detected reoccurring structures. Each element of the symmetry in (b) is represented by a unique color. Likewise each occurrence of repeating structures in (c) and (d) is represented by a unique color.

[Darom and Keller, 2012] Darom, T. and Keller, Y. (2012). Scale-invariant features for 3-d mesh models. *IEEE Transactions on Image Processing*, 21(5):2758–2769.

[Dunn and Parberry, 2011] Dunn, F. and Parberry, I. (2011). *3D math primer for graphics and game development*. CRC Press.

[Dutagaci et al., 2012] Dutagaci, H., Cheung, C. P., and Godil, A. (2012). Evaluation of 3d interest point detection techniques via human-generated ground truth. *The Visual Computer*, 28(9):901–917.

[Fortunato, 2010] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174.

[Heider et al., 2012] Heider, P., Pierre-Pierre, A., Li, R., Mueller, R., and Grimm, C. (2012). Comparing local shape descriptors. *The Visual Computer*, 28(9):919–929.

[Horn, 1986] Horn, B. K. P. (1986). *Robot vision*. the MIT Press.

[Huynh, 2009] Huynh, D. Q. (2009). Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164.

[Knopp et al., 2011] Knopp, J., Prasad, M., and Gool, L. V. (2011). Scene cut: Class-specific object detection and segmentation in 3d scenes. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pages 180–187. IEEE.

[Knopp et al., 2010] Knopp, J., Prasad, M., Willems, G., Timofte, R., and Van Gool, L. (2010). Hough transform and 3d surf for robust three dimensional classification. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV'10, pages 589–602, Berlin, Heidelberg. Springer-Verlag.

[Kuipers, 2002] Kuipers, J. B. (2002). *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press.

[Lancichinetti et al., 2011] Lancichinetti, A., Radicchi, F., Ramasco, J. J., and Fortunato, S. (2011). Finding statistically significant communities in networks. *PloS one*, 6(4):e18961.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[Mitra et al., 2006] Mitra, N. J., Guibas, L. J., and Pauly, M. (2006). Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (TOG)*, 25(3):560–568.

[Mitra et al., 2013] Mitra, N. J., Pauly, M., Wand, M., and Ceylan, D. (2013). Symmetry in 3d geometry: Extraction and applications. In *Computer Graphics Forum*. Wiley Online Library.

[Pauly et al., 2008] Pauly, M., Mitra, N. J., Wallner, J., Pottmann, H., and Guibas, L. J. (2008). Discovering structural regularity in 3d geometry. In *ACM Transactions on Graphics (TOG)*, volume 27, page 43. ACM.

[Petrelli and Di Stefano, 2011] Petrelli, A. and Di Stefano, L. (2011). On the repeatability of the local reference frame for partial shape matching. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2244–2251. IEEE.

[Podolak et al., 2006] Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., and Funkhouser, T. (2006). A planar-reflective symmetry transform for 3d shapes. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 549–559. ACM.

[Szeliski, 2011] Szeliski, R. (2011). *Computer vision: algorithms and applications*. Springer.

[Tombari et al., 2013] Tombari, F., Salti, S., and Di Stefano, L. (2013). Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision*, pages 1–23.

Bayer Interpolation with Skip Mode

D.G. Bailey, M. Contreras & G. Sen Gupta

School of Engineering and Advanced Technology
Massey University
Palmerston North, New Zealand

Abstract

The Bayer patterned colour filter array is commonly used with single chip cameras, with bilinear interpolation commonly used to demosaic the raw image to form a full colour image. When skip mode is used for sensor readout, the pixels are read out in 2×2 blocks. This requires modifying the interpolation weights. Several linear and bilinear interpolation schemes are compared. It is shown that with the output pixels located in the corners of the 2×2 Bayer blocks (for a half resolution output image) produces the best reconstruction in terms of PSNR, and output pixels located in the centre of the Bayer blocks gives the best visual reconstruction.

Keywords: CMOS camera, Bayer interpolation, Skipping mode, Demosaicing, Artefacts

1 Introduction

Most low cost colour cameras use a single chip where each pixel captures only a single colour channel of the colour image. This is achieved by integrating an array of colour filters, one for each pixel, when the sensor is fabricated. The most commonly used colour filter array is the Bayer pattern [Bayer, 1976] although other filter patterns are possible [Parulski, 1985]. The Bayer pattern, shown in Figure 1, has half green pixels, quarter blue pixels and quarter red pixels. To form a full colour image, it is necessary to interpolate the missing components from those that are available. This interpolation process is also known as demosaicing.

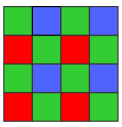


Figure 1: Colour filter array for the Bayer pattern

There is a trade-off between computational complexity and the quality of accuracy of the results. Three types of artefacts are commonly encountered from demosaicing:

- Blurring. The reduced sampling of channels results in a loss of fine details. Recovering the missing details by interpolating across edges will blur those edges.
- Colour bleeding. The samples for the different channels are in different locations. Interpolating them independently can result in the different colour channels being interpolated differently, resulting in unnatural colours, particularly visible around edges.
- Zipper effect. The alternating rows and columns within the colour filter array pattern can result in an alternating colour pattern most visible around edges and lines. This alternating pattern looks a little like a zipper.

 ISBN 978-0-9934207-0-2

The simplest interpolation approach is nearest neighbour interpolation. This uses a 2×2 window to copy the value from the nearest previously available pixel for a given colour channel. Since each colour channel is shifted slightly differently, this results can result in colour bleeding around edges.

The next simplest approach is bilinear interpolation, which averages the pixels on each side of the unknown channels using a 3×3 window. This gives significantly better results than nearest neighbour interpolation, while still being relatively simple to implement. However the quincunx sampling of the green channel is quite different from that of the red and blue, giving zipper artefacts around edges.

A simple improvement over basic bilinear interpolation is to detect the predominant orientation of any edges within the green channel (by taking the difference at red and blue pixels between the horizontally adjacent and vertically adjacent green pixels), and interpolating along the edges rather than across the edge. This removes much of the zipper artefact, and can reduce the blurring. For this reason, edge directed bilinear interpolation has probably become one of the most commonly used demosaicing algorithms.

There is a wide range of more advanced methods (see [Lu and Tan, 2003, Gunturk et al., 2002, Hsia, 2004, Li et al., 2008, Li and Randhawa, 2009]), which aim to improve the quality by reducing the artefacts. These generally require a larger window size, and result in increased computational complexity.

1.1 Camera Readout Modes

Low cost cameras are usually made using CMOS fabrication technology, and can result in quite large resolution sensors being available at relatively low cost. Since the pixels within a CMOS sensor are potentially individually addressable, sensor manufacturers have introduced a number of readout modes which enable a subset of the pixels to be read. This has the potential of reducing the resolution of the output image with the advantage of increasing the frame rate of the camera.

Windowing reads out only a rectangular subset of the pixels within the camera, directly reducing the image resolution. Only the pixels with a small portion of the complete field of view are read out. To read out the same field of view as before, a significantly shorter focal length lens must be used. Such short focal length lenses are prone to lens distortion.

Another readout mode available is skip mode [Fossum, 1997] where every second (or third, etc) pixel and row is read out. Such subsampling reduces the resolution without affecting the field of view, enabling the existing lens to be used. For single chip colour sensors, however, the skipping pattern has to take into account the pattern associated with the colour filter array. Taking every second pixel on every second row will only output a single colour channel. Therefore, for colour sensors the skip mode operates at a granularity of a 2×2 block rather than individual pixels [Micron, 2006]. The corresponding readout pattern is shown in Figure 2.

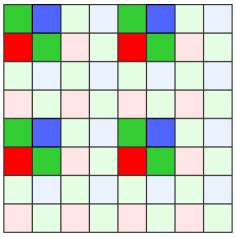


Figure 2: Pixel access pattern when accessing every second pixel in skip mode, with pale pixels not read out

When interpolating an image with skipping, the bilinear interpolation weights have to be adjusted to give evenly spaced output pixels. The best approach for accomplishing this has not been discussed before in the research literature. Different approaches for demosaicing with skipping every second pixel will be investigated and compared within this paper. Section 2 outlines the different interpolation strategies that will be considered. The methodology for comparing the different schemes is described in Section 3. The results are analysed in Section 4 before drawing conclusions in Section 5.

 ISBN 978-0-9934207-0-2

2 Bayer Interpolation Strategies

Using standard bilinear interpolation without skipping is clearly inappropriate because even rows and even columns are offset by 1 input pixel (or half an output pixel). The weights of the input pixels will clearly need to be adjusted. The fist question is where, relative to the input samples, should the output pixels be located. The most obvious choice is pixel centred, with the output aligned with one of the pixels (see the left pattern in Figure 2). Alternatively, the pixels could be aligned with the corners of the input pixels. Here there are two possible configurations or strategies (see Figure 2). The pixels can either be centred on the centre of a 2×2 block, or on the corners of the block. Other configurations are possible, but lack symmetry so will not be considered.

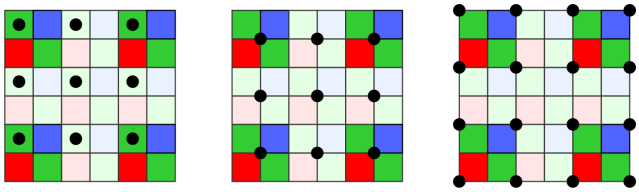


Figure 3: Positioning of the output samples. Black dots show the centres of the output pixels. From the left: pixel-centred, block-centred, and block-cornered configurations

The performance of each of these configurations will be considered. The grouping of pixels in blocks means that to access all of the neighbouring pixels required for bilinear interpolation, a 4×4 window in the input image is needed (or 5×5 for the block-cornered method). For the red and blue channels, the pixels are available on a regular grid, making bilinear interpolation straight-forward to estimate the value at arbitrary locations. For the green channel, there are two such grids. The grid with a pixel closer to the output position will generally give the better interpolation results. In some cases this is not clear (in particular with the block-cornered scheme), so the interpolation results from the two grids are averaged.

2.1 Pixel-centred

The weights for the pixel-centred configuration for the three colour channels for each of the output pixel samples are shown in Figure 4. Of the four pixels within the 2×2 block to align the output image to, we have chosen the green pixel on the first row of the block. This was chosen because there are twice as many green pixels as red and blue. Selecting the other green pixel would give similar results because of symmetry. Selecting either the red or blue, while possible, lacks symmetry in that the red and blue channels would require different sets of weights. In the results, this scheme is labelled *PC*.

Note that for the green channel, in the second and third column, there are simpler horizontal and vertical averages available as shown in Figure 5 (labelled *PCL*), interpolating using the nearest pixel gives a lower error. An alternative to consider using the nearest green is interpolating linearly along the diagonals, as shown Figure 5. This method is labelled *PCD*.

2.2 Block-centred

The weights for the block-centred configuration are shown in Figure 6. For the green channel, averages are taken between the nearest two green pixels. For the red and blue channels, the weights of the four adjacent input pixels are determined using bilinear interpolation. In the results, this scheme is labelled *BC*.

For the red and blue pixels in the centre of the block, it is also possible to interpolate simply along the diagonal as shown in Figure 7. The effect of this simplification will also be considered, and is labelled *BCD*.

2.3 Block-cornered

The final configuration considered is the block-cornered configuration, with the weights shown in Figure 8. In the results, this scheme is labelled *BN*.

 ISBN 978-0-9934207-0-2

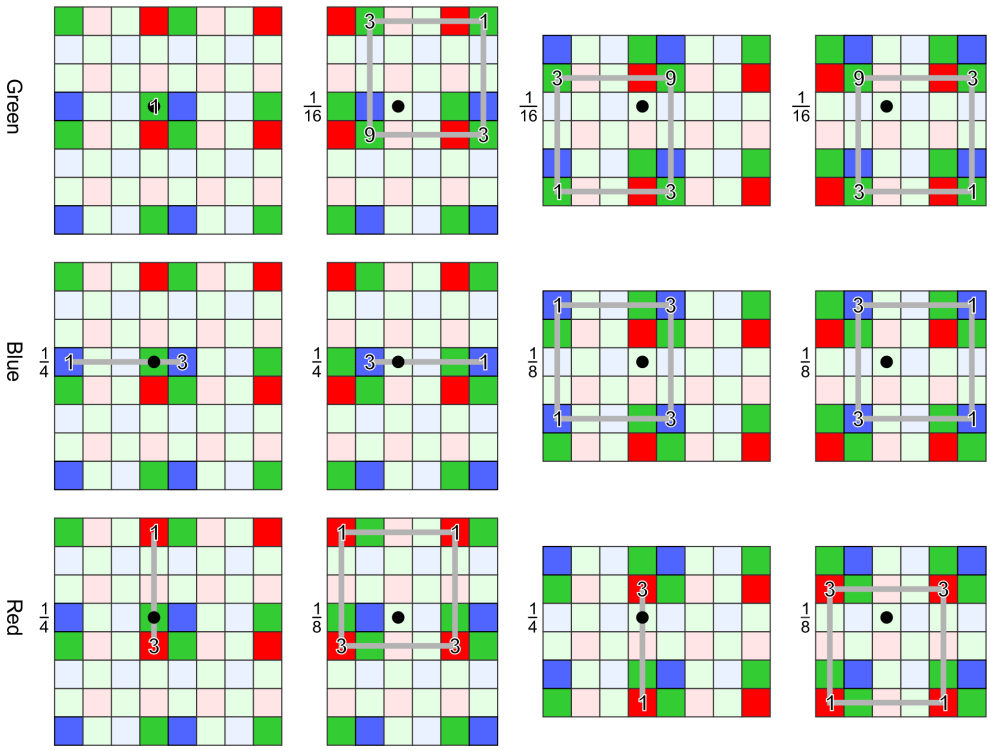


Figure 4: Weights for the pixel-centred scheme (*PC*)

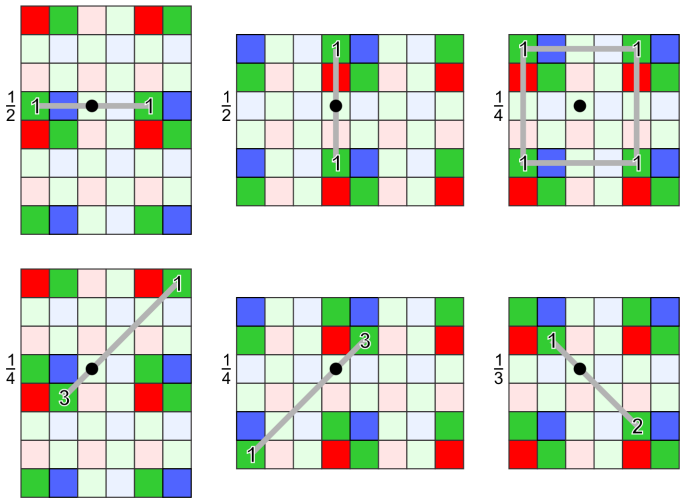


Figure 5: Alternative weights for the pixel-centred green channel (upper row: *PCL*, lower row: *PCD*)

For some of the outputs, a diagonal linear interpolation is also possible. The corresponding weights for these are shown on the left of Figure 9. Applying this to the green channel is labelled *BND1G* in the results, while extending it to the red and blue channels as well is labelled *BND1*. For the other green output pixels, it is also possible to make use of symmetry to perform a diagonal interpolation, as shown in the right of Figure 9. This is labelled *BND2G*.

3 Comparison Framework

To be able to objectively assess the accuracy of the demosaicing, it is necessary to have ground truth data. An image dataset commonly used for testing demosaicing algorithms is the Kodak photo CD image set. This is a set of full colour images [Kodak, 1991] of a range of scenes with varying amounts of fine detail and

 ISBN 978-0-9934207-0-2

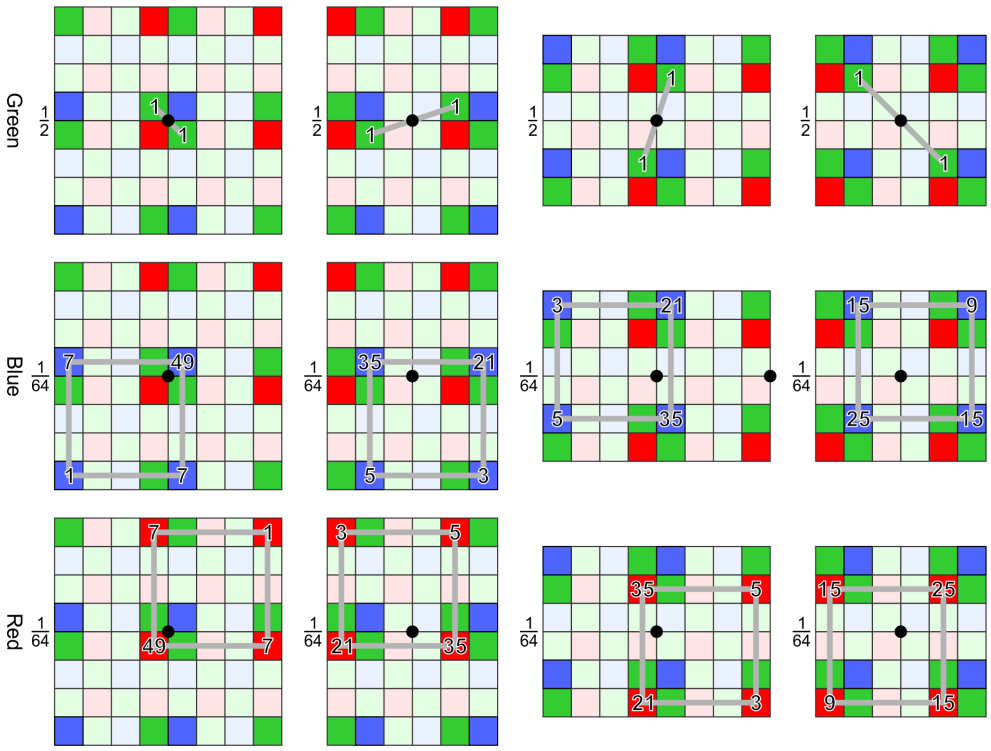


Figure 6: Weights for the block-centred scheme (*BC*)

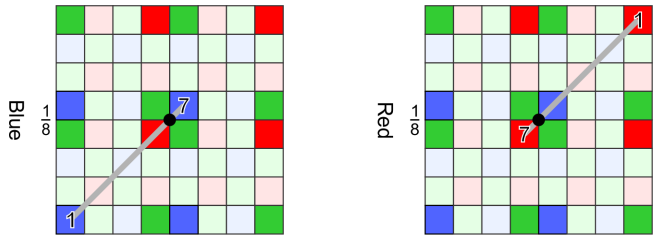


Figure 7: Alternative weights for the block-centred red and blue channels for diagonal interpolation (*BCD*)

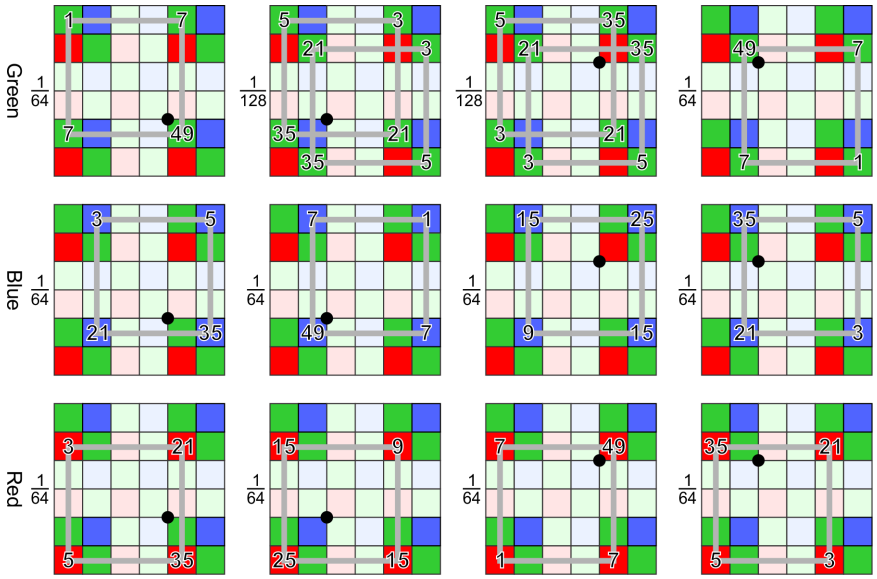


Figure 8: Weights for the block-cornered scheme (*BN*)

 ISBN 978-0-9934207-0-2

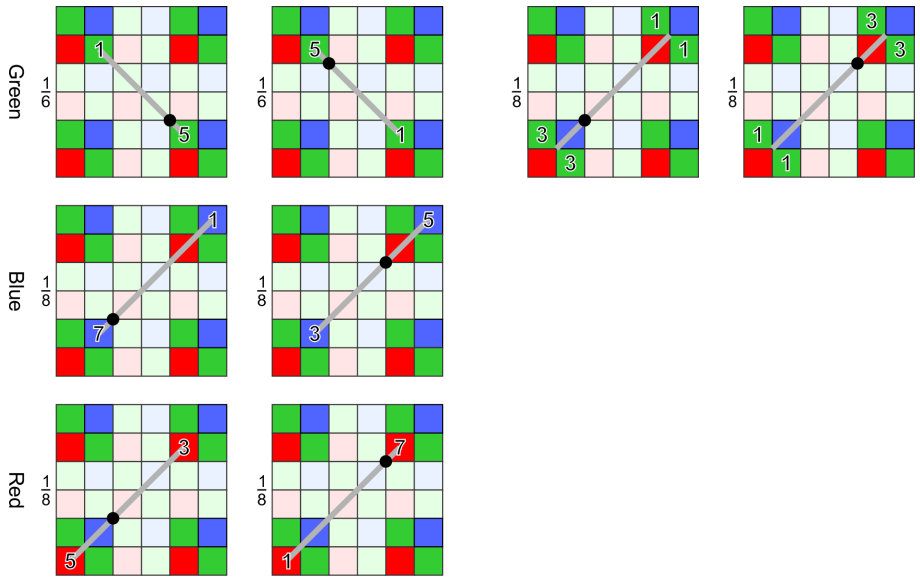


Figure 9: Alternative weights for the block-cornered scheme using diagonal interpolation (left *BND1G* and *BND1*, right *BND2G*)

structure. From these, image capture by a single chip sensor can be simulated. Since the unsensed colour channels are available in the original image, the demosaiced image can be compared with the original, and the error objectively determined. Let I be the original input image, downsampled by the skip factor, and R be the reconstructed image after demosaicing, then

$$E_{RMS} = \sqrt{\frac{1}{N}\sum^{N}(R-I)^2} \tag{1}$$

where N is the number of pixels in the image. From this, the $PSNR$ in dB can be derived

$$PSNR = 20\log_{10}\left(\frac{255}{E_{RMS}}\right) \tag{2}$$

One limitation of the PSNR is that it does not always agree with subjective visual assessment. For this reason, the results of the different interpolation schemes will also be assessed visually.

For the block centred and block cornered schemes, the output pixel grid is offset from the input by half a pixel horizontally and vertically. For this, the ground truth image was estimated by averaging 2×2 blocks of the input image before downsampling.

4 Results and Discussion

The different interpolation schemes are applied to each of the 24 images in the test set, with the average PSNR over the set calculated. The results are listed in Table 1. For all of the schemes, using the diagonal interpolation gives a small drop in the average PSNR (although for some images there may be a slight improvement). However, with the small dataset used, this drop is not statistically significant.

The block-centred scheme gave a significant improvement in PSNR over the pixel-centred scheme. The block-cornered scheme gave a small but statistically significant improvement over the block-centred scheme in terms of PSNR. As with the other methods, using diagonal interpolation did not improve the results on average (although *BND1G* gave the best PSNR for about one quarter of the images tested). However the difference in average PSNR between *BN* and *BND1G* is not statistically significant.

One new artefact resulting from using skip mode is a level of blockiness or jumping along diagonal lines and edges. This can be clearly seen in Figure 10, especially along the mast regions. This is most noticeable

 ISBN 978-0-9934207-0-2

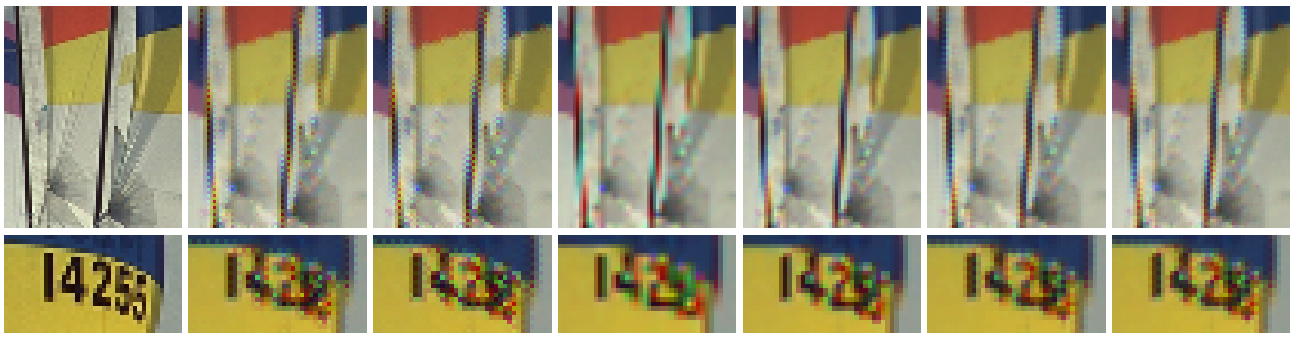
Scheme	Average PSNR	Number best	PSNR for Figure 10 image
PC	24.714 dB	0	26.405 dB
PCL	24.411 dB	0	26.107 dB
PCD	24.604 dB	0	26.350 dB
BC	26.269 dB	1	27.733 dB
BCD	26.226 dB	0	27.700 dB
BN	26.605 dB	17	28.061 dB
BND1G	26.601 dB	6	28.087 dB
BND1	26.331 dB	0	27.889 dB
BND2G	26.455 dB	0	27.905 dB

Table 1: PSNR averaged over 24 images of the Kodak image set, and for the image in Figure 10.

ground truth pixel-centred (*PC*) block-centred (*BC*) block-cornered (*BN*)

Figure 10: Example image showing the different demosaicing schemes

within the pixel-centred interpolation, but is also apparent with the other schemes. This is an inevitable result of the 2×2 block based sampling. Also apparent from Figure 10 is a significant blurring and loss of fine detail. This is primarily a consequence of using bilinear interpolation, and could perhaps be improved by using more advanced interpolation methods.



ground truth *PC* *PCD* *PCL* *BC* *BN* *BND1G*

Figure 11: Detail from the mast region and number regions

To facilitate visual assessment, part of the mast region if Figure 10 is expanded in Figure 11. The pixel-centred scheme (*PC*) has quite significant zipper artefacts. These can be almost completely eliminated by using the simpler horizontal and vertical averaging (*PCL*), although this is at the expense of increased blurring and colour bleed. Visually, using diagonal interpolation (*PCD*) is indistinguishable from the (*PC*) method.

Block-centred interpolation (*BC*) gives significant visual improvement of both colour bleed and geometric

 ISBN 978-0-9934207-0-2

distortion over the *PC* method, and appears to be free of zipper artefacts.

The block-cornered method appears to have slightly less colour bleeding than the other methods. The slight blurring makes the geometric distortion visually less noticeable. It does, however, have moderate zipper artefacts. In terms of visual quality, there is no noticeable difference between *BN* and *BND1G*.

Overall, in terms of visual evaluation, if the zipper artefact is deemed objectionable, then the block-centred method is preferable (*BC*), otherwise the visual assessment confirms the objective results given from the PSNR.

5 Conclusion

With CMOS sensors, skip mode readout allows the resolution of the image to be reduced without affecting the field of view of the camera. When skip mode is used with Bayer pattern sensors, pixels are read out in 2×2 blocks because of the colour filter array pattern. When demosaicing the raw image to produce a full colour image, aligning the output pixels based on the block structure gives better results than aligning them based on the input pixel locations. For downsampling by a factor of 2, aligning the pixels to the corners of the 2×2 Bayer blocks gives slightly better results than aligning them to the block centres. This reduces the geometric distortion resulting from the block structure of the sampling. Corner alignment does suffer from moderate zipper artefacts, so if these are likely to be an issue either visually or for subsequent processing, then block-centred alignment is preferable.

Future work involves investigating whether edge directed interpolation can reduce the artefacts (blur in particular) and give an improvement in PSNR. This, and other more advanced interpolation techniques, are complicated by the uneven sampling associated with skip mode readout.

References

[Bayer, 1976] Bayer, B. E. (1976). Color imaging array. USA Patent 3971065.

[Fossum, 1997] Fossum, E. R. (1997). Cmos image sensors: electronic camera-on-a-chip. *IEEE Transactions on Electron Devices*, 44(10):1689–1698.

[Gunturk et al., 2002] Gunturk, B., Altunbasak, Y., and Mersereau, R. (2002). Color plane interpolation using alternating projections. *IEEE Transactions on Image Processing*, 11(9):997–1013.

[Hsia, 2004] Hsia, S. C. (2004). Fast high-quality color-filter-array interpolation method for digital camera systems. *Journal of Electronic Imaging*, 13(1):244–247.

[Kodak, 1991] Kodak (1991). Kodak photo cd pcd0992. http://r0k.us/graphics/kodak/.

[Li and Randhawa, 2009] Li, J. and Randhawa, S. (2009). Color filter array demosaicking using high-order interpolation techniques with a weighted median filter for sharp color edge preservation. *IEEE Transactions on Image Processing*, 18(9):1946–1957.

[Li et al., 2008] Li, X., Gunturk, B., and Zhang, L. (2008). Image demosaicing: a systematic survey. In *Visual Communications and Image Processing 2008*, volume SPIE 6822, page 15 pages. SPIE.

[Lu and Tan, 2003] Lu, W. and Tan, Y. P. (2003). Color filter array demosaicking: new method and performance measures. *IEEE Transactions on Image Processing*, 12(10):1194–1210.

[Micron, 2006] Micron (2006). *MT9P031 5MP Image Sensor Product Brief*. Micron Technology Inc.

[Parulski, 1985] Parulski, K. (1985). Color filters and processing alternatives for one-chip cameras. *IEEE Transactions on Electron Devices*, 32(8):1381–1389.

 ISBN 978-0-9934207-0-2

Gradient Magnitude Based Normalised Convolution

Ahmad Al-Kabbany[1], Sonya Coleman[2], and Dermot Kerr[2]

[1]*VIVA Lab, School of Electrical Engineering and Computer Science, University of Ottawa, Canada.*
[2]*School of Computing and Intelligent Systems, University of Ulster, BT48 7JL, N. Ireland.*
alkabbany@ieee.org; {sa.coleman, d.kerr}@ulster.ac.uk

Abstract

Although image data can often be sparse for a variety of different reasons, standard image processing techniques require the use of complete image data. Therefore sparse image data must undergo reconstruction to yield complete images prior to any subsequent processing. Highly accurate image reconstruction techniques tend to be expensive to implement whilst simpler techniques, such as image interpolation, are usually not adequate to support subsequent reliable image processing. A common approach to image reconstruction is normalised convolution; we present a modified approach to normalised convolution which is based on the sparse image content and we demonstrate that accurate reconstruction is achieved yielding better image processing results than the current standard normalised convolution.

Keywords: Image reconstruction, Normalised convolution.

1 Introduction

Sparsity is a well-documented problem in all branches of science; numerous methods including statistical approaches and multi-resolution analysis do exist in the literature to overcome this problem whether in a time series, a 2-D planar image or an unordered set [Rybicki and Press, 1992, Ford and Etter, 1998]. One of the key points to be addressed when thinking of a solution for data incompleteness is the nature of such incompleteness. As usual, regularity is always easy to deal with and when the image sparsity is regular conventional interpolation techniques can reconstruct the image successfully [Jain, 1989]. However, this is not the case when the data are irregularly distributed. This situation is often encountered in computer vision applications; for example the use of omnidirectional cameras [Yagi and Kawato, 1990, Hong et al., 1992, Yamazawa et al., 1993] where unwarping the omnidirectional images to planar images results in incomplete projections [Scotney et al., 2006]. This sparsity, which should be handled before proceeding with feature detection and managing the process of feature detection on such sparse un-warped images, has attracted considerable research efforts over the last two decades. Normalised convolution is one of the methods used for interpolating irregularly sampled images. It was first introduced in [Knutsson and Westin, 1993] and has shown superiority over conventional grid-based techniques [Foster and Evans, 2008]. It also has illustrated remarkable capabilities in dealing with low subsampling rates [Piroddi and Petrou, 2004].

Feature detection is at the heart of all computer vision applications. There has been an incessant desire to develop reliable and stable feature detectors that could reveal the actual informative content of an image while withstanding possible image distortions and scale variations. In [Kerr et al., 2008], a novel technique was presented for corner detection adopting the finite-element method and working directly on sparse images. Within that framework, an adaptive gradient operator is formulated where an irregular image is represented by an irregular mesh of triangular elements and piece-wise linear basis functions. We have extended the work in [Kerr et al., 2008] by presenting a normalised convolution based approach called Gradient Magnitude-Weighted Normalised Convolution (GMWNC). Using the gradient operator proposed in [Kerr et al., 2008] as a weighting

 ISBN 978-0-9934207-0-2

function, GMWNC improves the reconstruction of sparse images compared with the conventional normalised convolution (NC). For comparative purposes, feature detection evaluation methods have been applied to the reconstructed images using GMWNC as well as standard normalised convolution (NC). While maintaining low computational complexity, the proposed algorithm results in improved feature detection with consistently improved root mean square error (RMSE) than the traditional NC approach.

The paper is organised as follows: in Section 2, an overview on the conventional NC technique is presented. In Section 3, the GMWNC is proposed. Results are shown in Section 4 and Section 5 presents a conclusion and suggested future work.

2 Normalised Convolution

Normalised convolution is an algorithm that operates on irregularly sampled or sparse data sets in order to fill-in the missing information. Accordingly, it has been used extensively to deal with image data incompleteness as a spatial interpolation technique. NC adopts a standard convolution filter, classically a Gaussian filter, in addition to a certainty map. The idea of constructing a certainty map was suggested to differentiate between the locations where we have a zero-valued pixel and the locations where we have missing data; this map is a simple binary filter [Piroddi and Petrou, 2004].

Conventional NC is called normalised averaging and it involves two convolutions and an element-wise division [Knutsson and Westin, 1993]. The first convolution is where the standard filter, namely the applicability filter is convolved with the sampled sparse data. This filter is responsible for the process of diffusing the information from the areas where it exists to the areas where it is missed, according to a certain profile. According to its properties, it defines the vicinity over which it operates. In the case of a Gaussian filter, interpolation takes place within the function support according to a Gaussian profile. Equations (1), (2) and (3) shows the three operations which are carried out in the simplest NC algorithm. The first step is to calculate

$$D(x, y) = f(x, y) \otimes g(x, y) \tag{1}$$

where $f(x, y)$ is the sampled input data and $g(x, y)$ is the applicability filter. The second step is to calculate

$$N(x, y) = c(x, y) \otimes g(x, y) \tag{2}$$

where $c(x, y)$ is the binary valued certainly map.

The second convolution can be thought of as the certainties associated with the interpolations that took place in the first convolution [Foster and Evans, 2008]. In order to normalise the first convolution, the reconstructed image $\hat{f}$ is determined by

$$\hat{f} = \frac{D(x, y)}{N(x, y)} \tag{3}$$

By using the available information and a map for data certainty, we are able to generate interpolated pixel values in the locations where none were originally present. The NC technique is superior to many conventional in-filling techniques such as the bi-linear and bi-cubic interpolation [Foster and Evans, 2008].

3 Gradient Magnitude-Weighted Normalised Convolution

Gradient Magnitude-Weighted Normalised Convolution GMWNC is a NC-based technique that enhances the performance of the conventional NC for reconstructing sparse images. The approach in based on that presented in [Kerr et al., 2008] and an overview is presented in the following subsections.

3.1 Overview of Gradient Operator Design

As in [Kerr et al., 2008], we consider sparse image to be represented by a spatially irregular sample of values of a continuous function $u(x, y)$ of image intensity on a domain Ω. The operator design procedure is then based

 ISBN 978-0-9934207-0-2

on the use of a mesh generated using Delaunay triangulation. With each node i in the mesh is associated a piecewise linear basis function $\phi_i(x, y)$ which has the properties $\phi_i(x_j, y_j) = 1$ if $i = j$ and $\phi_i(x_j, y_j) = 0$ if $i \neq j$ where (x_j, y_j) are the co-ordinates of the nodal point j in the mesh. We then approximately represent the image function u by a function $U(x, y) = \sum_{j=1}^{N} U_j \phi_j(x, y)$ in which the parameters $\{U_1, \ldots, U_N\}$ are mapped from the sparse image intensity values. The approximate image function representation is therefore piecewise linear on each triangle and has value U_j at node j.

The gradient operator design in [Kerr et al., 2008] is based on weak forms of operators in the finite element method [Becker et al., 1981, Scotney and Coleman, 2007]. In order to be directly applicable to sparse image data, the operator design needs to explicitly embrace the concept of operator size and shape, thus the design procedure explicitly embodies a size parameter σ that is determined by the local point density. Therefore, we use sets of Gaussian test functions $\psi_i^\sigma(x, y)$, $i = 1, \ldots, N$, when defining the derivative based operators; for first order operators respectively, this provides the functionals

$$E_i^\sigma(U) = \int_{\Omega_i^\sigma} \underline{b}_i \cdot \underline{\nabla} U \psi_i^\sigma \, d\Omega_i \tag{4}$$

where $\underline{b}_i$ is the basis function and U is the image. Each Gaussian function $\psi_i^\sigma(x, y)$ is restricted to have support over a neighbourhood Ω_i^σ, centred on node i, consisting of those triangular elements that have node i as a vertex and therefore that the integral in the definition of the functional E_i^σ need be computed by integration over only the neighbourhood Ω_i^σ rather than the entire image domain Ω. This process enables us to compute the gradient magnitude at each point within the sparse image for subsequent use in the reconstruction process.

3.2 Weighted Normalised Convolution

The algorithm commences by calculating the gradient magnitude responses across the sparse image using the gradient operators presented. Referring to equations (1) and (2) in Section 2, the gradient magnitude is equivalent to the applicability filter denoted here as the weighting function, and the presence of a gradient magnitude value implies a value of 1 in the certainty map in equation (2). Depending on whether the current location is non-zero-valued or not, $D(x, y)$ and $N(x, y)$ will be calculated and weighted by the gradient magnitude response. Equations (5) and (6) demonstrate the steps of carrying out the GMWNC algorithm. First, $D(x, y)$ is calculated as follows

$$D(x, y) = f(x, y) \otimes [g(x, y) * (\upsilon - GM(x, y))] \tag{5}$$

where υ is the maximum gradient magnitude response and $GM(x, y)$ is the gradient magnitude at the point (x, y). Similar to NC, $N(x, y)$ is identical to $D(x, y)$ with the sampled input data is replaced by the certainty filter $c(x, y)$.

$$N(x, y) = c(x, y) \otimes [g(x, y) * (\upsilon - GM(x, y))]. \tag{6}$$

This ensures that when the gradient magnitude is high (i.e. a potential feature point) the smoothing is reduced and when the gradient magnitude is low (i.e. a background point) the smoothing is increased, thus retaining the key images features during image reconstruction whilst removing noise. The reconstructed image is then calculated in the same way as indicated by equation (3).

4 Performance Evaluation

Improving the reconstruction of features is the main goal of the proposed technique. Therefore for evaluation purposes we apply a feature detector and use the well-known Figure of Merit (FoM) technique [Abdou and Pratt, 1979]. Taking into account the fairness of evaluation, FoM has been calculated over a range of signal to noise ratios, typically 100, 50, 20, 10, 5 and 1. In addition, the assessment was made using different percentages of sparsity. FoM was calculated using synthetic images for diagonal, vertical and horizontal edges. However, the results obtained for 75% of the image data were similar for both the proposed GMWNC and NC

 ISBN 978-0-9934207-0-2

techniques and are therefore not shown here; the significant improvements are found when the percentage data is reduced as low as 25%.

Figure 1 and Figure 2 present graphs for the Figure of Merit versus Signal to noise ratio using only 25% of the original image data for three edge types (diagonal, horizontal and vertical) using 3×3 and 7×7 filters respectively. In all cases we can see that the proposed GMWNC technique outperforms the traditional NC approach with respect to subsequent image processing, i.e., edge detection.

Edge image	Method	RMSE	
		25% data	75% data
Diagonal	NC	25.845	26.6573
	GMWNC	26.5877	26.0140
Horizontal	NC	28.0025	28.3764
	GMWNC	28.2964	28.1106
Vertical	NC	25.896	26.3009
	GMWNC	27.124	26.1031

Table 1: RMSE values for different sparsity ratios and edge orientations

In addition, Table 1 presents root mean squared errors (RMSE) for both approaches, comparing the reconstructed images with the original images using 25% and 75% data. The RMSE values demonstrate that both approaches have similar accuracy with respect to reconstruction; however the FoM results in Figure 1 and Figure 2 illustrate that our proposed approach yields better edge detection and hence feature reconstruction.

5 Conclusion

Techniques that are based on image reconstruction without prior image knowledge do not generally provide reliable mechanisms for accurate feature extraction. The most accurate reconstruction technique currently available that is also computationally efficient is the Normalised Convolution approach. In [Scotney et al., 2006], a design procedure was presented for edge detection operators for direct use on sparse image data. Here we extend this existing approach by applying it to sparse image data to derive knowledge of the image content that can be subsequently used to enhance the image reconstruction process. We have demonstrated the success of this approach by presenting root mean squared errors that are similar to those obtained using the standard Normalised Convolution method but also by demonstrating via the Figure of Merit technique that the images reconstructed using the proposed GMWNC approach yield better results with respect to edge detection than images reconstructed using standard NC even when using as little as 25% of the original image data. Having obtained accurate results for image reconstruction the focus of our further work will be the use of real omnidirectional images for robot localisation.

References

[Abdou and Pratt, 1979] Abdou, I. E. and Pratt, W. K. (1979). Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE*, 67(5):753–763.

[Becker et al., 1981] Becker, E. B., Carey, G. F., and Oden, J. T. (1981). Finite elements, an introduction: Volume i. *., 258*, page 1981.

[Ford and Etter, 1998] Ford, C. and Etter, D. (1998). Wavelet basis reconstruction of nonuniformly sampled data. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 45(8):1165–1168.

[Foster and Evans, 2008] Foster, M. P. and Evans, A. N. (2008). An evaluation of interpolation techniques for reconstructing ionospheric tec maps. *Geoscience and Remote Sensing, IEEE Transactions on*, 46(7):2153–2164.

[Hong et al., 1992] Hong, J., Tan, X., Pinette, B., Weiss, R., and Riseman, E. M. (1992). Image-based homing. *Control Systems, IEEE*, 12(1):38–45.

[Jain, 1989] Jain, A. K. (1989). *Fundamentals of digital image processing*. Prentice-Hall, Inc.

[Kerr et al., 2008] Kerr, D., Scotney, B., and Coleman, S. (2008). Interest point detection on incomplete images. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 817–820. IEEE.

[Knutsson and Westin, 1993] Knutsson, H. and Westin, C.-F. (1993). Normalized and differential convolution. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 515–523. IEEE.

[Piroddi and Petrou, 2004] Piroddi, R. and Petrou, M. (2004). Analysis of irregularly sampled data: A review. *Advances in Imaging and Electron Physics*, 132:109–165.

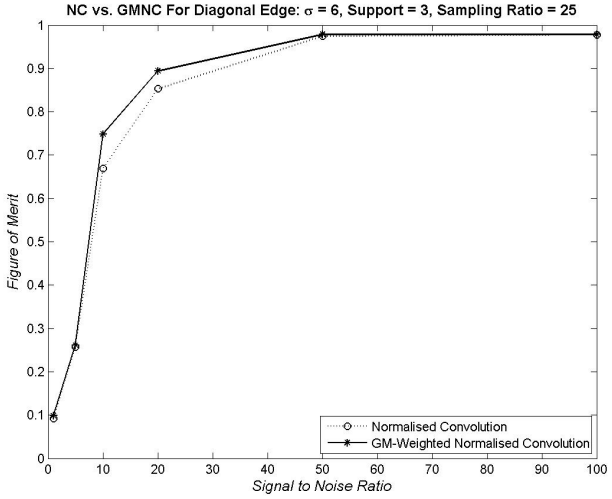
[Rybicki and Press, 1992] Rybicki, G. B. and Press, W. H. (1992). Interpolation, realization, and reconstruction of noisy, irregularly sampled data. *The Astrophysical Journal*, 398:169–176.

[Scotney et al., 2006] Scotney, B., Coleman, S., and Kerr, D. (2006). A graph theoretic approach to direct processing of sparse unwarped panoramic images. In *Image Processing, 2006 IEEE International Conference on*, pages 1557–1560. IEEE.

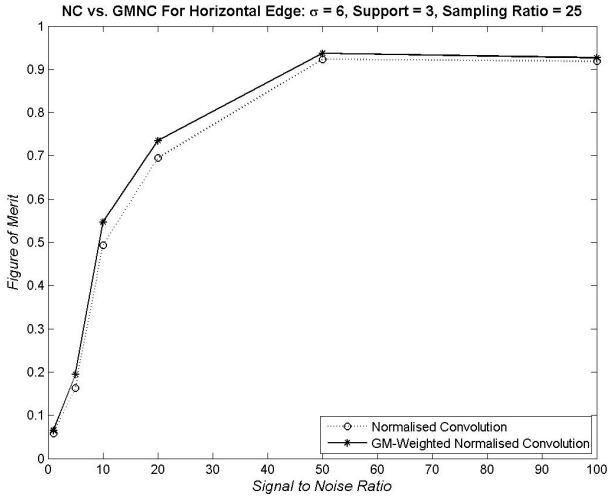
[Scotney and Coleman, 2007] Scotney, B. W. and Coleman, S. A. (2007). Improving angular error via systematically designed near-circular gaussian-based feature extraction operators. *Pattern Recognition*, 40(5):1451–1465.

[Yagi and Kawato, 1990] Yagi, Y. and Kawato, S. (1990). Panorama scene analysis with conic projection. In *Intelligent Robots and Systems' 90.'Towards a New Frontier of Applications', Proceedings. IROS'90. IEEE International Workshop on*, pages 181–187. IEEE.

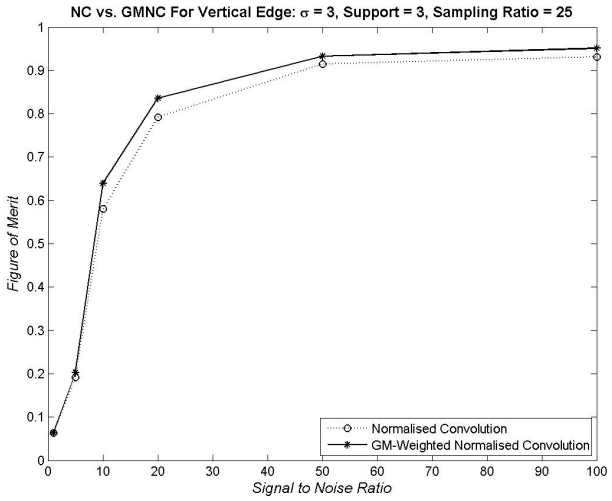
[Yamazawa et al., 1993] Yamazawa, K., Yagi, Y., and Yachida, M. (1993). Omnidirectional imaging with hyperboloidal projection. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 2, pages 1029–1034. IEEE.



(a) Diagonal Edge



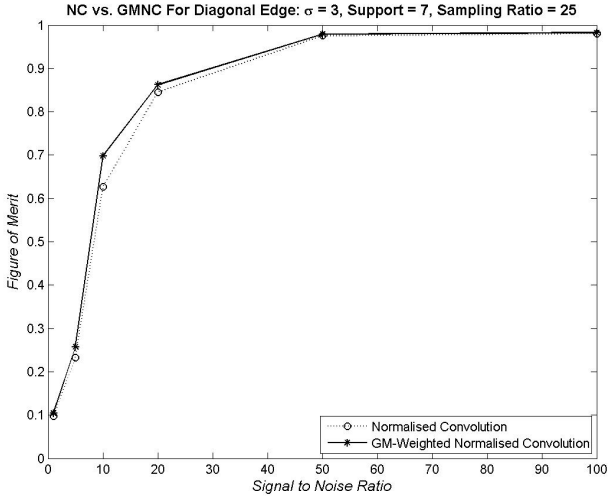
(b) Horizontal Edge



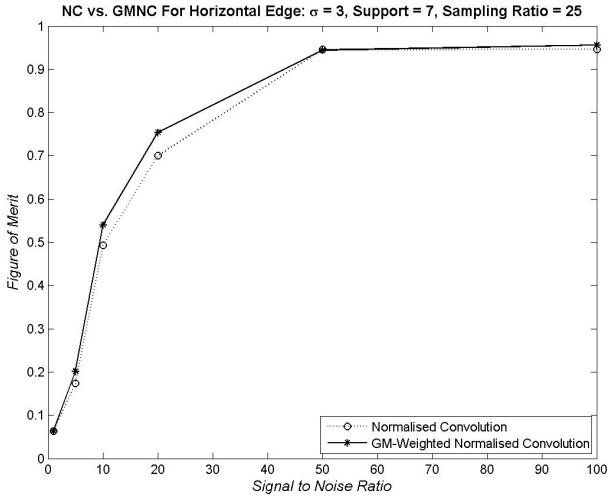
(c) Vertical Edge

Figure 1: FoM Vs SNR using images with 25% data and a 3×3 filter

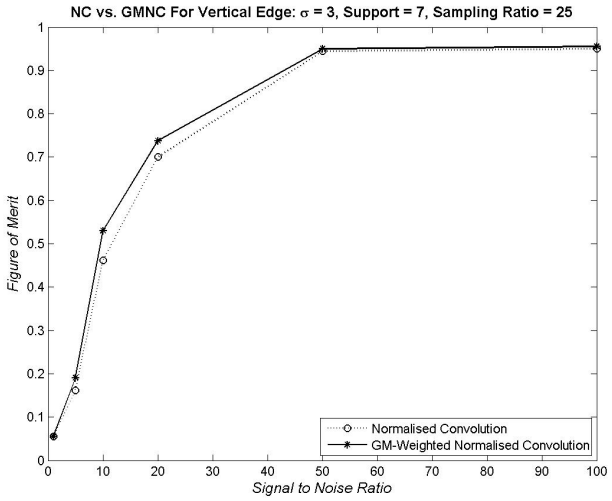
 ISBN 978-0-9934207-0-2



(a) Diagonal Edge



(b) Horizontal Edge



(c) Vertical Edge

Figure 2: FoM Vs SNR using images with 25% data and a 7×7 filter

 ISBN 978-0-9934207-0-2

PatchCity: Procedural City Generation using Texture Synthesis

John D. Bustard* and Liam P. de Valmency**

*Queen's University Belfast, UK
**University of Southampton, UK

Abstract

PatchCity is a new approach to the procedural generation of city models. The algorithm uses texture synthesis to create a city layout in the visual style of one or more input examples. Data is provided in vector graphic form from either real or synthetic city definitions. The paper describes the PatchCity algorithm, illustrates its use, and identifies its strengths and limitations. The technique provides a greater range of features and styles of city layout than existing generative methods, thereby achieving results that are more realistic. An open source implementation of the algorithm is available.

Keywords: procedural modeling, texture synthesis, example-based

1 Introduction

Procedural modeling techniques have substantially reduced the effort required to create synthesised environments. In the computer games industry, for example, sandbox games [Sony Online Entertainment, 2014] can now be set in vast explorable worlds, and the player presented with an ever-changing environment each time they enter the game. Similarly, in film, generative modeling has greatly simplified the creation of wide environment shots, backdrops, and the in-fill of detailed scenery [Planetside Software, 2014] [IDV Inc., 2014].

Existing procedural techniques for city generation use a variety of synthesis methods to produce road layouts and building placements. These include manually created heuristics [Parish and Müller, 2001], agent models [Lechner et al., 2003], tensor fields [Chen et al., 2008] and statistically based constraints [Groenewegen et al., 2009]. Such techniques can produce realistic approximations of cities but have limitations. In particular:

- The structure of a created city is an emergent property of the underlying heuristic rules that determine its shape. The variety of cities produced is therefore constrained by the creative power of those heuristics. As a result, existing procedural city techniques are often restricted to grid-based, metropolitan layouts.
- The fine tuning of a created city's appearance is achieved through adjusting the parameters of the generative algorithm. As the heuristics become increasingly complex, the tuning of the associated parameters also increases in difficulty [Lechner et al., 2003].
- When complex heuristics and constraints are applied, the time needed to generate structures can increase significantly. This can reach a point where construction can no longer occur in real-time, relying instead on offline generation; this may not be suitable for some applications.

This paper describes a texture synthesis approach to city generation, PatchCity, which was conceived as a way of addressing these limitations. The strategy is to extrapolate from data provided in the form of vector graphic city layouts to create a new, larger map, which maintains the consistency and visual style of the original. The advantages of this approach are:

 ISBN 978-0-9934207-0-2

- The variation in city styles and features is not constrained by the algorithm because they are produced directly from the patterns in the user-supplied input.
- By adjusting the input data, subtle changes in city appearance can be achieved in an intuitive way. In addition, texture synthesis methods can also be applied to fill holes around explicitly defined city features, enabling precise adjustment of the city structure as required.
- Because the method uses a look-up table of city patches, the city structure can be produced efficiently without requiring complex heuristic rules to create detailed results. In addition, the use of existing data sources can also handle building and other feature placement implicitly, which requires significant additional processing within other city generation algorithms.

Section 2 of the paper examines existing work in procedural city generation. Section 3 then describes the PatchCity algorithm and related work in texture synthesis. The section shows how vector-based texture synthesis can be performed using urban layouts and identifies the adjustment steps that are applied to improve the resulting structure. Section 4 presents the results of applying the PatchCity algorithm to four example urban locations. The section highlights the strengths and limitations of the algorithm. The paper concludes with a discussion of potential future work.

2 Procedural City Generation

CityEngine, designed by Parish and Müller [Parish and Müller, 2001] was the first algorithm developed for automatically constructing city road networks. CityEngine takes road network descriptions as input, and creates a skeleton city layout. The heuristic rules for the city's construction are expressed as a set of Lindenmayer System (L-system) formal grammars [Rozenberg and Salomaa, 1980]. During creation, consistency adjustments are also made to ensure, for example, that roads do not overlap. The heuristics and constraints require additional user input in the form of images representing water boundaries, terrain elevation, population density and street patterns. The use of standard highway patterns, such as grid-aligned, radial, and population-density seeking, helps give the generated city a realistic appearance.

CityEngine is effective in situations where a "good enough" output is acceptable, such as stylised video games. One problem with this approach, however, is that the indirect nature of L-system descriptions makes them difficult to adjust, especially as they increase in size to deal with detailed map features. Lechner et al. [Lechner et al., 2003] proposed an alternative agent-based approach that helps overcome these issues. Their technique delegates the handling of road and building placement to agents that follow specific rules, similar to those used by city planners. Different agent types are assigned specific tasks, such as extending roads into unexplored terrain, connecting existing roads to create shortcuts, and searching for available land on which to construct buildings. Building construction is based on the type of zone being built and the estimated land value.

Lechner's approach requires less user input than CityEngine, needing only one image. It also has the advantage of allowing the user to set global parameters to fine tune the appearance of the generated city. In later work, the authors combined their system with the SimCity engine for more aesthetically pleasing output [Lechner et al., 2004]. There are, however, two limitations with an agent-based approach. One is that basing generation on the emergent results of a set of heuristics limits the control over the resulting city appearance. The second is that generation speed is relatively slow.

Groenewegen et al. [Groenewegen et al., 2009] introduced a strategy for using real-world exemplar data in the generation of cities. The models govern how a city might be formed based on user-specified factors such as city size, continent, historical background and number of highways passing through it. The generation process divides a city into zones which represent various types of buildings, including light or heavy industrial zones, commercial districts and transport nodes. These zones are grouped according to their statistical likelihood of adjacency, based on the input parameters. For example, a city in Western Europe would have a different zone grouping to one in North America. The Groenewegen approach has the obvious overhead of requiring a new land use model for each specific country in which a city is generated, which implies having an understanding of

 ISBN 978-0-9934207-0-2

building patterns in that situation. Also, it focuses on the high-level structure for a city, meaning that additional work is needed to provide the detail in each case.

Another strategy for generating road networks from pre-existing data is described by Aliaga et al. [Aliaga et al., 2008], where new roads are produced by analysing statistical properties of an existing network. Their system uses a random walk algorithm, which iteratively adds road segments to the existing network in a way that mimics the properties of road intersection points in the original, such as mean/variance of road distances and angles, tortuosity and intersection hierarchy levels. As a result, the road network generated is one of the most accurate among those available. Its implementation is, however, much more complex than other approaches. Also, it can be biased by highly atypical streets from an example map, which may skew the intersection property calculations. Texture warping is used to fill space between roads to create convincing satellite imagery.

Tackling the difficulty of indirect adjustment of city descriptions, as required in CityEngine, Chen et al. [Chen et al., 2008] proposed an alternative interactive approach. This allows users to create a road network from scratch or modify an existing network. The underlying representation is a tensor field which can be manipulated directly or modified through the graph representing the resulting road network. The creation engine requires the user to know the general layout of the road network they wish to generate, which prevents creation of a random but realistic city derived from common city patterns. This becomes an advantage, however, if the user has a very specific city layout in mind. The key strength of the method is the realism of the output, which improves on that of CityEngine.

From this analysis, five desirable properties of a city generation algorithm are evident: realism, detail, automation, control and speed of execution. Each existing approach varies in the degree to which these properties are achieved. The next section presents an algorithm intended to improve on the detail and realism of existing results. The approach can be fully automatic and has the potential for low level adjustment through explicit patch placement and hole filling. It also operates sufficiently quickly for interactive applications.

3 PatchCity Approach

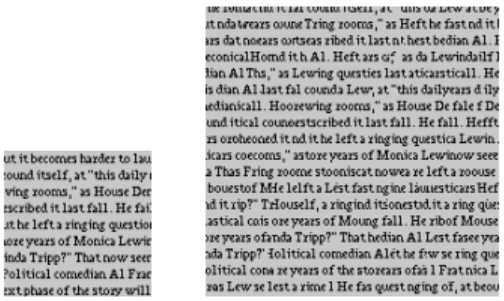


Figure 1: Results of applying texture synthesis to an image of text.

PatchCity generates new city structure through a technique based on texture synthesis. In general, texture synthesis algorithms aim to extrapolate from data provided in the form of an image to create a new, larger image which maintains the consistency and visual style of the original. Figure 1, for example, shows the technique applied to a block of text. The visual appearance of the generated block on the right strongly resembles that of the initial sample patch on the left from which it was produced. Note that no semantic information has been used in the synthesis process so the generated text is meaningless.

PatchCity is based on a similar synthesis approach but uses patches in vector graphic form that include important semantic information. More specifically, it uses patches in the form of maps from Geographic Information Systems (GIS) such as OpenStreetMap [Haklay and Weber, 2008]. Its approach means that it can handle building and other city feature placement, which previous city generation algorithms have had to process separately.

 ISBN 978-0-9934207-0-2

Current algorithms for texture synthesis largely fall into two categories: pixel-based [Efros and Leung, 1999] and patch-based methods [Efros and Freeman, 2001] [Kwatra et al., 2003]. One issue with per-pixel synthesis methods, such as non-parametric pixel sampling [Efros and Leung, 1999], is that they fail to preserve the high-level structure of road networks, resulting in disorganised clusters of road segments. Also, existing patch-based methods, such as Image Quilting or Graphcut, create frequent repetition of certain sampled patches, and have road structure disconnection along patch seams. PatchCity addresses these issues by performing texture synthesis using a vector representation. This format enables patch constraints to be specified more precisely and provides a representation that enables patches to be merged without noticeable errors.

3.1 Algorithm

The PatchCity algorithm takes one or more vector street maps as input, from which patches can be extracted. For image based texture synthesis, sampling of the original image(s) can take place in real-time, as patches can be calculated entirely using local pixel values. In a vector format, this process becomes more difficult, as roads will be represented as a series of points marking their path, and buildings by a set of points denoting their shape. To obtain the set of road points and buildings which lie within a patch, at a given point, requires clipping of path segments, as well as exclusion of buildings which do not fall entirely within the patch area. The first step in the algorithm is to divide each input map into its respective patches, and to store these in a lookup table.

Patch extraction takes place at a user-specified interval along each axis of the 2D image. Each extracted patch stores the paths which cross through its area, clipped to the boundaries of the map region from which it was taken. The Cohen-Sutherland algorithm [Sproull and Newman, 1973] is used to clip any lines that pass through the patch, and store them. Buildings that fall entirely within the patch are also stored.

To prevent repeated iteration through each path node within a patch when locating nodes that lie on its boundary, results are cached for further use. This includes the road count along each edge, the road locations (as a distance from the edge's top or leftmost point), and the road sizes. The use of the cache improves the efficiency of the algorithm and also enables boundary values to be compared more conveniently. Specifically, execution time is reduced because only those patches with equivalent road counts along the patch boundary need to be considered as candidates.

Map generation takes place in a top to bottom, left to right fashion, following a grid of patch-sized areas. In addition to the patch extraction interval, the user can also specify the width and height of the synthesised map (the number of patches along each axis), as well as the size of each of the patches themselves. For each unfilled space in the map, the algorithm will fetch the most appropriate patch from the lookup table.

While regular texture synthesis algorithms perform a nearest neighbour comparison with pixel values, the PatchCity algorithm takes a different approach. Firstly, the top-left patch in the map is initialised to a random fragment from the lookup table containing at least one road. Subsequent patches are chosen based on a cost function which takes two patches as arguments. In choosing a new patch from the lookup table, only those with an equivalent number of roads along the connecting edge are evaluated. The cost function must be calculated for each edge which connects the target patch location to those which have already been placed in the synthesised map. For patches in either the first row or the first column, only one evaluation takes place (against the patch to the left and the patch above respectively). For each of the remaining patches, the choice of patch is based on the sum of the cost function run against both adjacent patches. The algorithm chooses the candidate patch that returns the lowest value from the cost function; in the case of multiple patches with the same value, the system selects randomly from them.

The cost function itself attempts to minimise the visual error between placed patches, using two properties of the roads which lie along the seam: their location and size. Specifically, the function was designed to return a higher error cost when the roads on each side of the seam are further apart, and when they are of different sizes. The function is evaluated as:

 ISBN 978-0-9934207-0-2

$$cost(a,b) = \sum_{i=1}^{numRoads} (compat(a_i, b_i) * a_{i_{size}} * b_{i_{size}}) \quad (1)$$

where

$$dist(r1, r2) = |r1 - r2|^2 \quad (2)$$

$$ratio(s1, s2) = max(s1, s2)/min(s1, s2) \quad (3)$$

$$compat(a,b) = dist(a_{pos}, b_{pos}) * ratio(a_{size}, b_{size}) \quad (4)$$

It is much more visually jarring in a synthesised map when a major road is disconnected from another along a seam than when the same problem occurs with a side road or pathway. For this reason, each road pair cost is also multiplied by the product of their sizes. This results in a lower contribution to the cost total by smaller roads, and thus disconnects on major roads are discouraged more strongly.

The patches are therefore chosen as:

$$\arg\min_p \ cost(patch(x-1, y), p) \quad (5)$$

for top row patches,

$$\arg\min_p \ cost(patch(x, y-1), p) \quad (6)$$

for leftmost column patches, and

$$\arg\min_p \ (cost(patch(x-1, y), p) + cost(patch(x, y-1), p)) \quad (7)$$

for the remaining patch spaces.

For comparison purposes, each of the fragments stored by the lookup table is used as a template. On choosing a fragment to fill a space within the grid, the path and building data are copied to a new patch, so that the path and building data may be modified by further steps within the algorithm, without corrupting the patch data stored in the lookup table.

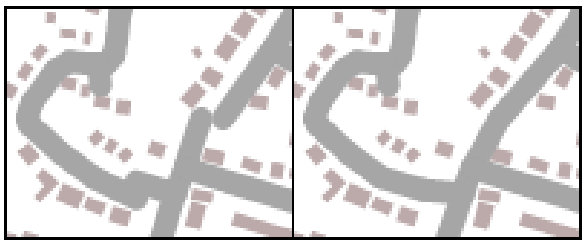


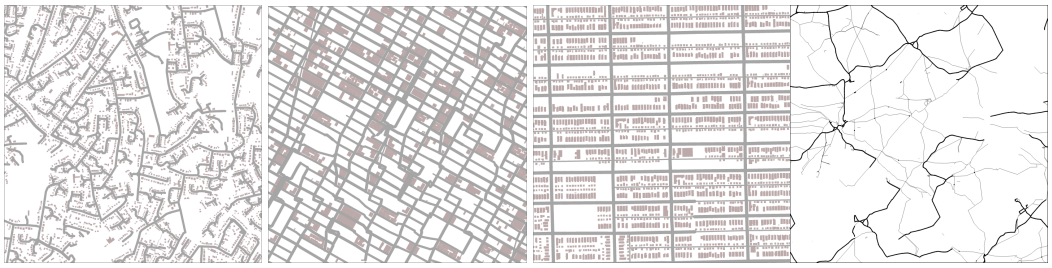
Figure 2: Fix step applied to disconnected roads. Before (left) and after (right).

Figure 3: Fix step applied to floating roads. Before (left) and after (right).

The next stage of the process is to deal with road disconnects along patch seams (Figure 2) and floating roads that are fully disconnected (Figure 3). To remove disconnection of roads along patch seams, the endpoint of

 ISBN 978-0-9934207-0-2

Figure 4: Test inputs. Left to right: Chicago, IL; Southampton, UK; Northampton UK; Times Square, NY.



(a) (b) (c) (d)

Figure 5: Cities generated from a single input map. a) Southampton. b) Times Square. c) Chicago. d) Northampton.

each road on a given patch edge is translated by half of the distance to the corresponding endpoint in the adjacent patch. This results in a join between the roads, at the cost of introducing some heuristic modification to the road network. The distance between corresponding road ends is usually small enough that these modifications are hard to detect in all but the most regularly structured maps. This issue is discussed further in the results section.

4 Results and Evaluation

Four test maps (Figure 4) were used to evaluate PatchCity. Together they represent four of the most common styles of road network: (i) irregular city roads with varying shapes and directions (Southampton); (ii) irregular, grid-aligned roads (New York); (iii) regular, grid-aligned roads with high building density (Chicago); and (iv) sparse, disorganised country roads (Northampton).

Plausibly realistic cities are produced (Figure 5). There are, however, some issues: (i) removal of floating road sections can produce large holes, (ii) buildings that cross patch boundaries are excluded, and (iii) cities with a grid structure can show a visible seam.

Floating roads start and end within the boundaries of the map area. The removal of these problematic roads is implemented by identifying the connected component subgraphs of the road network. Subgraphs that have no roads leading off any of the boundaries of the map are identified as floating roads. When floating roads are removed, buildings may become detached and so also need to be removed. This is achieved by creating a map which associates each building with the nearest path node. After removing a floating road, any building which maps to those paths is also removed. The removal of floating roads and their accompanying buildings can introduce noticeable holes within the map. Consequently this stage is optional as it may or may not improve the synthesised results. The algorithm could be improved with an additional step that connects floating road subgraphs to the rest of the map by inserting an additional connecting road.

In addition, during the population of the lookup table, extracted patches contain only those buildings that fall entirely within their boundaries. This is problematic when buildings are either too large to fit within the specified patch size, or large enough that they are excluded from most patches in their area. This results in maps which often contain large gaps in building density along seam lines (Figure 6). This issue could be addressed

by including the clipped buildings within a patch, and only inserting them into the final synthesised map if this does not result in a collision with neighbouring patch elements.

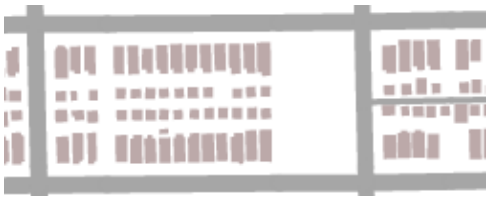


Figure 6: Example of building placement gaps occurring due to patch cropping.

A noticeable artifact within the synthesised Times Square result is that at scattered points around the map the road network falls out of line with its intended grid structure, creating a knock on effect for the subsequent patches (Figure 7). This problem exists largely because PatchCity operates on a fixed grid of lookup table patch locations, which may not align precisely with the city's grid structure. These artifacts could be reduced by searching in the local neighbourhood of lookup table patches prior to placement to find the best possible match between placed tiles.

Figure 7: Example of the synthesis method failing to fully preserve grid-based road alignment.

Despite these issues, grid network generation is still possible with the algorithm if the input road network has regularity, i.e. blocks within the input grid network are regularly spaced, as shown in the Chicago synthesis example (Figure 5 (a)).

Although no explicit optimisation has been performed, PatchCity's main synthesis algorithm takes place in real-time. The execution time for the selection of patches is proportional to the number of lookup table entries. In addition, the removal of floating roads requires a calculation of the connected components which is linear in the number of road segments and junctions. This implementation's measured performance is linear in the area of the output map. The ability to generate in real-time gives PatchCity a much wider range of applications. For example, its use can be easily imagined in an open world video game which generates the environment around the player as they travel, with the potential for 'infinite' worlds.

5 Conclusions and Further Work

The key strength of PatchCity is its use of structural example data as input. This enables the algorithm to generate a diverse range of city styles, from regular, grid-aligned modern cities to more organic, historic urban centers. PatchCity is also able to capture instances of uncommon city features such as shortcut pathways and roundabouts. This adds to the realism and variety of the output.

PatchCity's building synthesis is a result of the algorithm's extraction of building data from the input maps, providing building placement as a built-in element of its execution. As further work, additional city metadata could be included in the synthesis process. Terrain data, land use models, height maps and population density information could all be added to the input and integrated into the patch extraction and placement process. By adding more terms to the cost function, this additional data could provide more detailed and realistic city structures.

Further improvements in city quality could also be achieved by performing user evaluations of synthetic results. With such evaluations it may be possible to develop improved metrics for tile placement as well as

 ISBN 978-0-9934207-0-2

a more global method of ranking different synthesised cities. Knowledge of what constitutes a high quality output would allow the system to repeat the synthesis process until the quality reaches a specified threshold.

The PatchCity algorithm can also be viewed as a novel contribution to the field of texture synthesis. Highly stylised structural data often produces artifacts using traditional pixel based methods. By adapting the cost function described in Section 3.1, PatchCity could be generalised to other synthesis applications.

References

[Aliaga et al., 2008] Aliaga, D. G., Vanegas, C. A., and Beneš, B. (2008). Interactive example-based urban layout synthesis. In *ACM Transactions on Graphics (TOG)*, volume 27, page 160. ACM.

[Chen et al., 2008] Chen, G., Esch, G., Wonka, P., Müller, P., and Zhang, E. (2008). Interactive procedural street modeling. *ACM Transactions on Graphics (TOG)*, 27(3):103.

[Efros and Freeman, 2001] Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM.

[Efros and Leung, 1999] Efros, A. A. and Leung, T. K. (1999). Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE.

[Groenewegen et al., 2009] Groenewegen, S. A., Smelik, R. M., de Kraker, K. J., and Bidarra, R. (2009). Procedural city layout generation based on urban land use models. In *Proceedings of the 30th Annual Conference of the European Association for Computer Graphics (EurographicsâĂŹ09)*, pages 45–48.

[Haklay and Weber, 2008] Haklay, M. M. and Weber, P. (2008). Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18.

[IDV Inc., 2014] IDV Inc. (2014). Speedtree. http://www.speedtree.com.

[Kwatra et al., 2003] Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A. (2003). Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM.

[Lechner et al., 2004] Lechner, T., Watson, B., Ren, P., Wilensky, U., Tisue, S., and Felsen, M. (2004). Procedural modeling of land use in cities.

[Lechner et al., 2003] Lechner, T., Watson, B., and Wilensky, U. (2003). Procedural city modeling. In *In 1st Midwestern Graphics Conference*.

[Parish and Müller, 2001] Parish, Y. I. and Müller, P. (2001). Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 301–308. ACM.

[Planetside Software, 2014] Planetside Software (2014). Terragen. http://www.planetside.co.uk.

[Rozenberg and Salomaa, 1980] Rozenberg, G. and Salomaa, A. (1980). *Mathematical Theory of L Systems*. Academic Press, Inc.

[Sony Online Entertainment, 2014] Sony Online Entertainment (2014). Everquest Next. http://www.everquestnext.com.

[Sproull and Newman, 1973] Sproull, B. and Newman, W. M. (1973). *Principles of Interactive Computer Graphics*. McGraw-Hill Education, international edition.

Simplifying Genetic Algorithm: A Bit Order Determined Sampling Method for Adaptive Template Matching

Chao Zhang[†], Takuya Akashi[‡]

[†]*Graduate School of Engineering,* [‡]*Faculty of Engineering*

Iwate University, Japan

Abstract

In this paper, we address the problem of template matching with scaling and in-plane rotation. To solve this problem, huge amount of matching tests are potentially required and the common integral image is no longer applicable. The effectiveness of template matching can be enhanced by adaptive sampling methods under smooth assumption. As a solution, we simplify the simple genetic algorithm (GA) to a stochastic adaptive sampling method. Our method introduces two operations called global sampling (GS) and local sampling (LS) as the substitution of crossover and mutation. GS and LS utilize the property of binary code: the left-most bit of a binary-coded individual affects the phenotype most significantly and vice versa. By randomly updating the high-order bits, GS has the ability to sample in a wide scope of searching space to approach the ground truth fast but roughly. On the contrary, LS has the ability to refine the final result in a narrow scope which is hard to be realized by GS. We compare our method against the simple GA over 11 sequences, with each sequence contains 36 images. The results show that our method performs more accurate and faster than the simple GA on adaptive template matching.

Keywords: Adaptive template matching, In-plan rotation, Global sampling, Local sampling, Genetic algorithm

1 Introduction

Template matching is a classical research topic of computer vision. With the development of key point based local features, template matching has been less studied in recent years because of the inefficiency due to exhaustive searching. However, it is worth noting that under many situations, template matching is still an efficient way to solve problems especially when the key points are hard to be detected or easy to be mismatched. Common template matching problem without considering the rotation and scaling is easy to be solved exhaustively by a modern personal computer because the degree of freedom (DOF) is simply two. However, when in-plane rotation, x-axis scaling, and y-axis scaling are involved additionally, the DOF grows to five, and is no longer suitable for exhaustive searching, because each sample is in a five DOF space. Under the assumption that a template is smooth, we can approach the ground truth adaptively

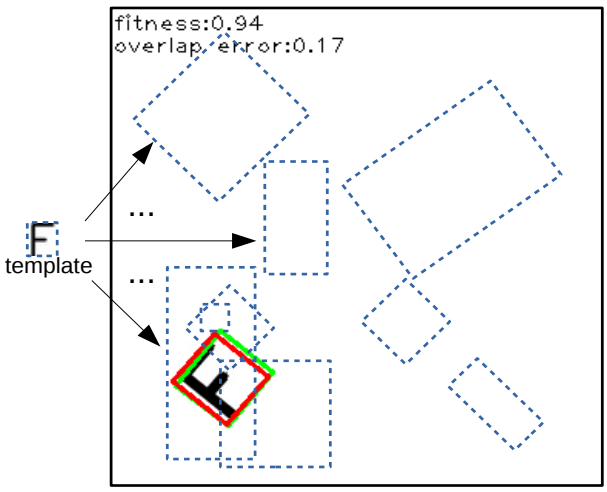


Figure 1: Template matching considering scaling and in-plane rotation. Red rectangle represents the ground true, green rectangle represents the matching result.

 ISBN 978-0-9934207-0-2

instead of estimating all the candidate regions. The "adaptiveness" can be realized from the perspective of optimization algorithms. In the situation of genetic algorithm (GA), we can treat the template matching problem as a minimization problem of grey value difference. When applying GA to template matching problem, achieving an accurate result usually requires a large number of populations and generations, because the optimization time of GA is large. We wish to develop an adaptive sampling method by controlling the scope of sampling space rather than utilizing evolutionary operations such as crossover, mutation. Figure 1 shows the task of our paper.

The rest of this paper is structured as follows. In Section 2, we survey template matching methods dealing with scaling and in-plane rotation. In Section 3, we introduce our method from two perspectives: 1) construction of a single individual, 2) GS and LS operations. In Section 4, we systematically compare our method against GA to show the enhancement. Finally, we conclude this paper in Section 5.

2 Related Work

In this section, we mainly survey previous works on template matching considering scaling and in-plane rotation. Despite the feature-based matching methods like SIFT [Lowe, 2004], ASIFT [Morel and Yu, 2009], direct template methods also play an important role.

To accelerate matching procedure, [Kim and de Araújo, 2007] apply cascaded filters to exclude areas which have low probability to be final result. [Akashi et al., 2007] treat template matching as an optimization problem under genetic algorithm framework and apply their method into real-time eye detection by inheriting previous frame's evolutionary result to the next. Genetic algorithm can evolutionarily select "promising" candidate areas to evaluate, thus can avoid exhaustive searching. [Korman et al., 2013] propose a method which can deal with affine transformation including scaling and rotation. It constructs a discrete sampling net to approach the ground truth sparsely. [Tsai and Chiang, 2002] decompose an input image into different multi-resolution levels in the wavelet-transformed domain, and then use only the pixels with high wavelet coefficients in the decomposed detail sub-image at a lower resolution level to compute the normalized correlation between two compared patterns. [Choi and Kim, 2002] propose a method based on the combination of the projection method and Zernike moments. At first stage, candidate regions are selected by computationally low cost features. After that, rotation invariant template matching is realized by performing Zernike moments on candidates.

3 Proposed method

3.1 Problem Description

Two greyscale images I_1 and I_2 are given as the input with each pixel's grey value in [0, 255]. I_1 is defined as a template image with size $n_1 \times n_1$. I_2 is defined as a target image with size $n_2 \times n_2$. For clarity, we consider I_1 and I_2 are both square images. An arbitrary transformation $T = \{r, t_x, t_y, s_x, s_y\}$ respects to a candidate region includes rotation r, x-axis's translation t_x, y-axis's translation t_y, x-axis's scaling s_x, y-axis's scaling s_y. Greyscale intensity difference is used to measure the similarity between I_1 and a candidate region in I_2, which can be represented as:

$$G(I_1, I_2, T) = \sum_{p_i \in I_1, p_i \in I_2} |p_i - p_i^T|, \tag{1}$$

where p_i is a pixel in template and p_i^T represents a pixel in I_2 after transformation T is involved.

Since minimizing the greyscale intensity difference is the task of our method, we can write it formally as:

$$\hat{T} = \arg\min_{T \in \mathbb{S}} G(I_1, I_2, T). \tag{2}$$

Sampling set $\mathbb{S}$ contains all the individuals we have estimated. Each individual is coded by a Gray code whose construction will be introduced in the next subsection.

 ISBN 978-0-9934207-0-2

3.2 Construction of each binary-coded individual

A binary Gray code is denoted by g, with each bit $g_i \in \{0,1\}$, where $|g| = 40$ in our setting. Specifically, $g_0 \sim g_7$ represents rotation, $g_8 \sim g_{15}$ represents x-axis translation, $g_{16} \sim g_{23}$ represents y-axis translation, $g_{24} \sim g_{31}$ represents x-axis scaling, $g_{32} \sim g_{39}$ represents y-axis scaling. The phenotype of $(i/8+1)$ th parameter ρ_i is calculated from corresponding bit part. As an example, the phenotype of rotation (first parameter) can be computed as follow,

$$\rho_0 = \text{decimal}(g_0 \sim g_7)/255 \times 2\pi. \tag{3}$$

After each parameter's phenotype is calculated, the transformation matrices between each p_i and p_i^T can be represented as:

$$M = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; R = \begin{pmatrix} \cos r & \sin r & 0 \\ -\sin r & \cos r & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; Tr = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{pmatrix}. \tag{4}$$

Matrix M represents the scaling, R is the rotation, and Tr is the translation.

3.3 Global sampling (GS) and Local sampling (LS)

For GS, we randomly reverse the bit values of high-order in each parameter with a certain GS probability α. On the other hand, we randomly reverse the bit values of low-order in each parameter with a certain LS probability β. The location to divide the bits into low-order or high-order is tunable. In our setting, the center position is used. For example, if we have a bit part $g_j \sim g_{(j+7)}$, after GS and LS operations are performed, the value of a certain bit g_i in $g_j \sim g_{(j+7)}$ can be represented as:

$$g_i = \begin{cases} 1-g_i & \text{with probability } \alpha, \text{ if i} < \text{j}+4 \\ 1-g_i & \text{with probability } \beta, \text{ if i} \geq \text{j}+4 \end{cases}, \; \alpha < \beta. \tag{5}$$

By defining this, the probability of parameters' changes can be controlled. Taking rotation as an example, after GS and LS, the probability of individual 10001000 (192°) to be within range [10000000, 10001111] ([180°, 202°]) is nearly $(1-\alpha)^4$. Since α is suggested to be set smaller than β, sampling procedure will be done mainly around the elite in each generation (similar with crossover), while has lower probability to start searching a new space (similar with mutation). When α equals to β, our algorithm degenerates to a completely random sampling method.

We use roulette selection scheme to select better individuals, which is the same with GA. Without selection, the "adaptiveness" can not be realized. In summary, our method can be concluded in Figure 2.

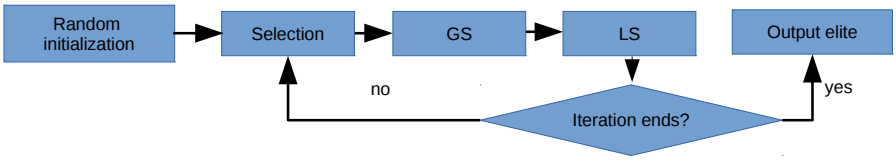


Figure 2: Overview of our algorithm.

4 Experiment

To evaluate our algorithm, we create a benchmark which contains 11 sequences, focusing on matching a capital alphabet "F" by a given template. In each sequence, the ground truth position of "F" is rendered at different places and rotated at an equal interval of 10 degrees. Thus, each sequence contains 36 images. Pixels in the ground truth rectangle are all in the target image. In our experiment, only one template is used, which has a

 ISBN 978-0-9934207-0-2

size of 15 × 17 pixels and shown in Figure 3. Overlap error is used to quantify the accuracy between matching result T and ground truth T', which is defined as:

$$O(T, T') = 1.0 - \frac{\text{area(T)} \cap \text{area(T}')}{\text{area(T)} \cup \text{area(T}')} \tag{6}$$

All the experimetns are done on a PC which has a Core i7 2.97GHz CPU and 16 GB memory.

For parameter setting of our method, we set α as 0.1 and β as 0.7. For GA, we set crossover rate as 0.9, mutation rate as 0.05. For clarity, population number and generation number are set as the same. We allow arbitrary rotation angles and both the x-axis and y-axis scaling are within [1.0,5.0].

Table 1 shows the minimum overlap error in each sequence by comparing our method with simple GA. Number refers both the population number and generation number. As we can see, the minimum overlap error of each sequence belongs to our method. Figure 4(a) ~ (k) show the comparative results of each sequence in curves. We can observe that with small population number and generation number, GA usually outperforms our method. However, with the increase of population number and generation number, our method reduce the overlap error rapidly. Figure 4(l) shows the comparative result of average processing time. Because our method does not need to swap the genes on couples of individuals like crossover, the processing time required is reduced, especially the population and generation number are large. Figure 5 shows some examples of matching results. The matching results can well fit the ground truths and appear to be robust with scaling and rotation.

Figure 3: Template used in experiment (truth size: 15 pixels × 17 pixels).

Table 1: Minimum overlap error in each sequence which is shown in bold.

	Seq01		**Seq02**		**Seq03**		**Seq04**		**Seq05**	
number	**Our**	GA	**Our**	GA	**Our**	GA	**Our**	GA	**Our**	GA
30	0.82	0.45	0.86	0.83	0.92	0.87	0.79	0.6	0.78	0.46
60	0.71	0.53	0.78	0.58	0.91	0.86	0.28	0.26	0.63	0.36
90	0.25	0.26	0.31	0.45	0.88	0.82	0.42	0.29	0.29	0.39
120	0.29	0.29	0.26	0.37	0.64	0.85	0.13	0.19	0.35	0.37
150	0.25	0.28	0.22	0.31	**0.57**	0.85	0.14	0.22	0.3	0.3
180	**0.12**	0.22	0.23	0.31	0.83	0.81	0.14	0.19	0.31	0.31
210	**0.12**	0.21	**0.18**	0.34	0.72	0.78	0.17	0.15	0.2	0.32
240	**0.12**	0.19	0.2	0.34	0.71	0.77	0.11	0.16	0.28	0.33
270	**0.12**	0.17	**0.18**	0.33	0.62	0.78	0.12	0.17	**0.15**	0.3
300	**0.12**	0.14	0.24	0.32	0.62	0.77	**0.1**	0.17	0.2	0.29
Seq06		**Seq07**		**Seq08**		**Seq09**		**Seq10**		**Seq11**	
Our	GA	**Our**	GA	**Our**	GA	**Our**	GA	**Our**	GA	**Our**	GA
0.67	0.5	0.55	0.5	0.81	0.44	0.8	0.5	0.41	0.61	0.63	0.72
0.44	0.44	0.77	0.27	0.19	0.32	0.55	0.38	0.42	0.24	0.25	0.38
0.43	0.33	0.27	0.23	0.35	0.36	0.64	0.3	0.14	0.2	0.36	0.31
0.26	0.37	0.27	0.24	0.19	0.22	0.29	0.27	0.25	0.26	0.22	0.39
0.35	0.37	0.15	0.18	0.17	0.21	0.26	0.32	0.11	0.23	0.2	0.3
0.24	0.37	0.17	0.14	0.11	0.15	0.27	0.31	0.13	0.19	0.26	0.34
0.18	0.28	0.17	0.17	**0.1**	0.16	0.19	0.29	0.11	0.18	0.25	0.28
0.21	0.31	0.13	0.17	**0.1**	0.17	0.22	0.28	0.11	0.17	0.23	0.27
0.23	0.27	0.11	0.15	0.12	0.16	**0.17**	0.3	**0.1**	0.14	0.26	0.29
0.24	0.27	**0.1**	0.15	**0.1**	0.16	0.18	0.26	0.11	0.17	**0.14**	0.28

 ISBN 978-0-9934207-0-2

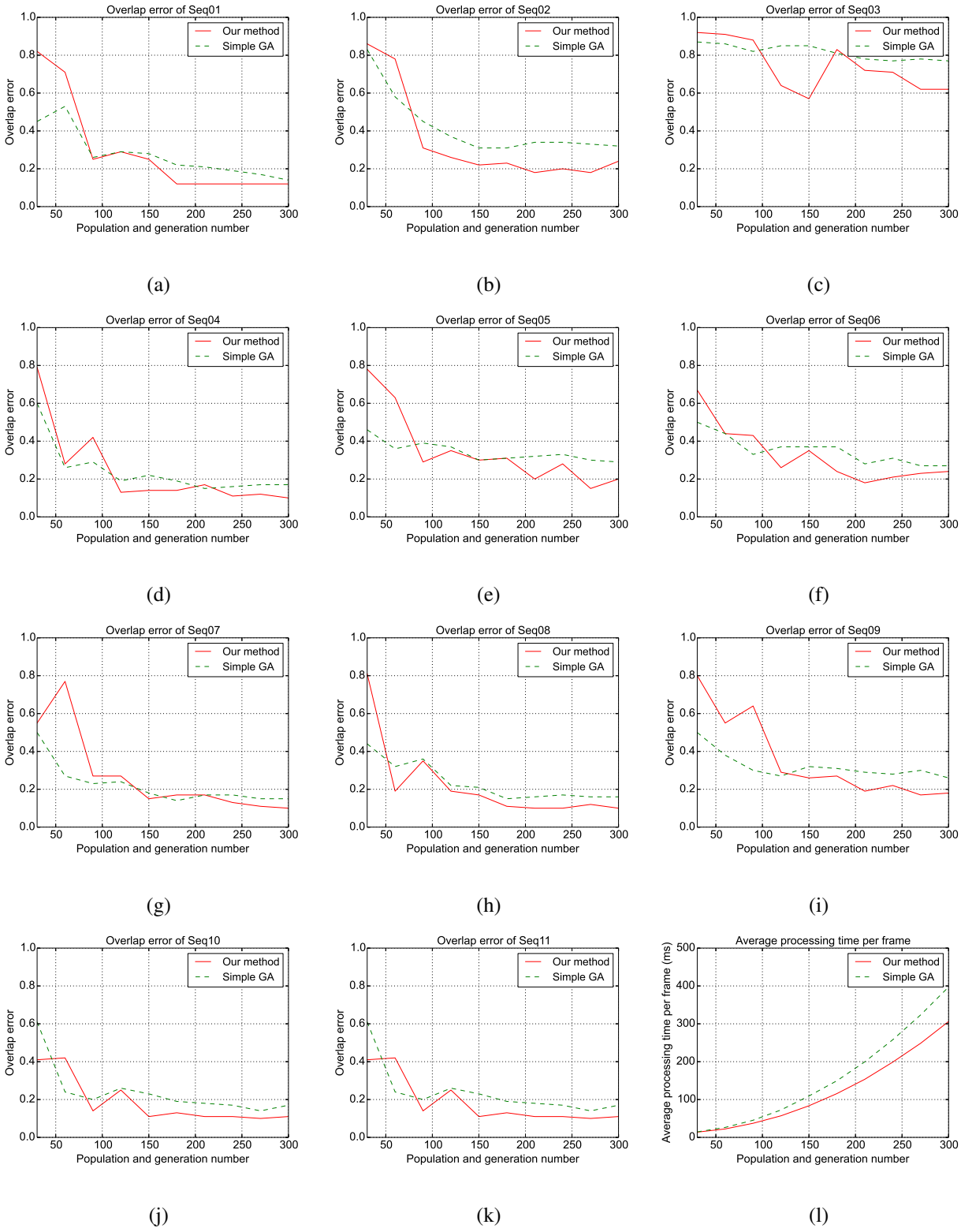


Figure 4: (a)~(k) Comparative results over each sequence. (l) Average processing time of each test image.

 ISBN 978-0-9934207-0-2

5 Conclusion

In this paper, we presented a method to solve template matching problem with scaling and in-plane rotation. For efficiency, we proposed global sampling operation and local sampling operation to simplify the simple GA algorithm. Instead of crossover and mutation, we locate and refine the matching result by controlling the search scope of the whole sampling space. Experiments have shown that our algorithm is more accurate and faster than existing simple GA algorithm. The drawbacks of our algorithm can be concluded as: 1) The smooth assumption limits the application of our algorithm. For template with large variation, we will need larger population and generation number to ensure the accuracy. 2) Due to the stochastic behaviour, there is no absolute assurance that our algorithm can find the global optimum solution by the limited matching tests and the results change with the change of random seed. As the future work, we plan to apply our algorithm to real-world applications.

References

[Akashi et al., 2007] Akashi, T., Wakasa, Y., Tanaka, K., Karungaru, S., and Fukumi, M. (2007). Using genetic algorithm for eye detection and tracking in video sequence. *Journal of Systemics, Cybernetics and Informatics*, 5(2):72–78.

[Choi and Kim, 2002] Choi, M.-S. and Kim, W.-Y. (2002). A novel two stage template matching method for rotation and illumination invariance. *Pattern recognition*, 35(1):119–129.

[Kim and de Araújo, 2007] Kim, H. Y. and de Araújo, S. A. (2007). Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast. In *Proceedings of the 2Nd Pacific Rim Conference on Advances in Image and Video Technology*, pages 100–113. Springer-Verlag.

[Korman et al., 2013] Korman, S., Reichman, D., Tsur, G., and Avidan, S. (2013). Fast-match: Fast affine template matching. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2331–2338. IEEE.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[Morel and Yu, 2009] Morel, J.-M. and Yu, G. (2009). Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469.

[Tsai and Chiang, 2002] Tsai, D.-M. and Chiang, C.-H. (2002). Rotation-invariant pattern matching using wavelet decomposition. *Pattern Recognition Letters*, 23(1):191–201.

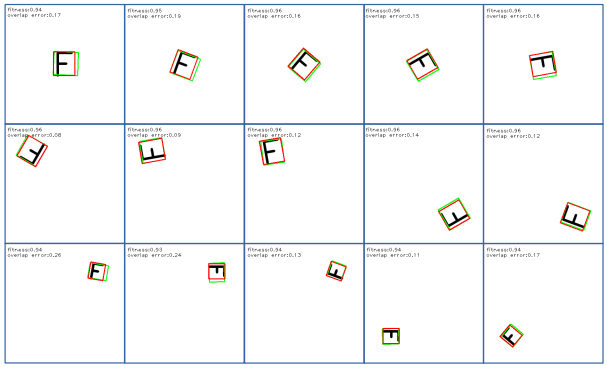


Figure 5: Examples of matching results. Red rectangle represents the ground true, green rectangle represents the matching result.

Automatic Segmented Area Structured Lighting

Kruti Goyal, Hadi Baghsiahi & David R Selviah

Department of Electronic and Electrical Engineering
University College London (UCL), Torrington Place WC1E 7JE London
d.selviah@ucl.ac.uk, http://www.ee.ucl.ac.uk/staff/academic/dselviah

Abstract

The aim of the research is to devise an automatic way to view and segment a scene of discrete 3D objects with or without ambient illumination and then to fully illuminate each object in turn without illuminating other objects or the background. The structured illumination must be controlled in time and space to have the same shape, size and position as the object. There is a need for such a system in the entertainment and visual arts industries for sound and light shows at night outdoors for selectively illuminating buildings, in caves or caverns for illuminating rock formations, or for illuminating mannequins, statues or waxwork figures in theatres sequentially in synchrony with a voiceover narration discussing each in turn. In these applications, such a technique has an advantage over the use of spotlights as only the object of interest is illuminated and not nearby objects or the background so helping the viewer to concentrate on just the object of interest. In this paper a video camera and projector system is reported with real time image processing feedback via a computer. The way in which the image processing algorithms in the feedback loop were developed to overcome various issues is explained.

Keywords: Object-Background segmentation, Structured illumination, Spotlight, Signal processing, Image alignment

1 Introduction

The research aims to design and demonstrate a camera and projector intelligent system to illuminate 3D Objects within a 3D scene while minimizing illumination of other nearby objects and the background of the scene. The system should be capable of illuminating each individual object alone and alternatively illuminating each individual object in a sequence in turn. The system can be used to illuminate all of the objects in the scene at the same time or a single object can be illuminated while the rest of the objects and the background are under background are kept dark. The research has various applications in the entertainment industry such as in outdoor sound and light shows [1] in which individual buildings [3] or statues are illuminated in turn either in time to music or synchronised with explanations about the objects given by a speaker [2], in underground cave tours where individual rock formations are illuminated in turn as a speaker explains them [4], recently it has also been popular to project images onto mannequins and statues

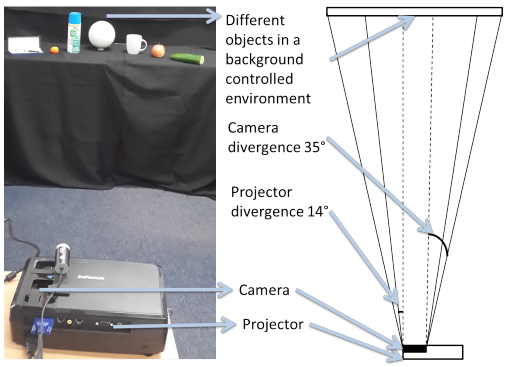


Figure 1: Experimental Arrangement and Bird view of the arrangement

 ISBN 978-0-9934207-0-2

to make them appear to be talking. In such situations, the events usually take place at night or in dark rooms and the difficulty is how to arrange the projector to mainly illuminate just one object without illuminating other nearby objects. The alignment and calibration of the projector can be very time consuming. In this paper we describe the development of a camera and projector system with image processing software to carry out this automatic alignment of the projected image and the 3D objects. In this paper only uniform illumination is projected although in future it could be projected moving images.

The arrangement used for the paper is as shown in figure 1. A video camera is placed on top of the projector, ans is secured to it, and the output of the camera and input of the projector are connected to a HP Pavilion DV6 3107ax laptop computer which requires 90 W AC power and has 500 GB memory. The video camera was chosen to be Microsoft Lifecam Cinema which has a USB 2.0 connection, a speed of 30 fps (frames per second) and is powered by the computer through the USB 2.0 link. The resolution of the camera was 960 × 544. Thw projector chosen for the project was an Infocus X9 DLP project with a brightness of 1800 ANSI lumen and 1400 × 1050 resolution. The camera is rotated so that its horizontal axis matches that of the projector so that the captured images do not need to be rotated. The camera is placed on the top of the projector to minimise the distance between the two. In the experiments reported in this paper the distance between the center of the lenses of the camera and projector is 8 cm. This distance is measured vertically as the camera lens was placed directly above the projector lens. A dark background is used to simulate the typical environments used in sound a light shows. This also helps to achieve proper object background segmentation due to the intensity difference between the background and the objects to be illuminated. A light background and dark foreground can be used alternatively, in which case the picture needs to be inverted before further processing.

The paper focuses on three main issues which need to be solved. Firstly, the system should be able to automatically detect and separate the objects from their background. Secondly, the projector must project image such that it matches the objects detected by the camera exactly. Finally, only the specified objects should be illuminated at a particular time keeping all of the other objects and the background in the dark. This whole process should be carried out automatically in realtime and should be robust to changes in the ambient light. In order to achieve this the camera-projector system is used in conjunction with image processing algorithms written in LabVIEW is used.

The surroundings contain 3D objects, therefore, the detection technique should be such that the appropriate information of the surroundings is retrieved when the 3D environment is converted into a 2D image. Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is graphical programming language used for computer vision and image processing and was chosen over MatLab due to its speed and graphical user interface which enables various functions such as different filters to be used without changing the code or need to recompile the code and due to LabVIEW's better hardware-software interface. LabVIEW image acquisition was used to capture the surroundings and the image was then processed as presented in section 2. The next step is to determine the best match between the projected illumination and the object such that the projected light is exactly on top of the object. Normalised cross correlation between the transformed captured image with and without projection was used for this purpose. The disadvantages of cross correlation includes sensitivity to illumination, rotation and scale changes. However, in this research they prove to be advantageous as due to the difference in normalised cross correlation value for different scale, illumination or rotation, the best match can be found. The objects were then selected and lit sequentially using Magic Wand Processing, a function provided by National Instruments as part of NI Vision, as presented in section 3.

The whole process takes about 35 seconds to one minute 15 seconds with a 4 GB AMD Phenom Quad-core processor and is independent of the number of objects used. However, the speed of processing is highly dependent on the processor used. Onboard FPGA image processing can give much higher processing speed than the laptop computer. The speed of the program also depends on the RAM available with higher RAM increasing the speed of processing significantly. The processing time is independent of number of objects detected, and, hence, any number of objects can be detected by this method.

 ISBN 978-0-9934207-0-2

2 Object-Background Segmentation and Image Processing

Each object needs to be segmented and separated from the background and other objects. Also the noise (unwanted particles and bright spots due to details of the scene and light sources and unwanted reflections) needs to be minimised [5–8]. LabVIEW 14.0 student edition together with IMAQ 2014, IMAQdx 2014, NI VISA 2014 and NI Vision acquisition 2014 and NI Vision assistant 2014 [9, 10], all provided by National Instruments, were used for this purpose as they conveniently have a variety of pre-written functions relating to image processing and computer vision and they provide various drivers for hardware integration.

2.1 Spatial Image Filtering

The captured RGB image is converted to greyscale and is segmentation filtered to separate the object from the background and from other objects. Different filters were tried to find the most appropriate filter. A Gaussian filter [11–14] was first applied to separate the objects from their background. It has the effect of keeping the important patterns while removing finer details, however, it also increases the noise present (figure 3). If the light is equally distributed across the image and bright, the Gaussian filter saturates and the whole image becomes white. In this case, the alternative is to use a low pass filter. The effect of the low pass filter is minimal, except that it smooths the boundaries of objects. If lowpass filter is used for figure 2, which is not uniformly lit, the object to background intensity difference will be low and the objects would not be detected, this is evident in later stages of image processing, that is after thresholding as shown in figure 4. Even when the kernel's (Convolution matrix determining the weight of the pixels under consideration and their neighbouring pixels) size is increased, the result is still the same as shown in figure 4. The final result after image transformations as described is section 2.2 results in a black image if a low pass filter is applied for non-uniformly lit environment. If no filter is applied, the result is the same as shown in figure 4. Hence, a Gaussian filter is suitable for non-uniformly lit enviorment such as that of figure 2, whereas lowpass filter is ideal for uniformly lit enviorment. Therefore, either a Gaussian or low pass filter is selected for object background segmentation depending on the ambient illumination conditions. A Gaussian kernel of size 3×3 with the following dimensions were used as they were found to give the best results.

$$G = \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$$

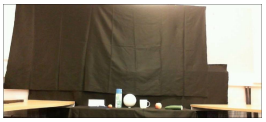


Figure 2: Captured image of the arrangement of 3D objects within the scene from the CCD camera

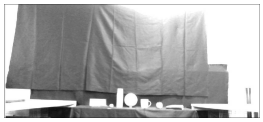


Figure 3: Image after application of a Gaussian Filter

Figure 4: Image after application of a Low Pass filter

2.2 Image Transformation and Morphology

The greyscale image is eroded [15] for four iterations to remove the finer details and bright spots, so that they do not connect two adjacent objects and make them appear as one, and also so that they are not mistaken as objects themselves. The number of iterations was varied and four was found to yield the best results. If a pixel under consideration has the binary value 1 and if all of its cardinal neighbouring (pixels to the immediate left and right, and top and bottom of the pixel under consideration) pixel values are 1 then the new pixel value is set to 1, otherwise it is set to 0. The erode operation is defined by:

 ISBN 978-0-9934207-0-2

$$X_{new} = \begin{cases} 0 \ if & x_{old} = 0 \\ 1 \ if & x_{old} \,\&\, (Cardinal = 1) \end{cases}$$

Next, the image is dilated for four iterations, to bring it back to its original size. The number of iterations was varied and four was found to yield the best results. The dilation operation sets the new value of the pixel to be 1 if the original pixel is 1 or if its cardinal neighbours are 1, otherwise the value is set to 0. The dilate operation is defined by:

$$X_{new} = \begin{cases} 1 \ if \ x_{old} \, OR (Cardinal = 1) \\ 0 \ if \quad otherwise \end{cases}$$

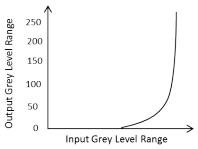


Figure 5: Exponential operation

A non linear exponential operation is applied to the pixel grey level intensities next using a look up table [16] to improve the contrast ratio and to remove noise due to intensity variation. This operation decreases the contrast and brightness in dark regions, whereas it increases the contrast in bright regions as can be seen from the non-linear curve in figure 5.
Next, the image is thresholded to separate the bright object from the dark background. A high and low threshold value is chosen, the pixels within this range is detected as the object whereas all the other pixels are darkened. As a black background is used and objects nearer to the camera are brighter than those further from the video camera, this essentially separates the object from its background. This step however, is greatly affected by the brightness of the room and the camera used, and needs to be varied in different ambient illumination. If the room is bright, the objects and background will reflect more light, and, hence, a high lower threshold value needs to be set. Whereas, if the room is dark, the reflected light from the objects and the background will be weaker, and, hence, a low lower threshold value needs to be set. Therefore, to avoid setting up a manual threshold every time the program runs, the maximum intensity of the greyscale image is found and a percentage of this (20 and 35 % were tried for the experiment) is taken as the lower threshold, the upper threshold is set to a constant, which is its highest value (255 for 8-bit image). Bright spots on the border of the image can be mistaken as objects and, hence, all the objects connected to the border are removed next. This step is optional, as in many cases the object can be near the border. The image is then equalised by distributing the pixels in intensity evenly and converting it to an 8−bit image from a binary image. These image transformation operations are as shown in figure 6. These transformation were applied on images taken in a room with ceiling light and windows opposite to the scene. If the room is completely dark in the detection stage, the program would not be able to segment the objects from its background. However, the sequential illumination of the objects can be done in a dark room using the information obtained in the previous stages. The thresholding operation is given by:

$$i = \begin{cases} 0 \ if & p < t_{low}, p > t_{high} \\ 1 \ if & t_{low} \leq p \leq t_{high} \end{cases}$$

3 Calibration and Projection

The video camera has a greater field of view than that of the projector. To detect the area of projection in the captured image, a white image with the same pixel area as the projector is projected to illuminate the field of

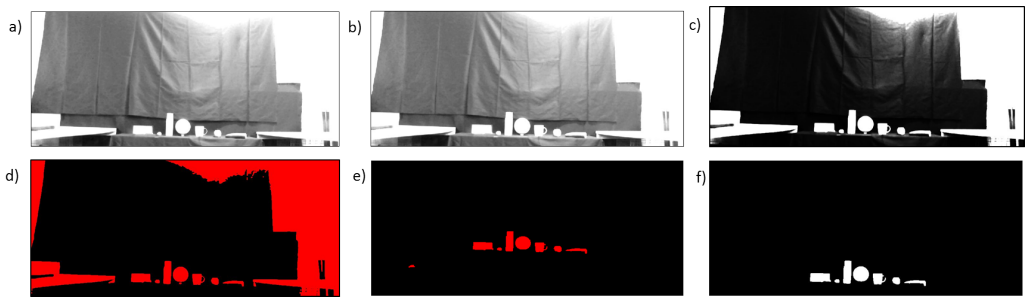


Figure 6: Image transformation: a) Erode b) Dilate c) Exponential d) Threshold e) Remove border objects f) Equalise

view of the projector. This area also called, region of interest (ROI), is then extracted from the captured image and image processing is subsequently only carried out within this area. This area selection can be carried out automatically by setting a high value for the lower threshold (around 230) to detect the area of projection, as it will very bright due to projection of a white image onto the scene. This area is then selected by using the magic wand function, which compares the intensity of the pixel under consideration with its neighbours and their neighbours, and selects all connected pixels which have a similar intensity within the set tolerance limit (20%). The coordinates for a particle under consideration are found using the particle analyser function in LabVIEW and correspond to the centroid of the bright particle with the maximum area. The maximum area is selected because there might be some other regions with high intensity which can be taken as the ROI resulting in the wrong selection. However, depending on the surface of the background, some regions in the projected area might not fall into the threshold range and are ignored, resulting in an error; this error, however, does not affect the final result significantly as the ROI selected is defined to be a rectangle.

After the ROI is extracted, one object is selected to be illuminated. Next, the processed image of the object is projected back through the projector onto the object. Automatic thresholding on the image can now be used (entropy [17–19] in this case), as the brightness of projected structured light is much greater than the other ambient illumination, hence, the ambient illumination can be ignored.

There may still be some difference between the edge of the area of the projected light and that of the real object due to the skew of the projector lens, the distance between the projector lens and the camera lens and error in locating and extracting the ROI. This error is corrected using a feedback loop. The scene onto which processed images are projected is continuously monitored using the video camera, and the latest captured and processed image (which is dominated by the illumination area shape) is correlated with the initial captured and processed image (Template). The dimensions and position of the projected image are altered until a maximum correlation is achieved. The use of one object instead of all, decreases the processing time and reduces errors. The projected structured light is rescaled, moved up and down and left and right, and the normalised cross correlation is calculated for each position and size. The final algorithm which moves the image up, down, left, right and varies the scale works well but it would be useful to incorporate additional transformations such as keystone to correct for the skew due to the 8 cm offset between the camera and the projector lens. Once the maximum correlation is reached when the projected image is exactly aligned with the real object, the calibration process ends. The dimensions and coordinates of the projected image are then recorded for further use.

The acquired illuminated image is then analysed using NI Vision particle analyser. Particle analyser can detect continuous regions or grouping of pixels with similar intensity, known as particles. The particle analyser then makes measurements of these regions or particles and also determine their coordinates. This is often referred to as BLOB (Binary Large Object) analysis. The particle analyser detects each particle and returns the coordinates of each object detected. The program then takes these pixel coordinates and uses magic wand processing which uses the intensity of neighbouring pixels, to select only one object and mask all the others, so that only one object can be illuminated at one time. The projected individual object images can then be projected sequentially to illuminate each object one after the other as required.

4 Experimental Results Discussion

The research investigated different algorithms, to select the best possible method. It was also conducted at different times of day to check the performance under different ambient background lighting conditions and at different locations with different 3D scenes to verify its robustness. The results obtained are discussed in this section. The area of interest is as shown in figure 7. The arrangement consist of a pen case, an orange, a bottle, a sphere, a mug, an apple and a cucumber, from left to right, the camera-projector system is 1.26 m from the objects. The sphere is illuminated for calibration purposes in figure 8 and 9, however, any object can be selected for this purpose. When the projection is of a different size than the object it is resized using a scale transformation as shown in figure 8. Next, the projection is moved in the x and the y directions to meet the edges of the object as shown in figure 9.

The objects were lit sequentially and all at once as shown in figures 10, 11, 12 and 13. The algorithm was

Figure 7: Projected image to detect ROI

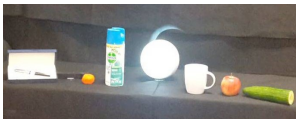


Figure 8: Resizing of the projection to match the sphere

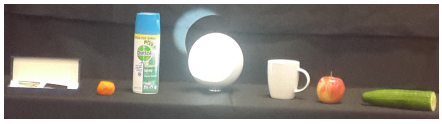


Figure 9: Shifting the projected light to the left

also applied to different types of object as can be seen in the figures. The objects had various shapes, colours, textures and reflectivities. All of the objects were placed side by side at different positions on the table and could all be lit at the same time. To check the validity of the algorithm objects were detected and illuminated from different positions. It was found that the closest position the camera can detect the object and area of interest without becoming saturated is 14 cm. The longest distance, however, is only limited by the power of the projected light and in this case is about 15-16 m. There is no limit on the area of the largest detectable object, as long as the camera can detect it and it comes within the projector's field of view. The smallest object which can be detected without being eliminated as background noise is about 5 pixels or 0.14 pixels per degree according to camera's resolution, and, hence, very small objects can be detected if they are close to the camera-projector system but the same object is neglected as noise when it is far from the camera-projector system. Objects were placed close and far from the system simultaneously and illuminated as shown in figure 14 and the system was shown to illuminate them all correctly. Complicated and large shapes such as human beings can also be detected by the system as can be seen in figure 15.

Figure 10: Light projected onto a Pen Case

Figure 11: Light projected onto the Sphere

Figure 12: Light projected onto a mug

Figure 13: Light Projected on all the objects

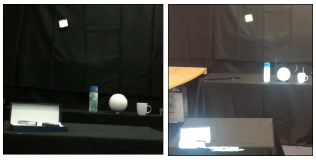


Figure 14: Arrangement (left) of and light projection (right) onto Pen case (closer to projector), post-it, bottle, sphere and cup

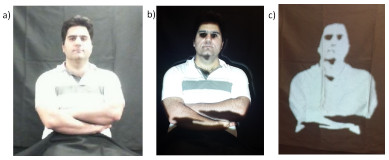


Figure 15: Detecting a complex subject: a) Scene with no projection on the person b) Light projected onto the person c) The detected person projected onto the background

 ISBN 978-0-9934207-0-2

The camera has a divergence of 35 degrees or horizontal divergence of 27 pixels/degree and vertical divergence of 16 pixels/degree, whereas, the projector has a horizontal divergence of 14 degrees or 100 pixels/degree and vertical divergence of 12 degrees or 87 pixels/degree. As the field of view of the camera is larger and the resolution is lower than the projector which decreases even more when part of the image is extracted, there is a difference between the processed image and the image which is projected the and this sometimes results in spillage of light into the background of about 1 pixel or more, even after the calibration is done. This is evident from the background illumination in figure 12 around the mug. This spillage cannot be seen by the camera due to the difference in divergence and point of view of the camera and can be solved by using another camera, which can view this spillage from another angle and counter it by manipulating the light to match the object.

5 Conclusions

The project's aim to make a robust system to illuminate discrete objects individually was achieved. The shifting and resizing process can be calibrated to enable the illumination to automatically illuminate an individual object to one pixel accuracy. The illumination became distorted when it is passes into the camera due to the magnification factor between the scene and the camera's detector array which affects the result of the filters and the result of correlation. This in turn affects the output shift and size of the projected image as due to distortion the position and size of the projection is deemed to be correct by the program, when in reality it may not be. These effects can be minimised by using a high resolution, anti-glare camera.
The camera used for this project has a resolution of 960 × 544 whereas the projector has a resolution of 1400 × 1050. In addition the angular divergence of the camera, 35 degrees, was much larger than that of the projector, 14 degrees, which aggravated the difference between their resolutions resulting in an angular resolution of 27 pixels per degree for the camera and 100 pixels per degree for the projector. Ideally the resolution of the camera in pixels per degree should be higher than that of the projector in pixels per degree. This can be achieved by a combination of selecting the number of pixels in the camera sensor plane and controlling the angular divergence by narrowing the cameras field of view using a zoom lens. Another limitation of this method is that some light was projected onto the background, which cannot be seen by the camera but is seen by the viewer at a different angle from the camera. The program's speed is processor dependent and takes about 35 seconds to 1 minute 15 seconds to complete with the current system. Therefore, the research was successfully able to achieve object-background segmentation, calibration and sequential projection. However, the aforementioned limitations such as projection on the backdrop of the object and effects due to environment; limit the application of this research and makes it viable for further investigation.

Acknowledgements

We express our gratitude to Dr. Sally Day for her support and her generosity in providing a work space and apparatus required.

References

[1] Dream makers. 2010. [Online]. Available from: http://www.soundandlight.com.eg/Shows/ GizaPyramidsHome/VideoPics/VideoTube2.aspx. [Accessed 20 February 2015].

[2] D. Bowies. 2015. [Online]. Available from: http://flavorwire.com/422969/deconstructing-david-bowies-diy-video-for-love-is-lost. [Accessed 20 February 2015].

[3] Shader Lamps. 2015. [Online]. http://web.media.mit.edu/ raskar/Shaderlamps/OtherImages/index.html. [Accessed 20 February 2015].

[4] Poole's Cavern. 2015. [Online]. Available from: http://www.poolescavern.co.uk/visitor-information/. [Accessed 20 February 2015].

[5] Szeliski, R. (2010). Computer Vision : Algorithms and Applications. Springer Science and Business Media, 1, 2, 3. London.

[6] Klinger, T. (2003). Image Processing with LabVIEW and IMAQ Vision. Prentice Hall Professional. Upper Saddle River, New Jersey.

[7] Pratt, W. (1991). Digital Image Processing, Second Edition. John Wiley and Sons, Inc. New York, NY. ISBN 0 – 471 – 85766. pp 561 – 563.

[8] Lewis, R. (1990). Practical Image Processing. Ellis Horwood Limited. New York, NY. ISBN 0 – 13 – 6383525. pp 90 – 91.

[9] Fairweather, I., Brumfield, A. (2011). LabVIEW: A Developer's Guide to Real World Integration. Chapman and Hall/CRC. Boca Raton, FL. ISBN 9781439839812.

[10] Posada-Gomez, R., Sandoval-Gonzalez, O.O, Sibaja, A.M., Portillo-Rodriguez, O., Alor-Hernandez, G. (2011). Digital Image Processing Using LabVIEW, Practical Applications and Solutions Using LabVIEW Software. Dr. Silviu Folea (Ed.). ISBN: 978-953-307-650-8. InTech, DOI: 10.5772/23285. Available from: http://www.intechopen.com/books/practical-applications-and-solutions-using-LabVIEW-software/digital-image-processing-using-LabVIEW. [Accessed 20 November 2014]

[11] National Instruments. (2000). IMAQ Vision for LabVIEW. Available from: http://www.ni.com/pdf/manuals/371007a.pdf. [Accessed 20 November 2014]

[12] Relf, C. G. (2004). Image Acquisition and Processing with LabVIEW. CRC Press LLC. Boca Raton, FL. 2^{ed}. USA. ISBN 0-8493-1480-1.

[13] Version, I., Bhuvaneswari, P., Therese, A. B. (2014). Edge Detection Techniques in Digital and Optical Image Processing. Int. Journal of Engineering Research and Applications. ISSN : 2248-9622. Vol. 4, Issue 5 (Version 3). pp.33-37.

[14] Ravi Kumar V. A., Nataraj, K. R., Rekha, K. R. (2012). Morphological Real Time Video Edge Detection in LabVIEW. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2). pp 3808-3811.

[15] Xiaobo, M., Jing, Y. (2011). Research on object-background segmentation of color image based on LabVIEW. 2011 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, DOI: 10.1109/CYBER.2011.6011791. pp 190-194.

[16] National Instruments. (2000). IMAQ Vision Concepts Manual, (322916). [online]. Available from: http://www.ni.com/pdf/manuals/322916a.pdf. [Accessed 26 November 2014].

[17] Gonzalez, R.C., Woods R.E, S.L. Eddins. (2009). Digital Image Processing Using MATLAB. 2nd ed., Gatesmark Publishing. Knoxville, TN. Prentice Hall. Chapter 11.

[18] Loomis, J. (1998). (2015). Entropy. [online]. Available from: http://www.johnloomis.org/ece563/notes/basics/entropy/entropy.html. [Accessed 25 November 2014].

[19] Iliev, P., Tzvetkov, P., Petrov, G. (2014). Motion Detection Using 3D Image Histogram Sequences Analysis. [online]. Available from: http://nbu.bg/PUBLIC/IMAGES/File/departamenti/telekomunikacii/1.pdf. [Accessed 25 November 2014].

Machine Learning in Prediction of Prostate Brachytherapy Rectal Dose Classes at Day 30

P. Leydon[1,2], F. Sullivan[2], F. Jamaluddin[2], P. Woulfe[2], D. Greene[3], K. Curran[1].

[1]*School of Medicine & Medical Science, University College Dublin, Ireland*

[2]*The Galway Clinic Ireland*

[3]*School of Computer Science & Informatics, University College Dublin, Ireland*

Abstract

A retrospective analysis of brachytherapy implant data was carried out on 351 patients that underwent permanent I^{125} brachytherapy for treatment of low-risk prostate cancer. For each patient, the dose received by $2cm^3$ of the rectum (D2cc) 30 days post implant was defined as belonging one of two classes, "Low" and "High" depending on whether or not it was above or below a particular dose threshold. The aim of the study was to investigate the application of a number of machine learning classification techniques to intra-operative implant dosimetry data for prediction of rectal dose classes determined 30 days post implant. Algorithm performance was assessed in terms of its true and false positive rates and Receiver Operator Curve area based on a 10-fold cross validation procedure using Weka software. This was repeated for a variety of dose class thresholds to determine the point at which the highest accuracy was achieved. The highest ROC areas were observed at a threshold of D2cc = 90 Gy, with the highest area achieved by Bayes Net (0.943). At more clinically useful thresholds of D2cc = 145 Gy, classification was less reliable, with the highest ROC area achieved by Bayes Net (0.613).

Keywords: Brachytherapy, Prostate Cancer, Machine Learning, Classification, Weka.

1 Introduction

Prostate cancer is the second most frequently diagnosed cancer and the sixth leading cause of cancer death in males, accounting for 14% (903,500) of the total new cancer cases and 6% (258,400) of the total cancer deaths in males in 2008 [Jemal et al., 2011].

The permanent transperineal interstitial placement of I^{125} seeds is a popular choice for Low Dose Rate Brachytherapy of Prostate Cancer. The deposited-seed positions are imaged and the plan optimized in real-time throughout the procedure. The dose distribution is updated dynamically based on the actual positions as the seeds are deposited. The clinical and technological improvements that have emerged over the last decade in low dose rate prostate brachytherapy such as the use of ultrasound and computed tomography-based treatment planning systems has led to a resurgence in the use of the technique for localized prostate cancer [Jemal et al., 2011].

As prostate cancer is being diagnosed at increasingly early stages, quality-of-life issues when choosing the primary treatment modality are becoming of greater concern. Prostate brachytherapy has many advantages over both external-beam radiation therapy and radical prostatectomy, however it has been shown that because the rectum often receives a large radiation dose this may lead to radiation-induced rectal injury such as rectal

 ISBN 978-0-9934207-0-2

bleeding, chronic radiation proctitis which if not managed correctly can result in fistula development and the eventual need for a colostomy.

With regards to rectal dosimetry, the guidelines given by Snyder et al based rectal dose constraints on an annular dose-volume histogram of the rectum, aiming to achieve less than 160 Gy to 1.3 cm^3 for I^{125} monotherapy. This rule is often modified by practitioners who prescribe 145 Gy minimum peripheral dose (mPD), and for doses to 2.0 cm^3 of the rectum to be less than 145 Gy [Nath et al., 2009].

Due to the permanent nature of the implant the dose is determined by integrating the dose rate from the time of implantation until the isotope has decayed to background levels, taking into account the physical decay of the sources only and assuming that the geometry of implant and anatomy established at the date of scan does not change over time.

To account for this time variance in prostate volume, seed position and the subsequent dosimetry calculations, the post-implant CT scan is often postponed for approximately 1 month after the procedure assuming that the prostate volume, seed positions etc will change very little afterwards [Steggerda et al., 2007]. Changes in rectal doses have been observed between measurements made intraoperatively and from a patient's post-implant CT. The differences have been largely attributed to changes in rectal proximity to the implanted seeds as periprostatic edema resolves [Nag et al., 1999]. Discrepancies in rectal doses may also be as a consequence of the different patient set-ups and the resulting variation in prostate positions during intra-operative transrectal ultrasound (dorsal lithotomy position) and postplanning CT (supine position) [Pinkawa et al., 2009].

The aim of the study was to investigate the application of machine learning techniques to intra-operative implant dosimetry data for prediction of rectal dose classes determined 30 days post-implant.

2 Methods

A retrospective analysis of brachytherapy implant data was carried out on 351 patients that underwent permanent I^{125} brachytherapy for low-risk prostate cancer. For each patient, the dose received by 2 cm^3 of the rectum (D2cc) at day 30 was defined as belonging to one of two classes, Low <(Dose Threshold), and High >(Dose Threshold). The original dataset, n = 351, contained approximately 40 intra-operative measurements which was then reduced to a selection of 10 predictors, using the "Best First" method in Weka [Hall et al., 2009]. Any data with missing values were excluded, leaving the final refined set of n = 347. The day 0 predictors used in this study were, (i) Treatment Type, (ii) Pre-Needle Prostate Volume, (iii) Post-Needle Prostate Volume, (iv) Dose to 90% of the Prostate (D90), (v) Minimum Dose to Prostate, (vi) Mean Dose to Prostate, (vii) Dose to 30% of the urethra (U30), (viii) Dose to 10% of the urethra (U10), (ix) Dose to 5% of the urethra (U5), and (x) the dose to 2 cm^3 of the rectum (D2cc).

Variable	Value ± SD	Range
Age	63.41 ± 6.99	42.07 - 83.27
Prostate Volume cc	33.47 ± 10.96	11.5 - 62
Treatment Type	Mono(258), Boost(83), Salvage(10)	
Activity per seed	0.51 ± 0.04	0.35 - 0.58
Total Activity mCi	27.37 ± 8.68	9.26 - 84
Number of Seeds	66.83 ± 16.23	26 - 105
Number of Needles	17.95 ± 2.84	10 - 25
Initial PSA ng/ml	7.61 ± 3.70	1.5 - 29
Gleason		(3+1) - (5+5)
ADT	~28%	
Pre Needle Volume cc	31.06 ± 10.62	10.23 - 62.3
Post Needle Volume cc	32.91 ± 11.91	11.4 - 64.7
D90 Day 0	155.30 ± 24.89	85.96 - 192.34
Minimum Dose Day 0	97.72 ± 22.08	11.62 - 179.18
Mean Dose Day 0	230.64 ± 32.89	155.49 - 284.53
U30 Day 0	163.48 ± 26.73	12.06 - 220.69
U10 Day 0	172.128 ± 27.48	110.44 - 248.24
U5 Day 0	175.92 ± 28.78	112.53 - 267.4
D2cc Day 0	82.41 ± 25.37	23.36 - 160.47
D2cc Day 30	111.29 ± 28.28	22.46 - 190.95

Figure 1: Patient dataset summary statistics.

Changes of dose distribution for both the prostate and the surrounding tissues after brachytherapy have been reported in literature, and have been largely attributed to factors such as edema, intra-observer variability, and the use of different imaging modalities with different patient set-up between day 0 and day 30 [Moorrees et al., 2012]. In Figure 2. these changes in rectal doses between day 0 and day 30 are evident, with an obvious overall shift towards higher rectal doses at day

 ISBN 978-0-9934207-0-2

30.

Machine learning techniques, developed using Weka software, were trained on the refined dataset to make predictions of rectal dose classification at day 30. Four algorithms were used;
(i) **Radial Basis Function network (RBF)** - Implements a normalized Gaussian radial basis function network, with k-means clustering algorithm providing the basis functions for logistic regression. Symmetric multivariate Gaussians are fit to the data from each cluster. All numeric attributes are standardized to zero mean and unit variance.
(ii) **J48** - A pruned decision tree based on the C4.8 algorithm [Quinlan, 1993]. The number of folds used was 3, with subtree raising when pruning, and without binary splits on numeric data or Laplace smoothing.

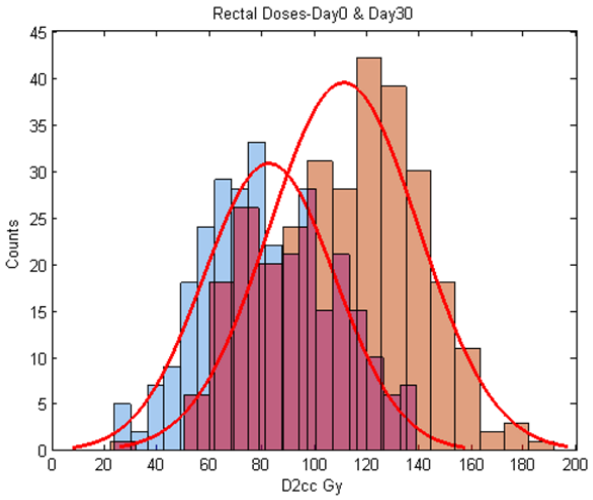


Figure 2: Patient dataset summary statistics.

(iii) **Random Tree** - Constructs a desecion tree that considers a number of randomly chosen attributes at each node. No pruning, and no backfitting was performed.
(iv) **BayesNet** - Uses the K2 hill climbing algorithm restricted by the Weka default variables. The maximum number of parents in the Bayes Net was set to 1, resulting in a Naive Bayes classifier. Conditional probabilites were determined by selecting the Simple Estimator option with alpha = 0.5.

The performance of a technique was assessed in terms of true and false positive rates, and area under Receiver Operator Curves based on a 10-fold cross validation procedure. This process was repeated for a number of different thresholds ranging from D2cc of 50 Gy to 180 Gy in order to determine the point at which an algorithm demonstrated the highest accuracy.

3 Results

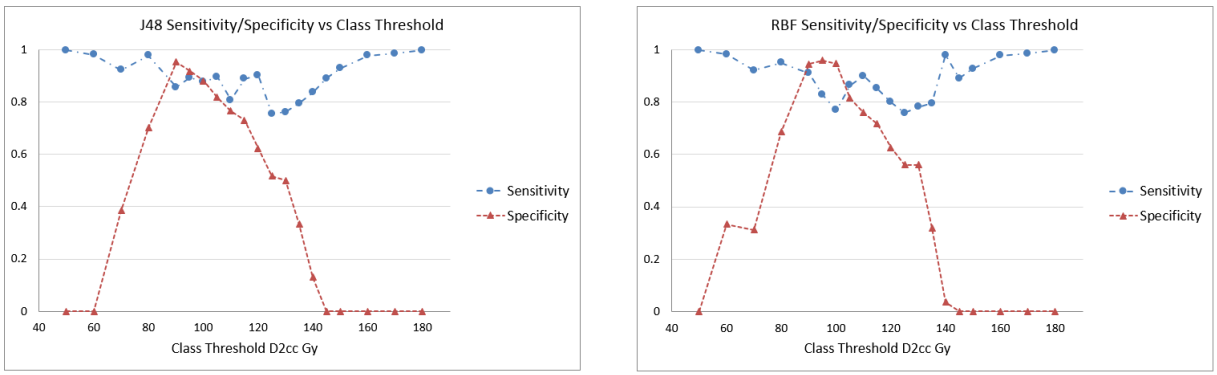


Figure 3: Shows both sensitivity and specificity results for J48 and Radial Basis Function algorithms at various dose thresholds.

The four algorithms displayed a similar performance overall in the threshold ranges tested, with similar changes in sensitivity and specificity clearly visible in Figures 3 & 4. The ROC areas in Figure 5 demonstrate the variation across the threshold range depending on the choice of algorithm; however all appear to give the

 ISBN 978-0-9934207-0-2

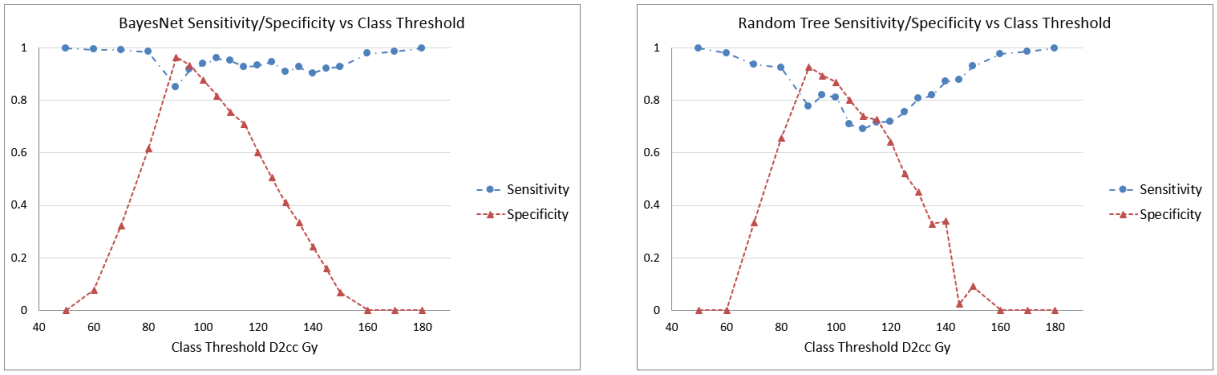


Figure 4: Shows both sensitivity and specificity results for BayesNet and Random Tree algorithms at various dose thresholds.

best result close to a threshold of 90 Gy, with ROC areas of 0.928 (J48), 0.927 (Radial Basis Function), 0.943 (BayesNet), and 0.851 (Random Tree). At the more clinically important threshold of 145 Gy algorithms display considerably less accuracy with ROC areas of 0.477 (J48), 0.660 (BayesNet), 0.613 (Radial Basis Function), and 0.444 (Random Tree).

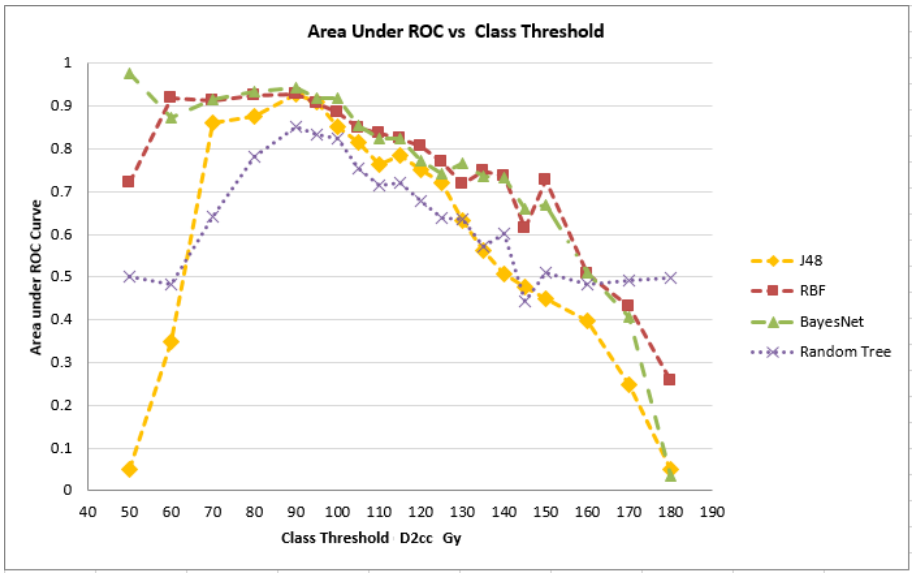


Figure 5: Reciever Operator Curves for all four algorithms.

4 Conclusion

All four of the algorithms appear to predict reliably at a rectal dose threshold of approximately 90 Gy. However, it is prediction at the higher dose thresholds that would be of most use in a clinical setting but as the rectal dose class threshold passes the 90 Gy mark predictions become steadily less reliable.

It may be worth exploring the effect that inclusion, or exclusion of combinations of the 10 predictor variables used in this study has on the accuracy of the algorithms. Some predictors may not be necessary for optimum training of algorithms and may be impeding reliable classification. The possible inclusion of any outliers in training data may also be reducing performance.

It has been by shown that implant variables, such as D90, do not depend on the post-implantation date of the scan supplying the images for dosimetry however some parameters associated with morbidities strongly changed within the first month after implantation, meaning that the time at which a scan is carried out may be an important variable to consider and one which was not included as part of this study [Steggerda et al., 2007].

Weka provides a much larger selection of classification algorithms than the four used in this study, such as Mutli Layer Perceptrons and Support Vector Machines. It may be that one of these other algorithms is more suited to this particular classification task, than those that were used. The inclusion of uncertainties associated with classification predictions would also be of use clinically.

References

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

[Jemal et al., 2011] Jemal, A., Bray, F., Center, M. M., Ferlay, J., Ward, E., and Forman, D. (2011). Global cancer statistics. *CA: a cancer journal for clinicians*, 61(2):69–90.

[Moorrees et al., 2012] Moorrees, J., Lawson, J. M., and Marcu, L. G. (2012). Assessment of i-125 seed implant accuracy when using the live-planning technique for low dose rate prostate brachytherapy. *Radiation Oncology*, 7(1):196.

[Nag et al., 1999] Nag, S., Beyer, D., Friedland, J., Grimm, P., and Nath, R. (1999). American brachytherapy society (abs) recommendations for transperineal permanent brachytherapy of prostate cancer. *International Journal of Radiation Oncology* Biology* Physics*, 44(4):789–799.

[Nath et al., 2009] Nath, R., Bice, W. S., Butler, W. M., Chen, Z., Meigooni, A. S., Narayana, V., Rivard, M. J., and Yu, Y. (2009). Aapm recommendations on dose prescription and reporting methods for permanent interstitial brachytherapy for prostate cancer: Report of task group 137. *Medical physics*, 36(11):5310–5322.

[Pinkawa et al., 2009] Pinkawa, M., Asadpour, B., Piroth, M. D., Gagel, B., Klotz, J., Fischedick, K., Borchers, H., Jakse, G., and Eble, M. J. (2009). Rectal dosimetry following prostate brachytherapy with stranded seeds–comparison of transrectal ultrasound intra-operative planning (day 0) and computed tomography-postplanning (day 1 vs. day 30) with special focus on sources placed close to the rectal wall. *Radiotherapy and Oncology*, 91(2):207–212.

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Steggerda et al., 2007] Steggerda, M. J., Moonen, L. M., van der Poel, H. G., and Schneider, C. J. (2007). The influence of geometrical changes on the dose distribution after i-125 seed implantation of the prostate. *Radiotherapy and oncology*, 83(1):11–17.

 ISBN 978-0-9934207-0-2

Resolution enhancement of thermal imaging

Colm Lynch[1,2], Nicholas Devaney[1], Alexandru Drimbarean[2]

1. Applied Optics Group, School of Physics,
National University of Ireland, Galway
2. Fotonation Ireland, Parkmore East Industrial Estate,
Galway, Ireland

Abstract

A method of enhancing the output of low resolution thermal sensors is investigated. Current thermal imaging arrays are limited in pixel count with a large pixel size, resulting in output images of relatively poor quality when compared with consumer camera modules in the visible spectrum. Due to physical limitations this is likely to remain true for several years to come. Utilising small shifts in the thermal channel, an enhanced output can be obtained that allows for a resizing of input images with no apparent loss in detail. Simulation and theory are presented, together with some initial results.

Keywords: Super-resolution, image-processing, thermal imaging

1 Introduction

Interest has recently been growing in thermal imaging for consumer markets, with the advent of smartphone-based thermal cameras. The current form factor of these cameras is limited and is likely to remain so due to physical and manufacturing constraints. Although high resolution thermal cameras are available, the sensor and system size greatly exceeds that available to smartphones. To this end a method of improving the output of thermal cameras, while retaining the same form and size is sought. Current consumer thermal camera resolutions are of the order of 200×150[1] pixels, which falls greatly below that of consumer grade imaging sensors in the visible spectrum. A more typical pixel count is 80×60[2] as is common with many thermal camera sensors. When imaging in the thermal spectrum, different materials are required. For standard uncooled thermal imaging, Vanadium Oxide is used as detector material and the resistivity change with incident thermal radiation is used to form an image. With a longer wavelength, the diffraction limited spot size increases, leading to a necessary increase in pixel size. Glass is opaque across thermal wavelengths and so metal and polymer lenses are used to focus light. Regardless of these differing materials, the underlying principles remain the same regarding image formation and the same treatment can be applied as with visible spectrum imaging [Dereniak and Boreman, 1996].

2 Aliasing

The detail captured by an optical system can be decomposed into constituent spatial frequencies. Higher spatial frequencies correspond to fine detail. For a diffraction-limited system the upper limit on the spatial frequencies passed by the optics is given by $\frac{1}{F\lambda}$, where λ is the wavelength and F is the F number, or ratio of focal length to lens diameter. Imaging with a mean wavelength $\lambda = 10\,\mu m$ and $F = 1.2$ gives a limiting spatial frequency of $80\,lp\,mm^{-1}$, measured in lines per mm.

[1] SeeK Thermal: 206×156 element array, FLIR One generation 2: 180×160 element array
[2] FLIR One generation 1: 80×60 element array

 ISBN 978-0-9934207-0-2

When discretely sampling a signal, to obtain a true representation of the signal we should sample at a higher frequency than those present in the signal. The minimum sampling frequency is twice that of the highest frequency present. If we sample below this frequency, higher frequency variations cannot be recorded faithfully and are misinterpreted as lower spatial frequencies. With a lens with maximum spatial frequency of $80\,lp\,mm^{-1}$ the signal should be sampled at $160\,lp\,mm^{-1}$. A square pixel, with size δ_x, has associated sampling frequency $\frac{1}{\delta_x}$, requiring a pixel size of $6.25\,\mu m$ to fully capture the information present in the scene.

The thermal camera used in this experiment has pixel size of $\delta_x = 50\,\mu m$ and so only frequencies below $10\,lp\,mm^{-1}$ are faithfully captured. Frequencies that reach the sensor above this are folded into the lower frequency range. This is referred to as aliasing and causes image artifacts.

Although frequencies above $\frac{1}{2\delta_x}$ are aliased, the information carried is still present. By applying known shifts to the image sequences, the frequency content across each pixel will vary. With a knowledge of the aliased components and their associated shifts, the corresponding spatial frequencies can be extracted [Ur and Gross, 1992]. In the case of aliasing, observing frequencies u below the $\frac{1}{2\delta_x}$ limit results in a corrupted signal containing frequencies above u_N, the Nyquist frequency. [Figure 1] The measured frequency, $f_{meas}(u)$ contains aliases of frequencies according to $f_{meas}(u) = f(u) + f(2u_N - u) + f(2u_N + u) + f(4u_N - u) + f(4u_N - u) + \dots$. The variation of spatial frequency content per pixel is of interest and for this reason shifts should be applied at a sub-pixel level to the thermal spectrum images. An adaptation of a super-resolution model by Gilman & Bailey [Gilman et al., 2008] was used in this work.

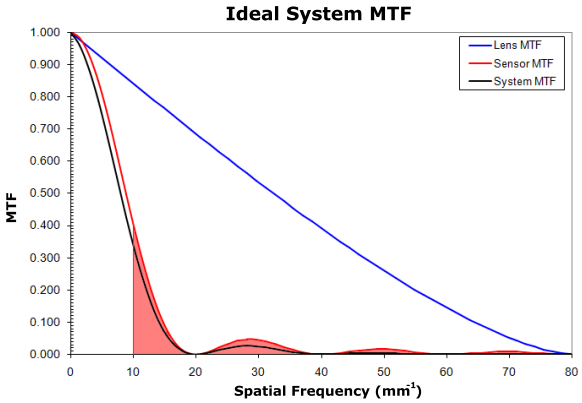


Figure 1: Theoretical modulation transfer functions (MTFs) for a thermal system with $F\# = 1.2$, $\lambda = 10\,\mu m$ and pixel size of $\delta_x = 50\,\mu m$. The lens MTF places an upper limit of 80 $lp\,mm^{-1}$ on the spatial frequencies passed. The sensor response is the typical Sinc function, where the first zero occurs at δ_x^{-1}. The system MTF is a multiplicative combination of lens and sensor MTFs. Spatial frequencies above $(2\delta_x)^{-1}$ are aliased and appear at lower spatial frequencies. Shaded regions denote spatial frequencies that reach the sensor but are misinterpreted as low spatial frequencies. It is in this shaded range that frequency recovery is possible.

3 Experimental considerations

3.1 Simulation

Implementation of the super-resolution model was first applied to a set of simulated low spatial frequency data. Utilizing knowledge of exact shifts presents an optimum scenario for which the algorithm can be tested for. By discretely binning high resolution images, a set of low resolution shifted images can be simulated. An input image from the visible spectrum was downsampled as the test input.

Pixel binning is necessary rather than interpolation, since this maintains higher spatial frequency content as aliases in the low resolution output, as would occur in real systems. Applying the super-resolution algorithm to this input set of data resulted in an output image without discrete jagged edges as visible in the interpolated version of input frames. [Figure 2] Small details that were not resolvable in input images became clearer in the super-resolution output.

3.2 Experimental setup

A joint imaging system consisting of a camera operating in the visible spectrum and a thermal imaging camera was constructed with fixed spacing. To co-register the systems, a dichroic beamsplitter was used. Any movement observed by the visible camera is subject to fixed transformations consisting of scaling and reflec-

 ISBN 978-0-9934207-0-2

Figure 2: Comparison of super-resolution results applied to simulated low spatial frequency images. (left) images obtained through Sinc interpolation of input images, (right) results of super-resolution. Sharp discontinuities across edges and periodic gradients are visible in the interpolated output. Upon application of super-resolution, more continuous detail is visible along edges. The input sequence for the super-resolution process was 23 randomly shifted images.

tion around the vertical axis in the thermal spectrum. As the visible pixels are much smaller than those in the thermal spectrum, shifts can be measured in the visible spectrum with ease and converted to sub-pixel thermal shifts. Many methods exist for image registration and sub-pixel shift estimation[Foroosh et al., 2002, Lucas et al., 1981, Mas Candela et al., 2012]. In any potential final consumer product, input from both cameras would likely be present and a simple phase correlation method was implemented in determining interframe shifts. Other local methods are to be implemented in the future.

Upon characterizing the entire system, the scaling for transformations between visible shifts and thermal shifts was measured to be 0.0769. For a shift of one pixel in the thermal spectrum, the visible channel would have undergone a shift of 13 pixels.

An autoregressive model was developed to simulate hand motion, as measured via gyroscopic data using a smartphone. These generated shifts were then applied using a Newport Fast-Steering Mirror. By imaging a target through a mirror with tip and tilt control it is possible to produce small, precise shifts, measurable by the visible channel.

To produce a scene with sufficiently high spatial frequencies, a heated wire was used. As a point spread function cannot be created with ease in the thermal spectrum, it was necessary to use a 1 dimensional integration - a line spread function. This can be repeated for any orientation. A crosshair allowed for registration in both x and y directions and for visualization of super-resolution along orthogonal axes.

Upon imaging with sub-pixel shifts, an output was obtained using the super-resolution algorithm.

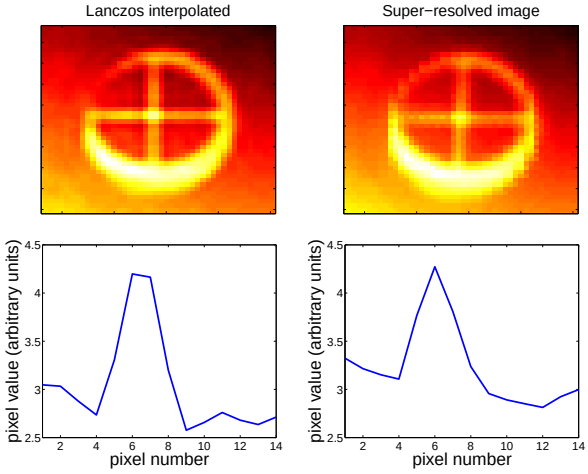


Figure 3: Results of the super-resolution technique applied to a images containing real shifts. (left) Sinc interpolation of an input frame, (right) output from super-resolution algorithm. The crosshair appears better resolved upon application of the super-resolution algorithm. A finer profile is visible and undulating features present in interpolations of input frames are not present. The annular object is the mirror mount and is not super-resolved. A total of 7 input frames were used.

 ISBN 978-0-9934207-0-2

[Figure 3] With a spatial frequency increase, an image size increase is possible with no apparent loss in detail.

3.3 Objective quality metrics

To truly determine if the output image contains sensible high spatial frequencies, a performance metric is required. The MTF itself is not sufficient as a measurement, as high frequencies in the MTF are not necessarily associated with meaningful detail. Work is ongoing by using methods such as SSIM (structural similarity), etc.

4 Further work

Work is ongoing in manipulation of temporal variations in an image sequence to extract additional information from the scene. As part of this, an improvement in spatial resolution could be advantageous. If any output from the super-resolution algorithm is to be useful in temporal processing, the number of frames needed per output super-resolved frame must be minimized so that the temporal information is left mostly unaltered. A previous paper has suggested that the minimum number of images necessary to sample at $1/K$ times the Nyquist frequency is K images [Brown, 1981]. Extending this to the two-dimensional realm results in K^2 samples for a $1/K$ sampling. A new model of super-resolution is being developed to reduce the disparity between this theory and observations to date.

5 Conclusion

It has been observed that a modest resolution increase is possible with the use of sub-pixel shifts. Sub-pixel shifts were simulated and applied to produce an output image with improved resolution.

Acknowledgments

This work was made possible with the support of the Irish Research Council and Fotonation Ireland.

References

[Brown, 1981] Brown, J. L. (1981). Multi-channel sampling of low-pass signals. *Circuits and Systems, IEEE Transactions on*, 28(2):101–106.

[Dereniak and Boreman, 1996] Dereniak, E. L. and Boreman, G. D. (1996). *Infrared detectors and systems*. Wiley New York.

[Foroosh et al., 2002] Foroosh, H., Zerubia, J. B., and Berthod, M. (2002). Extension of phase correlation to subpixel registration. *Image Processing, IEEE Transactions on*, 11(3):188–200.

[Gilman et al., 2008] Gilman, A., Bailey, D. G., and Marsland, S. R. (2008). Interpolation models for image super-resolution. In *Electronic Design, Test and Applications. DELTA. Fourth IEEE International Symposium on*, pages 55–60.

[Lucas et al., 1981] Lucas, B. D., Kanade, T., et al. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2*, volume 81, pages 674–679.

[Mas Candela et al., 2012] Mas Candela, D., Ferrer Crespo, B., Sheridan, J. T., Espinosa Tomás, J., et al. (2012). Resolution limits to object tracking with subpixel accuracy. *Optics letters*, 37(23):4877–4879.

[Ur and Gross, 1992] Ur, H. and Gross, D. (1992). Improved resolution from subpixel shifted pictures. *CVGIP: Graphical Models and Image Processing*, 54(2):181–186.

Interpolating eigenvectors from second-stage PCA to find the pose angle in handshape recognition.

M. Oliveira & A. Sutherland

School of Computing
Dublin City University, Ireland

Abstract

In this paper we present a new technique to interpolate eivenspaces for handshape recognition. We use a two stage Principal Component Analysis (PCA) to reduce the dimensionality and extract features and then interpolate. We propose to apply PCA over the manifolds generated by a first stage PCA, creating a second stage PCA. From these new manifolds, we use splines to create new artificial eigenspaces in between different pose angles. Some results are presented such as accuracy of recognition of the correct pose angle gap and a plot of the original and artificial projection of the eigenspaces.

Keywords: PCA, Handshapes Recognition, Splines.

1 Introduction

One very important application of Computer Vision (CV) is Human Computer Interaction (HCI). HCI can be used to let humans interact with a computer or a mobile device using cameras. Handshape recognition is a natural interaction for that propose. However, it is not an easy task since the hand is a deformable object [Binh et al., 2005].

There are different techniques to extract features of images. Appearance-based methods is one of them. Usually, these methods have the advantage of real-time performance. In addition, we can find different techniques to build classifiers in this category [Wu et al., 2001].

According to [Han and Liu, 2014] Principal Component Analysis (PCA) is a very useful technique for dimensionality reduction and feature selection. It is possible to apply PCA in numerous fields, such as face and handshape recognition.

The use of PCA with more than one stage was successfully applied in [Chan et al., 2014], for face recognition. For that reason we have decided to apply PCA for handshape recognition, over the manifolds extracted from a first stage PCA. At that point, we have a second stage PCA where we could interpolate, using splines, some points and recreate any shape in between pose angles. We used the second stage PCA for reduce even more the dimensionality and make it possible to interpolate in few dimensions.

2 Dataset

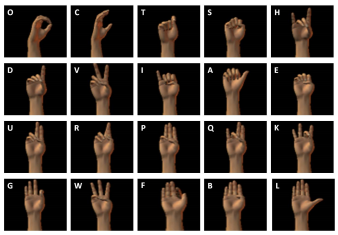
The first step was to compute PCA over a range of raw images. The dataset used was created by Farouk [Farouk et al., 2013], which contains 20 handshapes, Figure 1a. Each image was translated 5 pixels horizontally and vertically, creating a total of 121 images for each handshape. In total we have 2420 images for each pose angle.

These images were converted to grey scale and blurred using a Gaussian blurring function. It helps to reduce the non-linearity in the manifolds within the eigenspaces. It is more efficient to analyse a flat manifold

 ISBN 978-0-9934207-0-2

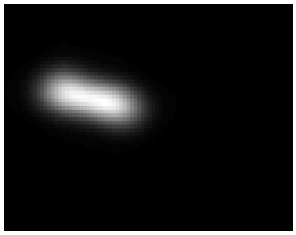
than a curved one, because curves tend to overlap. In this stage a 10 pixel radius of blurring was applied, though the blurred image (Figure 1c) shows much less detail it is still good for recognition, since PCA uses every pixel for computation.

Each image in Farouk's dataset [Farouk et al., 2013] has 640×480 pixels. In this paper we reduce each of these images to 12.5% of the original size turning into 80×60 pixels, in order to manipulate these images in a personal computer. In Figure 1a is possible to view the Irish Sign Language alphabet, Figure 1b shows the image corresponding to the letter *A* in this alphabet and Figure 1c shows the same shape after being converted to grey scale and blurred.



(a) Original images from Farouk's dataset

(b) "A" Before blurring



(c) "A" After convert to grey scale and blurring

Figure 1: Irish sign language alphabet, followed by handshape of letter *A* and the same image blurred.

2.1 Training Dataset

The training dataset consists of the same 20 handshapes from Farouk [Farouk et al., 2013]. However, for training data they were rotated from 0 to -90 degrees, at intervals of 10 degrees, via Matlab, from the original position.

2.2 Testing Dataset

In the testing dataset we obtained the same 20 images and rotated them from -5 degrees to -85 degrees in intervals of 10 degrees. The main idea of this testing dataset is that each angle is the midpoint of a consecutive pair of angles in the training dataset.

3 Principal Component Analysis

The first step was to combine images from the training dataset into the same array and then compute PCA. Since each image has 60×80 pixels when vectorized it turned into 4800 pixels in a row, for each image. As a result, we have a 4800×4800 covariance matrix. By applying PCA to the covariance matrix we obtained 4800 eigenvectors.

Multiplying the 10 first (most significant according to the eigenvalues) eigenvectors by each set of images we have one eigenspace with 10 dimensions.

3.1 Perpendicular distance

In order to find the distance between a new unknown image and a space, we used perpendicular distance. Which is the distance of a point to an eigenspace measured along a line perpendicular to the eigenspace.

$$dis = \sqrt{\sum (p-o)^2 - \sum (p*v - o*v)^2}$$

Where o is the origin (mean of all projected images at the same angle), v is the eigenspace and p is the new image. This equation returns the distance between a new image and a space.

 ISBN 978-0-9934207-0-2

3.2 Second Stage PCA

In order to interpolate eigenspaces it is essential to have just a few dimensions and elements in each eigenspace. That is why we computed PCA again for each angle of the dataset separately. In other words, a new PCA is computed for projected images for each pose angle from 0º to -90º at intervals of 10 degrees. As a result we have 10 new eigenspaces. Algorithm 1 shows how this second stage PCA is computed.

Data: projection of each pose angle in the first stage PCA
Result: new eigenspace of the second stage PCA
initialization;
for *each pose angle projection* **do**
  compute PCA;
end

Algorithm 1: Pseudo-code for the second stage PCA

3.3 Interpolating by splines

Once we have these 10 new eigenspaces, it is possible to interpolate new spaces in between. These spaces consist of 10 eigenvectors each with 10 dimensions. Then we took the most significant eigenvectors (according to the eigenvalues) and interpolate all elements across the 10 different spaces. We used splines to interpolate a curve in a space of 10 dimensions (each dimension represents an element of the eigenvector). In our case we considered only 4 eigenvectors. Figure 2 shows the interpolation of each one of these 4 eigenvectors.

Having some spaces interpolated between a pair of angles it is possible to determine any point in between. As we have 10 spaces and each one is 10 degrees apart we interpolated 90 points in between all 10 spaces, then that each new space corresponds to one angle from 0 to 90 degrees.

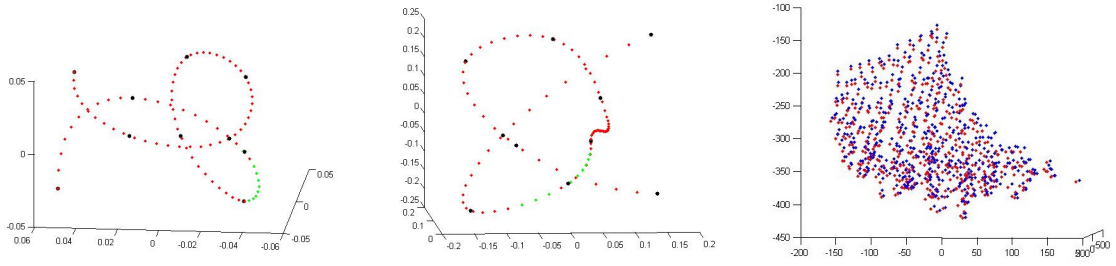
3.4 Methodology and Results

In the second stage PCA we obtained 10 new eigenspaces, from these spaces it is possible to measure the perpendicular distance of a new unknown image from a space. Table 1 shows the accuracy as a percentage of the recognition of unknown images in an unknown pose angle against the ten spaces. Each set of these unknown images are at an intermediate pose angle, then we know the neighbour angles in advance. This accuracy was measured after computing perpendicular distance of each image against all 10 eigenspaces and taking the shortest one. The percentage is out of 2420 images, e.g. 95.61% means 2313 images out of 2420 were classified correctly.

Table 1: Accuracy of recognizing correct gaps in the first stage PCA

Angle	Accuracy %	Angle	Accuracy %	Angle	Accuracy %
5	95.61	35	86.44	65	82.31
15	75.61	45	88.80	75	76.15
25	84.79	55	82.68	85	93.51

In Figure 2 we can view the quality of the interpolation between spaces. In Figures 2a and 2b each axis represents one element, in total we have 10 elements. However, here we are plotting only 3 of them, because plots are 3 dimensional. Each plot represents one eigenvector. The green dots are projections of the 3 elements in the artificial eigenspace for the 10 interpolated points.



(a) The first eigenvector (b) The second eigenvector (c) Projection of images in an artificial (red) and real eigenspace (blue).

Figure 2: Interpolated eigenvectors, where black are the original points, red are the interpolated and the green are the 10 first interpolated in Figures (a) and (b). Figure(c) shows a projection of 242 images in an artificial eigenspace in red, and the same projection into a real one in blue.

Figure 2c shows a projection of a set of 242 images in pose angle -25 into a 3 dimensional space. Blue points represent a projection of these images in a real eigenspace and the red points represent the same projection into an artificial eigenspace, both for the same pose angle. It is clear that points are close to each other.

4 Conclusion

In this work PCA was used as a feature extraction and dimensionality reduction method. Gaussian blur was used to reduce the non-linearity of the manifolds in the PCA spaces. An incoming object can be classified by computing the perpendicular distance between this point (new image) and a space. Splines were used to interpolate between eigenspaces creating artificial new ones. In summary, interpolated eigenvectors might be useful to create artificial new eigenspaces from a dimensionality-reduced space.

As future work we intend to try different numbers of eigenvectors to observe how these numbers influence the results. We intend to find a classifier for images in the interpolated space. We will try a new technique to interpolate eigenvectors in any dimension. Finally, we will test different levels of blurring.

Acknowledgements

This research was funded by CAPES/Science without Borders - Brazilian Program - Process: 9064-13-3.

References

[Binh et al., 2005] Binh, N. D., Shuichi, E., and Ejima, T. (2005). Real-time hand tracking and gesture recognition system. In *Proceedings of International Conference on Graphics, Vision and Image Processing (GVIP-05*, pages 362–368.

[Chan et al., 2014] Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., and Ma, Y. (2014). PCANet: A simple deep learning baseline for image classification? *CoRR*, abs/1404.3606.

[Farouk et al., 2013] Farouk, M., Sutherland, A., and Shokry, A. (2013). Nonlinearity reduction of manifolds using Gaussian blur for handshape recognition based on multi-dimensional grids. In Marsico, M. D. and Fred, A. L. N., editors, *ICPRAM*, pages 303–307. SciTePress.

[Han and Liu, 2014] Han, F. and Liu, H. (2014). Scale-invariant sparse PCA on high-dimensional meta-elliptical data. *Journal of the American Statistical Association*, 109(505):275–287.

[Wu et al., 2001] Wu, Y., Lin, J. Y., and Huang, T. S. (2001). Capturing natural hand articulation. In *In ICCV*, pages 426–432.

Range Image Feature Extraction using a Hexagonal Pixel-based Framework

Bryan Gardiner
Ulster University
Northland Road, Londonderry, BT48 7JL
b.gardiner@ulster.ac.uk

Sonya Coleman
Ulster University
Northland Road, Londonderry, BT48 7JL
Sa.coleman@ulster.ac.uk

Abstract

Research to date within the area of range image processing has highlighted the many advantages of using range data to represent scenes in a 3-D manner. However, given the large volume of data associated with range images, processing and extracting relevant information from these images has presented a challenge. Due to the irregular distribution of data in many range image types, a resampling process is required to map the irregular data to a regular grid structure, providing an opportunity to resample directly to a uniform hexagonal framework. This paper therefore proposes a novel framework for range image feature extraction, which utilises the efficiency and accuracy of the hexagonal pixel-based framework.

Keywords: Hexagonal Framework, Range Images, Feature Extraction.

1 Introduction

Due to the recent development of low-cost RGB-D sensors such as the Microsoft Kinect, the use of range images have become prominent in computer vision techniques, providing an approach to obtain reliable descriptions of 3-D scenes. A range image may be described as a 2-D image that contains distance measurements from a selected reference point or plane to surface points of objects within a scene [16]. Range images provide additional information over conventional intensity images allowing more information about the scene to be captured [3]. It should be noted a range image contains information about only the visible surfaces of the objects, and not their hidden surfaces, and hence is often referred to as 2 ½ -D information [16].

Range cameras can operate according to a number of different techniques, for example, laser scanning [4], stereovision [15], pattern projection [18] and time-of-flight lasers [12]. All of these range finders have their advantages and disadvantages depending on their application, where some example application areas include object recognition [13], surface reconstruction [17], surveillance [5], robot navigation [19], etc. Many of these application areas are dependent on the completion of feature extraction on range images, and the process of acquiring these features is significantly different from that when using conventional intensity images. There are two factors to consider when processing range images: the irregularity of the distribution of the range data [2], and the types of features that are acquired from range data feature extraction. Depending on the hardware sensor used to capture the range data, the acquired data may be distributed regularly or irregularly; this should be considered before processing the image. Unlike intensity images, where edges can be found by significant changes in grey level values and hence may be modeled as ramps or steps, range data extracts edges in three different categories: step edges, crease edges and smooth edges.

A recent concept for image representation is the use of hexagonal pixels, introducing the area of hexagonal image processing. As well as the factor of hexagonally structured images mimicking the

 ISBN 978-0-9934207-0-2

structure found in the human fovea (a small region within the retina, consisting of a high density of cones shaped and placed in a hexagonal arrangement [9]), hexagonal grids have other advantages over the conventional rectangular grid. Equidistance of all pixel neighbours facilitates the implementation of circular symmetric kernels that is associated with an increase in accuracy when detecting edges, both straight and curved [1], and the improved accuracy of circular and near circular image processing operators has been demonstrated in [6]. Additionally, better spatial sampling efficiency is achieved by the hexagonal structure, leading to improved computational performance. In a hexagonal grid with unit separation of pixel centres, approximately 13% fewer pixels are required to represent the same image resolution as required on a rectangular grid with unit horizontal and vertical separation of pixel centres [14].

In previous work, the authors have shown how the hexagonal framework can be used to improve both efficiency and accuracy with respect to feature extraction on conventional intensity images [8]. This paper progresses this work by developing an approach for range data feature extraction that utilises the advantages acquired from using the hexagonal pixel-based framework. In Section 2 the range image representation using a hexagonal structure is outlined, and Section 3 details the process of applying hexagonal feature extraction operators to range images and determining the features using this approach. Section 4 presents resultant features maps and a conclusion is presented in Section 5.

2 Range Image Representation

If we consider a range image to be represented as a spatially irregular sample of values of a continuous function $u(x, y)$ of depth value, an additional process is required to map the range data to a regular rectangular grid for processing. As resampling is taking place in this process, it provides an opportunity to resample directly onto a regular hexagonal pixel-based structure, utilising the advantages obtained from processing in a hexagonal framework. For regularly distributed range images, for example those captured using an RGB-D sensor, the hexagonal resampling can be completed using the approach of [20] where hexagonal pixels are created through clusters of sub-pixels. We have modified this technique slightly by representing each pixel by a nxn block of sub-pixels. Each sub-pixel inherits the original pixel intensity, as in [14], in order to create an effect that enables sub-pixel clustering; this limits the loss of image resolution. With this re-sampled hexagonal image, it is possible to represent the range image by using an array of samples of a continuous function $u(x, y)$ of range data on a domain Ω. Figure 1 (a) represents an image composed of hexagonal pixels with nodes placed in the centre of each pixel, overlaid by the triangular finite element mesh. These nodes are the reference points for the computation of finite element techniques throughout the domain Ω.

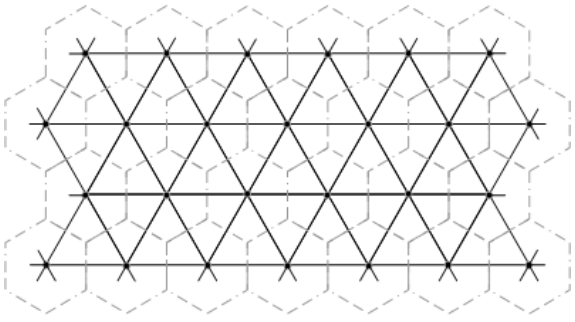


Figure 1 (a). Hexagonal array of pixels and overlying mesh

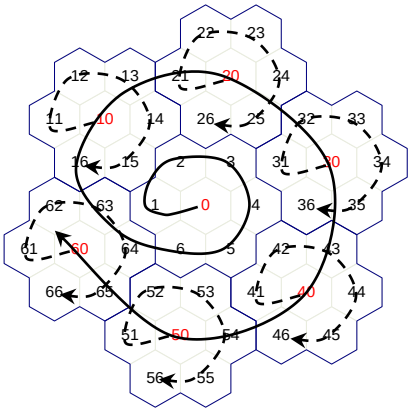


Figure 1 (b). 1D spiral addressing scheme

As discussed in previous work [8], an efficient approach to addressing hexagonal pixel-based images is via the use of the Spiral Architecture. The addressing of the Spiral Architecture originates at the centre of the hexagonal image and spirals out using one dimensional indexing. This structure facilitates the use of base seven numbering to address each pixel within the image, which permits the grouping of pixels in clusters for efficient pixel access; these clusters are shown in Figure 1 (b).

 ISBN 978-0-9934207-0-2

3 Feature Extraction Approach

As it is not possible to apply conventional square operators to images represented on a hexagonal grid structure, we have shown in recent work [11] how a finite element based approach can be used to create a hexagonally structured Linear-Gaussian operator (H_1) based on the construction of two independent directional derivative operators aligned in the *x*- and *y*- directions. The resultant kernels derived from [11] are presented in Figure 2 and will be used for feature extraction on hexagonal range images.

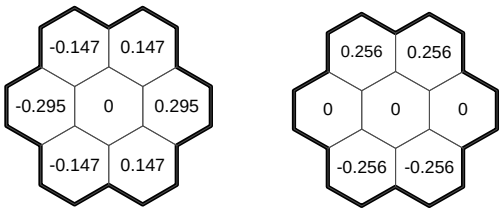


Figure 2. *x*- and *y*-components of operator H_1

Applying the presented edge detection operator to an intensity image will identify edges at the maxima of the gradient output, however the same thresholding does not identify edges in a depth image but instead will identify surfaces [7]. Determining features in hexagonal range images requires that the change point of the surfaces, i.e. slope direction, is identified. This can be achieved by using the first order operator response to identify the different surfaces in the depth image and from this, detecting significant change in gradient output can be determined as an edge point. If we consider the gradient magnitude response at any point in the hexagonal range image as $\underline{\nabla}(P_x)$, where *x* denotes the pixel location using the 1-D addressing scheme presented in Section 2, and using a threshold value of *T*, we can characterize each of the roof edges in the range image using the following set of rules:

For $(P_R) \in (P_{(x+_s 4)}), (P_{(x+_s 5)}), (P_{(x+_s 6)})$

If $\left|\underline{\nabla}(P_x) - \underline{\nabla}(P_R)\right| > T$ then $\underline{\nabla}(P_x)$ is an edge point.

(P_R) is obtained by determining the location of neighbouring pixels in tri-directions from the pixel (P_x) as demonstrated in Figure 3. Spiral addition $(+_s)$ is used to obtain pixel neighbours in a spiral framework as previously described in [10]. Utilising this set of pixel values will ensure adequate comparison of neighbouring edge points when thresholding the obtained feature map.

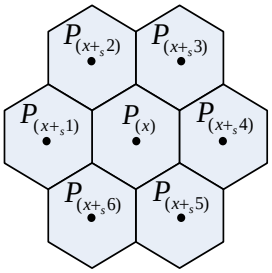


Figure 3. Neighbouring pixel locations for determining features in hexagonal range image

4 Resultant Feature Maps

To demonstrate the framework described in Section 3, the Linear-Gaussian operator was applied to various depth images that have been resampled to a hexagonal pixel-based image structure. Using the proposed thresholding technique, specifically developed for hexagonal images, edge features were identified within the depth images. An example of an original intensity image, the associated depth image and the resultant feature map acquired from the proposed framework is presented in Figure 4 (a), (b) and (c) respectively. Visual results show merit in using depth information for feature extraction, for example, successful identification of edges of two occluding objects, i.e. the bowl and hat, have been completed using the proposed range image approach, see Figure 4 (c).

5 Conclusion

This paper presents a novel approach to range image feature extraction that utilises the advantages obtained from hexagonal pixel-based images. As resampling is often required to overcome the irregular distribution of range image data, resampling to a hexagonal framework does not require any additional steps, however as a processing framework it offers additional efficiency and accuracy when compared with the conventional rectangular framework. Visual results have shown the benefit of this approach and if used in conjunction with feature extraction on intensity images should provide further accuracy of detecting edge features in 3-D scenes.

(a) Original intensity image (b) Hexagonal depth image (c) Hexagonal feature map

Figure 4. Hexagonal pixel-based range images

6 References

[1] Allen, J. D., “Filter Banks for Images on Hexagonal Grid”, Signal Solutions, 2003.

[2] B.K. Lee, K. Yu, M. Pyeon, “Effective reduction of horizontal error in laser scanning information by strip-wise least squares adjustments”, ETRI J., 25 (2), pp. 109–120, 2003.

[3] Bellon, O., Silva, L., “New improvements on range image segmentation by edge detection techniques”, Proc. Workshop on Artificial Intelligence and Computer Vision, 2000.

[4] Brenner, Claus. "Building reconstruction from images and laser scanning." International Journal of Applied Earth Observation and Geoinformation 6.3 (2005): 187-198.

[5] Choi, Wongun, Caroline Pantofaru, and Silvio Savarese. "Detecting and tracking people using an rgb-d camera via multiple detector fusion." Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE, 2011.

[6] Coleman, S.A., Scotney, B.W., Herron, M.G., “A Systematic Design Procedure for Scalable Near-Circular Laplacian of Gaussian Operators”, 17th International Conference on Pattern Recognition ICPR’04, Vol. 1, pp. 700 – 703, 2004.

[7] Coleman, S.A., Suganthan S., and Scotney, B.W., "Gradient operators for feature extraction and characterisation in range images." Pattern Recognition Letters 31.9, pp. 1028-1040, 2010.

[8] Coleman, Sonya, Bryan Gardiner, and Bryan Scotney. "Adaptive tri-direction edge detection operators based on the spiral architecture." IEEE International Conference on Image Processing (ICIP), 2010.

[9] Curcio, C.A., et al., “Human Photoreceptor Topography”, Journal of Comparative Neurology, Vol. 292, pp. 497-523, 1990.

[10] Gardiner, B., “Hexagonal Image Processing,” Ph.D. dissertation, Ulster University, 2010.

[11] Gardiner, Bryan, Sonya Coleman, and Bryan Scotney. "A design procedure for gradient operators on hexagonal images." International Machine Vision and Image Processing Conference, IMVIP 2007.

[12] Ghobadi, Seyed Eghbal, et al. "Detection and classification of moving objects-stereo or time-of-flight images." Computational Intelligence and Security, 2006 International Conference on. Vol. 1. IEEE, 2006.

[13] L. Bo, X. Ren, D. Fox, “Unsupervised Feature Learning for RGB-D Based Object Recognition”, In International Symposium on Experimental Robotics, (ISER), June 2012.

[14] Middleton L., Sivaswamy J., Hexagonal Image Processing; A Practical Approach, Springer, 2005.

[15] Murray, Don, and James J. Little. "Using real-time stereo vision for mobile robot navigation." Autonomous Robots 8.2 (2000): 161-171.

[16] P.J. Besl, “Active, optical range imaging sensors”, Machine Vision Applications, Vol. 1, pp. 127–152, 1988.

[17] R.Newcombe, S.Izadi,O.Hilliges,D.Molyneaux,D.Kim,A.Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking.” IEEE International Symposium on Mixed and Augmented Reality, 2011.

[18] Salvi, Joaquim, Joan Batlle, and E. Mouaddib. "A robust-coded pattern projection for dynamic 3D scene measurement." Pattern Recognition Letters 19.11 (1998): 1055-1065.

[19] Sturm, Jürgen, et al. "A benchmark for the evaluation of RGB-D SLAM systems." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.

[20] Wuthrich, C.A., Stucki, P., “An Algorithm Comparison between Square and Hexagonal Based Grids,” CVGIP: Graphical Models and Image Processing 53, pp. 324-339,1991.

Investigation of Face Tracking Accuracy by Obscuration Filters for Privacy Protection

J. Sato & T. Akashi

Graduate School of Engineering
Iwate University, Japan

Abstract

By developing many applications such as social networking service, privacy protection of images and videos becomes one of the important tasks to manage data. In order to obscure personal information such as a face, there are some research using filters. Although they can obscure the information, the data utility is basically lost. This is a big problem because the lost data cannot be used for any vision applications such as a surveillance system. Therefore, we investigate the obscuration filters, which are blur, pixelation, and shuffle and they are commonly used for the research of the privacy protection, based on face tracking since it is applied to many applications. From experimental results, the detection accuracy with shuffle is higher.

Keywords: Face Tracking, Privacy Protection, Obscuration Filter

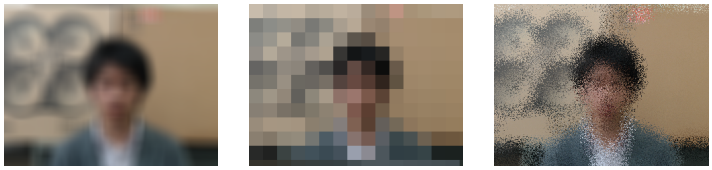
1 Introduction

Since captured images and videos can extract some personal information, such as faces and expressions, they sometimes give a stress to people. In the computer vision, there are some research about obscuration methods to protect the privacy [Upmanyu et al., 2009, Zhang et al., 2005]. Filtering the image is one of the easy technique and commonly used in a news program. For example, blurring and pixelating methods are famous. However, since they degrade the utility of the whole image, the degraded images cannot be applied to any vision applications. Therefore, it is necessary to investigate a practical filter. In this research, we evaluate the filters based on face tracking because it is applied to many applications. Since the obscured image does not have local features, such as corners and edges, the tracking is difficult task. Also, multi-view face must be considered. In order to address these problems, color histogram template matching with genetic algorithm (GA) [Oikawa et al., 2015] is used. Also, another obscuration filter, which is shuffle and proposed in [Oikawa et al., 2015], is investigated.

2 Obscuration Filter for Privacy Protection

Because it is difficult to define an infringement of privacy, there is no definitive obscuration filter. In this research, the commonly used obscuration filters and shuffling [Oikawa et al., 2015] are focused. Blurring is a smoothing algorithm and used for denoising. A centered pixel value in a filter is calculated by averaging all pixel values in the filter. Figure 1(a) shows an example of the blurring. This algorithm is easy, however there is a disadvantage that edges are lost. Pixelating is an algorithm, which produces mosaic effect. After a target image is segmented into rectangles with an arbitrary size, average pixel values in each rectangle are calculated. The averaged values are used in each rectangle as new pixel values. Figure 1(b) shows an example of the pixelating. An obscuration filter of shuffling pixels is proposed by [Oikawa et al., 2015]. Because a centered pixel in a filter is changed by other pixel, which is selected randomly in the filter, the color information is not lost significantly. Also, since when the large size filter is used, spatial information is lost. Hence, it can protect the privacy. Figure 1(c) shows an example of the shuffling pixels.

 ISBN 978-0-9934207-0-2



(a) (b) (c)

Figure 1: Example of results of obscuration filters: a) blurring; b) pixelating; c) shuffling.

3 Face Tracking by Color Histogram Template Matching with GA

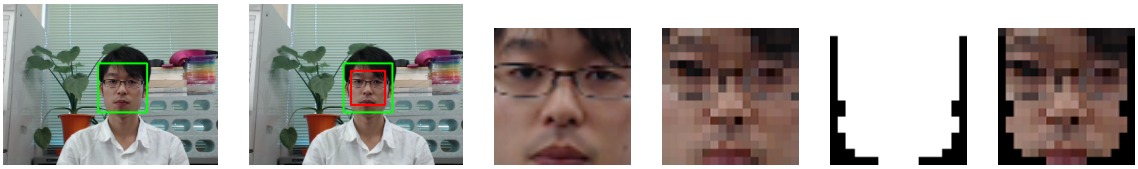
Face tracking on privacy protected images is difficult since some local features are lost. Also, if a practical use is considered, there are many real problems, such as multi-view face (appearance change), noise, and illumination change. Oikawa et al. [Oikawa et al., 2015] address these problems using color histogram template matching with genetic algorithm (GA). From the next section, the method is explained.

3.1 Creation of Color Histogram Template

Firstly, a frontal face, which is tracked in privacy protected images, is detected on an original image using a face detector [Lienhart and Maydt, 2002] (Figure 2(a)). Then, the 70% of the detected region is extracted to remove background (Figure 2(b) and (c)). After that, the extracted region is resized to 1/9 (Figure 2(d)). In order to remove the background near the chinbone, a half ellipse mask is created (Figure 2(e)). This mask is created based on the size of the resized image. By masking the extracted region, the background is removed (Figure 2(f)). Next, the resized image is converted to YCrCb color space, and color histograms of Cr and Cb components are created. In addition, Cr and Cb color histograms from a forehead are created for higher tracking accuracy. The forehead region is determined using the 70% region of the face. Let w and h be width and height of the 70% region. x and y are the upper left vertex. The width and height of the forehead region are w and $1/3h$. The upper left vertex is $(x, y-2/5h)$. Figure 3(a) shows the forehead region. Figure 3(b) and (c) represent the extracted image and resized image to 1/9. By using four color histogram templates, the face can be tracked.

3.2 Template Matching with GA

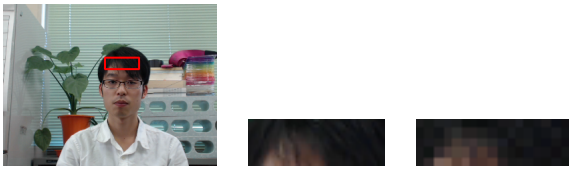
Most object detection methods use sliding window. However, scanning the whole image in every frame is inefficient. For fast processing, GA is applied to template matching [Oikawa et al., 2015]. Firstly, N individuals are generated randomly. Each individual has parameters of a parallel translation (x, y), scale factors of x and y axis, and an angle of rotation θ as binary. After each individual is decoded to real numbers, candidate regions are generated using the rectangles, which are resized to 1/9 from the 70% of the face detection results and a forehead, and the real numbers. Next, four color histograms, which are Cr and Cb components, are



(a) (b) (c) (d) (e) (f)

Figure 2: Creation of color histogram template from a face: a) frontal face detection; b) extraction of 70%; c) extraction result; d) resized to 1/9; e) half ellipse mask; f) masking result.

 ISBN 978-0-9934207-0-2



(a) (b) (c)

Figure 3: Creation of color histogram template from a forehead: a) decision of the forehead region; b) extraction of the region; c) resized to 1/9.

created. Then, squared euclidean distance between these histograms and template are calculated for objective values. After that, fitness values are calculated by normalizing the objective values to [0.0,1.0]. Based on the fitness value, genetic operations, such as selection, crossover, and mutation are performed. This procedure is iterated multiple times (generation iteration). When a termination condition is satisfied, the generation iteration is stopped, and an elite individual, whose fitness value is the highest in the final generation, is acquired as a detection result.

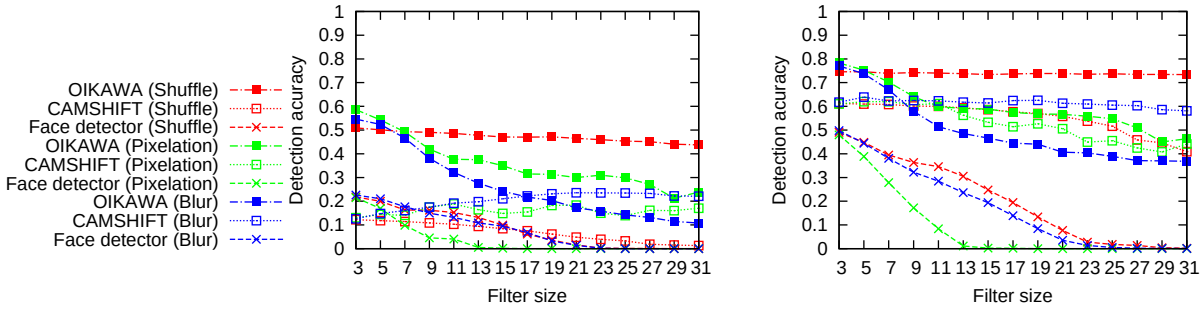
4 Experiment

Used video sets were the same to [Oikawa et al., 2015]. They were 15 video sequences and five subjects appeared with three different backgrounds. The only one subject appeared in each video with the great head motion. The image size was 320×240 pixels. In order to judge results as success or failure, bounding box evaluation, which evaluates results based on an overlap ratio (PASCAL threshold) between a system output and a ground truth, was used [Everingham et al., 2010]. A threshold of the overlap ratio is generally 0.5. However, this value is a little tight in this experiment because the local features are lost by the obscuration filters. Therefore, 0.4 and 0.3 were used. As comparative methods, CAMSHIFT [Bradski and Clara, 1998] and haar-like face detector [Lienhart and Maydt, 2002] were used. This is because that CAMSHIFT uses color information and is robust to loss of the local features, and the face detector is a popular method. Used computer had CPU of Intel Core (TM) i7-3770S (3.10 GHz) and RAM of 16.0 GB.

5 Result, Consideration, and Conclusion

Table 4(a) and (b) show detection accuracy (DA) with different PASCAL thresholds. Although filter size changes, DA of OIKAWA's method is stable and the highest in most filter sizes. The DA of face detector is lower in all the filters. The reason is that the local features are lost. This method uses some relationships of a difference of intensity of the face and they are extracted by using enormous face images and a machine learning. However, the learned relationships cannot be used for the filtered images. The DA of OIKAWA's method is higher in most filter sizes with shuffling. Since the shuffling only changes locations of each pixel, color histogram template is not changed drastically (Figure 5(a)). In contrast, because CAMSHIFT uses a high probability skin color and main tracking is skin, the output rectangle is larger than the ground truth and it causes the low DA. Hence, in order for the high DA, using another color feature of face parts such as palpebral fissures is necessary for CAMSHIFT. Because the pixelating and blurring filters change the color information drastically compared to the shuffling (Figure 5(b) and (c)), the DA of the OIKAWA's method and the CAMSHIFT is low. From these results, the shuffling with color histogram template matching is a better method for the face tracking.

Next, the average processing time by each obscuration filter per one image is considered. The time when the filter size is 3×3 and 31×31 are used is investigated. When the size is 3×3, the times by shuffling, pixelating, and blurring are 1.2, 0.5, and 1.0 ms. When the size is 31×31, the times are 1.0, 0.3, and 1.1 ms. All the filters do not take a long time. Therefore, if we consider the face tracking accuracy, OIKAWA's method with shuffling is the best. However, if we consider a practical application, it is necessary to improve the OIKAWA's method



(a) (b)

Figure 4: Detection accuracy with different PASCAL thresholds: a) PASCAL=0.4; b) PASCAL=0.3.

(a) (b) (c)

Figure 5: Example of detection results. The size for all obscuration filters is 31×31. Red and green rectanges represent the tracking result by CAMSHIFT and OIKAWA's method and there is no detection by the face detector: a) shuffling; b) pixelating; c) blurring.

because the DA is low when the PASCAL threshold is high. This is because that this method only uses color histograms. For more stable system, additional features are required. In the future, we investigate this.

References

[Bradski and Clara, 1998] Bradski, G. R. and Clara, S. (1998). Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal Q2 '98*, pages 1–15.

[Everingham et al., 2010] Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *Int. J. Computer Vision*, 88:303–338.

[Lienhart and Maydt, 2002] Lienhart, R. and Maydt, J. (2002). An extended set of haar-like features for rapid object detection. pages I–900–I–903. ICIP.

[Oikawa et al., 2015] Oikawa, D., Sato, J., and Akashi, T. (2015). Improved face tracking with privacy protection using half ellipse and additional region matching. pages CD–ROM. IWAIT and IFMIA.

[Upmanyu et al., 2009] Upmanyu, M., Namboodiri, A. M., Srinathan, K., and Jawahar, C. V. (2009). Efficient privacy preserving video surveillance. pages 1639–1646. ICCV.

[Zhang et al., 2005] Zhang, W., ching S. Cheung, S., and Chen, M. (2005). Hiding privacy information in video surveillance system. pages II–868–II–871. ICIP.

Cone detection and blood vessel segmentation on AO retinal images

L. Mariotti & N.Devaney

Applied Optics Group, School of Physics
National University of Ireland, Galway

Abstract

With the advent of Adaptive Optics (AO) for the high-resolution imaging of the retina, it is possible to study the individual photoreceptors and their spatial distribution *in vivo*. The presence of the blood vessels affects negatively the detections, and for this reason clinicians select manually the regions devoid from the vessels to be analysed. We present here a method that we developed for the segmentation of blood vessels in AO retinal images. With the choice of a suitable cone detection algorithm, we are now able to automatically analyse the images acquired by an AO fundus camera in their entirety.

Keywords: image processing, ophthalmology, Adaptive Optics, segmentation

1 Introduction

Thanks to the advent of Adaptive Optics (AO) in vision science, it has become possible for clinicians to study the human retina *in vivo* with high-resolution images. The detection of cone photoreceptors is important, as their spatial distribution is strongly related to the quality of the subject's vision and can indicate the presence of clinical conditions (Figure 1). In previous work [Mariotti and Devaney, 2015] we implemented and compared the most commonly used cone detection algorithms.

It is common practice for clinicians to select small rectangular windows on the image in which to perform cone detection and to analyse the metrics related to cone spacing. The windows are selected in order to exclude the shadow projected by blood vessels on the photoreceptor layer, as the resulting detections would be unreliable. There are already algorithms in the literature proposed for the segmentation of retinal vessels, but they are developed for low-resolution retinal images, and did not perform well on our high-resolution images.

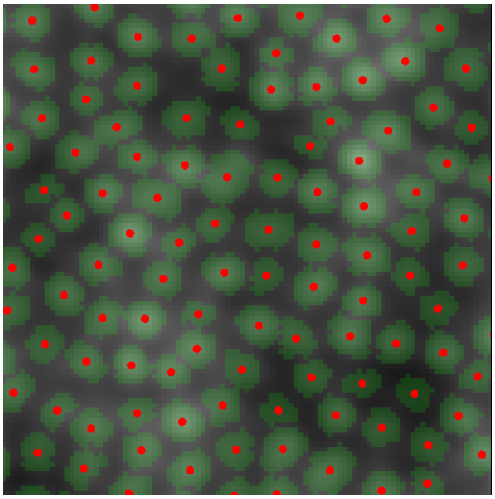


Figure 1: Example of cone detection with segmentation

2 Method

The two cone detection algorithms that we use in our analysis were developed by [Li and Roorda, 2007] and [Chiu et al., 2013]. The Li and Roorda algorithm detects the cones as the local maxima in the image. The Chiu

 ISBN 978-0-9934207-0-2

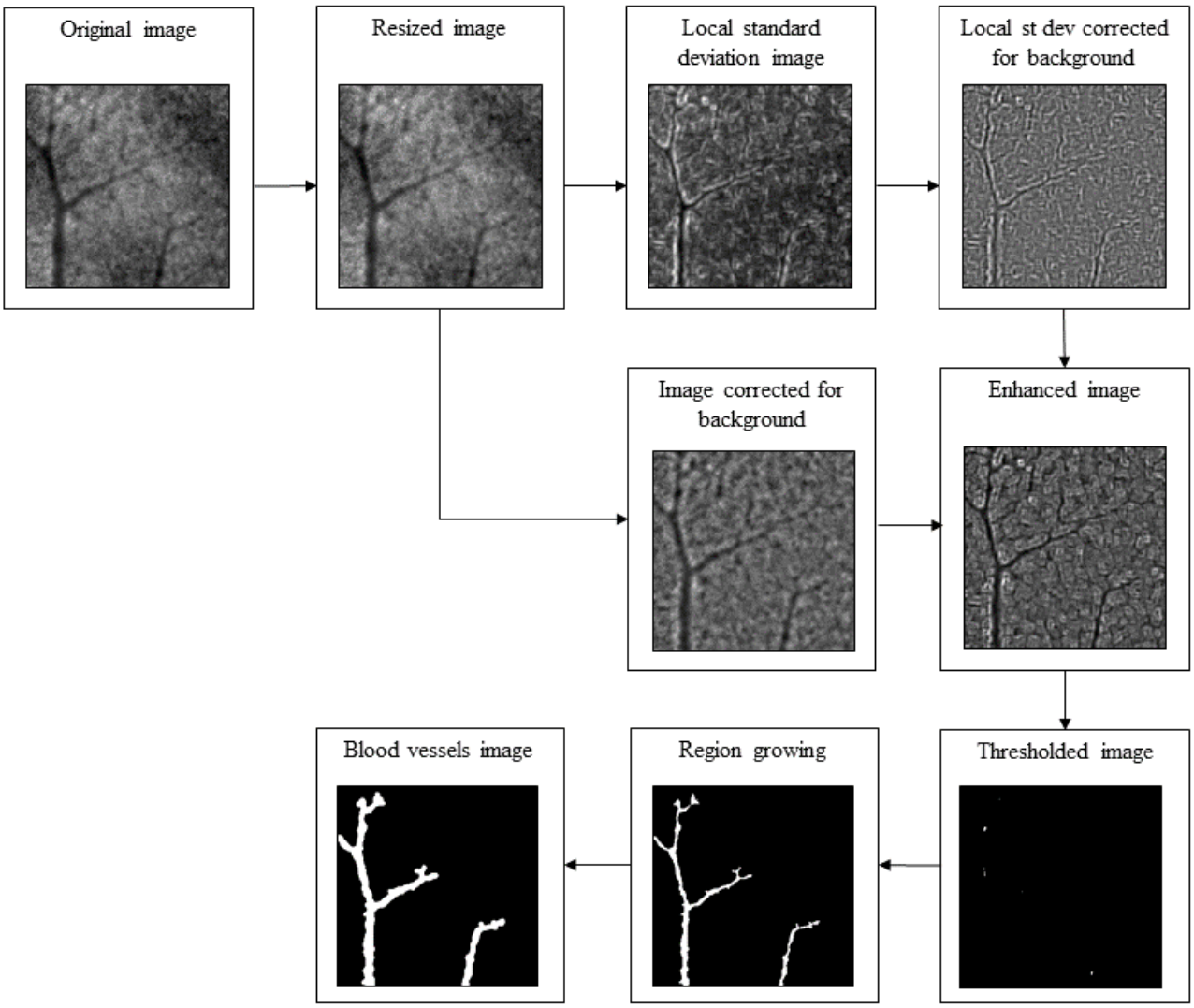


Figure 2: Diagram showing the selection of the blood vessels from the original image

et al. algorithm extends the detection by segmenting the cone profiles around the local maxima (Figure 1). A portion of the image surrounding each maximum is transformed into quasi-polar coordinates, then the contour of the cone is segmented as a layer using graph theory and dynamic programming.

In order to segment the blood vessels from the rest of the image, we examine the local standard deviation of the images, as this will be reduced where there are the vessel shadows. We downsize the images by a factor of four, in order to discard the high-frequency spatial information given by the cones. The local standard deviation of the images is calculated, and multiplied by the original image to enhance the blood vessels, after correcting both images for low-frequency background variations.

The vessel profiles were then found using region growing around the minimum intensity pixels of the enhanced image. The threshold for selecting the seed points and the parameter that regulates the extension of the region growing are chosen manually after visual inspection of the results (however, when using the same instrument it does not vary significantly between patients). A diagram of the procedure is shown in Figure 2.

 ISBN 978-0-9934207-0-2

3 Conclusions

The segmentation of the vessels allows us to perform a complete analysis on the cone mosaic, increasing the area and the number of cones that can be included in the study (Figure 3) and removing the necessity to manually select windows. This simplification will facilitate clinical use, resulting in an increased number of patients that can be examined.

We thank Marco Lombardo for acquiring the images (with *rtx1* camera of Imagine Eyes, Orsay, France) and the Irish Research Council for financial support.

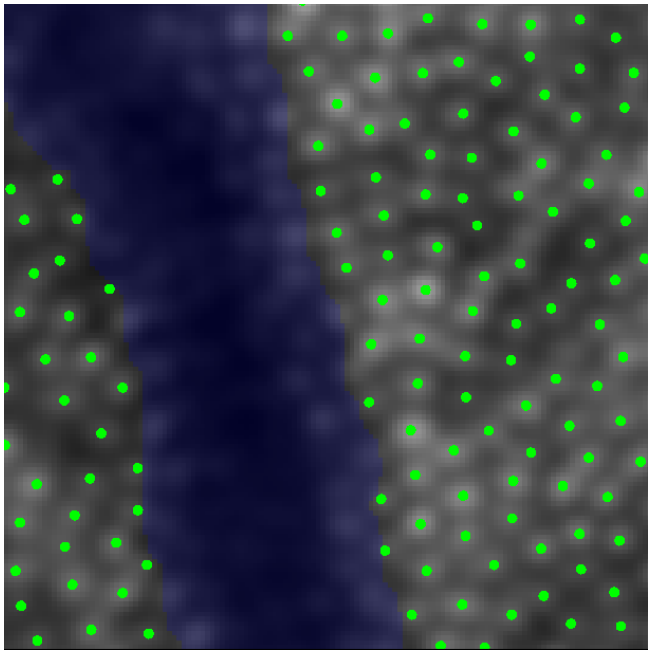


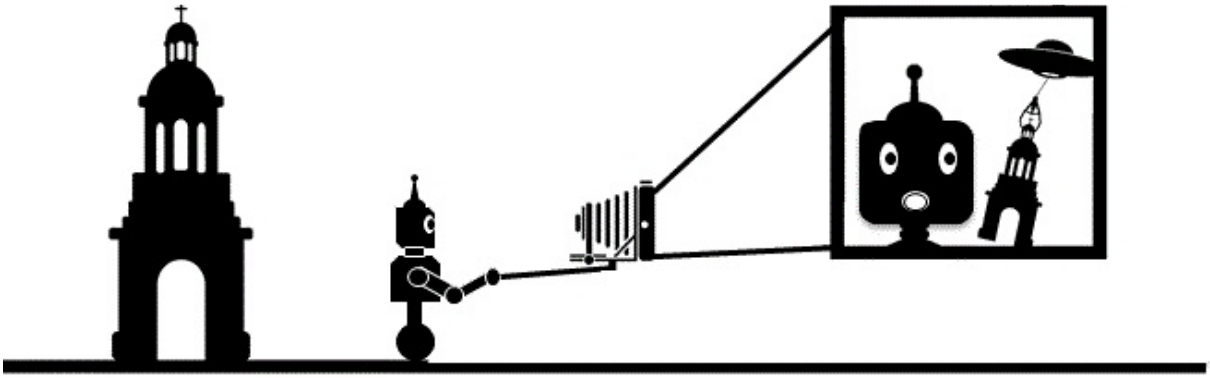
Figure 3: Detail of cone detections (green dots) with the exclusion of the blood vessel profile (blue)

References

[Chiu et al., 2013] Chiu, S. J., Lokhnygina, Y., Dubis, A. M., Dubra, A., Carroll, J., Izatt, J. A., and Farsiu, S. (2013). Automatic cone photoreceptor segmentation using graph theory and dynamic programming. *Biomed. Opt. Express*, 4(6):924–937.

[Li and Roorda, 2007] Li, K. Y. and Roorda, A. (2007). Automated identification of cone photoreceptors in adaptive optics retinal images. *JOSA A*, 24(5):1358–1363.

[Mariotti and Devaney, 2015] Mariotti, L. and Devaney, N. (2015). Performance analysis of cone detection algorithms. *J. Opt. Soc. Am. A*, 32(4):497–506.



IRISH MACHINE VISION & IMAGE PROCESSING

Conference proceedings 2015

26 - 28 August 2015

Trinity College Dublin

Dublin 2, Ireland

Published by the Irish Pattern Recognition & Classification Society (web: iprcs.org)

ISBN 978-0-9934207-0-2