UNIVERSITY OF DUNDEE

BMVA

School of Computing
UNIVERSITY OF DUNDEE

# BMVC'11 UK Student Workshop Proceedings

Edited by

**Jianguo Zhang**
School of Computing
University of Dundee

**ISBN: 1-901725-44-8**

**Published by :**
**The British Machine Vision Association and Society for Pattern Recognition**
http://www.bmva.org/

**Edited by:**
**Jianguo Zhang**
School of Computing
University of Dundee
UK

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use. The publisher makes no representation, express or implied, with regard to the accuracy of the information in this book and cannot accept any legal responsibility for any error or omission that may be made. Printed and bound in the United Kingdom.

BMVC
22
longer
DUNDEE 2011

# Workshop Program

Proc. of the 3rd British Machine Vision UK Student Workshop (BMVC'11 WS, Dundee, Scotland, 2nd September, 2011)

# Shared Parts for Deformable Part Models

Patrick Ott
http://www.comp.leeds.ac.uk/ott/
Mark Everingham
http://www.comp.leeds.ac.uk/me/

School of Computing
University of Leeds
West Yorkshire, UK

A recent approach to object detection, which stands out for its success on recent PASCAL Visual Object Class (VOC) challenges [1] is the deformable part-based model (DPM) of Felzenszwalb *et al*. [2]. The DPM method incorporates a higher degree of *learnt* invariance by partitioning the object model into a set of local parts which are allowed to move around subject to soft spatial constraints. In addition, the DPM method uses the idea of a mixture model to capture the large variation in appearance that an object category may present.

One might hope to increase the accuracy of the DPM by increasing the number of mixture components and parts to give a more faithful model, but multiple reasons prevent this approach from being effective: (i) since mixture components have separate parts the number of parameters increases linearly with the number of mixture components; (ii) since training examples are assigned to a single mixture component, the amount of training data per component decreases linearly with the number of components.

We propose extensions [3] to the DPM allowing for more powerful detectors to be built. The key idea is to *share* parts between detectors so that (i) the overall number of parameters is reduced, encouraging generalization from finite training data; (ii) training examples are shared across all relevant parameters, such that the paucity of available training data has less negative impact. This results in more compact models and allows training examples to be shared by multiple components, ameliorating the effect of a limited size training set.

Parts are shared both (i) *within* object categories, *e.g*. a 'wheel' part may be shared across mixture components representing different viewpoints of a car; (ii) *across* object categories, *e.g*. the same wheel part may be shared by detectors for both cars and motorbikes. We pose learning of the extended DPM with shared parts as minimization of a novel energy function allowing simultaneous learning of parts for multiple object classes and mixture components.

We report state-of-the-art results on the PASCAL VOC dataset [1] and establish the effectiveness of part sharing over detectors which do not share parts.

**Patrick Ott Biography:** Patrick Ott obtained his Diploma in Computer Science and Business at the Anhalt University of applied Sciences (Germany) and the Hangzhou Dianzi University (China). He is currently a final-year PhD student in the vision group at the University of Leeds, supervised by Mark Everingham.

## References

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 88(2), 2010.

[2] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), 2010.

[3] P. Ott and M. Everingham. Shared parts for deformable part-based models. In *CVPR*, 2011.

Proc. of the 3rd British Machine Vision UK Student Workshop (BMVC'11 WS, Dundee, Scotland, 2nd September, 2011)

# Cross Spectral Face Recognition between Near Infrared and Visible Faces

Debaditya Goswami
d.goswami@surrey.ac.uk
David Windridge
d.windridge@surrey.ac.uk
Chi Ho Chan
chiho.chan@surrey.ac.uk
Josef Kittler
j.kittler@surrey.ac.uk

Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford
Surrey, UK

**Abstract**

Performing facial recognition by comparing Near Infrared (NIR) probe images against a gallery of visible-light (VIS) images is a relatively new method of countering illumination variation problems in face recognition. In this paper we describe an approach which learns the relationship between NIR-VIS image pairs to perform classification. A series of preprocessing algorithms, followed by LBP representation are used to transform the image-pairs into a separate, lower-dimensional subspace where a relationship between the modalities is learnt using Canonical Correlation Analysis. We demonstrate how the novel combination of LBP feature histograms and CCA classification performs in comparison with several other pre-processing techniques.

## 1 Introduction

Traditional authentication mechanisms like keys, ID-cards, tokens are no longer sufficient to provide a truly robust security solution. The integrity of such security systems is based on the premise that the owner of the authenticating device is not an impostor. Biometric systems are based on who the user is or how the user behaves. Due to a combination of factors, including greater computing power, storage and breakthroughs in facial image processing - interest in facial features as a biometric has increased.

A major problem facing face recognition systems in the visible light (VIS) spectrum is that of illumination variation[1]. To combat this, various approaches have been tried, including the use of 3-D [4], Thermal IR [13][10] and even fusion of images from multiple spectra [18][5].However, such techniques require acquisition of new databases, and interoperability with existing optical databases is minimal.

The use of Near Infrared (NIR), usually described as lying in the 800 - 1000 nm range, in face-recognition to counter illumination variation has become well-established [25] [14] [24] [15]. The advantages over facial images recorded in the visible light spectrum are that indirect illumination can be largely eliminated. This is due to the fact that when capturing images in the NIR spectrum, it is very unlikely that there are multiple sources of NIR illumination present. Again, these methods are subject to the same restrictions as other non-VIS methods. In typical, real-life identification scenarios it will not be the case that the *gallery* image with which one is attempting to seek a correlation is acquired in the NIR modality. A more useful scenario would be matching visible images to NIR images when attempting person identification. In such cases, the use of a NIR *probe* image is still advantageous, providing a fixed reference point, as opposed to an attempt to match pairs of visible images with arbitrarily varying illuminations.

However the use of NIR introduces certain systematic deviations into the probe-gallery comparison. These include differing absorption levels, subsurface scattering functions for human skin [9] and various other spectral inequalities. Matching NIR probe images against VIS gallery images is quite new in the field of face recognition. A general learned association algorithm that can be applied to heterogeneous image problems is proposed in [16], and referred to as Common Discriminant Feature Extraction (CDFE) by the authors. This was originally developed to enable recognition between VIS images and their sketch images. It is then used to perform face recognition between VIS and NIR images. Li *et al* [22] use a system of canonical correlation analysis (CCA) in which the learning mechanism is implemented between features in linear discriminant analysis (LDA) subspaces.

Reiter *et al* [19] use CCA and active appearance models [7] to predict NIR face depth maps from RGB face data, thereby demonstrating the relation between NIR and VIS from a simulation-based perspective. While not a true face recognition system when considered on its own, the method described here could be modified for use in such a system. In [23], the authors tackle multiple problems in their work. Using the MBGC Portal Challenge dataset, they attempt to match partial NIR video clips to a gallery of full frontal VIS images.

The use of Local Binary Patterns (LBP) in heterogeneous face recognition has also been attempted. In [6], the authors combine LBP feature representation and patch-based manifold learning to synthesize a virtual sample from an input image. Use of the LBP operator is made again in [12], where the authors additionally extract histograms of oriented gradients (HoG) feature descriptors in an effort to increase face recognition performance.

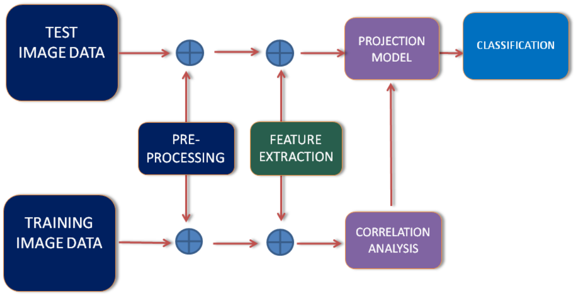


Figure 1: An overview of the system described in this paper. The feature extraction stage consists of combinations of LBP, PCA and LDA which feeds into CCA. The learned model is then used to transform the test images prior to classification.(Refer Section 3 for details)

In this paper, we attempt to use the novel combination of LBP histograms of corresponding NIR-VIS image pairs to learn a relationship using CCA. LBP features should be ideally matched to the NIR-VIS matching problem in that they largely remove all large-scale

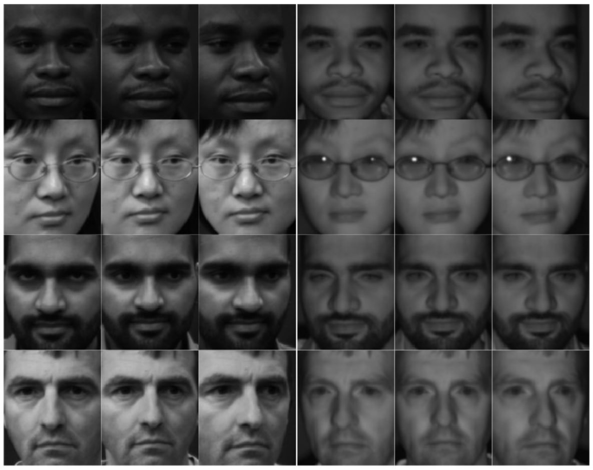


Figure 2: Images from the dataset with the pose-reflection (-10/0/+10) for both VIS (left) and NIR (right)

morphological considerations from the CCA matching (since CCA can at best approximate a linear filter). Performing CCA directly between image pairs is prohibitively expensive computationally due to a significant overhead. Hence a number of experiments, including projection into a lower dimensional subspace (using PCA/LDA) before CCA are described. Figure 1 shows an overview of the proposed system. The rest of the paper is organised in the following manner: Section 2 contains some background theory on key aspects of the algorithm. Section 3 details the methodology and procedures used in the experiments. Section 4 discusses and analyses the results after experimentation. Section 5 concludes by summarising the experimental procedure and results, followed by avenues of future work.

# 2 Theoretical Background

## 2.1 Spectral Differences between VIS and NIR

VIS images are captured in the 0.4 - 0.7 $\mu$m band. NIR images on the other hand, lie in the 0.8 - 1.0 $\mu$m band (invisible to the naked eye). The popularity of NIR as a comparable source to VIS is based on the fact that it is far enough from the visible light spectrum so as to be unaffected by ambient visible light sources, and yet exhibit similar facial structures (as seen in Figure 2). However, there exist a number of non-trivial difficulties in comparing the two modalities. These include differential visibility of features at differing wavelengths, and also the effects of excessive feature diffusion (which increases in magnitude as we move along the NIR spectrum). Difficulties also exist if the information available to the differing spectral modalities is of a differing quality [8]. Of these, the differential visibility of features represent potentially the most significant obstacle for face matching.

In this paper, we attempt to transform the VIS and NIR images into an alternate space which enhances their commonality, while retaining enough discriminative information between subjects. This is done with various combinations of LBP histogram quantisation and dimensionality reduction (PCA/LDA), followed by application of CCA.

## 2.2 Local Binary Patterns

$$LBP(x_c, y_c) = \sum_{n=0}^{7} 2^n s(i_n - i_c), \quad (1)$$

where $n$ is the number of neighbours (8 in this case) over the central pixel $c$, $i_c$ and $i_n$ are the gray level values of the image at $c$ and $n$, and $s(u)$ is 1 if $u \geq 0$ and 0 otherwise.

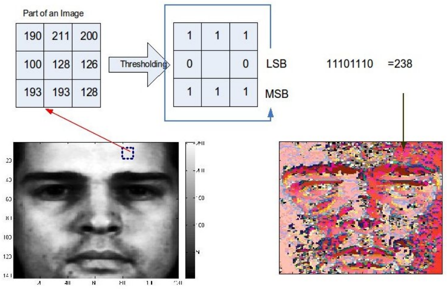


Figure 3: The LBP operator (left), and uniform LBP image representation (right)

Local Binary Patterns [17] (LBP) are a form of texture representation that employs structured ordinal contrast encoding to capture the complex image micro-structure. Figure 3 shows the basic LBP operator and its effect on an image. LBP is insensitive to monotonic gray-level transformations, which makes it a highly robust representation for use in cross spectral face recognition. One of the more interesting aspects of LBP relates to the concept of uniform LBP patterns. Ojala *et al*, state that approximately 90% of discriminative information in an image is restricted to uniform variants of LBP. A uniform LBP contains at most two bit-wise transitions. It describes 2 types of patterns - rotational patterns, such as edges and two non-rotational patterns, such as a bright spot or a flat area.

Considering the inherent spectral properties (monotonic invariance) of NIR images, an LBP histogram representation of such an image would enable comparison with a VIS image possessing a relatively higher degree of variation. Hence a corresponding representation of a VIS image by its LBP histogram can be used to eliminate a large portion of the spectral differences between VIS and NIR images. However, it should be noted that skin texture can still vary between NIR and VIS, and LBP does not account for these differences. For the purposes of the experiment, multi-region (8x8, non-overlapping segments) uniform LBP feature histograms were used[2]. This has been demonstrated to improve face recognition accuracy. However, it should be noted that excessive segmentation can render the system vulnerable to misalignment or mis-registered faces. The uniform LBP feature histograms were obtained from the images in the following manner:

1. Perform uniform LBP on the overall image
2. Divide image into 8x8, non-overlapping segments.
3. Populate LBP histogram bins for each uniform LBP image segment

A significant benefit of using uniform LBP in this system was the reduction in number of bins per LBP segment down to 59 from 256, resulting in an overall 64x59 matrix representing the image. This meant that for the next stage of the process involving CCA modeling, rasterised multi-region LBP vectors (1 x 3776) further reduced computational overhead.

### 2.3 Canonical Correlation Analysis

Canonical Correlation Analysis is a multivariate regression method introduced by Harold Hotelling [11]which attempts to describe the relationship between 2 data sets in terms of their cross-covariance matrices. In face recognition, CCA is a powerful tool that can be used to identify correlation between 2 sets of images (VIS and NIR for example). Consider 2 random, correlated vectors $x$ and $y$, analogous to VIS and NIR image pairs respectively. CCA can be used in this case to obtain pairs of directions $w_x$ and $w_y$ which maximise the correlation between the projections $x = \mathbf{w}_x^T\mathbf{x}$ and $y = \mathbf{w}_y^T\mathbf{y}$. The directions can then be obtained as a maxima of the following function:

$$\rho = \frac{E[xy]}{\sqrt{E[x^2]E[y^2]}} = \frac{E[\mathbf{w}_x^T\mathbf{x}\mathbf{y}\mathbf{w}_y^T]}{E[\mathbf{w}_x^T\mathbf{x}\mathbf{x}^T\mathbf{w}_x]E[\mathbf{w}_y^T\mathbf{y}\mathbf{y}^T\mathbf{w}_y]} \tag{2}$$

This can be represented:

$$\rho = \frac{\mathbf{w}_x^T\mathbf{C}_{xy}\mathbf{w}_y}{\sqrt{\mathbf{w}_x^T\mathbf{C}_{xx}\mathbf{w}_x\mathbf{w}_y^T\mathbf{C}_{yy}\mathbf{w}_y}} \tag{3}$$

where $\mathbf{C}_{xx} \in \mathbb{R}^{p \times p}$ and $\mathbf{C}_{yy} \in \mathbb{R}^{q \times q}$ are the *within-set covariance matrices*, while $\mathbf{C}_{xy} \in \mathbb{R}^{p \times q}$ is the *between-set covariance matrix.* The correlation score, $S$ can be calculated using normalised cross correlation between the output CCA projection vectors $x$ and $y$. Namely:

$$S = \frac{x.y}{||x||.||y||} \tag{4}$$

One of the major strengths of CCA is the fact that almost any 2 signals, with some degree of correlation, can be used as input vectors to learn a correlation-relationship between them. However, it should also be noted that CCA is highly dataset dependent. Given a lack of observation samples, it was necessary to artificially inflate the training data by including images which had been flipped about their vertical axis. Dimensionality reduction also plays an important role in the CCA learning process. Computing CCA directly on the images is a very computationally intensive task, and results in a high level of sparsity in any projection matrices. In this paper, we demonstrate the effectiveness of CCA in face recognition by using a combination of modalities, input vectors and combinations of both. (Further details in Section 3)

## 3 Methodology

### 3.1 Data

The data collected consists of 150 subjects for both NIR and VIS. As seen in Figure 4, for each subject and modality there are 3 deviations from -10 through frontal to +10. The illumination profile is set to simulate a point-source of light on the forehead for a fully frontal face. Although every attempt was made during data capture to ensure that the poses were under strict control, there were still a few cases of systematic errors. As a result, not all the images were strictly of the deviations intended. This seemed to have no negative effect on the learned model, which is shown to be relatively insensitive to small deviations. Manually annotated eye-locations for the data were used for the pre-processing.

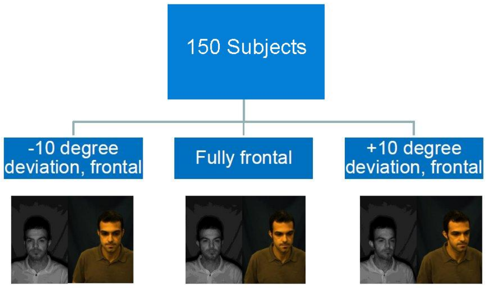


Figure 4: The dataset contained 150 exclusive subjects, with varying poses of -10/0/+10 degrees. Corresponding NIR and VIS pairs for each subject and pose were collected.

## 3.2 Data Preparation

The image data for both VIS and NIR was put through a series of pre-processing algorithms prior to calculating their LBP representations. First, the face area was detected using manually annotated eye locations.These detected faces were scaled and cropped to a 140x128 resolution. Following this, the images were transformed to grayscale and normalised to ensure zero-mean and unit variance. The normalised, cropped faces are then either piped into a sequential photometric chain of further pre-processing algorithms prior to LBP, or directly into the LBP histogram generator. To compare the baseline performance for cross-spectral face recognition, the photometric pre-processing outlined by [21] is used. This sequential chain of pre-processing algorithms (SQ preprocessing hereafter) ensures that the effects of illumination variation are countered to a large extent, while still retaining distinguishable facial features for use in face recognition. The data is preprocessed using the following chain of algorithms:

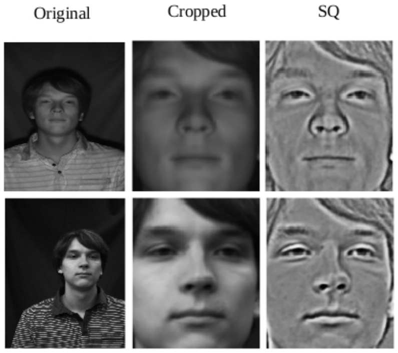


Figure 5: This figure shows the image pipeline that was fed into the Feature Selection stages of the process (NIR above, VIS below). From left to right, the original image was cropped and then pre-processed using the sequential chain of photometric algorithms.

- **Gamma correction**: This algorithm enhances the local dynamic range of the image in darker sections, while compressing it in bright regions and at highlights.

- **Difference of Gaussian Filtering**: Essentially a bandpass filter, DoG filtering removes non-essential shading from face images while still retaining the broader, more distinguishable features.
- **Contrast Equalisation**: Global rescaling of the image intensity values to ensure that there are no 'extreme' values

In addition, experiments substituting the SQ preprocessing for histogram equalisation were also conducted. This enabled us to further investigate the effects of monotonic invariance between NIR-VIS in the LBP space. Following the determination of the final input vectors to the CCA algorithm, a combination of dimensionality reductions were used to ensure greater computational efficiency, as well as investigating the benefits of greater inter/intra-class disparity. To this end, standard PCA and LDA algorithms, which are well documented in face recognition were used [3].

### 3.3 Experimental Procedure

The main aim of the experiments was to identify and evaluate the best algorithms for NIR → VIS face matching. Figure 6 shows the overall methodology for the experiments conducted in this paper. The combination of sequential chain pre-processing, LBP histogram generation, dimensionality reduction and use of CCA lead to a number of testable algorithms, which are outlined in Table 1. The CCA relationship for all experiments was derived using

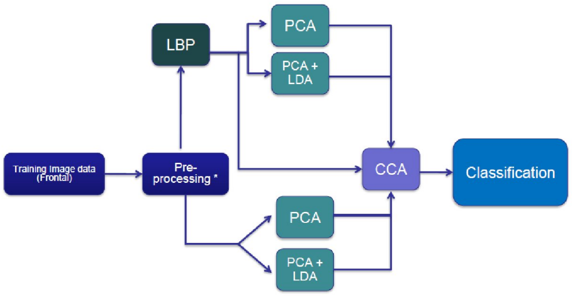


Figure 6: Experimental Procedure

NIR and VIS fully frontal images only. To simulate multiple samples per pose per subject, it was necessary to use images that had been flipped about their vertical axis (especially for LDA). CCA is well documented to be highly dependent on the size of the training set [20], and smaller training sets can result in over-fitting. Purely frontal cross-modal CCA experiments are thus omitted, since including them would mean a reduction in training images resulting from the split into training and test frontal images. Instead both the +/-10 degree deviated images are used for testing. As the results show, the CCA model is not affected greatly by the pose variation and performs comparably to pose-invariant, cross-spectral face matching results with pure LBP. A set of match scores were obtained using normalised cross-correlation (Refer to Eqn 4) on the CCA projections of the probe and gallery datasets. The classification was performed using a minimum distance classifier, to obtain Rank 1 recognition performance for all the experiments. Table 1 shows a list of all the experiments carried out, along with the performance of each algorithm.

| **Probe** | **Gallery** | **Algorithm** | **Performance (%)** |
|---|---|---|---|
| NIR frontal | VIS frontal | LBP | 54 |
| NIR frontal | VIS frontal | histeq+LBP | 53.33 |
| NIR frontal | VIS frontal | SQ+LBP | 85.67 |
| VIS deviated | VIS frontal | LBP | 99 |
| VIS deviated | VIS frontal | histeq+LBP | 99 |
| VIS deviated | VIS frontal | SQ+LBP | 99.33 |
| NIR deviated | NIR frontal | LBP | 98.67 |
| NIR deviated | NIR frontal | histeq+LBP | 98 |
| NIR deviated | NIR frontal | SQ+LBP | 99 |
| NIR deviated | VIS frontal | LBP | 46.44 |
| NIR deviated | VIS frontal | histeq+LBP | 44 |
| NIR deviated | VIS frontal | SQ+LBP | 65.67 |
| NIR deviated | VIS frontal | LBP+CCA | 95 |
| NIR deviated | VIS frontal | PCA+CCA | 86.67 |
| NIR deviated | VIS frontal | PCA+LDA+CCA | 75.67 |
| NIR deviated | VIS frontal | LBP+PCA+CCA | 95.33 |
| NIR deviated | VIS frontal | LBP+PCA+LDA+CCA | **98.33** |
| NIR deviated | VIS frontal | SQ+LBP+PCA+CCA | 94.67 |
| NIR deviated | VIS frontal | SQ+LBP+PCA+LDA+CCA | 97 |

Table 1: Detailing the algorithms and modal relationships tested. The best performing algorithms are highlighted in red. Note the highest performing combination of SQ+LBP+PCA+LDA+CCA with a recognition rate of 98.33%

## 4 Results

Figure 7 shows the recognition rates for the unimodal experiments using LBP histograms (only). It can be seen that pose-variation has some effect on the recognition performance, but only if the matched modalities are distinct. For unimodal systems, the base LBP classification is 99.3% for the VIS deviated → VIS frontal [SQ+LBP]

The results of classification using CCA can be seen in Figure 8 . Consider the figures when compared with their input CCA vectors. There are 3 types -Cropped faces without SeqChain or LBP, Cropped face LBP histograms without SeqChain and Cropped face LBP histograms **with** SeqChain. It is clearly seen that the use of LBP has a significant effect on the recognition performance.

There are several points of interest here. The drop in performance between PCA+CCA and PCA+LDA+CCA when using **non-LBP** face images can be attributed mainly to a rudimentary LDA training. With a high ratio of features:samples/class, it is quite possible that the LDA projection removes essential, discriminatory data from the vectors, giving rise to a performance deficit when compared with the corresponding LBP performance.

Another point of interest to note here is that the SQ preprocessing does not add anything to the CCA algorithm. In fact, a reduction in performance is seen. Considering the sequence of preprocessing algorithms that are used, it is apparent that the NIR-VIS image pairs are already transformed into a space which increases their inherent similarity, thereby reducing the effect CCA has on the final correlation measure.

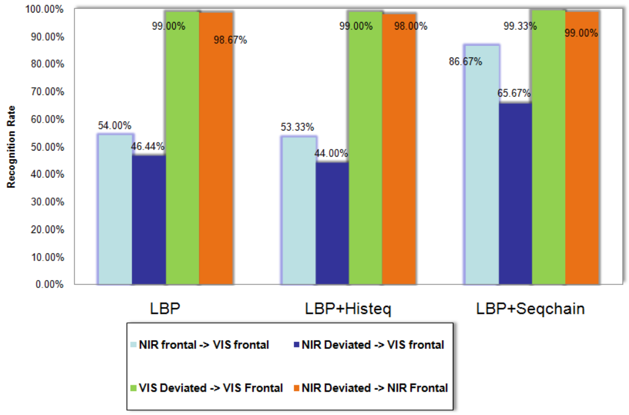


Figure 7: Rank 1 Recognition Performance for pure LBP classification (without CCA), where SQ=Sequential Chain Preprocessing, histeq=Histogram Equalisation and LBP=Local Binary Patterns

Finally, compare the sets of CCA (Fig.8) vs non-CCA (Fig.7) results. It can be seen that the peak performance for cross-spectral face matching using CCA is 98.33%. The highest performance of VIS→VIS matching is 99.33%. This is despite the fact that the CCA is trained using frontal and tested using deviated images. Additionally, the way in which the data collection was undertaken results in little to no illumination variance between the gallery images (which are scenarios where the benefits of cross-spectral matching become fully apparent). So we can see that the performance of cross-spectral matching using LBP features combined with CCA gives a very comparable benefit when compared to direct VIS→VIS face matching.

## 5 Conclusions

In this paper, we have attempted to solve the problem of NIR →VIS, cross-spectral face matching using a unique combination of LBP histograms to learn a relationship between the 2 modalities with CCA. We hypothesised that this would enable a purer comparison between NIR and VIS, by removing a large amount of the differences attributable to their separate spectral and textural properties. A series of experiments to verify this using combinations of cross-modal, cross-pose, pre-processed input images were conducted. Results indicated that the best cross-modal performance was obtained by using the LBP+PCA+LDA+CCA algorithm, which gave a recognition rate of 98.67%. This is comparable with a maximum performance of 99.33% using LBP/VIS deviated-VIS frontal (without CCA).

An important point to note is that the benefits of cross-spectral matching are truly apparent in scenarios which involve a high variability in illumination profiles between the source and probe images. The use of NIR imagery at the probe point enables the regularisation of illumination, thereby enabling a more robust face recognition solution. Given that the data used in the experiments did not contain significant amounts of illumination variation with respect to VIS probe and VIS gallery images, the strong performance of NIR-VIS matching

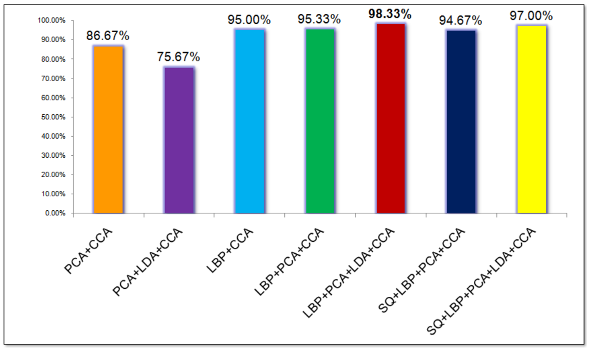


Figure 8: Rank 1 Recognition Performance for CCA classification. For each type of input vectors there are a set of pre-processing and dimensionality reduction techniques. Note that the CCA training is always done using NIR frontal and VIS frontal images.

is potentially significant.

Future experiments which incorporate a greater volume of data samples (and variability) are planned. We also note that the progress of NIR-VIS face matching to date has been mainly through model and learning-based approaches. Relatively few techniques attempt to physically model the relationship between the modalities, and this is therefore an avenue of further exploration.

# References

[1] Y. Adini, Y. Moses, and S. Ullman. Face recognition: The problem of compensationg for changes in illumination direction. In *IEEE Transactions on Pattern analysis and Machine Intelligence*, 1997.

[2] Timo Ahonen, Abdenour Hadid, and Matti PietikÃď'inen. Face recognition with local binary patterns. In *ECCV*, 2004.

[3] Peter N. Belhumeur, Jo ao P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, volume 19, pages 711–720, 1997.

[4] K. W. Bowyer, Chang, and P. J. Flynn. A survey of 3d and multi-modal 3d+2d face recognition. In *In Proceedings of International Conference Pattern Recognition*, page 358C361, August 2004.

[5] Hong Chang, A. Koschan, M. Abidi, S.G. Kong, and Chang-Hee Won. Multispectral visible and infrared imaging for face recognition. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–6, June 2008. doi: 10.1109/CVPRW.2008.4563054.

[6] Jie Chen, Dong Yi, Jimei Yang, Guoying Zhao, Stan Z. Li, and Matti PietikÃď'inen.

Learning mappings for face synthesis from near infrared to visual light images. In *IEEE conference on Computer Vision and Pattern Recognition*, 2009.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearence models. In *IEEE Transactions PAMI*, volume 23(6), pages 681–685, 2001.

[8] Omolara Fatukasi, Josef Kitter, and Norman Poh. Quality controlled multimodal fusion of biometric experts. In *CIARP*, pages 881–890, 2007.

[9] D Goswami, D Windridge, and J Kittler. Subsurface scattering deconvolution for improved nir-visible facial image correlation. In *8th IEEE International Conference on Automatic Face and Gesture Recognition*, September 2008.

[10] J. Heo, B. Abidi, S. G. Kong, and M. Abidi. Performance comparison of visual and thermal signatures for face recognition. In *Biometric Consortium Conference*, 2003.

[11] H Hotelling. Relations between two sets of variates. In *Biometrika*, volume 28, 1936.

[12] Brendan Klare and Anil K. Jain. Heterogeneous face recognition: Matching nir to visible light images. In *International Conference on Pattern Recognition*, 2009.

[13] S. G. Kong, J. Heo, B. Abid, J. Paik, and M. Abidi. Recent advances in visual and infrared face recognition - a review. In *Computer Vision and Image Understanding, 97(1):103C135*, January 2005.

[14] Stan Z. Li, RuFeng Chu, ShengCai Lao, and Lun Zhang. Illumination invariant face recognition using near-infrared images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 29, pages 627 – 639, April 2007.

[15] S. Liao, Z.Zhu, Z.Lei, L.Zhang, and S.Z.Li. Learning multi-scale block local binary patterns for face recognition. In *International Conference on Biometrics*, pages 828–837, 2007.

[16] Dahua Lin and Xiaoou Tang. Inter-modality face recognition. In *Computer Vision âĂŞ ECCV 2006*, volume Volume 3954/2006 of *Lecture Notes in Computer Science*, pages 13–26. Springer Berlin / Heidelberg, july 2006.

[17] T Ojala, M Pietikainen, and T Maenpaa. A comparative study of texture measures with classification based on feature distributions. In *Pattern Recognition*, volume 29, 1996.

[18] Z.H. Pan, G. Healey, M. Prasad, and B. Tromberg. Face recognition in hyperspectral images. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 25 no. 12, pages 1552–1560, December 2003.

[19] Michael Reiter, Rene Donner, Georg Langs, and Horst Bischof. 3d and infrared face reconstruction from rgb data using canonical correlation analysis. In *International Conference on Pattern Recognition*, 2006.

[20] Quan-Sen Sun, Pheng-Ann Heng, Zhong Jin, and De-Shen Xia. Face recognition based on generalized canonical correlation analysis. In *Lecture Notes in Computer Science, Advances in Intelligent Computing*, volume 3645/2005, pages 958–967, 2005.

[21] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *Analysis and Modelling of Faces and Gestures*, volume 4778/2007, pages 168–182, 2007.

[22] Dong Yi, Rong Liu, RuFeng Chu, Zhen Lei, and Stan Z. Lei. Face matching between near infrared and visible light images. In *International Conference, ICB 2007, proceedings*, volume 4642/2007, pages 523 – 530, August 2007.

[23] Dong Yi, ShengCai Liao, Zhen Lei, Jitao Sang, and Stan Z. Li. Partial face matching between near infrared and visual images in mbgc portal challenge. In Massimo Tistarelli and Mark S. Nixon, editors, *ICB*, volume 5558 of *Lecture Notes in Computer Science*, pages 733–742. Springer, 2009. ISBN 978-3-642-01792-6. URL http://dblp.uni-trier.de/db/conf/icb/icb2009.html#YiLLSL09.

[24] S.Y. Zhao and R.R. Grigat. An automatic face recognition system in the near infrared spectrum. In *Proc. IntâĂŹl Conf. Machine Learning and Data Mining in Pattern Recognition*, pages 437–444, July 2005.

[25] X Zou, J Kittler, and K Messer. Ambient illumination variation removal by active near-ir imaging. In D Zhang and A Jain, editors, *Proceedings of the International Conference on Biometrics*, pages 19–25. Springer, January 2006.

Proc. of the 3rd British Machine Vision UK Student Workshop (BMVC'11 WS, Dundee, Scotland, 2nd September, 2011)

# Psychologically inspired dimensionality reduction for 2D and 3D Face Recognition

Mark F. Hansen
mark.hansen@uwe.ac.uk
Gary A. Atkinson
gary.atkinson@uwe.ac.uk
Melvyn L. Smith
melvyn.smith@uwe.ac.uk
Lyndon N. Smith
lyndon.smith@uwe.ac.uk

The Machine Vision Laboratory
University of the West Of England
Bristol, UK

### Abstract

We present a number of related novel methods for reducing the dimensionality of data for the purposes of 2D and 3D face recognition. Results from psychology show that humans are capable of very good recognition of low resolution images and caricatures. These findings have inspired our experiments into methods of effective dimension reduction. For experimentation we use a subset of the benchmark FRGCv2.0 database as well as our own photometric stereo "Photoface" database. Our approaches look at the effects of image resizing, and inclusion of pixels based on percentiles and variance. Via the best combination of these techniques we represent a 3D image using only 61 variables and achieve 95.75% recognition performance (only a 2.25% decrease from using all pixels). These variables are extracted using computationally efficient techniques instead of more intensive methods employed by Eigenface and Fisherface techniques and can additionally reduce processing time tenfold.

## 1 Introduction

Automatic face recognition has been an active area of research for over four decades and a key part of this research is understanding how different data representations affect recognition rates and efficiency. Digital images of faces have a very high data dimensionality: a $200 \times 200$px image defines a point in a 40000-dimensional space, making computation a slow and resource hungry process. This is compounded when faces images are extended into 3D models. Reducing the dimensionality of the data without discarding the discriminatory information is the aim of this research. If a face can effectively be reduced down from many thousands of dimensions of raw data to a few tens of dimensions as in this paper, then storage needs become far less and processing becomes far faster. This has obvious applications for industrial and commercial implementations.

In this paper, we prove the following contributions for both the FRGCv2.0 database [11] and our own photometric stereo database [22] captured using the "Photoface" device [6]:

1. Optimal recognition results for close-cropped faces are obtained when the resolution is reduced to a mere 10x10 pixels.
2. The exclusive use of just 10% of the data (chosen to be those pixel locations with the greatest variance) is sufficient to maintain recognition rates to within 10% of those rates that include all of the data.
3. When combining the above two contributions we perform recognition at an accuracy of 96.25% for 40 subjects using only 61 dimensions (pixels). This compares to 98% when the full 80x80 resolution is used on all data.

Ultimately we aim to compare dimension reduction techniques based on a percentile and variance based inclusion principle (to exclude 90% of the data) with a baseline condition containing all pixels.

Our own database, Photoface, provides over 3000 sessions of 457 individuals, and scans are captured using photometric stereo [17] which results in estimated surface normals at each pixel. Full details of the actual device used can be found in [6] and an example of a scan can be seen in Fig. 1. The FRGCv2.0 database, which we also use in this paper, does not provide the surface normals. They can be calculated by numerically differentiating the point cloud data. We also include experiments on the depth map images to rule out any errors introduced by differentiation.

Using 3D data for face recognition allows for pose and illumination correction which are two commonly cited problems with conventional 2D images. Better recognition rates have also been reported using 3D over 2D data [4], although this is not always replicated [7]. One reason for this may be the representation of the 3D data used in the analysis. Gökberk *et al*. [5] performed recognition experiments using numerous 3D representations. They concluded that *'... surface normals are better descriptors than the 3D coordinates of the facial points.'* This is at odds with most research which uses the 3D point coordinates as a starting point. Surface normals are used in the experiments performed in this paper for this reason.

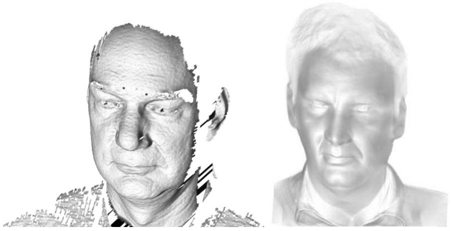


Figure 1: Examples of FRGCv2.0 (left) and Photoface (right) 3D scans. NB They are not of the same person.

There are many mathematical techniques for dimensionality reduction, and in particular the Eigenface [15] and Fisherface [2] (based on Principle Components Analysis (PCA) and Fisher's Linear Discriminant (FLD) respectively) techniques are commonly used in face recognition. With an added dimension, 3D face models potentially compound the problem for large data storage. Recent techniques such as sparse representation (such as non-negative matrix factorization) and manifold learning (such as local linear embedding [8]) show that effective methods of dimension reduction are a key topic. Methods that can reduce the amount of data without discarding discriminatory information are essential for faster processing and optimal solutions. There have been many attempts in the literature to extend and generalise PCA, FLD and other methods [19, 20, 21] in order to improve robustness to pose, illumination, etc, typically at the expense of computational efficiency. The main contribution of this paper by contrast, is to show that for the constrained case of frontal 2.5D data, then the efficiency can be improved even compared to PCA by using more direct analysis without the need to project into a new subspace.

Caricaturing essentially enhances those facial features that are unusual or deviate sufficiently from the norm. It has been shown that humans are better able to recognise a caricature than they are the veridical image [10, 12]. This finding is interesting as caricaturing is simply distorting or adding noise to an image. However this noise aids human recognition and this, in turn, provides insights into the storage or retrieval mechanism used by the human brain.

Unnikrishnan [16] conceptualises an approach similar to face caricatures for human recognition. In this approach, only those features which deviate from the norm by more than a threshold are used to uniquely describe a face. Unnikrishnan suggests using those features whose deviations lie below the $5^{th}$ percentile and above the $95^{th}$ percentile, thereby discarding 90% of the data. Unnikrishnan provides no empirical evidence in support of his hypothesis, so an aim of this paper is to test the theory experimentally. We do this in two ways: the first directly tests his theory, finding the thresholds for each pixel which represent the $5^{th}$ and $95^{th}$ percentile values and only including those pixels in each scan which lie outside them (outliers). The second is loosely based on Unnikrishnan's idea, and looks at the variance across the whole database to calculate the pixel locations with the largest variance. Only the pixels at these locations are then used for recognition.

An obvious method of reducing the amount of data is to downscale the images. A great deal of research has gone into increasing the resolution of poor quality images (super-resolution [1, 18], hallucinating [23]) by combining images or using statistical techniques to reproduce a more accurate representation of a face (*e.g.* from CCTV footage). By contrast, little research attempts to directly investigate resolution as a function of recognition rates on 3D data. Toderici *et al.* state that there is little to be gained from using high resolution images [14], Boom *et al.* state that the optimum face size is $32 \times 32$ px for registration and recognition [3], a view which is reinforced by a more recent study by Lui *et al.* who state that optimum face size lies between 32 and 64 pixels [9]. These experiments have used 2D images. Chang *et al.* use both 2D and 3D data and conclude that there is little effect of decreasing resolution up to 25% on 2D data and 50% on 3D [4] using PCA. In summary, the research suggests that relatively low resolutions give optimum recognition (for the given recognition algorithms). These findings are conducive to the fact that the same appears to be true of human recognition [13].

# 2 Methods and data

This section details the datasets, preprocessing steps, and the methods used in the experiments.

## 2.1 Data and preprocessing

Experiments were performed on 10 sessions of 40 subjects facing frontally without expression on the FRGCv2.0 and our own photometric stereo database. 2D and 3D data are used in separate experiments.

The FRGCv2.0 dataset comes in point cloud format which is converted to a mesh via uniform sampling across facets. Noise is removed by median smoothing and holes filled by interpolation. Normals are then estimated by differentiating the surface. The depth map images are all normalized to have a minimum value of 0.

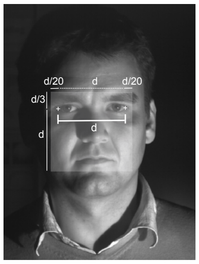


Figure 2: The cropped region of a face. The distance between the anterior canthi ($d$) is used to calculate the cropped region.

Data is cropped for both databases as follows: the median anterior canthi and nose tip across all sessions are used for alignment via linear transforms. The aligned images are then cropped into a square region as shown in Fig. 2 to preserve main features of the face (eyes, nose, mouth), and exclude the forehead and chin regions which can frequently be occluded by hair.

Our 2D experiments are based on data as follows: the accompanying colour image for each FRGCv2.0 scan is converted to greyscale, aligned and cropped in the same way as the 3D scan. The 2D images in the Photoface database are the estimated albedo images which are also aligned and cropped in the same way as the 3D data. Due to memory limitations, both the 2D and 3D data are then resized to $80 \times 80$ px and are reshaped into a 6400-dimension and a 12800-dimension (x and y components of the surface normals are concatenated) vector respectively.

## 2.2 Calculating outliers and variance

The thresholds for each pixel are calculated which represent the $5^{th}$ and $95^{th}$ percentile values. We are interested in the norm across the whole dataset for each pixel location rather than the norm for each image. For the 2D images, percentile values are calculated for the greyscale intensity value for each pixel location. There are 400 sessions, so there are 400 values for each pixel from which we calculate the percentile thresholds. The same process is performed for 3D surface normal data, giving $x$ and $y$ surface normal component thresholds for each pixel. Pixels which have a value between the $5^{th}$ and $95^{th}$ percentile are discarded, leaving only the 10% outlying data. We shall refer to this as the "percentile inclusion criterion". Examples can be seen in Fig. 3.

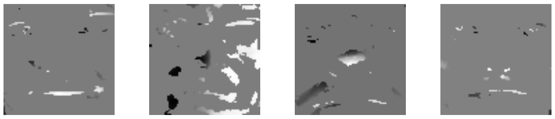


Figure 3: Examples of the $y$-components of the surface normals that have values outside the $5^{th}$ and $95^{th}$ percentiles for four subjects which are used for recognition.

The above method extracts the least common data from each session and that is what is used for recognition. Alternately, we can use the greyscale variance at each pixel location as a measure of discriminatory power. If a pixel shows a large variance across the dataset, then this might make it useful for recognition (assuming that variance within the class or subject is small). Therefore the standard deviation of each pixel is calculated over all the sessions. Whether or not a particular pixel location is used in recognition depends on whether or not the variance is above a pre-determined threshold. Examples of the use of different thresholds are shown in Fig. 4. We refer to this as the "variance inclusion criterion".

## 2.3 Image resizing

The effect of different resizing techniques on linear subsampling are investigated in terms of their effect on recognition as a function of resolution. Resizing is performed via the Matlab

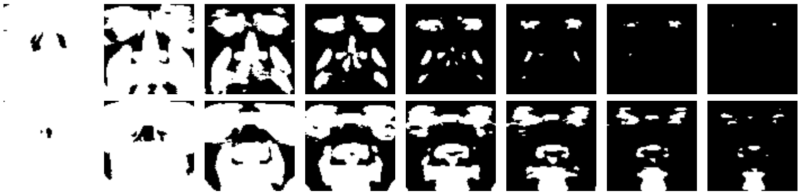


Figure 4: Examples of the regions which remain for $x$ (top row) and $y$-components (bottom row) as the threshold variance is increased from left to right. White regions are retained and black regions are discarded.

`imresize()` function using the deafult bicubic kernal type and with antialiasing on, as these settings were found to provide the best performance.

### 2.4 Recognition algorithm

Our experiments used to test recognition accuracy employ the leave-one-out paradigm. This dictates that every session is used as a probe against a gallery of all other sessions once. There are therefore 400 classifications per condition of which the percentage correctly identified is shown.

As the purpose of this research is feature extraction efficiency, the actual choice of classifier is not so important. We therefore implement Pearson product-moment correlation coefficient (PMCC) as a similarity measurement between a probe vector and the gallery vectors. The gallery session with the highest coefficient is regarded as a match. Experimentally, we found that PMCC gives similar performance on baseline conditions to the Fisherface algorithm but is approximately eight times faster.

## 3 Results

### 3.1 Dimensionality reduction via the percentile inclusion criterion

Unnikrishnan's theory states that we should expect reliable performance using only the data which lies outside the $5^{th}$ and $95^{th}$ percentiles [16]. Table. 1 shows recognition rates on 2D and 3D data using both all data and the outliers only. Note in particular that, for the 3D surface normal data, the rates drop by under 10% when using outlier data only. This effect seems limited to the surface normal data and is not seen in either the 2D or depth map data. We have included results from a fusion technique using the Photoface surface normal data combined with the albedo image. There is a small decrease in baseline performance and using only the outlying data leads to a severe decrease of about 34%.

| | | Baseline (All pixels) | Outliers (10% of pixels) |
|---|---|---|---|
| 2D | FRGC | 90 | 73.75 |
| | Photoface | 98 | 64 |
| 3D | FRGC Surface normals | 90.25 | 84.25 |
| | FRGC Depth map | 71.5 | 23.25 |
| | Photoface | 98.25 | 89.25 |
| Fusion | Photoface 2D + 3D | 97 | 63.25 |

Table 1: Baseline versus outlier performance (% correct).

Fig. 5 shows a plot of recognition rate as a function of which percentile range is used for recognition on 3D Photoface data. It should be noted that similar patterns of results were found for all datasets (2D, 3D and FRGC). As predicted, the figure shows that the best recognition performance is obtained using the most outlying percentiles. As expected also, the recognition rate reduces as the percentile ranges used tend toward the inliers. However, for the most inlying data of all (i.e. percentiles 45–55) we find a significant increase in performance. Contrary to Unnikrishnan's theory, this implies that there is discriminative data that is useful for face recognition in the most common data as well as the most outlying.

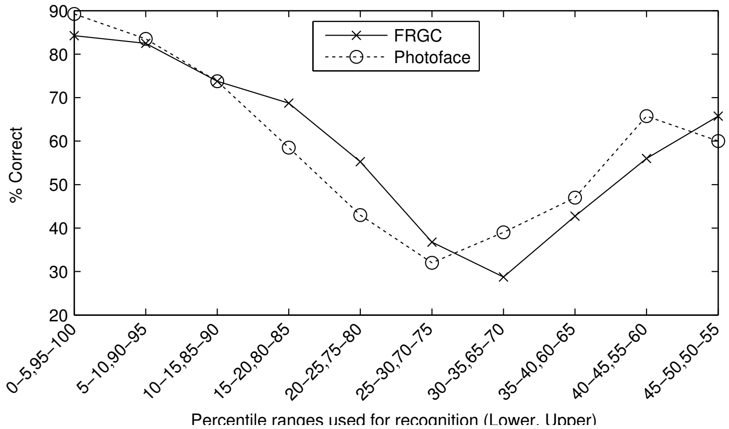


Figure 5: Recognition performance using pairs of percentile ranges for 3D data.

In a related experiment, we used single 5% ranges of data for recognition (i.e. $[0^{th} - 5^{th}], [5^{th} - 10^{th}]$ etc.) as shown in Fig. 6. Note that the increase in recognition performance for the most inlying data is not replicated. The slightly lower performance compared with Fig. 5 is because only 5% of the data is used instead of 10%.

Performance increases by combining ranges are not always observed. Consider, for example, the $25 - 30^{th}$ and $70 - 75^{th}$ percentiles for the FRGCv2.0 data. Individually the two percentiles give a performance around the 50% mark in Fig. 6, but when combined, the performance drops to around 40% in Fig. 5.

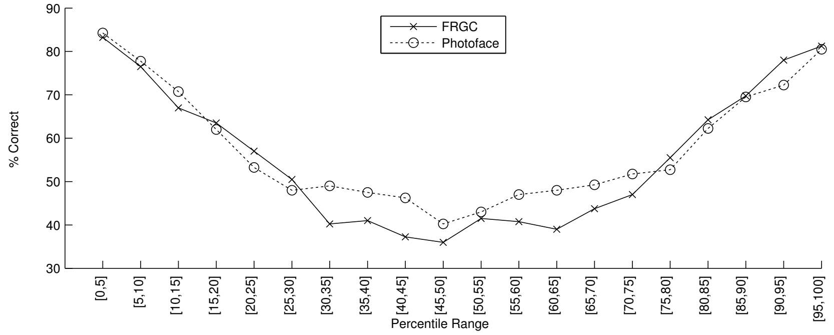


Figure 6: FRGC and Photoface data show a marked symmetry across ranges of percentiles.

## 3.2 Dimensionality reduction via the variance inclusion criterion

One problem with the above method is that the outlying points tend to be scattered across different parts of the images, making inter- and intra-comparisons between individuals somewhat unstructured. For the next method therefore, we use the same pixel locations in our recognition test for all images. Instead of using the percentiles defined within a single image as an inclusion criterion, we use the variance of a particular pixel across all subjects as explained in Sec. 2.2.

Fig. 7 shows plots combining the number of pixels which remain as we remove those with least variance (bar plot) against the recognition performance (line plot). It is apparent that we can achieve close to optimal performance while losing a large proportion of the pixels. We can discard approximately 75% of the least varying pixels and observe a corresponding drop of less than 10% in recognition performance on the FRGC data. Indeed, for Photoface data specifically, we only lose a few percent.

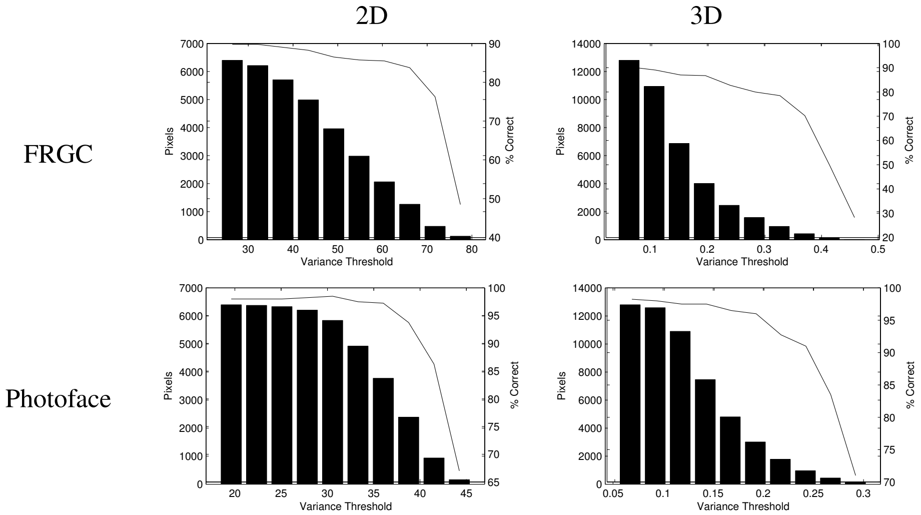


Figure 7: Recognition (line) as a function of retained pixels (bar chart). The pattern is shown in both sets of data (FRGC on the top row and Photoface on the bottom). 2D (grayscale for FRGC and albedo for Photoface) on the left, and surface normal data is shown on the right.

Table 2 shows a performance comparison of the two types of inclusion criteria when only 10% of pixels are retained. It is clear that by discarding the data that varies the least, we can maintain reasonably high recognition rates.

| | Percentiles | Variance |
|---|---|---|
| FRGC | 84.25% | $\approx$ 79% |
| Photoface | 89% | $\approx$ 92% |
| Processing time | 800.64s | 180.95s |

Table 2: A comparison of recognition performance using percentiles and variance methods to select the most discriminatory 10% of the data. The processing time includes the calculation of the outliers/most varying pixels and 400 classifications

The processing time improvement for the variance approach is due to having decreased the vector size by 90 %. This compares to 973.09s for the equivalent Fisherface analysis which provides an accuracy of 99.5% so both methods offer considerable time savings at a small cost to accuracy.

### 3.3 Optimisation of Image resolution

Finally the effect of image resolution on 3D recognition performance is shown in Fig. 8. This clearly shows that a resolution of $10 \times 10$ px provides optimal or close to optimal recognition performance (the result for $40 \times 40$ px is 0.25% higher for FRGC) on both 3D datasets. The same pattern appears in the 2D Photoface database, but there is a small decrease of just under 3% for the 2D FRGC data. Nonetheless, if we take the $10 \times 10$ px as an optiumum size, this figure is lower than often reported in the literature. This may be because the data used in these experiments is already highly cropped, and other research may be using other metrics such as the distance across the uncropped head. Although not shown in the figure, not antialiasing the resampled images led to poorer performance in all cases.

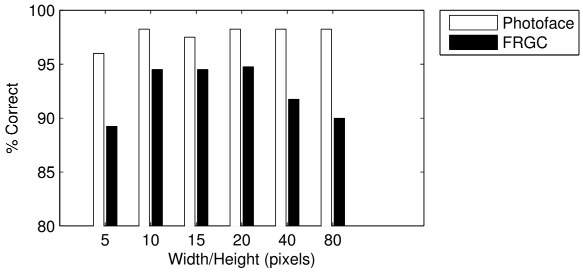


Figure 8: The effect of resolution on 3D recognition performance. Recognition rates for $10 \times 10$ px are 94.75% for FRGC data and 98.25% for Photoface data.

Combining the optimal resolution of $10 \times 10$ px with the variance method above we can achieve virtually the same recognition performance as an $80 \times 80$ px image but using only 64 pixels for FRGC data and 61 pixels for Photoface data. Recognition rates of 87.75% and 96.25% are recorded (a loss of only 7% and 2% respectively). The processing time is also reduced to 10.5s for variance analysis and 400 classifications. The same analysis using the Fisherface algorithm takes 118s and achieves a comparable rate of 89.25%.

## 4 Discussion

This paper describes methods to effectively reduce data dimensions while maintaining recognition performance. Computationally efficient methods using variance analysis and image resizing have been shown to be powerful means of reducing data but maintaining discriminatory information. Table 3 compares commonly used dimension reduction techniques of PCA and Fisherface with our variance and percentile inclusion criterion techniques at different resolutions in terms of classification accuracy and processing time. All experiments were carried out in Matlab on a Quad Core 2.5GHz Intel PC with 2GB ram running Windows XP. Only one percentile inclusion criterion result has been included as performance (especially processing time) was not at the same level as other conditions.

The number of components which are used for PCA depends on the specific test as follows: 61 components (61PCA, row 6 of Table 3) were chosen for a direct comparison

| | Res. (px) | Data Reduction | Classifier | No. Dimensions | % Correct | Proc. time(s) |
|---|---|---|---|---|---|---|
| 1. | 10x10 | None | PMCC | 200 | 98.25 | 12.02 |
| 2. | 10x10 | VI | PMCC | 19 (10%) | 82.75 | 12.52 |
| 3. | 10x10 | VI | PMCC | 61 | 95.75 | 13.02 |
| 4. | 10x10 | PCA | Euc. dist. | 21 | 94.5 | 92.47 |
| 5. | 10x10 | PCA | PMCC | 21 | 92.25 | 97.16 |
| 6. | 10x10 | 61PCA | Euc. dist. | 61 | 96.25% | 102.91 |
| 7. | 10x10 | VI → 15PCA | PMCC | 61 → 15 | 89.75 | 128.54 |
| 8. | 10x10 | VI → FF | Euc. dist. | 19 → 19 | 90.5 | 129.74 |
| 9. | 80x80 | None | PMCC | 12800 | 98.25 | 129.86 |
| 10. | 10x10 | VI → 15PCA | PMCC | 19 (10%) → 15 | 79 | 132.56 |
| 11. | 10x10 | VI → FF | Euc. dist. | 61 → 39 | 99 | 134.69 |
| 12. | 10x10 | FF | Euc. dist. | 39 | 100 | 144.25 |
| 13. | 80x80 | VI | PMCC | 1235 (10%) | 92.25 | 180.95 |
| 14. | 80x80 | VI → 15PCA | PMCC | 1235 (10%) → 15 | 85.25 | 331.40 |
| 15. | 80x80 | VI → FF | Euc. dist. | 1235 (10%) → 39 | 90.75 | 549.25 |
| 16. | 80x80 | PCA | Euc. dist. | 61 | 96.75 | 573.52 |
| 17. | 80x80 | PI | PMCC | 12800 | 89 | 800.64 |
| 18. | 80x80 | FF | Euc. dist. | 39 | 99.5 | 973.09 |

Table 3: A comparison of our variance (VI) and percentile (PI) inclusion techniques with PCA and Fisherface (FF) algorithms sorted by processing time.

with the 61 variables of the variance inclusion criterion which gave good performance in Fig. 7. 15 components (15PCA condition, rows 7, 10 & 14) were chosen arbitrarily as an extra step after the variance inclusion criterion for its low dimensionality and relatively good performance. For other tests using PCA, the number of components are chosen which describe 85% of the variance. Some entries in the “No. Dimensions” column have (10%) shown next to them. This is a reminder that only 10% of the data remains after applying the variance inclusion criterion. Finally some of the rows contain a “→” symbol representing a combination of processes eg Variance Inclusion followed by Fisherface.

Generally resizing the image to 10x10 pixels gives a clear processing time advantage with little or no compromise on accuracy. Without additional dimensionality reduction we achieve a recognition rate of 98.25% (row 1). We are able to reduce the dimensionality by a further $\frac{2}{3}$ and only lose 2.5% performance by additionally using the variance inclusion criterion to select 61 pixel locations (row 3). This appears to give the best compromise in terms of the number of dimensions, processing time and accuracy . The Fisherface algorithm gives excellent performance (10x10 Fisherface gives 100% accuracy, row 12) but at the cost of processing time.

These results only apply to the simplest case in face recognition – the frontal, expressionless face. The variance inclusion algorithm would be unlikely to produce similarly good results if expressions were present in the dataset, as these are likely to produce areas of high variance which will not be discriminatory. Nonetheless these could be used for the purposes of expression analysis instead of recognition or alternatively areas which change greatly with expression could be omitted from the variance inclusion criterion.

It is clear that effective dimensionality reduction can be achieved via more direct, psychologically inspired models in contrast to conventional mathematical tools such as PCA. Processing speed is also drastically increased – if we perform recognition by the Fisherface algorithm on $80 \times 80$ pixel images, it takes 973.09s. Using $10 \times 10$ pixel images, processing time drops to only 13.02s using our proposed variance inclusion method to extract 61 pixel locations with only a 3.75% drop in performance.

# 5 Conclusion

We have presented a number of important findings that affect face recognition performance regarding the effects of optimum image size and the use of different variance measures to select discriminatory data. The findings have implications on real-world applications in that they point to computationally attractive means of reducing the dimensionality of the data. Empirical support of Unnikrishnan's hypothesis [16] regarding the use of outlying percentile ranges is provided on both the FRGCv2.0 database as well as our own photometric stereo face database. One of the most promising results comes from resizing the original 3D data from 80x80 pixels to 10x10 pixels and applying the variance based inclusion approach yielding an accuracy of 95.75% using just 61 dimensions and the fact that this heuristic was inspired by the human process of caricaturing. Using this combination of techniques, processing speeds can be also be increased tenfold over the conventional Fisherface algorithm.

# References

[1] Simon Baker and Takeo Kanade. Limits on Super-Resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2002.1033210.

[2] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997. ISSN 0162-8828.

[3] B. J. Boom, G. M. Beumer, L. J. Spreeuwers, and R. N. J. Veldhuis. The effect of image resolution on the performance of a face recognition system. In *9th International Conference on Control, Automation, Robotics and Vision, 2006. ICARCV'06*, page 1–6, 2006.

[4] K. Chang, K. Bowyer, and P. Flynn. Face recognition using 2D and 3D facial data. In *ACM Workshop on Multimodal User Authentication*, pages 25–32, 2003.

[5] B. Gökberk, M. O. İrfanoğlu, and L. Akarun. 3D shape-based face representation and feature extraction for face recognition. *Image and Vision Computing*, 24(8):857–869, 2006.

[6] Mark F. Hansen, Gary A. Atkinson, Lyndon N. Smith, and Melvyn L. Smith. 3D face reconstructions from photometric stereo using near infrared and visible light. *Computer Vision and Image Understanding*, 114(8):942–951, August 2010. ISSN 1077-3142.

[7] M. Hüsken, M. Brauckmann, S. Gehlen, and C. Von der Malsburg. Strategies and benefits of fusion of 2D and 3D face recognition. In *IEEE workshop on face recognition grand challenge experiments*, page 174, 2005.

[8] B. Li, C. H Zheng, and D. S Huang. Locally linear discriminant embedding: An efficient method for face recognition. *Pattern Recognition*, 41(12):3813–3821, 2008. ISSN 0031-3203.

[9] Y. M Lui, D. Bolme, B. A Draper, J. R Beveridge, G. Givens, and P. J Phillips. A meta-analysis of face recognition covariates. In *Proceedings of the 3rd IEEE international conference on Biometrics: Theory, applications and systems*, page 139–146, 2009.

[10] R. Mauro and M. Kubovy. Caricature and face recognition. *Memory & Cognition*, 20 (4):433–440, 1992.

[11] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Proc. CVPR*, volume 1, 2005.

[12] G. Rhodes, S. Brennan, and S. Carey. Identification and ratings of caricatures: Implications for mental representations of faces. *Cognitive Psychology*, 19(4):473–497, 1987.

[13] Pawan Sinha, Benjamin Balas, Yuri Ostrovsky, and Richard Russell. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948–1962, November 2006. ISSN 0018-9219.

[14] George Toderici, Sean O'Malley, George Passalis, Theoharis Theoharis, and Ioannis Kakadiaris. Ethnicity- and gender-based subject retrieval using 3-D Face-Recognition techniques. *International Journal of Computer Vision*, 89(2):382–391, 2010. doi: 10.1007/s11263-009-0300-7.

[15] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[16] M.K. Unnikrishnan. How is the individuality of a face recognized? *Journal of Theoretical Biology*, 261(3):469–474, December 2009. ISSN 0022-5193. doi: 16/j.jtbi.2009.08.011.

[17] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Opt. Eng.*, 19(1):139–144, 1980.

[18] J. Yang, J. Wright, T. S Huang, and Y. Ma. Image super-resolution via sparse representation. *Image Processing, IEEE Transactions on*, 19(11):2861–2873, 2010. ISSN 1057-7149.

[19] Jian Yang, David Zhang, Alejandro F. Frangi, and Jing-yu Yang. Two-Dimensional PCA: a new approach to Appearance-Based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2004.10004.

[20] M. H Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *fgr*, page 0215, 2002.

[21] J. Ye, R. Janardan, Q. Li, et al. Two-dimensional linear discriminant analysis. *Advances in Neural Information Processing Systems*, 17:1569–1576, 2004.

[22] Stefanos Zafeiriou, Mark F. Hansen, Gary A. Atkinson, V. Argyriou, Maria Petrou, M.L. Smith, and L.N. Smith. The PhotoFace database. In *Proc. Biometrics Workshop of Computer Vision and Pattern Recognition*, pages 161–168, Colorado Springs, USA, 2011. IEEE.

[23] Y. Zhuang, J. Zhang, and F. Wu. Hallucinating faces: LPH super-resolution and neighbor reconstruction for residue compensation. *Pattern Recognition*, 40(11):3178–3194, 2007. ISSN 0031-3203.

Proc. of the 3rd British Machine Vision UK Student Workshop (BMVC'11 WS, Dundee, Scotland, 2nd September, 2011)

# Recognition of Everyday Domestic Activities Using a Depth Sensor

Lulu Chen
l.chen@pgr.reading.ac.uk
Hong Wei
h.wei@reading.ac.uk
James Ferryman
j.m.ferryman@reading.ac.uk

Computational Vision Group
University of Reading
UK

**Abstract**

For future domestic integrated monitoring systems, we propose an approach that uses both spatial and temporal properties to describe an action using the data now available from commodity depth sensors. It is applicable to various domestic applications, such as home care of frail elderly people who live alone. The novelty is in building on readily available software algorithms and hardware devices to produce a continuous stream of high-level generic events that could be exploited by application programs. Building on lower level vision algorithms, such as object detection and tracking, we focus on the higher level interpretation of human activities in terms of their interaction with and manipulation of other physical objects. We model an action using 3D spatial information from the depth image, and temporal reasoning. The method can achieve high recognition accuracy and is fast, so it could be run as a background process feeding events to any listeners that have subscribed. To illustrate the approach, we provide drinking as an example, as this can be an important activity for monitoring elderly people's health. We tested our method with real users and the results show a good confidence against different realistic room settings.

## 1 Introduction

The level of technology in the home continues to increase, and is a great help in our daily lives. Basic sensors such as those for temperature, water usage, passive infrared motion, and carbon monoxide, are being joined by smart electricity meters that perform continuous monitoring to encourage energy conservation. With the rapid development of camera technology, cameras have been getting smaller, cheaper, and lower power, so a computer vision system could be integrated along with lighting and other electrical items in the home.

We aim to develop a home monitoring system to interpret everyday activities. This could be used for many purposes, but would be particularly beneficial for the health care of elderly people who live alone. The system would provide them with interaction or assistance based on the understanding of their activities. For instance, after consultation with home care nurses, we learned that one of the major problems for elderly people is that they suffer dehydration because they do not remember to drink enough water. Thus, recognition of drinking activities is one feature that would provide useful information to care givers or even just as a reminder to the elderly people themselves. Our work mainly focuses on higher level human activity interpretation, with preconditions assuming some lower level image processing has been done.

Two big challenges for home monitoring are the complexity in the environment, which makes foreground extraction difficult, and the indoor lighting condition, which changes over time causing most feature extraction algorithms based on colour images to fail. Thus, instead of using conventional colour cameras, we propose to use a depth sensor, which generates depth information that is more stable and robust against indoor lighting changes. In the scene of a home environment, we assume the objects of interest have been identified and tracked (including a person and relevant objects). Therefore, 3D motion trajectories of these objects can be estimated from the depth information. Then distances between the objects, which indicate interaction and manipulation, can be calculated using their 3D coordinates. Actions are then detected using a spatio-temporal reasoning method.

The rest of this paper is organised as follows. Recently new depth sensing technology has made it cheap and robust to obtain 3D positions, especially with the appearance of Microsoft Kinect. A background introduction of depth sensors along with related work on activity recognition based on vision systems is provided (Section 2). Objects of interest can be detected and tracked in 3D using ambient sensors, and actions of interest are modelled and recognised based on the spatial 3D information provided from the depth sensor and temporal interval reasoning (Section 3). As an example, we analysed the performance with five people who each performed a drinking action. Results and discussion are presented (Section 4). We will apply this method to other activities and further additions are proposed (Section 5).

## 2 Background

Our proposed algorithm addresses the general area of human activity recognition (Section 2.1) and in particular it processes spatial and temporal trajectories (Section 2.2). Accurate trajectories can now be obtained due to advances in depth sensors (Section 2.3).

### 2.1 Human Activity Recognition

Human activity recognition has a wide range of potential real-life applications to help improve human lives, including the areas of visual surveillance, human-computer interaction, game controls, sports video analysis, virtual reality and video retrieval. Activity recognition is usually based on computer vision because it is non-intrusive, but extracting the salient events from all the data is a challenge. To recognise daily activities at home can be even more complex, due to the large variety of room settings, furniture occlusion, complexity of the interactions between humans and their environment, and strong diversities of characteristic human behaviour. Since action recognition has been widely studied in computer vision, much work has been done related to human action recognition and significant progress has been seen in recent years [5, 19].

However, previous work focussed on home activity monitoring and everyday activity interpretation has not been gained much attention. There are a few works related to fall detection addressing the problem of falling down faced by elderly people [11, 12, 21]. Fall detection based on vision is helpful for elderly people's care, because users do not need to wear sensors on their bodies. But, falls are not the only problem for elderly people. Snoek et al. [14] presented an approach for assessing the usability of a faucet for elderly people that combines vision and audio processing. We are developing a method for monitoring users' everyday activities, to contribute to the development of home care.

## 2.2 Actions as Spatial and Temporal Trajectories

Action modelling is mainly based on spatial and temporal cues. Spatial trajectory based on interest point or object tracking has been an approach to describe actions. Velocity, direction and maxima of the trajectory curvature have been commonly applied [9, 11, 21]. However, a single trajectory only presents motion cues, rather than providing any spatial information such as shapes [22]. In recent literature, some methods presented template modelling by directly extracting a space-time shape from a 3D spatial-temporal space $(x,y,t)$.

The original work on motion history image (MHI) presented by Bobick and Davis [2] has been referenced and built upon in various ways by other work. A popular extension of MHI introduced by Weinland et al. [18] is motion history volumes (MHVs) using visual hulls instead of silhouettes in MHI. More recent work uses space-time volume built by accumulating torso silhouettes over time [4, 22] instead of using multiple views. These methods can successfully construct shapes of actions, however, due to large variations in body shapes and individual action styles the model classification can be very difficult. Moreover, the applied actions cannot be complex and do not allow much obstruction of body parts by each other, so they are not practical for real-life action interpretation. Our method is inspired by these methods. Instead of using a conventional colour image and trying to extract features and build shapes using 2D features, we use 3D location information captured from a depth sensor with temporal reasoning built on top, which can be simply applied to realistic everyday domestic actions.

## 2.3 Depth Sensors

Depth map acquisition based on the principle of stereo vision as an important computer vision research field can be dated back to 1960s. Since then much research has contributed to calculating a depth map from stereo views. In fact, due to the complexities in the stereo geometry calculation, camera systems, and correspondence matching, reconstructing the depth map still remains a challenge in computer vision. This makes the depth map reconstructed from stereo vision still impractical for use in real-time and in everyday life. In the past a few years, depth sensing has drawn great attention from researchers, and commercial development has made progress. Time-of-Flight (ToF) cameras were made available by a few companies for research use [8, 15]. However, according to their very high price and limited imaging range and resolution, ToF cameras cannot currently contribute to everyday life applications. Schwarz et al. [13] presented a method using ToF camera to track human poses and recognise activities. Their method requires a prior motion model which is composed of a set of low-dimensional manifold embeddings. In 2010, Microsoft released an imaging device called Kinect [17], which is affordably priced for normal people's use, and which indicates the possibilities of everyday applications based on depth sensors. The Kinect device include an RGB camera and a depth sensor, which consists of an infrared laser projector combined with monochrome image sensor. The infrared laser projector projects a fixed infrared structured light pattern of dots, which is captured by the image sensor. The correspondences between projector and sensor are used to reconstruct the depth of the scene.

Unlike a conventional RGB colour image, pixel values in a depth image indicate calibrated distance between camera and scene, which provides a so-called 2.5-dimensional version of the scene: the 3D location of points visible to both the projector and the camera is known, but nothing is known about the parts of the scene that are not visible. Given depth information, background subtraction, one of the main subjects in computer vision, can be simplified, and contour detection within certain distance ranges becomes easier. Since the low-cost depth sensor's release, the term Natural User Interface (NUI) for user interfaces based on hand gesture recognition has rapidly become popular [20]. In this work, we use Kinect as the capture device, whose depth sensor gives a 640 by 480 resolution image at 30 frames/sec. The depth range is around 0.8 m to 3.5 m, with a resolution of about 1 cm [10].

# 3 Proposed Algorithm

Our algorithm has three steps, which are described in the sections below, and runs in real time on a conventional home computer to provide continuous monitoring. Objects of interest are identified and tracked, the distances between them are repeatedly calculated in a device-independent way, and spatio-temporal reasoning is applied to create a stream of application-related events.

## 3.1 3D Trajectory

Data in a depth map represents the 3D locations $(x,y,z)$ in the scene. With this, understanding human actions will not be limited to the 2D feature world, but performed in the real 3D space, which provides richer and more useful information. In this work we only focus on higher level action recognition, so we assume lower level processes, such as feature detection, object detection and tracking, have been done. We also assume the detection and tracking is implemented using just the colour images. Images from the colour camera and depth sensor have been registered, therefore the position of the object tracking results from the colour image can be given in the depth image. From the results of tracking, 3D trajectories of the objects of interest can be obtained from depth map. For instance, in a drinking action, we focus on three objects: mouth, hand and cup.

Figure 1 presents four frames extracted from a video sequence containing a drinking action. The images on the top row are from the colour camera, and those on the bottom row are the corresponding frames from the depth sensor. In the first colour image the objects of interest are highlighted: face (red), hand (blue) and cup (green). The depth and RGB cameras in general will be at different positions, but their output can be registered to provide RGB and depth values for one scene position at a single point in the images, by projecting one image into the plane of the other. Then the corresponding locations can be drawn on the depth images (shown on the depth images with the same colours) based on co-registration of the two images. The dashed lines on the depth images demonstrate the movement of the objects. We can imagine the relationship change between them over time. Figure 2 is a 3D plot of these trajectories. To represent each object we choose a single centre point for it. A complete drinking action includes the the person picking up the cup, drinking, and putting the cup back down.

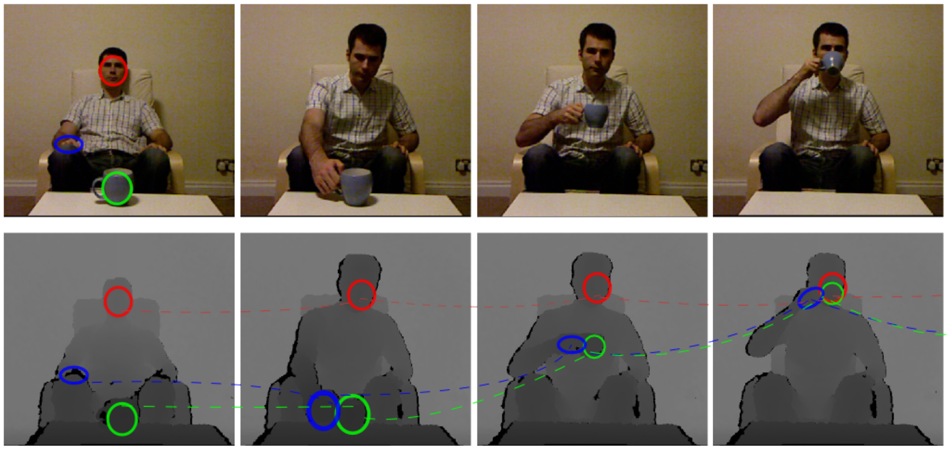


Figure 1: A drinking action from one experimental participant. Top row: four frames from colour camera. Bottom row: corresponding frames from depth sensor. Coloured lines demonstrate the movement of three objects: face, hand and cup.

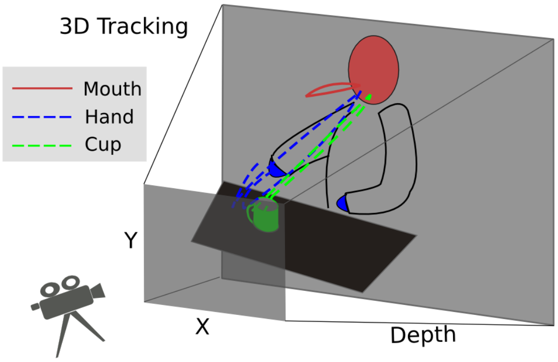
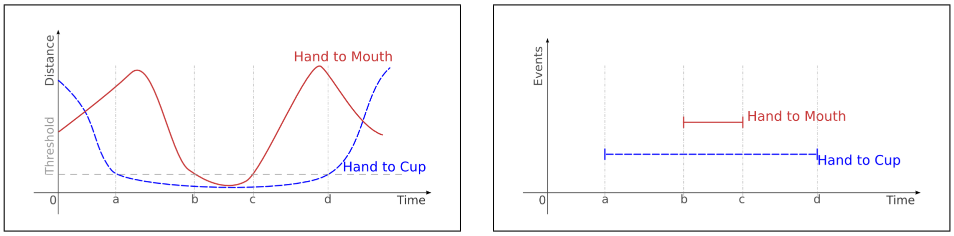


Figure 2: 3D plot of trajectories of tracked objects.

## 3.2 Distance Measurement

Everyday activities involve interaction and manipulation from humans to other physical objects. The aim of this work is to interpret these interactions. Trajectories from object tracking only contain position values to model an action, but the logical relationship between objects is also useful, and easier to process at higher levels of abstraction. To represent the relationship between objects' motions, we apply a distance measurement based on their trajectories. From 3D trajectories obtained from the depth map, we can calculate the relative distance between each pair of objects over time. We calculate the relationship between each pair of objects in the 3D coordinates using a simple Euclidean distance measurement, which is updated in real time at the frame rate of the sensor. Figure 3(a) illustrates the distances using a drinking action as an example. The red continuous line in Figure 3 shows the hand-to-mouth distance over time, and the blue dashed line shows the hand-to-cup distance.

The spatial representation we obtain is the distances between objects. To convert the re-



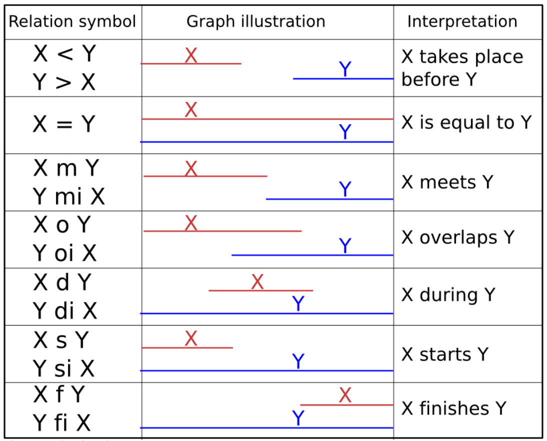
(a) (b)

Figure 3: Distance between objects (a) and event intervals (b).

lationship into logical events, a threshold is applied (the gray horizontal line in Figure 3(a)). The value of the threshold is application-dependent, that is, it is set based on the application needs using an actual measurement in metres. This distance is hardware-independent, which is possible because the depth map provides actual 3D positions in metres relative to the sensor position, and the distance calculation removes the effect of the arbitrary choice of world co-ordinate origin and axis directions. When the distance between two objects becomes smaller than the threshold, an event is considered to be happening. This binary decision is shown in Figure 3(b), which is an interval plot containing detected events. From a to d on the horizontal time axis of both plots show the correspondence between them. During interval a-to-d, the hand-to-cup distance is under the threshold, so an event is shown for that period in plot (b). Similarly, during interval b-to-c, the hand-to-mouth event is drawn in plot (b).

## 3.3 Spatio-Temperal Reasoning

To model and combine actions we apply a temporal reasoning method. Allen's Interval Algebra operators introduced by James F. Allen describe the possible relations between time intervals, which can be used as a basis for reasoning about temporal descriptions of events [1]. There are thirteen basic relation operators presented in his work (illustrated in Figure 4).



| Relation symbol | Graph illustration | Interpretation |
|---|---|---|
| X < Y<br>Y > X | X<br>Y | X takes place before Y |
| X = Y | X<br>Y | X is equal to Y |
| X m Y<br>Y mi X | X<br>Y | X meets Y |
| X o Y<br>Y oi X | X<br>Y | X overlaps Y |
| X d Y<br>Y di X | X<br>Y | X during Y |
| X s Y<br>Y si X | X<br>Y | X starts Y |
| X f Y<br>Y fi X | X<br>Y | X finishes Y |

i stands for inverse

Figure 4: The thirteen basic relationships between two intervals ($X$ and $Y$) in Allen's Interval Algebra [1].

In Figure 3(b) the red continuous line presents the interval of the hand-to-mouth event, and the blue dashed line presents the interval of the hand-to-cup event. The hand-to-mouth event happened during the hand-to-cup event, which follows Allen's Interval Algebra rule of 'during' ($X$ d $Y$). Drinking can be modelled as a combination of the two basic events, and therefore recognised as a new higher-level event. In other words, we can define the drinking action as follows:

$$\begin{aligned} \text{Event hand-to-mouth} &= M\,, \\ \text{Event hand-to-cup} &= C\,, \\ \text{Thus, Event drinking} &= D \quad = \quad M \,\mathrm{d}\, C\,. \end{aligned} \tag{1}$$

Using the basic relationships, events can be combined in different ways to create many other new events, which means that multiple events can be detected simultaneously. These could be more complex, and involve multiple objects, which we will consider in future work.

# 4 Experiment

In this section the method and lower-level algorithms we used are described, then the results are discussed.

## 4.1 Method

To analyse the performance of our approach, we used the action of drinking as an example, and tested it with five people. Each subject was asked to act a drinking action while in a sitting position. The scene was set up differently in each case, to test the robustness of the method. The aspects to be considered include: age, lighting condition, sitting distance from the camera and which hand was used for picking up the cup. Table 1 lists the situation of the subjects. The sensor used in the experiment was Microsoft Kinect, which has a colour camera and a depth sensor. The registration between the colour and depth images was done using OpenNI API framework [6] based on the device intrinsic parametres. Face detection was implemented based on Viola-Jones object detection algorithm [16]. The position of the mouth was defined to be one third up from the bottom-centre of the bounding box of the face. The cup and hand tracking was implemented using the Continuously Adaptive Mean Shift (CamShift) algorithm [3]. The distance threshold for both hand-to-mouth and hand-to-cup was set at 10 centimetres.

| Subject | Age | Day/Night | Sitting Distance | Right/Left Hand |
|---|---|---|---|---|
| 1 | 20s | daylight | 1.5m | left |
| 2 | 30s | daylight | 2.1m | right |
| 3 | 30s | indoor lighting plus sunlight | 2.1m | right |
| 4 | 50s | indoor lighting | 1.7m | left |
| 5 | 60s | indoor lighting | 1.7m | right |

Table 1: Experimental subjects.

### 4.2 Results and Discussion

Figures 5, 6 and 7 show the results from three of the five subjects. In each figure, the top plot shows the distances over time, and the bottom plot show the event intervals based on the threshold. In Figure 5, the subject performed one drinking action including picking up the cup, drinking, and putting down the cup. From the distance plot, we can see that hand-to-mouth becomes small during hand-to-cup being small. After applying the threshold the event intervals are calculated, and they are shown on the bottom plot. This drinking action happened in the anticipated way, with hand-to-mouth happening during hand-to-cup.

Another subject acted rather differently in the experiment (Figure 6). At the beginning the subject attempted to pick up the cup twice, but did not perform the drinking. The third time, the subject actually completed a drinking action. In the interval plot, we can see that the hand-to-cup event happened three times, and only on the third time did the hand-to-mouth event happen during it.

In real cases, it is common to see people have multiple sips before they put down the cup, and this is what happened in this case. Figure 7 shows a very clear distance change of hand-to-mouth, with it twice becoming small while the hand-to-cup distance remains under the threshold. The result from the interval plot at the bottom indicates the accurate recognition of two drinking events within one hand-to-cup event.

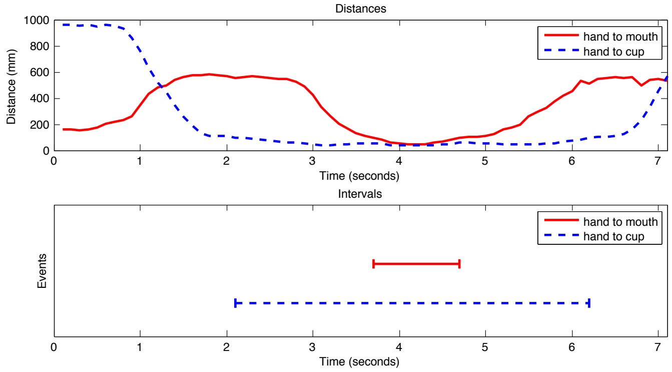


Figure 5: Subject 1 results: distances (top) and intervals (bottom).

During all the experiments, the object tracking was accurate and the interest objects were tracked from the beginning to the end. However, sometimes the tracking could be lost due to the robustness of the tracking algorithm. If tracking of an object is lost, we simply define the state of all events that rely on that object's position to be unknown. It is then left to the software that is receiving those events to decide what to do with that information. The camera was all set up in front of the person while the person was sitting on a chair during the experiments. However, the camera angle can be changed because as long object tracking is maintained, a different viewpoint will not affect the distance measurement or spatio-termporal reasoning from depth information.

## 5 Conclusion and Future Work

Depth images provide robust location information for objects in 3D space, and sensors are becoming cheap, which allows everyday monitoring applications with useful features at low

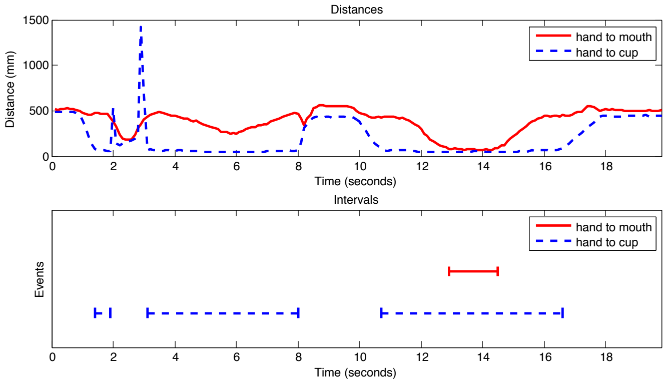


Figure 6: Subject 2 results: distances (top) and intervals (bottom).

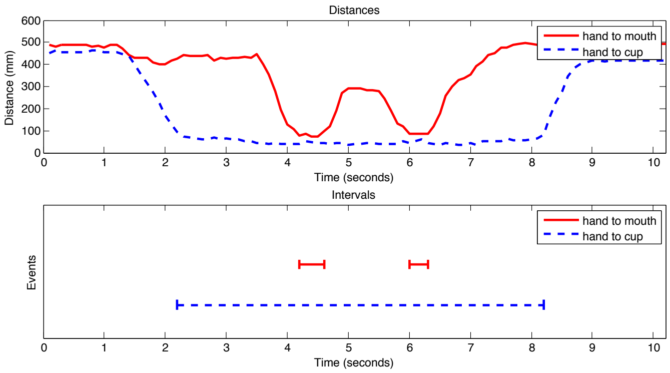


Figure 7: Subject 3 results: distances (top) and intervals (bottom).

cost. We propose a home monitoring method for describing and recognising human daily activities based on spatial (distance measurement) and temporal (interval algebra) cues. We tested the method using a drinking action with real users against different indoor settings, in fact the method can be applied to various indoor activities, such as reading newspapers, using TV remote control, sitting down and using a phone.

The main contribution of this work is that we employ vision-based human action recognition in a home monitoring system, targeting real use in daily life, particularly for elderly people's home care, where the computer vision literature is lacking. Also, the use of a depth sensor provides 3D information that has a great advantage over conventional colour cameras alone. We intend our continual ambient home monitoring system to output a stream of high level events. Application programs would then use a publish/subscribe system to receive the events and produce useful services for users, such as activity logs or alerts.

We have demonstrated the proposed method by detecting drinking. It can be easily applied to other activities, especially when there are multiple events involving multiple objects happening interactively and simultaneously. For instance, a person sitting on a chair by a table drinking tea while reading a paper. The method is expected to handle this complexity. A previous work demonstrated a temporal network structure to efficiently respond to multiple events in interval algebra [7].

In future work, we will test with more subjects, and especially with elderly people. The test will not be limited to drinking events, but various indoor activities. Additionally, the current software only tracks one hand at a time, so we will track both hands simultaneously and accept events from either one. Currently we use a simple threshold to convert distances to intervals, but we will investigate hysteresis thresholding to avoid any instability if a distance lingers close to the threshold. The relationship description between objects is not limited to Euclidean distance, and in future work other distance measures could be applied to different action events. Detection and tracking algorithms are conventionally applied to RGB or greyscale images, and this is what we have implemented. However, we are investigating applying or adapting these algorithms for use on depth images or a combination of RGB and depth images. In the long term, our proposed monitoring system could be considered as a component of infrastructure in the home, and implemented as a service oriented architecture. Other assistant services would be built on top of it, for the inhabitant(s) of the home, care givers, home automation or robotics.

## References

[1] James Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26 (11):832–843, 1983.

[2] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Inelligence*, 23:257–267, March 2001.

[3] Gary R. Bradski. Computer vision face tracking for use in a perceptual user interface, 1998.

[4] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:2247–2253, December 2007.

[5] TB Moeslund, A Hilton, and V Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[6] OpenNI. http://www.openni.org/, accessed March 2011.

[7] C.S. Pinhanez and A.F. Bobick. Human action detection using pnf propagation of temporal constraints. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 898 –904, June 1998.

[8] PMDTechnologies Products. http://www.pmdtec.com/, accessed March 2011.

[9] Cen Rao, Alper Yilmaz, and Mubarak Shah. View-invariant representation and recognition of actions. *Int. J. Comput. Vision*, 50:203–226, November 2002.

[10] PrimeSense reference design. http://www.primesense.com/?p=514, accessed March 2011.

[11] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Monocular 3D Head Tracking to Detect Falls of Elderly People. *Proceedings of the 28th IEEE EMBS Annual International Conference*, 1:6384–6387, 2006.

[12] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Fall detection from human shape and motion history using video surveillance. In *21st International Conference on AINAW*, volume 2, pages 875–880, 2007.

[13] L. A. Schwarz, D. Mateus, V. Castaneda, and N. Nava. Manifold learning for tof-based human body tracking and activity recognition. In *British Machine Vision Conference (BMVC)*, August 2010.

[14] J. Snoek, B. Taati, Y. Eskin, and A. Mihailidis. Automatic segmentation of video to aid the study of faucet usability for older adults. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 63 –70, June 2010.

[15] Mesa Imaging SwissRanger SR4000. http://www.mesa-imaging.ch/prodview4k.php, accessed March 2011.

[16] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.

[17] Microsoft Corp. Redmond WA. Kinect for xbox 360.

[18] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Comput. Vis. Image Underst*, 104(2):249–257, 2006.

[19] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Compouter Vision and Image Understanding*, 115(2):224–241, 2010.

[20] Daniel Wigdor and Dennis Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann, 2011.

[21] G. Wu. Distinguishing fall activities from normal activities by velocity characteristics. *Journal of Biomechanics*, 33(11):1497–1500, 2000.

[22] Alper Yilmaz and Mubarak Shah. Actions sketch: A novel action representation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1 of *CVPR*, pages 984–989, Washington, DC, USA, 2005.

# Motion Detection in Moving Transport

Sriram Varadarajan
Queen's University Belfast

Paul Miller
Queen's University Belfast

Huiyu Zhou
Queen's University Belfast

Jianguo Zhang
University of Dundee

**Abstract**

The aim of this paper is to evaluate the performances of existing motion detection algorithms on a moving transport where there exists the main problem of a large dynamic background. This paper provides a comparison of four well known background subtraction algorithms on a bus sequence. The results show that these conventional methods are not really suitable for this scenario and the analysis is provided with explanations for each algorithm. A few suggestions have also been proposed to minimise this problem.

## 1 Introduction

Detecting moving objects in a scene is one of the most fundamental and important tasks in video surveillance. There are many challenges that one may encounter while performing this task. The difficulty of the task may vary depending on a lot of factors such as the environment, lighting conditions, weather conditions, complexity of the scene etc. The surveillance could be taking place in an indoor or an outdoor environment. The system has to adapt to gradual illumination changes caused by the time of the day and also some sudden change in lighting. The scene could vary from a very simple one with a homogeneous background, to a complex scene with a cluttered background. There could also be some occlusions of the objects being observed. In complex scenes, there might be specific objects that should be considered as a background even though they are moving. These could be shadows of the actual objects or branches of trees swaying in the wind. These types of objects are referred to as dynamic background. The algorithm should be able to filter out these objects while segmenting the foreground.

In this paper, the scene under study is a moving transport carriage such as a bus. This type of scenario is very complex because of the design of the bus. There are various challenges associated with this scene. There could be sudden and extreme lighting changes when the bus moves into or comes out of a subway. There would also be constant changes in the direction of light falling onto the bus based on the direction in which the bus is moving and also the time of the day. However, the main problem in this scene is the moving background outside the bus that is observed through the windows. When the bus is in motion, the scene outside is random and constantly changing. The observed objects could be trees, buildings, sign posts, other vehicles or even people walking by the side of the road. These form a significant portion of the frame captured by the camera. They pose the real test in this problem as it is extremely difficult to isolate and eliminate this moving background from the scene. The window region cannot be masked out of the frame physically as a person inside the bus standing in front of the window would be masked out as well. The different algorithms are briefly explained in section 2 of the paper. Section 3 deals with the experiments and the results followed by discussion and conclusion in the remaining sections of the paper.

# 2 Methodology

The different methods used in the experiments are discussed in this Section. A simple yet efficient method for background subtraction with respect to a median background [1] is given in Subsection 2.1. An adaptive Gaussian Mixture Model technique is presented in Subsection 2.2. This method is based on Zivkovic's publication [2]. In Subsection 2.3, an approach based on Eigenbackground by Oliver et al. [3] is described. Subsection 2.4 outlines a method based on Colour Difference Histogram [4].

## 2.1 Adaptive Median method

This is one of the simplest methods of background subtraction where the background model is the median of N frames. The background estimate is the median at each pixel location over the sampling size. A recursive filter is used to estimate the median by using the following update equation where B denotes the median background and I denotes the current image:

$$B_{t+1}^{c} = \begin{cases} B_t^c + 1 & if\ I_t^c > B_t^c \\ B_t^c - 1 & if\ I_t^c < B_t^c \\ B_t^c & if\ I_t^c = B_t^c \end{cases} \tag{1}$$

The pixel classification is done with the help of a threshold T.

$$|I_t(x,y) - B_t(x,y)| > T \tag{2}$$

This method is computationally efficient, simple and robust to noise.

## 2.2 Adaptive Mixture of Gaussians

The Gaussian Mixture Model (GMM) is one of the most widely used approaches in the field of background subtraction. Each pixel in the image is modelled by a probability density function. The foreground pixels are then detected based on the variance in the pixel intensity level. Usually in Gaussian Mixture Model based approach, the number of components for each Gaussian is fixed. This approach is an adaptive form of GMM where the number of components of the mixture is varied for each pixel in an on-line procedure [2]. The pixel distributions are updated depending on a time period T. The GMM with K components is given is

$$\hat{p}(\vec{x}|\chi_T, BG + FG) = \sum_{i=1}^{K} \hat{\pi}_i * \eta(\vec{x}; \widehat{\vec{\mu}_i}, \hat{\sigma}_i^2 I) \tag{3}$$

where K is the number of Gaussian distributions, $\widehat{\vec{\mu}_i}$ are the estimates of the means and $\hat{\sigma}_i$ are the estimates of the variances of the Gaussian components. New data samples are added by updating the weight, mean and variance with the help of a constant $\alpha$ which is assumed to be 1/T. The component with the largest weight $\hat{\pi}_i$ is set as the owner of the sample, if it is deemed to be 'close' enough. The closeness is calculated by using the Mahalanobis distance and checking if it is less than a threshold value [2]. The closest component is then set value of 1 while all the other components are set a value of 0. The background model can be approximated by using the first B largest clusters by

$$\hat{p}(\vec{x}|\chi_T, BG) = \sum_{i=1}^{B} \hat{\pi}_i * \eta(\vec{x}; \widehat{\vec{\mu_i}}, \hat{\sigma}_i^2 I) \tag{4}$$

B is calculated by first sorting the components in descending order of weights and then applying,

$$B = \arg\min_b (\sum_{i=1}^{b} \hat{\pi}_i > (1 - c_f)) \tag{5}$$

where $c_f$ is a measure of how long an object can remain as a foreground without affecting the background model.

## 2.3 Eigenbackground based approach

In this method, an eigenspace is built to model the background for segmentation. In order to build the eigenspace, principal component analysis (PCA) is used. This helps in reducing the dimensionality and hence the complexity of the problem. The background model is first considered for N images from which the mean background image $\mu_b$ and its covariance matrix $C_b$ are calculated. For performing PCA, this matrix has to be diagonalised by

$$L_b = \Phi_b C_b \Phi_b^T \tag{6}$$

where $\Phi_b$ is the eigenvector matrix of the covariance and $L_b$ is the corresponding diagonal matrix of its eigenvalues.

While performing PCA, only the M largest eigenvalues are used to reduce the dimensionality of them. Therefore, only M eigenvectors are used to get a $\Phi_M$ matrix. The values used for modelling the background are stored in a feature vector form created by $I_i - \Phi_{M_b}^T X_i$ where $X_i = I_i - \mu_b$ is the mean normalised vector form of the image.

This eigenspace effectively models the probability distribution function of the background and so, the moving objects do not provide significant information to this model. This means that the actual background alone can be obtained from the sum of the eigenvectors.

Once the background model is obtained, the Euclidean distance between the eigenbackground image $B_i = \Phi_{M_b} X_i$ and the input image $I_i$ is calculated as

$$D_i = |I_i - B_i| > T \tag{7}$$

This distance is compared to a threshold $t$ and the moving objects are segmented when this distance is above the threshold.

## 2.4 Colour Difference Histogram based segmentation

This method involves background extraction by clustering of each pixel over a sequence of images by using their colour difference. Then the minimum distance is found by

$$D_i(x, y, m) = \min_{\forall n_{xy}} D_i\left(x, y, n_{xy}\right), \qquad where\ m \subset n_{xy} \tag{8}$$

The minimum distance is then compared with a threshold to obtain a classification of the pixel into that cluster. If the classification is not possible, it is set into a new cluster and the values are updated and the number of pixels in the new cluster is set to 1.

$$\begin{gathered} n_{xy} = n_{xy} + 1 \\ RGB\left(x, y, n_{xy}\right) = RGB_i(x, y) \\ C\left(x, y, n_{xy}\right) = 1 \end{gathered} \tag{9}$$

where $n_{xy}$ is the cluster number at the pixel (x,y) and RGB(x,y,$n_{xy}$) denotes the red, green and blue values of the current pixel (x,y) in the cluster $n_{xy}$. Once all the background pixels have been extracted, the background model is then used for segmentation of the foreground.

The segmentation is based on the colour difference histograms for all three colours Red, Green and Blue between the input image and the extracted background image. According to the authors, these histograms resemble a Gaussian distribution with zero mean with the foreground pixels showing a variance from the mean value. Hence, first turning points for the three colours are used to obtain a threshold value. A turning point in the histogram is defined as a point where there are maxima or minima.

$$\begin{gathered} If\ H_{i_{sRGB}}[k-1] \geq\ H_{i_{sRGB}}[k]\ and \\ H_{i_{sRGB}}[k] \leq\ H_{i_{sRGB}}[k+1], \\ HRGB_i = k \end{gathered} \quad (10)$$

where k ranges from 1 to 252.

This threshold value is used to compare the colour difference image of the next input image and the background image. If the difference is more than the threshold, then the corresponding pixel is a foreground object. Otherwise, it is considered a background.

# 3 Experiments and Analysis

The above algorithms were evaluated on a video sequence captured from a fixed camera inside a bus. The sequence has the bus at rest initially, and then the empty bus moving for a while and finally a person enters the bus and sits in one of the seats at the front of the bus. It is 1000 frames in length at a frame rate of 29 frames per second. Each frame size is 320x240 pixels. The ground truth of the bus was manually labelled for each of the 1000 frames for comparison with the results of the algorithms. Since pixel level segmentation is under consideration, this process is generally time consuming as the outline of the person changes slightly every four or five frames on an average due to the capturing limitations of the camera.

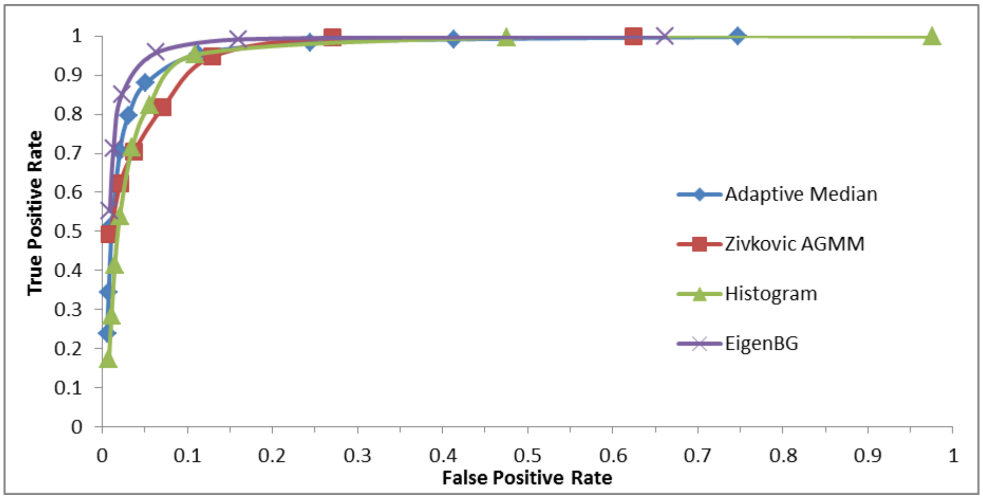


Figure 1: ROC plots for the four algorithms.

The Receiver Operating Characteristic (ROC) plot for the four algorithms is shown in Figure 1. This shows the plot of the false positive rate or 1-specificity versus the true positive rate or the sensitivity. It is interesting to note that all the algorithms are inclined towards the perfect classification side (FPR-0, TPR-1). This would imply that these algorithms seem to work fine with the sequence. But, when viewing the outputs qualitatively, it could be seen that the moving background problem persists in all the algorithms. The reason the false positive rates are so low for high true positive rates is that the number of true negatives is very high compared to the number of false positives. This is illustrated in the example below.

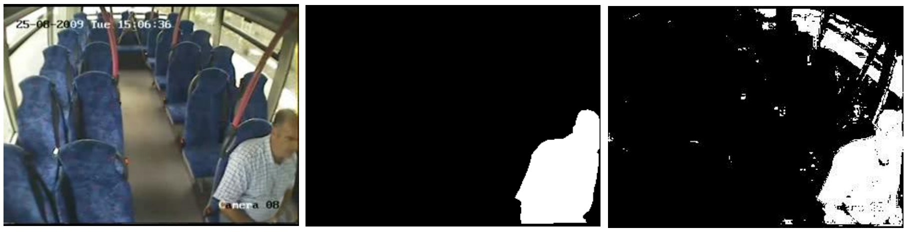


Figure 2: Sample Frame, Ground Truth and Typical Segmentation Output

In figure 2, the False Positive Rate (FPR) is FP/(FP+TN), which is 0.0611. The True Positive Rate (TPR) is TP/(TP+FN), which is 0.9547. Hence, although the FPR seems to be very low (around 6%) for a high TPR (around 95%), the dynamic background can be seen quite clearly here because FP forms 5.5% of the image (4233/76800) and TP forms 9.5% of the image (7274/76800). In other datasets, it is noticed that the dynamic background does not contribute to even 1% of the image. Even in cases where it is around 5% of the image, the true positive pixels form more than at least 30% of the image and hence the false positive pixels do not form a significant part of the segmented frame. This can be inferred from the standard datasets available. This difference makes this problem a challenging one.

Another way to analyse the algorithms is by using Recall-Precision plot which is shown for the different algorithms in Figure 3. These plots compare the Recall of the algorithm which is given by the True Positive Rate and the Precision of the algorithm which is the ratio of True Positives (the person in this case) to the total number of Positives (the pixels detected as foreground). A high recall implies that all the true positive pixels have been segmented properly, but a lot of false positives are also present in the result. With respect to the video sequence, it means that the person is completely segmented without any false negative pixels, but in addition to that, there are a lot of useless pixels also present in the segmented frame. The frame shown in figure 2 is actually an example of high recall. Higher precision means better relevance and more precise results, but may imply fewer results returned. This means, to achieve a high precision, all the pixels segmented must be true positive pixels, but this does not necessarily mean that all the true positive pixels would be segmented. There could be some pixels on the person missing but there should be very less or no false positives present like the moving background.

The ‘good’ corner of the plot is the top-right corner which has a high precision value for a high recall value. Towards this end, all the true positive pixels are segmented while the false positive pixels are discarded. From figure 3, it can be seen that the algorithms are not very accurate when they have a high recall, i.e. all the true positive values (in this case, the person) is present at lower thresholds, but at the same time, there are a large number of false positive values as well. This can also be seen in the Recall-Precision plots. The precision drops rapidly towards the higher end of the recall axis.

A particular thing to note is that none of the algorithms are able to attain anywhere close to 100% precision, even at low recall values. The highest is around the region of 85% with the Eigenbackground based approach. This is because of the ever presence of the moving background in all the results. A high precision should mean there should be no false positives in the segmented output. Since all the results have significant false positives, the precision is not very high.

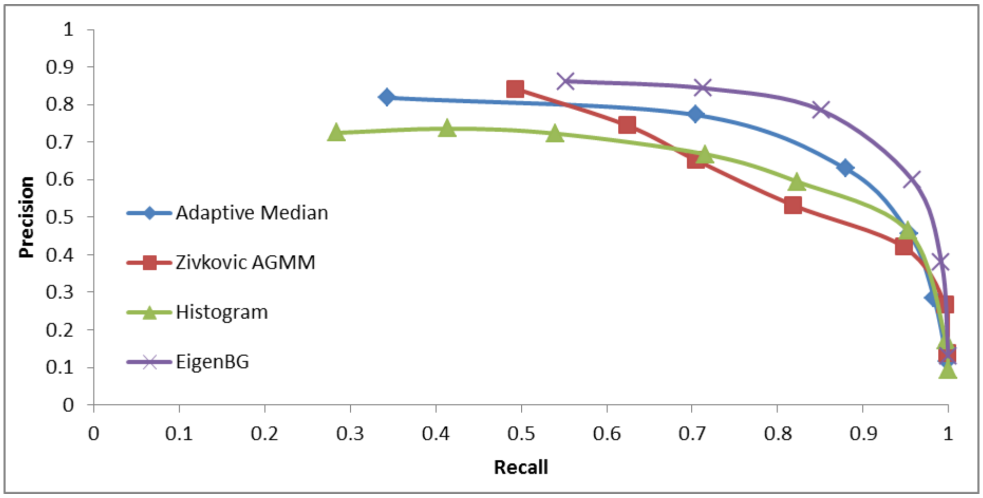


Figure 3: Precision Recall plots for the four algorithms.

The following part is the analysis of each algorithm against the video in detail. An example frame with the corresponding segmentation is shown for each algorithm in the image below. It might seem logical that the threshold corresponding to high true positive rate and low false positive rate is the best threshold. However, the important thing to consider here is a proper balance between the true positives and the false positives. It depends on whether a perfect segmentation is required or a noiseless output is required. At first glance, a threshold value producing TPR around 0.9 with FPR around 0.1 might seem the best choice. However, increasing the threshold value at the expense of TPR is not a bad choice. This is because we are looking to solve the moving background problem which forms the majority of the false positive pixels.

## 3.1 Adaptive Median method

The adaptive median method was experimented with various thresholds levels. As the threshold value increases, the number of positive detections decreases because more pixels fall in between the threshold value and the mean of the distribution.

It was observed that the adaptive median technique does not work really well with a considerable lighting change. This is because the variance of the pixels is not effectively modelled in adaptive median method. The seats are segmented as foreground here because the colour intensity level has just changed and the simple median background is not able to track this change properly as can be seen in figure 4. There are some false negatives as the algorithm has yet to adapt to the lighting change completely. These false negatives could be reduced further by using a connected components algorithm and filling the missing pixels.

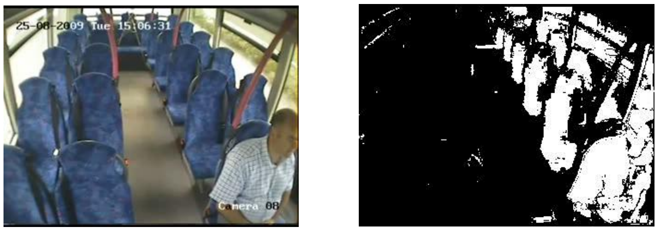


Figure 4: Example segmented frame output for Adaptive Median method.

## 3.2 Adaptive Mixture of Gaussians

Seven different threshold values were tried for the adaptive Mixture of Gaussians method. These threshold values are compared with the Mahalanobis distance value to determine the closeness of the pixel with the components. Here again, as the threshold value increases, the number of positives decreases as many pixels would fall within the threshold value and get subtracted as background.

From the outputs of the Mixture of Gaussians approach, it was observed that the shadow problem still exists. This method, however, adapts to a sudden lighting change in a much better way than the adaptive median method. The different Gaussians for each pixel effectively model the pixel distribution under different lighting conditions. Therefore, the segmentation process is not affected by the lighting changes in the sequence.

A main problem of this method is that the foreground starts to move into background after a short while. This can be seen in figure 5. In general motion segmentation scenes, a foreground object can be considered to become a background object once it remains static after a fixed amount of time. In this case however, the subject has to remain a foreground object as long as he remains in the view of the camera. This problem can be reduced while updating the background model by adjusting the value of $c_f$ and $\alpha$.

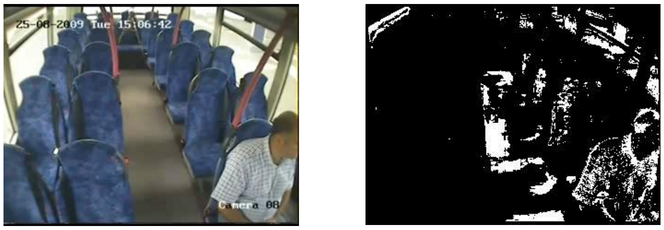


Figure 5: Example segmented frame output for Adaptive Mixture of Gaussians method.

## 3.3 Eigenbackground based Approach

In the Eigenbackground based approach, the threshold value is used to compare the Euclidean distance between the Eigenbackground image and the input image. Six different thresholds were used for this approach and the results are shown in the ROC plot in figure 1.

The Eigenbackground approach results show that the algorithm adapt well to lighting conditions as there are very few false positives inside the bus throughout the length of the sequence. The moving background problem exists here as well although it is not very pronounced compared to the previous two algorithms. At higher thresholds, the number of false positives (window region) is lower compared to the other algorithms; also there is a high sensitivity. The Eigenbackground approach based on eigenvalue decomposition effectively models the different illumination changes, either global or local. This is because the Eigenbackground is basically an image that is formed by combining background frames under various lighting conditions. This means any illumination change in the input frame is handled very well by the background model. This is illustrated in figure 6.

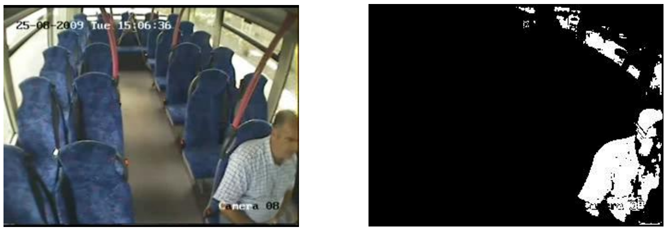


Figure 6: Example segmented frame output for Eigenbackground method.

### 3.4 Colour Difference Histogram based segmentation

The colour difference histogram based motion segmentation algorithm was tried with seven different thresholds ranging from very low positive values to very high positive values. It is similar to the Mixture of Gaussians approach in the sense that both the algorithms use clustering of pixels based on probability. Therefore, similar results to the Mixture of Gaussians approach can be seen here. The algorithm handles lighting changes very well; however, there are a large number of false negatives as time progresses. This could be corrected by updating the background model suitably, like in the case of the Adaptive Mixture of Gaussians approach. An example frame is shown in figure 7.

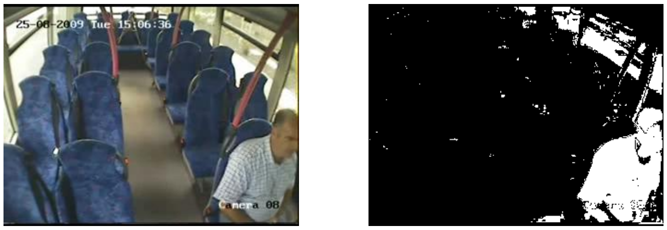


Figure 7: Example segmented frame output for Colour difference histogram based method.

## 4 Discussion

The moving background problem is seen to exist in all the approaches. This is because all these approaches are based on colour intensity values and the moving background also exhibits change in intensity in the same way as the foreground. The algorithms view the moving background in the same way as the foreground and hence, they are not able to effectively eliminate them. The algorithms, though different by definition are all similar approaches under a same framework. For instance, the Mixture of Gaussians and the Colour Difference Histogram approach perform pixel clustering depending on the variance of the pixels. The Eigenbackground approach uses PCA which is also used to model the distribution based on the variance. On the other hand, these algorithms are also contrasting in the sense that the MoG approach and the colour histogram approach build models for each pixel in the image whereas the Eigenbackground approach builds the model based on the variance in the entire image.

Out of the four algorithms, it can be easily seen the Eigenbackground based approach performs the best. This is inferred from the results, the ROC plots and the R-P plots. The adaptive median approach does not adapt well to the lighting changes, but removes the moving background fairly reasonably. The statistical approaches work well with lighting changes but fail to remove the moving background.

The Eigenbackground approach is still prone to errors due to the moving background problem. There is scope for improvement by adaptively varying the threshold along with the Eigenbackground model. The above approach uses a constant threshold throughout the sequence. Further, the threshold values in the window region could be updated in a manner different from the rest of the bus. This is possible because the pixel distribution in the window regions is different from the rest of the bus. The pixels in the window region mostly lie towards the white end of the colour map. Any moving object observed outside the bus is different in colour from the actual background and is usually darker. This means that the shift in the pixel distribution would be from 1 to 0 for example, if the colour model is expressed as a fraction with 1 representing white and 0 representing black.

Inside the bus, however, this shift would be different when there is a foreground object. This is because, in the original background, the window regions are bright in intensity compared to the internal structure of the bus. Therefore, the shift in the window region would be mostly from white towards black while the shift inside the bus would be either in the other way or there would be very little shift. In the case that there is a shift towards 1 inside the bus, this would still be different from the one occurring in the moving background region because the shift in the moving background would be starting at a much higher value compared to the one inside the bus. This information could be used while choosing the threshold for the Eigenbackground approach.

In a typical bus, the camera usually captures the fixed background and then the moving background without any foreground when the bus is empty initially. This moving background could be modelled by a learning algorithm initially and this model developed for the moving background could be used in conjunction with the Eigenbackground model for the fixed background to adaptively remove the moving background when there are foreground objects as well in the scene.

Another way to tackle this problem is to combine the Eigenbackground approach with a pixel concurrence technique similar to the one used by Seki et al [5]. This could be done by using image blocks and finding the correlation between neighbouring blocks. The spatial variations of the pixels could be combined with the temporal variation of the Eigenbackground approach to provide a better performance. These hypotheses would be experimented in the immediate future.

# 5 Conclusion

In this paper, four well known background subtraction algorithms have been evaluated with the bus sequence. These algorithms have been known in literature to work with common dynamic background video sequences. However, it could be seen from our results that these algorithms don't work very well with the bus sequence as the percentage of the moving background in the image is very large. The work in this paper is the initial step in developing a novel motion detection algorithm for moving transport and our future work is concerned with improving the accuracy of detection and reducing the false positive rate caused by the scene outside the window by experimenting on the hypotheses discussed above.

# References

[1] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, no. 3, pp. 187-193, 1995.

[2] Z. Zivkovic, "Improved adaptive Gausian mixture model for background subtraction," in *International Conference on Pattern Recognition*, 2004, pp. 28-31.

[3] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, August 2000.

[4] CC. Chiu, MY. Ku, and LW. Liang, "A Robust Object Segmentation System Using Probability-Based Background Extraction Algorithm," *IEEE Transactions on Circuits and Systems For Video Technology*, vol. 20, no. 4, April 2010.

[5] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, "Background Subtraction based on Cooccurrence of Image Variations," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

Proc. of the 3rd British Machine Vision UK Student Workshop (BMVC'11 WS, Dundee, Scotland, 2nd September, 2011)

# Fast, Approximate 3D Face Reconstruction from Multiple Views

Tim Rawlinson
tpr@dcs.warwick.ac.uk
Abhir Bhalerao
abhir@dcs.warwick.ac.uk

Department of Computer Science
University of Warwick
Coventry, UK

**Abstract**

3D face reconstruction is essential for a number of applications. Most approaches are aimed at producing accurate models and make use of high levels of control to achieve this, such as through restricted motion, structured light, or specialist equipment. However, there is also a considerable market for reconstructions in uncontrolled environments but where a lower level of accuracy is acceptable, for example, in avatar generation for games. It is this that we are trying to achieve, specifically: fast, approximate reconstruction using multiple views from a single low-cost camera in a largely uncontrolled environment.

We present a multi-step process combining: an extended Viola-Jones face detector and Active Appearance Models (AAM) for feature detection; Sparse Bundle Adjustment (SBA) for reconstruction; refinement; and texture generation, to produce a high-quality 3D model of a face. We describe the method in detail and present several visually convincing results.

## 1 Introduction

3D face reconstruction has many uses, for example, in face recognition, film character creation, and teleconferencing. Applications can be broadly categorised using two factors: the required accuracy and the extent to which the environment can be controlled. Solving face reconstruction with high accuracy and little control is a hard problem, however, compromising one for the other makes the problem more manageable. There are many methods for reconstruction that produce accurate models and that are highly-constrained and/or require many resources, such as laser-scanning, structured light, or multi-view stereo. Approaches to face reconstruction in uncontrolled environments but where a lower level of accuracy is acceptable are fewer, despite the fact that is there is potentially a considerable market for such techniques, for example, in avatar generation for games. It is this gap that we are trying to fill, specifically: fast, approximate reconstruction using multiple views from a single low-cost camera in a largely uncontrolled environment.

There are several difficulties inherent in using a few, uncalibrated, low-quality views. Being restricted to a single camera means images taken of the face will not be simultaneous and so there is likely to be minor variation in the shape between views. Also, in many situations the required views are produced by rotating the head, rather than the camera, introducing further deformation. The quality of images retrieved from a low-cost camera

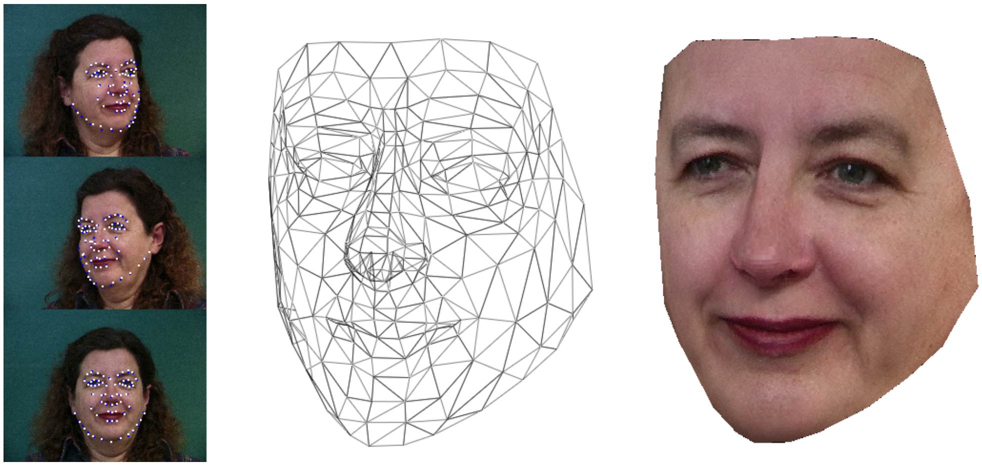


Figure 1: A face from the IMM face database, reconstructed from three views. The left-most images show the faces and their detected feature points. The right two show the reconstruted mesh, without and with texture.

are often poor, suffering from low dynamic range, noise, compression artefacts, etc., as well as being of a low resolution — although this is gradually improving as camera technology advances and devices become cheaper. Also, due to the likely use of the system, lighting conditions are liable to be non-ideal and changing and the direction from which the photos are taken unknown.

There are many different approaches to face reconstruction and shape reconstruction in general. A few examples include shape-from-shading [14, 17], model fitting [15], and space carving [7].

One approach that encompasses similar elements to our method is the work of Breuer et al. [2], which uses Support Vector Machines (SVMs) to detect feature points and then fits a 3D Morphable Model (3DMM) [1] to a single view to produce a 'plausible and photo-realistic' model. Similar work includes that of Gu and Kanade [5]. Our method is most similar to that of Zhang et al. [18], which uses multiple consecutive monocular views to perform triangulation of a set of manually marked feature points and then fits a high-resolution mesh to match. While these all produce reasonable results, they suffer from a reasonably long run-time, taking up to several minutes — a time that may be unacceptable in many applications. In comparison, our method is fast, taking less than a second, and requires no manual initialisation.

The rest of this paper describes the method in detail followed by presentation of results and discussion. Figure 1 shows an example reconstruction.

## 2 Method

The process requires one or more images of the face from approximately known views. In the majority of cases we use three images: one from the front and two three-quarter profiles. Though the method does work with a single view, results are much less accurate. Note that our system can also be run on a video sequence of a person rotating their head, automatically

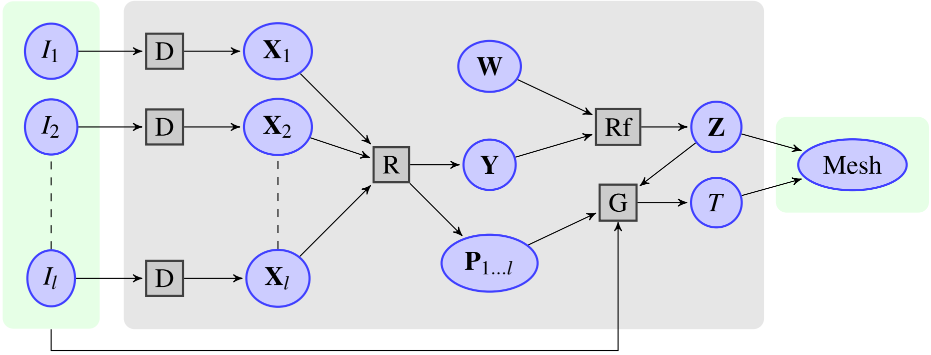


Figure 2: Overview of the face reconstruction process. $I_{1\ldots l}$ are the input images, D is the feature detection process, $\mathbf{X}_{1\ldots l}$ are the detected feature sets, R is the reconstruction step, $\mathbf{Y}$ and $\mathbf{P}_{1\ldots l}$ are the reconstructed coarse model and camera information, Rf is the model refinement stage with weights, $\mathbf{W}$, as an additional input, G is the texture generation process, and $\mathbf{Z}$ and $T$ are the refined model and generated texture that are combined to produce the final mesh.

selecting the required views.

Given multiple input images we produce a high-resolution, reconstructed 3D mesh: $\mathbf{I} \to \{\mathbf{Y}, T, f, M\}$, where $\mathbf{I} = \{I_1, \cdots, I_l\}$ is a set of $l$ input images, $\mathbf{Y}$ is a reconstructed set of $n$ 3D vertices with triangulation $f$, $T$ is a generated texture image, and $M\colon \mathbb{N} \to \mathbb{R}^2$ is a mapping from vertex indices to texture coordinates. $M$ and $f$ are precomputed as they are independent of the input images. The process has four distinct stages: face and feature detection, model reconstruction, model refinement, and texture generation. Figure 2 shows an overview of the process.

## 2.1 Feature Detection

The initial stage of the process uses Active Appearance Model (AAM) [3] to detect feature points of the face.

First the face is detected, giving a bounding box $b_k = \{x_k, y_k, w_k, h_k\}$, with a centre at $(x_k, y_k)$ and a width and height of $w_k$ and $h_k$ respectively. An extended Viola-Jones face detector [9, 16], which employs a boosted cascade of Haar-like features, is used to detect the face region. The choice of detector is largely inconsequential as it is simply used to initialize the AAM, using the mean points, $\overline{\mathbf{X}}$, and a learned scale $\alpha$ and translation $(\beta_x, \beta_y)$ relative to the detected face rectangle,

$$\mathbf{X}_{k0} = \alpha \overline{\mathbf{X}} \begin{bmatrix} w_k & 0 \\ 0 & h_k \end{bmatrix} + \mathbf{1}([\beta_x \ \beta_y] + [x_k \ y_k]).$$

They are then fit independently to each image using a AAM with model $M_k$:

$$\mathbf{X}_k = \mathrm{AAM}(I_k, M_k, \mathbf{X}_{k0}).$$

We use the I1I2I3 colour space, as proposed by Ohta et al. [12], where the image channels are strongly decorrelated. This has been shown by Ionita et al. [6] to be more effective than

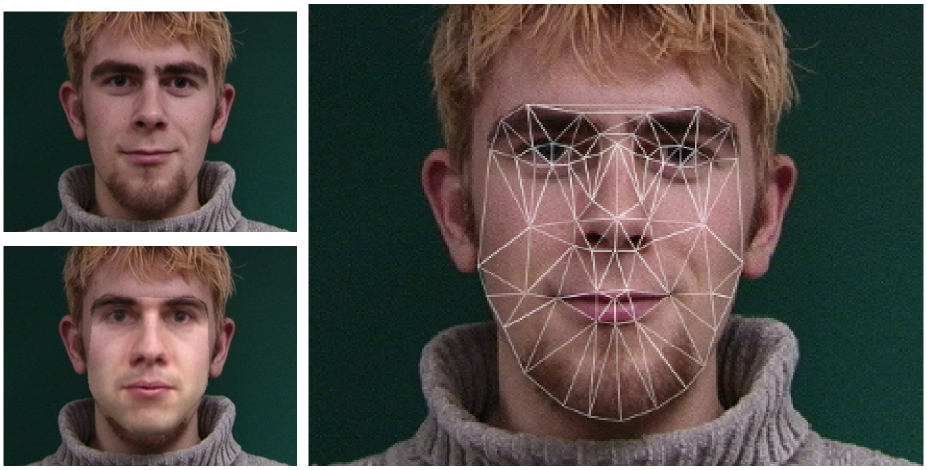


Figure 3: Upper-left shows the detected face, lower-left shows the texture recovered as part of the AAM process, and right shows the detected features.

using a single grey channel or simple RGB. As in [4] we build a separate model for each view, with mirrored views being considered the same. Figure 3 shows the detected feature points.

A weakness of using an AAM as a feature detector is that many of the points lie on edges and so tend to fit to a line rather than a specific point. There is therefore not necessarily a point-to-point correspondence between views but rather one from line-to-line. This is handled during the reconstruction phase.

When using a video sequence the optimal views can be selected automatically by fitting all models at each frame and using the frames that have the lowest reconstruction error for each model. This allows a short video of a person slowly turning their head to be used rather than having them correctly position their head for each view and then manually taking a photograph.

## 2.2 Model Reconstruction

Once the feature points have been detected their 3D positions can be reconstructed. Part of this process also involves reconstructing the parameters of the camera used to take the images. This information is later used to refine and texture the model.

Rather than considering the images taken from a stationary camera with the head rotating, as is the more likely case, we reconstruct the model assuming a stationary head and rotating cameras. The two can be considered equivalent assuming the model does not change between views. We therefore need to recover one set of intrinsic parameters and as many extrinsic as there are views.

Given corresponding feature points in each image, $\mathbf{X}_k$, we reconstruct their 3D positions, $\mathbf{Y}$. 3D vertices are projected to 2D by:

$$\mathbf{X}_k = \mathbf{P}_k \mathbf{Y},$$

where $\mathbf{P}_k$ is a projection matrix composed as

$$\mathbf{P}_k = \mathbf{C} \left( \mathbf{R}_k | \mathbf{t}_k \right).$$

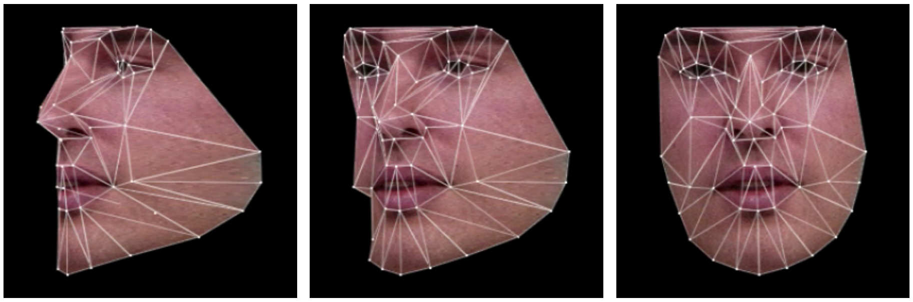


Figure 4: A coarse face model reconstructed from three views.

$\mathbf{R}_k$, $\mathbf{t}_k$, and $\mathbf{C}$ form the standard camera and transformation matrices describing the position, direction, and intrinsic properties of the camera.

The fitting is performed using Sparse Bundle Adjustment (SBA) [10], which is based on the Levenberg-Marquardt algorithm but optimized for use when the relationship between parameters is sparse, by minimising the reprojection error:

$$\{\mathbf{Y},\mathbf{P}\} \approx \underset{\{\mathbf{Y},\mathbf{P}\}}{\arg\min} \sum_{k=1}^{l} \|\mathbf{X}_k - \mathbf{P}_k\mathbf{Y}\|.$$

The process for fitting summarised as follows:

1. **Initialize the 3D model** This is done by using the $x$ and $y$ values from a single view, usually the frontal, and the $z$ values from the mean head aligned to that view.

2. **Initialize the intrinsic camera parameters** These can be set to either likely values or known values. It is assumed that all images are taken with the same camera with unchanged intrinsic parameters.

3. **Initialize the extrinsic camera parameters** The positions and rotations of the camera can be estimated based on the views used to train the AAMs.

4. **Fit the camera parameters** This is done using SBA by minimizing the reprojection error. In many situations the intrinsic camera parameters are known and these can be omitted from the optimization to improve accuracy.

5. **Move line-points** As previously mentioned, many points lie on lines and so do not necessarily correspond between views. To overcome this we move the relevant detected points along their lines to minimize the distance to their corresponding point when the current model is projected into that view.

6. **Fit the vertex positions** Again this is done using SBA in a similar manner, optimizing for vertex points rather than the camera parameters.

7. **Repeat steps 4-6 until converged** When the change between iterations falls below some threshold. Usually only a few ($< 5$) iterations are required.

Figure 4 shows the reconstructed 3D positions of feature points on a face after this process.

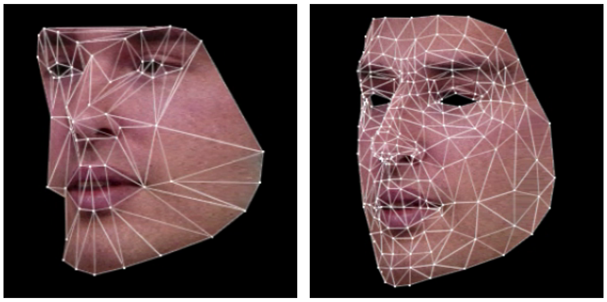


Figure 5: A coarse face model, $\mathbf{Y}$, and subsequent refinement, $\mathbf{Z}$. The mesh has gone from 64 points to 267.

## 2.3 Model Refinement

The reconstructed 3D model points, $\mathbf{Y}$, are too few to form a useful mesh and so we fit an existing mesh with a higher number of vertices.

This is achieved by 'morphing' the higher resolution model's vertices, $\overline{\mathbf{Z}}$, to produce the final set of vertices, $\mathbf{Z}$. Each feature point in the model is treated as a control point for the mesh. Each control point has a manually specified weighted influence over the rest of the mesh that is Gaussian with distance from the control point, so nearby points are more affected than those that are far away. This is coded as a matrix of weights that operate on the deviation from the mean:

$$\mathbf{Z} \approx \overline{\mathbf{Z}} + \mathbf{W}(\mathbf{Y} - \overline{\mathbf{Y}}),$$

where $\mathbf{W}$ is $n \times m$ matrix of weights.

As an additional step we further morph the model to improve alignment of edges with those seen in the images, in particular the jaw/cheek line. We find the vertices in the high-resolution mesh that correspond to the edges and move them along their normals until they lie on the edge when projected onto the images. Again they have a weighted influence over the rest of the model so the entire mesh is morphed to conform to the silhouette. Figure 5 shows the result of this refinement.

## 2.4 Texture Reconstruction

The face model is textured using UV-mapping, which copies triangles of texture from a texture image to their projection positions. This can be achieved using a function such as a piecewise-affine warp (PAW) and requires a map, $M\colon \mathbb{N} \rightarrow \mathbb{R}^2$, from model vertex indices to texture coordinates.

$M$ is produced using a least squares conformal mapping [8], which tries to preserve angles between each face, so the mapping is very good locally but scale is not preserved. It is precomputed using the mean model vertices, $\overline{\mathbf{Z}}$.

During rendering, every triangle, $\mathbf{t} \in f$, in the model is warped from its position in the texture image, $M(\mathbf{t})$, to its projection, $\mathbf{P}_k\mathbf{t}$, in the output. In order to recover the model's texture we do the reverse of this and produce a texture image, $T_k$, for each input image, $I_k$:

$$T_k = \mathrm{PAW}(I_k, \mathbf{Z}, f, M, \mathbf{P}_k).$$

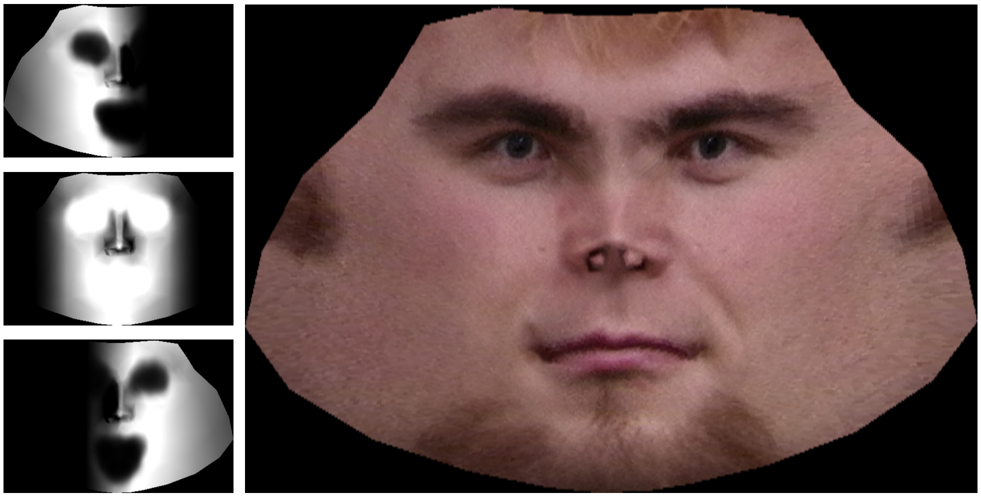


Figure 6: The left most images show the weightings, $\mathbf{W}_k$, used when combining texture maps from multiple views. The right image shows one such merged texture map, $T$.

However, we require a single texture image and so we merge these, weighted by the cosine of the angle between the surface normal, $\mathbf{n}_k(u,v)$, and the vector from that point on the surface to the camera, $\mathbf{c}_k(u,v)$:

$$T(u,v) = \frac{\sum_{k=1}^{l} \mathbf{W}_k(u,v) T_k(u,v)}{\sum_{k=1}^{l} \mathbf{W}_k(u,v)},$$

where $\mathbf{W}_k(u,v) \approx \mathbf{c}_k(u,v) \cdot \mathbf{n}_k(u,v)$.

In the case that a pixel in a texture map is not visible in the corresponding view, that is the surface is occluded or the angle between the camera and the surface normal is greater than 90 degrees, then its weight is set to be zero.

Merging images from multiple views will inevitably lead to blurring where the reconstructed model or its projection is not perfect. This is particularly noticeable along edges. As the camera positions are known approximately beforehand, the weights can be precomputed and manually adjusted to favour a single view in areas with strong edges, i.e., the eyes and mouth. This can clearly be seen in figure 6.

## 3 Results and Discussion

Figures 1, 7, 8, and 9 show example reconstructions. Similarly to the results of Breuer et al. [2], they are 'plausible and photo-realistic'.

All were produced using three images of the subjects face. The Active Appearance Models were built using 64 points and trained on the IMM face database [11]. The final meshes have just 267 vertices, due to a commercial requirement at the time, though the system has since been used with the Basel face model [13], which has some 50,000 vertices. The first two reconstructions are from the IMM face database and were omitted when training. The second two show real-life reconstructions produced in an office environment with subject

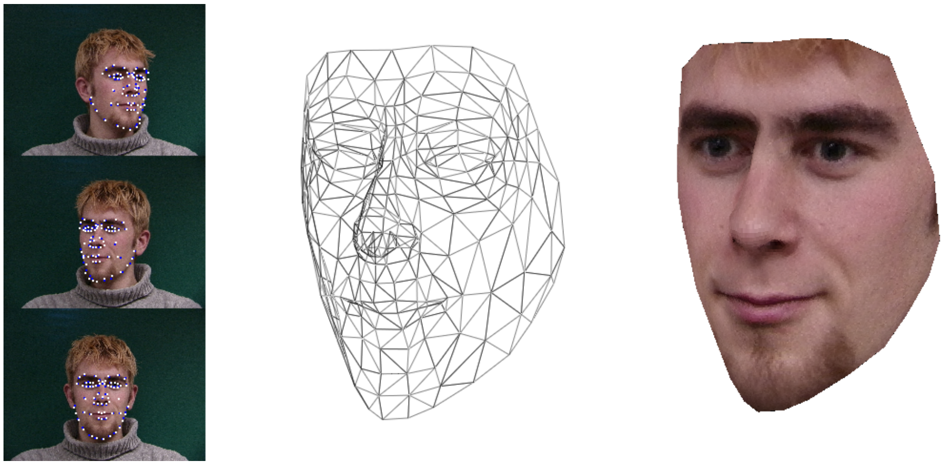


Figure 7: A second example of a face from the IMM face database reconstructed from three views.

unfamiliar with the system; images were chosen automatically from a short video sequence with a resolution of 640 ×480pixels.

A quantitative evaluation of the process as a whole is currently difficult to produce as a dataset comprising high-quality 3D reconstructions as well as multiple views in uncontrolled conditions is required: two requirements that almost preclude each other. Such a dataset could be produced synthetically and this is indeed planned as future work. However, the relaxed requirements on the accuracy of the reconstructions means that an absolute comparison is of a lesser importance than the robustness of the method and the visual quality of the results. Indeed, Breuer et al. [2] provide a numerical result by asking several participants to visually rate the accuracy as either very good, good, acceptable, or bad; though this approach seems unsatisfying.

The implementation is largely unoptimised but, for example, we are achieving a sub-second run-time to reconstruct faces as in figures 8 and 9 on an Intel® Core™2 Duo CPU @ 2.20 GHz with 2 GB of RAM.

There is considerable scope in the presented method for improvements through future work, both in the individual steps and in the system as a whole. Two principal directions of further work are:

- A weakness in fitting individual AAMs to each view is that each must be separately trained and so the system is restricted to only accepting images from views for which it has been built. We propose to develop a method for overcoming this by dynamically interpolating between AAMs to produce a new model specific to each view.

- An alternative means of overcoming this issue is to move from using several 2D AAMs to a single 3D one. This also has the advantage of removing the need for reconstructing the 3D positions of the points, though it requires fitting the camera parameters during the model fitting process. Consolidating the individual models would also allow the fitting to occur simultaneously in all views and thereby potentially reducing the error.

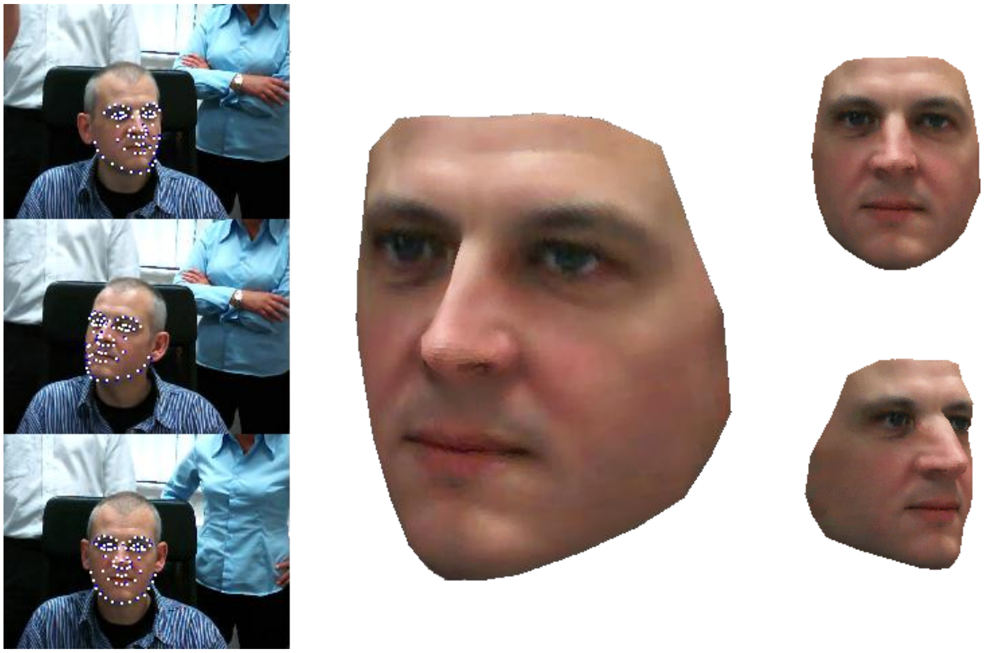


Figure 8: The reconstruction of a real-life head. The three views were chosen automatically from a short video sequence captured using a webcam.

## 4 Acknowledgements

This system was developed under the sponsorship of the Royal Commission for the Exhibition of 1851 and in collaboration with Warwick Warp Ltd.

## References

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

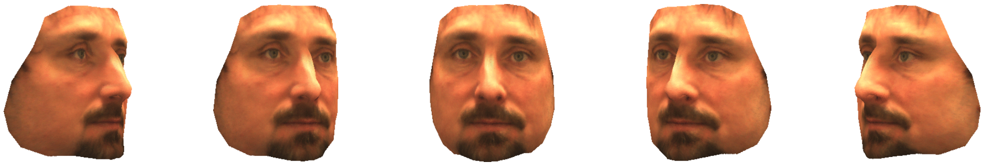


Figure 9: Five views of a head reconstructed from three images chosen automatically from a video sequence.

[2] P. Breuer, KI Kim, W. Kienzle, V. Blanz, and B. Schölkopf. Automatic 3D face reconstruction from single images or video. 2007.

[3] G.J. Edwards, C.J. Taylor, and T.F. Cootes. Interpreting face images using active appearance models. *Automatic Face and Gesture Recognition, IEEE International Conference on*, 0:300, 1998.

[4] G.J. Edwards, T.F. Cootes, and C.J. Taylor. Advances in active appearance models. *Computer Vision, IEEE International Conference on*, 1:137, 1999.

[5] Lie Gu and Takeo Kanade. 3D alignment of face in a single image. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, pages 1305–1312, Washington, DC, USA, 2006. IEEE Computer Society.

[6] Mircea C. Ionita, Peter Corcoran, and Vasile Buzuloiu. On color texture normalization for active appearance models. *Trans. Img. Proc.*, 18:1372–1378, June 2009.

[7] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[8] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics*, 21(3):362–371, 2002.

[9] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *Proc. IEEE ICIP 2002*, volume 1, pages 900–903, September 2002.

[10] M.I.A. Lourakis and A.A. Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)*, 36(1):2, 2009.

[11] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann. The IMM face database - an annotated dataset of 240 face images, 2004.

[12] Y. Ohta, Takeo Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13(1):222 – 241, July 1980.

[13] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. Genova, Italy, 2009. IEEE.

[14] E. Prados and O. Faugeras. Shape from shading. *Handbook of mathematical models in computer vision. New York: Springer*, pages 375–88, 2006.

[15] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. 2003.

[16] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE CVPR 2001*, pages 511–518, 2001.

[17] R. Zhang, P.S. Tsai, J.E. Cryer, and M. Shah. Shape-from-shading: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):690–706, 1999.

[18] Zhengyou Zhang, Zicheng Liu, Dennis Adler, Michael F. Cohen, Erik Hanson, and Ying Shan. Robust and rapid generation of animated faces from video images: A model-based modeling approach. *Int. J. Comput. Vision*, 58:93–119, July 2004.

Proc. of the 3rd British Machine Vision UK Student Workshop (BMVC'11 WS, Dundee, Scotland, 2nd September, 2011)

# Boundary Extraction in Images Using Hierarchical Clustering-based Segmentation

Arul N. Selvan[1]
a.n.selvan@SHU.ac.uk
http://www.shu.ac.uk/research/meri/mmvl/research/projects/medical/index.html

[1] Materials & Engineering Research Inst.
Sheffield Hallam University,
Sheffield. S1 1WB. UK.

## Abstract

Hierarchical organization is one of the main characteristics of human segmentation. A human subject segments a natural image by identifying physical objects and marking their boundaries up to a certain level of detail [1]. Hierarchical clustering based segmentation (HCS) process mimics this capability of the human vision. The HCS process automatically generates a hierarchy of segmented images. The hierarchy represents the continuous merging of similar, spatially adjacent or disjoint, regions as the allowable threshold value of dissimilarity between regions, for merging, is gradually increased. HCS process is unsupervised and is completely data driven. This ensures that the segmentation process can be applied to any image, without any prior information about the image data and without any need for prior training of the segmentation process with the relevant image data.

The implementation details of HCS process have been described elsewhere in the author's work [2]. The purpose of the current study is to demonstrate the performance of the HCS process in outlining boundaries in images and its possible application in processing medical images.

# 1 Introduction

Segmentation can be thought as a process of grouping visual information, where the details are grouped into objects, objects into classes of objects, etc. Thus, starting from the composite segmentation, the perceptual organization of the image can be represented by a tree of regions, ordered by inclusion. The root of the tree is the entire scene, the leaves are the finest details and each region represents an object at a certain scale of observation [1]. Since the early days of computer vision, the hierarchical structure of visual perception has motivated clustering techniques to segmentation [3], where connected regions of the image domain are classified according to an inter-region dissimilarity measure.

Hierarchical Clustering-based Segmentation (HCS) implements the traditional bottom-up approach, also called agglomerative clustering [4], where the regions of an initial partition are iteratively merged. HCS procedure automatically generates a hierarchy of segmented images. The hierarchy of segmented images is generated by partitioning an image into its constituent regions at hierarchical levels of allowable dissimilarity between its different regions. At any particular level in the hierarchy, the segmentation process will cluster together all the pixels and/or regions which have dissimilarity among them less than or equal to the dissimilarity allowed for that level.

It should be noted that HCS process is quite different from Hierarchical image segmentation methods discussed, for example, in [5]. The hierarchy in the study [5] refers

doi:18/07/2011

to the multi-resolution hierarchy in scale space. Also HCS process, unlike other segmentation methods like [6], [7] and [8], is not an iterative optimization process. Instead at each level HCS process yields an optimized segmentation output related to the dissimilarity allowed for that level.

Details of the implementation of the HCS process are described in the author's work in [2]. The purpose of this paper is two fold firstly to present the results of the objective evaluation of the performance of HCS process as a segmentation tool and secondly to present the possibility of how HCS process can be used as a computer aided monitoring (CAM) tool to assist radiologists to monitor abnormalities in X-ray mammograms.

The rest of the paper is organized as follows. Firstly the operation of the HCS process is outlined. Also the facilities offered by the GUI to display the HCS output is discussed. Secondly the performance of the HCS process is evaluated. Finally the success of HCS in segmenting medical images and using HCS process as an aid for radiologists are discussed.

## 2 Overview of the HCS Process

Following is a high-level description of the HCS process [2] (See Figure 1 for a flow chart representation) :

1. Give each pixel in the image a region label as follows :
   If an initial segmentation of the image is available, label each pixel according to this pre-segmentation. The initial segmentation can be obtained by prior class information (for e.g. based on the user information).
   If no initial segmentation is available, label each pixel as a separate region.
   Set current dissimilarity allowed between regions, *dissimilarity_allowed*, equal to zero.
2. Calculate the dissimilarity value, (*dissimilarity_value*), between all pairs of regions in the image.
   Set *threshold_value* equal to the smallest *dissimilarity_value*.
3. If the *threshold_value* found, in step 2, is less than or equal to the current *dissimilarity_allowed* value, then merge all those regions having *dissimilarity_value*, between them, less than or equal to the *threshold_value*.
   Otherwise go to step 6.
4. If the number of regions merged in step 3 is greater than 0, then reclassify the pixels on the border of the merged regions with the rest of the regions until no more reclassification is possible. After all the possible border pixels are reclassified, among the merged regions, store the region information for this iteration as an intermediate segmentation and go to step 2.
   Otherwise, if the number of regions merged in step 3 is equal to 0 then, go to step 5.
5. If the current number of regions in the image is less than the pre-set value, *check_no_regions*, then go to step 7. Otherwise, go to step 6.
6. If the current value of *dissimilarity_allowed* is less than the maximum possible value then increase the *dissimilarity_allowed* value by an incremental value, *dissimilarity_allowed* = *dissimilarity_allowed* + *dissimilarity_increment*, and go to step 2. Otherwise go to step 7.
7. Save the region information from the current iteration as the coarsest instance

The above steps ensures that the segmentation does not depend on the order in which the image regions are processed. Normally in agglomerative clustering methods the cluster structure depends on the order in which the regions are considered [4]. The brute force approach, followed by the HCS process, where only those regions with the smallest overall dissimilarity are merged in each step, is the only solution to overcome this effect [9].

The feature measure used by the HCS process to estimate the similarity is the actual distribution of the pixel values in a region surrounding the locations. This technique may be considered to work in a way similar to the human visual system where features for texture (region) segmentation are not consciously computed [10].

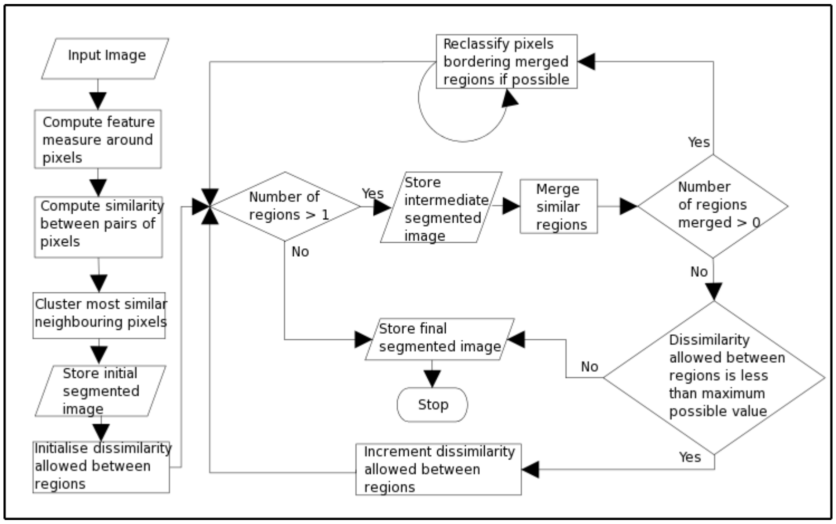

Input Image
Compute feature measure around pixels
Compute similarity between pairs of pixels
Cluster most similar neighbouring pixels
Store initial segmented image
Initialise dissimilarity allowed between regions
Reclassify pixels bordering merged regions if possible
Number of regions > 1
Yes
No
Store intermediate segmented image
Merge similar regions
Number of regions merged > 0
Yes
No
Store final segmented image
Stop
Dissimilarity allowed between regions is less than maximum possible value
No
Yes
Increment dissimilarity allowed between regions

Figure 1: Flow chart illustrating the different operations of the HCS process [2]

## 2.1 Graphical User Interface (GUI) for Displaying HCS Results

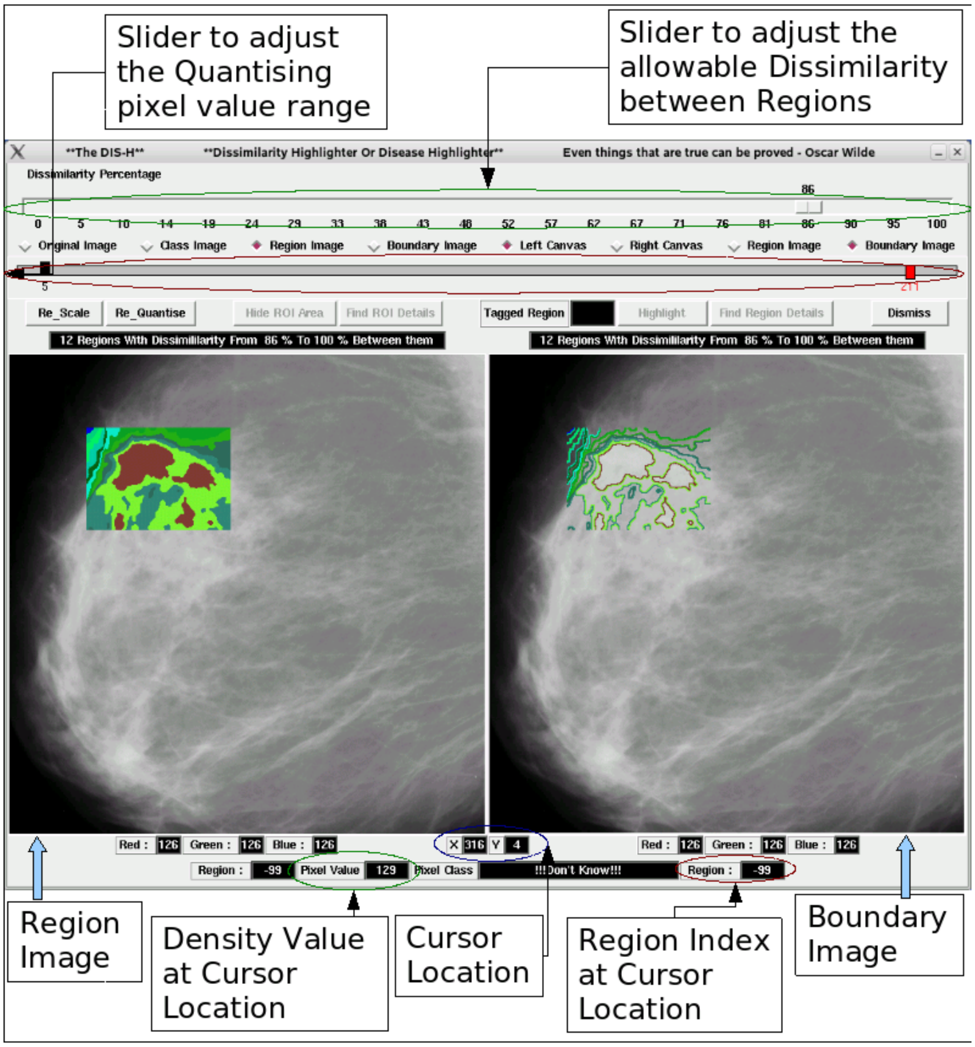
The HCS process generates a hierarchy of segmentation results, associated with a set of dissimilarity values. The segmentation output is stored at the end of the HCS processing. Subsequently the GUI can be used to reproduce the resulting segmentation images associated with a dissimilarity value instantaneously. The GUI is designed in such a way as to make it easy for the user to view all the different solutions and select the most suitable. The easy viewing and scrutinizing of the segmentation output is achieved by the GUI by having the following facilities :

1. The hierarchical segmentation output can be viewed by using a slider bar giving the dissimilarity index.
2. Individual region properties like the number of pixels, the lowest, highest and average pixel value and the distribution of the pixel values within the region can be scrutinized.
3. The original image or another segmented image at a different level of dissimilarity can be compared with a segmented image by displaying them alongside each other and a dual cursor moves simultaneously on both the images.

The image shown in Figure 2 gives a screen captured image of the GUI and the user controls provided for the above listed facilities. The GUI facility will be useful for the user to choose the right segmentation to identify the regions which correspond to the healthy and diseased part of the breast tissue in X-ray mammograms (Section 3).

## 2.2 Objective Evaluation of the Performance of the HCS Process

To evaluate the performance of segmentation methods like HCS, which yield a hierarchy of boundaries, Precision-Recall curves, (Berkley benchmark) an alternative to Receiver Operating Characteristic (ROC) curve, is more suited [11]. The procedure to benchmark the performance of any boundary detecting algorithm, like HCS, is described as follows :



**The DIS-H** **Dissimilarity Highlighter Or Disease Highlighter** Even things that are true can be proved - Oscar Wilde

○ Orginal Image ○ Class Image ● Region Image ○ Boundary Image ● Left Canvas ○ Right Canvas ○ Region Image ● Boundary Image

12 Regions With Dissimillarity From 86 % To 100 % Between them

12 Regions With Dissimillarity From 86 % To 100 % Between them

Red : 126 Green : 126 Blue : 126 X 316 Y 4 Red : 126 Green : 126 Blue : 126

Region : -99 Pixel Value 129 Pixel Class !!!Don't Know!!! Region : -99

Figure 2: Annotated screen captured image of the GUI, to visualize HCS process output.

For each image a collection of human-marked boundaries, marked by different human subjects, constitutes the ground truth. The output of the boundary detecting algorithm, will be a boundary map with one pixel wide boundaries. The objective is to find out how well the algorithm's boundary map matches with that of the human subjects'.

Traditionally, one would "binarize" the boundary map by choosing some threshold. There are two problems with thresholding a boundary map: [12]

1. The optimal threshold depends on the application, and
2. Thresholding a low-level feature like boundaries is likely to be a bad idea for most applications, since it destroys much information.

For these reasons, the Berkley benchmark operates on a non-thresholded boundary map. Nevertheless, one need to threshold the boundary map in order to compare it to the ground truth boundaries, but this is done at many levels. At each level, the two quantities -- precision and recall – are computed. In this manner the precision-recall curve for the boundary detecting algorithm is produced [12].

The performance of the boundary detecting algorithm is measured by estimating the summary statistic as follows : The F-measure, which is the harmonic mean of precision and recall is defined at all points on the precision-recall curve. The maximum F-measure value across an algorithm's precision-recall curve is its summary statistic [12].

A subset of one of the natural scenes of the Berkley database [12] (Image ID 42049) (Figure 3 a) was used to evaluate the performance of the HCS process, as discussed above. Figure 3 shows the human subjects' hand drawn boundaries (Top row) and some of the HCS process intermediate segmentation output (Bottom row). Figure 4 shows the precession and recall plot for 228 segmentation levels of the HCS process. For the HCS process the maximum F-measure value was found to be 0.82 (Figure 4).

The corresponding scores for other segmentation algorithms ranges from 0.92 [13] to 0.76 [14]. (See [12]). The algorithm which has got the highest score of 0.92 is supervised [13] while HCS process is unsupervised. Moreover the human subjects' boundaries, with which the algorithms' output is compared, outline only the major structures in the scene. For example none of the human subjects has outlined the border of the bird's legs. In contrast the HCS process has not only outlined the legs of the bird but has also outlined the subtle difference in the chest plumage of the bird (Figure 5). This outlining of subtle differences within an otherwise a homogeneous region is found very useful in highlighting dissimilarities in diagnostic images. Because tissue abnormalities, in medical images are indicated, by part of the image being dissimilar from other homogeneous areas representing healthy parts.

# 3 Medical Image Segmentation Using HCS Process

The goal of medical image segmentation is to separate the image into regions that are meaningful for a specific task. This task may, for instance, involve the detection of specific section of organs or quantitative measurements made from the images. Medical image segmentation is a difficult task because of issues such as spatial resolution, poor contrast, ill-defined boundaries, noise, or acquisition artefacts [15]. The medical images in the Figures 6 and 7 illustrate this difficulty. Figure 6 shows a CT image of a section of the brain. The image area within the region of interest (rectangular area outlined in black) is made up of three different types of regions viz. Grey matter, White matter and the stroke affected area. The stroke affected area has been outlined in white by an expert. Figure 7 shows the segmentation result obtained by Hierarchical Segmentation (HSEG) [16] which is very similar to HCS process. Figure 7 illustrates the difficulties faced by the HSEG process to segment medical images. Although the image pixels within the region of interest (ROI), have been segmented into three classes colour coded as Red (the diseased area), Green (white matter) and blue (Grey matter) it has misclassified some of the pixels not belonging to the diseased area as being diseased as well. This can be seen by the presence of red coloured pixels at the other end of the ROI outside the area outlined by the expert.

Figure 8 shows the segmentation of the same CT image (Figure 6) by the HCS process. It can be seen that the HCS process has successfully delineated the three different types of regions viz. Grey matter, White matter and the stroke affected area. Comparing the segmentation results of the HCS process Figure 8 and with that of the HSEG process Figure 7, it can be seen that the HSEG process segmentation (Figure 7) is suboptimal while the HCS process is able to achieve a smooth segmentation as can be seen in Figure 8.

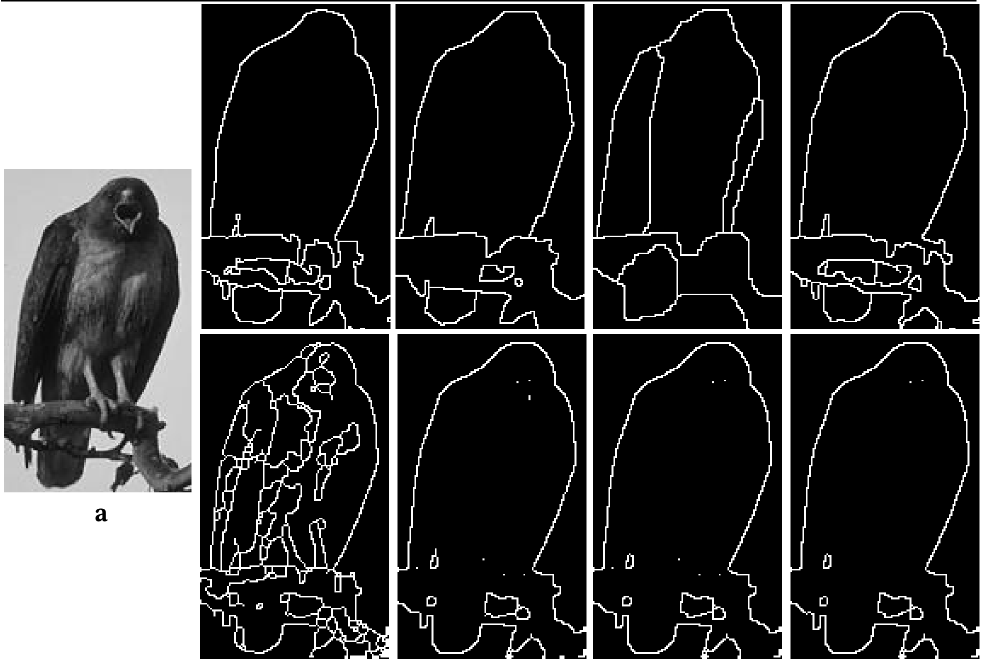


Figure 3: A natural scene from the Berkley database (subset of Image ID 42049) (a). Top row images are boundaries, marked by 4 different human subjects. Bottom row images are the HCS process boundary output when there were 22, 15, 10 and 6 regions.

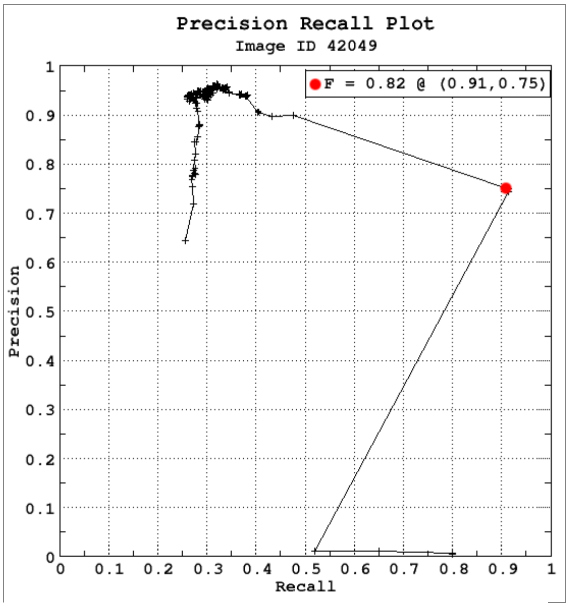


Figure 4: Precision and Recall plot for 228 levels

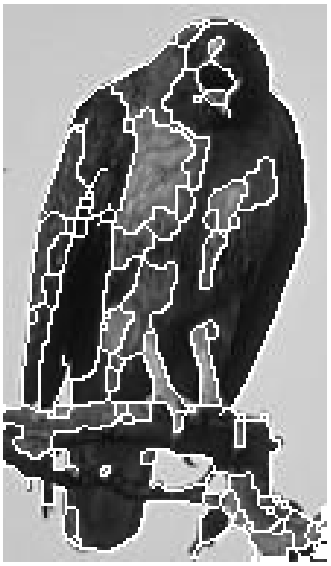


Figure 5: HCS process segmentation of the natural scene (Figure 3 a)

## 3.1 Border Pixel Re-Classification

One of the unique feature of the HCS process is the border pixel classification operation (Figure 1). Border pixel reclassification is necessary because the merging process starts

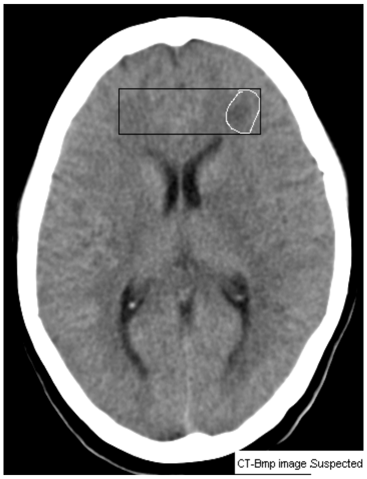


Figure 6: CT image showing the suspected area outlined in white by a neuroradiologist.

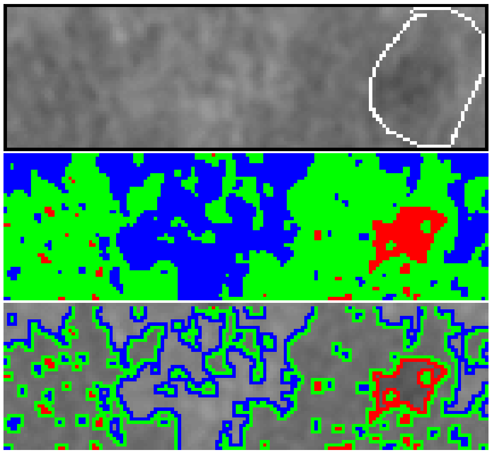


Figure 7: Segmentation of the Grey matter, White matter and Stroke affected regions and their boundaries by HSEG [16]

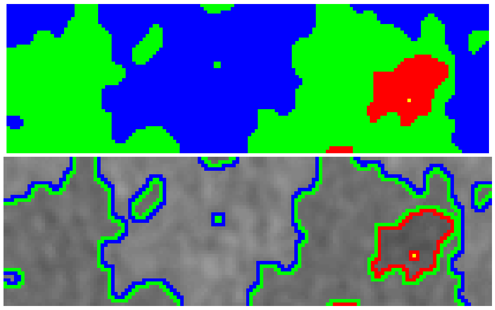


Figure 8: Segmentation of the Grey matter, White matter and Stroke affected regions and their boundaries by HCS [2]

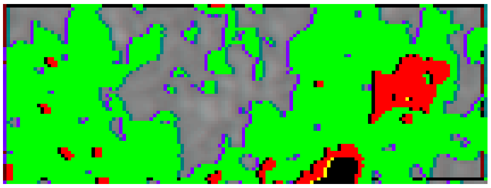


Figure 9: Segmentation of the Grey matter, White matter and Stroke affected regions by HCS without border pixels reclassification, where misclassification had occurred.[2]

with individual pixels and the pixels are merged to form regions and subsequently regions are merged to form bigger regions and so on. During the initial merging pixels/regions belonging to different types/classes might get merged. This could happen, for example when comparing pixels/regions over a small neighbourhood, where the dissimilarity between pixels/regions belonging to different types/classes of regions might be smaller than the dissimilarity between pixels/regions belonging to the same type/class of region because of the local in-homogeneity. However as the regions grow, during border pixel reclassification, those pixels which had been merged as a result of comparison with smaller neighbourhood might subsequently be reclassified by properly comparing them with the rest of the pixels from a larger neighbourhood.

Border pixel reclassification was considered only for those pixels on the boundary of the clusters which had been merged with other clusters. These boundary pixels were removed one at a time from their original clusters. The pixel removed was considered as a region of its own and the similarity between the one pixel region and the regions bordering it (which include the original cluster to which it belonged to) were found and the single pixel region was merged with the most similar bordering region.

Figure 9 shows the intermediate segmentation results of the HCS process, performed without border pixels reclassification. In Figure 9 the pixels belonging to the larger clusters of the major classes White Matter and the infarct are coloured as Green and Red. Pixels clustered as belonging to the Grey matter class are not coloured. From Figure 9 it can be seen that the cluster belonging to the major class infarct has misclassified pixels on the other part of the image.

Comparing the segmentation output of the HSEG (Figure 7) with that obtained by HCS without border pixel reclassification (Figure 9), it can be seen that they are almost the same. Since the similarity measures adopted by the HCS and the HSEG processes may or may not be the same it can be safely assumed that border pixel reclassification plays a crucial role in obtaining a smooth segmentation.

Although only one example was shown to compare the performance of the HSEG with that of HCS. It is also shown how HCS process output is very similar to that of HSEG when HCS process does not perform border-pixel reclassification. From this it can be inferred that HCS process with border-pixel-reclassification will always give a better segmentation output when compared to HSEG.

## 3.2 HCS Process as a Computer Aided Monitoring (CAM) Tool

X-ray mammography is the most effective tool for the detection and diagnosis of breast cancer, but the interpretation of mammograms is a error-prone task. Hence, computer aids have been developed to assist the radiologist. While Computer-aided detection (CADe) systems address the problem that radiologists often miss signs of cancers, Computer-aided diagnosis (CADx) systems assist them to classify the lesions as benign or malignant [17].

This study demonstrates a novel alternative system namely computer-aided monitoring (CAM) system. The designed CAM system can be used to objectively measure the properties of suspected abnormal areas in mammograms. Thus it can be used to assist the clinician to objectively monitor abnormalities. In brief the designed CAM system works as follows : Using the approximate location and size of an abnormality, obtained from the user, the HCS process automatically identifies the more appropriate boundaries of the different regions within a region of interest (ROI), centred at the approximate location. From the set of, HCS process segmented, regions the user identifies the regions which most likely represent the abnormality and the healthy areas. Subsequently the CAM system compares the characteristics of the user identified abnormal region with that of the healthy region; to differentiate malignant from benign abnormality. An example follows.

Figure 10 shows a X-ray mammogram (mdb102) of a dense glandular breast having a malignant asymmetry class of abnormality. Using the information provided in the mini-MIAS database [18] the approximate boundary of the abnormality (Red circle Figure 10 a), was located. The HCS process was applied within a ROI centred on the abnormality. Inspecting the HCS process output the user selected the region corresponding to the abnormality (Red Figures 10 b, c and d). The area within the approximate circular boundary, other than the abnormality, was selected as healthy, (Green Figure 10 d). Inspecting the HCS process output the user also selected a location, within the abnormal area, which was considered as the core of the abnormality (Yellow Figure 10 d and e). To estimate the dissimilarity between the abnormal and the healthy regions, the HCS process was applied only to the pixel locations within the abnormality (Red Figure 10 d and e) and the healthy (Green Figure 10 d and e) areas of the image. As the HCS process goes about merging similar regions within the abnormality and the healthy areas, the maximum average dissimilarity, measured between the cluster having the user tagged location (within

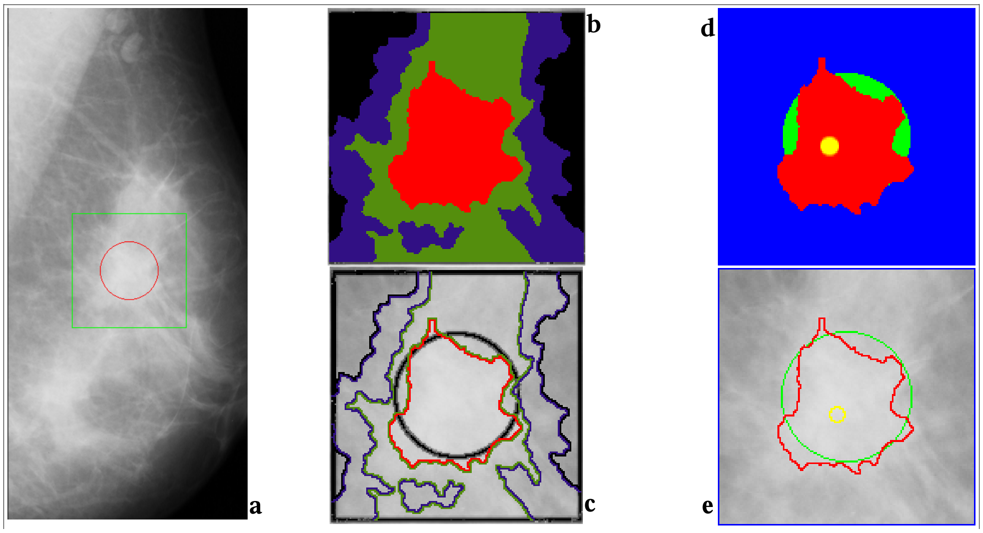


Figure 10: mini-MIAS mammogram (Image-ID mdb102) with the location and the approximate boundary of the abnormality circled in Red by the user (a).HCS process intermediate segmentation of four regions and their boundaries (b and c). Regions, and their boundaries, identified by the user as healthy (Green) and abnormal (Red) (d and e).

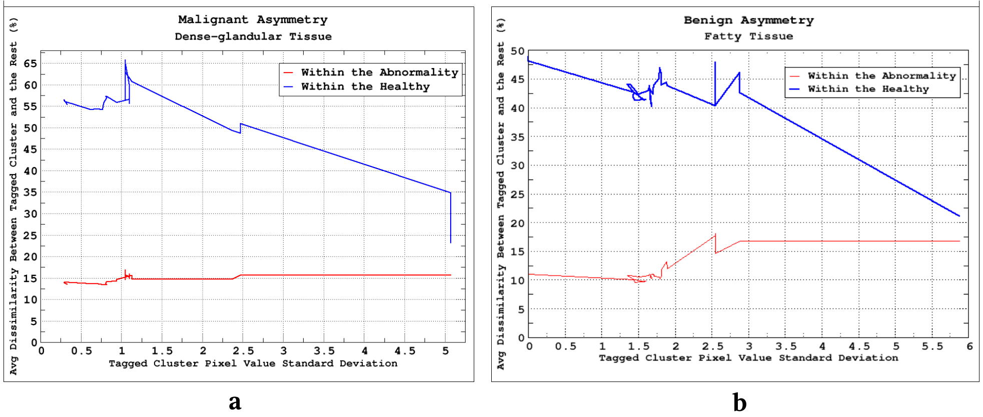


Figure 11: For mini-MIAS Mammogram images mdb102 and mdb097 graphs (a and b) showing the dissimilarity between the cluster having the user tagged location and other clusters belonging to the abnormality area and the healthy area.

the abnormality) and the clusters within the healthy region, is estimated. The heuristic used for differentiating malignant from benign abnormality is, if the value of the above estimated measure is less than fifty percent then the abnormality is benign else malignant. Graph shown in Figure 11 (a) demonstrates how the above measure and the criteria is able to classify the abnormality, under consideration, as malignant. Similarly the graph shown in Figure 11 (b) demonstrates how a benign abnormality is correctly classified.

The technique used to rate the HCS process based CAM effectiveness was to give it one malignant case and one benign case and then test its ability to determine which is which. The designed CAM process achieved 100% success rate in classifying malignant from benign asymmetric and circulant class of abnormalities in 16 mammograms from the mini-MIAS. The HCS process classifications were matched with that given in the mini-MIAS database [18].

# 4 Conclusion

Radiologists' expertise in reading diagnostic images calls for a combination of perceptual skills to find what may be faint and small features in a complex visual environment, and interpretive skills to rate their (the features') significance [19]. Computing systems are more consistent in their perceiving ability but they cannot match human interpretive skills. Drawing a line so as to limit the system's interpretive function has the virtue of achieving a complementary synthesis of system and radiologists' strengths [20]. However in practice the question of where to draw the line in computer aided detection/diagnostic systems between perception and interpretation is problematic [21].

The HCS process based CAM system developed in this study might be able to address the above problem. Work is in progress to establish whether the HCS based CAM system augments the diagnostic capabilities of the radiologists.

# References

[1] P. Arbelaez. Boundary Extraction in Natural Images Using Ultrametric Contour Maps. *Proceedings 5th IEEE Workshop on Perceptual Organization in Computer Vision* (POCV'06). June 2006. New York, USA.

[2] A. N. Selvan. Highlighting Dissimilarity in Medical Images Using Hierarchical Clustering Based Segmentation (HCS). *M. Phil. dissertation*, Faculty of Arts Computing Engineering and Sciences Sheffield Hallam Univ., Sheffield, UK, 2007.

[3] R. Ohlander, K. Price, and R. Reddy. Picture segmentation by a recursive region splitting method *Computer Graphics Image Processing*, 8:313-333, 1978.

[4] M. Nadler, and E. P. Smith. *Pattern Recognition Engineering* John Wiley and Sons inc. 1993.

[5] P. Schroeter J. Bigün Hierarchical image segmentation by multi dimensional clustering and orientation-adaptive boundary refinement, *Pattern Recognition*,Vol. 28, No. 5, pp. 695-709 1995;

[6] M. Jeon, M. Alexander, W. Pedrycz and N. Pizzi Unsupervised hierarchical image segmentation with level set and additive operator splitting, *Pattern Recognition Letter* 26 1461-1469 2005.

[7] H. Zhou, G. Schaefer, A. Sadka and M.E. Celebi Anisotropic mean shift based fuzzy C-means segmentation of dermoscopy images *IEEE Journal of Selected Topics in Signal Processing,* Vol. 3, No. 1, 26-34, 2009.

[8] H. D. Cheng and Ying Sun, A hierarchical approach to color image segmentation using homogeneity, *IEEE Transactions on Image Processing*, VOL. 9, NO. 12, 2071-2082, 2000.

[9] C. Thies, A. Malik, D. Keysers, M. Kohnen, B. Fischer and T.M. Lehmann, Content-based retrieval in medical image databases by hierarchical feature clustering, *Procs SPIE 2003;* 5032: 598-608. 2003.

[10] U. Bhattacharya, B. B. Chaudhuri, S. K. Parui, An MLP-based texture segmentation method without selecting a feature set *Image and Vision Computing* 15 937-948, 1997.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics *International Conference on Computer Vision*, Vancouver, Canada. 2001.

[12] Pablo Arbelaez, Charless Fowlkes and David Martin,. The Berkeley Segmentation Dataset and Benchmark June, 2007. http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

[13] Piotr Dollar, Zhuowen Tu, and Serge Belongie, Supervised Learning of Edges and Object Boundaries, *IEEE Computer Vision and Pattern Recognition* (CVPR), June, 2006.

[14] Stella X. Yu Segmentation Induced by Scale Invariance *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, June 20-26, 2005.

[15] M. Sonka, and J. M. Fitzpatrick. *Handbook of Medical Imaging Volume 2 Medical Image Processing and Analysis*. Published by SPIE-The international society for optical engineering. 2000..

[16] J. C. Tilton. Hierarchical Image Segmentation *On-line Journal of Space Communication Issue* No. 3 Winter 2003 Research and Applications.

[17] A. L. Elter M, A. Horsch. CADx of mammographic masses and clustered microcalcifications: a review. *Medical Physics* 2009 Jun;36(6):2052-68. 2009.

[18] J. Suckling, J. Parker, DR. Dance, S. Astley, I. Hutt, C. Boggis, I. Ricketts, E. Stamatakis, N. Cerneaz, SL. Kok, P. Taylor, D. Betal, and J. Savage. The Mammographic Image Analysis Society Digital Mammogram Database. *Exerpta Medica. International Congress Series* 1069 pp375-378, 1994. http://peipa.essex.ac.uk/info/mias.html

[19] Tabar L. and Dean P;. *Teaching atlas of Mammography* New York: Thieme,1985.

[20] E. Claridge, Experts' assessment as a "gold standard" for characterization of lesions, *In proceedings of Medical Image Analysis and Understanding* Oxford 1997.

[21] M. Hartswood, R. Procter, L. Williams, R. Prescott. and P. Dixon, Drawing the line between perception and interpretation in computer-aided mammography *In:Proceedings of the First International Conference on Allocation of Functions*: Allocation of Functions; 01 Oct 1997-03 Oct 1997; Galway. International Ergonomics Association Press; 1997. p. 275-291

Proc. of the 3rd British Machine Vision UK Student Workshop (BMVC'11 WS, Dundee, Scotland, 2nd September, 2011)

# Learning soft combination of spatial pyramid for action recognition

Wenqi Li
wyli@dundee.ac.uk
Jianguo Zhang
jgzhang@computing.dundee.ac.uk

School of Computing
University of Dundee
Dundee, UK

**Abstract**

This paper proposed a new approach to produce video descriptors based on the state-of-the-art spatial pyramid method. At different resolution level of spatial pyramid, video descriptors are extracted using either Bag-of-Features (BoG) representation or sparse coding representation. Instead of concatenating descriptors directly, binary support vector machines are constructed for each level in order to capture characteristics at different resolutions. The experiments are conducted with Gabor detector and 3D Histogram of oriented Gradient (HoG) description. The proposed approach is tested on KTH action dataset and Youtube action dataset with a three-level pyramid in both spatial and temporal scale. The performance are compared with standard Spatial-temporal Matching kernel , and experiments show promising results.

## 1 Introduction

Bag-of-Features representation based on local descriptors has been a very popular model for action classification tasks. Visual codebook is first constructed by applying a clustering algorithm, *e.g*. K-means, on descriptors set extracted from training videos. Each centre of cluster is presented as a "*visual word*". Then a histogram is formed by accounting the number of descriptors that are quantized to each visual word.

Since histogram only captures statistical characteristics of the descriptors set, BoF representation ignored all spatial and temporal information. To overcome this limitation, one extension of BoF is Spatial Pyramid Matching kernel (SPM) approach proposed by Lazebnik *et al*. [7]. SPM approach partitions image into increasingly fine sub-regions and computes histograms of local features found inside each sub-region. The final feature vector for video is a histogram from each sub-region concatenated with different weight according to level of resolution.

By using SPM representation, a typical video descriptor constructed with 3 uniform levels and 1024 visual words would have 36864 attributes, it would be computationally expensive to applying non-linear multi-class SVM on both training and testing stage, especially for large datasets which usually involve more than thousands videos.

In this paper, we developed a new approach to combine bag-of-features descriptors from each pyramid level using support vector machine (SVM). We derive descriptors of sub-regions using standard bag-of-features representation, descriptors from the corresponding sub-region are classified using multi-class SVMs. The SVM decision values against each action type can be viewed as an abstract description of the sub-region. Therefore, an efficient descriptor would be formed by concatenating weighted decision values. The final descriptor again would be then classified by standard non-linear SVM. Our approach builds SVM for each sub-region separately; a single classification task only involves the same number of attributes as visual codebook. Meanwhile, the tasks can be easily distributed to multicore processors and computer clusters. Furthermore, in our approach, the number of SVM decision values only depends on the number of training class labels, by concatenating SVM decision values we significantly reduce the dimensionality of the final video descriptors. By using this approach we achieved better performance than standard SPM approach.

We also adopted *sparse coding* (SC) representation for descriptors set based on Yang *et al*.'s work [12] on image classification. This extension of SPM model replaced the standard K-means vector quantization (VQ) with SC representation of descriptors set by relaxing the restrictive cardinality constraint of VQ. Additionally, unlike the original SPM that performs spatial pooling by computing histograms, they use max spatial pooling that is more robust to local spatial translations and more biological plausible. We integrated SC representation in our approach, as a comparison to traditional SPM, showed better classification accuracy.

The rest of paper is presented as follow: Section 2 introduces related woks used in the process, Section 3-7 demonstrate the applied techniques, Section 8 presents experiment setup and the evaluated results by adopting our approach, Section 9 draws conclusions and proposes future work.

## 2 Related Works

A popular architecture for solving video classification problem is (1)extracting low-level descriptors *e.g*. SIFT, at interesting point locations, then (2)combining these local descriptors into a global representation of video. Our combining strategy based on SPM as an extension to BoF is similar to [4, 6, 7, 12]. Differently we learn decision values for each SPM subregion using non-linear SVM, the decision values are concatenated with different weights to form a stronger video descriptor. Sub-region learning stages are independent to each other, the process can be parallelized efficiently.

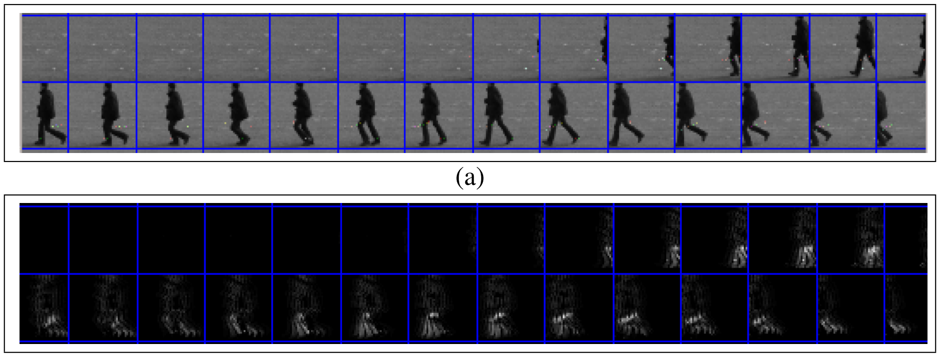
## 3 Extracting Low-level Descriptors

The low-level descriptors extraction methods are generally following existing popular methods. This section would take a brief description of the techniques being used.

### 3.1 Local Motions Detection

Given video data consisting of a sequence of frames, interest points are detected using 3D Gabor detector developed by Dollár *et al*. [2]. The response function has the form:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{1}$$



(a)

(b)

Figure 1: An example from KTH action dataset of original frames (a) and spatial-temporal interest point detected by Gabor detector (b)

In the function, $g(x,y;\sigma)$ is the 2D Gaussian smooth kernel for spatial dimensions, $h_{ev}$ and $h_{od}$ are a quadrature pair of 1D Gabor filters applied temporally. They are defined as:

$$h_{ev}(t;\tau,\omega) = -\cos(2\pi t\omega)\, e^{-t^2/\tau^2} \tag{2}$$

$$h_{od}(t;\tau,\omega) = -\sin(2\pi t\omega)\, e^{-t^2/\tau^2} \tag{3}$$

Where $\omega = 4/\tau$. The interest points are the local maxima of the responsefunction. One example is presented in Fig. 1. The parameters are set as $\sigma = 2$, $\tau = 1.37$.

### 3.2 Local Descriptor

In video, a local descriptor is a vector describing a interest point and its' near cuboid region. Both the gradient descriptor and the optical flow descriptor are equally effective in describing the motion information [10]. In this paper we adopted the 3D histograms of oriented gradients descriptor similar to Laptev *et al.* [5]. The cuboids near interest points is first smoothed, and gradients are calculated by dividing the cuboid into $(n_x \times n_y \times n_t)$ grids. A weighted histogram is finally accumulated based on oriented direction. We set the parameters as $n_x = n_y = 4$, $n_t = 2$. The histograms from different cuboids are concatenated over these sub-regions. Then Principle Component Analysis is done on the raw descriptor reducing number of elements to 200.

## 4 Descriptors Encoding

Here we borrow the notation in [12] for explanation. Let $\mathbf{X}$ be a set of descriptors in a D-dimensional feature space described above, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]^T \in \mathbb{R}^{M\times D}$. The encoding problem can be re-formulated into a matrix factorization problem:

$$\min_{U,V} \quad \sum_{M} \|\mathbf{x}_m - \mathbf{u}_m \mathbf{V}\|^2 \tag{4}$$

$$subject\ to \quad certain\ constraints\ on\ \mathbf{U}\ and\ \mathbf{V} \tag{5}$$

Where $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_M]^T$ is encoded descriptors set, V is referred as a visual codebook. By defining different constraints on the problem, we can have BoF encodings or SC encodings described as follows:

### 4.1 Bag-of-Features

In Eq.5, we define the constraints as

$$Card(\mathbf{u}_m) = 1, |\mathbf{u}_m| = 1, \mathbf{u}_m \geqslant 0, \forall m \tag{6}$$

Where $Card(\mathbf{u}_m) = 1$ is a cardinality constraint, meaning that only one element of $\mathbf{u}_m$ is nonzero, $\mathbf{u}_m \geqslant 0$ means that all the elements of $\mathbf{x}_m$ are nonnegative, and $|\mathbf{u}_m|$ is the L1-norm of $\mathbf{u}_m$, the summation of the absolute value of each element in $\mathbf{u}_m$. After optimization, the index of the only nonzero element in $\mathbf{u}_m$ indicates which cluster the vector $\mathbf{x}_m$ belongs to. The optimization is now converted into a standard BoF model.

This optimization problem can be solved by first calculating codebook $\mathbf{V}$ with K-means clustering algorithm, and then for each descriptor $\mathbf{x}_m$, find the membership indicator $\mathbf{u}_m$. In the coding phase, given a descriptor $\mathbf{x}_m$, the membership can also be learned with respect to the pre-calculated codebook $\mathbf{V}$.

### 4.2 Sparse Coding

If we adding a sparsity (number of nonzero elements) constraint on $\mathbf{u}_m$ and a L2-norm constraint on $\mathbf{v}_k$, the problem would be converted into SC:

$$\min_{U,V} \quad \sum_M \|\mathbf{x}_m - \mathbf{u}_m\mathbf{V}\|^2 + \lambda |\mathbf{u}_m| \tag{7}$$

$$subject\ to \quad \|\mathbf{v}_k\| \leqslant 1, \forall k = 1, ..., k \tag{8}$$

An algorithm proposed by [8] can solve the optimization problem efficiently. In the training stage a set of training descriptors are used to solve Eq.7 with respect to $\mathbf{U}$ and $\mathbf{V}$. For the coding stage, the SC codes are obtained by optimizing Eq.7 with fixed codebook $\mathbf{V}$. The SC coding can achieve a lower reconstruction error compared with standard BoF. So we choose to evaluate SC encoding methods along with BoF.

## 5 Spatial-Temporal Pooling

Given a encoded descriptor set $\mathbf{U}$, standard BoF combines them with histogram pooling, while max pooling function shows better performance on SC coding methods according to [12]. We evaluated both function in our experiments. This section describes these pooling strategies.

### 5.1 Histogram Pooling

The histogram pooling function is:

$$\mathbf{z} = \frac{1}{M} \sum_M \mathbf{u}_m \tag{9}$$

In standard BoF, orderless histogram is formed by accumulating descriptors as the video descriptor $\mathbf{z}$. In the extension of BoF, SPM constructs histograms on each sub-region of the video, and then concatenate them as a video descriptor.

### 5.2 Max Pooling

. The max pooling function is, every element of final descriptor $\mathbf{z}$ is:

$$z_j = \max\left\{|u_{1,j}|, |u_{2,j}|, ..., |u_{M,j}|\right\} \tag{10}$$

The max pooling function can also be applied to SPM. By changing sub-region, the encoded descriptors set $\mathbf{U}$ changed according to the scope.

## 6 Support Vector Machine

We use C-support vector classification [1] methods to complete action classification task. Given training feature vectors $\mathbf{x} \in R_n, i = 1, ..., l$, a vector $\mathbf{y} \in R_l$ such that $y_i \in \{1, -1\}$, The dual form of optimisation problem can be written as:

$$\min_{\alpha} \quad \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \tag{11}$$

$$subject\ to \quad \mathbf{y}^T\alpha = 0, \tag{12}$$

$$0 \leq \alpha \leq C, i = 1, ..., l \tag{13}$$

Where $\mathbf{e}$ is the vector of all ones, $C > 0$ is the upper bound, $Q$ is an $l$ by $l$ positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ is the kernel. We uses radial basis function (RBF) kernel for evaluation:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right), \gamma > 0 \tag{14}$$

Once the SVM model is trained, we label a new video fame sequences denoted by $\mathbf{x}_T$ descriptor with decision function:

$$sgn\left(\sum_{i=0}^{l} y_i\alpha_i K(\mathbf{x}_i, \mathbf{x}_T) + b\right) \tag{15}$$

LibSVM [1] is used for model training and testing.

## 7 Our Approach

We use the 3D Gabor detector for interest points detection. HoG feature descriptors for feature extraction as described before. We keep location information of every interest points throughout the process.

With standard BoF approach, We sample descriptors from training sets for computational efficiency, use K-means algorithm, define centres of clusters as words to create codebook. The measurement of clustering metric is Euclidean distance. $K$ parameter in K-means is set to be 1024. Therefore every descriptors from both training and testing sets can be assigned to a unique integer label in range 1 to 1024.

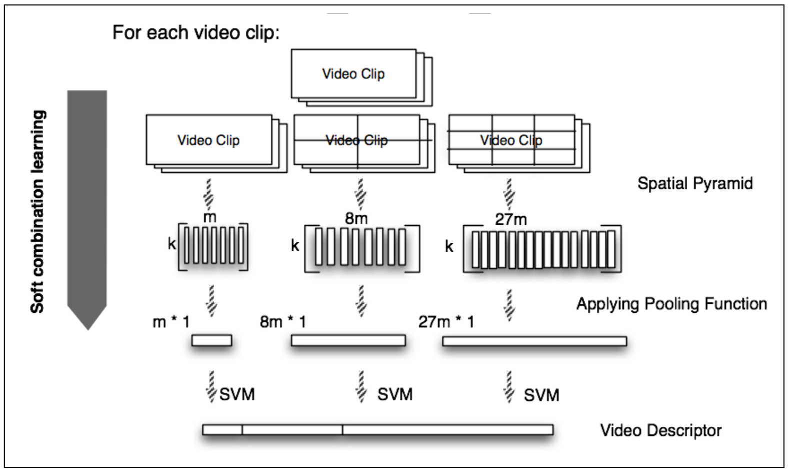


Figure 2: Soft combine strategy overview

With SC approach, we sampled descriptors from training sets, learned codebook by optimizing Eq.7 with respect to both **U** and **V**. Then in the coding stage we optimize descriptors codes **U** in Eq.7 with fixed codebook **V**.

Thereafter, a spatial-temporal pyramid with uniformly distributed grids is constructed on each video. As the resolution level increases, the videos are divided into several sub-regions in both spatial and temporal scale. Then we derive BoF feature vectors for all sub-regions. The BoF feature from the same level can be concatenated in sequence and then trained with SVM. Each level would therefore owned an SVM model that is ready for predicting. According to Eq.15, every descriptor would obtain decision values using the pre-trained SVM models. The decision values can then be concatenated with weight to form a descriptor per video (shown in Fig.2). That is to say, for a descriptor $\mathbf{d}_{pq}$ located at $p^{th}$ level, in $q^{th}$ sub-region, it can be tested in the trained model $M_p$ (note that this model is trained based on all descriptors in the training set which are located in level $(p)$ in different video clips). Ignoring the hard decision, for a two-class action classification, decision value is:

$$Decision\,Value\ R = \sum_{i=0}^{l} y_{ip}\alpha_{ip}K_p\left(\mathbf{x}_{ip},\mathbf{x}\right) + b_p \tag{16}$$

Instead of concatenating BoF features from each pyramid level directly, we form decision values as descriptor for each level and concatenate them with different weight as the final descriptor. We learn decision values in the early stage because decision values explain the descriptors set in a more concise way. For sub-regions evaluation, the length of encoded descriptors is fixed to the size of codebook, thus reduced computational complexity. Furthermore, descriptors for different sub-regions are independent, this would enable parallelized computing to make the overall process more efficient. For example in Fig.3, the most important sub-region is the centred one, high decision values from this sub-region with third level weight on the values, makes them much possible to be classified in the same class.

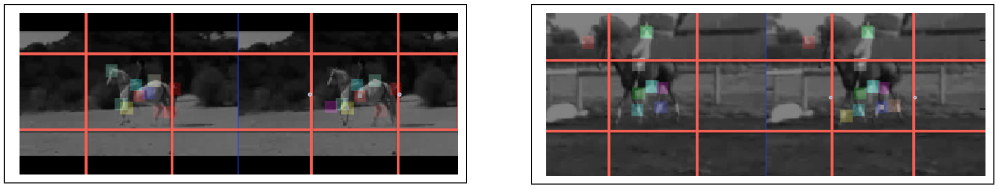


Figure 3: Sample frames from youtube action dataset, both are from videos with different actor riding horses.

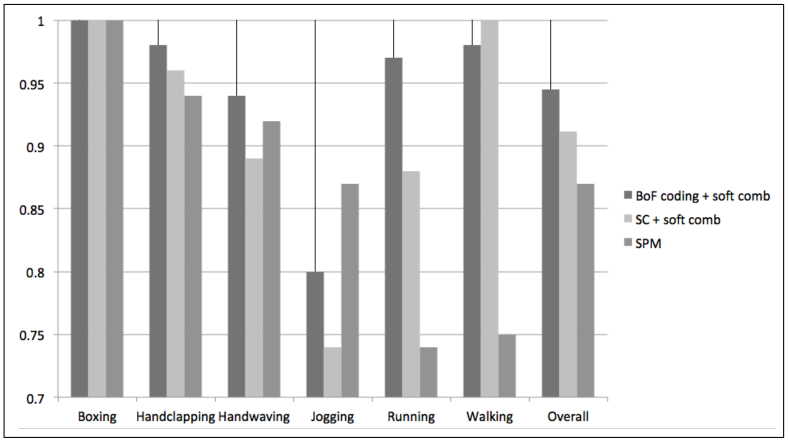


Figure 4: Average accuracies against KTH action classes.

# 8 Experiments

## 8.1 on KTH Action Dataset

KTH action dataset [11] is one of the largest video clips dataset for human actions. It includes 599 clips varying in six different classes: boxing, hand clapping, hand waving, jogging, running and walking. The scene includes indoors and outdoors with different figures. In order to avoid large memory consumptions, all pixel data are transformed into 8-bit greyscale values in the actual testing. In our experiments we simply choose 299 clips as training samples and the others as a testing set. A $3 \times 3 \times 3$ spatial-temporal pyramid is build on every video, weight are assigned as $[\frac{1}{4}, \frac{1}{4}, \frac{1}{2}]$. Non-linear SVM training process is conducted using LibSVM, 5-fold cross validation is also done on training set.

We designed three sets of experiments on KTH set: (1) Combining BoF coding with our proposed method, (2) Combining SC coding with our proposed method, (3) standard SPM. Fig.4 shows the average accuracies for each class and average overall performance. Our proposed methods perform better than the standard SPM. The best result is achieved by combining SPM coding with our method: 94.5% overall accuracy.

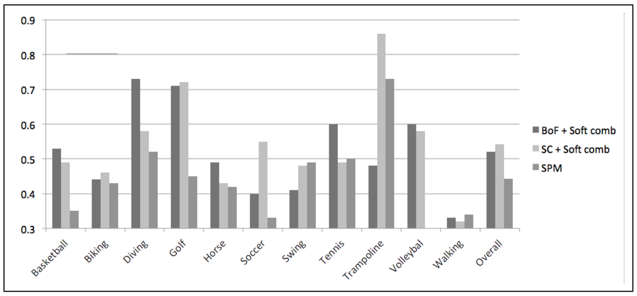


Figure 5: Average accuracies against Youtube action classes

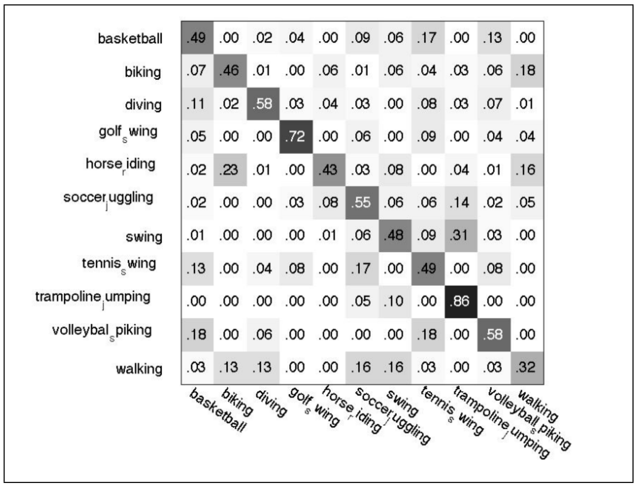
### 8.2 on Youtube Action Dataset

We also evaluated our methods on the more challenging dataset, the YouTube Action Dataset. The dataset introduced by [9] contains 11 different categories: basketball shooting, biking/cycling, diving, golf swinging, trampoline jumping, volleyball spiking, and walking with a dog. There are about 100 clips for every action group. The dataset is very challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. We performed our experiments on total 1587 video clips. As some of the videos involve very fast motions, The parameters of Gabor detector are set as $\sigma = 3$, $\tau = 2$. We first converted all video frames to grey scale to avoid extra memory consumption, and also limited number of raw descriptors to 400 per video.

Similar to the KTH setup, we evaluated $3 \times 3 \times 3$ spatial-temporal pyramid on each video, with weight vector $[\frac{1}{4}, \frac{1}{4}, \frac{1}{2}]$, with three different combining strategies: (1) BoF coding with our proposed method, (2) SC coding with our proposed method, (3) standard SPM. Fig.5 shows the average accuracies for each class and average overall performance. Again, our proposed methods perform better than the standard SPM. The best result is achieved by combining SC coding with our proposed method: 54.2%. Fig.6 shows the confusion matrix based on SC coding with our method. The results for classifying "walking with dog" has very low accuracies for all these three approaches, which can be explained by the fact that the video clips are of low quality, at the same time the motions in that group are relatively slower than those in the other groups, the cuboids extracted is not large enough to capture the motions.

## 9 Conclusions

We proposed a novel approach for combining descriptors extracted from spatial-temporal pyramid. The approach showed better performance than the standard SPM approach on KTH action set and Youtube action dataset. At the same time, the proposed approach can be separated into independent small processes, which is suitable for multicore and computer cluster processors. The overall performance on Youtube dataset is not satisfactory in compared with 75.2%, the state-of-the-art performance reported in [3], mainly because the size of cuboids in descriptors extraction stage is fixed, it failed to pick up motions with varies speed. That is: with too large cuboids in both spatial and temporal scale, much motion caused by noises are captured, with smaller cuboids might cause important motions rejected.



| | basketball | biking | diving | golf$_s$wing | horse$_r$iding | soccer$_j$uggling | swing | tennis$_s$wing | trampoline$_j$umping | volleyball$_s$piking | walking |
|---|---|---|---|---|---|---|---|---|---|---|---|
| basketball | .49 | .00 | .02 | .04 | .00 | .09 | .06 | .17 | .00 | .13 | .00 |
| biking | .07 | .46 | .01 | .00 | .06 | .01 | .06 | .04 | .03 | .06 | .18 |
| diving | .11 | .02 | .58 | .03 | .04 | .03 | .00 | .08 | .03 | .07 | .01 |
| golf$_s$wing | .05 | .00 | .00 | .72 | .00 | .06 | .00 | .09 | .00 | .04 | .04 |
| horse$_r$iding | .02 | .23 | .01 | .00 | .43 | .03 | .08 | .00 | .04 | .01 | .16 |
| soccer$_j$uggling | .02 | .00 | .00 | .03 | .08 | .55 | .06 | .06 | .14 | .02 | .05 |
| swing | .01 | .00 | .00 | .00 | .01 | .06 | .48 | .09 | .31 | .03 | .00 |
| tennis$_s$wing | .13 | .00 | .04 | .08 | .00 | .17 | .00 | .49 | .00 | .08 | .00 |
| trampoline$_j$umping | .00 | .00 | .00 | .00 | .00 | .05 | .10 | .00 | .86 | .00 | .00 |
| volleybal$_s$piking | .18 | .00 | .06 | .00 | .00 | .00 | .00 | .18 | .00 | .58 | .00 |
| walking | .03 | .13 | .13 | .00 | .00 | .16 | .16 | .03 | .00 | .03 | .32 |

Figure 6: Confusion matrix for youtube actions based on SC coding with proposed method.

In the future work, we would like to focus on optimizing the descriptor extraction process to improve the performance on challenging datasets.

# References

[1] Chih-Chung and Chih-Jen Lin. Libsvm: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjin/libsvm.

[2] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005.

[3] Nazli Ikizler-Cinbis and Stan Sclaroff. Object, scene and actions: Combining multiple features for human action recognition, 2010. In ECCV.

[4] H Jhuang, T Serre, L Wolf, and T Poggio. A biologically inspired system for action recognition. *IEEE 11th International Conference on Computer Vision (2007)*, 1(2): 1–8, 2007.

[5] Ivan Laptev. On space-time interest points, 2005. IJCV, 64:107-123.

[6] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies, 2008. In Conference on Computer Vision & Pattern Recognition, Jun 2008.

[7] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, 2006. In CVPR (2), pages 2196-2178.

[8] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In *NIPS'06*, pages 801–808, 2006.

[9] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild", 2009.

[10] J. Niebles and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words, 2008. IJCV, 79:299-318.

[11] Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach, 2004. In ICPR.

[12] Jianchao Yang, Kai Yu, Yihong Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1794–1801, 2009.

# Facial point detection in visible and thermal imagery

Brais Martinez
http://ibug.doc.ic.ac.uk/people/bmartinez

Department of Computing
Imperial College London

A precise localisation of a set of facial points can contribute to a large degree to many facial analysis problems. Traditionally, this challenging problem has been tackled through an exhaustive search approach. A set of classifiers account for distinguishing between the local texture around the facial points and its surrounding area. Then, the classifiers are evaluated over regions of interest to obtain the detection. It is also common to use a shape model to code the spatial relations between the different points[1]. The potential presence of multiple positives per point and the high computational complexity of exhaustive search are natural limitations to this methodology. As an alternative, several works have proposed an estimation-based approach. In these works the facial points are located iteratively. At each round, an inference of the point location is obtained directly from local image textures, and a shape restriction/correction can be applied over the estimation output [2]. This approach is potentially very fast and yields only one estimate. As a counterpart, erroneous estimates can greatly affect the iterative process, it is possible to converge to local minima or to have loops in the estimates, and an accurate first estimate is needed. In [3] we propose an iterative, estimation-based exploration, where the tested locations are sampled stochastically over a region of interest dynamically defined by the estimates obtained up to date. The obtained algorithm can detect reliably facial points with low computational complexity. We complement these search methodologies with a probabilistic shape model. We code the spatial relations using a Markov Network. With this model, we are capable of identifying anthropomorphically unfeasible estimates. For these cases, a correction based on the feasible estimates can be computed.

**Brais Martinez** received a Ph.D. in Computer Science from the Universitat Autonoma de Barcelona in 2010. He is currently a Research Associate at the Intelligent Behaviour Understanding Group led by Maja Pantic at the Imperial College of London.

# References

[1] B. Martinez, X. Binefa, M. Pantic. Facial Component Detection in Thermal Imagery. In *IEEE Int'l Conf. Computer Vision and Pattern Recognition - Workshops*, 2010.

[2] M.F. Valstar, B. Martinez, X. Binefa, M. Pantic. Facial Point Detection using Boosted Regression and Graph Models. In *IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2010.

[3] B. Martinez, M.F. Valstar, X. Binefa, M. Pantic. Local Evidence Aggregation for Regression Based Facial Point Detection. Submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

# Segmentation and shape classification of nuclei in DAPI images

Violet Snell
V.Snell@surrey.ac.uk
William Christmas
W.Christmas@surrey.ac.uk
Josef Kittler
J.Kittler@surrey.ac.uk

Centre for Vision, Signal Processing
and Speech
University of Surrey
Guildford GU2 7XH, UK

**Abstract**

This paper addresses issues of analysis of DAPI-stained microscopy images of cell samples, particularly classification of objects as single nuclei, nuclei clusters or non-nuclear material. First, segmentation is significantly improved compared to Otsu's method [5] by choosing a more appropriate threshold, using a cost-function that explicitly relates to the quality of resulting boundary, rather than image histogram. This method applies ideas from active contour models to threshold-based segmentation, combining the local image sensitivity of the former with the simplicity and lower computational complexity of the latter.

Secondly, we evaluate some novel measurements that are useful in classification of resulting shapes. Particularly, analysis of central distance profiles provides a method for improved detection of notches in nuclei clusters. Error rates are reduced to less than half compared to those of the base system, which used Fourier shape descriptors alone.

## 1 Introduction

We consider improving the automated processing of DAPI-stained cervical smear images, for an existing diagnostic screening application. DAPI is a fluorescent stain which binds strongly to DNA, allowing visualisation of the cell nucleus. For diagnosis of potential chromosomal abnormalities, the DAPI images are combined with FISH[1] markers in identified areas of interest.

Objects identified as a single nucleus would be expected to have a single set of chromosomes, and larger numbers of FISH signals would indicate potential abnormality, requiring further investigation. Slides also include clusters of partially overlapping nuclei, which are further processed by more complex segmentation algorithms to determine a possible split between them, and assign the corresponding FISH signals. As the incidence of single nuclei in each slide is around twice that of clusters, the early separation of classes avoids a significant computational load in additional segmentation.

The process starts with identification of regions of interest and segmentation of the nuclear objects, improvements to which are described in section 2. Classification into single,

[1] Fluorescence in situ hybridization (FISH) is a technique for locating specific DNA sequences on chromosomes using fluorescent chemical markers.

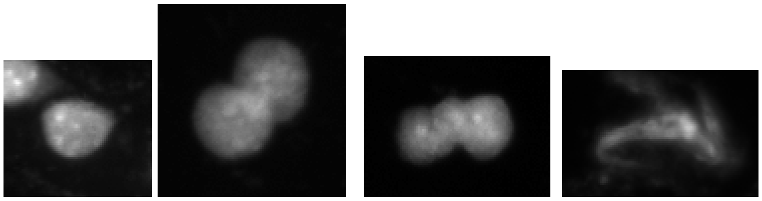

(a) Single (b) Double cluster (c) Triple cluster (d) Debris

Figure 1: Examples of each object type

cluster or debris objects is a supervised problem, with relevant region cut-outs of over 14,000 DAPI images provided with manually assigned category labels. We find a set of feature measurements that improves this classification, as described in Section 3. Some examples of each object type are shown in Fig. 1.

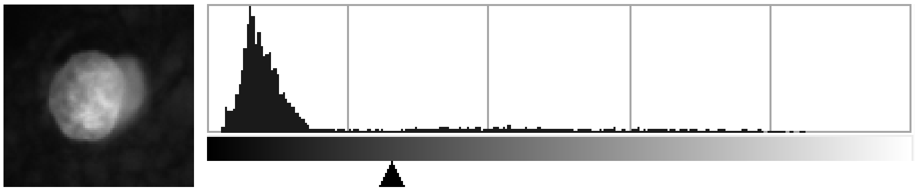
# 2 Threshold selection

## 2.1 Background

In the current implementation, object boundaries are obtained by thresholding the source images, with a separate threshold for each object calculated using Otsu's method [5], which is very widely used for segmentation tasks. On examination of the resulting masks, we note that there is a significant incidence of poor segmentation within the data set. The outlines are frequently wiggly, and sometimes quite far from the edge of the object (some examples are shown in Fig. 3). As the decision on object class is largely based on the shape of the segmented outline, it is important to improve these so that they match the actual boundary as closely as possible.

In most of the problematic cases the cause is a threshold that is set too high, sometimes by as much as 20 to 30 8-bit levels. We find an explanation for this in [4], which demonstrates that the method is inherently biased towards the class with the higher variance. In our image set the foreground nuclear objects exhibit significantly stronger texture, and therefore variance, than the background which is mostly black and plain. This also explains why the problem is particularly acute for clusters, which may contain constituent nuclei of quite different brightness levels, and overlapping zones which are brighter than either, increasing the variance.

Fortunately the problem can definitely be solved by choosing a different threshold. Very good shapes can be obtained on cell and nucleus images using thresholding methods, so we do not have to resort to more complex and computationally intensive methods of segmentation [6]. Threshold selection methods based on histogram properties prove unsuitable for this data set, as the histograms are noisy and very variable. Simple clustering methods also fail to improve on the base-line. Both of these approaches make certain assumptions about the underlying distributions of pixel values, which do not hold for our type of images. Fig. 2 shows an example histogram, which illustrates the difficulty of selecting the correct threshold based on the grey-level distribution.



(a) Image

(b) Histogram

Figure 2: Example image and its histogram. The optimal manually selected threshold is indicated by the triangle.

## 2.2 Method

To automatically find the best threshold we consider what qualities would be sought in a boundary by a human operator faced with the same task: a reasonable match with the visual edge of the object, and *a priori* knowledge of its generally elliptical shape. We therefore use two measurements to assess each candidate threshold: the average gradient across the boundary, to assess the matching of edge position, and the shape ratio of area to the square of the perimeter, which reflects smoothness of the outline.

The shape ratio is defined as

$$S(t) = A(t)/l^2(t) \tag{1}$$

where $A$ is area enclosed by contour and $l$ is length of the contour resulting from threshold $t$. It penalises thresholds that are too high, as they slice through the textured foreground resulting in very wiggly outlines which accumulate a lot of perimeter length for relatively little area. This ratio is a common morphological measurement in analysis of nuclear images [3], sometimes referred to as circularity or compactness.

The gradient measurement is normalised by average brightness of the object, to compensate for changes in illumination between images:

$$G(t) = \frac{\sum\limits_{p \in C(t)} |I(p+\delta) - I(p-\delta)|}{l(t) \cdot \bar{I}} \tag{2}$$

where $\delta$ is a unit vector perpendicular to the contour $C(t)$ at point $p$, $I(p)$ are image pixel values and $\bar{I}$ is the average brightness of object pixels. The gradient is clearly highest when the boundary matches the steepest part of the grey-scale slope around the object.

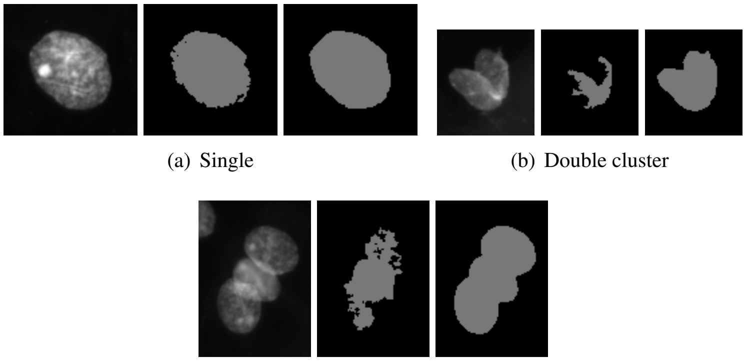
The two measurements are combined in a weighted sum to produce a single quality metric:

$$T_{opt} = \arg\max_t \{G(t) + w_S \cdot\ S(t)\} \tag{3}$$

with weight $w_S$ estimated from the ratio of sample variances of the two parameters across the image set as

$$w_S = \sqrt{\frac{\sum_i Var_t[G_i(t)]}{\sum_i Var_t[S_i(t)]}} \tag{4}$$

where $i$ is the image index within the data set. As there is no specific reason for either of the measurements to have more influence on the outcome than the other, this weight balances the contributions from each measurement to overall cost-function variance.



(a) Single

(b) Double cluster

(c) Triple cluster

Figure 3: Examples of improved threshold selection, showing grey-scale source image, mask from Otsu's method, and the improved mask for each one.

Thresholds from the original minimum variance estimate down to 30 grey levels below it are evaluated. This range is established empirically, and may not be appropriate for other image sets. We find that an exhaustive search is required, as the effects of noise and texture produce substantial variation in the quality metric at intermediate levels, precluding the use of search optimisation techniques.

Lowering the threshold presents an additional challenge, as the object may join up with other objects or sections of objects which are present nearby within the image cut-out. To avoid this eventuality the search is terminated early if this condition is detected. The condition needs to be distinguished from gradual increase in object area or joining up with parts of the object itself which had been separated by the excessively high original threshold. This can be done by monitoring the number of pixels added by each lowering of threshold - a large step increase indicates coalescing with another object, as the previously disjoint contours are bridged by a pixel of lower intensity.

## 2.3 Results

We estimate using Eq. (4) that for optimal balance between searching for a rounded shape, but also matching the object's edge within the image, weights of around $w_S = 12$ produce the best overall goodness measure. This retains reasonably sharp notches in clusters, to help distinguish them from single nuclei, but corrects most of the deformities introduced by threshold bias, as illustrated in Fig. 3. The process is not particularly sensitive to the precise values of the weights; for example, a 10% increase in shape weight affects the resulting threshold in less than 4% of cases.

The improvement in mask shape resulting from this process is consistent and visually apparent. As there is no such thing as 'perfect' segmentation (unless the source images are artificially generated), it is difficult to provide an objective ground-truth and measure the improvement numerically. Any automatic method of assessing the resulting boundary would likely involve the very measures that are being optimised by the process. However, as illustrated in Fig. 4, there is a very large proportion of images that are improved by a moderate refinement of threshold, and a considerable number that require a large adjustment.

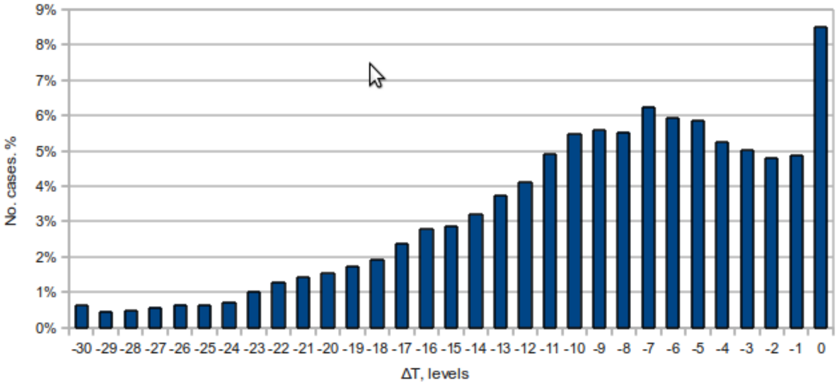


Figure 4: Frequency of threshold changes

## 2.4 Discussion

In choosing the threshold there is sometimes a conflict between obtaining a good outline and keeping it separate from other nearby objects. This could be addressed by an adaptive method, raising the threshold for the part of the boundary that is detected as being close to another object, while keeping it low elsewhere. However, we do not find this necessary in this particular application.

The method described here borrows from active contour models [1] the idea of combining a shape measurement (often referred to as *internal energy*) with object features in the form of gradients (*external energy*). However, by reducing the search space to one dimension, we no longer require an iterative approximation or differential equations. Both the complexity of implementation and computational load involved are greatly reduced, while retaining the flexibility to define an image-based measure and adjust weights in a way that is suited to the particular application. While the computational cost is higher than basic histogram-derived methods, it is still very minor in terms of the overall image-analysis process. As a very large number of nuclei is processed for each patient sample, the cost does need to be kept low.

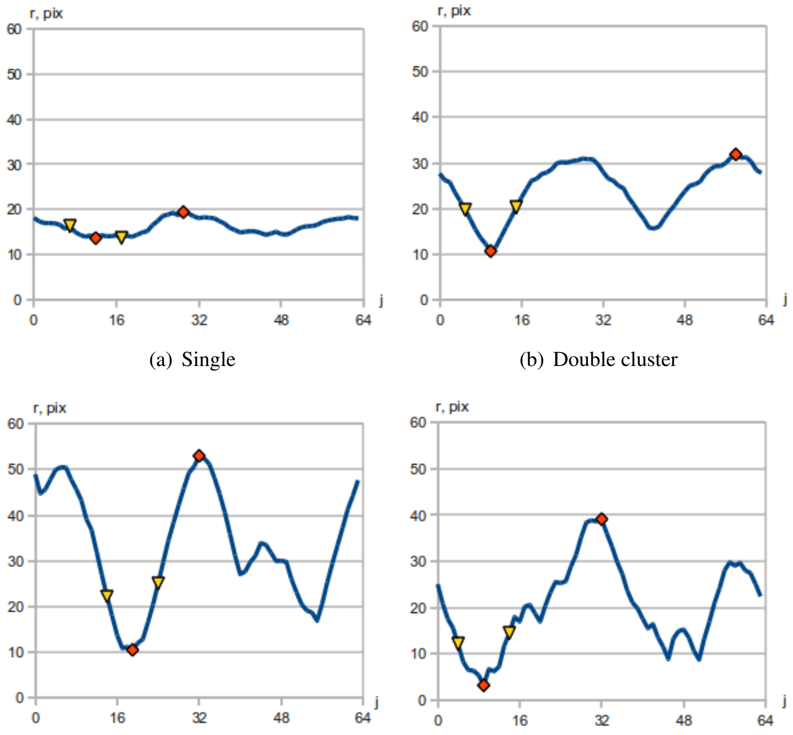
# 3 Discriminating Features

## 3.1 Background

Many different measurements and features have been used for automated analysis of fluorescent nuclear images [3]. In this study we are primarily interested in discriminating single nuclei from touching or overlapping clusters, as well as rejecting non-nuclear material; this decision is mostly based on the shape of the object. Our labelled set includes 7376 examples of single nuclei, 4550 of clusters, and 2178 of debris.

A common approach to shape classification is the use of Fourier Descriptors of the central distance profile [7]. Distance from the object centroid to the edge is calculated at points equally spaced around the perimeter, separated by regular arc-length intervals. The profiles are then subjected to the Fourier frequency transform, with resulting coefficients used as classification features. The existing application uses 10 lowest terms of the 64 point transform, and this is used as the base-line for later comparisons.

### 3.2 Method



(a) Single

(b) Double cluster

(c) Triple cluster

(d) Debris

Figure 5: Typical examples of central distance profiles for different object types, showing min and max positions (diamonds) and sides of lowest trough (triangles).

We suggest that in this case some valuable information can be derived from direct analysis of the central distance profiles, rather than their Fourier coefficients. We define the distance from the object centroid $\bar{p}$ for points $p_j \in C$ along the contour $C$ as

$$r(j) = \|p_j - \bar{p}\|_{L2}, j \in \{1..K\} \tag{5}$$

where K points are spaced at equal arc-lengths around the contour. We use 64 points for compatibility with FFT processing and comparable figures to the existing system. The equally-spaced points are obtained by re-sampling integer pixel positions from the segmentation contour. Some typical examples of these profiles for the different classes in this study are shown in Fig. 5.

Characteristically, most single nuclei show a very small amount of variation in their profiles, when compared to the other object types. Most variation in single nucleus profiles comes from the elongation of the elliptical shape, which is smooth, whereas clusters tend to have much sharper notches and angles between the nuclei, which result in much deeper, but also sharper, troughs in the profiles. Debris profiles tend to be much noisier and generally less consistent in their shape.

Based on the observations above, two types of measurement suitable for classification are derived from the profiles: the first is a ratio between the minimum and the maximum of the profile, indicating the relative depth of the biggest trough within the profile. This is equivalent to the $R_{min}/R_{max}$ ratio that is sometimes used in nucleus analysis [3]. The second set of measurements assesses the steepness of the sides of the lowest trough, by taking gradients either side of the minimum. To reduce the effect of noise, these are taken as differences from the minimum to values several points away from the minimum position; it has been established experimentally that 5 points (out of the total of 64) is optimal for overall classification performance on this data set. The two gradients, from left and right, are sorted into the larger and the smaller, as no significance can be attached to the orientation. Both are scaled by the DC term of the Fourier transform, representing average radius, to provide size invariance.

To improve the accuracy, we include a number of other measurements in the feature vector. Only the 6 lowest Fourier descriptor terms are found to carry useful information (with higher harmonics not bringing any improvement in classification). Additionally, the feature vector includes the shape ratio (defined in Section 2), as well as object area and perimeter on their own; and another morphological measure based on the relative difference in area between the object and its convex hull.

$$C = \frac{A_{hull} - A}{A} \tag{6}$$

This concavity measure is aimed particularly at identifying the debris objects, which frequently have shapes with random protrusions and irregular edges. We also find it beneficial to include features derived from the image content, such as mean and standard deviation of pixel values within the object boundary; cross-boundary gradient (also described in Section 2) and a spot filter energy total. The last two values are normalised by the average brightness to compensate for variations in overall luminosity.

We use Support Vector Machine (SVM) classifiers with a radial basis kernel, provided by LibSVM wrapper for WEKA [2], with 10-fold cross-validation to obtain stable accuracy figures.

## 3.3 Results

The central distance trough measurements provide a very strong contribution to distinguishing single nuclei from other object types. Even when used on their own, without any additional measurements, accuracy of 96.3% can be achieved on this particular data set.

With a total of 17 features we can obtain accuracy of 98.7% for identification of single nuclei, or 98.0% for the full 3-class separation (see Table 2 for full confusion matrix). This compares favourably with the currently used method based on 10 complex pairs of Fourier descriptors, which was able to identify single nuclei with accuracy of 97.1%, but was not suitable for separating clusters from debris (overall error rates of around 7.8%). The standard deviation of the results obtained from multiple cross-validation experiments is 0.25%, and the results are summarised in Table 1.

The feature set is found to be robust to noise, particularly to the effects of poor segmentation: the accuracy achieved on images segmented with the original Otsu's threshold is

| | 2-class (Single vs others) | 3-class (Single, Cluster, Debris) |
|---|---|---|
| 10 x complex FDs | 97.1% | 92.2% |
| Proposed feature set | 98.7% | 98.0% |

Table 1: Accuracy rate comparison

| Classified as→<br>Actual class ↓ | Single | Cluster | Debris |
|---|---|---|---|
| Single | 99.0% | 0.8% | 0.2% |
| Cluster | 2.1% | 96.8% | 1.1% |
| Debris | 1.5% | 2.5% | 96.0% |

Table 2: Confusion matrix from 10-fold cross-validation using proposed feature set

only 0.2% lower for 2 classes, and 0.3% for 3 classes, despite quite strong degradation of boundary shape in many cases.

### 3.4 Discussion

The error rate of new proposed method is significantly lower than that achieved by the use of Fourier descriptors alone. For an application aimed at filtering out just single nuclei, the error rate is less than half (1.3% vs 2.9%), a statistically significant difference of over 6 standard deviations. We also develop a selection of features which is able to differentiate clusters from non-nuclear debris in the same step. Use of statistics and measurements of the image content (rather than shape alone) definitely contributes towards this ability, as debris is generally characterised by a more smeary texture than genuine nuclear objects.

The final confusion matrix is reasonably well balanced, with no one combination a dominating source of error. Although the results quoted are for RBF SVM classifiers only, several other types of classifier were evaluated (k-nearest neighbour, perceptron neural network, Bayesian estimators and other kernels for SVM), but none could match the performance of RBF SVM on this data set.

## 4 Conclusions

We describe two areas of improvement in an application which promises great advances in accuracy and availability of early detection of cancers and precancerous conditions.

While the segmentation improvements described in Section 2 have a relatively small impact on classification performance, the general approach is potentially useful in other segmentation applications. Although the computational cost of assessing each threshold is relatively high when compared to histogram methods, the connection to spatial layout of the pixels facilitates a much more meaningful decision in terms of the object of interest. On the other hand, the restriction of search space to one dimension allows a favourable complexity comparison to two-dimensional segmentation methods which optimise some measure of a boundary's desirability. Moreover, the method is extremely flexible, allowing a choice of weights and perhaps different shape and edge measures that are more suitable for a particular type of images.

It is difficult to assess the precise contribution to accuracy from any one feature, as it is the combination of all the measurements in the feature vector that determines the classifier's overall generalisation ability. Among the features that have been evaluated, direct measurements on the central distance profile are notable for their novelty and efficacy. While Fourier analysis of these profiles is widely used for general shape matching, the new measures are more tailored to the specific task of detecting notches between overlapping or touching nuclei within a cluster. Similar techniques could also be used in other notch detection applications.

## 4.1 Acknowledgements

The authors acknowledge the support from Ikonisys, Inc. in providing the large quantities of labelled data, as well as their expertise and support in the medical and biological aspects of the subject. This research is funded by EPSRC.

# References

[1] D. Cohen. On active contour models and balloons. In *CVGIP: Image Understanding*, volume 53, pages 211–218. Academic Press, March 1991.

[2] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1), 2009.

[3] S. Heynen, E.A. Hunter, and J.H. Price. Review of cell nuclear features for classification from fluorescence images. *Proceedings of the SPIE - The International Society for Optical Engineering*, 3921:54–65, 2000. ISSN 0277-786X.

[4] Z. Hou, Q. Hu, and W.L. Nowinski. On minimum variance thresholding. *Pattern Recognition Letters*, 27(14):1732–1743, 2006. ISSN 0167-8655.

[5] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9(1):62–6, 1979/01/. ISSN 0018-9472.

[6] M Sezgin and B Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13:146–168, 2004.

[7] D Zhang and G Lu. A comparative study on shape retrieval using Fourier descriptors with different shape signatures. In *International Conference on Intelligent Multimedia and Distance Learning*, pages 1–9, Fargo, ND, USA, June 2001.

Proc. of the 3rd British Machine Vision UK Student Workshop (BMVC'11 WS, Dundee, Scotland, 2nd September, 2011)

# Texture Segmentation by Evidence Gathering

Ben Waller
bmw306@ecs.soton.ac.uk
Mark Nixon
msn@ecs.soton.ac.uk
John Carter
jnc@ecs.soton.ac.uk

School of Electronics and Computer Science
University of Southampton
Southampton, UK

**Abstract**

A new approach to texture segmentation is presented which uses Local Binary Pattern data to provide evidence from which pixels can be classified into texture classes. The proposed algorithm, which we contend to be the first use of evidence gathering in the field of texture classification, uses Generalised Hough Transform style R-tables as unique descriptors for each texture class and an accumulator is used to store votes for each texture class. Tests on the Brodatz database and Berkeley Segmentation Dataset have shown that our algorithm provides excellent results; an average of 86.9% was achieved over 50 tests on 27 Brodatz textures compared with 80.3% achieved by segmentation by histogram comparison centred on each pixel. In addition, our results provide noticeably smoother texture boundaries and reduced noise within texture regions. The concept is also a "higher order" texture descriptor, whereby the arrangement of texture elements is used for classification as well as the frequency of occurrence that is featured in standard texture operators. This results in a unique descriptor for each texture class based on the structure of texture elements within the image, which leads to a homogeneous segmentation, in boundary and area, of texture by this new technique.

## 1 Introduction

Texture is an important property of images, representing the structural and statistical distribution of elements throughout the image. Images can contain a single texture, for example an image of a brick wall, or multiple textures of varying distribution throughout the image such as a satellite image containing textures representing urban areas, fields, forest and water. Image segmentation by texture has a wide range of applications, from analysis of medical images [7] to remote sensing [8]. Additionally there are industrial applications of texture analysis which include visual inspection and defect detection [11].

Texture descriptors can be divided into two types; structural and statistical. Structural approaches apply a transform, such as the Fourier transform, to the image and then obtain a set of measurements which describe the texture [13]. Statistical approaches classify textures by measuring a property of the image and comparing the rate of occurrence of this to that obtained from training images. A well-known example of this is the co-occurrence matrix

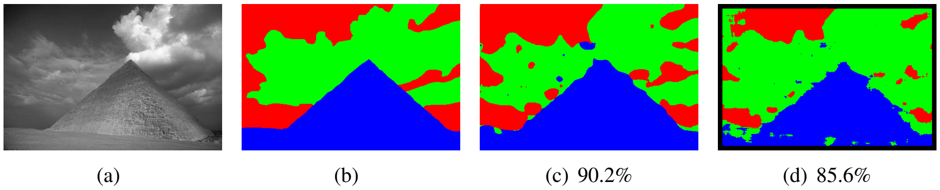


Figure 1: BSDS Pyramid: a) original image; b) manual segmentation; c) segmentation using the EGTS algorithm; d) segmentation using histogram comparison.

[5], developed by Haralick *et al*. in 1973, where the number of pairs of pixels separated by a particular distance with a specific intensity are counted. The matrix of number of pairs is used as the texture descriptor for classification. Another popular and more modern operator is Local Binary Patterns (LBP) which uses the intensity at a point to threshold surrounding pixels to produce a code representing the texture pattern at that point [16]. A histogram of the texture codes is used as the texture descriptor. Both operators are well established and the LBP has continued to receive significant attention over the years with many published extensions and applications [2, 4].

Texture classification typically relies on using a measure of similarity between a texture sample and known texture classes to classify the sample. Segmentation is usually performed either by classification of each pixel separately via a windowing method [10] or by an iterative split and merge algorithm [14]. Both methods are computationally inefficient since they require the same information to be processed multiple times.

Unlike texture, the field of template matching in computer vision has benefited from the use of evidence gathering approaches, most notably present in the Hough Transform, which accumulates votes for each pixel every time evidence indicating the presence of a desired shape at that point is found [6]. The Hough Transform was extended further to accommodate many different shapes such as circles and ellipses, and a generalised form of the Hough Transform was developed to be able to search the image for any arbitrarily shaped object [1]. The advantages of this evidence gathering method include scale and rotation invariance and resilience to noise and occlusion.

We describe a new approach to texture segmentation which determines texture class through evidence gathering. The Local Binary Pattern (LBP) operator is used as the mechanism to gather evidence and Generalised Hough Transform (GHT) style R-table and accumulators are employed to store this evidence and vote accordingly. This new approach is the first use of evidence gathering to determine texture and has been demonstrated to give very good results for texture segmentation, as illustrated by Figure 1, while maintaining smooth texture boundaries and minimising noise. The proposed algorithm will be referred to henceforth as the Evidence Gathering Texture Segmentation (EGTS) algorithm. To show the advantages of this new technique our evaluation compares results from EGTS with a pure LBP algorithm known as histogram comparison. This demonstrates that using evidence gathering in conjunction with an established texture descriptor can yield improved segmentation accuracy.

Section 2 summarises the existing work on texture classification and evidence gathering that is used as the basis for this new method and Section 3 describes the new EGTS algorithm. Section 4 provides experimental results and Section 5 concludes the paper.

# 2 Background

## 2.1 Local Binary Patterns

Local Binary Patterns are texture descriptors which label individual pixels in an image with a code corresponding to the local texture pattern surrounding the pixel. First introduced by Ojala *et al*. [15], the earliest form of the LBP used the centre pixel of a 3x3 grid, $g_c$, to threshold each of the eight neighbouring pixels $g_0$ to $g_7$. This produced an eight bit binary code which represents the texture element present at that point. The LBP was later extended to give the texture pattern for $P$ points on a circle of radius $R$. It was observed that certain fundamental patterns make up the majority of all LBP patterns observed [16]. These were found to be the patterns which had at most two zero to one transitions ($U(LBP_{P,R}) \leq 2$) and are called *uniform* LBP patterns. All of the uniform patterns are labelled according to the number of '1' bits in the code. When $P$ is equal to eight, there will be ten different patterns: the uniform patterns from '0' to '8' and the pattern '9' which is the agglomeration of all other patterns. Since the sampling positions for the neighbouring points are arranged in a circle, the uniform LBP is, by nature, rotation invariant. This is because if the sample image texture is rotated, the LBP code produced will still have the same number of zero to one transitions and the same number of '1' bits, resulting in an identical uniform LBP code regardless of the order of the bits. The rotation invariant uniform LBP code for a point, $LBP^{riu2}_{P,R}$, is calculated by:

$$LBP^{riu2}_{P,R} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{if } U(LBP_{P,R}) \leq 2 \\ P+1 & \text{otherwise} \end{cases} \tag{1}$$

where

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} \left| s(g_p - g_c) - s(g_{p-1} - g_c) \right| \tag{2}$$

and

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Textures can be classified using the LBP by obtaining a histogram of the uniform patterns and using a dissimilarity metric to compare this to histograms obtained from known texture classes. Segmentation can be performed by performing classification on a pixel by pixel basis by obtaining the histogram of a region centred on the pixel.

## 2.2 Generalised Hough Transform

The Generalised Hough Transform [1] uses an evidence gathering approach to determine the location of previously defined arbitrary shapes within an image. An arbitrary shape can be described by the following set of parameters:

$$\mathbf{a} = \{\mathbf{y}, s, \theta\} \tag{4}$$

where $\mathbf{y} = (x_r, y_r)$ is the reference origin for the shape, $s$ is the scale of the shape and $\theta$ is the orientation of the shape. For each edge point $\mathbf{x}$ on the shape the gradient is calculated and the vector $\mathbf{r}$ between $\mathbf{x}$ and $\mathbf{y}$ is stored in a table called the *R-table*. The R-table contains a series of bins, each representing a range of gradients. The **r** vector for each edge point is stored in the bin that matches the calculated gradient, resulting in the R-table containing a complete description of the shape. The algorithm for using the R-table to find shapes with in an image is described by Ballard [1] as:

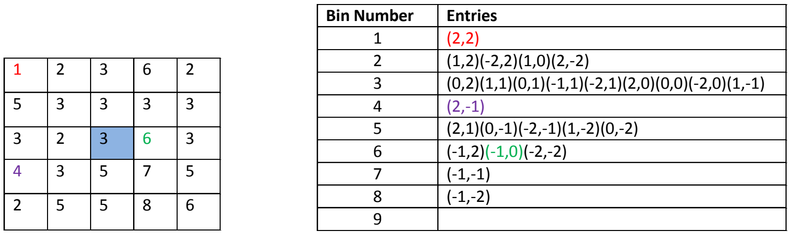
"For each edge pixel **x** in the image, increment all the corresponding points **x** + **r** in the accumulator array $A$ where **r** is a table entry indexed by $\theta$, i.e., $\mathbf{r}(\theta)$. Maxima in $A$ correspond to possible instances of the shape $S$."

# 3 Evidence Gathering Texture Segmentation

A new evidence gathering approach to texture segmentation is proposed which uses the principles of template matching present in the Generalised Hough Transform (GHT) and modifies it to match texture instead of shape. The technique exploits a property of the Local Binary Pattern (LBP) texture descriptor which is that if there is structure in the image space, there must be structure in the LBP space. Mäenpää and Pietikäinen observed in [9] that each LBP code limits the set of possible codes adjacent to it. This implies that the arrangement of LBP codes within a texture element is not random and that taking a histogram of the codes reduces the available information further to that originally lost in the LBP process. It is possible for several textures to have the same histogram, rendering such methods incapable of distinguishing between them. By storing the LBP code along with its offset to the centre of the texture region for each pixel, this structural information is not lost and a unique descriptor is produced which can be used in the classification and segmentation of images. The descriptor is unique because it can be used to regenerate the array of LBP codes that represent the texture sample, unlike a histogram of LBP codes which cannot. The new algorithm is thought to be the first use of evidence gathering in texture segmentation and achieves high efficiency by transferring the principles of low computational complexity present in the GHT method to the field of texture analysis.

## 3.1 Method

As with the GHT, before sample images can be analysed, an R-table must be generated for each known texture class. This describes the structure and composition of a section of the texture and is used to classify the texture class of the sample images. Sub-images, or cells, are taken from the training images and the LBP code is calculated for each pixel within the cell. The R-table contains a number of bins equal to the number of different LBP codes that exist for the version of the LBP that is being used. For LBP $P$ values of eight, the number of bins will be ten; one for each of the nine uniform LBP codes and a miscellaneous bin for all other codes which are not classified as one of the uniform patterns. For each pixel in the cell an entry is submitted to the bin corresponding to the LBP code for that pixel. The entry is a two dimensional vector **r**=$(x_r,y_r)$ representing the translation from the pixel to the reference point of the cell. This reference point is usually chosen to be the centre. In Figure 2, the top left pixel (shown in red) in the cell has an LBP code of '1' and so an entry is made in the '1' bin with the vector (2,2) which maps the top left pixel to the centre. The size and number of cells taken from the training images are not fixed and these parameters can be tailored for different applications. The size of the cell should be large enough to contain at least one full example of the repeating pattern in the texture. Having multiple cells for each texture class will provide more evidence for classification during the segmentation process.



| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 6 | 2 |
| 5 | 3 | 3 | 3 | 3 |
| 3 | 2 | 3 | 6 | 3 |
| 4 | 3 | 5 | 7 | 5 |
| 2 | 5 | 5 | 8 | 6 |

(a) Cell

| Bin Number | Entries |
|---|---|
| 1 | (2,2) |
| 2 | (1,2)(-2,2)(1,0)(2,-2) |
| 3 | (0,2)(1,1)(0,1)(-1,1)(-2,1)(2,0)(0,0)(-2,0)(1,-1) |
| 4 | (2,-1) |
| 5 | (2,1)(0,-1)(-2,-1)(1,-2)(0,-2) |
| 6 | (-1,2)(-1,0)(-2,-2) |
| 7 | (-1,-1) |
| 8 | (-1,-2) |
| 9 | |

(b) R-table

Figure 2: Example LBP values for a 5x5 pixel cell and corresponding R-table. The reference point for the cell is shaded in blue.

The following equation is used to calculate the R-table entry for each pixel $\mathbf{x}$ = *(x,y)* in a cell of centre $\mathbf{c}$ = $(x_c, y_c)$:

$$\mathbf{r} = \mathbf{c} - \mathbf{x} \tag{5}$$

where the R-table index is the LBP code calculated by Equation 1 at the point $\mathbf{x}$ = *(x,y)*.

As with the GHT, evidence is stored in an array called the accumulator, and a separate accumulator is maintained for each of the texture classes that are being searched for. In the segmentation of sample images, the LBP code for each pixel in the entire image is calculated. The entries in the R-tables represent the possible locations of the current pixel relative to the reference point of the cell. For the example in Figure 2, if a pixel in the sample image had an LBP code of '6', it could correspond equally to any of the three positions within the cell also with that LBP code. For each in turn, votes are made for the area that would cover the entire cell positioned on that pixel. Rephrasing Ballard, the algorithm becomes: For each pixel $\mathbf{x}$ in the image, increment all the corresponding points in a cell centred on the point $\mathbf{x}$ + $\mathbf{r}$ in the accumulator array $A$ where $\mathbf{r}$ is a table entry indexed by the LBP code at point $\mathbf{x}$. Maxima in $A$ correspond to possible instances of the texture $T$.

Voting is done in blocks rather than for individual pixels because texture covers an area and a single pixel on its own does not contain texture. The area covered by each block vote is equal to the area of the cell from which the evidence was gathered. The three block votes for an LBP code of '6' using the R-table in Figure 2(b) are shown in Figure 3. The algorithm is effectively searching the sample image for the texture structure observed in the training cell. In Figure 3, it can be seen that four of the pixels in the image were within all three possible cells for that R-table and hence these pixels have a higher probability of belonging to that texture class. The equations for calculating the coordinates of the four corners of the rectangle covering the voting block for each R-table entry, where the reference point is the centre of the cell, are as follows:

$$\text{Top left} = \mathbf{x} + \mathbf{r} + (-\frac{c_w}{2}, -\frac{c_h}{2}) \tag{6}$$

$$\text{Top right} = \mathbf{x} + \mathbf{r} + (\frac{c_w}{2}, -\frac{c_h}{2}) \tag{7}$$

$$\text{Bottom left} = \mathbf{x} + \mathbf{r} + (-\frac{c_w}{2}, \frac{c_h}{2}) \tag{8}$$

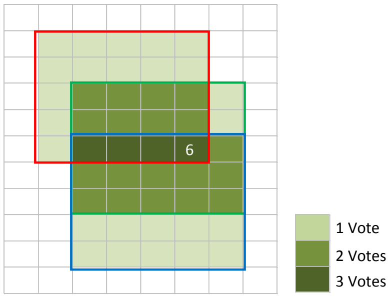


Figure 3: Accumulator showing block votes for three R-table entries, bordered by red, green and blue rectangles.

$$\text{Bottom right} = \mathbf{x} + \mathbf{r} + \left(\frac{c_w}{2}, \frac{c_h}{2}\right) \tag{9}$$

where $c_w$ and $c_h$ are the cell width and cell height respectively.

An accumulator for each texture class maintains the number of votes for each pixel for that texture. If there is more than one cell for a texture class, the votes of the subsequent cells are added to the accumulator for the first cell. When the voting process is finished, the higher the number of votes for each pixel, the higher the probability of the pixel belonging to that texture class. It is important to note that analysis of a single pixel yields evidence for many other pixels. This works because if there is structure in the texture, the LBP code at a point is related to those around it. Using a higher number of cells per texture class increases the amount of evidence used to classify pixels and leads to a higher accuracy. Segmentation is performed by filling an accumulator for each texture class and assigning each pixel to the texture class with the highest number of votes at that point.

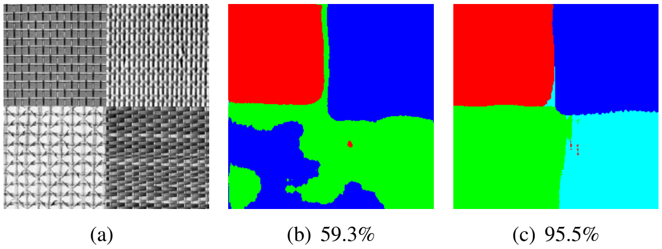
## 3.2 Extensions

### 3.2.1 Multi-scale Support

Multi-scale versions of the LBP operator can be obtained from the individual histograms of the LBP at different scales by extending the measure of dissimilarity to compare over multiple histograms. The multi-scale LBP has been demonstrated to give better results than the single scale version [16]. The EGTS algorithm can be similarly extended to support multiple scales by calculating the votes for each pixel at each scale and then adding them together. In Figure 4(b) it can be seen that not all textures are identified correctly using an LBP radius of 1, however when these results are combined with those obtained from an LBP radius of 2, as seen in Figure 4(c), a vastly improved segmentation is obtained.

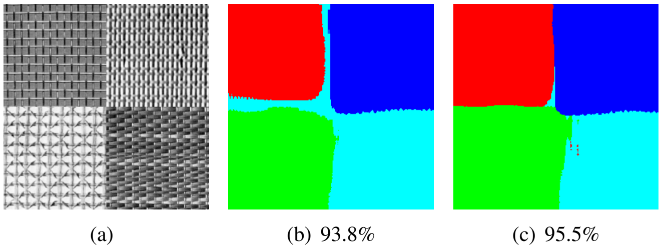
### 3.2.2 Matched Voting

An issue with the original form of the EGTS algorithm is overvoting. Since most modern LBP variants only have ten different codes many votes are made for the wrong texture since



(a) (b) 59.3% (c) 95.5%

Figure 4: Multiscale: a) original image; b) Segmentation results using LBP radius of 1 and nine cells of 32x32 pixels c) Segmentation results using LBP radius of 1 and 2 and nine cells of 32x32 pixels.



(a) (b) 93.8% (c) 95.5%

Figure 5: Matched Voting: a) original image; b) Results using radius of 1 and 2 and nine cells of size 32x32 pixels without using matched voting; c) Results under the same conditions using the matched voting extension.

there will always be an element of overlap in the code occurrence. The LBP methodology still works; there will always be more votes for a perfect sample than for a different texture, however the presence of noise or a slightly distorted texture sample can reduce the contrast of votes between texture classes. A solution to this problem is the matched voting extension. In the standard version of the algorithm the LBP code of the pixel being classified is matched to those of the training cells. However if we revisit the theory of structure present in the LBP space it is apparent that if there is also a match between the LBP codes of the neighbouring pixels in the sample image and the neighbouring pixels in the training cell there is a higher chance of the pixel belonging to that texture class. The matched voting extension awards one extra block vote per correctly matched neighbouring pixel. Tests have shown that allowing the neighbouring LBP codes to match any of the neighbouring codes in the R-table gives the best contrast increase while maintaining the rotation invariant properties and number of votes for correct textures. This means that in the example in Figure 3, the three entries in the R-table will not be treated equally and will be assigned votes dependant on how closely the structure matches. Each R-table entry is now required to contain the LBP codes for the neighbouring pixels as well as the vector from the pixel to the centre of the cell. Figure 5 shows the typical performance increase when matched voting is used instead of standard voting.

### 3.2.3 Normalisation

It can be observed that different textures have different voting strengths. This means that some textures could give a larger number of votes for an incorrect texture than another texture

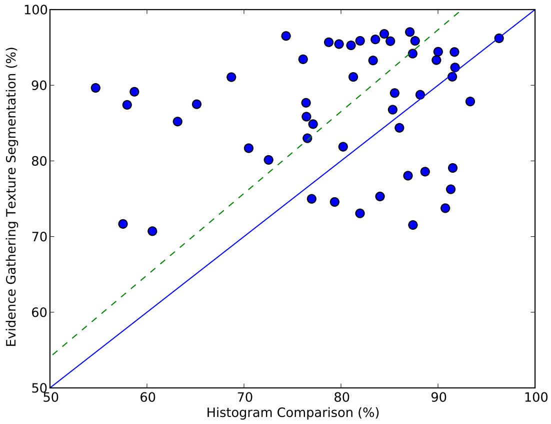


Figure 6: Segmentation accuracy of mosaics from the Brodatz subset using both the new evidence gathering algorithm and the histogram comparison algorithm. The solid blue line represents the line of equality and the dashed green line is the trend line.

could give for a correct match. This leads to cases where votes from one texture overpower those from another, distorting the segmentation results. A solution is to normalise the voting, whereby the votes from each texture are weighted according to their strength factor. One way of calculating the strength factor is to add up the total number of votes for the texture over the entire image and divide by the number of pixels. When all votes for a texture are divided by its strength factor the stronger textures will have their influence over the regions of other textures weakened, reducing the "overspill" effect. The equation for performing normalisation on an accumulator $A$ of size $w$ by $h$ is:

$$A_{norm}(x,y) = \frac{A(x,y) * w * h}{\sum_{a=0}^{w} \sum_{b=0}^{h} A(a,b)} \qquad (10)$$

If normalisation is required where one texture is weaker than the others, its use can restore the texture boundaries to their correct locations. Better results can sometimes be obtained from manual assignment of the strength factors, leading us to believe that a machine learning approach is the best way of obtaining the optimum strength factor during the training stage.

# 4 Results

## 4.1 Texture Mosaics

A subset of 27 textures from the Brodatz album [3] was used to generate 50 mosaics containing four randomly selected textures. This subset is included in the supplementary material. For each texture in the subset, the bottom right quarter was used to generate the mosaics, and the top left quarter was used to provide training data for segmentation. Segmentation was performed using the EGTS algorithm, employing LBP radii of both 1 and 2 for multi-scale

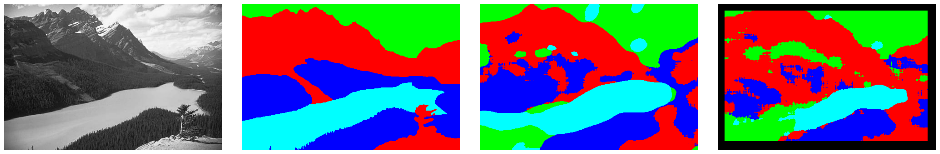

(a) (b) (c) 76.7% (d) 67.8%

Figure 7: BSDS Mountain: a) original image; b) manual segmentation; c) segmentation using the EGTS algorithm; d) segmentation using the HC algorithm.

support and segmenting using 9 cells of 32x32 pixels each. The matched voting and automatic normalisation features were also enabled. The standard method of image segmentation using a texture classification algorithm classifies each pixel individually by taking a window centred on it and performing comparison against the training data [17]. For comparison, we have used the LBP segmentation from [10] which uses this method to segment each of the 50 texture mosaics. For simplicity, this algorithm will be referred to as Histogram Comparison (HC). Results from the 50 tests are shown in the graph in Figure 6, where the preponderance of results exceeding the line of equality shows the superiority of the new approach. Segmentation accuracy was calculated by comparing the results pixel-by-pixel against the ground truth. Our EGTS algorithm achieved an average segmentation accuracy of 86.9% and standard deviation of 8.12 over the fifty tests compared with an average of 80.3% and standard deviation of 10.36 achieved by HC.

### 4.2 Real Images

Results from two real images from the Berkeley Segmentation Dataset [12] have been included. The first is an Egyptian pyramid shown in Figure 1. The results obtained from our EGTS algorithm and the standard HC algorithm are shown in Figures 1(c) and 1(d) respectively. A manual segmentation of the image is included in Figure 1(b) and the segmentations are compared to this ground truth to obtain a numerical indicator of their quality which is shown in the captions. Both algorithms provide a good segmentation of the image, however it is apparent that that the EGTS algorithm provides a much smoother boundary between the textures along with a higher segmentation accuracy. The second image is of a mountain scene and results are shown in Figure 7. Our algorithm provides a significantly better result than the HC algorithm and again features smoother boundaries between textures and lower noise within texture segments.

## 5 Conclusions

In this paper, we have presented a new method for image texture segmentation which we contend to be the first use of an evidence gathering approach in the field of texture analysis. In contrast to conventional methods which compare measurements from a sample of an image to training data to classify a single pixel, our approach compiles information gathered from each pixel into evidence to support the classification of nearby pixels into each known texture class. Each pixel is then classified into the class for which it has the most evidence. We have performed a statistical test using a subset of the Brodatz texture database and our EGTS algorithm gives a higher average performance and lower standard deviation than the HC algorithm under the same conditions. The lower standard deviation implies that in addition to performing better on average, our algorithm is also more robust. Two tests on real images from the Berkeley Segmentation Dataset show significantly higher segmentation accuracies are obtained from the EGTS algorithm. Our results also provide noticeably smoother texture boundaries and reduced noise within texture regions. The proposed EGTS algorithm is an implementation of a higher order texture descriptor; classifying texture based on the structure of the individual elements which make up the texture. Existing "low order" descriptors use the rate of occurrence of the texture elements to classify the textures, providing a descriptor which is not necessarily unique to a single texture class. By contrast, our EGTS algorithm generates a unique R-table for each texture which not only supplies information on the occurrence of texture elements, but also their structure. Further work will focus on parameter optimisation and developing a colour version of the algorithm.

# References

[1] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[2] HS. Bhatt, S. Bharadwaj, R. Singh, and M. Vatsa. On matching sketches with digital face images. In *Proc. BTAS*, 2010.

[3] P. Brodatz. *Textures: A photographic album for artists and designers*. Dover Publications New York, 1966.

[4] Z. Guo, L. Zhang, D. Zhang, and S. Zhang. Rotation invariant texture classification using adaptive lbp with directional statistical features. In *Proc. ICIP*, 2010.

[5] R.M. Haralick, K. Shanmugam, and I.H. Dinstein. Textural features for image classification. *IEEE TSMC*, 3(6):610–621, 1973.

[6] J. Illingworth and J. Kittler. A survey of the Hough transform. *CVGIP*, 44(1):87–116, 1988.

[7] J. Kontinen, J. Röning, and R. MacKie. Texture features in the classification of melanocytic lesions. In *Proc. ICIAP*, pages 453–460, 1997.

[8] A. Lucieer, A. Stein, and P. Fisher. Texture-based segmentation of high-resolution remotely sensed imagery for identification of fuzzy objects. In *Proc. GeoComputation*, 2003.

[9] T. Mäenpää and M. Pietikäinen. Texture analysis with local binary patterns. *Handbook of Pattern Recognition and Computer Vision*, pages 197–216, 2005.

[10] T. Mäenpää, M. Pietikäinen, and T. Ojala. Texture classification by multi-predicate local binary pattern operators. In *ICPR*, volume 15, pages 939–942, 2000.

[11] T. Mäenpää, M. Turtinen, and M. Pietikäinen. Real-time surface inspection by texture. *Real-Time Imaging*, 9(5):289–296, 2003.

[12] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, volume 2, pages 416–423, 2001.

[13] M.S. Nixon and A.S. Aguado. *Feature extraction and image processing*. Academic Press, 2008.

[14] T. Ojala and M. Pietikäinen. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, 32(3):477–486, 1999.

[15] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[16] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 24(7):971–987, 2002.

[17] M. Petrou and P.G. Sevilla. *Image processing: dealing with texture*. Wiley, 2006.

# Model Constraints for Non-Rigid Structure from Motion

Lili Tao
lltao@uclan.ac.uk

Bogdan J. Matuszewski
bmatuszewski1@uclan.ac.uk

Stephen J. Mein
sjmein@uclan.ac.uk

Applied Digital Signal and Image
Processing Research Centre
University of Central Lancashire
Preston, UK

**Abstract**

In this paper, we propose a new method for the recovery of non-rigid structures from two dimensional (2D) image sequences captured by an orthographic camera.

We first describe a general idea of the Structure from Motion (SfM) and the basic reconstruction algorithms for both rigid and non-rigid objects using Singular Value Decomposition (SVD) and iterative optimization based methods. Current methodologies apply a non-linear optimization method to minimize image re-projection error for non-rigid object reconstruction and recovery of camera parameters. Although such methods are proven and widely adopted, their success strongly depends on the quality of the initial estimate. This initialisation oversensitivity can be reduced by introduction of shape constraints through integration of the prior information in the cost function. This inspired us to propose a new approach to estimate a shape-varying object using prior learned three dimensional (3D) deformation shape model. Results of the proposed method are shown on synthetic and real data of articulated face.

## 1 Introduction

Simultaneous shape reconstruction of 3D deformable objects observed in a video sequence, and estimation of the corresponding camera motion trajectories, using an uncalibrated camera system, is one of the fundamental problems in the computer vision – known as Structure from Motion (SfM). This technique has a wide range of applications including robot navigation, augmented reality and biomedical engineering. One common approach is based on feature point reconstruction [2, 4, 5] followed by curve based method which tackles occlusions well [1, 13].

An effective technique for shape recovery of an object is the classical algorithm of point based SfM with factorization, as proposed by Tomasi and Kanade [14]. In this, a factorization algorithm based on the Singular Value Decomposition (SVD) was used for reconstruction of a rigid object under an orthographic projection model. The algorithm factorizes the measurement matrix into shape and rotation matrices under a rank constraint. The subsequent work has focused on the factorization approach applied to multiple rigid objects [3, 7] and articulated rigid objects [15]. The reconstruction of rigid objects has been well-established for some time now, however in real environments many objects of interest are subject to deformation over time, therefore the research has expanded into the Non-Rigid Structure from Motion (NRSfM) [2, 17, 18].

Most factorization based SfM techniques begin with the assumption of an affine camera model. Thus approximating the real projection as either weak perspective or paraperspective, this is only valid in the case when the size of the object is relatively small compared to the distance between camera centre and the object. One possible solution for reconstruction under perspective projection model is to scale the image measurement matrix by using the reconstructed projective depths, then subsequently factorise the scaled matrix and impose the metric constraints in order to obtain camera parameters and object structure. Perspective reconstruction has been successfully applied where the object model can be assumed rigid: Sturm and Triggs [12] described a non-iterative factorization method for uncalibrated cameras, extending Tomasi and Kanade's algorithm [14]. Han and Kanade [6] proposed an alternative method using bilinear projective factorization algorithm; this iteratively improves the depth information, eliminating need for fundamental matrices calculation. On the other hand, NRSfM with perspective camera [9, 10, 19] can be seen as an extension of the classical reconstruction under orthographic projection.

The main contribution of this paper is a novel approach for reconstruction of 3D deformable structures, such as articulated face, from 2D video sequences taken by an orthographic camera. We proposed to add specific constraints within the state-of-art batch-processing scheme previously proposed by Del Bue [4].

The advantage of this approach is that the proposed constraints reduce a likelihood of a non-linear optimization procedure converging to a local minimum. Furthermore, the final results are not strongly dependent on the initial estimate used in the optimization process, ensuring the system does not require complex initialisation.

# 2 Related Work

Structure and motion recovery from image sequences is one of the fundamental problems in computer vision. To extend rigid SfM to the case of recovering 3D deformable objects, Bregler *et al.* [2] first described a low rank shape model to represent varying shapes. They factorize the 2D data matrix, using SVD, into object configuration weights, a camera motion matrix and 3D basis shapes used to represent the reconstructed object structure. Following this idea, factorization for articulated NRSfM has been proposed [11], but the accuracy of these methods strongly depends on the initial affine decomposition. Small inaccuracies in the affine values greatly affect the subsequent estimation process. To eliminate the ambiguity, Xiao *et al.* [18] proposed a closed-form solution to focus on deformable structure from a sequence of images taken with an uncalibrated camera. They employ the traditional orthonormality constraints, but also introduce basis constraints to further determine shape basis, however this method does not cope well with the noisy data. To overcome this, the iterative optimization methods [17], based on bundle adjustment [16], were subsequently introduced.

One of the fundamental issues when solving NRSfM problems is that they have an inherent high number of degrees of freedom (dof) which together with motion degeneracy (very limited camera motion during data acquisition) may result in worthless reconstructions. Del Bue demonstrated an alternative approach of bundle adjustment which introduces object shape prior information [4]. This approach can improve performance for both rigid and non-rigid SfM, obtaining reliable 3D reconstructions when an appropriate initial guess is provided. But in practice, when only constrained by minimization of the 2D re-projection error and a single basic shape, the optimization of large number of variables without a high quality initial guess, often results in convergence to a local minimum.

The remainder of this paper is organized as follows: Section 3 gives a short review of existing non-rigid factorization methodologies, including a mathematical representation of deformable object using a linear combination of a set of basic shapes and an adjustable set of configuration weights representing the shapes in each frame. This section also provides description of the reconstruction problem. Section 4 introduces the proposed shape model including estimate of the weight probability density function. Then the non-linear optimization is described in Section 5, and Section 6 presents results on synthetic and motion capture based data respectively.

# 3 Problem Statement

Given a point in world coordinate system, denoted as $\mathbf{s}_n = [x_n, y_n, z_n]^T$ and transformed into $m^{th}$ image coordinate system through rotation $\mathbf{R}_m$ and translation $\mathbf{t}_m$, its orthographic projection $\mathbf{x}_{mn}$ onto $m^{th}$ image, is given by:

$$\mathbf{x}_{mn} = \begin{bmatrix} u_{mn} \\ v_{mn} \end{bmatrix} = [\mathbf{R}_m | \mathbf{t}_m] \cdot \begin{bmatrix} \mathbf{s}_n \\ 1 \end{bmatrix} = \begin{bmatrix} r_{m1} & r_{m2} & r_{m3} & t_{xm} \\ r_{m4} & r_{m5} & r_{m6} & t_{ym} \end{bmatrix} \cdot \begin{bmatrix} x_n & y_n & z_n & 1 \end{bmatrix}^T \tag{1}$$

$\mathbf{x}_{mn}$ represents the $n^{th}$ 3D point $\mathbf{s}_n$ projected onto $m^{th}$ image; the orthographic camera matrix $\mathbf{R}_m$ only encodes the first two rows of rotation matrix with rotation constraint $\mathbf{R}_m \mathbf{R}_m^T = \mathbf{I}$. It can be shown that when $\mathbf{x}_{mn}$ are given with respect to the origin at the centre of gravity calculated for all projected points in the $m^{th}$ frame, $\mathbf{t}_m = \mathbf{0}$.

Consider a set of monocular 2D video sequences, tracking $P$ feature points in $F$ video frames, the $2F \times P$ observation matrix can be expressed as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{x}_{11} & \cdots & \mathbf{x}_{1P} \\ \vdots & \mathbf{x}_{mn} & \vdots \\ \mathbf{x}_{F1} & \cdots & \mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_F \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 & \cdots & \mathbf{s}_P \end{bmatrix} = \mathbf{MS} \tag{2}$$

Where $\mathbf{M}$ is a stack of motion (rotation) matrices representing camera orientation for each frame and $\mathbf{S}$ represents all 3D feature points on reconstructed objects concatenated into a single matrix.

## 3.1 Rigid structure reconstruction

To reconstruct a rigid object or a static scene, factorization is a long-standing and well-known algorithm [14]. Kanatani and Sugaya provided comprehensive descriptions and complete derivation of this technique [8]. Given its simplicity this is widely exploited in many applications and also frequently used as first step in optimization procedure to reconstruct time-varying shape structure.

The key idea is to factorize observation matrix $\mathbf{W}$ into two factors $\mathbf{M}$ and $\mathbf{S}$. Decomposition of $\mathbf{W}$ with a straightforward SVD based factorization leads to the affine approximation $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ of the real camera motion and shape represented by $\mathbf{M}$ and $\mathbf{S}$ respectively.

$$SVD(\mathbf{W}) = \mathbf{U}_{2F\times3} \mathbf{\Sigma}_{3\times3} \mathbf{V}_{3\times P} = \hat{\mathbf{M}}_{2F\times3} \hat{\mathbf{S}}_{3\times P} = (\hat{\mathbf{M}}\mathbf{H})(\mathbf{H}^{-1}\hat{\mathbf{S}}) = \mathbf{MS} \tag{3}$$

In order to recover correct rotation and shape a unique linear transformation $\mathbf{H}$ can be found enforcing the orthonormality of the rotation in $\hat{\mathbf{M}}$.

## 3.2 Non-rigid structure reconstruction

Compared with the rigid model, a non-rigid object contains more degrees of freedom and is much more complicated to recover since the shape of the object varies over time. Assuming that all the points detected in the image are represented with respect to their centre of gravity, the dynamic structure under orthographic projection can be represented as:

$$\begin{bmatrix} \mathbf{x}_{11} & \cdots & \mathbf{x}_{1P} \\ \vdots & \mathbf{x}_{mn} & \vdots \\ \mathbf{x}_{F1} & \cdots & \mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R}_F \end{bmatrix} \begin{bmatrix} -\mathbf{S}_1- \\ \vdots \\ -\mathbf{S}_F- \end{bmatrix} = \mathbf{RS} \tag{4}$$

Describing the deformation using $F$ independent shapes $\mathbf{S}_m = [\mathbf{s}_{m1} \cdots \mathbf{s}_{mP}]$, with $\mathbf{s}_{mn}$ representing coordinates of the $n$th 3D feature point in frame $m$, may entail more unknown variables than constraints. However, it is clear that motion is not random, feature points are highly correlated in time and space. Therefore, an object is unlikely to deform completely arbitrarily over time. Using basic shapes and basic trajectories are two major approaches to determine structure which lies in a lower dimensional subspace.

Using a shape model to represent the non-rigid structure is one way of reducing dimensionality of the problem [4, 5]. A linear combination of $K$ basis shapes, $B_d$, could be used to mathematically represent a morphable 3D model represented in each frame.

$$\mathbf{S}_m = \alpha_1\mathbf{B}_1 + \alpha_2\mathbf{B}_2 + \cdots + \alpha_K\mathbf{B}_K = \sum_{d=1}^{K} \alpha_d\mathbf{B}_d \tag{5}$$

Where basic shapes $\mathbf{B}_d$ are unknown but fixed, whilst deformation coefficients $\alpha_d$ are adjustable over time. Figure 1 illustrates an example of deformable model. As show "symbolically" in the figure, second basis shape provides a greater contribution than any other basic shape.

The whole shape matrix $\mathbf{S}$ can be rearranged as:

$$\mathbf{S} = \begin{bmatrix} -\mathbf{S}_1- \\ \vdots \\ -\mathbf{S}_F- \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1K} \\ \vdots & \alpha_{mn} & \vdots \\ \alpha_{F1} & \cdots & \alpha_{FK} \end{bmatrix} \begin{bmatrix} -\mathbf{B}_1- \\ \vdots \\ -\mathbf{B}_K- \end{bmatrix} \tag{6}$$

To deal with the case of non-rigid shapes under orthographic camera model, a low rank shape model has proved a successful representation. The advantage of this approach is that it can tackle the problem without any prior information about the object or the scene, or any other multiple views and 3D input. The core of this method is to express the measurement matrix as a trilinear product of three matrices: pose, basic models and time varying coefficients. Such that:

$$\mathbf{W} = \begin{bmatrix} \mathbf{R}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R}_F \end{bmatrix} \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1K} \\ \vdots & \alpha_{mn} & \vdots \\ \alpha_{F1} & \cdots & \alpha_{FK} \end{bmatrix} \begin{bmatrix} -\mathbf{B}_1- \\ \vdots \\ -\mathbf{B}_K- \end{bmatrix} = \begin{bmatrix} \alpha_{11}\mathbf{R}_1 & \cdots & \alpha_{1K}\mathbf{R}_1 \\ \vdots & \ddots & \vdots \\ \alpha_{F1}\mathbf{R}_F & \cdots & \alpha_{FK}\mathbf{R}_F \end{bmatrix} \begin{bmatrix} -\mathbf{B}_1- \\ \vdots \\ -\mathbf{B}_K- \end{bmatrix} = \mathbf{MB} \tag{7}$$

Where motion $\mathbf{M}$ is a $2F\times3K$ matrix which contains rotations $\mathbf{R}_m$ with weighting factors $\alpha_{mn}$ and basic shapes $\mathbf{B}_d$ have size $3K\times P$, the rank of $\mathbf{W}$ must be at most $3K$ in the absence of noise.

The limitation of this approach is that the motion matrix is non-linear, when an inaccurate set of basis shapes have been chosen, it may not be possible to remove the affine ambiguity.

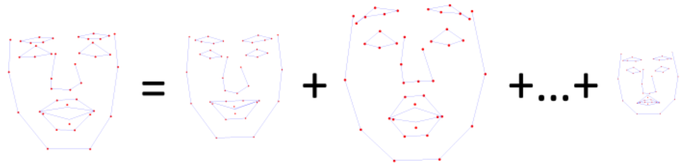


Figure 1: A graphical representation of the deformable shape model as a weighted superposition of several basic shapes (shown shapes do not represent a true appearance of the basic B$_i$). The size of the shape visually encodes the corresponding shape's weight.

# 4 Prior information learning

As mentioned in previous section, the results obtained without using any prior information about the shape and/or trajectory are sensitive to the level of noise present in the data and the algorithm initialisation. The higher number of degrees of freedom may lead to smaller re-projection error, but result in unrealistic reconstruction shapes. Appropriate prior shape information can help to improve the results. The key idea in our method is to use a learned shape space model.

## 4.1 Proposed shape model

Our method departs somewhat from the linear combination of weighted shape basis model presented in preceding section. We propose to use standard Principle Component Analysis (PCA) to obtain constraints on the basic shapes.

Principle Component Analysis is a useful statistical technique for reducing the problem dimensionality. In the last two decades, it has been employed in a wide range of applications across many areas of computer vision. In this application the idea is to represent each of the shapes in the training dataset in a low dimensional shape space that reduces the large number of observed variables into a small number of principal components. Suppose a training dataset has $N$ shapes and the set of points in $i^{\mathrm{th}}$ shape are represented by $\mathbf{X}_i$. The mean shape, $\bar{\mathbf{X}}$, of all the training dataset is given by: $\bar{\mathbf{X}} = \sum_{i=1}^{N} \mathbf{X}_i$ and eigenshapes $\mathbf{E}_i$ and eigenvalues $\lambda_i$ are obtained from the covariance matrix, defined as $\mathbf{C} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathbf{X}_{\mathrm{i}} - \bar{\mathbf{X}})(\mathbf{X}_{\mathrm{i}} - \bar{\mathbf{X}})^{\mathrm{T}}$. Any of the shapes from the training dataset can be then approximated by:

$$\mathbf{X}_{\mathrm{i}} \cong \bar{\mathbf{X}} + \boldsymbol{\gamma}_i \mathbf{E} = \bar{\mathrm{X}} + \begin{bmatrix}\gamma_{i1} & \gamma_{i2} & \dots & \gamma_{i(K-1)}\end{bmatrix}\begin{bmatrix}\mathbf{E}_1 \\ \mathbf{E}_2 \\ \vdots \\ \mathbf{E}_{K-1}\end{bmatrix} \tag{8}$$

$K$-1 is the number of dimensions after reducing the dimensionality. $\gamma_{\mathrm{i}}$ describes the contribution of $i^{\mathrm{th}}$ eigenshape and is calculated using the inner product between $\mathbf{E}_i$ and. $\mathbf{X}_i - \bar{\mathbf{X}}$.

Inspired by the idea of PCA and following the deformable shape representation described in equation (6), our proposed shape model is given by:

$$\mathbf{S}_m = \mu\mathbf{B}_0 + \begin{bmatrix}\alpha_{m1} & \cdots & \alpha_{m(K-1)}\end{bmatrix}\begin{bmatrix}\mathbf{B}_1 \\ \vdots \\ \mathbf{B}_{K-1}\end{bmatrix} \tag{9}$$

There are *K* basic shapes, $\mathbf{B}_0$ as the first basic shape is similar to mean shape $\bar{\mathbf{X}}$ computed from all the faces in the training datasets. Therefore $\mu$ is a scaling factor for first basic shape which controls the overall size of the shape. The rest of the basic shapes, $\mathbf{B}_1$ to $\mathbf{B}_{K-1}$ are forced to be close to the corresponding eigenshapes. The basic shapes are only “encourage” to be close to the mean shape and eigenshapes instead of to being exactly the same.

By stacking the shapes $\mathbf{S}_m$ for each time instant, then projecting them onto the 2D images using orthographic projection model, equation (9) can be re-written in compact matrix form:

$$\mathbf{W} = \begin{bmatrix} \mu_1\mathbf{R}_1\mathbf{B}_0 \\ \vdots \\ \mu_F\mathbf{R}_F\mathbf{B}_0 \end{bmatrix} + \begin{bmatrix} \alpha_{11}\mathbf{R}_1 & \cdots & \alpha_{1(K-1)}\mathbf{R}_1 \\ \vdots & \ddots & \vdots \\ \alpha_{F1}\mathbf{R}_F & \cdots & \alpha_{F(K-1)}\mathbf{R}_F \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_{K-1} \end{bmatrix} \tag{10}$$

## 4.2 Prior

Given that deformation is not random, thus with prior knowledge it is possible to restrict the estimated deformation of the object; if it is known how the weighting coefficients $\alpha_{md}$ are distributed in *K*–1 dimensional space. If the prior is not applied to constrain the weights, it may lead the reconstructed shapes representing not feasible deformations.

To further constraint the reconstructed shapes, a prior probability on the values of the weighting coefficients is added to the model. The Parzen window density estimation in the face-eigen space was used for this purpose.

$$p(\boldsymbol{\alpha}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h^2}\phi\left(\frac{\boldsymbol{\gamma}_i - \boldsymbol{\alpha}}{h}\right) \tag{11}$$

Where *N* is the number of shapes used to estimate the probability density function, and $\phi(\bullet)$ is a kernel function. For the isotropic Gaussian kernel function the estimate of the density function is given by:

$$p(\boldsymbol{\alpha}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{\left\|\boldsymbol{\gamma}_i - \boldsymbol{\alpha}\right\|^2}{2\sigma^2}\right) \tag{12}$$

The dimensionality of this function is defined by the number of eigenshapes used in the approximation. An example 2-D weights probability distribution is shown in Figure 2.

# 5 Non-linear Optimization

As the information about shapes and weights probability distribution is learned in advance, the optimization process comes down to minimizing a cost function built as a superposition of four components

The first component of the cost function measures the re-projection error between the feature points detected in the observed images and corresponding projection of 3D points in the estimated shapes. The re-projection error is given by:

$$\varepsilon_{re} = \sum_{m=1,n=1}^{F,P}\left\|\mathbf{w}_{mn} - \tilde{\mathbf{w}}_{mn}\right\|^2 \text{ with } \tilde{\mathbf{w}}_{mn} = \mathbf{R}_m\sum_{d=0}^{K-1}\alpha_{md}\mathbf{B}_{dn} \tag{13}$$

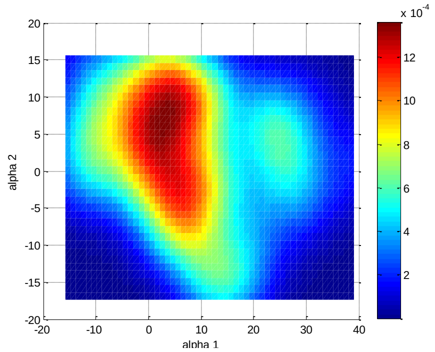


Figure 2: Probability distribution of configurations for first two basic shapes in 2-D.

Assuming that the reconstructed object is viewed by an orthographic camera, rotation matrix $\mathbf{R}_m$ represents orthographic camera matrix. The second component of the cost function enforces orthonomality of all $\mathbf{R}_m$ and is expressed as:

$$\varepsilon_{rot} = \sum_{m=1}^{F} \left\| \mathbf{R}_m \mathbf{R}_m{}^T - \mathbf{I} \right\|^2 \tag{14}$$

The prior on the shape basis given in equation 15 is included as the third component of the cost function:

$$\varepsilon_{bs} = \left\| \mathbf{B}_0 - \bar{\mathbf{X}} \right\|^2 + \sum_{d=1}^{K-1} \left\| \mathbf{B}_d - \mathbf{E}_d \right\|^2 \tag{15}$$

Given that the reconstructed object is not part of the training dataset, we are much more concerned about recovering the 3D shapes rather than having accurate basic shapes.

Last but not least, following discussion in Section 4.2, the fourth components of the cost function introduces constraints on the weighting coefficients. We restrict the search for optimal weights within the high probability region of the learned weights probability distribution leading by maximising $p(\boldsymbol{\alpha}_\mathrm{m})$.

The overall proposed cost function combines minimization of the re-projection error with efficient constraints for rotation matrices, shape basis, as well as weighting coefficients:

$$\min_{\mathbf{R}_m, \mathbf{B}_d, \boldsymbol{\alpha}_m} \left( \varepsilon_{re}\left(\mathbf{R}_m, \mathbf{B}_d, \boldsymbol{\alpha}_m\right) + \beta_1 \varepsilon_{rot}\left(\mathbf{R}_m\right) + \beta_2 \varepsilon_{bs}\left(\mathbf{B}_d\right) - \beta_3 p(\boldsymbol{\alpha}_m) \right) \tag{16}$$

Where scalars $\beta_1, \beta_2, \beta_3$ are the design parameters controlling importance of each constraint in the cost function. A non-linear optimization based on bundle adjustment using Levenberg-Marquardt algorithm was applied to minimize this cost function. The adopted method is efficient and is able to converge to the correct solution even in cases where the initial point is far from the expected minimum.

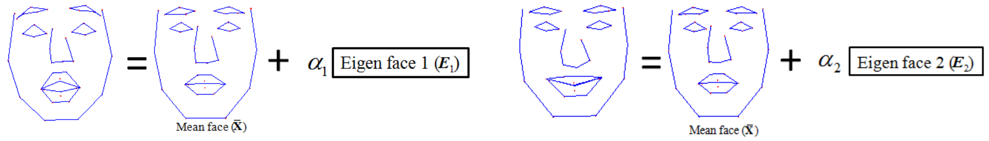
Rather than initialize the data using the method described in Section 3.2, the initialization is obtained using the generalized SVD (GSVD) [4] followed by orthonormal decomposition [5]. An initial shape is given by a rigid shape which is computed from measured data and prior shape model. To initialize rotation and weights, orthonormal decomposition is applied to decompose the remaining factors in the observation matrix.

# 6 Results and Evaluation

The experiments evaluating the proposed methodology are based on recovery of an articulated face model. In the case of face reconstruction, the estimated shape can be accurately represented using a model with a relatively small number of degrees of freedom, therefore allowing for linear deformations. Firstly we introduce the training data and show the learned shape model. Then, the accuracy of the proposed method is analysed with the use of synthetic data. Finally, the results obtained for real data, captured by a dynamic 3D scanner, are shown to further validate the proposed approach.

## 6.1 Shape model

The training datasets are taken from the BU-3DFE[1] and Hi4D-ADSIP[2] database. A total number of 2400, rigidly co-registered, 3D face images of different people with different facial expressions were used for learning the shape model and the distribution of weights. Figure 3 shows two examples of the shapes built using the learned mean face and eigenfaces.


=
Mean face ($\bar{\mathbf{X}}$)
+ $\alpha_1$
Eigen face 1 ($\boldsymbol{E}_1$)
=
Mean face ($\bar{\mathbf{X}}$)
+ $\alpha_2$
Eigen face 2 ($\boldsymbol{E}_2$)

(a) (b)

Figure 3: (a) Shape consisting of mean face and first eigenface; (b) Shape consisting of mean face and second eigenface.

## 6.2 Evaluation on synthetic data

In the first experiment, we generated a synthetic sequence to simulate a human face, allowing deformation. In real case, inaccurate tracking will affect the 2D measurement input data, which would lead to a failure of most previously proposed approaches, as those are very sensitive to noise. This experiment is designed to test the performance of our method in terms of its sensitivity to different levels of Gaussian noise added to the measurement data $\mathbf{W}$. Figure 4 shows the performance of our method with different noise levels and a varying number of basic shapes used for the reconstruction. The input data was composed of a total of 42 feature points over 60 frames. The number of basic shapes used for the reconstruction was set at 3, 5, 7, 10, 15 and 20, and levels of additive noise as 5% (Noise = 5% $\|\mathbf{W}\|$ ), 10% and 15%. The error was measured by $\frac{\|\text{reconstructed shape}-\text{true shape}\|}{\|\text{true shape}\|}\%$. As expected, increasing the number of basic shapes decreases the error efficiently due to greater number of eigenfaces used to constrain the reconstructed shapes. With a noise level 0%, the recovered shape is very similar to the true shape with the reconstruction error close to zero. With noise present in the measurements, reasonably accurate shapes are still obtainable, showing that the method is robust.

The method introduced by Del Bue [4] uses rigid model as prior shape. This works well only in the case when initial estimate is close to the global optimum and measurements are relatively accurate. In order to compare our method with Del Bue's, both algorithms were tested using the same 2D input sequence. Results are shown in Figure 6. Although the measurement errors are small for both methods for low level of the measurement noise, the proposed method performs noticeably better when the noise level increases.

[1] BU-3DFE database has been obtained from Binghamton University USA;

[2] Hi4D-ADSIP database is available from the ADSIP Research Centre at the University of Central Lancashire

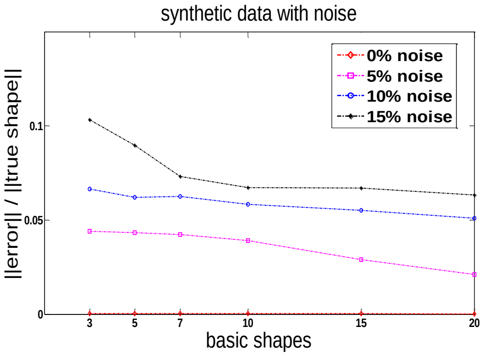


Figure 4: Reconstruction error given with respect to noise level and number of basic shapes employed.

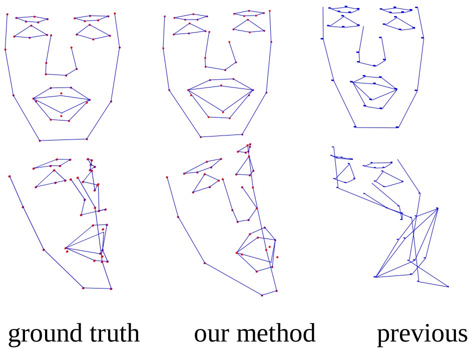


Figure 5: Our method offers improved facial reconstruction, evident from the side view.

| #Basic shapes | | 3 | 5 | 7 | 10 | 15 | 20 |
|---|---|---|---|---|---|---|---|
| 3D error (%) | 34 P | 12.56 | 12.22 | 10.63 | 10.28 | 8.62 | 7.50 |
| | 68 P | 16.45 | 13.31 | 12.10 | 10.82 | 8.60 | 7.34 |
| 2D re-projection error (%) | 34 P | 0.39 | 0.22 | 0.14 | 0.11 | 0.076 | 0.053 |
| | 68 P | 0.66 | 0.26 | 0.20 | 0.14 | 0.064 | 0.049 |

Table 1: Error with different number of basic shapes and points (40 frames)

| #Basic shapes | | 3 | 5 | 7 | 10 | 15 | 20 |
|---|---|---|---|---|---|---|---|
| 3D error (%) | 30 F | 12.56 | 12.22 | 10.63 | 10.28 | 8.62 | 7.50 |
| | 60 F | 17.16 | 13.99 | 13.98 | 12.54 | 10.34 | 8.21 |
| 2D re-projection error (%) | 30 F | 0.39 | 0.22 | 0.14 | 0.11 | 0.076 | 0.053 |
| | 60 F | 0.9 | 0.39 | 0.27 | 0.22 | 0.085 | 0.059 |

Table 2: Error with different number of basic shapes and frames (34 points)

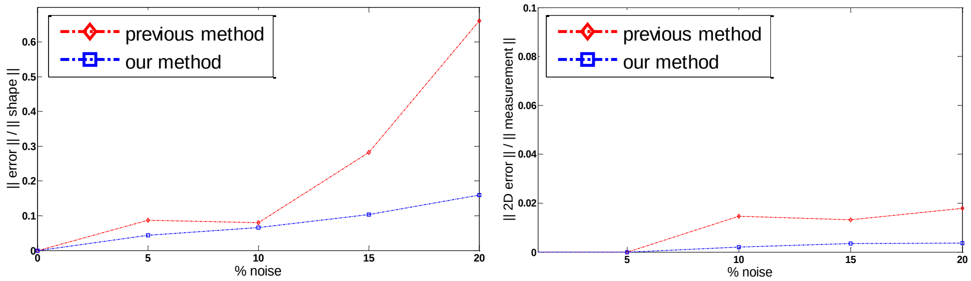


Figure 6: Reconstruction (left) and measurement (right) error, varying the noise level for comparison of the existent and our method.

## 6.3 Motion capture based data

Ground truth data of an articulated face was captured using Passive 3-D scanner with 3D tracking of 34 and 68 feature points. The points were projected onto the images sequences with different number of frames (30 and 60 frames respectively) under orthographic camera model. The results are listed in Table 1 and 2, evaluated in terms of 3D shape error and 2D re-projection error, for cases where differing number of basic shapes are used for the reconstruction.

Figure 5 compares the Del Bue's approach and ours. Although the frontal view appears well reconstructed in both cases, the side view demonstrates our method performs better for recovering depth information.

The results shown in figure 6 are for tracking 42 points over 60 frames video sequences. We present both front view and side view of a selection of facial reconstructions extracted from the sequence.

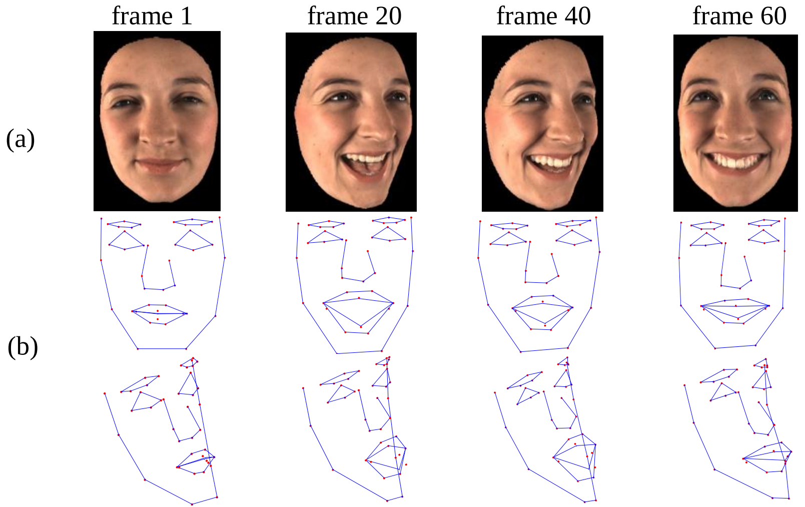


Figure 6: (a) 4 frames with different facial expression extracted from the 2D input video sequence. (b) Front and side view of reconstructed faces.

# 7 Conclusion

We have developed several extensions for recently proposed algorithm for recovering 3D deformable object and camera pose from a video sequence. The proposed extensions include use of learned shape model and distribution of the weights, in the cost function which improves performance of the optimization process.

The current approach relies on a linear subspace model to represent the deformations of the object of interest. However, this approach is only applicable to a relatively simple non-rigid object, especially when the reconstructed object is based on only a small number of basic shapes. Further work is therefore required to constrain shapes to a smooth manifold representing learned complex shape variability. This approach will be more accurate and well-adapted to large deformation models which cannot be accurately represented by a linear subspace.

# References

[1] A. Bartoli, E. von Tunzelmann, and A. Zisserman. Augmenting images of non-rigid scenes using point and curve correspondences. In *Proc. IEEE Conference on Computer Vision And Pattern Recognition*, Jun 2004.

[2] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[3] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3): 159-179, 1998.

[4] A. Del Bue. A factorization approach to structure from motion with shape priors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 2008.

[5] A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using ranklet-based tracking and non-linear optimization. In *Image and Vision Computing*, volume 25, pp 297–310, March 2007.

[6] M. Han, T. Kanade, Reconstruction of a scene with multiple linearly moving objects, In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina, pp. 542-549, June 2000.

[7] M. Han and T. Kanade. Multiple motion scene reconstruction from uncalibrated views. In *Proc. International Conference on Computer Vision, volume 1*, pp 163-170, July 2001.

[8] K. Kanatani, Y. Sugaya, Factorization without factorization: complete recipe, Memories of the Faculty of Engineering, Okayama University 38 (1-2), pp. 61 – 72, 2004.

[9] X. Lladó, A.D. Bue, and L. Agapito. Non-Rigid Metric Reconstruction from Perspective Cameras. *Image Vision Computing*, 28:1339–1353, 2010.

[10] X. Lladó, A.D. Bue, A. Oliver et al. Reconstruction of non-rigid 3D shapes from stereo-motion. *Pattern Recognition Letters* vol. 32, Article 7, 2011.

[11] P. Marco *et al*. Factorization for non-rigid and articulated structure using metric projections. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, Florida, 2009.

[12] P. F. Sturm , B. Triggs, A Factorization Based Algorithm for Multi-Image Projective Structure and Motion, In *Proc. European Conference on Computer Vision* Volume II, p.709-720, 1996

[13] M. Prasad, A. W. Fitzgibbon, A. Zisserman, L. Van Gool. Finding Nemo: Deformable Object Class Modelling using Curve Matching. In *Proc .IEEE Conference on Computer Vision and Pattern Recognition*, 2010

[14] C. Tomasi, T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision 9(2):137-154*, 1992.

[15] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, California, volume 2, pp 1110-1115, June 2005.

[16] W. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment: A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: theory and Practice*, LNCS, pp 298-375. Springer Cerlagm 2000.

[17] G. Wang, H.T. Tsui and Q.M.J. Wu, Rotation constrained power factorization for structure from motion of nonrigid objects, *Pattern Recognition Letters* 29 (1), pp. 72–80, 2008.

[18] J. Xiao, J. Chai, T. Kanade, A closed-form solution to non-rigid shape and motion recovery, In: *Proc. European Conference on Computer Vision*, Prague, Czech Republic, pp. 573–587, 2004.

[19] J. Xiao and T. Kanade. Uncalibrated perspective reconstruction of deformable structures. In *Proc. International Conference on Computer Vision*, Beijing, China, October 2005.

# Index of Authors