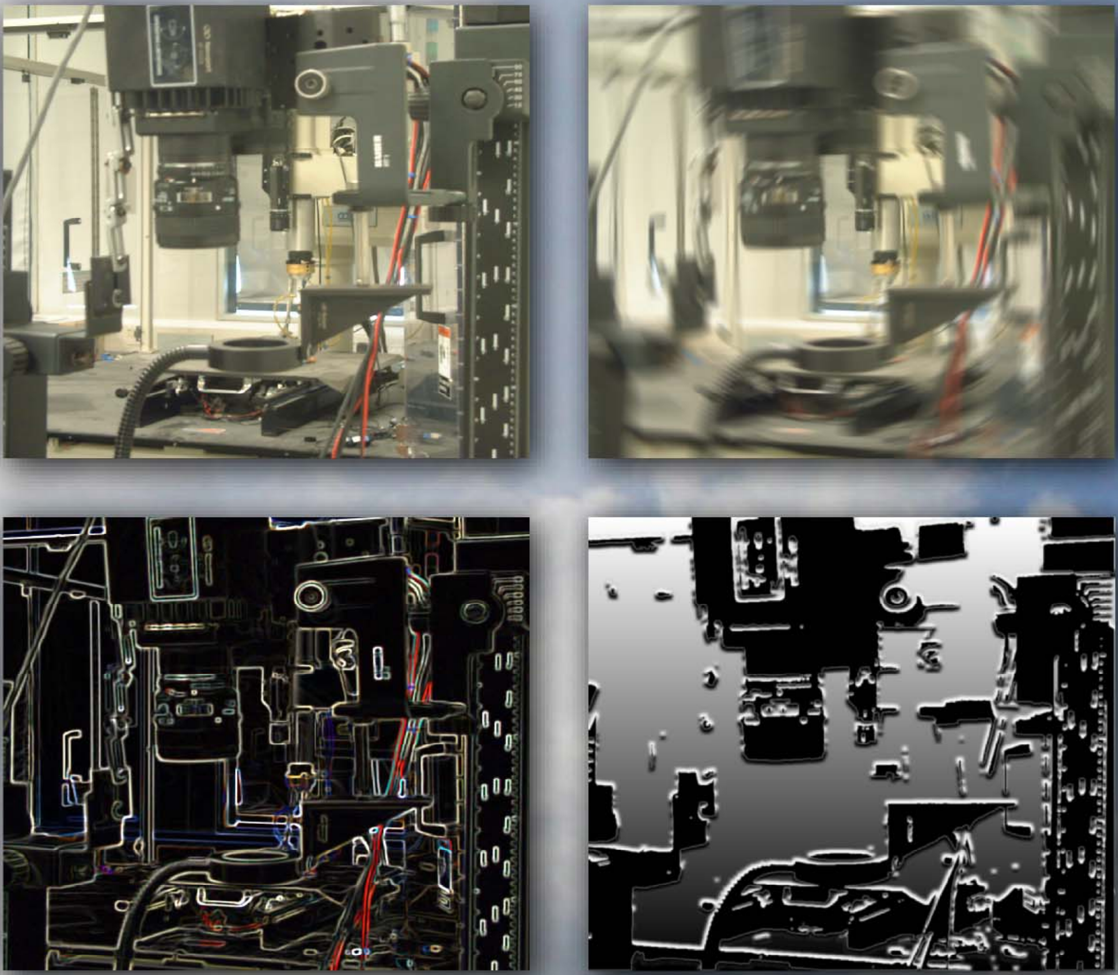
IMVIP 2006

Proceedings of the Irish Machine Vision and Image Processing Conference, 30th August to 1st September 2006



Derek Molloy, Ovidiu Ghita & Robert Sadleir (Eds.)

DCU
Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

IPRCS
Irish Pattern Recognition and Classification Society

IMVIP 2006

Proceedings of the Irish Machine Vision and Image Processing Conference 2006

Dublin City University
30th August to 1st September 2006.

Edited by: Derek Molloy, Ovidiu Ghita, Robert Sadleir

IPRCS – Irish Pattern Recognition and Classification Society

Derek Molloy, Ovidiu Ghita, Robert Sadleir (Eds.)

IMVIP 2006

Irish Machine Vision and Image Processing Conference 2006

30th August to 1st September 2006.

Published by:
Vision System Group,
Research Institute for Network and
Communications Engineering (RINCE),
Faculty of Engineering and Computing,
Dublin City University, Dublin 9, Ireland.

ISBN-10: 0-9553885-0-3
ISBN-13: 978-0-9553885-0-7

British Library Cataloguing in Publication Data. A catalogue for this book is available from the British Library.

Printed by:
Gemini International Ltd.,
Plato Business Park,
Damastown,
Dublin 15,
Ireland.

Foreword

On behalf of the Local Organising Committee, we would like to welcome all speakers and delegates to the 2006 Irish Machine Vision and Image Processing Conference. This year IMVIP 2006 is being hosted by the Research Institute for networks and Communications Engineering, Faculty of Engineering and Computing, Dublin City University.

IMVIP 2006 is a single-track conference consisting of high quality previously unpublished contributed papers. The conference emphasises both theoretical research results and practical engineering experience in all areas. In total the programme committee reviewed 47 contributions, from which 21 were chosen for oral presentation and 14 were chosen to be represented by posters. Full papers were subjected to a double-blind review process by the programme committee.

IMVIP 2006 is the 10th conference in the series. Previous IMVIP conferences organised by Magee College, University of Ulster (1997), NUI, Maynooth (1998), Dublin City University (1999), Queen's University of Belfast (2000), NUI, Maynooth (2001), NUI, Galway (2002), University of Ulster, Coleraine (2003), Trinity College Dublin (2004) and the last conference IMVIP 2005, was hosted by Queen's University, Belfast.

We would like to thank the members of the Programme Committee for providing their expertise in the review process. This conference could not have taken place without your expert input. We are extremely grateful to Professor Paul Whelan for his constant guidance and to our colleagues in the Vision Systems Group for all their assistance. We would also like to thank the staff members of the School of Electronic Engineering and RINCE for their support, in particular Ger Lardner for her help and advice.

We are grateful to our invited speakers for taking the time to present at the conference: Professor Daniel Rueckert (Imperial College London) and Professor John Barron (University of Western Ontario).

IMVIP 2006 is run in association with the Irish Pattern Recognition and Classification Society (IPRCS[1]), a member organisation of the International Association for Pattern Recognition (IAPR).

Vision Systems Group,
Dublin City University,
Dublin 9.
Ireland.

Derek Molloy
Robert Sadleir
Ovidiu Ghita
Eamonn Boyle

August 2006.

[1] See: www.**iprcs**.info

Acknowledgements

We would like to express our gratitude to our sponsors, RINCE (Research Institute for Networks and Communications Engineering), The Office of the Vice-President for Research, DCU and the Vision Systems Group, DCU.

Programme Chair:

- Derek Molloy, Dublin City University, Ireland

Technical Programme Chairs:

- Eamonn Boyle, Dublin City University, Ireland
- Ovidiu Ghita, Dublin City University, Ireland
- Robert Sadleir, Dublin City University, Ireland

Programme Committee:

- Adam C. Winstanley, National University of Ireland, Maynooth
- Ahmed Bouridane, The Queen's University of Belfast
- Alan Smeaton, Dublin City University
- Alistair Sutherland, Dublin City University
- Andrew Donnellan, Tallaght Institute of Technology
- Andy Shearer, National University of Ireland Galway
- Anil Kokaram, Trinity College Dublin
- Bruce G Batchelor, University of Wales
- Bryan W. Scotney, University of Ulster Coleraine
- Danny Crookes, The Queen's University of Belfast
- David Hogg, University of Leeds, UK
- David Vernon, University of Genoa, Italy
- Don Braggins, Machine Vision Systems Consultancy, UK
- Fionn Murtagh, Royal Holloway University, UK
- Hiroshi Sako, Hitachi Central Research Laboratory, Japan
- Hugh McCabe, Institute of Technology Blanchardstown
- James Mahon, Agilent Technologies
- Jane Courtney, Dublin Institute of Technology
- Joe Connell, Cork Institute of Technology
- John Barron, The University of Western Ontario, Canada
- John Mc Donald, National University of Ireland Maynooth
- John Winder, University of Ulster Newtownabbey
- Jonathan G Campbell, Letterkenny Institute of Technology
- Kenneth Dawson-Howe, Trinity College Dublin
- Naomi Harte, Trinity College Dublin
- Noel Murphy, Dublin City University
- Paul F. Whelan, Dublin City University
- Philip Morrow, University of Ulster Coleraine
- Pierre Soille, SAI, EC Joint Research Centre, Italy
- Richard Reilly, University College Dublin
- Robert Fisher, University of Edinburgh, UK
- Rozenn Dahyot, Trinity College Dublin

Contents

Invited Papers

Medical and Biomedical Imaging

Object and Event Recognition

2D, 3D Scene Analysis and Visualisation

Active Vision, Tracking and Motion Analysis

Data Clustering and Texture Analysis

Segmentation

Applications, Architectures and Systems Integration

Posters

Invited Papers

Medical Image Registration in Healthcare, Biomedical Research and Drug Discovery

Professor Daniel Rueckert
Visual Information Processing
Department of Computing, Imperial College
180 Queen's Gate
London SW7 2BZ, UK

D.Rueckert@imperial.ac.uk

Abstract: Imaging technologies are developing at a rapid pace allowing for in-vivo 3D and 4D imaging of the anatomy and physiology in humans and animals. This is opening up unprecedented opportunities for research and clinical applications ranging from imaging for drug discovery and delivery, over imaging for diagnosis and therapy, to imaging for basic research such as brain mapping. In this talk we will focus on how computational techniques based on non-rigid image registration can be used to address the image analysis challenges in healthcare, biomedical research and drug discovery.

Severe Storm Detection and Tracking in 3D Doppler Radar Imagery

J. L. Barron, R. E. Mercer
Dept. of Computer Science
University of Western Ontario
London, Ontario, Canada, N6A 5B7
{barron, mercer}@csd.uwo.ca

P. Joe
King City Radar Station
Meteorological Service of Canada
Toronto, Ontario, Canada, M3H 5T4
Paul.Joe@ec.gc.ca

Abstract

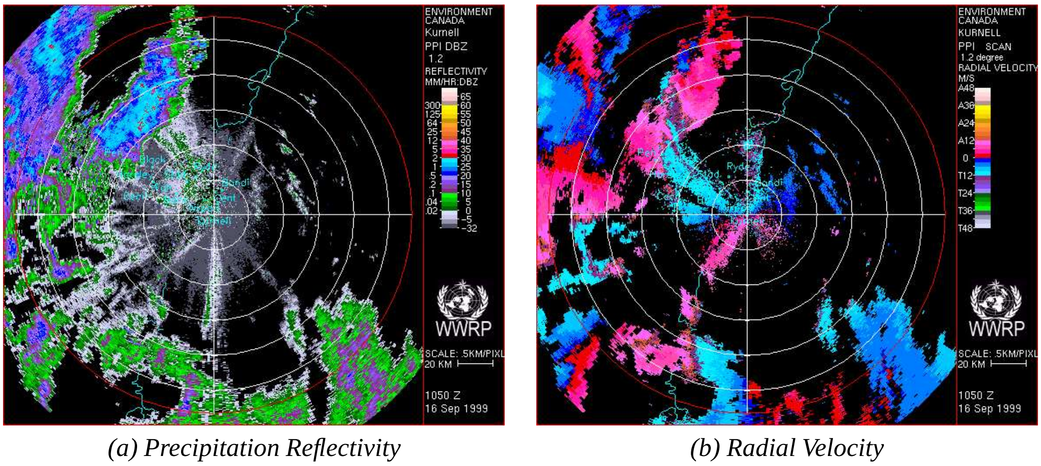
We describe our project for detecting and tracking severe weather in Doppler radar datasets. We present a history of our work, starting in 1994 with a 2D storm tracking algorithm. This tracking algorithm is posed in a relaxation labelling framework using compatibility functions that are based on the notions of fuzzy storms and fuzzy algebra using 2D reflectivity data. We extended this work in 2001 using two types of 3D Doppler radar data: (1) Doppler reflectivity data to detect and track storms as deformable 3D objects and (2) Doppler radial velocity data to compute 3D optical flow to predict 3D storm motion. Our current work is tornado signature detection in Doppler radar reflectivity data.

Keywords: Severe Weather Storm Detection and Tracking, Doppler Reflectivity (Precipitation Density) and Radial Velocity Datasets, Relaxation Labelling, 3D Fuzzy Algebra, 3D Optical Flow via Least Squares and Regularization, Tornado Signature Detection.

1 Introduction

Severe weather storms are localised events, usually affecting areas smaller than tropical cyclones and floods, so their devastating impact is often underestimated. These storms, which are more common than any other natural hazard, can occur everywhere, causing deaths and property damage. For this reason, the forecasting of severe storms is both critical and necessary.

We use both 3D Doppler reflectivity (precipitation density) [18] and radial velocity [2] data to detect and track storms in a relaxation labelling framework using a "fuzzy" algebra [16]. Typical reflectivity/radial velocity images produced by Doppler radar are shown in Figure 1. We first present fuzzy algebra and fuzzy storms in Section 2, then present the calculation of 3D optical flow from radial velocities in Section 3, then present the relaxation labelling algorithm that integrates this data via compatibility factors in Section 4 and finally present some experimental results in Section 5.



(a) Precipitation Reflectivity

(b) Radial Velocity

Figure 1: The Doppler radar imagery obtained from the King City Doppler radar at time 199909161050 [1999, Sept. 16, 10:50 hours] for elevation level 3. (a) Precipitation Reflectivity and (b) Radial Velocity.

Our original storm detection and tracking work (circa 1994) involved using 2D images of Doppler precipitation images. We detected 2D "fuzzy" storms and tracked them using a relaxation labelling algorithm. Because of page restrictions, in this paper we describe only our latest 3D tracking algorithm. The relaxation labelling essentially remains unchanged but 2D fuzzy storm, initially represented as fuzzy circles, evolved first into 3D fuzzy spheres and currently into 3D fuzzy ellipsoids. We did not use Doppler radial velocity in our 2D work to compute 2D optical flow (although some recent work demonstrated this is feasible) but do use it in our 3D work to compute 3D optical flow. This flow is a good predictor of a storm's immediate motion and we have incorporated it into our tracking algorithm. A description of our 2D work can be found in a journal paper [5] and a book chapter [11].

2 3D Fuzzy Algebra

Severe weather storms are not rigid and therefore can't be tracked using their center of masses or contour outlines. Two possible modifications to the current rigid object tracking methodology are possible: create algorithms to deal with this non-rigidity of the tracked objects, for example, using snakes [14] or find a modified representation of the objects and use the original (essentially unchanged) rigid object tracking algorithms with this new representation. We have chosen the latter approach in designing a tracking algorithm that uses the notion of "fuzzy" storms to capture the uncertainty in a storm's location. We use fuzzy vectors to get the correspondence of a fuzzy storm in adjacent images, thereby describing the movement of a storm between two adjacent images.

2.1 3D Fuzzy Storms

Our storm detection program groups connected sets of precipitation reflectivity voxels above a threshold (30 here) into potential storms. Even though the voxels are not uniformly spaced or sized (they are locally), we use a simple 3D floodfill algorithm to do this. Each voxel has coordinates denoted as (x_1, x_2, x_3). (μ_1, μ_2, μ_3) denotes the center of mass of a storm. We treat each set of 3D Doppler storm voxels as a 3D multivariate normal distribution [6] and compute a 3×3 covariance matrix Σ (symmetric positive and semi-definite), where the $(i,j)^{th}$ element, σ_{ij}, is given by;

$$\sigma_{ij} = \sigma_{ji} = \varepsilon[(x_i - \mu_i)(x_j - \mu_j)], \;\; i,j = 1, \ldots, 3. \tag{1}$$

σ_{ij} is computed as:

$$\sqrt{\frac{\sum_{x_i, x_j \in R}(x_i - \mu_i)(x_j - \mu_j)}{|R| - 1}}, \tag{2}$$

where $|R|$ is the number of storm voxels in the ellipsoid. We compute the eigenvalues, λ_i, and the corresponding eigenvectors, $\hat{e}_i$ of Σ and use each eigenvector as one of the ellipsoid axes and $\sqrt{\lambda_i}$ as the corresponding radii. We use these ellipsoids to represent hypothesised fuzzy storms in the 3D Doppler precipitation reflectivity data.

2.2 Ellipsoidal Fuzzy Algebra

The definitions for ellipsoidal fuzzy algebra are basically the same as for spherical fuzzy algebra [16], except where changes are needed when ellipsoids instead of spheres are used.

Definition 1. *A* **3D fuzzy point** $E\langle c, r\rangle$ *is defined as an ellipsoid with center* $c = (x, y, z)$*, three radii* $r = (r_x, r_y, r_z)$ *and three mutually orthogonal direction vectors* $e = (\hat{e}_x, \hat{e}_y$ *and* $\hat{e}_z)$.

The point can be anywhere in the ellipsoid including the center. Two fuzzy points $E_1\langle c_1, r_1, e_1\rangle$ and $E_2\langle c_2, r_2, e_2\rangle$ are identical if and only if $c_1 = c_2$, $r_{x_1} = r_{x_2}$, $r_{y_1} = r_{y_2}$, $r_{z_1} = r_{z_2}$, $e_{x_1} = e_{x_2}$, $e_{y_1} = e_{y_2}$ and $e_{z_1} = e_{z_2}$. Figure 2 shows 2 such fuzzy points.

Definition 2. *A* **fuzzy vector** *from a fuzzy point* E_1 *to another fuzzy point* E_2 *is defined as the infinite set of all displacement vectors from points in* E_1 *to points in* E_2*, see [16].*

Definition 3. *The* **fuzzy length** *or* **fuzzy magnitude** *of a fuzzy vector* $\overrightarrow{E}$ *is a set of lengths or magnitudes of all vectors in* $\overrightarrow{E}$ *and is defined as* $\|\overrightarrow{E}\|$.

Consider two fuzzy points E_1 and $E_2 \in \mathbf{E}$ (the set of all fuzzy vectors). The displacement vector from a point in E_1 to any point in E_2 can be defined as $\overrightarrow{E_1E_2}$ since a fuzzy point is only a set of the Euclidean points in three dimensions. The set of all such vectors is the fuzzy vector from E_1 to E_2, i.e.

$$\overrightarrow{E_1E_2} = \{\; \overrightarrow{e_1e_2} \mid e_1 \in E_1 \; and \; e_2 \in E_2 \;\}. \tag{3}$$

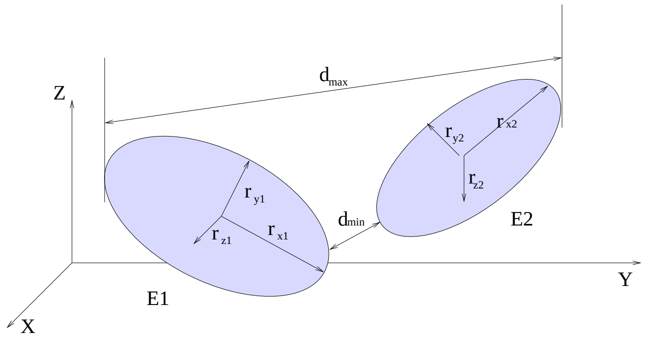


Figure 2: The closest distance d_{min} and furthest distance d_{max} between any two points respectively on the surfaces of the fuzzy points P_1 and P_2.

with fuzzy magnitude:

$$\|\overrightarrow{E_1E_2}\| = \{\ \|\overrightarrow{e_1e_2}\| \mid \overrightarrow{e_1e_2} \in \overrightarrow{E_1E_2}\ \}. \tag{4}$$

As apparent in Figure 2, we can express d_{min} and d_{max} given by the variables c_1, r_1, e_1 and c_2, r_2, e_2 as:

$$d_{min} = \min\{\ \|\overrightarrow{e_1e_2}\| \mid \overrightarrow{e_1e_2} \in \overrightarrow{E_1E_2}\ \} \text{ and} \tag{5}$$

$$d_{max} = \max\{\ \|\overrightarrow{e_1e_2}\| \mid \overrightarrow{e_1e_2} \in \overrightarrow{E_1E_2}\ \}. \tag{6}$$

A fuzzy magnitude is then the interval $[d_{min}, d_{max}]$.

Definition 4. *The* **fuzzy angle** *subtended by a non-zero fuzzy vector $\overrightarrow{Q}$ relative to another non-zero fuzzy vector $\overrightarrow{P}$ is defined as the set of angles subtended by any displacement vector $\overrightarrow{q}$ in $\overrightarrow{Q}$ relative to another displacement vector $\overrightarrow{p}$ in $\overrightarrow{P}$ having a touching head and tail, respectively. The set can be denoted as $\langle \overrightarrow{P}, \overrightarrow{Q} \rangle_\theta$.*

Consider the three fuzzy points: $E_1 = \langle c_1, r_1, e_1 \rangle, E_2 = \langle c_2, r_2, e_2 \rangle$ and $E_3 = \langle c_3, r_3, e_3 \rangle$. We can pick any point e_1 in E_1, e_2 in E_2 and e_3 in E_3 to from a pair of displacement vectors $\overrightarrow{e_1e_2}$ and $\overrightarrow{e_2e_3}$. The angle between these two vectors can be calculated by the dot product of the two vectors:

$$\cos\theta = \frac{\overrightarrow{e_1e_2} \cdot \overrightarrow{e_2e_3}}{\|\overrightarrow{e_1e_2}\|\ \|\overrightarrow{e_2e_3}\|}. \tag{7}$$

We can define the minimum and maximum angles as:
$\theta_{min} = \min\langle \overrightarrow{E_1E_2}, \overrightarrow{E_2E_3} \rangle_\theta$ and $\theta_{max} = \max\langle \overrightarrow{E_1E_2}, \overrightarrow{E_2E_3} \rangle_\theta$. The fuzzy angle is then the interval $[\theta_{min}, \theta_{max}]$.

We currently compute the fuzzy distances and angles by brute force; we just enumerate all distances and angles between the voxels in the ellipsoids. The intersection of two arbitrarily oriented ellipsoids is an open problem (for example, in 3D video game playing software[1]) and a closed form solution for this calculation is an active area of current research.

3 3D Optical Flow

The 3D motion constraint equation can be derived in a similar fashion, as for the 2D motion constraint equation given in equation (given in [9, 15]):

$$I_X U + I_Y V + I_Z W + I_t = 0, \tag{8}$$

where 3D velocity $\vec{V}$ has components U, V and W and I_X, I_Y, I_Z and I_t are the X, Y, Z and t intensity derivatives. This work has mostly been motivated by and applied to medical applications, for example, to compute 3D optical flow for CT (Computed Tomography) [7], MRI (Magnetic Resonance Imaging) [10], and PET (Positron Emission Tomography) [12] datasets.

We can rewrite equation (8) as

$$\vec{V} \cdot \hat{n} = V_n, \tag{9}$$

where $\vec{V_n} = V_n\hat{n}$ is the normal velocity which can be written in terms of intensity derivatives as:

$$\vec{V_n} = \frac{-(I_X, I_Y, I, Z)I_t}{||(I_X, I_Y, I_Z)||_2^2}. \tag{10}$$

[1] www.magic-software.com

This is similar to plane normal velocities for 3D range flow derived in [20].

To compute the 3D full velocity $\vec{V} = (U, V, W)$ from radial velocity $\vec{V_r}$ we use the dot product to obtain:

$$\vec{V} \cdot \hat{r} = V_r, \tag{11}$$

which is one equation in three unknowns. We note that the normal direction in equation (9) has been replaced by the radial direction, $\hat{r}$, here. This is the 3D motion constraint equation for 3D optical flow, except we use radial (rather than normal) velocities to set up and solve linear systems of equations in small neighbourhoods.

To solve for $\vec{V}$ at a point, we select a small local neighbourhood about the point. Currently, we use a $7 \times 7 \times 7$ neighbourhood (determined by trial and error). Before the least squares estimation, the radial velocity data is smoothed using a $7 \times 7 \times 7$ averaging filter. The experimental results presented in [3, 4] show that smoothing the data before the computation and using $7 \times 7 \times 7$ neighbourhoods for least squares integration produces the best result.

Since the neighbourhood is small compared to the whole Doppler dataset, we can assume that all points in the neighbourhood move with the same full velocity $\vec{V}$. Since the radial velocities $\vec{V_r}$ for different points satisfy different motion constraint planes, their intersection defines the common 3D full velocity $\vec{V} = (U, V, W)$, where U, V, W are three components of the velocity vector respectively along the X, Y, and Z axes. This forms the basis of our least squares computation. Full details are given in [3]. The problem with the calculation is that the radial velocity values are not as accurate when they were converted to unsigned 8-bit numbers in the acquisition stage. The algorithm presented here starts with these inaccurate least squares velocities and regularizes them to obtain more accurate and smoother velocity fields.

3.1 Least Squares Regularized Flow from Radial Velocities

To constrain our regularization to give a smooth full velocity field close to the true full velocity we use the computed least squares flow as a third consistency constraint in the regularization:

$$\begin{aligned} \int\!\!\int\!\!\int \underbrace{(\vec{V} \cdot \hat{r} - V_r)^2}_{\textit{Motion Constraint Equation}} \quad + \\ \alpha^2 \underbrace{(U_X^2 + U_Y^2 + U_Z^2 + V_X^2 + V_Y^2 + V_Z^2 + W_X^2 + W_Y^2 + W_Z^2)}_{\textit{Smoothness Constraint}} + \\ \underbrace{\beta^2((U - U_{ls})^2 + (V - V_{ls})^2 + (W - W_{ls})^2)}_{\textit{Least Squares Velocity Consistency Constraint}} \partial X \partial Y \partial Z, \end{aligned} \tag{12}$$

where $\vec{V_{ls}} = (U_{ls}, V_{ls}, W_{ls})$ is computed least squares 3D velocity. The first two constraints enforce 3D Horn and Schunck like constraints with respect to the 3D motion constraint equation and global smoothness in the 3D velocity field. The idea here is to compute a smooth regularized velocity compatible with the local least squares velocities. α and β are Lagrange multipliers which specify the relative importance of the various constraints. Based on trial and error, we use $\alpha = 5.0$ and $\beta = 1.0$ to obtain the results in this paper. Since $\nabla^2 U = U_{XX} + U_{YY} + U_{ZZ}$, $\nabla^2 V = V_{XX} + V_{YY} + V_{ZZ}$ and $\nabla^2 W = W_{XX} + W_{YY} + W_{ZZ}$ and expanding $\vec{V} \cdot \hat{r}$ as $Ur_1 + Vr_2 + Wr_3$, we can rewrite the Euler-Lagrange equations that minimize this functional as:

$$(Ur_1 + Vr_2 + Wr_3)r_1 + \beta^2 U = \alpha^2 \nabla^2 U + \beta^2 U_{ls} + V_r r_1, \tag{13}$$

$$(Ur_1 + Vr_2 + Wr_3)r_2 + \beta^2 V = \alpha^2 \nabla^2 V + \beta^2 V_{ls} + V_r r_2, \tag{14}$$

$$(Ur_1 + Vr_2 + Wr_3)r_3 + \beta^2 W = \alpha^2 \nabla^2 W + \beta^2 W_{ls} + V_r r_3. \tag{15}$$

The approximations $\nabla^2 U \approx \bar{U} - U$, $\nabla^2 V \approx \bar{V} - V$ and $\nabla^2 W \approx \bar{W} - W$ let us rewrite the Euler-Lagrange equations in matrix form as:

$$A \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} (\alpha^2 \bar{U} + \beta^2 U_{ls} + V_r r_1) \\ (\alpha^2 \bar{V} + \beta^2 V_{ls} + V_r r_2) \\ (\alpha^2 \bar{W} + \beta^2 W_{ls} + V_r r_3) \end{bmatrix}, \tag{16}$$

where

$$A = \begin{bmatrix} (r_1^2 + \alpha^2 + \beta^2) & (r_1 r_2) & (r_1 r_3) \\ (r_1 r_2) & (r_2^2 + \alpha^2 + \beta^2) & (r_2 r_3) \\ (r_1 r_3) & (r_2 r_3) & (r_3^2 + \alpha^2 + \beta^2) \end{bmatrix}. \tag{17}$$

The Gauss Seidel iterative equations can be written as:

$$\begin{bmatrix} U^{n+1} \\ V^{n+1} \\ W^{n+1} \end{bmatrix} = A^{-1} \begin{bmatrix} (\alpha^2 \bar{U}^n + \beta^2 U_{ls} + V_r r_1) \\ (\alpha^2 \bar{V}^n + \beta^2 V_{ls} + V_r r_2) \\ (\alpha^2 \bar{W}^n + \beta^2 W_{ls} + V_r r_3) \end{bmatrix}. \tag{18}$$

We perform this iteration until $||\vec{V}^{n+1} - \vec{V}^n||_2 < 0.001$. Typically, we require about 120 iterations to achieve convergence.

4 3D Tracking via Relaxation Labelling

Fuzzy storms are then tracked over time using an incremental relaxation labelling algorithm with compatibility factors to incorporate both fuzzy storm tracking and their optical flow. We used a modified version of Bernard and Thompson's relaxation labelling algorithm [1], to use a temporal smoothness constraint as well as a spatial smoothness constraint. Krezeski [13] added *property coherence* to the algorithm. Property coherence [19] allows multiple features of a storm to be tracked over time in addition to the location of the storm center. A number of storm properties are considered: *size, minimum length, maximum length, angle, orientation* and *velocity*. The *velocity displacement* property is the only use made of 3D optical flow in our work.

Let S_k be an hypothesized fuzzy storm center in the kth image. A disparity represented by a fuzzy vector $\overrightarrow{S_j S_{j+1}}$ is constructed from S_j to S_{j+1} if the *infimum* of the fuzzy length of the fuzzy vector is less than a threshold, T_d, which is set to a default value of 10 pixels; and concurrently, the two fuzzy storm centers have compatible sizes.

4.1 Size Compatibility

We define a property function, f_s, to measure the *size-compatibility* of two fuzzy storm centers $S_1 = \langle c_1, r_1 \rangle$ and $S_2 = \langle c_2, r_2 \rangle$ as:

$$f_s(S_1, S_2) = \begin{cases} 1 - \frac{|r_1 - r_2|}{\max(r_1, r_2)} & \text{if } r_1 > 0 \text{ or } r_2 > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{19}$$

A size-compatibility threshold, T_{sc}, is set to 0.5. Note that if T_{sc} is set to 1, then a disparity will be constructed between two fuzzy storm centers only when they have exactly the same size. On the other hand, if T_{sc} is set to 0, then the size-compatibility criterion is effectively removed.

4.2 Length Compatibility

The *length compatibility* function C_d:

$$C_d\left(\overrightarrow{SC_j S_C j + 1}, \overrightarrow{SC_{j+1} SC_{j+2}}\right) = \begin{cases} 1 - \frac{|d_1 - d_2|}{\max(d_1, d_2)} & \text{if } d_1, d_2 > 0, \\ 1 & \text{otherwise,} \end{cases} \tag{20}$$

where $d_1 = \min ||\overrightarrow{SC_j SC_{j+1}}||$ and $d_2 = \min ||\overrightarrow{SC_{j+1} SC_{j+2}}||$.

4.3 Angle Compatibility

The *angle compatibility* function C_θ:

$$C_\theta\left(\overrightarrow{SC_j SC_{j+1}}, \overrightarrow{SC_{j+1} SC_{j+2}}\right) = \begin{cases} 1 - \frac{1 + \cos\left(\max\left\langle \overrightarrow{SC_j SC_{j+1}}, \overrightarrow{SC_{j+1} SC_{j+2}} \right\rangle_\theta\right)}{2} & \text{if } d_1, d_2 > 0, \\ 1 & \text{otherwise,} \end{cases} \tag{21}$$

where $\max\left\langle \overrightarrow{SC_j SC_{j+1}}, \overrightarrow{SC_{j+1} SC_{j+2}} \right\rangle_\theta$.

4.4 Orientation Compatibility

Storms are spread along two axes but the third vertical axis is always close to $90°$. This results in a simple orientation compatibility calculation: compute the angle between the major axes of ellipsoids. The smaller this angle is the closer the orientations are (and the higher the compatibility is). The *orientation compatibility* function is:

$$C_o\left(\overrightarrow{SC_jSC_{j+1}}, \overrightarrow{SC_{j+1}SC_{j+2}}\right) = \frac{f_o(\overrightarrow{SC_j, SC_{j+1}}) + f_o(\overrightarrow{SC_{j+1}, SC_{j+2}})}{2}, \tag{22}$$

where $f_o(\overrightarrow{SC_j, SC_{j+1}})$ is the angle between the two fuzzy storm centers' major radii. C_o values lie in the interval $[0, 1]$.

4.5 Velocity Compatibility

Given a storm's velocity (the optical flow vector nearest its 3D center of mass), we can predict where a storm should move in the next dataset. Figure 3 shows how the predicted and actual displacements of a storm might overlap.

The *velocity compatibility* function C_v is:

$$C_v\left(\overrightarrow{SC_jSC_{j+1}}, \overrightarrow{SC_{j+1}SC_{j+2}}\right) = \frac{f_v(\overrightarrow{SC_j, SC_{j+1}}) + f_v(\overrightarrow{SC_{j+1}, SC_{j+2}})}{2}. \tag{23}$$

Suppose $SC_j = (C_{xj}, C_{yj}, C_{zj})$ is the center of the j^{th} fuzzy storm from one Doppler radar dataset and $SC_{j+1} = (C_{x(j+1)}, C_{y(j+1)}, C_{z(j+1)})$ is the center of the $(j+1)^{th}$ fuzzy storm from the second Doppler radar data collected after time δt and $\vec{V_j} = (V_x, V_y, V_z)$ is the full velocity of SC_j. We can calculate SC'_{j+1} as $SC_j + \vec{V_j}\delta t$ and compare it to SC_{j+1} as a test of the storm track goodness. A *velocity intersection* function $f_v(SC_j, SC_{j+1})$ can be defined as:

$$f_v(\overrightarrow{SC_j, SC_{j+1}}) = \frac{\textit{Intersect\_Volume}(SC'_j, SC_{j+1})}{\textit{Min\_Volume}(SC'_j, SC_{j+1})}. \tag{24}$$

4.6 Overall Compatibility

We measure the total compatibility between two adjacent disparities using a weighted sum of these components: *size compatibility* C_s, *length compatibility* C_d, *angle compatibility* C_θ, *orientation compatibility* C_o and *velocity compatibility* C_s. The overall compatibility function is defined as:

$$C = w_sC_s + w_dC_d + w_\theta C_\theta + w_oC_o + w_vC_v, \tag{25}$$

where w_s, w_d, w_θ, w_o and w_v are normalized weights such that $w_s + w_d + w_\theta + w_o + w_v = 1$. These weight values are determined empirically [21]. Values for these and other weight coefficients (below) were chosen by empirical observation.

4.7 The Relaxation Labelling Algorithm

Two adjacent disparities are connected together if their compatibility value is greater than a threshold:

$$C\left(\overrightarrow{S_jS_{j+1}}, \overrightarrow{S_{j+1}S_{j+2}}\right) > T_c, \tag{26}$$

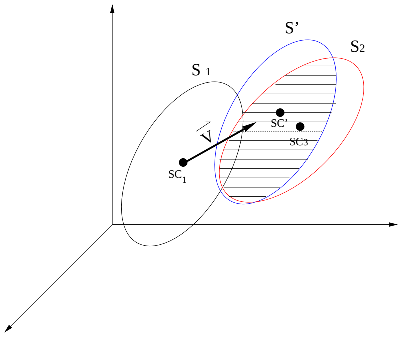


Figure 3: SC_1 is the center of storm S_1. SC_2 is the same storm in next image. $\overrightarrow{V}$ is the full velocity of SC_1. If the internal time between two adjacent images in this image sequence is δt, we calculate $SC' = SC_1 + \overrightarrow{V}\delta t$. Then we can get the intersection volume between SC' and SC_2. We can use this volume to judge if two adjacent disparities should be connected together in a track.

where T_c is currently 0.2. When all qualified adjacent disparities have been linked together, the certainty of each disparity is refined iteratively by relaxation on the overall compatibility among its adjacent disparities. Consider a disparity $d = \overrightarrow{S_j S_{j+1}}$. The initial certainty of the disparity, denoted as $p_0(d)$, is set to $f_s(S_j, S_{j+1})$. During each iteration, we apply both spatial and temporal consistency constraints to compute the supporting and contradictory evidence of the disparity using the compatibility values. Let E_s and E_c denote the supporting and contradictory evidence, respectively. Let n_s and n_c denote the number of supporting disparities and the number of contradictory disparities, respectively. These four quantities are reset at the start of each iteration.

- To apply the temporal consistency constraint, for each adjacent disparity d_t to d of the form $d_t = \overrightarrow{S_{j-1} S_j}$ or $d_t = \overrightarrow{S_{j+1} S_{j+2}}$, we compute the compatibility at the k^{th} iteration $(k > 0)$ between the two disparities as:

$$C_k(d, d_t) \quad = \quad w_1 C(d, d_t) + w_2 \left(\frac{p_{k-1}(d) + p_{k-1}(d_t)}{2} \right), \tag{27}$$

  where w_1 and w_2 are normalized weights that sum to 1. We currently use $w_1 = 0.4$ and $w_2 = 0.6$. If $C_k(d, d_t) > T_k$ (we use $T_k = 0.6$), then we add $p_{k-1}(d)$ to E_s and increment n_s by 1. Otherwise, we add $p_{k-1}(d)$ to E_c and increment n_c by 1.

- To apply the spatial consistency constraint, for each disparity d_s which has the same head storm or tail storm as d, if $p_{k-1}(d) \geq p_{k-1}(d_s)$, then we add $p_{k-1}(d)$ to E_s and increment n_s by 1. Otherwise, we add $p_{k-1}(d)$ to E_c and increment n_c by 1.

- The certainty of the disparity d at the k^{th} iteration is modified by:

$$p_k(d) = \begin{cases} \frac{1}{2}\left(1 + \frac{w_s E_s - w_c E_c}{w_s E_s + w_c E_c}\right) & \text{if } E_s \neq 0 \text{ or } E_c \neq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{28}$$

  where w_s is the weight of the supporting evidence and is computed as $w_s = \frac{n_s}{n_s + n_c}$ and w_c is the weight of the contradictory evidence and is computed as $w_c = \frac{n_c}{n_s + n_c}$.

- The iterative process stops at the k^{th} iteration when the certainty of each disparity has converged to the desired level of confidence (ε), say to n decimal places (we use $n = 6$):

$$\varepsilon = \mid p_k(d) - p_{k-1}(d) \mid < 10^{-n}, \tag{29}$$

  for each disparity d or the maximum number of iterations has been reached: $k \rightarrow T_k$, where T_k is currently set to 20.

Once the relaxation process has converged, we construct a set of all longest tracks such that each disparity has a final certainty over a threshold T_p (we use $T_p = 0.85$). We choose a subset of these tracks, with the condition that no storm lies upon more than one chosen track.

We have changed the implementation of the algorithm from processing of a complete image sequence of known length (i.e. *batch mode*) to be in *incremental mode*. Given n images the hypothesized storm disparities are relaxed. When an $(n+1)^{th}$ image is added, the relaxation is restarted using the results for the first n images plus the hypothesized storms of the $(n+1)^{th}$ image. We have observed empirically that the result is always the same as if all $n+1$ images had been initially relaxed. The difference in computation speed between the two methods is insignificant since relaxation converges within ten iterations in either mode. The incremental algorithm allows us to view the current storm tracks as the data become available.

5 Experimental Results

Each dataset consists of 15 elevations of precipitation density (reflectivity) and radial velocity of moving precipitation reflectivity data. At each elevation the data consists of 360 rays of reflectivity/radial velocity data (1 ray for each degree of a circle) and each ray consists of 600 individual reflectivity and radial velocity values. Figure 4 shows a diagram of the 3D structure of the data.

To obtain a smooth visualisation of the 3D storms, we have used a cubic β-Spline to display the storm tracks (with tension $t = 1$ and skew $s = 1$). All images of the experimental results are displayed by our X windows system, with window size of 900×900 pixels. Detected storms are coloured as red masses with a thick yellow circle representing the fuzzy storms. Each storm is numbered for later

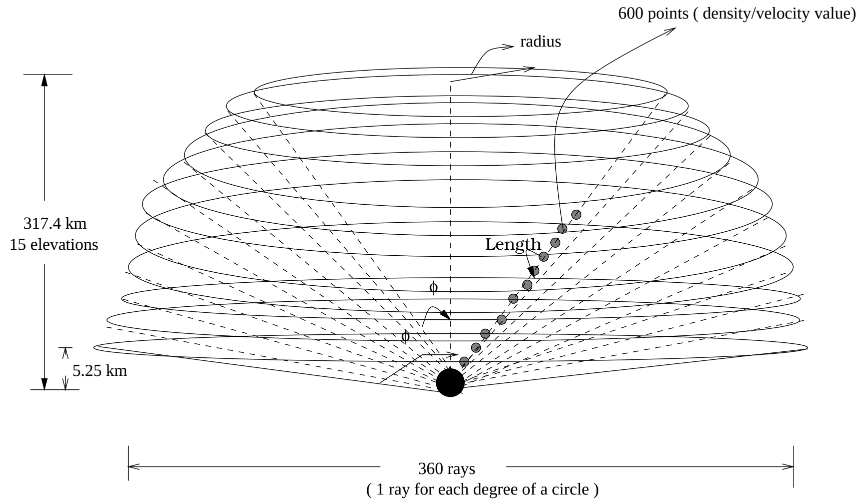

600 points (density/velocity value)
radius
317.4 km
15 elevations
Length
ϕ
ϕ
5.25 km
360 rays
(1 ray for each degree of a circle)

Figure 4: The structure of 3D Doppler radar data. The length of a cell in each dataset is 1km and there are 15 elevations ranging from a minimum angle, ϕ_{min} (cone angle), of $58°$ to a maximum angle, ϕ_{max}, of $89.5°$. The height of the rays range from a minimum of 5.25km to a maximum of 317.4km. The cone radii range from 508.84km to 599.97km.

tracking. The tracking paths are drawn as β splines with arrowheads indicating the direction of each storm motion.

We present results for detecting and tracking storms in the 3D Doppler reflectivity data that was collected at the Kurnell Radar Station in Australia at intervals of 10 minutes on September 16, 1999 (dataset 1) and September 25, 2000 (dataset 2). The name of each image file, for example, 199909161050, gives the time and date of the image (1999 September 16, 10:50am). Figure 5 shows 4 adjacent images of track 1 of a long ellipsoidal-shaped storm. Figure 6a shows all the tracks for this sequence. The oblong nature of the storm caused the fuzzy center to move too much from 199909161340 to 199909161350, breaking the track into two pieces, tracks 1 and 2. Future work includes extending fuzzy spheres to fuzzy ellipses to handle this case. Figure 7 shows the track at 20 minute intervals for the 7^{th} storm found in dataset 2 (these images were sampled every 5 minutes) while Figure 6b shows all the tracks found. In both cases, the tracking agreed with meteorologist predictions.

We use perspective projection to project 3D velocity vectors onto a 2D image plane for display purposes. A 3D point $\vec{P}(x, y, z)$ which moves with full velocity $\vec{V}$ will reach $\vec{P'}(x', y', z')$ after time interval t:

$$\begin{aligned} \vec{P'} &= \vec{P} + \vec{V}t \\ &= (X, Y, Z) + (V_X t, V_Y t, V_Z t) \\ &= (X + V_X t, Y + V_Y t, Z + V_Z t). \end{aligned} \tag{30}$$

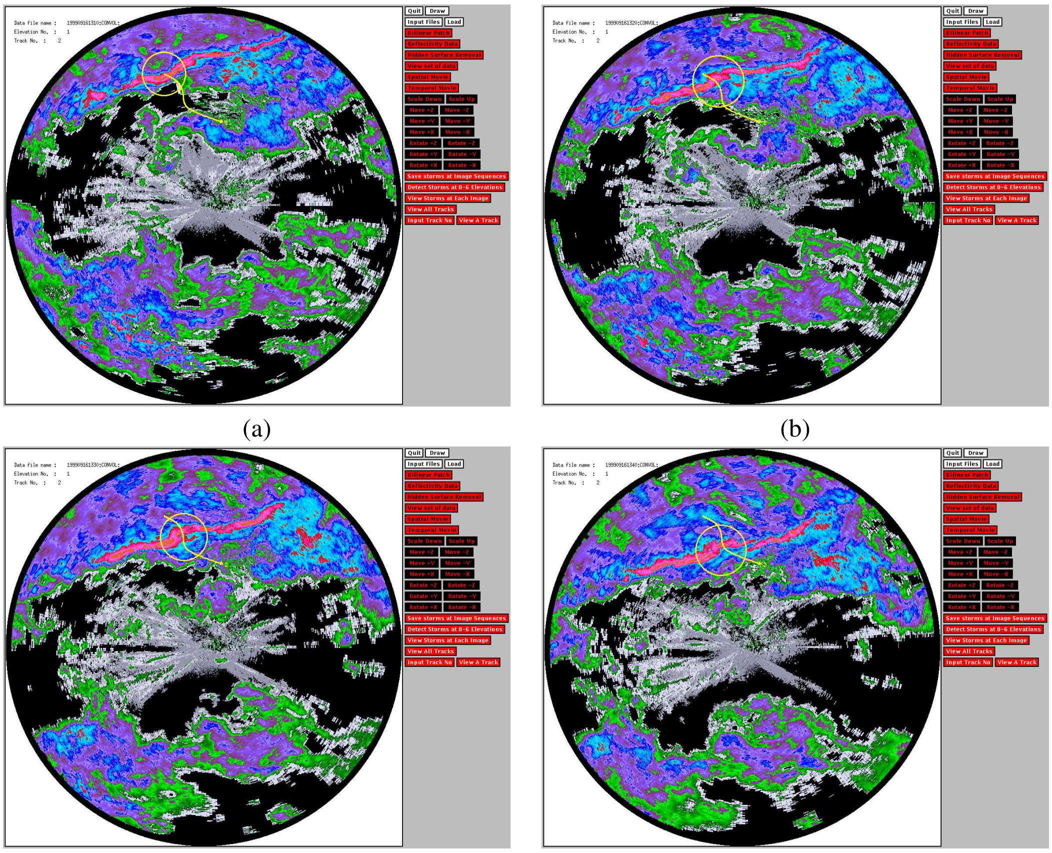
Then, the projections of $\vec{P}$ and $\vec{P'}$ onto a 2D XY image plane as $\vec{p}$ and $\vec{p}\,'$ respectively are:

$$\vec{p} = (\frac{fX}{f+Z}, \frac{fY}{f+Z}), \tag{31}$$

$$\vec{p}\,' = (\frac{f(X+V_X t)}{f+(Z+V_Z t)}, \frac{f(Y+V_Y t)}{f+(Z+V_Z t)}), \tag{32}$$

where $\vec{v} \approx \vec{p}\,' - \vec{p}$ is the 2D projection velocity of $\vec{V}$. f is the focal length of our virtual camera which we arbitrarily set to obtain “nice” looking images. Figure 8 shows the regularized least squares calculation of the velocity field for time 200009251510 with the radial velocities initially pre-smoothed with a 7 × 7 × 7 averaging filter.

Table 1 shows that the velocity intersection values, f_v, are higher for fuzzy storms represented as ellipsoids than as spheres [18, 23, 22]. Figure 9 shows the 3D velocity values computed for the ellipsoid center of storm 2. This is the same dataset showing a large oblong storm moving from northeast to southwest [18]. The velocities are shown as white (yellow in colour) vectors and can be seen to point in the direction of the storm's displacement.



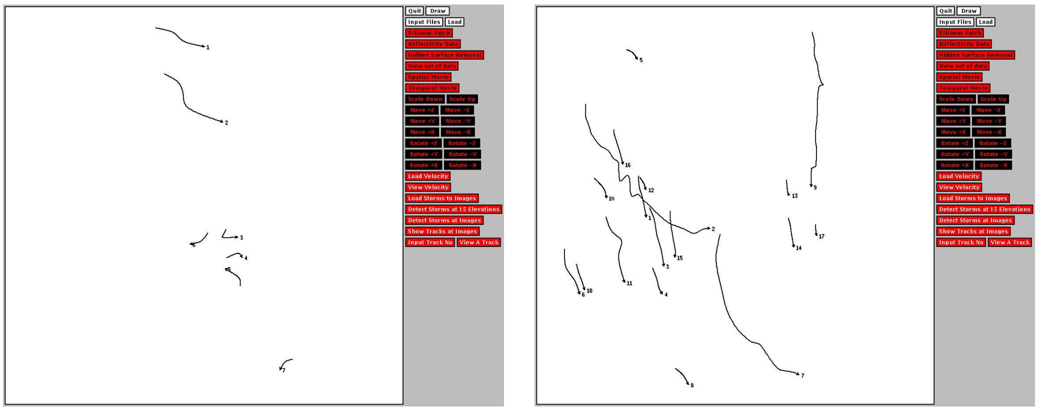
Data file name : 199909161310:CONVOL:
Elevation No. : 1
Track No. : 2

Data file name : 199909161320:CONVOL:
Elevation No. : 1
Track No. : 2

Data file name : 199909161330:CONVOL:
Elevation No. : 1
Track No. : 2

Data file name : 199909161340:CONVOL:
Elevation No. : 1
Track No. : 2

Figure 5: Four tracks of the 2^{nd} storm in images (a) 199909161310, (b) 199909161320, (c) 199909161330 and (d) 199909161340, all at elevation 1.



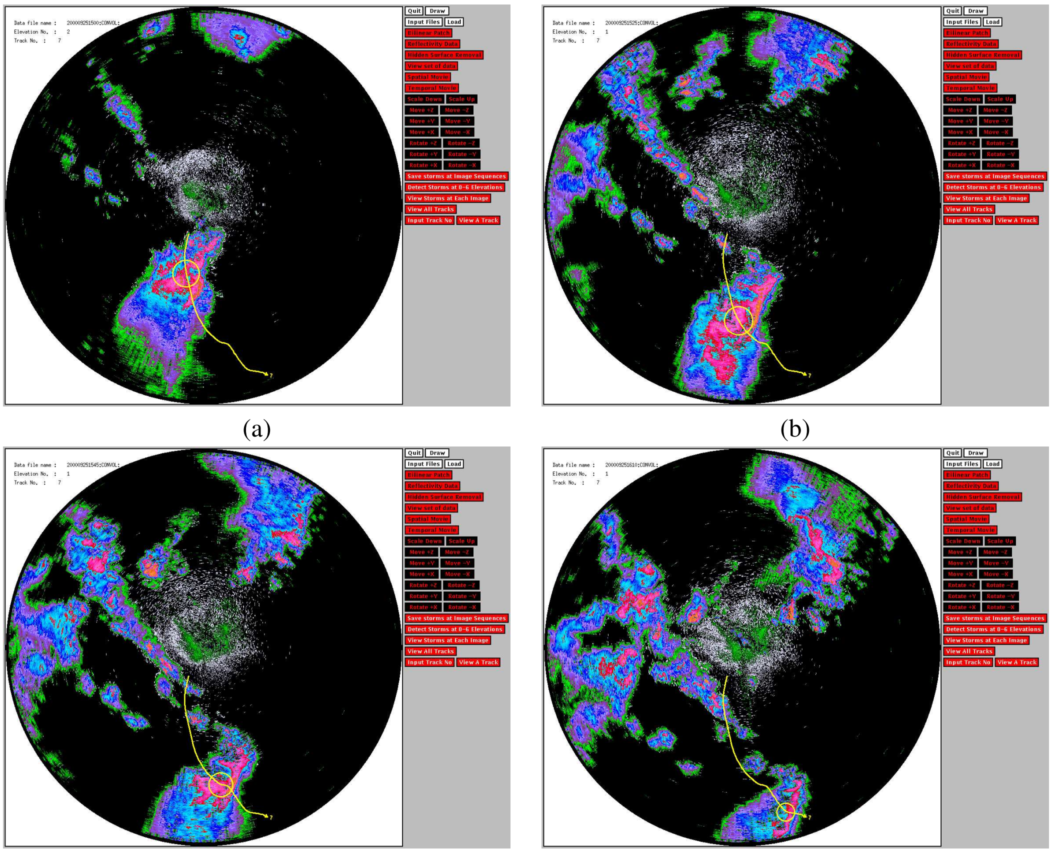
a

b

Figure 6: All storm tracks in (a) dataset 1 [September 16, 1999] and (a) dataset 2 [September 25, 2000].

Image$_1$-*Image*$_2$	*Sphere*	*Ellipsoid*
199909161310-1320	57.42	87.62
199909161320-1330	65.08	90.17
199909161330-1340	67.90	95.00

Table 1: Velocity Intersection Values, f_v, of the Predicted and Actual Fuzzy Spherical Storms and Fuzzy Ellipsoidal Storms. The first column gives image sequence numbers, *Sphere* is the fuzzy storm intersection volume percent using spheres and *Ellipsoid* is the fuzzy storm intersection volume percent using ellipsoid.



Data file name : 200009251500:CONVOL:
Elevation No. : 2
Track No. : 7

(a)

Data file name : 200009251525:CONVOL:
Elevation No. : 1
Track No. : 7

(b)

Data file name : 200009251545:CONVOL:
Elevation No. : 1
Track No. : 7

(c)

Data file name : 200009251610:CONVOL:
Elevation No. : 1
Track No. : 7

(d)

Figure 7: Four tracks of the 7^{th} storm in images (a) 200009251505, (b) 200009251525, (c) 200009251545 and (d) 200009251615. The storm at 200009251505 is at elevation 2, the others are all at elevation 1.

6 Tornado Signature Detection

We have devised a skeleton-based method to detect hook echoes in Doppler precipitation density data of tornadoes [8]. We use the HSV (Hierarchic Voronoi Skeleton) algorithm [17] to compute the "backbone" skeleton of a storm and use six features of hook echoes (curvature, orientation, thickness variation, boundary proximity, southwest localization and storm size) to detect these signatures. Figure 10a shows a Doppler image from the May 3^{rd}, 1999 Oklahoma tornado storm. Figure 10b shows the computed backbone skeleton with circles centered at each skeleton node touching at least 2 boundary points on the storm's outline. Note that this "classical" hook echo has significant counter-clockwise curvature in its backbone skeleton, has a fat-thin-fat intensity profile about the hook echo (we are the first to propose the use of this morphological property) and is located at the storm's southeast boundary. We applied our algorithm on several Doppler datasets of severe weather storms in Oklahoma from 1999 and 2003 containing numerous tornado hook echo signatures and verified its effectiveness through a CSI analysis.

7 Conclusions and Future Work

We have shown an effective tracking algorithm that uses relaxation labelling to track a number of storm properties, including size, length, angle, orientation and velocity displacement. Future work includes tracking storms among overlapping Doppler radars and integrating wind profiler and overlapping Doppler data together.

Acknowledgments

The authors gratefully acknowledge support from NSERC (Natural Sciences and Engineering Research Council of Canada) Discovery grants and an AES (Atmospheric Environment Services) subvention. The

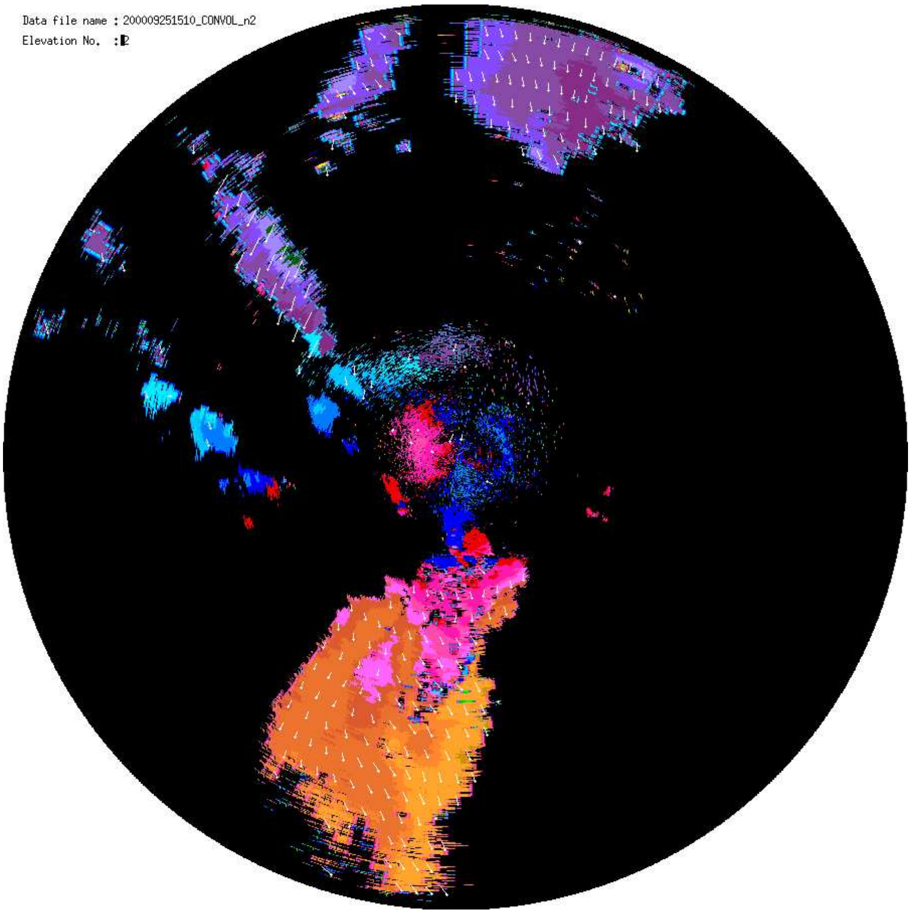


Figure 8: Computed full 3D velocity field at elevation 2 from the real Doppler radial velocity dataset at time 200009251510 using global regularization with the least squares velocity constraint.

following undergraduate and graduate students have worked on this project since 1991 (in alphabetical order): W.-K. Chan, C. Chen, X. Chen, D. Cheng, W.-K. Choi, M. Falla, M. Garden, D. Krezeski, M. Liqun, W. Qiu, D. Reynolds, G. Salonikidis, X. Tang, C. Turner, H. Wang, and H. Zhang.

References

[1] S. T. Barnard and W. B. Thompson. “Disparity analysis of images”. *IEEE Trans. Patterm Analysis and Machine Analysis*, 2(4):333–340, 1980.

[2] J.L. Barron, R.E. Mercer, X. Chen, and P. Joe. 3d velocity fields from 3d doppler radial velocity. *Intl. Journal of Imaging Systems and Technology*, 15(3):189–198, 2005.

[3] X. Chen, J.L. Barron, R.E. Mercer, and P. Joe. 3d least squares velocity from 3d doppler radial velocity. In *Vision Interface (VI2001)*, pages 56–63, June 2001.

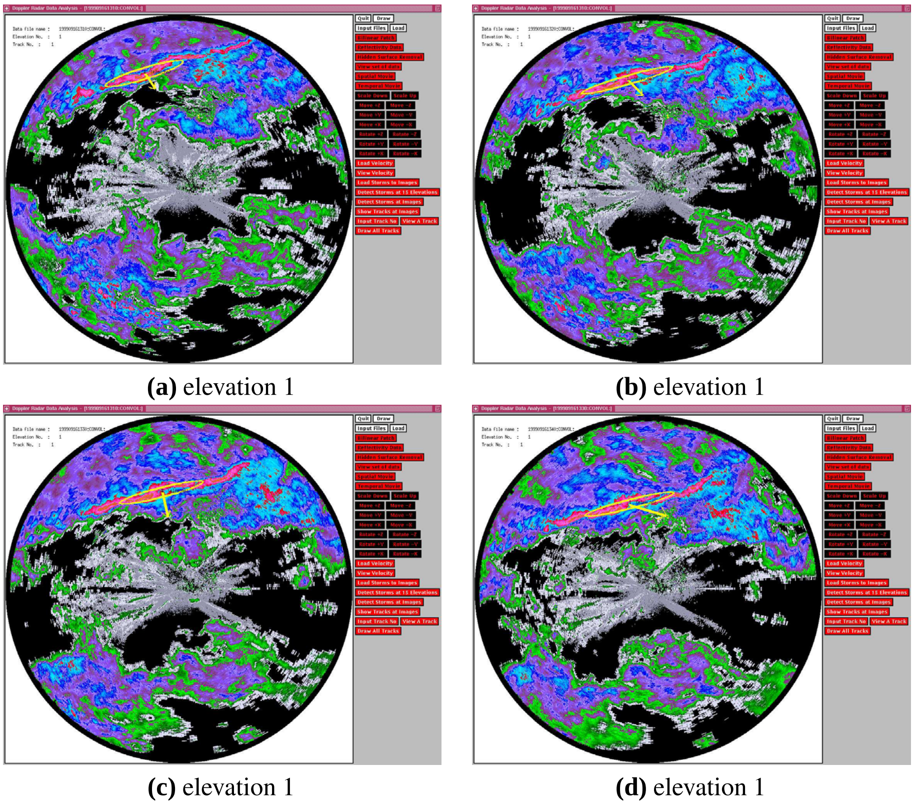
[4] X. Chen, J.L. Barron, R.E. Mercer, and P. Joe. 3d regularized velocity from 3d doppler radial velocity. In *IEEE Intl. Conf. on Image Processing (ICIP2001)*, volume 3, pages 664–667, 2001.

[5] D. Cheng, R.E. Mercer, J.L. Barron, and P.Joe. Tracking severe weather storms in doppler radar images. *Int. Journal of Imaging Systems and Technology*, 9:201–213, August 1998.

[6] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley and Sons, 2nd edition, 2002.

[7] M.A. Gutierrez, M.S. Rebelo, S.S. Furuie, L. Moura, C.M.C. Moro, C.P. Meio, and J.C. Meneghetti. A polar map representation of myocardial kinetic energy from Gated SPECT. *18th Annual International Conference of the IEEE*, pages 660–661, 1996.

[8] Wang H, R.E. Mercer, J.L. Barron, and P. Joe. Skeleton-based hook echo detection in doppler radar precipitation density imagery. *submitted*, 2006.


Doppler Radar Data Analysis - [199909161310:CONVOL:]

Data file name : 199909161310:CONVOL:
Elevation No. : 1
Track No. : 1

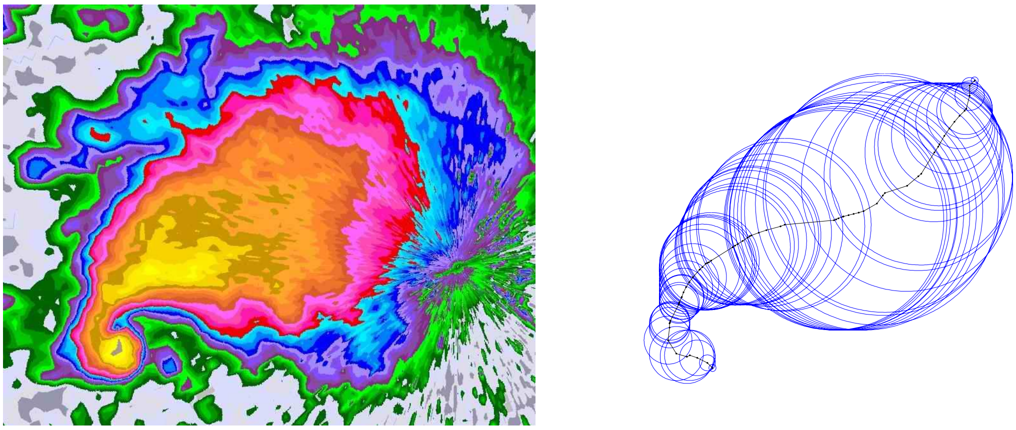
(a) elevation 1

(b) elevation 1

(c) elevation 1

(d) elevation 1

Figure 9: Velocity at the center of mass of the second storm in the reflectivity images: (a) 199909161310, (b) 199909161320, (c) 199909161330 and (d) 199909161340.


(a)

(b)

Figure 10: (a) A classic hook echo (colored red, orange and yellow). The tornado associated with this echo was part of the May 3^{rd}, 1999 Oklahoma severe weather storm. (b) The backbone skeleton for this hook echo. Each circle is the smallest circle that has a skeleton node as its center that can be completely inscribed in the storm.

[9] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–204, 1981.

[10] S.-H. Huang, S.-T. Wang, and J.-H. Chen. 3D motion analysis of MR imaging using optical flow method. *17th Annual International Conference of the IEEE*, pages 463–464, 1995.

[11] J.L.Barron, R.E. Mercer, D Cheng, and P. Joe. Tracking ‘fuzzy’ storms in doppler radar images. In Bernd Jähne, Horst Haußecker, and Peter Geißler, editors, *Computer Vision and Applications Handbook*, pages 807–820. Academic Press, Boston, January 1999.

[12] G.J. Klein and R. H. Huesman. A 3D optical flow approach to addition of deformable PET volumes. *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pages 136–143, 1997.

[13] D. Krezeski, R. E. Mercer, J. L. Barron, P. Joe, and H. Zhang. “Storm tracking in Doppler radar images”. In *Proc. International Conf. on Image Processing (ICIP94)*, volume III, pages 226–230, 1994.

[14] F. Leymarie and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:617–634, 1993.

[15] B. D. Lucas and T. Kanade. An iterative image-registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130. DARPA, 1981. (see also IJCAI81, pp674-679).

[16] R.E. Mercer, J.L. Barron, D. Cheng, and A. Bruen. Fuzzy points: Algebra and application. *Pattern Recognition (PR2002)*, 35(5):1153–1166, May 2002.

[17] R. L. Ogniewicz and O. Kuebler. Hierarchic voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995.

[18] W. Qiu, R.E. Mercer, and J.L. Barron. 3d storm tracking in 3d doppler precipitation reflectivity datasets. In *Irish Machine Vision and Image Processing conference (IMVIP2001)*, pages pp79–86, 2001.

[19] I. K. Sethi, Y. K. Chung, and J. H. Yoo. “Correspondence using property coherence”. *SPIE Applications of Artificial Intelligence X: Machine Vision and Robotics*, 1708:653–662, 1992.

[20] H. Spies, B. Jähne, and J. L. Barron. Range flow estimation. *Computer Vision Image Understanding (CVIU2002)*, 85(3):209–231, March 2002.

[21] X. Tang. Tracking 3d doppler weather storms using fuzzy ellipsoids and radial velocity. Master’s thesis, Dept. of Computer Science, The University of Western Ontario, 2003.

[22] X. Tang, J.L. Barron, R.E. Mercer, and P. Joe. Tracking 3d storms using fuzzy points represented as ellipsoids. In *Irish Machine Vision and Image Processing Conference (IMVIP2003)*, pages 73–82, 2003.

[23] X. Tang, J.L. Barron, R.E. Mercer, and P. Joe. Tracking weather storms using 3d doppler radial velocity information. In *13th Scandinavian Conference on Image Analysis (SCIA2003)*, pages 1038–1043, 2003.

Medical & Biomedical Imaging

Evaluation of Skin Lesion Asymmetry using Fourier Descriptors

K.M. Clawson[1], P.J. Morrow[1], B.W. Scotney[1], O.M. Dolan[2], D.J. McKenna[2]

[1]School of Computing and Information Engineering
University of Ulster
Coleraine BT52 1SA
clawson-k2@ulster.ac.uk, pj.morrow@ulster.ac.uk, bw.scotney@ulster.ac.uk

[2]Department of Dermatology
Royal Hospitals Trust
Grosvenor Road, Belfast BT12 6BA
olivia.dolan@royalhospitals.n-1.nhs.uk, johnmckenna@doctors.org.uk

Abstract

Malignant melanoma is the most serious form of skin cancer. Early diagnosis and prompt surgical excision of malignant melanomas is essential. The development of automated systems aimed at accurately classifying suspicious pigmented lesions as benign or malignant may improve diagnostic accuracy preoperatively. In this paper we review techniques used in the detection of lesion asymmetry, a strong diagnostic indicator of melanoma. In addition, the feasibility of utilising Fourier descriptors as a shape asymmetry measure is evaluated. It is determined that, although not offering a full measure of lesion asymmetry alone, Fourier descriptors allow accurate determination of a lesion's closest axis of symmetry, which may then be utilised in new or existing asymmetry measures.

Keywords: Medical Imaging, Melanoma, Asymmetry, Skin Lesion

1 Introduction

Malignant melanoma is a cancer which originates in melanocytes, the cells which produce skin pigmentation [1]. The incidence in Caucasian populations is rising dramatically with a doubling time of approximately 10 years. It is the most serious form of skin cancer and although representing only 5% of skin cancers diagnosed it accounts for over 90% of subsequent mortalities [2]. Prompt recognition and early diagnosis significantly improves survival rates as tumours are detected at an early stage. Patients diagnosed with thinner tumours have a much lower risk of metastatic disease and therefore improved survival [3]. When recognised early in proliferation, malignant melanomas can be excised surgically with a good prognosis (5 year survival rates approaching 95%) [4]. The early identification of malignant tumours is therefore paramount. Dermatoscopy is a technique used to improve preoperative diagnosis of pigmented skin lesions. It helps clinicians differentiate between benign pigmented lesions such as moles and malignant melanoma. A variety of systems have been devised to help in the assessment of melanocytic lesions by dermatoscopy [5]. The ABCDE rule of dermatoscopy is an algorithmic system which attempts to quantify malignancy through analysis of lesion features (figure 1). The *asymmetry*, *border* (irregularity and variegation), *colour* (homogeneity and number of colours), *diameter* (greater or less than 6mm), and *evolution* of a lesion are considered [6]. A possible criticism of such techniques is that feature evaluation is open to subjectivity [7], which can cause reduced sensitivity and specificity of diagnosis [4]. There has therefore been recent focus on the development of computerised systems which assist diagnosis. Such systems automate feature identification, allow more precise definition of features, and diminish variability of feature analysis [7, 22].

This paper aims to identify the range of skin lesion *asymmetry* quantifiers which have been developed using digital technologies and explores the viability of integrating Fourier analysis, specifically Fourier descriptors, into a shape asymmetry measure. Section 2 details existing approaches to lesion asymmetry measurement whilst section 3 discusses the Fourier approach and describes experimental work undertaken. Results and discussion are provided in sections 4 and 5 respectively.

2 Current Approaches to Asymmetry Measurement

The analysis of lesion asymmetry is an important diagnostic indicator [4]. Asymmetry may be measured across zero, one or two orthogonal axes and considers variations in surface colour, texture and/or shape [5]. Multiple digital techniques have been developed to automate this process, involving the extraction of information from binary, greyscale and colour images. Simplistic techniques evaluate asymmetry across the entire lesion, for example “circularity” may be calculated, representing the correlation between a lesion surface area and the area of that lesion’s best-fit circle [8-10]. This method is invariant to scale or translation [11] and considered highly intuitive as the shape of a regular benign lesion is often circular. However, when a lesion border is highly irregular circularity measures will not be representative of image shape, thereby allowing similar results to be obtained for dissimilar lesions [12]. An alternative approach is proposed in [8]: Lesions are segmented into 256 sectors, defined by axes crossing through the centroid. Area differences between these sectors are exploited to calculate an asymmetry measure ranging between 0 (symmetrical) and 10 (asymmetrical). Similarly, in [10], Andreassi et al propose a percentage symmetry measure based on the variance of area differences between lesion segments formed using 180 axes passing through the lesion centroid.

In [13] Colot et al. define a colour symmetry measure based on the distribution between a lesion’s centroid and “clear” and “dark” pixels within the lesion. Colour symmetry is given by:

$$s_C = \frac{\left|\overline{d_D} - \overline{d_C}\right|}{\overline{d_G}} \tag{1}$$

where $\overline{d_G}$ is the average distance to the lesion centroid, $\overline{d_C}$ is the mean distance between the clearer pixels and the lesion centroid and $\overline{d_D}$ is the mean distance between the darker pixels and the lesion centroid. A similar method is discussed in [14] which calculates a lesion’s “*dark distribution factor”*, defined as the distance between the lesion centroid and the centroid of dark pigmentation islands. Seidenari et al [15] quantify pigment asymmetry using colour averaging inside pixel blocks. Essentially a three-step process, their proposed algorithm subdivides an image into a $n \times n$ grid, ascertains the average lesion colour per pixel block and calculates the Euclidean distance (for red, green and blue planes) between each block. Subsequent measures considered for feature analysis include the mean, variance and maximum colour change across all blocks. The algorithm applied in [15] is computationally intensive, requiring $n(n-1)/2$ comparisons per calculation, where n is the number of valid (lesion) pixels in the grid.

Asymmetry has also been measured across a lesion’s principal axes of symmetry. The principal axis of symmetry (major axis or major product of inertia) is the axis which globally maximises symmetry across a lesion; the minor axis runs orthogonal to this [16]. Axes of symmetry may be calculated using standard algorithms, such as those detailed in [17]. An alternative approach is to assume that the major and minor chords of a near-symmetrical object are a close enough approximation of the true principal axes for meaningful results to be obtained [1]. In [1] the image is reflected across its major and minor axes. For each reflection, the area of the image on one side of the axis is subtracted from the reflected image on the other side resulting in two area differences. Asymmetry of shape is subsequently calculated using:

$$Asymmetry = \frac{\Delta \mathrm{A}_{\min}}{\mathrm{A}} \bullet 100\% \tag{2}$$

where $\Delta A_{\min}$ is the smallest absolute area difference and A is the lesion area. This “folding” or “reflection” operation has been replicated in [6, 18-20]. In [5] Ganster et al. calculate feature ratios between different segments of a lesion. The binary mask is split into its right and left half, lower and upper half, and into four quadrants based on the location of the major and minor axes. Shape asymmetry is compared across these quadrants using:

$$R_i = \frac{Q_i}{\sum_{j \neq i} Qj} \tag{3}$$

where Q_i represents the feature value (i.e. the perimeter, area or form-factor) of quadrant *i*.

In [6] luminance thresholding is applied to identify regions of solid pigmentation. The asymmetry of pigment distribution is computed on a per quadrant basis using:

$$\lambda_i = \frac{A_i{}^P D_i{}^P}{A_i{}^L D_i{}^L} \tag{4}$$

where A_i^P is the area of the dark pigment region within each quadrant, A_i^L is the area of the lesion in each quadrant, D_i^L is the distance from the lesion centroid in the quadrant and the centroid of the entire lesion, and D_i^P is the distance from the dark pigment centroid (per quadrant) and the centroid of the entire lesion. Finally, an overall (per lesion) measure of asymmetry is found using:

$$\alpha = \sum_{i=1}^{N} \frac{(\lambda_i - \mu_i)^2}{N} \tag{5}$$

where μ_i is the mean pigment asymmetry index calculated across all four quadrants. One criticism of the technique proposed in [6] is that two images displaying significantly different pigment features could yield similar results using equation (5). For example an image with a large island of pigmentation close to the centroid of the whole lesion could potentially have a similar λ to a lesion with a small island of pigmentation close to the border (i.e further away from the lesion centroid). Therefore it could be implied that equations (4) and (5) do not necessarily provide a representative measure of asymmetry.

Gutkowicz-Krusin et al [21] propose an alternative method for pigment asymmetry quantification. Initially an image is segmented via thresholding of luminance and the intensity $I_L(x,y)$ of a pixel is set to 0 if it is greater than the threshold value I_{th}. The lesion intensity moment is then calculated for each point $f(x,y)$ in the image, and the image is rotated by angle θ to align its principal axes parallel to the image axes. Pigment asymmetry is measured as $A = Ax + Ay$ where:

$$A_x = \frac{\sum_n \sum_y \left| I_L(x_c + n, y) - I_L(x_c - n, y) \right|}{\sum_x \sum_y I_L(x, y)} \tag{6}$$

and

$$A_y = \frac{\sum_x \sum_n \left| I_L(x, y_c + n) - I_L(x, y_c - n) \right|}{\sum_x \sum_y I_L(x, y)} \tag{7}$$

and I_L corresponds to intensity distribution. If I_L in equations (6) and (7) is replaced by values across the lesion’s binary mask, the result indicates which fraction of pixels do not have a counterpart when the image is reflected across its principal axis (i.e. shape asymmetry).

Schmid-Saugeon et al regard asymmetry calculation as an optimisation problem where a given symmetry measure should be maximised [7, 16]. They propose a generic approach based on the concept that any asymmetry can be regarded as noise and represented by the mean square error (MSE) between an image and its corresponding reflection across a particular axis. The peak signal to noise ratio (PSNR) of the image can then be used as a symmetry measure:

$$PSNR = 10\log_{10}\left(\frac{(N_q - 1)^2}{MSE}\right) \tag{8}$$

where N_q is the number of quantisation levels in the image. Finally, the PSNR determined can be used to derive a symmetry coefficient. In [7] and [16] the computational intelligence techniques of genetic algorithms and self organising maps have been applied to locate the axis which maximises the symmetry coefficient for a given image. This approach has been applied to binary, colour and textural information.

It is difficult to gauge the accuracy of techniques developed for measuring the asymmetry of pigmented skin lesions because the majority of approaches developed are integrated into systems which automate the extraction of multiple features (for example, asymmetry, border irregularity and colour). Where results for individual asymmetry measures have been provided, the maximum accuracy rates obtained (based on successfully classifying a lesion as symmetric or asymmetric) varies from 73.2% [7] to 93.5% [1]. It is important to appreciate however that in [1] the test sets used were comprised of 39 and 46 lesions respectively. In order to be certain of the validity of this method, testing on a larger data set is desirable.

3 Fourier Analysis

Fourier descriptors are a popular method for generating boundary based shape descriptors. An application of discrete Fourier analysis, these descriptors transform spatial boundary representations into the frequency domain, utilising the concept that any wave shape can be approximated using a summation of sine and cosine functions. Although two-dimensional discrete Fourier analysis has been utilised for feature extraction and image registration purposes [22], the specific use of Fourier descriptors remains unexplored within the context of melanoma asymmetry analysis. Given N object boundary points defined in two-dimensional Cartesian space and expressed as the sequence $s(n) = [x(n),y(n)]$ for n=0,1,2,...,N-1, each coordinate pair may be regarded as a complex number such that $s(n) = [x(n)+iy(n)]$ for n=0,1,2,...,N-1. Using this approach the x-axis and y-axis are treated respectively as the real (*Re*) and imaginary (*Im*) parts of a sequence of complex numbers which may in turn be used to generate a set of complex coefficients or Fourier descriptors using:

$$a(u) = \frac{1}{N}\sum_{n=0}^{N-1} s(n)e^{-i2\pi un/N} \tag{9}$$

for u=0,1,2,…,N-1. The result of equation (9) is N frequency coefficients, each of which corresponds to a complex sinusoid or basis function. Lower frequencies capture the gross essence of the object shape, whereas higher frequencies represent finer details. In addition, the magnitude or spectrum of each frequency can be calculated for analysis using:

$$|a(u)| = \left[\mathrm{Re}^2(a(u)) + \mathrm{Im}^2(a(u))\right]^{1/2} \tag{10}$$

Fourier descriptors are advantageous in that they can be made invariant to rotation, translation, scaling, and start point. It takes approximately N^2 multiplications and summations to produce a series of Fourier coefficients, but computational expense may be minimised by applying the transform on a subset of boundary points, for example points sampled at regular intervals.

3.1 Fourier Properties of Geometrically Symmetric Images

Our goal is to evaluate the usefulness of Fourier descriptors as a method for quantifying the shape asymmetry of an object. In order to identify how boundary symmetry is represented within Fourier descriptors, equation (9) was implemented on manually generated boundary point samples corresponding to images which are geometrically symmetric across either one or two orthogonal axes. The number of sample points was kept constant for all images (*N*=64), resulting in 64 frequency coefficients per image, labelled *f*(0) to *f*(63). Initially the symmetry axis of each image was aligned parallel to the y-axis. In subsequent tests each image was rotated about its centroid, thereby varying the angle of inclination of the major axis (figure 2). This was achieved by applying an affine transform on the sample points. Changes in real and imaginary parts of each complex coefficient were recorded, along with trends apparent within frequency magnitudes. In addition, the ratio between real and imaginary parts of each complex coefficient, and variance of these ratios, was computed.

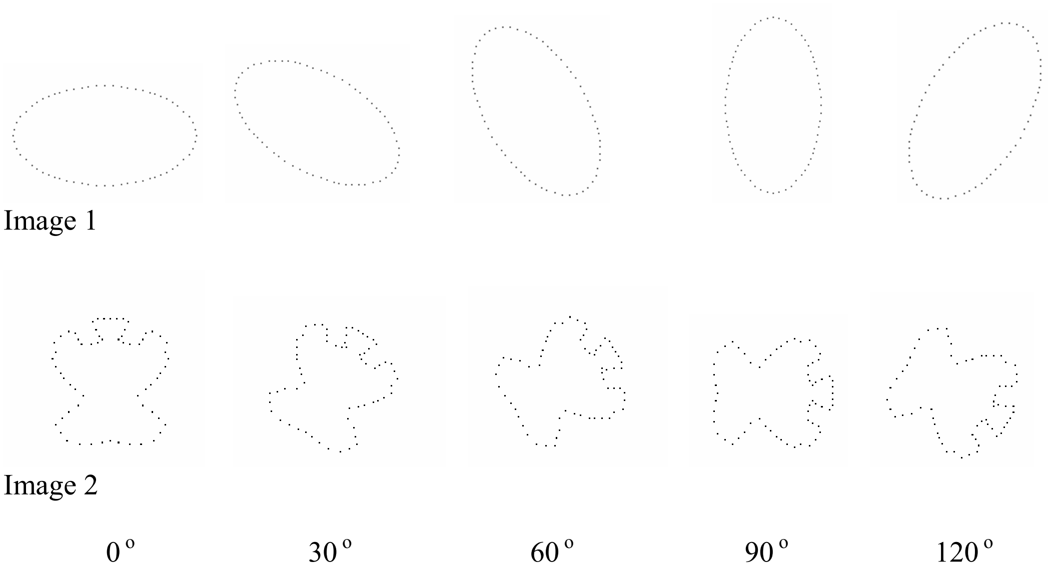


Figure 2. Examples of manually generated boundary points corresponding to images with either 1 or 2 axes of geometric symmetry, rotated at varying angles.

3.2 Fourier Properties of Non-Geometrically Symmetric Images

As an extension to 3.1, Fourier descriptors were generated for non-geometrically symmetric images. As an initial test images 1 and 2 were modified so they display reduced levels of symmetry (figure 3). Fourier descriptors were also generated and evaluated using coordinate sets corresponding to (a) 'basic' shapes with at least one clearly perceptible axis of symmetry (table 2) and (b) the boundaries of pigmented skin lesions (figure 4). The lines intersecting boundary points in figure 4 indicate the axis of symmetry which has been calculated (see section 4). Images used here were from a set supplied by the Royal Victoria Hospital, Belfast, and comprised one symmetric and two less symmetric lesions.

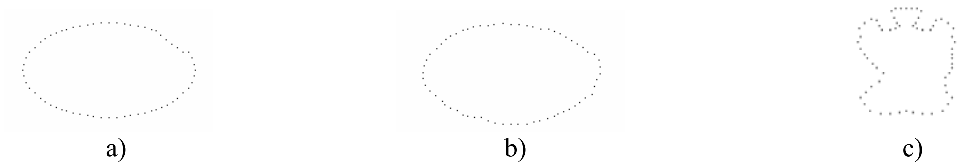


Figure 3. Images with varying degrees of symmetry: a) Fig. 2 Image 1 with reduced geometrical symmetry, b) Fig. 2 Image 1 with "rough" symmetry, c) Fig. 2 Image 2 made asymmetric

4 Results

Using Fourier descriptors one can identify the inclination of an axis of symmetry for any geometrically symmetric shape via values of real and imaginary parts of the complex coefficients. On examination of the Fourier coefficients it is apparent that if an axis of symmetry lies vertically, the real part of each complex coefficient from *f*(1) onwards is approximately equal to zero. If we have perfect symmetry around a vertical axis, then for each $k, k = 1,...,N$, we may match our data points in pairs:

$$(x(k), y(k)) = (-x(N-k), y(N-k)) \tag{11}$$

Hence, when computing the Fourier descriptor $a(u)$, the k'th and (N-k)th terms, $a_k(u)$ and $a_{N-k}(u)$, are given by

$$a_k(u) = (x(k) + iy(k))e^{-i2\pi uk/N} \tag{12}$$

and

$$a_{N-k}(u) = (x(k) + iy(k))e^{-i2\pi u(N-k)/N} \tag{13}$$

Therefore the sum $a_k(u) + a_{N-k}(u) = 2i(-x(k)\sin(2\pi uk/N) + y(k)\cos(2\pi uk/N))$ is purely imaginary. A similar argument holds when a symmetry axis is horizontal, specifically that the imaginary part of each complex coefficient from $f(1)$ onwards will be approximately zero.

As an extension of this property, using the relationship between phase angle and angle of rotation it can be easily shown that if an axis of symmetry lies at any other inclination θ then there exists a relationship between the ratio ($ratio_r$) of real and imaginary parts for specific (even numbered) coefficients and the angle of inclination of the axis θ. Specifically,

$$ratio_r = -\tan(\theta) \tag{14}$$

and

$$\theta = arcTan(-ratio_r) \tag{15}$$

When descriptors are constructed using the boundary points of less symmetric or non geometrically symmetric shapes, the $ratio_r$ of individual coefficients varies with little or no pattern across most frequencies. However computing equation (15) using average $ratio_r$ of the most significant frequencies for each image, excluding $f(0)$, remains indicative of the location of the object's axis of symmetry. Table 1 illustrates this, along with the observation that using $f(1)$ alone to estimate the angle of inclination proves an accurate method, even when applied to "asymmetrical" images, such as the image in figure 3(c) .

Actual angle of symmetry axis (degrees)	Estimated angle of symmetry axis using $f(1)$ only		Estimated angle of symmetry axis (2 most significant frequency coefficients)		Estimated angle of symmetry axis (3 most significant frequency coefficients)	
	Figure 3(b)	Figure 3(c)	Figure 3(b)	Figure 3(c)	Figure 3(b)	Figure 3(c)
0	0.6305	-0.7986	1.1499	0.8929	1.1472	-0.7854
20	20.6305	19.2014	21.1516	20.9111	21.1483	19.2623
40	40.6305	39.2014	41.1538	40.9353	41.1498	39.3239
60	60.6305	59.2014	61.1583	60.9817	61.1528	59.4381
80	80.6305	79.2014	81.1800	81.2034	81.1672	79.9312
100	100.6305	99.2014	101.1259	100.6327	101.1312	98.3706
120	120.6305	119.2014	121.1420	120.8086	121.1419	118.9804
140	140.6305	139.2014	141.1459	140.8515	141.1445	139.1028
160	160.6305	159.2014	161.1481	160.8748	161.1460	159.1665

Table 1. Using significant frequencies to estimate the location of an images axis of symmetry

In order to test the robustness of this apparent trend between the ratio of real and imaginary parts of $f(1)$ and the symmetry axis phase, equation (15) was applied to four further sets of boundary points, where each set represented a 'basic' shape with bilateral symmetry across at least one axis (Table 2). Resultant statistics indicate this approach is accurate: the maximum error incurred across all images (Table1 and Table 2) was 0.8 degrees and standard error was 0.067.

Therefore similar application using descriptors corresponding to digital representations of melanomas can provide a method of estimating the angle of inclination of a lesion's principal axis. When equation (15) is applied to images in figure 4(a) – (c) using *ratio*$_r$ of $f(1)$ the axes of symmetry were computed as being 112, 41 and 113 degrees respectively, measured clockwise from the vertical axis (Figure 4).

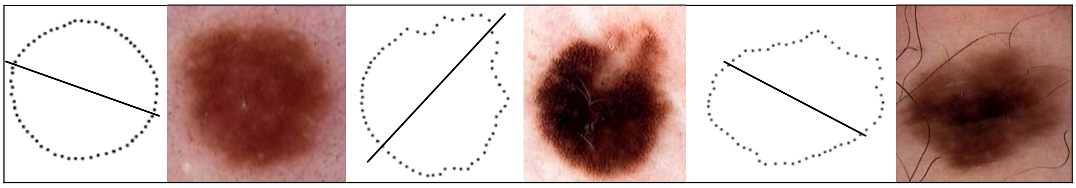
	star			**cross**			**polygon**			**Rectangle**		
θ	Re $f(1)$	Im $f(1)$	$\approx\theta$	Re $f(1)$	Im $f(1)$	$\approx\theta$	Re $f(1)$	Im $f(1)$	$\approx\theta$	Re $f(1)$	Im $f(1)$	$\approx\theta$
0	-0.27	-48.80	-0.31	0.65	-73.39	0.51	-0.13	-52.51	-0.14	-0.13	-88.39	-0.08
20	16.44	-45.94	19.69	25.71	-68.74	20.51	17.84	-49.39	19.86	30.11	-83.10	19.92
40	31.16	-37.55	39.69	47.67	-55.8	40.51	33.65	-40.31	39.86	56.71	-67.79	39.92
60	42.13	-24.63	59.69	63.88	-36.13	60.51	45.41	-26.37	59.86	76.48	-44.31	59.92
80	48.01	-8.73	79.69	72.39	-12.1	80.51	51.69	-9.25	79.86	87.02	-15.48	79.92
100	48.10	8.21	99.69	72.16	13.39	100.51	51.73	8.99	99.86	87.07	15.22	99.92
120	42.39	24.17	119.69	63.23	37.26	120.51	45.54	26.14	119.86	76.61	44.08	119.92
140	31.57	37.21	139.69	46.67	56.64	140.51	33.85	40.14	139.86	56.91	67.62	139.92
160	16.94	45.76	159.69	24.49	69.19	160.51	18.08	49.30	159.86	30.35	83.01	159.92

Table 2. Comparison of actual symmetry axis phase (θ) and estimated symmetry axis phase ($\approx\theta$) calculated using $f(1)$.

5 Discussion

Analysis of Fourier descriptors allows us to propose a rule base for differentiating between images which display geometric symmetry and those which do not. If the variance of ratio$_r$ of specific (even numbered) complex coefficients is approximately equal to zero, then the object boundary is geometrically symmetric. Otherwise the object boundary is non-geometrically symmetric. However, experimental work suggests that the use of Fourier descriptors alone does not allow determination of whether a melanoma may be defined as symmetric or asymmetric.

Nevertheless, it has been evidenced that Fourier descriptor coefficients may be used to accurately identify a melanoma's principal axis of symmetry. As the location of this axis is fundamental amongst numerous existing asymmetry measures, it would be valuable to assess the accuracy of these methods when Fourier descriptors are used to locate the axis. Possible approaches which could be assessed include those identified in [1, 16-19] and [4]. Therefore further work will consider a more formalised analysis of the trends already identified, and will compare accuracy of the Fourier method of symmetry axis location with existing methods.



a) – Symmetric b) – Slightly Asymmetric c) - Asymmetric

Figure 4. Coordinate sets corresponding to the boundaries of pigmented skin lesions, and axis of symmetry calculated using equation (16).

References

[1] Stoecker, W. Li, W. and Moss, R. (1992) "Automatic Detection of Asymmetry in Skin Tumors," *Computerized Medical Imaging and Graphics,* vol. 16, pp. 191-197.

[2] Soyer, H. Argenziano, G. et al. (2004) "Three-Point checklist of dermoscopy." *Journal of Dermatology,* vol. 2, pp. 27-31.

[3] Polsky, D. (2005), The ABCDEs of Melanoma: an evolving concept. *Journal of Drugs in Dermatology*

[4] Roberts, D. Anstrey, A. Barlow, et al (2002) “UK Guidelines for the management of cutaneous melanoma,” *British Journal of Dermatology,* vol 146 pp 7-17.

[5] Ganster, H. Pinz, P. et al (2001) "Automated melanoma recognition," *Medical Imaging, IEEE Transactions on,* vol. 20, pp. 233-239.

[6] Chang, Y. Stanley, R.J. et al (2005)"A systematic heuristic approach for feature selection for melanoma discrimination using clinical images," *Skin Research and Technology,* vol. 11, pp. 165.

[7] Schmid-Saugeon, P. Guillod, J. et al (2003) "Towards a computer-aided diagnosis system for pigmented skin lesions " *Computerized Medical Imaging and Graphics,* vol. 27, pp. 65-78.

[8] Seidenari, S. Pellacani, G. et al (1999) "Digital videomicroscopy and image analysis with automatic classification for detection of thin melanomas," *Melanoma Research,* vol. 9, pp. 163-171.

[9] Cascinelli, N. Ferrario, M. et al (1992) "Results obtained by a computerized image analysis system designed as an aid to diagnosis of cutaneous melanoma," *Melanoma Research,* vol. 2, pp. 163-170.

[10] Andreassi, L. Perotti, R. et al (1997) "Digital dermoscopy analysis for the differentiation of atypical nevi and early melanoma," *Arch Dermatol,* vol. 135, pp. 1459-1465, 1999.

[11] Ng, V. and Cheung, D. "Measuring asymmetries of skin lesions," *Systems, Man, and Cybernetics,* vol. 5, pp. 4211-4216.

[12] Lee, T. and Fung, B.(2003) "Determining the asymmetries of skin lesions with fuzzy borders," *Proceedings Third IEEE Symposium on BioInformatics and BioEngineering,* pp. 223-230.

[13] O. Colot, A. Devinoy and Sombo, D. et al (1998) "A Colour Image Processing Method for Melanoma Detection," *Medical Computing and Computer Assisted Intervention, International Conference on* pp. 562-569.

[14] Tomatis, S. Carrara, M. et al (2005) "Automated melanoma detection with a novel multispectral imaging system: results of a prospective study," *Physics in Medicine and Biology,* vol. 50, pp. 1675-1687.

[15] S. Seidenari, S. Pellacani, G. and Grana, C. (2005) "Pigment distribution in melanocytic lesion images: a digital parameter to be employed for computer-aided diagnosis," *Skin Research and Technology,* vol. 11, pp. 236-241.

[16] Schmid-Saugeon, P. (2000) "Symmetry axis computation for almost-symmetrical and asymmetrical objects: Application to pigmented skin lesions," *Journal of Medical Image Analysis,* vol. 4, pp. 269-282.

[17] Leou, J. and Tsai, W. (1987) "Automatic rotational symmetry determination for shape analysis," *Pattern Recognition,* vol. 20, pp. 571-582.

[18] Kjoelen, A. Thompson, M. et al (1995)"Performance of AI Methods In Detecting Melanoma," *Artificial Intelligence Performance,* vol. 14, pp. 411-416.

[19] Hoffmann, K. Gambichler, T. et al (2003)"Diagnostic and neural analysis of skin cancer (DANAOS). A multicentre study for collection and computer-aided analysis of data from pigmented skin lesions using digital dermoscopy," *British Journal of Dermatology ,* vol. 149, pp. 801-809.

[20] Ercal, F. Chawla, A. et al (1994) "Neural network diagnosis of malignant melanoma from color images," *Biomedical Engineering, IEEE Transactions on,* vol. 41, pp. 837-845.

[21] Gutkowicz-Krusin, D. Elbaum, M. et al (1997) "Can early malignant melanoma be defferentiated from atypical nevus by in vivo techniques? Part II. Automatic machine vision classification," *Journal of Skin Research and Technology,* vol. 3, pp. 15-22.

[22] I. Maglogiannis, S. Pavlopoulos and D. Koutsouris, "An Integrated Computer Support Acquisition, Handling, and Characterization System for Pigmented Skin Lesions in Dermatological Images," *IEEE Transactions on Information Technology in Biomedicine,* vol. 9, pp. 86-98, 2005.

A New Method for Vertebrae Contours Detection in X-ray Images

Mohammed Benjelloun, Horacio Téllez, Saïd Mahmoudi
Computer Science Department, Faculty of Engineering, rue de Houdain 9
Mons, B-7000, Belgium
Mohammed.Benjelloun, Horacio.Tellez, Said.Mahmoudi@fpms.ac.be

Abstract

This paper describes a new method for individual vertebra segmentation and identification in medical images. X-ray images of the spinal column are analysed in order to extract a closed contour for each vertebra. To achieve this goal, we proceed by steps: the starting one is a segmentation approach based on the selection of each vertebra region using a new template matching method. We use these regions information to identify each vertebra by its contour. For this task, we propose a polar signature representation of the contour, combined with a second tempalte matching process. After that, we apply an edge closing method exploiting polynomial fitting. Also, the extraction of some parameters characterizing each vertebra, related to their form, position and orientation, allows determining vertebrae motion induced by their movement between two or several positions.

Keywords: Contour detection, Region vertebra selection, Vertebrae analysis, polynomial fitting, polar signature, Motion estimation.

1 Introduction

Medical staffs often examine X-rays of spinal columns to determinie the presence of abnormalities or dysfunctions and to analyse the vertebral mobility. Medical image processing and analysis applications automate some tasks dealing with the interpretation of these images. These applications use objective image parameters to measure, compare and detect the changes between images.

X-ray images segmentation is an essential task for morphology analysis and motion estimation of the spinal column. Several methods have been proposed in the literature to analyse and to extract vertebrae contours from X-ray images [Rico et al., 2001]. Extensive research has been done by Long et al. [LR and GR, 2001, LR and GR, 2000] to automatically identify and classify spinal vertebrae. They formulated the problem of spine vertebrae identification by three level of processing: In the first stage they used an heuristic analysis combined with an adaptive thresholding system to obtain basic orientation data, providing basic landmarks in the image; in the second stage, boundary data for the spine region of interest were defined by solving an optimization problem; the third stage was expected to use deformable template processing to locate individual vertebrae boundaries at finely grained level. The main drawback of this approach is the need of a good grayscale thresholding. Stanley and Long [RJ et al., 2004] proposed a new method of subluxation detection. They used the spatial location of each vertebra in the spinal column and the variation in its position. They applied a second order spinal column approximation by using vertebral centroids. The goal of their approach is to quantify the degree to which vertebrae areas within the image are positioned on their posterior sides.

In an other work, Rodney and Thoma [Long and Thoma, 1999] described a reliable method for automatically fixing an anatomy-based coordinate system in the image with an adaptive thresholding system. Kauffmann et al. [Kauffman and Guise., 1997] first detected the axis of the spinal column by manually placing points along it and fitting a curve through them. The fitted curve was used to initialize and rigidly match templates of vertebral body with the image data to obtain vertebral outlines. Verdonck et al. [Verdonck et al., 1998] manually indicated specific landmarks in the image and finded others using an interpolation technique. The landmarks, together with a manually indicated axis of the spinal column, are used to automatically compute endplates on veretbrae and the global outline of the spine.

Techniques using Hough transform [Howe et al., 2004, Tezmol et al., 2002] and Active Shape Models [Roberts et al., 2003] are other examples of the various approaches developed. These methods use a large set of templates to capture the great variability in vertebrae shapes. But, in most of the cases, it leads to prohibitive computation time, as in the case of Hough transform, and usually needs a large and accurate training set in the case of Active Shape Models.

Other contour detection methods use a global description such as Canny [Canny, 1986] or Deriche filter [Deriche and Faugeras, 1990]. Also, standard gradient-based edge detection techniques work well only when images contain clear intensity differences at boundaries. Applied to X-ray images of the spinal column, the results of these methods are generally presented by open contours, without the possibility to restore or to isolate the precise forms of each vertebra. Moreover, it exists just a few detection methods of local closed contours, without any convincing results. Indeed, the use of global methods for vertebrae contours detection is difficult because of the nature and the non-homogeneity of gray scale levels in spinal X-ray images.

In order to improve these results, it is recommended to use a specific segmentation to these images. To overcome limitations of edge detectors in this kind of images, we propose a new method for X-ray images segmentation based on a region vertebrae selection before the effective edge detection.

On the other hand, we propose to use the segmentation results for vertebral mobility estimation. The purpose of the diagnosis is to extract some quantitative measures of particular changes between images acquired at different moments. For instance, to measure the vertebrae mobility, images in flexion, neutral and extension position are respectively analyzed. Measuring each vertebra movement allows to determine the mobility of the vertebrae, in relation to each other, and to compare the corresponding vertebrae between several images.

This paper is organized in the following way. Section 2 describes the region vertebra selection process. Section 3 presents how to combine regions information with a template matching process using a polar coordinate system, and this in order to extract vertebrae contours. Finally, in section 4, we present some experimental results and their use in vertebral mobility analysis.

2 Region vertebra selection

This first pre-processing step allows the creation of a polygonal region for each vertebra. Each region represents a specific geometrical model based on the geometry and the orientation of the vertebra. However, it is difficult to achieve fully and automatically the segmentation of X-ray spinal images with current computer vision methods. So, we propose a supervised process where the user have to click once inside each vertebra that we want to delineate and analyse. We initially place a click towards the center of each vertebra. These clicks represent the starting points $P(x_i, y_i)$ for the construction of vertebrae regions, figure (1-a), [Benjelloun et al., 2006]. After this, we compute the distance between each two contiguous points $(D_{i,i+1})$ and the line L_1, which connects the contiguous points, by a first order polynomial, equation (1).

$$L_1 = f\left[a, b; P(x_i, y_i), P(x_{i+1}, y_{i+1})\right] \tag{1}$$

The function L_1 will be used as reference for a template displacement by the function $T(x, y)$ defined in equation (2). This template function represents an inter-vertebral model, which is calculated according to the areas shapes between vertebrae. We use the L_1 function and the inter vertebral distances to calculate the inter vertebral angles (α_{iv}) and to determine a division line for each inter vertebral area. To determine the points representing borders areas, we displace the template function $T(x, y)$, equation (2), between each two reference points

$P(x_i, y_i)$ and $P(x_{i+1}, y_{i+1})$, along the line L_1. Then, we calculate the correlation degree D_C between the template function and the image $I(x, y)$.

$$T(x,y) = \left(1 - e^{(-rx^2)}\right) \quad with \quad r = k/D_{i,i+1} \tag{2}$$

with k empirical value.

The correlation degree is a measure of similarity which permits to obtain the ideal template function that joins perfectly the borders between the vertebrae areas. The maximum correlation value D_C between the template function $T(x, y)$ and the image $I(x, y)$ for all the analysed positions will correspond to the most stable position. This position corresponds to an angle α_{iv} and a position $P(x_{iM}, y_{iM})$ for the template function, i.e. the position on the image in which the template function $T(x, y)$ is best placed.

In figure (1-a), the click points $P(x_i, y_i)$ are represented. In figure (1-b), we present the inter-vertebral points $P(x_{iM}, y_{iM})$ given by the proposed procedure. In figure (1-c), boundary lines between vertebrae are traced according to the angle α_{iv} given by the same procedure and centered on the points $P(x_{iM}, y_{iM})$). To obtain vertebral regions, we connect the extreme points of the boundary lines, figure (1-d).

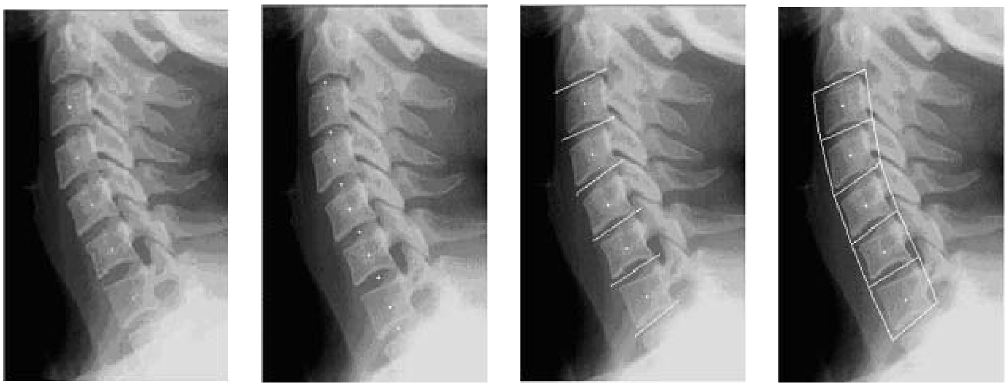


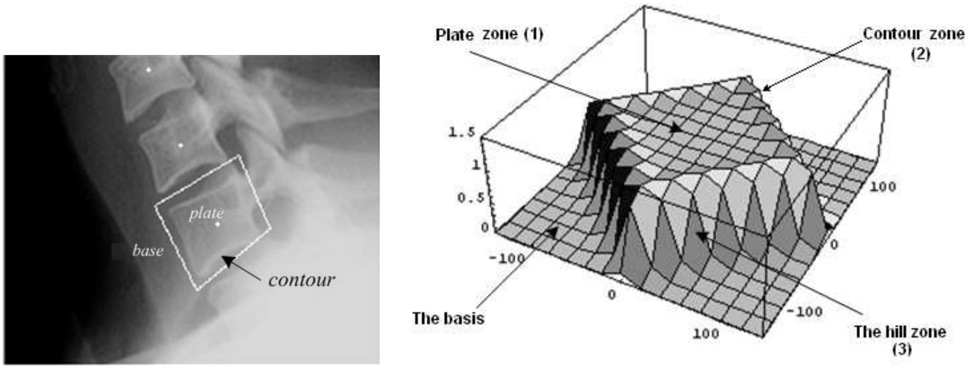
Figure 1: Results obtained by the process of vertebral regions selection. (a): original image reference; (b): inter vertebral points given by the template matching process; (c): boundary lines between vertebrae; (d): vertebrae regions.

3 Contours vertebrae detection

After the first step of region localisation, we proceed to vertebrae contours detection. For this second step, we propose a polar signature method [Lie and Chen, 1986], figure (5), combined with a second template matching approach. This method is applied to each vertebra region. A general approach to determine the polar signature of objects boundaries is illustrated in figure (4-b). We choosed to use this polar signature approach in order to explore all regions points likely to be corresponding to vertebrae contours.

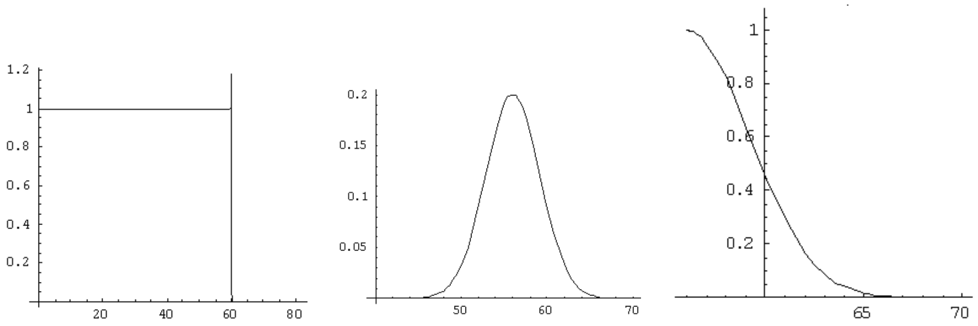
The approach that we choose for selecting the points of vertebrae contours is based on a template matching method. For this, we use a mathematical model based on the specific shape of each vertebra. To achieve the task of contour detection, we use a polar signature system, figure (4-b), associated to the proposed template matching method inside each region. This process allows the computation of some parameters characterizing each vertebra, like their positions, dimensions, orientation, and other cervical information.

The template function used for this seconde template matching process is defined according to the radial intensity distribution on each vertebra. So, from the figures (2-a, 2-b), we can notice three different zones. The first zone is relatively plate, figure (2-b, 3-a). The second one presents the areas surrounding the edge or the effective contour. This zone is generally



(a) (b)

Figure 2: (a): The vertebra Representation inside its region, as well as the three zones. (b): Three dimensional synthetic modeling of vertebrae intensity distribution (three zones: the plate zone, the effective contour and the zone towards the basis).



(a) Zone relatively plate (b) Vertebra contour (c) Interface towards the basis of the vertebra

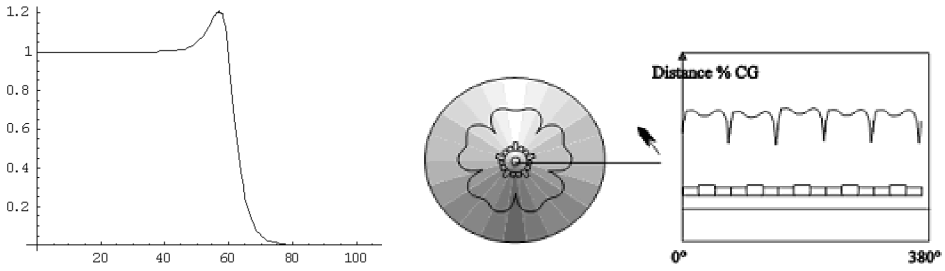
Figure 3: Decomposition of the vertebra model in three components.

similar to a Gaussian function, figure (2-b, 3-b). The third zone is the interface between the two preceding zones (gaussian and basis). This one presents the radial intensity relaxation until the basis. The third zone is similar to a hill, and can be simply represented by an exponential decreasing function figure (2-b, 3-c). These three zones could be described by a mathematical model composed of three terms, figure (3). We propose to modelize these three behaviours by a single function given by the equation (3), figure (4-a). We use this function as template for the task of contours vertebrae detection.

$$
\begin{aligned}
f_1(r) &= e^{-Tr} \\
f_2(r) &= e^{-T(r_0-r)} \\
f_F(r) &= \frac{f_1(f_1+f_2)}{(f_1^2+f_2^2)}
\end{aligned}
\tag{3}
$$

With r : the radial value and r_0 : the parameter which fixes the maximum value of f_F. T : A constant determining the decreasing speed related to the third zone.

We represent the sequence of points belonging to each vertebra contour by the vector $V[P(x_i, y_i)]$. To extract this sequence, we use a template matching process by using the function f_F, equation (3), and the image, figure (5-a). So, the first step is the template function displacement inside each vertebra region. For this, we use a polar signature system associated to each region, figure (5-b). We use the click points as the center points of this polar coordinate system. For the beginning direction, we chose the average direction between the frontal line



(a) (b)

Figure 4: (a): The template function proposed; (b): An example of contours points representation obtained by a polar signature approach.

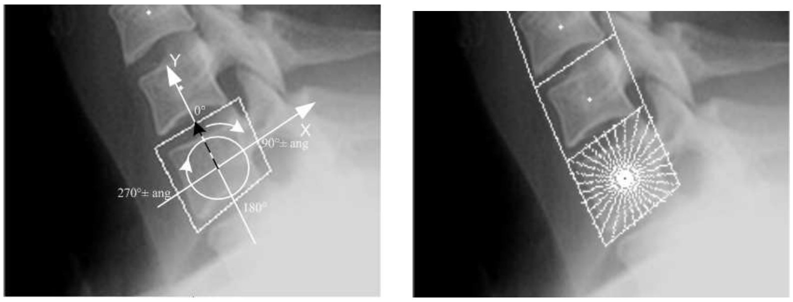
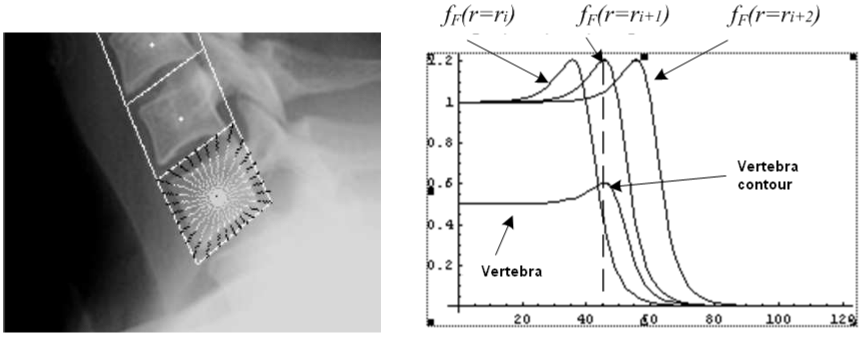


Figure 5: (a): The polar signature system; (b): The polar signature direction.



(a) (b)

Figure 6: (a): The intensity level representation of the template function f_F on the image, every $12°$ with a fixed value of r_0; (b): Synthetic representation of the template displacement for different value of r_0.

direction and the posterior line delimiting each region. We make turn the radial vector $360°$ around the central points with a step parameter, $\Delta\alpha$, figure (5-a, 5-b).

In figure (6-a), we represent the intensity level variation belonging to the function f_F inside vertebra region. We can notice that r_0 determines the maximum value of this function. Hence, for each value of α, we change also the value of r_0 in order to displace the maximum value of template function, figure (6-b).

The points $P(x_{\alpha i}, y_{\alpha i})$ corresponding to the maximum correlation value, $Max[Cr(\alpha, r_0)]$, between the template function and the image, for each degree α, will be taken as points of the contour $V[P(x_i, y_i)]$, figure (7-a). The figure (7-a) represents the set of points $V[P(x_i, y_i)]$ obtained by this template matching process combined with the polar signature system using the step $\Delta\alpha = 12°$.

Once the set $V[P(x_i, y_i)]$ of the contours points obtained, we proceed to extract each vertebra face by a process of corners vertebrae estimation, explained in next section (3.1). After this we apply a polynomial fitting, section (3.1), to each face in order to get a closed contour for each vertebra.

3.1 Polynomial fitting

To carry out calculations the most exactly, we start by detecting the four points which determine the corners of each vertebra, figure (7-a). This allows the separation of $V[P(x_i, y_i)]$ in four set of points, corresponding to the posterior face $VPF[P(x_i, y_i)]$, frontal face $VFF[P(x_i, y_i)]$, superior face $VSF[P(x_i, y_i)]$ and the inferior face $VIF[P(x_i, y_i)]$ of the vertebra, figure (7-b).

If the vertebrae were sufficiently square, the corners points would be localised at the points corresponding to the angles $45°$, $135°$, $225°$ and $315°$. Since vertebrae shapes are rather rectangular, that is not always the case. Therefore, a first step for corners detection consists in the Width/Height ratio (H/W) estimation. This estimation allows a better positioning of the angles corresponding to the changes of directions, i.e. the angles corresponding to the passage from each face to another.

To estimate the Height (H), we calculate the average distance between the central point $(P(x_i, y_i))$ and a fixed number of M points, $(M = 10)$, $(P(x_j, y_j))$ obtained by the previous step, around the directions 0° and 180°. For the width (W), we proceed in a similar way but around the directions 90° and 270°.

$$H = 1/M \sum_{j=\theta-M/2}^{j=\theta+M/2} D_\theta \quad W = 1/M \sum_{j=\theta-M/2}^{j=\theta+M/2} D_\theta$$

$$D_\theta = \sqrt{\left(P(x_i, y_i) - P(x^{\theta j}, y^{\theta j})\right)^2}$$

To determine corners position, we proceed in a similar way, but with the factor of correction (H/W). I.e. around the directions $45°H/W$, $135°H/W$, $225°H/W$ and $315°H/W$. So, from each group of points around these directions, we take the most distant sub-group from the center. From this sub-group we keep as corner position $(P_{corner}(x, y))$ the average position of the sub-group points, figure (7-a).

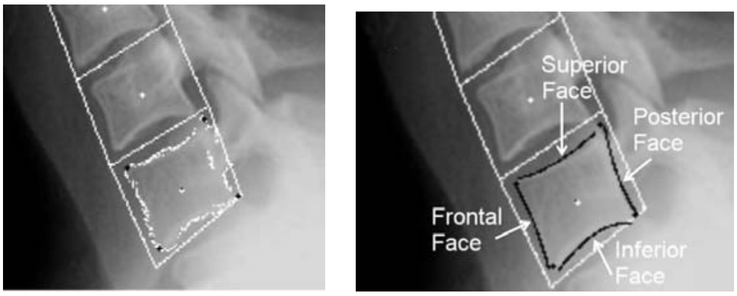


Figure 7: (a): Contour and corners points; (b): Polynomial fitting for each face of the vertebra

For a better approximation of vertebrae contours, we apply an edge closing method to the contours obtained. For this final step, we use a second degree polynomial fitting [Keren and Gotsman, 1999, Keren, 2004], applied to each face of the contour. We achieve this 2D polynomial fitting by the least square method, figures (7-a, 7-b).

4 Experimental results

We apply our method to a large set of X-ray images of the spinal column. The figure (8) shows the results obtained by applying the proposed method to two X-ray images of the spinal column. We notice that the process of region selection, figure (1), gives good results and

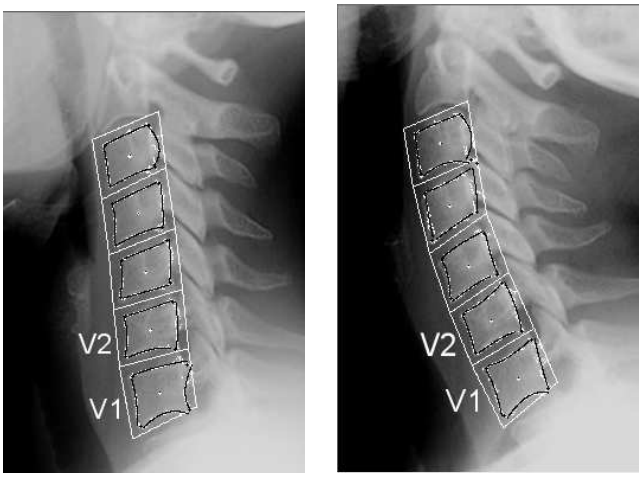
PF: Posterior face	FF: Frontal face	SF: Superior face	IF: Inferior face

Image (a)					Image (b)				
	PF	FF	SF	IF		PF	FF	SF	IF
V1	92.7	98.7	20.4	16.4	V1	105.4	111.5	33.5	31.0
V2	86.9	92.6	18.4	15.12	V2	107.2	112.0	38.1	31.2
V3	89.4	91.9	24.3	17.9	V3	99.4	109.7	29.9	35.4
V4	86.1	95.5	24.6	21.6	V4	93.6	99.7	22.1	30.5
V5	94.6	98.5	28.8	21.4	V5	97.0	97.0	25.4	16.9

Table 1: Orientation angles, in degree, of the vertebrae $V1$ to $V5$ in figures 8-a and 8-b, .

permits to isolate each vertebra separately in a polygonal area. On the other hand, contours extracted with the polar signature system combined with template matching process are given with high precision. The great advantage of our method is the fact that segmentation results are presented by closed contours. This will essentially facilitate the use of these results for image indexing and retrieval.

Vertebral mobility is estimated by the computation of the orientation angles belonging to each face of the contour.



(a) (b)

Figure 8: Final results, the points in white are the contours points obtained by the template matching process with polar signature, and points in black are the resulting faces by polynomial fitting.

In the table (1), we present some quantitative measurements of the orientation angle for each vertebra face. So, we present the orientation angles for the five vertebrae belonging to the images (a) and (b) in figure (8). This allows motion head estimation and vertebral mobility computation.

5 Conclusion

In this paper, a new method for vertebra segmentation has been proposed. The goal of this work was to develop a closed contours detection method aiming to represent each vertebra separately. This method permits to overcome some classical problems related to closed contours extraction and edge closing. Our approach lies on three steps. First, we proposed a new template matching method for the selection of each vertebra region. In the second step we formulated a new mathematical model for vertebra edge and used it in a template matching process by the mean of a polar signature system. The goal of this step was to extract the effective contour of each vertebra. Finally, we applied an edge closing method exploiting a polynomial fitting.

We have applied, with successful results, the method to a large set of real images. Vertebral mobility analysis has been represented by the angular variations measurements. For this task, we calculated the angular variations between two consecutive vertebrae within the same image.

The major advantage of the proposed polar coordinates contours description is the facility and the precision of the results. Nevertheless, if the precision is obtained by increasing azimuths number, computing time can be costly according to images complexity. In our future work, we are aiming to limit the number of clicks initially placed by the user to only one. Currently we are developing a content based image retrieval system by using the results presented in this paper.

References

[Benjelloun et al., 2006] Benjelloun, M., Téllez, H., and Mahmoudi, S. (2006). Template matching method for vertebra region selection. 2nd IEEE International Conference On Information and Communication Technologies: From Theory to Applications. (ICTTA'06), Damascus, Syria, April 2006.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence" Vol 8, No.6.

[Deriche and Faugeras, 1990] Deriche, R. and Faugeras, O. (1990). 2d curve matching using high curvature points: Application to stereo vision. Int Conf on Pattern Recognition, vol 1, pp 240-242.

[Howe et al., 2004] Howe, B., Gururajan, A., Sari-Sarraf, H., and Long, R. (2004). Hierarchical segmentation of cervical and lumbar vertebrae using a customized generalized hough transform. Proc. IEEE 6th SSIAI, p. 182-186, Lake Tahoe, NV. March.

[Kauffman and Guise., 1997] Kauffman, C. and Guise., J. (1997). Digital radiography segmentation of scoliotic vertebral body using deformable models. SPIE Medical Imaging vol 3034, pp 243-251.

[Keren, 2004] Keren, D. (2004). Topologically faithful fitting of simple closed curves. IEEE Transactions on PAMI 26(1).

[Keren and Gotsman, 1999] Keren, D. and Gotsman, C. (1999). Fitting curves and surfaces with constrained implicit polynomials. IEEE Transactions on PAMI 21(1).

[Lie and Chen, 1986] Lie, W. N. and Chen, Y. C. (1986). Shape representation and matching using polar signature. Proc. Intl. Comput. Symp. 1986, 710-718, 1986.

[Long and Thoma, 1999] Long, L. and Thoma, G. (1999). Segmentation and feature extraction of cervical spine x-ray images. Proc. SPIE Medical Imaging 1999: Image Processing, vol. 3661, San Diego, CA, February 20-26, 1037-1046,.

[LR and GR, 2000] LR, L. and GR, T. (2000). Use of shape models to search digitized spine x-rays. Proc. IEEE Computer-Based Medical Systems, 255-60, Houston, TX, June,.

[LR and GR, 2001] LR, L. and GR, T. (2001). Identification and classification of spine vertebrae by automated methods. Proc. SPIE Medical Imaging 2001: Image Processing, vol. 4322.

[Rico et al., 2001] Rico, G., Benjelloun, M., and Libert, G. (2001). Detection, localization and representation of cervical vertebrae. Computer Vision Winter Workshop, Slovenia, pp. 114-124, February.

[RJ et al., 2004] RJ, S., LR, L., S, A., GR, T., and Edward, D. (2004). image analysis techniques for the automated evaluation of subaxial subluxation in cervical spine x-ray images. Proceeding of the 17th IEEE symposium on computer-based medical systems CMBS04 2004.

[Roberts et al., 2003] Roberts, M., Cootes, T., and Adams, J. (2003). Linking sequences of active appearance sub-models via constraints: an application in automated vertebral morphometry. BMVC2003, Vol.1 pp.349-358,.

[Tezmol et al., 2002] Tezmol, A., Sari-Sarraf, H., Mitra, S., Long, R., and Gururajan, A. (2002). A customized hough transform for robust segmentation of cervical vertebrae from x-ray images. Proc. 5th IEEE Southwest Symposium on Image Analysis and Interpretation, santa Fe, NMexico, USA.

[Verdonck et al., 1998] Verdonck, B., R.Nijlunsing, Gerritsenand, F., Cheung, J., Wever, D., Veldhuizen, A., Devillers, S., and Makram-Ebeid, S. (1998). In proceeding of International Conference of Computing and Computer Assisted Interventions, LNCS, pages 822-831. Springer, 1998,.

Evaluation of 3D gradient filters for estimation of the surface orientation in CTC

Tarik A. Chowdhury, Ovidiu Ghita and Paul F. Whelan
Vision Systems Group
School of Electronic Engineering
Dublin City University
Dublin 9, Ireland
tarik@eeng.dcu.ie

Abstract

The extraction of the gradient information from 3D surfaces plays an important role for many applications including 3D graphics and medical imaging. The extraction of the 3D gradient information is performed by filtering the input data with high pass filters that are typically implemented using 3×3×3 masks. Since these filters extract the gradient information in small neighborhood, the estimated gradient information will be very sensitive to image noise. The development of a 3D gradient operator that is robust to image noise is particularly important since the medical datasets are characterized by a relatively low signal to noise ratio. The aim of this paper is to detail the implementation of an optimized 3D gradient operator that is applied to sample the local curvature of the colon wall in CT data and its influence on the overall performance of our CAD-CTC method. The developed 3D gradient operator has been applied to extract the local curvature of the colon wall in a large number CT datasets captured with different radiation doses and the experimental results are presented and discussed.

Keywords: 3D gradient operators, CTC, polyp detection, local curvature.

1 Introduction

Computer Tomography Colonagraphy (CTC) [1-4] is a rapidly evolving minimally invasive technique for early detection of colorectal polyps and nowadays the medical community views this medical procedure as a viable alternative to optical colonoscopy. As the performance of the CT imaging modalities is constantly improving, recent studies have demonstrated that the sensitivity in polyp detection offered by the CTC compares favorably with the sensitivity offered by the optical colonoscopy [5,6]. With the introduction of the new generation of multi-slice CT scanners that are able to produce high resolution CT data, the CT datasets generate a sheer volume of information required to be interpreted by the radiologist and this task is performed by analyzing either the 2D axial views or the 3D surface of the colon wall. The visual analysis of the CT data is a time consuming procedure and the examination results are biased by the subjectivity and the experience of the radiologists. This fact encouraged the development of automated computer aided detection (CAD)-CTC systems that are able to produce reproducible results with high sensitivity in detection of clinically significant polyps (>5mm). The main problem associated with the current range of developed CAD-CTC systems is the large number of false positives that are generated by other colon structures that mimic the shapes of the polyps (haustral folds, residual material, etc) [7-9]. The large number of false positives is generated by the subtle difference in shape between the polyps and other colon structures but also by the errors in the assessment of the local curvature (convexity) of the colon wall. In this paper we attempt to evaluate the contribution of the image noise (and the partial volume effects generated by the relatively low resolution in the *z* axis) in the estimation of the local curvature of the colon wall.

In order to determine the surface orientation we need to extract first the local derivatives from the 3D data. In 2D data the normal vector to a curve can be calculated by computing the local derivatives in the *x* and *y* direction using high pass local operators. The CTC datasets are 3D and most algorithms that perform automatic identification of colorectal polyps evaluate the local curvature of the colon wall by calculating in 3D the partial derivatives [7]. As the polyps are structures with a well-defined convex appearance we need to evaluate the measure of convexity by evaluating the normal intersection around suspicious colon structures. Nonetheless, this method will be successful if we are able to accurately extract the surface orientation, i.e. the normal vector. In this regard an elegant solution to this problem is the extension of the calculation of the normal vector in 2D to 3D data. In this regard, Zucker and Hummel [10] proposed a mathematical model to determine the optimal 3D gradient operators. In their formulation they determined the masks of the gradient operator using a functional analysis and their theory was just in fact the generalization of the ubiquitous 2D Sobel operator [11]. The optimal gradient operator described in their paper is a 3×3×3 anti-symmetric operator that is applied in 3 different directions (the Zucker-Hummel operator uses radial functions that smooth the calculated gradient. This will have a positive effect in cases where the Zucker-Hummel operator is applied to noisy data). One limitation of this operator is the small kernel that is used to sample the gradient in *x*, *y*, *z* directions and as a consequence the results are only modest in extracting the local orientation for complex surfaces such as roofs or cavities. This is highlighted in our experiments where is indicated that the Zucker-Hummel operator is in many cases outperformed by the standard 3D Sobel operator. The aim of this paper is to evaluate the influence of the selection of the gradient operator on the overall performance of our CAD-CTC and to detail the mathematical model that will allow us to implement optimized 3D gradient operators.

2 Mathematical background of gradient detection

In image processing the gradient operators are widely used to identify strong features in the image such as edges or the local orientation of the curves and surfaces. The extraction of local derivative from a continuous signal can be done by applying directly the well-known derivative formula:

$$Der(f(x)) = \lim_{\alpha \to 0} \frac{f(x+\alpha) - f(x)}{\alpha} \tag{1}$$

When designing a gradient operator we should bear in mind that the image data is discrete and we cannot apply the finite differences without compromising the accuracy of the gradient approximation [12-15]. Thus we have to assume that the original continuous optical signal that generates the image has been uniformly sampled at a rate of *T* samples per length. Using the Nyquist sampling theorem the continuous signal can be reconstructed from these discrete samples as follows:

$$f(x) = \sum_{k=1}^{N} f[k] s(x - KT), \; s(x) = \frac{\sin x}{x} \tag{2}$$

In equation 2 the term *f*[*k*] represent the discrete sampled signal and *s(x)* defines the sampling function that can be approximated with the *sinc* function. Hence, to obtain the gradient of the discrete signal we have to derivate the reconstructed signal *f(x)* that is depicted in equation 2.

$$Der(f(x)) = \sum_{k=1}^{N} f[k] der(s(x - KT)) = \sum_{k=1}^{N} f[k] s'(x - KT) \tag{3}$$

where s' represents the derivative of the *sinc* function. As the derivative of the *sinc* function is dependent on the sampling frequency, it is worth noting that the spectrum of the discrete signal is bounded by $2\pi / T$ that is in agreement with the sampling theorem. We note that the derivative of *sinc* signal decays relatively slowly and the implementation of an optimal gradient filter would require large filters that are not feasible to be applied in practice due to the onerous computational cost required to extract the partial derivatives. Next, we will introduce a practical method to design one-dimensional (1-D) gradient filters whereas the generalization to multiple dimensions is a relatively simple task. In order to design gradient operators that are to be applied to discrete signals we have to consider several constraints. The vision literature indicates that the gradient filters are anti-symmetric and usually have an odd order. Thus, the 1-D gradient filter can be represented in the following generic form:

$$d(k) = [d_{-N}, d_{-N+1}, \ldots, d_{-1}, 0, d_1, \ldots, d_{N-1}, d_N], \quad d_{-k} = -d_k, \quad k = 1, \ldots, N \tag{4}$$

In order to design 1-D derivative filters we need to impose several constraints for parameters d_k as illustrated in the following expressions [14]:

$$\sum_{k=-N}^{N} d_k = 0 \tag{5}$$

$$\sum_{k=-N}^{N} d_k k = 0 \tag{6}$$

In this way, equation 5 translates in the requirement that the derivative filter should have the sum of the coefficients equal to 0, while equation 6 can be used to select the values for d_k coefficients. The derivative operator has to fulfill the condition illustrated in equation 5 to achieve insensitivity to DC signals. Since the derivative filters are anti-symmetric the first coefficient of the operator can be determined using the following relationship:

$$d_1 = \frac{1}{2} - \sum_{k=2}^{N} d_k \tag{7}$$

Using the formulas illustrated in equations 4 to 7, the 5×5×5 derivative filter that is applied to extract the gradient in the x direction has the following mask [-1 8 0 -8 1]/12 • [1 4 6 4 1]/16, where • defines the convolution operator. To extract the gradient for other directions we need to rotate the 5×5×5 mask in the direction required for a particular axis. It can be noted that this operator, as expected, represents the direct extension of the 5×5 Sobel operator to the 3D case. Using equations 5 to 7, we have developed a new 5×5 gradient operator that implements a two-peak operator. Since the gradient operator has two lobes it will provide improved performance when applied to data with step discontinuities or 3D CT datasets defined by a low signal to noise ratio such as the low-dose CT data.

$$\begin{matrix} -.125 & -.25 & -.5 & -.25 & -.125 \\ -.25 & -.5 & -1.0 & -.5 & -.25 \\ -.5 & -1.0 & -2.0 & -1.0 & -.5 \\ -.25 & -.5 & -1.0 & -.5 & -.25 \\ -.125 & -.25 & -.5 & -.25 & -.125 \end{matrix} \qquad \begin{matrix} -.25 & -.5 & -1.0 & -.5 & -.25 \\ -.5 & -1.0 & -2.0 & -1.0 & -.5 \\ -1.0 & -2.0 & -4.0 & -2.0 & -1.0 \\ -.5 & -1.0 & -2.0 & -1.0 & -.5 \\ -.25 & -.5 & -1.0 & -.5 & -.25 \end{matrix} \qquad 0_{5\times5}$$

$$\begin{matrix} .25 & .5 & 1.0 & .5 & .25 \\ .5 & 1.0 & 2.0 & 1.0 & .5 \\ 1.0 & 2.0 & 4.0 & 2.0 & 1.0 \\ .5 & 1.0 & 2.0 & 1.0 & .5 \\ .25 & .5 & 1.0 & .5 & .25 \end{matrix} \qquad \begin{matrix} .125 & .25 & .5 & .25 & .125 \\ .25 & .5 & 1.0 & .5 & .25 \\ .5 & 1.0 & 2.0 & 1.0 & .5 \\ .25 & .5 & 1.0 & .5 & .25 \\ .125 & .25 & .5 & .25 & .125 \end{matrix}$$

Figure 1. The masks of the 5×5×5 3D OptDer operator to extract the gradient in the z axis (the mask 0_{5x5} indicates a 5×5 mask where all elements are zero).

In our experiments we have evaluated the efficiency of several filters including the 3×3×3 Zucker-Hummel operator, 5×5×5 Sobel operator and 5×5×5 optimized operator – OptDer filter (is a two peak derivative operator. For more details about the implementation of optimal derivative filters refer to [12,13]). A particular interest we had in assessing the performance of these gradient operators when applied to CTC datasets that have been acquired with different x-ray dose levels (the lower the radiation dose the higher the image noise). However in our experiments it has became clear that the 3×3×3 gradient masks are inefficient in sampling the correct curvature of the colon wall when dealing with irregular surfaces. The experimental data indicated that the optimized 5×5×5 gradient operator was able to return improved performance (this operator has been designed using the masks illustrated in Figure 1). The measurements were performed on CTC prone and supine views where the reconstruction interval was set to 1.5mm. The tests were conducted on phantom (synthetic) data and on real patient data. Of particular interest was the evaluation of the level of false positives detected by the automated CAD-CTC system and a detailed performance of our system is illustrated in Tables 1 to 5 where different gradient operators are evaluated.

3 CAD-CTC Polyp Detection Algorithm

We have developed an automated CAD-CTC method designed to identify the colorectal polyps in CT data [16] that evaluates the local morphology of the colon wall. Initially, the colon is segmented using a seeded 3D region growing algorithm that was applied to identify the interface between the air voxels and the colon tissue, which assures the extraction of the colon wall. After the identification of the colon wall, for each colon wall voxel the surface normal vector is calculated using the Hummel-Zucker operator, Sobel and OptDer operator. The normal vectors sample the local orientation of the colon surface and the suspicious candidate structures that may resemble polyps are extracted using a convexity analysis where the colon suspicious surfaces are detected by evaluating the distribution of the normal vectors intersections in the 3D space (for a detailed description of this algorithm refer to [16]). This method is able to correctly identify all polyps above 3mm but it is worth noting that this is achieved at the cost of a high level of false positives. In order to reduce the level of false positives, statistical features [16] including the standard deviation of surface variation, ellipsoid fitting error, sphere fitting error, three axes of the ellipsoid and the Gaussian sphere radius are calculated for each candidate surface that has been identified by the convexity method described before. These features are used as inputs for a nearest neighbor classifier that is trained to decide whether the surface under investigation belongs to a polyp or a fold. The classifier was trained using a collection of 64 polyps and 354 folds that were selected by a radiologist from Mater Misericordiae Hospital, Dublin.

4 Results and Discussions

In our tests we have used 52 standard dose (100mAs) patient datasets (prone and supine views) with 75 polyps, 9 low dose (13-50mAs) patient data with 2 small polyps and phantom data (low-dose and standard dose) with 48 polyps of various sizes and shapes. All patients were scanned at 120kVp, 13mAs-100mAs, 2.5mm collimation, 3mm slice thickness, 1.5mm reconstruction interval, and 0.5s gantry rotation. The CT acquisition was performed with the patient head-first supine position and then repeated with the patient in the prone position. The CT protocol mentioned before generates CT datasets where the number of axial slices is in the range 200-350 and is dependent on the height of the patient.

Table 1: Sensitivity for synthetic phantom data (polyps >=10mm).

mAs	Total	Sensitivity (%)		
		Zucker	Sobel	OptDer
100	14	100	100	100
40	14	100	100	100
30	14	100	92.85	100
20	14	100	100	100
13	14	92.85	92.85	100

Table 2: Sensitivity for synthetic phantom data (polyps [5-10)mm).

mAs	Total	Sensitivity (%)		
		Zucker	Sobel	OptDer
100	20	100	100	100
40	20	100	100	100
30	20	95	90	95
20	20	100	100	95
13	20	95	95	100

Table 3: Sensitivity for synthetic flat polyps.

mAs	Total	Sensitivity (%)			False Positive per dataset		
		Zucker	Sobel	OptDer	Zucker	Sobel	OptDer
100	9	55	55	44.44	1	1	1
40	9	33.33	33.33	44.44	2	1	1
30	9	44.44	44.44	55	0	2	2
20	9	11.11	33.33	44.44	2	2	2
13	9	22.22	22.22	44.44	2	2	3

Table 4: Sensitivity for polyps >=5mm in real patient standard dose (100mAs) data.

mAs	Total	Sensitivity (%)			False Positive per dataset		
		Zucker	Sobel	OptDer	Zucker	Sobel	OptDer
100	18	88.89	88.89	88.89	4.32	4.69	4.71

Table 5: Sensitivity for polyps <5mm in real patient's standard and low dose data

mAs	Total	Sensitivity (%)		
		Zucker	Sobel	OptDer
100	48	60.41	60.41	68.75
13-40	2	50	100	100

When the CAD-CTC system was applied on phantom data the OptDer operator shows 100% sensitivity for polyp >=10mm for datasets acquired with radiation doses in the range 100-13mAs where the Zucker-Hummel and Sobel operators shows 92.85% sensitivity at 30mAs and 13mAs radiation doses (see Table-1). Figure 2(a) illustrates the 3D surface extraction for a 12mm polyp when the Zucker-Hummel operator was applied for detection of the surface normal vectors and Figure 2(b) shows the surface extraction when the OptDer operator has been used. Figure 3 illustrate the surface extraction for an 8 mm phantom polyp from a dataset scanned with 13mAs. It can be noted that in both cases the CAD-CTC system achieved a more accurate surface extraction when the OptDer operator was employed. Due to incomplete surface segmentation our CAD-CTC system missed the polyp illustrated in Figure 2 when the Zucker-Hummel operator was used to extract the surface normal vectors (see Table 1), whereas the polyp was correctly detected when the OptDer operator was applied. In Figures 2 and 3 it can be also observed that the OptDer operator generates better surface normal concentration than the Zucker-Hummel operator. The application of the OptDer operator to extract the surface normal vectors offers better detection for polyps in the range 5-10mm than the Sobel operator (see Table 2). It also provides a better detection of flat polyps when compared to the performance of the Zucker-Hummel and Sobel operators (see Table 3). When the Zuker-Hummel, Sobel and OptDer operators were used to calculate the surface normals of the colon wall for standard dose real patient datasets, the sensitivities for the detection of polyps >=5mm were 88.89% (see Table 4) for all operators, but the OptDer operator provides higher sensitivity (see Table 5) in the detection of small polyps (<5mm) than the other two operators. Table 5 indicates that the overall sensitivity for polyp detection was highest when the OptDer operator was used and the experimental data indicates that this operator outperformed the Zucker-Hummel and the Sobel operators especially when the system is applied to low-dose datasets. The level of noise sampled by the standard deviation calculated in a local 5×5×5 neighborhood increased with a factor of 2.67 (SD = 26.59 for 100mAs and SD = 70.95 for 13mAs) when the scan was performed at 13mAs when compared to the case when the phantom was scanned with 100mAs radiation dose. The relation between the noise level and the radiation dose is illustrated in Figure 4.

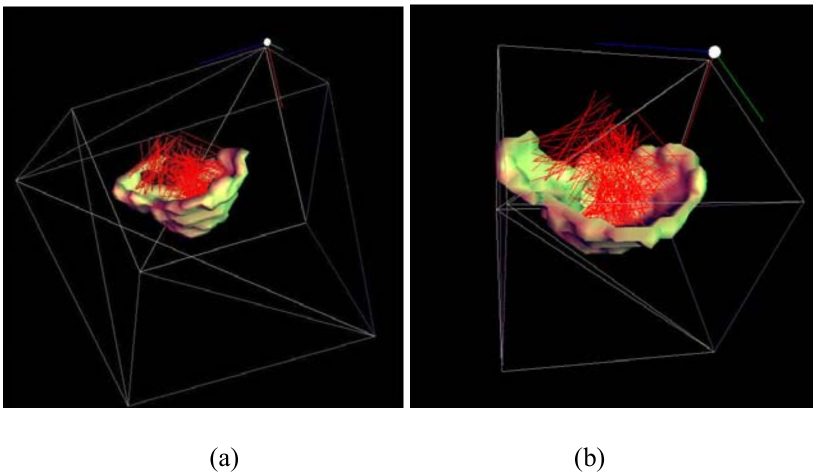


Figure 2. 3D surface extraction of a 12mm phantom polyp (radiation dose 13mAs). (a) The 3D surface extracted by the CAD-CTC system using the Zucker-Hummel operator. (b) The 3D surface extracted by the CAD-CTC system using the OptDer operator.

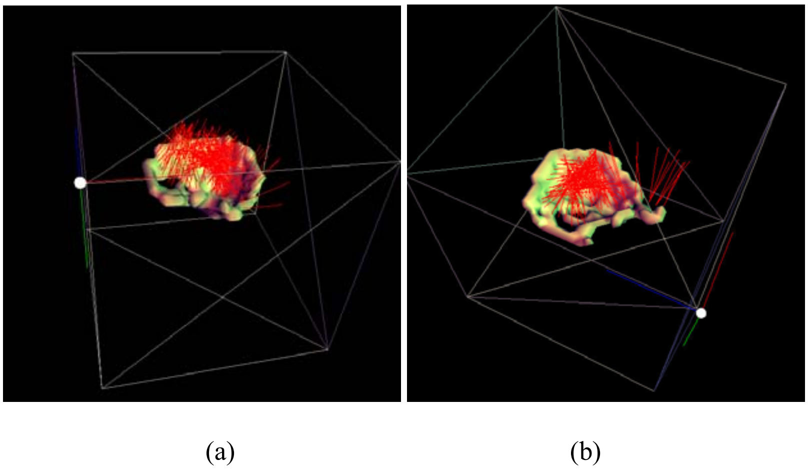


Figure 3. 3D surface extraction of a 12mm phantom polyp (radiation dose 13mAs). (a) The 3D surface extracted by the CAD-CTC system using the Zucker-Hummel operator. (b) The 3D surface extracted by the CAD-CTC system using the OptDer operator.

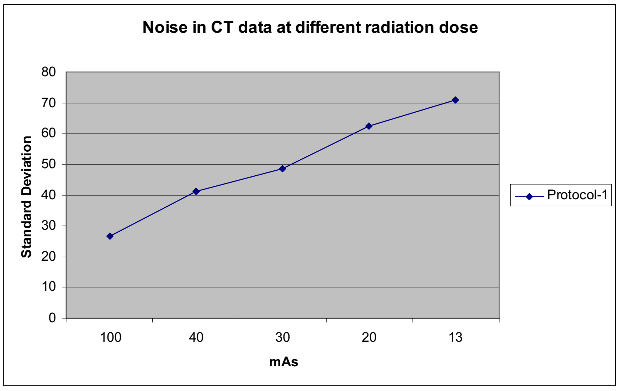


Figure 4. The relationship between the noise level and the radiation dose.

5 Conclusions

In this paper we have addressed the problem of robust calculation of the surface curvature in 3D CT data. As numerous automated CAD-CTC systems identify the colorectal polyps based on analysing the local convexity of the colon surface, one of the most important steps in this analysis is the precise identification of normal vector. In this regard, we have investigated a number of 3D gradient operators and we have conducted the experiments on a large number of synthetic and real patient data. Experimental data indicated that the commonly used 3D gradient operators such as Zucker-Hummel and Sobel fail to correctly determine the normal vector when dealing with datasets characterized by a low signal to noise ratio. To address this problem we have proposed a new gradient operator that was able to return better performance when applied to CT data that is acquired with different radiation dose levels.

Acknowledgements

We would like to acknowledge the contributions of our clinical partners in this project: Dr. Helen Fenlon (Department of Radiology) and Dr. Padraic MacMathuna (Gastrointestinal Unit) of the Mater Misericordiae Hospital, Dublin. This work was supported under an Investigator Programme Grant (02/IN1/1056) by Science Foundation Ireland (SFI).

References

[1] Vining D.J, Gelfand D.W., Bechtold R.E (1994), Technical feasibility of colon imaging with helical CT and virtual reality, AJR, 162, 104.

[2] Johnson C.D., Hara A.K., Reed J.E. (1998), Virtual endoscopy: what's in a name?, AJR, 171, 1201-2.

[3] Hara A.K., Johnson C.D., Reed J.E., Ahlquist D.A., Nelson H., Ehman R.L., McCollough C., Ilstrup D.M. (1996), Detection of Colorectal Polyps by CT Colonography: Feasibility of a novel technique, Gastroenterology, vol. 100:284-290.

[4] Johnson C.D., Hara A.K. (2000), CT colonography: the next colon screening examination?, Radiology, vol. 216, pp. 331-341.

[5] Fenlon H.M., Nunes D.P., Schroy P.C., Barish M.A., Clarke P.D., Ferrucci J.T. (1999), A comparision of virtual and convention colonoscopy for the detection of colorectal polyps, N. Engl J Med, 341:14961503.

[6] Pickhardt P.J., Choi J.R., Hwang I., Butler J.A., Puckett M.L., Hildebrandt H.A., Wong R., Nugent P.A., Mysliwiec P.A., Schindler W.R. (1997), Virtual colonoscopy for colorectal polyp detection, RBM; 19(5): 143-147.

[7] Summers R.M., Johnson C.D., Pusanik L.M., Malley J.D., Youssef A.M., Reed J.E. (2001). Automated polyp detection at CT colonography: Feasibility assessment in a human population, Radiology 219:51-59.

[8] Yoshida H., Masutani Y., MacEneaney P., Rubin D.T., Dachman A.H. (2002), Computerized detection of colonic polyps at CT colonography on the basis of volumetric features: Pilot study, Radiology, vol. 222, pp. 327-336.

[9] Paik D.S., Beaulieu C.F. et al. (2004), Surface normal overlap: a computer-aided detection algorithm with application to colonic polyps and lung nodules in helical CT, IEEE Transactions on Medical Imaging, 23(6):661-675.

[10] Zucker S.W., Hummel R.A. (1981), A Three-Dimensional edge operator, IEEE Transactions on Pattern Analysis and Machine Intelligence, 3(3), pp.324-331.

[11] Whelan P.F., Molloy D. (2000), Machine Vision Algorithms in Java: Techniques and Implementation, Springer (London), ISBN 1-85233-218-2

[12] Farid H., Simoncelli E.P. (2004), Differentiation of Multi-Dimensional Signals, IEEE Transactions on Image Processing, 13(4):496-508.

[13] Simoncelli E.P. (1994), Design of multi-dimensional derivative filters, IEEE International Conference on Image Processing, vol. 1:790-793.

[14] Jahne B., Scharr H., Korkel S. (1999), Principles of filter design, Handbook of Computer Vision and Applications, Academic Press.

[15] Freeman W.H., Adelson E. H (1991), The design and use of steerable filters, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13:891-906.

[16] Chowdhury T.A., Ghita O., Whelan P.F. (2005), A statistical approach for robust polyp detection in CT colonography, 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1-4 September 2005, Shanghai, China

Object & Event Recognition

Space Variant Feature Extraction for Omni-directional Images

Dermot Kerr[1], Bryan Scotney[2], Sonya Coleman[1]

[1]School of Computing and Intelligent Systems
University of Ulster
Magee, BT48 7JL
{kerr-d2, sa.coleman}@ulster.ac.uk

[2]School of Computing and Information
Engineering
University of Ulster
Coleraine, BT52 1SA
bw.scotney@ulster.ac.uk

Abstract

In recent years, the use of omni-directional cameras has become increasingly more popular in vision systems and robotics. To date, most of the research relating to omni-directional cameras has focussed on the design of the camera or the way in which to project the omni-directional image to a panoramic view rather than on how to process these images after capture.

Typically images obtained from omni-directional cameras are transformed to sparse panoramic images that are interpolated to obtain a complete panoramic view prior to low level image processing. This interpolation presents a significant computational overhead with respect to real-time vision. We present an approach to real-time vision that projects an omni-directional image to a sparse panoramic image and directly processes this sparse image. Feature extraction operators previously designed by the authors are used in this approach but this paper highlights the reduction of the computational overheads of processing images arising from omni-directional cameras through efficient coding and storage, whilst retaining accuracy sufficient for application to real-time robot vision.

Keywords: Omni-directional imaging, Feature detection.

1 Introduction

Panoramic vision systems have been developed based on the natural biological vision systems of arthropods such as insects [1]. The arthropod's vision system is simple when compared with the complex eye movement system of humans. Arthropods have compound eyes that, although immobile with respect to their body, have a natural advantage gained from a wide field of view [2]. This wide field of view would be of benefit to mobile robot localisation, as it allows many landmarks to be simultaneously present in the scene, without the need for an elaborate pan-tilt active vision system that would mimic human gaze control. Catadioptric cameras combine conventional cameras and mirrors to obtain a 360° omni-directional view similar to the wide field of view that a compound eye obtains. The camera is pointed vertically towards a mirror, with the optical axis of the camera lens aligned with the mirror's axis. The mirror profiles that can be used are conical or convex, such as spherical, parabolic, and hyperbolic. These cameras offer advantages to robot navigation systems mainly due to the fact that they provide the robot with a complete 360° omni-directional view of its immediate surroundings [3, 4], allowing many landmarks to be in view of the robot simultaneously, and they offer the ability to the robot to change viewing direction instantaneously. They enable a single feature to be tracked from varying viewpoints, whereas if a fixed camera were used, the feature may track out of the field of view. However, their disadvantage is that they capture images of a lower resolution than a standard image. The reduced resolution may be a problem for tasks such as fine manipulation or control, but for navigation tasks the wide field of view would be of greater value than higher resolution.

Previous research using omni-directional images for robot navigation includes Gaspar [1], in which the omni-directional image is un-warped to both panoramic and birds-eye-view images. An appearance based navigation system is used that has a set of reference birds-eye-view images and a topological map of the environment for navigation. Another appearance based method is used by Matsumoto [5] in which omni-directional images are un-warped to a panoramic view before template matching of images is performed. Yagi [6] also un-warps the omni-directional image to a panoramic view before detecting edges using the Sobel edge detector. The edge maps are then used to create a one dimensional projections relating to the edge density of the horizontal axis of the edge map for the purpose of navigation. A different approach is taken by Vlassis [3], where the omni-directional image is not un-warped. In this approach the Sobel edge detector is used to extract edges from the omni-directional image, and the edge pixels are then fed to a Parzen density estimator to calculate the edge density. Principal component analysis is then used to reduce the dimensionality of the feature vector that is used in the navigation system.

As discussed above, the omni-directional image may be used in the original format [3], or it may be un-warped to a rectangular panoramic image through the use of an un-warping algorithm [7, 8, 9]. In order to un-warp the omni-directional image to a panoramic image, it is typical for the circular omni-directional image to be projected onto a cylinder, transforming the omni-directional image to an un-warped rectangular panoramic image. Previous research has highlighted advantages of un-warping the omni-directional image. Matsumoto [5] highlights the fact that a system that can deal with shifts of panoramic images using template matching in hardware has lower computational and storage costs than a system that can cope with rotations of omni-directional images. Krose [10] also cites this as a benefit, along with the fact that standard feature trackers may be used on the panoramic image.

In this paper we discuss our method of extracting features directly from a sparse un-warped panoramic image. We will show that our method retains the accuracy of traditional 4- and 8-pixel neighbourhood feature extraction operators on complete images, whilst achieving a reduced computational overhead due to avoidance of image interpolation. Section 2 discusses methods of un-warping an omni-directional image to a panoramic image. Section 3 introduces our method of space variant feature detection, while Section 4 discusses implementation and implementation issues such as efficient coding and storage. In Section 5 we present processing times and sample edge maps using various edge detection methods for comparison, and Section 6 summarises our findings and outlines future work.

2 Sparse Panoramic Images

We obtained omni-directional images, as shown in Figure 1, using a mobile robot with a catadioptric camera, shown in Figure 2.

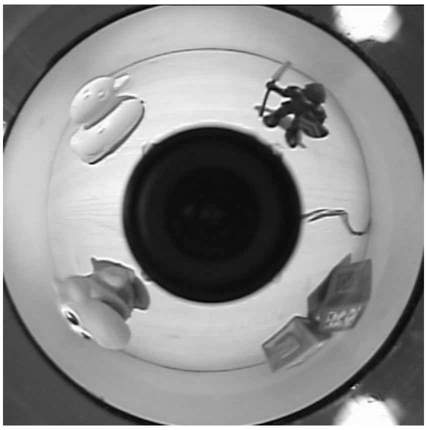


Figure 1. Omni-directional image

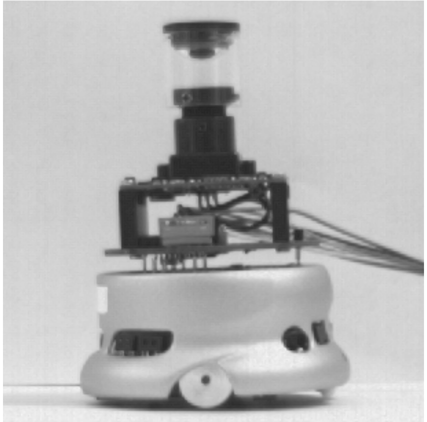


Figure 2. Khepera robot used to obtain images

One approach to un-warping of omni-directional images is by back-projection of each pixel in an un-warped panoramic image to a position in the omni-directional image by means of a polar-to-cartesian transformation:

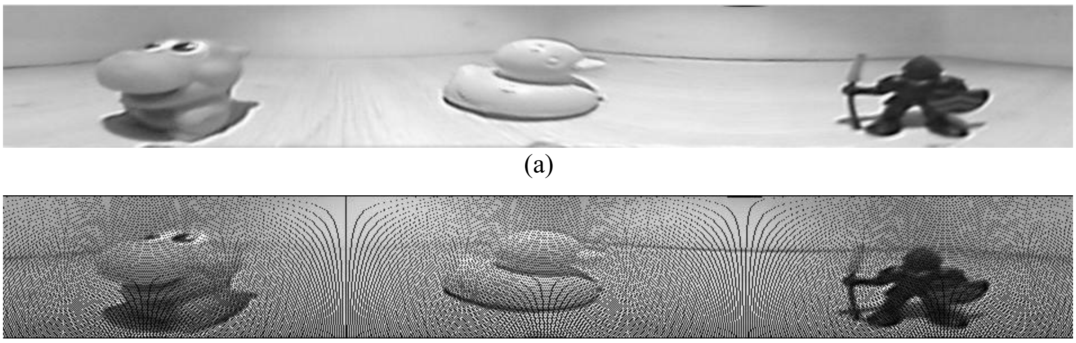
$$g(r,\theta) = f(x,y) = f(r\cos\theta, r\sin\theta) \quad (1)$$

where $g(r,\theta)$ represents the polar coordinate image and $f(x,y)$ represents the rectangular coordinate image. The value $g(r,\theta)$ in the omni-directional image is obtained by bilinear interpolation of the four pixel values in the 4-pixel neighbourhood of (r,θ). A three-quarter section of the un-warped version of the omni-directional image shown in Figure 1 obtained by using back-projection method is shown in Figure 3(a). Although bilinear interpolation is generally cheap to implement, such a back-projection has the disadvantage of relying on secondary reconstructed image data.

A second approach to un-warping is to forward-project each pixel in the omni-directional image to a position in the un-warped panoramic image by means of a cartesian-to-polar transformation:

$$f(x,y) = g(r,\theta) = g\left(\sqrt{x^2+y^2}, \tan^{-1}\frac{y}{x}\right) \tag{2}$$

where $f(x,y)$ represents the rectangular coordinates image and $g(r,\theta)$ represents the polar coordinates image. This un-warping approach leads to a sparse representation of the panoramic view, as shown in Figure 3(b), where the missing values depicted in black. Typically post-processing would be required, such as applying interpolation to this image, in order to obtain a complete data set before any further image processing. Our approach however is to work directly on the space variant panoramic image, thus avoiding the use of secondary reconstructed image data values.



(a)

(b)

Figure 3. (a) Back-projected un-warped image; (b) Forward-projected un-warped image

3 Space Variant Operators

In order to extract features from the space variant panoramic image described in Section 2 in real-time, we use the family of autonomous finite element based image processing operators presented in [11] for use on non-uniformly sampled intensity images and in [12] for use on range images. Here, the term autonomous indicates that these operators were developed in such a way that they can change size and shape across the image plane in accordance with the local pixel distribution as illustrated in Figure 4.

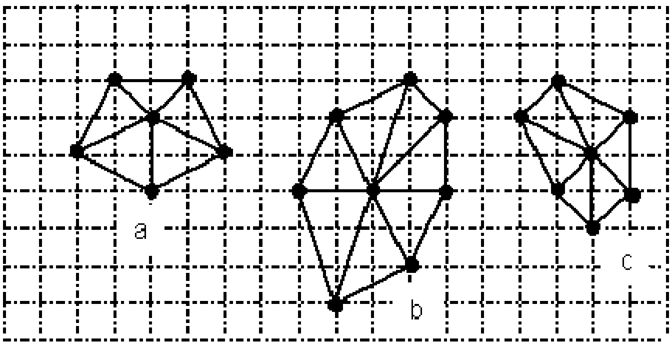


Figure 4. Neighbourhoods of Autonomous Operators

These feature detection operators correspond to weak forms of operators in the finite element method and can be based on first or second order derivative approximations, corresponding to a first directional derivative $\partial u/\partial b \equiv \underline{b} \cdot \underline{\nabla} u$ and a second directional derivative $-\underline{\nabla} \cdot (\mathbf{B}\underline{\nabla} u)$, and are defined by the functionals

$$E_i^\sigma(U) = \int_\Omega \underline{b}_i \cdot \underline{\nabla} U \psi_i^\sigma d\Omega \text{ and } Z_i^\sigma(U) = \int_\Omega \underline{\nabla} U \cdot (\mathbf{B}_i \underline{\nabla} \psi_i^\sigma) d\Omega \,. \tag{3}$$

Here, U is our representation of the image data, $\underline{b} = (\cos\theta, \sin\theta)$ is an image-dependent unit direction, $\psi_i^{\sigma_m}(x,y) = \frac{1}{2\pi\sigma_m{}^2} e^{-\left(\frac{(x-x_i)^2+(y-y_i)^2}{2\sigma_m{}^2}\right)}$ and **B** = $\underline{b}\ \underline{b}^{\mathrm{T}}$. In the special case of the Laplacian operator, represented by Z_i^σ, **B** is taken to be the identity matrix. By choosing each function ψ_i^σ to be a Gaussian function restricted to a neighbourhood Ω_i^σ, each operator is restricted to the neighbourhood Ω_i^σ.

As these operators can be applied directly to space variant image data, we can apply them directly to the forward-projected un-warped panoramic image without the need for additional image reconstruction. The application of these operators to space variant panoramic images yields edge maps that are comparable to those obtained using well-known image processing operators on complete panoramic images, as will be illustrated in Section 5.

4 Implementation

As shown in Figure 2, our system uses a Khepera II miniature robot manufactured by K-Team Corporation of Switzerland. The Khepera II has the optional K2D camera turret, which utilises a camera pointed upwards, looking at a spherical mirror to acquire omni-directional images. 3-channel colour omni-directional images are captured using a Matrox Meteor II frame-grabber at full resolution of 510×492. Images are stored, processed, and displayed on our graphics workstation using the *Open Source Computer Vision Library (OpenCV)* [13]. The 3-channel omni-directional images were then converted to 1-channel grey-scale images using the `ConvertToGrey()` function in *OpenCV*, before being centred and cropped to a final resolution of 452×452. An example of a cropped omni-directional image is shown in Figure 1.

The image is un-warped using the forward-projection method, as described in Section 2, to create a space variant panoramic image. The space variant image data are used in Delauney triangulation to construct a triangular mesh, defined by a set $T = \{e_m\}$ of triangular elements, on which our feature detection operators are constructed. To create this mesh we have incorporated the '*Triangle*' meshing library [14] into our application. Information on nodal connectivity is stored by creating a list of connected nodes for each node in the image; the compactness of the list structure can be improved by judicious global ordering of the mesh nodes.

With each node (x_i, y_i) is associated an image value U_i and a piecewise linear basis function $\phi_i(x,y)$ spanning a neighbourhood Ω_i^σ comprised of the set S_i^σ of triangular elements that share (x_i, y_i) as a vertex. The approximate image representation may then be written as

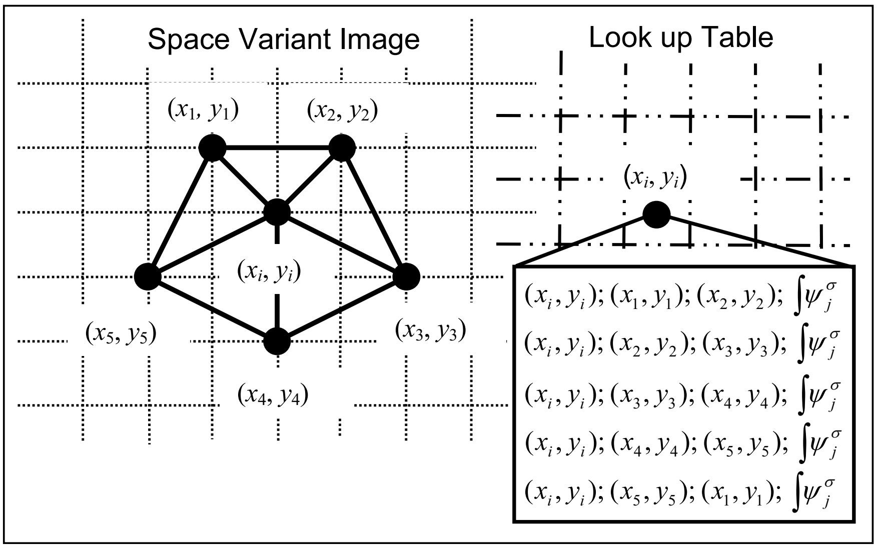
$$U(x,y) = \sum_{j=1}^{N} U_j \phi_j(x,y) \tag{4}$$

and is therefore piecewise linear on each triangle, and this image representation is used in equation (3) to develop feature detection operators that correspond to local derivative approximations.

In order to compute the weights in an operator we need to compute *element integrals* $k_{ij}^{m,\sigma}$, each defined over an element e_m in the neighbourhood set S_i^σ. For example, a contribution to a gradient operator $E_i^\sigma(U)$ would be $k_{ij}^{m,\sigma} = \int_{e_m} \frac{\partial \phi_j}{\partial x} \psi_i^\sigma dxdy$, whilst a contribution to a Laplacian operator $Z_i^\sigma(U)$ would be $k_{ij}^{m,\sigma} = \int_{e_m} \frac{\partial \phi_j}{\partial x} \frac{\partial \psi_i^\sigma}{\partial x} dxdy$. Such element integrals are accurately and efficiently evaluated using low-order (3-point) Gaussian quadrature. In finite element analysis, the $k_{ij}^{m,\sigma}$ values are stored in small *finite element matrices*: for a triangular element, each finite element matrix is 3×3 in size, with each row corresponding to a particular node in the triangle. A complete set of space variant autonomous operators, $E_i^\sigma(U)$ or $Z_i^\sigma(U)$, may thus be computed by *assembling* the $k_{ij}^{m,\sigma}$ values for each element e_m in S_i^σ according to information on nodal connectivity routinely stored in look-up-tables; i.e., each operator weight may be computed as $K_{ij}^\sigma = \sum_{\{m|e_m \in S_i^\sigma\}} k_{ij}^{m,\sigma}$.

The amount of computation required to carry out image un-warping by forward projection followed by construction of the triangular mesh and then the corresponding feature detection operators is greater than that required to construct a complete un-warped image by backward projection followed by application of traditional 4- or 8-pixel low level processing techniques. Hence, it might seem that real-time application is precluded. However, this is not the case, as the cartesian-to-polar transformation required for forward projection needs to be computed only once. Each pixel in an omni-directional image transforms to a fixed position in the un-warped panoramic image, and in all subsequent un-warped panoramic images. Hence for the robot to process a whole sequence of omni-directional images, we need to construct the triangular mesh and operators only once. By storing the operators in an efficient look-up-table, computation becomes sufficiently reduced to enable real-time application.

Whilst the operators could be constructed by computing and storing the finite element matrices described above, our performance experiments show that it can be more efficient to use a combination of stored information and some on-the-fly-computation; the best balance is achieved by minimizing the number of floating-point values that need to be addressed in look-up-tables. Since the image representation in equation (4) is piecewise linear on each element, the derivative terms, $\frac{\partial \phi_j}{\partial x}$ and $\frac{\partial \phi_j}{\partial y}$, in the element integrals $k_{ij}^{m,\sigma}$ are constant within an element, depending only on the locations of the three nodes of the element. Hence, rather than store and retrieve three fully computed element integrals $k_{ij}^{m,\sigma} = \int_{e_m} \frac{\partial \phi_j}{\partial x} \psi_i^\sigma dxdy$ or $k_{ij}^{m,\sigma} = \int_{e_m} \frac{\partial \phi_j}{\partial y} \psi_i^\sigma dxdy$ as floating point values on each element for each operator, it is more efficient to store and retrieve the single floating point value $\int_{e_m} \psi_i^\sigma dxdy$ for each element and multiply this value on-the-fly by the three constant values corresponding to $\frac{\partial \phi_j}{\partial x}$ or $\frac{\partial \phi_j}{\partial y}$. Figure 5 shows a graphical representation of how the operator data is stored in the look-up-table



$(x_i, y_i); (x_1, y_1); (x_2, y_2); \int \psi_j^\sigma$
$(x_i, y_i); (x_2, y_2); (x_3, y_3); \int \psi_j^\sigma$
$(x_i, y_i); (x_3, y_3); (x_4, y_4); \int \psi_j^\sigma$
$(x_i, y_i); (x_4, y_4); (x_5, y_5); \int \psi_j^\sigma$
$(x_i, y_i); (x_5, y_5); (x_1, y_1); \int \psi_j^\sigma$

Figure 5. Graphical representation of operator data storage in look-up-table

The procedure for the initial creation of the operators and subsequent processing of images is summarised in the following pseudo-code:

```
Capture first omni-directional image
Convert to grey-scale and crop
Un-warp to sparse panoramic view
Apply triangulation using ‘Triangle’
For i=1 to numNodes
/* for each node in the sparse image*/
     for j=1 to numElements
      /* for each element in a local neighbourhood */
      compute scale parameter $\sigma$ for element
      compute and store $\psi_j^\sigma$ integral in look-up-table
      store element node co-ordinates
      }
}

/* for all subsequent omni-directional images */
Convert to grey-scale and crop
Un-warp to sparse panoramic view
For i=1 to numNodes
/* for each node in the sparse image */
     for j=1 to numElements
      /* for each element in look-up-table */
      compute $X = \sum \int \frac{\partial \varphi_i}{\partial x} \psi_j^\sigma dxdy \times U_i$
      compute $Y = \sum \int \frac{\partial \varphi_i}{\partial y} \psi_j^\sigma dxdy \times U_i$
      operator response=$\sqrt{X^2 + Y^2}$
      }
}
```

In addition to the adaptive method described above, for purposes of comparison we have also implemented the back-projection technique described in Section 2, producing a complete image that may be used with standard feature detection techniques. We have applied the Sobel, Prewitt, and Canny edge detection operators to this full image for comparison with our results presented in Section 5.

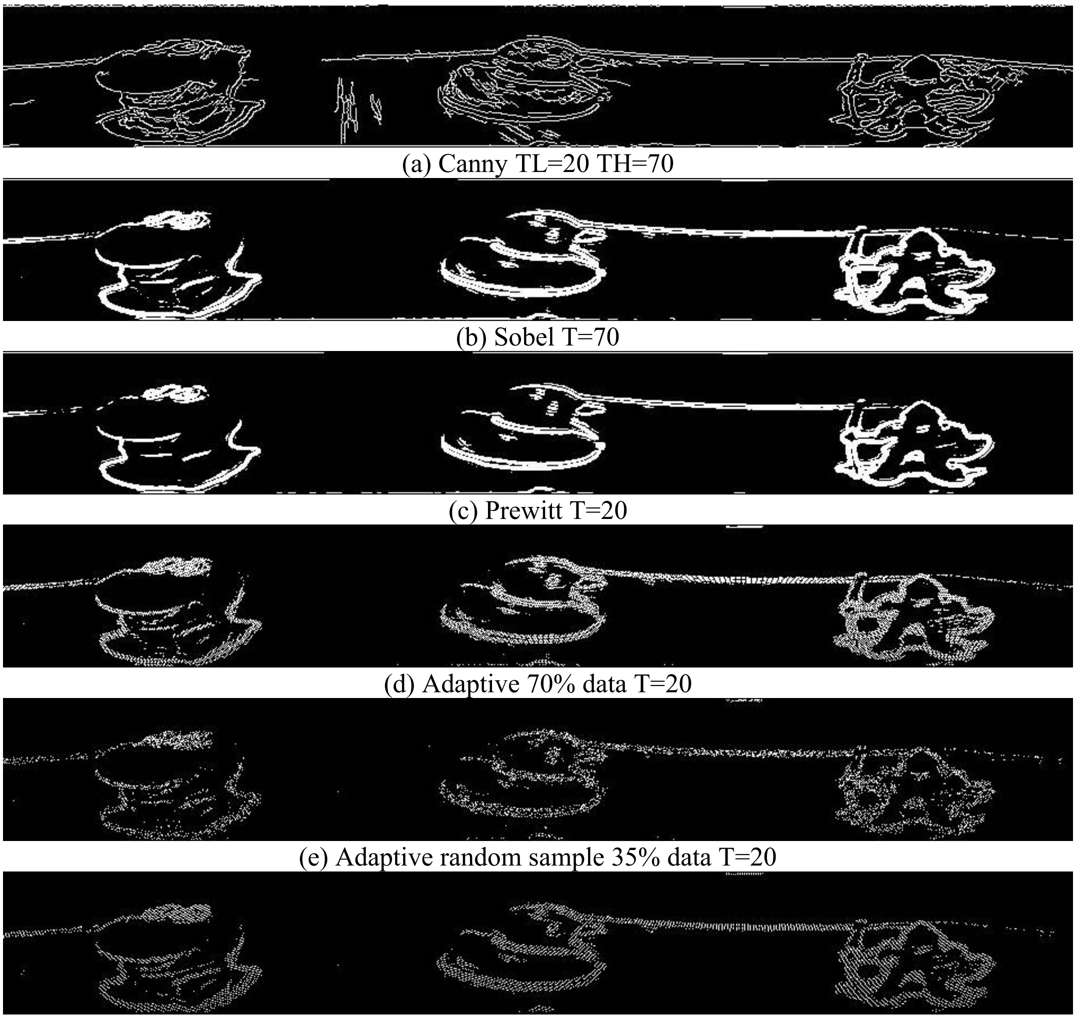
5 Experimental Results

We present processing times that consist of un-warping time and time taken to extract edges from the un-warped images. The space variant operator uses sparse images generated by the forward-projection un-warping technique described in Section 2. The Canny, Sobel, and Prewitt operators are used on complete un-warped images obtained by the back-projection technique also described in Section 2. Each algorithm has been run 100 times on a graphics workstation, consisting of a Pentium 4, 3.4 GHz processor and 2GB RAM, and the average processing time calculated. The space variant operator is comparable in processing time to both the Sobel and Prewitt operators, and is faster than the Canny operator. Comparative results for processing are presented in Table 1. These processing times do not include file I/O or capture times. With the space variant method it is also possible to reduce the quantity of data that are used for detecting features. In our experiments we have un-warped the omni-directional image to an extra-sparse panoramic image by using both random and regular sampling of the original omni-directional image data. This has the effect of reducing the quantity of data that needs to be processed, whilst still maintaining the ability to detect features using our space variant technique. Average processing times incorporating random and regular sampling have also been included in Table 1.

Operator	Processing Time (Seconds)
Space Variant (70% data)	0.069309
Space Variant (35% data) random sample	0.051391
Space Variant (35% data) regular sample	0.056221
Prewitt	0.068846
Sobel	0.068541
Canny	0.112076

Table 1. Processing times for detecting edges

In Figure 6 we present edge maps obtained using various feature detection techniques on the panoramic image shown in Figure 3. We have selected the visually best edge maps obtained with each feature detector at threshold T, and in the case of the Canny feature detector, threshold low TL, and threshold high TH. Figures 6(d), 6(e), and 6(f) illustrate that our approach yields feature maps containing sufficient information to recognise the objects in Figure 3. Our system captures images at a frequency of 25 Hz and using the space variant approach with a random sample we can forward un-warp and perform edge detection at a frequency of ~20 Hz, while reverse un-warping and using the Canny edge detection method a frequency of ~9 Hz is obtained. These results indicate that real-time performance approaching 20 Hz is achievable using the space variant approach, and also indicate that this approach would be suitable for use in a real-time robot localisation system as no significant delay between capture and processing is present.



(a) Canny TL=20 TH=70

(b) Sobel T=70

(c) Prewitt T=20

(d) Adaptive 70% data T=20

(e) Adaptive random sample 35% data T=20

(f) Adaptive regular sample 35% data T=20

Figure 6. Sample image sections and associated edge maps

6 Summary and Further Work

We have shown that through the use of appropriate data structures a family of autonomous finite element based image processing operators can process un-warped sparse omni-directional images in real-time, producing feature maps of a quality suitable for object recognition. Such an approach has the facility to reduce the quantity of input data further, thereby further reducing the processing times to become significantly faster than the traditional methods (approximately 30% reduction in processing time), whilst still producing edge maps suitable for object recognition. A slight computational time penalty is unavoidable with the initial construction of the mesh, however this is minimised as this mesh has to only be constructed on an initial image and not for subsequent images. The accuracy of the operator is also reliant on the accuracy of the underlying triangulation process. We have previously shown in [12] that the autonomous finite element based image processing operators can successfully process sparse range images. The approach of using autonomous finite element based image processing operators may also be used on other images with a non-uniform resolution such as log-polar images [15]. The next stage in our work is to develop a localisation system for the mobile robot using the feature maps of the robot's environment obtained by the current vision system.

7 Acknowledgements

This work was funded by the Nuffield Foundation under the research grant number NAL/00839/G.

8 References

[1] Gaspar, J., Winters, N., Santos-Victor, J., "Vision-based Navigation and Environmental Representation with an Omni-directional Camera" *IEEE T-RA, Vol. 16, No.6, pp.890-898, 2000.*

[2] Argyros, A.A., Tsakiris, D.P. & Groyer, C., "Biomimetic centering behavior", *IEEE Robotics & Automation Magazine., Vol. 11, No 4, pp. 21-30, 2004.*

[3] Vlassis, N., Motomura, Y., Hara, I., Asoh, H. & Matsui, T., "Edge-based features from omnidirectional images for robot localization", *Proc. IEEE ICRA, pp. 1579-1584, 2001.*

[4] Winters, N., Gaspar, J., Lacey, G., Santos-Victor, J., "Omni-directional Vision for Robot Navigation" *IEEE Workshop on Omnidirectional Vision, p.21, 2000.*

[5] Matsumoto, Y., Ikeda, Y., Inaba, M., Inoue, H. "Visual navigation using omnidirectional view sequence". *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp. 317--322, 1999.*

[6] Yagi, Y., Hamada, H., Benson, N., Yachida, M. "Generation of stationary environmental map under unknown robot motion". *Proc. IEEE/RSJ IROS, Vol. 2, pp. 1487-1492, Japan, 2000.*

[7] Torii, A. & Imiya, A. 2004, "Panoramic image transform of omnidirectional images using discrete geometry techniques", *3DPVT'04., pp. 608-615, 2004.*

[8] Nayar S.K., "Omnidirectional Vision" *Proc. 8th Int. Symp. Robotics Research, Japan 1997*

[9] Peri, V. & Nayar , S., "Generation of perspective and panoramic video from omnidirectional video*", In Proc. of 1997 DARPA Image Understanding Workshop, New Orleans., 1997.*

[10] Krose, B., Bunschoten, R., Hagen ST, Terwijn, B. & Vlassis, N., "Household robots look and learn: environment modeling and localization from an omnidirectional vision system", *IEEE Robotics & Automation Magazine, vol. 11, no. 4, pp. 45-52, 2004.*

[11] Coleman, S.A., Scotney, B.W. "Autonomous Operators for Direct use on Irregular Image Data" *Proceeding of the 2005 ICIAP, pp. 296-303, LNCS 3617, Springer Verlag.*

[12] Coleman S.A., Scotney B.W., Kerr D., "Direct Feature Extraction on Range Image Data", *Proc. of Irish Machine Vision and Image Processing Conference, QUB, Belfast, 2005.*

[13] Bradski, G.R., Pisarevsky, V. "Intel's Computer Vision Library: applications in calibration, stereo segmentation, tracking, gesture, face and object recognition", *IEEE CVPR, Vol. 2, pp. 796-797, 2000.*

[14] Shewchuk, J., "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator", *Applied Computational Geometry', Springer-Verlag, Berlin, Vol. 1148, pp. 203-222, 1996.*

[15] Wallace, R., Ong, P., Bederson, B., Schwartz, E., "Space variant image processing", *International Journal of Computer Vision, Vol. 13, No. 1, pp. 71-90, 1994.*

Object Recognition Using Artificial Neural Network: Case Studies for Noisy and Noiseless Images

Mahua Bhattacharya^{1,2}
1 Computer Science and Engineering
West Bengal University of Technology
BF -142, Sector I
Salt Lake City Kolkata 700064, India
bmahua@hotmail.com

Arpita Das^{2}
2 Institute of Radio Physics & Electronics
University of Calcutta
92, A.P.C. Road
Kolkata 700009, India
ar_das @rediffmail.com

Abstract

The artificial neural networks are now becoming popularly used for pattern recognition / classification problems and appropriate architecture and model have been developed for classification of patterns. In present paper authors have developed a Neural Network model for object recognition based on boundary features (Fourier descriptor) and regional features (Textural descriptors) of the objects. This neural network model has been further used for the classification of the two sets of images based on Fourier descriptors of the edge points of the images and on the basis of regional descriptors. Initially the network has been trained with noise free images for both the sets of images and then it is trained with noisy images. Finally the performance of the network architecture has been studied. The network has been successfully implemented on medical images of section of human brain having space occupying lesion and on few images of galaxy.

Keyword : Classification, Fourier Descriptors, neural network, performance

1. Introduction

Recognition is the process that assigns a label to an object based on its descriptors. The ultimate objective of digital image processing is to develop methods for correct recognition and classification of individual objects. A pattern is an arrangement of descriptors. A pattern class is a family of patterns that share some properties. Pattern recognition by machine involves the techniques for assigning patterns to their respective classes- as it is done normally in human brain. Machine perception of patterns [1],[2],[3],[4],[9],[10] can be viewed as a two fold task, (i) consisting of learning the invariant and common properties of a set of samples (ii) characterizing a class and deciding that a new sample is a possible member of class. Therefore, the task of pattern recognition by computer can be described as a transformation from the measurement space M to the feature space F and finally to the decision space D [4]. The approaches of pattern recognition are divided into two principal areas: decision-theoretic and structural. The first category deals with patterns described using quantitative descriptor, such as length, area, and texture. The second category deals with patterns best described by qualitative descriptors. The patterns used to estimate the adjustable parameters of the descriptor are usually called training patterns, and a set of such patterns form each class referred to a training set. The process by which a training set is used to obtain decision functions is called learning or training. Learning of descriptor is the central theme of the recognition process. Neural Network organized as nonlinear computing elements (called neurons), works as a way in which neurons are interconnected and performed in the brain. A multilayer Feedforword Neural Net is able to classify the objects / patterns correctly with the help of linear decision functions. The Network is trained with the basic method, called *generalized delta rule for learning by backpropagation* and then used in multiclass pattern recognition problems. Neural networks [4],[7],[11],[12],[13],[14],[15], recognize the patterns and adapt themselves with changing environments. The resulting models are referred to various names, including *neural networks, layered self-adaptive network models, parallel distributed processing* etc. The McCulloch-Pitts model of a neuron gives a simplified mathematical definition of an artificial neural system in analogy with biological nervous system. Every neuron model consists of a processing element with synaptic input connections and a single output. The signal flow of neuron inputs, is considered to be unidirectional

Neural Networks – A Soft-Computing Approach : In ANN model[5],[6],[7],[8] the perceptrons, when trained with linearly separable training sets would converge to a solution in a finite number of iterative steps, The solution takes the form of coefficients of hyperplanes capable of correctly separating the classes represented by patterns of the training set. More recent results dealing with the development of new training algorithms of multiplayer perceptrons [8] would supply an extended power of perceptron models. A single layer perceptron model, in principle is used to map the input data onto the required final output. It learns a linear decision function that correctly classifies only two linearly separable training sets. The response of this basic system is based on weighted sum of its inputs; that is

$$d(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + w_{N+1} \qquad \ldots\ldots\ldots(1)$$

This is a linear decision function. Having weights w_i, i =1, 2,…., n, . The perceptron architecture is trained with linearly separable training sets by a hyperplane (in 2-D it is a straight line only). All points on the one side of the hyperplane will be given output +1, and others contribute the output as −1. Thus the hyperplane form a decision surface separating one category from other. When the regions are associated with particular classification having more complex shape, several hyperplanes are formed. Under this condition multilayer perceptron modeling offers much more powerful result than single layer architecture. It is very convenient to refer the intermediate layers of a multilayer structure, such as layer A, layer B as the hidden layer. It is also noted that, each neuron has the same form as the single-layer with the exception that the hard limiting activation function has been replaced by a soft-limiting "sigmoid" function. Multilayer Feedforword network architectures consist of several layers of perceptron computing elements. It focuses on decision functions of multiclass pattern recognition problems and independent of whether or not the classes are linearly separable. The architecture consists of layers of structurally identical computing nodes (neurons) arranged so that the output of every neuron in one layer feeds into the input of every neuron in the next layer. The number of neurons in the first layer, called layer A, is N_A. The number of neurons in output layer, called layer Q, is denoted N_Q. The number N_Q equals W, the number of pattern classes that the neural network has been trained to recognize. The network recognizes a pattern vector x as belonging to class w_i , if the ith output of the network is "high" while all other outputs are "low".

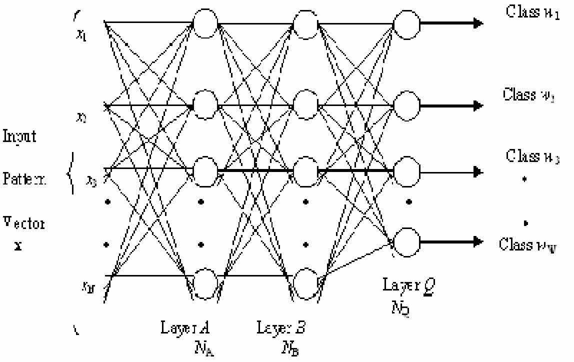


Fig.1 Multilayer Feedforword Neural Network.

In present paper authors have designed ANN model for the classification of the two sets of images using both Fourier descriptor and regional descriptors. Initially the network was trained with noise free images for both the sets and then with noisy images. Finally the performance of the network architecture has been studied at different noise levels. The network has been implemented on medical images of section of human brain having space occupying lesions and on few galaxy images for the purpose of classification. The novelty of the work is to propose an approach for determination of the nature of brain lesion either benign or malignant and to find finally the gradation of benignancy or malignancy. The benign lesions consist of smooth and circumscribed boundaries of almost round or oval shaped margins and malignant lesions consist of margins of multiple protrusions. Fourier descriptors can easily detect the margin of the tested lesions. Texture of the benign lesions are generally smooth whereas the malignant lesions are having coarse or irregular surface. Regional descriptors detect the relative smoothness of the lesions. The proposed technique stated in this paper is capable to identify the lesions in benign/malignant category and which may show new directions in clinical diagnosis. In case of galaxy images the position of stars changing from one galaxy to another galaxy can be detected using Fourier descriptors and using regional descriptors by detecting the change in intensity and smoothness of the galaxy images.

2. Overview of the work : Different Steps for Pattern Recognition

A brief overview of the different steps in object recognition with the help of Multilayer-Feedforword Neural Network is given below

***(i) Image Acquisition**:* The required images of section of human brain having space occupying lesion using CT and MR modalities have been converted in digital format. Set I galaxy images and Set II Brain lesions

(ii) ***Feature Selection:*** Feature selection is the choice of descriptors in a particular application. An image can be described depending on two choices, (a) to describe the image in terms of its external characteristics, such as its boundary and edge points, (b) to describe the image in terms of its internal or regional properties such as texture, color. In present problem both of the features of the pattern have been considered.

Digital Gray level Image
Fig-2(a)

Edges of the image
Fig-2 (b)

(iii)Boundary Description: Boundary or edge points of the images can be represented by the Fourier descriptors. Fig-2(b) shows k-points digital boundary in the x-y plane. The boundary starts from an arbitrary point (x_0,y_0) to the coordinate pairs (x_1,y_1), (x_2,y_2),…..(x_{k-1},y_{k-1}) in traversing the boundary. These co-ordinates are represented by the form $x(k) = x_k$ and $y(k) = y_k$. Thus the boundary can be represented as $s(k) = [x(k), y(k)]$ for $k = 0, 1, 2,\ldots, k-1$. Each co-ordinate pair can be treated as a complex number so that, $s(k) = x(k) + j*y(k)$, for $k=0, 1, 2, \ldots, k-1$. The x-axis is treated as real axis and y-axis as the imaginary one. The Discrete Fourier Transform (DFT) of s (k) is given below

$$a(u) = (1/K) \times \sum_{k=0}^{K-1} s(k) \times e^{-(j2\pi u k/K)} \ldots\ldots(1)$$

for $u = 0, 1, 2, \ldots, K-1$.The complex coefficient a (u) are called the Fourier descriptor of the edge Points. The inverse Fourier Transform of those coefficients are given as

$$s(k) = \sum_{u=0}^{K-1} a(u) \times e^{(j2\pi u k/K)} \ldots\ldots(2)$$

for $k = 0, 1, 2, \ldots, K-1$.

Let us suppose that instead of all Fourier coefficients, only the first 'P' coefficients are used and which is equivalent to set $a(u) = 0$ for $u > (P-1)$. The result of the following approximation is that the high frequency components containing the finer details are lost. Only the overall global shape of the images is identified.

In case of boundary detection we propose the Fourier descriptors as the key features. Firstly the edge points are found out. One – third of the maximum edge points have been searched which has given as P.

(iv)Regional Description: Regional descriptors are the internal property of the image. An important regional descriptor is the texture content. There is no formal description of texture, but it measures the properties such as smoothness, coarseness and regularity of the images.

(v)Statistical Approaches: One of the simplest approaches for describing texture is to use statistical moments of the gray level histogram of an image. Let z be a random variable denoting the gray levels which is range of $(0, L-1)$ and let $p(z_I)$, $I = 0, 1, 2, \ldots, L-1$, be the corresponding histogram (where L → number of distinct gray levels).

The n-th moment of z about the mean is given as

$$\mu_n(z) = \sum_{I=0}^{L-1} (z_I - m)^n p(z_I) \ldots\ldots(3)$$

where m is the mean value of z,

$$m = \sum_{I=0}^{L-1} z_I\, p(z_I) \ldots\ldots(4)$$

The second moment [the variance $\sigma^2(z) = \mu_2(z)$] is of particular importance in texture description. It is a measure of gray level contrast that can be used to establish relative smoothness. Another important regional descriptor is measured as the power of the images. The total power of the image is defined below:

The DFT of an image $f(x,y)$ of size M × N is given by

$$F(u, v) = (1/MN) \times \sum_{x=0}^{(M-1)} \sum_{y=0}^{(N-1)} f(x, y) \times e^{-j2\pi(u x/M + v y/N)} \ldots\ldots(5)$$

For $v = 0, 1, 2, \ldots, (N-1)$

for $u = 0, 1, 2, \ldots, (M-1)$, for $v = 0, 1, 2, \ldots, (N-1)$

As the analysis of complex numbers, F(u, v) can be expressed in terms of polar coordinate, $F(u, v) = |F(u, v)| \times e^{-j\phi(u, v)}$

where $|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$

and $\phi(u, v) = \tan^{-1}[I(u, v)/R(u, v)]$

P(u, v) → called the power spectrum of the Image $f(x, y)$ is defied as

$$P(u, v) = | F(u, v) |^2 = R^2(u, v) + I^2(u, v) \ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Thus the total power P_T is defined as

$$P_T = \sum_{u=0}^{(M-1)} \sum_{v=0}^{(N-1)} P(u, v) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

To describe an image in terms of its regional characteristics, the mean and standard deviation have been selected as textural feature and total power has been selected as another regional feature.

(vi)Design of Neural Network for the classification of the test images : The experiment was performed with two sets test images as shown below. Each set is comprising of three classes and each class having 15 images. To train the network twenty images from each set were used.

SET-I Images : (Galaxy Images)

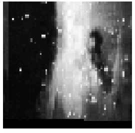
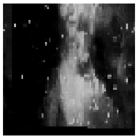
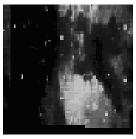




Fig-A class A	Fig-B class B	Fig-C class C
Mean 86.5718	Mean 50.67	Mean value =86.4
Std. dev.=76.72	Std. dev.=57.72	Std dev. = 52.87
Scaled Power	Scaled Power	Scaled Power
=224.46	= 98.97	= 83.00
edge points=172	Edge points=187	Edge points=171

SET-II Images : (Segmented human Brain Lesions)

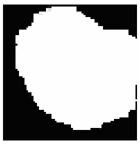


Fig-X class-X	Fig-Y class-Y	Fig-Z class-Z
Mean value=112.39	Mean value 150.6	Mean value 120.5
Std dev.=117.6	Std. dev 125.24	Std dev 90.8
Scaled Power 445.9	Sc. Power 647.1	Sc. Power 382.4
Edge points. 239	Edge points 163	Edge Points 161

Fig.(3) Test Images

A decision has been made on the basis of Fourier descriptor of the edge points, the proposed architecture of the Neural Network are :

60 **:** 30 **:** 3 for SET-I and 80 **:** 40 **:** 3 for SET-II

Number of neurons in the input layer = number of input pattern vectors = 60 & 80for two sets respectively. Number of units in the single hidden layer = 30 & 40 for two sets respectively; number of neurons in the output layer = number of pattern classes = 3 & 3 for two sets respectively. The network architecture has been chosen for achieving the optimum result i.e. optimum number of training cycles and optimum network complexity. Similarly to make a decision on the basis of regional descriptors of the images, the proposed architecture of the Neural Network

3 **:** 30 **:** 3 for SET-I and 3 **:** 20 **:** 3 for SET-II

Sigmoidal transfer function of the neurons throughout the experiment is given below

$f(\text{net}) = \{2/(1+e^{-(2 \times \text{net})})\} - 1$, where ‘net’ is the network function and learning rate of the Network = 0.01

3. Object Recognition Based on Neural Network Model

The Neural Network (60 **:** 30 **:** 3 Architecture) has been trained with Fig-A, Fig-B, Fig-C in SET-I and then Fig-A is correctly classified by the Network..

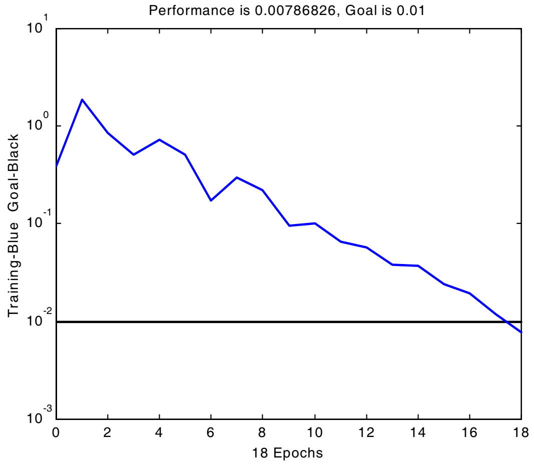
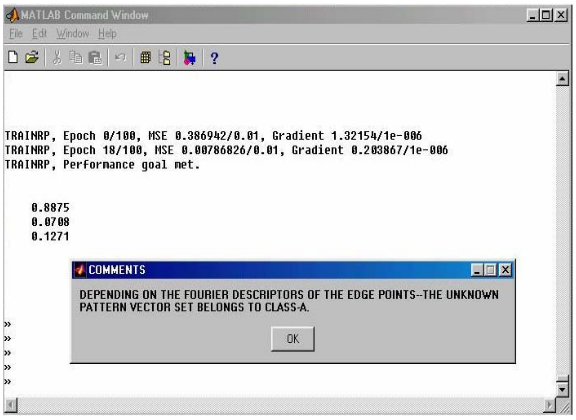


Fig-4**:** Training of the Network for the images of SET-I on the basis of Fourier Descriptor of the Edge points



```
TRAINRP, Epoch 0/100, MSE 0.386942/0.01, Gradient 1.32154/1e-006
TRAINRP, Epoch 18/100, MSE 0.00786826/0.01, Gradient 0.203867/1e-006
TRAINRP, Performance goal met.


    0.8875
    0.0708
    0.1271

»
»
»
»
»
```

COMMENTS

DEPENDING ON THE FOURIER DESCRIPTORS OF THE EDGE POINTS--THE UNKNOWN PATTERN VECTOR SET BELONGS TO CLASS-A.

Fig. 5 MATLAB Simulation Result with Fig-A of SET-I.

The Neural Network (3 **:** 30 **:** 3 Architecture) is trained with Fig-A, Fig-B, Fig-C in SET-I and then Fig-A is correctly classified by the Network.

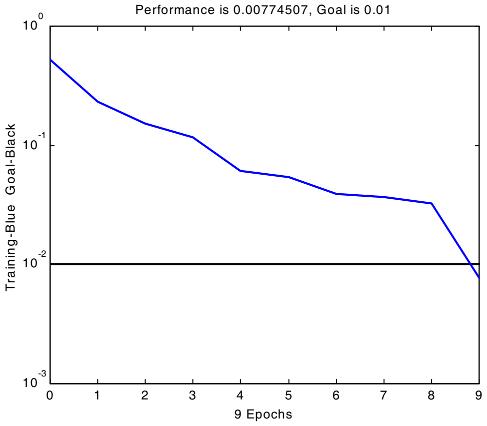
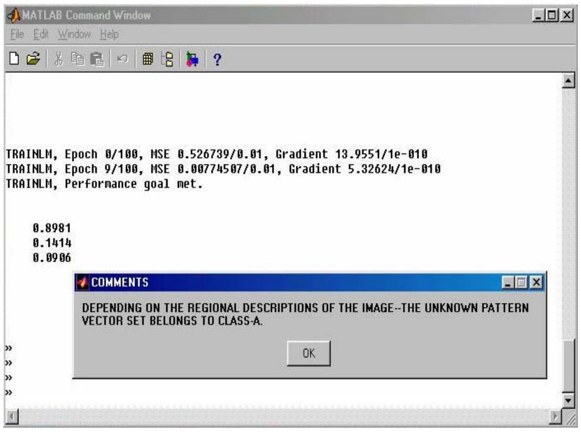


Fig-6 **:** Training of the Network for the images of SET-I (on the basis of regional descriptions of the Images)



```
TRAINLM, Epoch 0/100, MSE 0.526739/0.01, Gradient 13.9551/1e-010
TRAINLM, Epoch 9/100, MSE 0.00774507/0.01, Gradient 5.32624/1e-010
TRAINLM, Performance goal met.


    0.8981
    0.1414
    0.0906
```

COMMENTS

DEPENDING ON THE REGIONAL DESCRIPTIONS OF THE IMAGE--THE UNKNOWN PATTERN VECTOR SET BELONGS TO CLASS-A.

```
»
»
»
»
```

Fig–7 **:** MATLAB Simulation Result with Fig–A of SET-II.

Thus depending on both Fourier descriptor of the edges and regional descriptors of the Images of SET-I, Fig-A is correctly classified, similarly Fig-B & Fig-C are also.

4. Study for Network Intelligence

Noise-Free Case : Initially the network was trained with noise free images for both SET-I & SET-II. Let's consider Gaussian Noise with zero mean and variance of 0.01, 0.02, 0.03 and 0.04 are added with Fig-A of SET-I and Fig-Y of SET-II. It has been observed that correct classification is possible up to the noise level of variance 0.03 for both sets of images, and then misclassification was observed.

Table I shows the deviation of output neuron corresponds to class-A (belongs to SET-I) & class-Y (belongs to SET-II), increases from the Target value (here 0.9 for each output neuron), with increase of test noise level.

Table –I

Test Noise Level	0	.01	.02	.03	.04
Deviation from Target (class –A)	.0381	.0888	.1197	.152	.2107
Deviation from Target (class-Y)	.0154	.0215	.0897	.1605	.4462

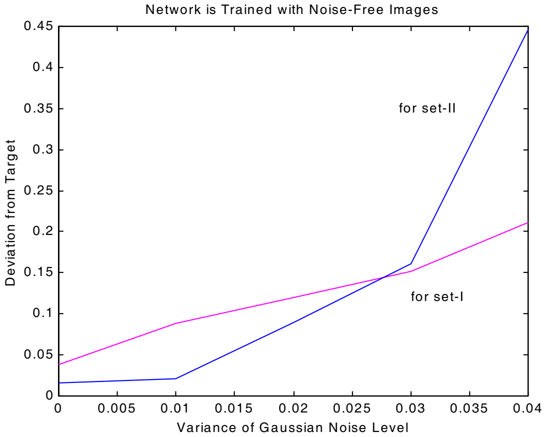


Fig-8 **:** Performance of the Neural Network as a function of test noise level.

Noisy Case: Network is now trained with the noisy images (Gaussian Noise with zero mean and variance of 0.02 is added to both sets of images).

In this case Table –II, shows that the deviation of the output neuron corresponds to class-A of SET-I & class-Y' of SET-II increase much slowly with the increase of test noise level in compare to the Noise-Free Case.

Table II

Test Noise Level	.01	.04	.08	.10	.12	.15	.18	.2
Deviation from Target (class-A)	.008	.01	.038	.049	.088	.14	0.18	.24
Deviation from Target (class-Y)	.027	.029	.142	.13	.18	.21	0.28	.32

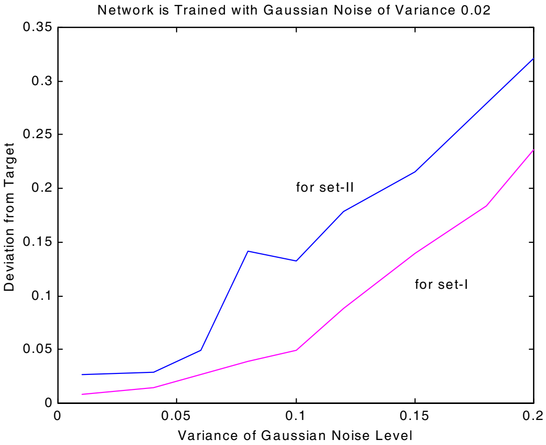


Fig-9 : Performance of the Neural Network as a function of test noise level.

5 Performance Analysis of Network Architecture

Fig.10 shows the learning profiles for the training of single hidden layer network. The number of training cycles required to achieve the performance goal largely vary with respect to the number of hidden layer units. Hidden layer sizes covering the range from 10 to 50 neurons in both the cases of SET-I & SET-II images have been tested to find Optimum Architecture for these specific problems. The network with less than 10 hidden neurons is not capable to learn all patterns, as a result, comparably large number of training cycles are required. On the other hand, hidden layer with more than 50 neurons have showed difficulty for learning, because the weight space has been too complicated.

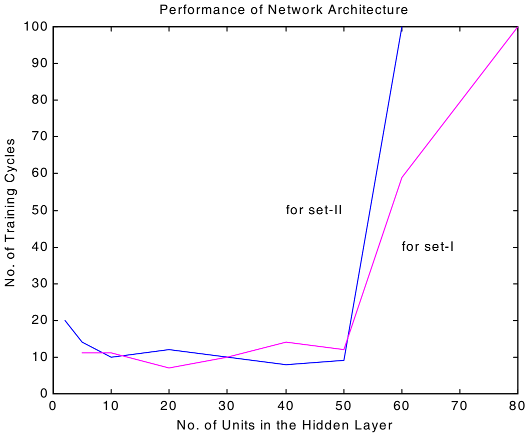


Fig-10 : Learning Profiles for the Training of SET-I & SET-II images with Single Hidden Layer Network.

Fig.11 shows the learning profiles for the training with four, three, two and single hidden layer network respectively for both SET-I & SET-II images. As expected, the number of training cycles required to achieve the performance goal, increases with increase of hidden layer numbers. Since a three-layer network architecture is capable for generating any arbitrary complex decision surfaces, this architecture is able to classify the unknown input patterns correctly. But it also be noted that network with two hidden layers shows minimum deviated output from the corresponding Target value.

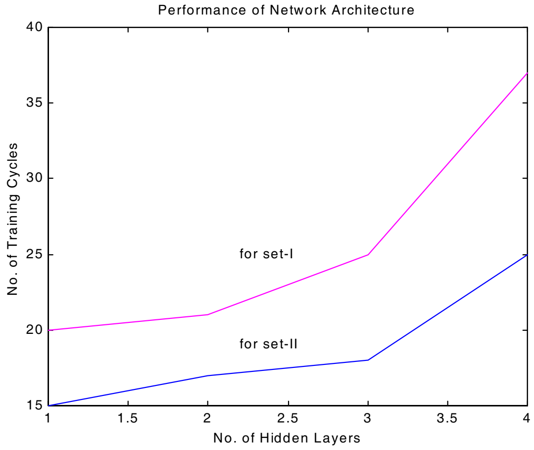


Fig-11 : Learning Profiles for the Training of SET-I & SET-II images with 4, 3, 2, 1 Hidden Layer Network.

Neural Network systems with modifiable weights have reached in the market recently. Furthermore, it seems certain that the proliferation of application-specific VLSI Neural Networks is of crucial importance for long-term success in technology. In long-term, it must be expected that Artificial Neural Systems will be used in applications involving speech detection, neuro-computing, decision-making, optimization of a function, quality control systems and Robot Kinematics.

References

[1] L.A. Zadeh (1994), Fuzzy logic, Neural Net and Soft Computing, *Communications of the ACM* 37: 77 - 84, .

[2] S.K. Pal and S. Mitra (1999), *Neuro – fuzzy pattern recognition : Methods in Soft Computing,* John wiely New York .

[3] R.O. Duda and P.E. Hart (1973), *Pattern Classification and Scene Analysis* , John Wiely New York

[4] S.K. Pal (2002), Soft computing Pattern Recognition, Case Generation and Data Mining , *Fuzzy Set Theory , Its Mathematical Aspects and Applications*, 51- 60 Allied Publishers.

[5] Hertz J, Krogh A and Palmer R[1991], *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Company.

[6] Sankar K. Pal, Sushmita Mitra, Pabitra Mitra(2003), Rough –Fuzzy MLP : Modular Evolution, Rule Generation , and Evaluation, *IEEE Trans. On Knowledge and Data Engineering ,* vol 15, no.1 ,Jan /Feb , pp : 14 -25.

[7] Y.H Pao(1989), *Adaptive Pattern Recognition and Neural Networks*, Reading, MA: Addison –Wesley.

[8] S.K. Pal and S. Mitra (1992), Multilayer Perceptron, Fuzzy Sets and Classification, *IEEE Transaction Neural Networks* , vol 3, pp: 683 -697.

[9] J.C. Bezdek (1981), *Pattern Recognition with Fuzzy Objective Functions*, Plenumm Press NY , 1981

[10] N.R. Pal and K Chintalapuri(1997) A Connectionist System for Feature Selection, Neural, *Parallel and Scientific Computation*, vol.5, no.3 pp : 359 -381, 1997.

[11]. B.Zheng, W.Qian and L. Clarke(1994), Multistage Neural Network for Pattern Recognition in Mammogram Screening, *IEEE International Conference on Neural Networks (*ICNN), pp.3437-3447.

[12]. Adlassnig, K. P. Fuzzy(1982) Nneural Network Learning Model for Image recognition. *Integrated mputer-Aided Engineering*, pp. 43-55,

[13]. Kim, J.S. and H. S. Cho (1994) A fuzzy logic and neural network approach to boundary detection for noisy images. *Fuzzy Sets and Systems*, pp. 141-159.

[14] B.Windrow and R.Winter(1988), Neural nets for adaptive filtering and adaptive pattern recognition. *IEEE Transactions on Computer*, pp. 25-39, March.

[15]C.-T. Lin and C.S.G. Lee(1991), Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on Computers*, vol-40, no-12, pp.1320-1336, December.

Facial Expression Classification using Kernel Principal Component Analysis and Support Vector Machines

J. Ghent
Computer Vision and Imaging Laboratory
Department of Computer Science
NUI Maynooth
jghent@cs.nuim.ie

J. Reilly
Computer Vision and Imaging Laboratory
Department of Computer Science
NUI Maynooth
jreilly@cs.nuim.ie

J. McDonald
Computer Vision and Imaging Laboratory
Department of Computer Science
NUI Maynooth
johnmcd@cs.nuim.ie

Abstract

This paper details a novel procedure for accurately classifying lower facial expressions. A shape model is developed based on an anatomical analysis of facial expression called the *Facial Action Coding System* (FACS). This model analyzes the movement in shape due to the formation of a specific expression. We apply *Kernel Principal Component Analysis* (KPCA) to the shapes in the training set and classify new unseen expressions by using *Support Vector Machines* (SVMs). We further analyse our model by attaching a probability measure to the outputs.

Keywords: Facial expression classification, *Kernel Principal Component Analysis* (KPCA), *Support Vector Machines* (SVMs).

1 Introduction

Of all the human senses, vision is the most informative with the majority of activity in our brain being concerned with visual processing. One of the most interesting and difficult visual processing tasks is facial image analysis. Of major importance here is the classification of facial expressions.

This paper details a technique that enables a computer to classify specific changes to the shape of a mouth. The approach taken employs psychological tools, computer vision techniques, and machine learning algorithms. We construct a training set such that every image in the training set depicts desired expressions. These expressions are anatomically analysed using a system for measuring expression called the *Facial Action Coding System* (FACS) [Ekman et al., 1978]. The expression we classify in this paper are all lower facial expressions and therefore only the shape of the mouth is analysed. A model is developed based on the shape of each mouth in the training set using KPCA.

We use the outputs of the shape model to train an SVM to classify an observed expression. SVMs are a new generation of learning system based on recent advances in statistical learning theory [Campbell, 2002, ScholKopf and Smola, 2002, Rogers et al., 2004, Rogers, 2004]. SVMs deliver state-of-the-art performance in real-world applications such as text categorisation, hand-written character recognition, image classification and bioinformatics [Guyon, 2006]. SVMs are based on a combination of techniques. A principal idea behind SVMs is the *kernel trick*, where data is transformed into a high-dimensional space making linear discriminant functions practical. SVMs also use the idea of *large margin classifiers*. This ensures the hyperplane is positioned in an optimal location seperating the two classes.

Considerable literature on facial expression classification exists. Techniques range from template based methods [Lyons et al., 1999], to neural network based methods [Er et al., 2002], or a combination of the two [Pantic and Rothkrantz, 2000, Ghent and McDonald, 2005b]. However, perhaps the most substantial work in this area has been done by Bartlett *et al*. Bartlett proposes a technique which combines Gabor wavelets and SVMs to classify *Action Units* (AUs) with 93.3% accuracy [Bartlett et al., 2004, Bartlett et al., 2003]. Again in [Littlewort et al., 2004], Littlewort and Bartlett propose a similar technique which classifies AUs with 97% accuracy. In [Littlewort-Ford et al., 2001], Bartlett used SVMs again to successfully distinguish between genuine and fake smiles.

More relevantly, Bartlett employed linear SVMs with PCA to classify facial actions with 75% accuracy [Littlewort et al., 2006] and concluded that there existed an incompatibility between PCA and SVMs for facial expression classification [Bartlett et al., 2005]. This approach performed PCA on Gabor wavelets of images from the training set prior to applying the SVM. We hypothesised that the non-linear nature of facial expression prohibited higher classification accuracy in [Bartlett et al., 2005] using this method. With this in mind we propose a technique that uses KPCA in conjunction with SVMs to classify facial expressions.

The rest of this paper is structured as follows: Section 2 documents our approach, section 3 details experiments and results, and we conclude with some final remarks. This paper extends our previous work detailed in [Ghent and McDonald, 2005b, Ghent and McDonald, 2005a] and [Ghent, 2005].

2 Proposed Methodology

Measuring facial expressions is a non-trivial task as everyone's face is unique. Several methods have been proposed, however, the technique we use must measure expression consistently independent of identity. In this paper we use the *Facial Action Coding System* (FACS), which measures expression by the movement of muscles in the face. This system is based on an anatomical analysis of facial expressions. A movement of a muscle or in some cases a group of muscles is known as an *Action Unit* (AU). All expressions can be described using the AUs defined by the FACS. The FACS allows us to subdivide our training data into subsets where the variation in each subset is precisely characterised. This provides the basis for accurate classification of expression independent of subject. We use FACS coded images to build a statistical model of shape based on point distribution.

2.1 KPCA

We label every image in the training set with a set of landmark points. These points are located around key areas such as the eyes, nose, mouth and eyebrows. The mean shape of the face is calculated and every image is aligned to the mean shape using *Generalised Procrustes Alignment* (GPA)[Gower, 1975]. This technique aligns two shapes with respect to translation, rotation and scale by minimising the weighted sum of the squared distances between the corresponding landmark points. The aligned landmark points are analysed using *Kernel Principal Component Analysis* (KPCA). This technique is similar to standard PCA except the data is projected into a higher dimensional feature space prior to performing eigenvector decomposition.

We project the data into feature space through the use of the *kernel trick*. This *kernel trick* permits the computation of dot products in high dimensional *feature spaces*, using functions defined on pairs of input patterns.

More specifically, mapping from one space to a higher dimensional space involves a mapping from $\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i)$, however, with an appropriate choice of kernel there exists a mapping ϕ such that

$$(\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)) = K(\mathbf{x}_i, \mathbf{x}_j). \tag{1}$$

This means that the inner products of the feature space can be calculated without computing $\phi(x)$ directly. This allows us to work in an extremely high dimensional feature space. The choice of kernel is still a matter of debate, however, in this paper we use a *Gaussian* kernel. The Gaussian kernel is defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)/2\sigma^2}. \tag{2}$$

The difference between KPCA and PCA is illustrated in Figure 1. It can be seen from this figure that the first principal component clusters the data using KPCA while standard PCA illustrates the most significant mode of variation.

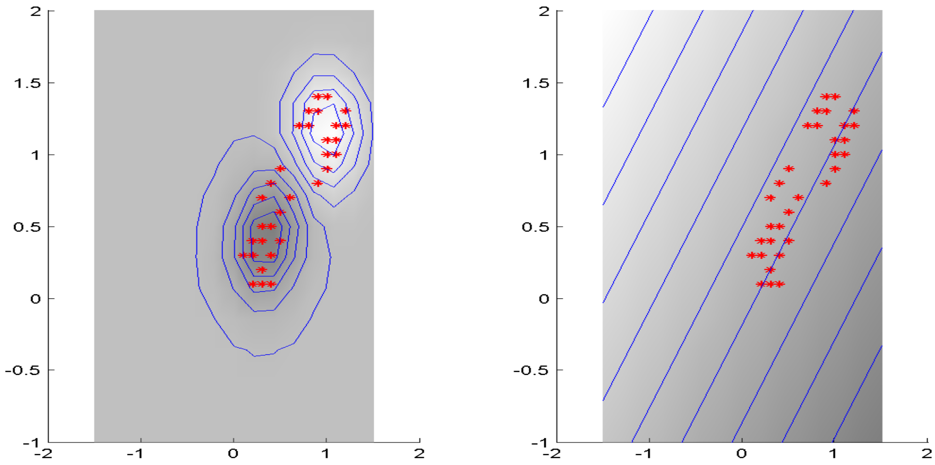


Figure 1: *An comparison using sample data between KCPA and PCA. KPCA is shown to the left of this figure and PCA is shown on the right.*

2.2 Support Vector Machines

The SVM algorithm can be separated into two distinct procedures, the *kernel method*, which we have already discussed, and the *base algorithm*. Suppose we have a dataset $(x_1, y_1), ..., (x_m, y_m) \in \mathbf{X} \times \{\pm 1\}$ where $\mathbf{X}$ is some space from which the x_i have been sampled. We can construct a dual Lagrangian of the form

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{3}$$

which are subject to the constraints

$$\alpha_i \geq 0 \quad \forall i \qquad and \qquad \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{4}$$

Further details of the construction of this equation can be found in [Ghent, 2005]. The solution to Equation 3 is a set of α values which are used in the decision function

$$f(\mathbf{z}) = sign\left(\sum_{i=1}^{m} y_i \alpha_i (\mathbf{x}_i \cdot \mathbf{z}) + b\right) \tag{5}$$

here $\mathbf{z}$ is an input and b is the bias. The resulting α_i values that are non-zero correspond to the support vectors. If $\alpha_i = 0$ then these points make no contribution to the decision function. The value of each α_i also carries information about the importance of particular datapoints in the training set. This insight can be utilised to deal with outliers and erroneous datapoints [Campbell, 2002]. Imposing a box constraint on the α's can limit the effect of outlying input data. The box constraint is given by

$$C \geq \alpha_i \geq 0 \tag{6}$$

where C is known as the soft margin parameter. The value of C is set using a standard optimisation approach, details can be found in [Ghent, 2005].

2.3 Measure of Confidence

It is possible to extract probabilities from SVM outputs which can be used as a post processing tool for classification problems. An SVM has a confidence measure which is inherent in the technique. The further a test point is from the separating hyperplane the greater the degree of confidence should be in the classification of that point. This distance can be mapped to a probability using a technique devised by Platt [Platt, 1999]. We use a parametric model to fit the posterior probability $P(y = 1|f)$ directly. The parametric function is

$$P(y = 1|f) = \frac{1}{1 + exp(Af + B)}. \tag{7}$$

A and B can be found from the training set by minimising the negative log likelihood function

$$min \left[-\sum_i t_i log(p_i) + (1 - t_i) log(1 - p_i) \right] \tag{8}$$

where p_i is (7) evaluated at f_i (the real value output of input i). This is minimised using a Levenberg-Marquardt algorithm. Once the sigmoid is found using the training set we can calculate the probability an unseen shape has of belonging to the class in question.

3 Experiments and Results

In this paper we classify AU's associated with the mouth. We classify four expressions; AU20+AU25, AU12, AU10+20+25 and AU25+AU27. The effect each of these AUs have on the mouth is illustrated in Table 1.

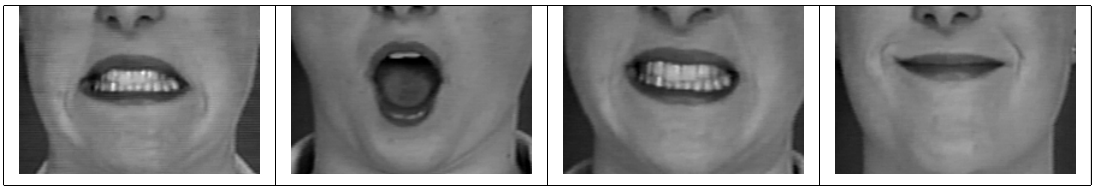


Table 1: *This table illustrates the effect of portraying four different expressions. The AUs portrayed, from left to right are; AU20+25, AU25+27, AU10+20+25, and AU12,*

We calculate our shape space by performing KPCA on the training data, as outlined in Section 2.1. The training data consists of just one subject performing the four desired AUs we wish to classify. We project new unseen data to be classified into the shape space and use these outputs as inputs to the SVM classifier. As there exists four expressions to be separated, the one-against all approach yields four separate SVM classifiers. This approach requires, at most, four evaluations to acquire a result. The results from the one-against-all approach are detailed in Table 2.

In Table 2 NTs is the number of test shapes, C is the soft margin variable, σ is the kernel parameter, and Ts is the percentage of correctly classified test data.

AU	NTs	C	σ	Ts
A-v-all	116	0.2	0.1	82.756
B-v-all	116	0.9	0.5	91.3794
C-v-all	147	0.1	0.2	93.8776
D-v-all	147	0.1	0.2	93.1972
$Average$				90.01

Table 2: *This table details the results from a one-against-all approach to classifying four multiple AU expressions. In the table above A = AU20+AU25, B = AU25+AU27, C = AU10+AU20+AU25 and D = AU12.*

It can be seen from Table 2 that a one-against-all SVM classifies four primary facial expression with an average accuracy of 90.01%. This is an encouraging result for three main reasons. Firstly, there exists a large amount of variance in the training set which would complicate the separation task. Secondly, the test data is completely unseen from the expression space i.e the test data was not used in calculating the expression space. This means that there exists enough variance in the expression space to accurately describe unseen shapes of individuals. And thirdly, there is significant overlap between the expressions we wish to classify, for example, it can be seen from Table 1 that two of the expression are extremely similar, this makes the separation task significantly more difficult.

Unfortunately, there exists no human classification baseline data of these AU's to compare our systems performance with. However, Bartlett has shown that naive human subjects classify single AU's with an accuracy of 77.9% while expert FACS coders classify AU's with an accuracy of 94.1% [Bartlett et al., 2003]. Naive subjects were provided with a guide sheet of the AU's which depicted examples of each AU and were also provided with written descriptions of each AU. Furthermore, alternative techniques for classifying multiple AUs achieve results of 83.34% [Abboud et al., 2004] and 86.0% [Michel and Kaliouby, 2003].

We extend our approach by incorporating a confidence measure associated with each new unseen shape. This information can be used to provide a measure of how confident we are that an unseen input belongs to a particular class. For example, in Figure 2 we input shapes into a probability function designed to recognise AU20+AU25. Each subject's expressions range from neutral to AU20+AU25. At neutral, represented by 1 on the *x-axis*, there is a low probability of the shape belonging to class AU20+AU25, however, once the expression is formed the likelihood of that shape belonging to class AU20+AU25 increases significantly. This probability measure makes no inference as to the intensity of the expression in question.

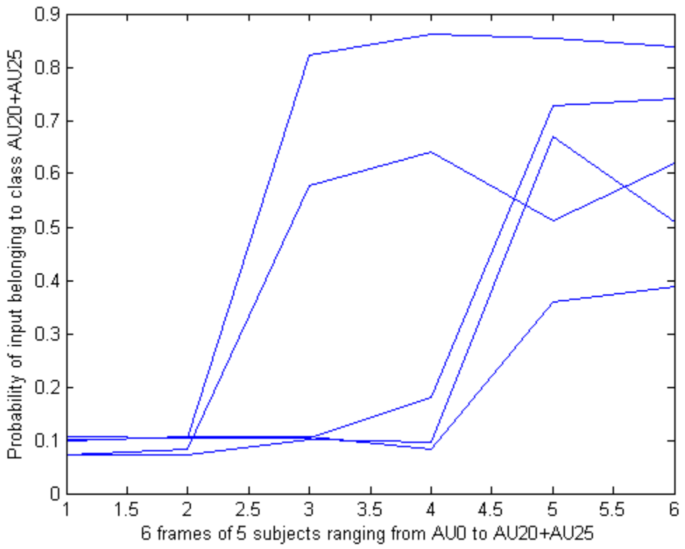


Figure 2: *The probability of a sequence of shapes belonging to one class*

This property of our approach is emphasized in Figure 3, here, 18 shapes are passed into four probability functions, each designed to measure the likelihood of an input belonging to a particular class. The input to this experiment was 18 shapes ranging from low intensity AU12 to high intensity AU12. As the diagram shows, the probability of the inputs belonging to class AU12 is greater than the probability of the same inputs belonging to any other class. It should also be noted that there is no significant difference between the likelihood of a low intensity example of AU12 and the likelihood of a high intensity example of AU12. The reason for this is that a high intensity expression is not necessarily at a greater distance from the separating hyperplane in an SVM.

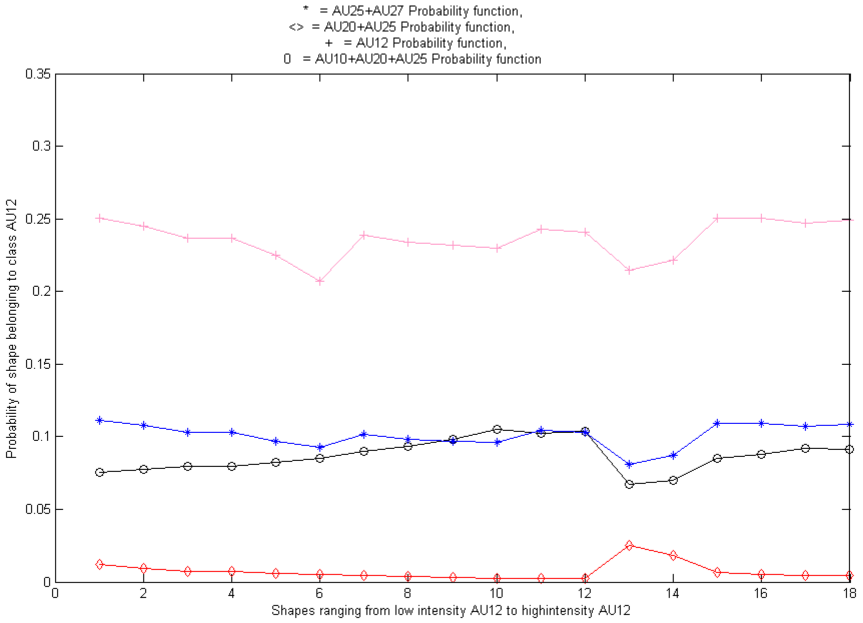


Figure 3: *The probability 18 inputs ranging from low intensity AU12 to high intensity AU12 belonging to a specific class.*

As can be seen from Figure 4, the confidence measure can also be used to aid in the classification process. This figure shows the probability of an input belonging to class AU20+AU25. The input data in this experiment is a sequence of extreme examples of AU20+AU25 as shown by several subjects. It should also be noted some of the inputs are classified as belonging to class AU12. This attribute again suggests that the most extreme expression is not necessarily going to return the highest probability of an input belonging to a particular class.

4 Conclusion

The accurate classification of facial expressions is a growing problem within several domains. The solution described in this paper takes a multidisciplinary approach drawing together psychological tools, statistical models and machine learning techniques. We first built a shape model that was based on an anatomical analysis of facial expression (FACS). The FACS provided us with a universal method of analyzing facial expression and allowed for the classification of facial expressions independent of subject (age, sex, skin, colour, etc.).

The shape model was calculated by using KPCA to lower the dimensionality of the problem. A one-against-all SVM was used to classify four expressions (AU20+AU25, AU25+AU27, AU10+AU20+AU25 and AU12). A one-against-all SVM classified multiple AU's with an average of 90% accuracy. A Gaussian kernel was used in each SVM and the value of the Gaussian (σ) and the soft margin parameter (C) were calculated using cross validation. Finally the data was further analysed by extracting probabilities from the outputs of the SVM's and establishing a confidence measure.

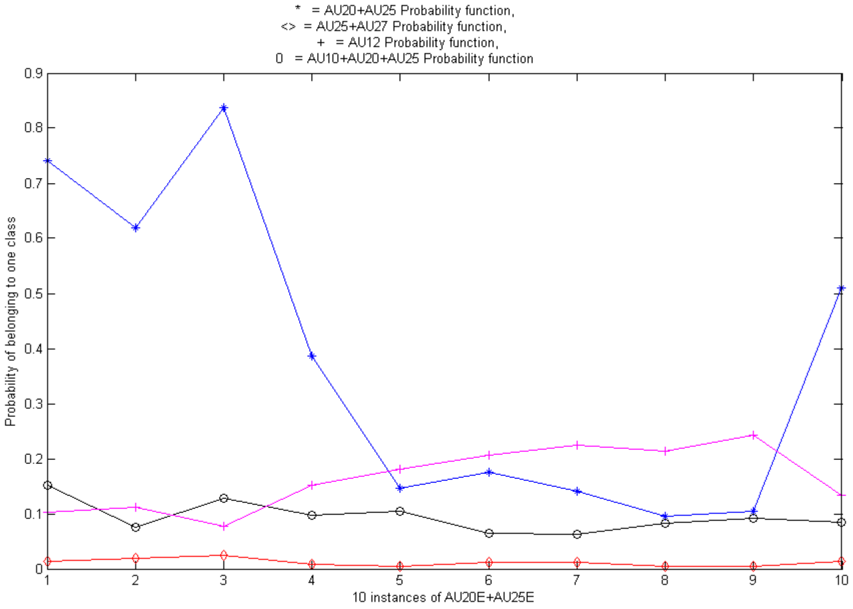


Figure 4: *The probability of 10 inputs belonging to a specific class. The inputs in this experiment all represent shapes portraying extreme examples of AU20+AU25.*

4.1 Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland.

References

[Abboud et al., 2004] Abboud, B., Davoine, F., and Dang, M. (2004). Facial expression recognition and synthesis based on an appearance model. *Signal Processing: Image Communication, ELSEVIER*.

[Bartlett et al., 2005] Bartlett, M. S., Littlewort, G., Frank, M., lainscsek, C., Fasel, I., and Movellan, J. (2005). Recognising facial expression: Machine learning and application to spontaneous behaviour. *IEEE International conference on computer vision and pattern recognition*.

[Bartlett et al., 2004] Bartlett, M. S., Littlewort, G., Lainscsek, C., Fasel, I., and Movellan, J. (2004). Machine learning methods for fully automatic recognition of facial expressions and facial actions. *IEEE International conference on systems, man and cybernetics*, pages 592–597.

[Bartlett et al., 2003] Bartlett, M. S., Movellan, J., Littlewort, G., Braathen, B., Frank, M. G., and Sejnowski, T. J. (2003). Towards automatic recognition of spontaneous facial actions. *In Paul Ekman, editor, what the face reveals*. Oxford University Press.

[Campbell, 2002] Campbell, C. (2002). kernel methods: A survey of current techniques. *Neurocomputing*, 48:63–84.

[Ekman et al., 1978] Ekman, P., Friesen, W., and Hager, J. (1978). Facial action coding system. *Consulting Psychologists Press*.

[Er et al., 2002] Er, M. J., Wu, S., Lu, J., and Toh, H. L. (2002). Face recognition with radial basis function neural networks. *IEEE transactions on neural networks*, 13(3):697–710.

[Ghent, 2005] Ghent, J. (2005). *A Computational Model of Facial Expression*. PhD thesis, National University of Ireland Maynooth, Co. kildare, Ireland.

[Ghent and McDonald, 2005a] Ghent, J. and McDonald, J. (2005a). Facial expression classification using a one-against-all support vector machine. *Proceedings of the Irish machine vision and image processing conference*.

[Ghent and McDonald, 2005b] Ghent, J. and McDonald, J. (2005b). Holistic facial expression classification. *Opto-Ireland*.

[Gower, 1975] Gower, J. C. (1975). Generalised procrustes analysis. *Psychometrika*, 40:33–50.

[Guyon, 2006] Guyon, I. (2006). Svm application list. http://www.clopinet.com/isabelle/Projects/SVM/applist.html.

[Littlewort et al., 2004] Littlewort, G., Bartlett, M. S., andJ. Chenu, I. F., Kanda, T., Ishiguro, H., and Movellan, J. (2004). Towards social robots: automatic evaluation of human-robot interaction by face detection and expression classification. *Advances in Neural Information Processing Systems*, 16:1563–1570.

[Littlewort et al., 2006] Littlewort, G., Bartlett, M. S., Fasel, I., Susskind, J., and Movellan, J. (2006). Dynamics of facial expression extracted automatically from video. *Computer vision and Image understanding*.

[Littlewort-Ford et al., 2001] Littlewort-Ford, G., Bartlett, M. S., and Movellan, J. R. (2001). Are your eyes smiling? detecting genuine smiles with support vector machines and gabor wavelets. *Proceedings of the 8th annual joint symposium on neural computation*.

[Lyons et al., 1999] Lyons, M. J., Budyek, J., and Akamatsu, S. (1999). Automatic classification of single facial expressions. *IEEE transactions on pattern analysis and machine intelligence*, 21(12):1357–1362.

[Michel and Kaliouby, 2003] Michel, P. and Kaliouby, R. E. (2003). Real time facial expression recognition in video using support vector machines. *Proceedings of HCI International Conference*.

[Pantic and Rothkrantz, 2000] Pantic, M. and Rothkrantz, L. J. M. (2000). Automatic analysis of facial expressions: the sate of the art. *IEEE transactions on pattern analysis and machine learning*, 22(12).

[Platt, 1999] Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularised likelihood methods. *In Alexander J. Smola, Peter Bartlett, Scholkopf Bernhard, and Dale Schuurmans, editors, Advances in large margin Classifiers*.

[Rogers, 2004] Rogers, S. (2004). *Machine Learning Techniques for Microarray Analysis*. Faculty of engineering mathematics, University of Bristol.

[Rogers et al., 2004] Rogers, S., Williams, R. D., and Campbell, C. (2004). *BioInformatics with computational intelligence paradigms*, chapter Class prediction with Microarray Datasets. Springer-Verlag.

[ScholKopf and Smola, 2002] ScholKopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

VALIDATION AND PERFORMANCE EVALUATION OF AN AUTOMATED VISUAL FEATURE EXTRACTION ALGORITHM

Premit Patel and Karim Ouazzane,
London Metropolitan University,
166-200 Royal Holloway Road.
N7 8DB London, UK.
Permit.patel@londonmet.ac.uk

Abstract

Audio-Visual Speech Recognition improves speech recognition by taking advantage of the fact that 'Human speech production and perception are bimodal' thus integrating visual feature cues into the recognition. A novel algorithm for the robust and reliable automatic extraction of visual feature points for audio-visual speech recognition is described. The accuracy of the automatic feature extraction algorithm is evaluated by comparing its results with manual feature extraction findings. The performance of various modules of feature extraction process has also been evaluated. The results presented demonstrate a highly accurate automated feature extraction algorithm with an average error of about 1-2 pixels for the most important visual features such as mouth width and height.

Keywords: Lipreading, viseme extraction, visual feature validation

1 Introduction

The ideal relation between the audio and visual sensory information in human speech recognition can be demonstrated with audio-visual illusions such as the McGurk effect [1], where the listener perceives something else than what is said acoustically due to the influence of a conflicting visual cues. These observations provide a motivation for attempting to integrate vision with speech in a computer speech recognition system. The main objective is to combine the visual and acoustic speech information so that recognition performance follows the human characteristic that bimodal speech recognition system's efficiency is always higher than that from either audio or visual modality alone.

Petajan [2] is often credited for building the first audio-visual speech recognition system. Using a single talker and custom hardware to quantify mouth opening together with linear and dynamic time warping, he showed that an audio-visual system was better than either alone. Since then, numerous researchers have demonstrated the use of visual speech cues of the speakers face, primarily the lip movements, for automatic speech recognition. However, robust and accurate visual feature extraction is a difficult object recognition problem, due to high variation in pose, lighting and facial makeup. Most of the existing approaches use constraints such as the subjects lips marked with colour or a reflective marker, the lip movements recorded with a head mounted camera, hand segmentation of the lip region or use of very controlled lighting conditions and limited subjects.

Most of such lip reading systems involve segmenting the mouth area from an image sequence of a person articulating a word, extract relevant features, and use them to be able to classify the word from those visual features vectors. A solution is presented which is capable of locating, extracting and tracking visual features reliably from a variety of subjects without the use of artificial aids to enable its operation in real world application. This solution reliably locates and tracks selected visual features in a sequence of images for an uttered alphabet.

In this paper, the results of automated feature extraction algorithm are validated against the manually extracted features by the user. The performance of different modules of the visual feature extractor is also evaluated.

2 Previous Work

Since the study by McGurk and MacDonald [1], there have been many systems that combine audio and visual information for automatic speech recognition. The most crucial elements of such audio-visual speech recognition system are accurate audio and visual feature extraction and fusion of audio with visual features. Over the last two decades many techniques have been proposed in the visual feature extraction area. These techniques can be mainly categorized into two groups, namely, image based (pixel based) [3,4] and model based (lip contour based) [5,6]. In the image based method the mouth area pixels are used as input of the recognition engine (e.g. HMM or ANN). The system is trained with typical pixel patterns associated with particular lip movements. The problem with such image-based system is that the input vector needs high dimensional space and it contains high redundant data. This is resolved by employing a principal component analysis (PCA) or linear discriminant analysis (LDA), which reduces the dimensionality of the input vector and defines the main directions of variation. In the model-based methods, a model of the visible speech articulators (e.g. lip contour) is built which is described by set of parameters. A brief review of existing lip segmentation techniques is given here.

One of the most common methods for lip feature extraction is the use of the grey-scale domain and edge detection [8,9]. Other technique [10] by Lewis and Powers uses colour spectrum and focuses on the green and blue colour. The rational is that as the face and lips are predominantly red, such that any contrast that may develop would be found in the green or blue colour range, excluding red. The researchers extracted mouth region using the following formula.

$$Log\,(\,G\,/\,B\,) <= \beta \qquad (1)$$

Using the log scale further enhances the contrast between distinctive areas. Here, the threshold β is manually calculated and varied to identify mouth area and lip features [10].

Coianiz et al [12] have used the hue, saturation and intensity colour space to extract lip pixels. The reason HSI space is preferred is that it disentangles illumination from colour, such that variations in lighting should not cause great variation in hue. The possibility of a pixel being part of lip pixels is based on a redefined hue value, h0 that defines lip hue [10].

$$\begin{aligned} f(h) &= 1 - (h - h0)^2 / w^2 &&, \mid h - h0 \mid <= w, \\ &= 0 &&, \text{otherwise.} \end{aligned} \qquad (2)$$

Another feature extraction technique proposed by Yang and Waibel [11], uses red and green components for lip segmentation,

$$L < R/G < U \qquad (3)$$

Where R and G are red and green components of pixel colour respectively and L and U are lower and upper boundaries that define which values are considered as lip pixels values. Though the above techniques are tested on various speakers and it works quite well finding most lip pixels with some noises, there are few difficulties associated with the above techniques such as i) it doesn't detect lips when the person is wearing a lipstick ii) it is hard to locate lips of a person with brown screen or wearing reddish make up iii) it is hard to locate outer boundary of lower lip due to shadows and a gradual colour change at outer boundary.

3 Visual Feature Extraction Process

Speech is composed of individual speech sounds known as phonemes, and when spoken, some of the phonemes show specific facial movements including the lips, tongue, jaw and the visibility of teeth. For most people a particular facial expression is created every time these visually distinctive phonemes are uttered. Such visual features need to be used to represent the phonemes uttered. The most of the information related to facial movement is conveyed in a lip movement. Such a lip movement could be described by variations in visual features such as height, width, area and contour. Thus, any particular phoneme enunciated could be described by variations in such visual features. For training purposes, some visual features such as lips width, height, perimeter, and area need to be manually extracted from face image sequences by user. Other features such as tongue, teeth visibility, mouth opening degree and time taken to utter an alphabet are also significant. These visual features are selected to collectively represent the mouth shape, lip movement and other important related information for the alphabets uttered.

The main difficulty in integrating lip movements cues into an automated speech recognition system is to find a robust and accurate method for extracting important visual features. The technique should be able to locate and tract lips in faces of various speakers and should be robust to variance in lighting, face rotation and scale. Most of the visual speech recognition solutions choose to approximate the lips contours using splines, or active contours, and perform tracking frame by frame of the lips movements. The drawback of the solution is high computational cost, and it usually needs pre-defined starting control points. A new approach is presented in this paper that uses a different and much smaller set of features for the lips that are easier to extract, and more stable. The following section describes manual feature extraction process followed by automated visual feature extraction algorithms.

3.1 Manual Feature Extraction

The visual features discussed so far are static parameters, that is, they illustrate the situation at a particular point in time but not the dynamic patterns that cause the change from one situation to the next. For lip movement modeling, it is crucial that features are extracted not just for a particular time during a speech but through out the duration of phoneme utterance. It is observed that obtaining visual features such as width, height or area at a particular time is not sufficient rather, the variations in feature values over time need to be extracted that can illustrate the lip movement during the speech generation. Such variations occur due to different vocal tract configurations based on different articulator positions and speaker characteristics.

For lip reading, the shape variation is of most interest in the analysis of phoneme-viseme relationships, because it enables studying similarities and differences between phonemes. Here, feature vectors are created from the extracted lip features to analyse lip movement for the alphabet spoken. Depending on average values of a particular feature, different thresholds are obtained for the related feature. Symbols N, W and M are used to describe narrow, wide and medium values of any feature respectively.

$$v < lt \Rightarrow \text{N}$$
$$v > ut \Rightarrow \text{W}$$
$$lt < v < ut \Rightarrow \text{M}$$

Here, v denotes feature value such as lip area, lt denotes lower threshold and ut denotes upper threshold. The increase, decrease or no change in feature value from previous frame is denoted by signs '+', '-' or '=' respectively.

3.2 Automated Feature Extraction

Automatic visual feature extraction involves extracting lip contour points, tongue visibility and teeth visibility. The lip contour points are used to find lip gap width and lip gap height for a speaker. The main steps employed to precisely extract these visual features are eyes detection, nostrils detection and accurately mouth area estimation. A flow chart for feature extraction process is given in figure 1.

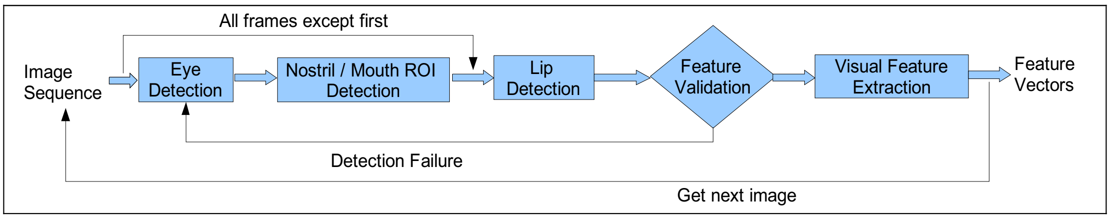


Figure 1 Feature extraction flow chart

Initially, the skin colour is obtained from a face image. The change in RGB values at row containing eyebrow is always over a certain threshold. After obtaining eyebrow location, sclera (The tough white fibrous outer envelope of tissue covering the entire eyeball except the cornea) pixels are counted and compared against sclera threshold, which is dependent on the face image size. This is carried out to precisely locate eyes. The approximate bounding box for the nose lays few rows below eyes. Nostrils are located by finding small dark region (shadows at nose opening). Mouth area is roughly located using eyes, nostrils location and knowledge about human face structure. After locating the mouth area, lips are segmented and a contour is extracted. Some of the existing methods for lip segmentation and associated difficulties are briefly described in the previous work section. To overcome those difficulties in lips segmentation, a novel algorithm to augment finding of lip pixels is presented. The change in red, green, blue colour components or their ratios is asymmetrical, however, it is observed that the change in hue value from skin to lips remains moderately constant regardless of person's skin colour. Thus, hue colour component is used to detect lip region [7].

$$Hue < \alpha \textit{ (without lipstick)} \tag{4}$$

$$Red\ Change + Green\ Change + Blue\ Change > \beta \textit{ (with lipstick)} \tag{5}$$

$$Red\,/\,Green > \partial \textit{ (with high illumination)} \tag{6}$$

Here α is a hue threshold such that any pixel having hue value less than α, falls in lip pixels category. This hue threshold α is calculated using hue value of the subject's skin. Sometimes the hue value for skin and lip is very low, especially in people with brown or black skin. It is also important to find the major change in hue values, which gives outer lip boundaries. This method of lip segmentation works well for most of the cases, except for subjects with lipstick colour dissimilar to lip colour or highly illuminated image frames. In such cases, the alternative methods given by equation (5) and (6) needs to be employed for lip segmentation. The threshold β is a colour change threshold to detect major colour change in mouth region of interest, which gives upper and lower lip boundaries and a lipstick colour. Here β is varied dynamically to allow variations in skin/lip colour for various people. The technique given by equation (6) is employed when lip pixels segmented using equation (4) or (5) are below or above certain thresholds. This indicates lips detection error. In such a case the outer lip colour is found using equation (6) and the whole mouth area is rescanned to find lip candidate pixels. Subsequently, lip features such as lip height, width, area, and perimeter are obtained by finding lip contour and lip corners. Other visual features such as teeth visibility and tongue visibility are extracted by thresholding method. Some results of lip segmentation are given in the figure 2.

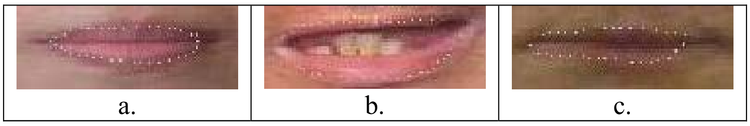


Figure 2 Results of lip segmentation (a. Subject A, b. Subject B, c. Subject E)

4 Visual Feature Validation

Initially, visual features extraction and recognition are carried out on alphabets instead of phonemes for simplicity. The algorithm described above for feature extraction was tested on 5 speakers [3 males, 2 females] of different race. Each subject was video recorded uttering alphabets ‘A’ to ‘Z’ and in reverse order ‘Z’ to ‘A’. The videos were recorded using a digital camera without using any markers or makeup under normal lighting conditions. A software ‘BimodalSR’ was developed for automatic visual feature extraction, error detection, feature validation and viseme classification. The difference between the feature locations found using above described algorithms and those found manually by human observer was used as the error criterion.

The comparison shows that the manual and automatic feature extractions yield similar results. They only differ at about 1-2mm for the mouth width and mouth height. This is a very accurate result given that a lip-tracking algorithm with no additional markers or made-up lips is used. Other features such as area and perimeter are found to be less accurate with average difference of around 10-12 pixels. Such inaccuracy is due to inability of finding the fine contour detail in some cases. It is also observed that in some of the cases, lip contour identification by human observer was error prone due to the gradual colour change near corners and shadows near bottom lip. Looking at each feature point separately reveals that the detection of upper boundary of upper lip is very accurate. The bottom edge of lower lip was detected with average inaccuracy of 1 pixel. However, the difference in the horizontal positions (lip corners) was 2-3 pixels on average because the inner lip contour could not be clearly distinguished from the lip flesh near the lip corners. The error occurred in automated feature detection and extraction is further reduced when vectors are obtained from feature values. Since the vectors are obtained using independent thresholds and more emphasis is given to the direction of change in vector, very highly accurate visual feature vector extraction and viseme classification is achieved.

Overall, the visual assessment of the extracted feature locations shows a high degree of accuracy. The algorithm fails only in a few frames, which are well detected by the confidence measure system. To validate the results acquired, the software compares the automatically extracted feature positions with the results from a manually extracted feature position values. Although the manual selection of features potentially introduces a new source of error, the human observer, it gives a clear clue on the efficiency of the automatic feature extraction algorithm. It is also used for system training and viseme classification. Furthermore, using the results, the mouth shapes such as fully open, rounded, closed, partially open were easily identified.

The performance of the visual feature extraction technique depends very much on accurate detection of mouth area, which is dependent on detection of eyes and nostrils. The eye detection technique imposes limitation on how much a face can be inclined. It allows around 40-50 degrees of inclination angle.

5 Performance Evaluation

The software ‘BimodalSR’ for automated visual feature extraction from an image sequence was tested on a system (Intel Pentium 4 CPU 2.80GHz, 512MB RAM, windows XP). The average time taken by different modules are given in table 1.

Module	Execution Time (ms)
Eye Detection	45-50ms
Mouth Detection (incl. Nostril Detection)	8-10ms
Lip Detection	18-20ms
Feature Extractor	3-5ms

Table 1 Execution time taken by various modules.

The percentage execution time breakdown for various modules is shown in figure 3. The chart shown below indicates that more than 50% of total execution time is taken by the eye detection module. Since eye detection is only carried out for the first image or in a case of tracking failure, very high overall speed for feature extraction is achieved.

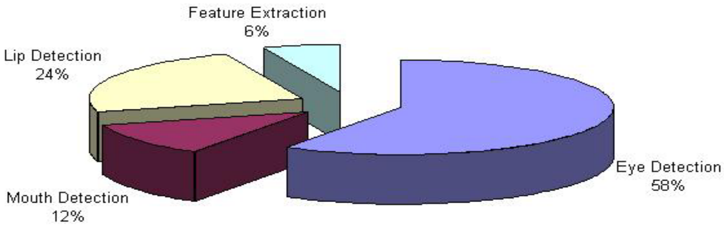


Figure 3 Execution time breakdown of feature extraction process modules

6 Conclusion

A robust approach for facial speech feature extraction has been described which uses a combination of algorithms and iterative thresholding method for lip segmentation and lip movement tracking for different subjects. High performance for locating and tracking visual features was achieved without the need of highlighting the lips with a lipstick or a reflective marker. The results of the feature extraction system were quite encouraging with only 1-2 pixels error in the lip corner detection whereas the detection of upper and lower lips boundaries was highly accurate. Other features such as area and perimeter are found to be less accurate with average difference of around 10-12 pixels. However, the feature detection error occurred is further reduced when vectors are obtained from feature values. This is achieved by emphasising more on the direction of the change in vector and using independent thresholds for diverse real-time situations.

Furthermore, these results strongly indicate the importance of using various approaches in diverse real-time situations for visual speech recognition. However, further investigation is required to test these algorithms on a larger database and for continuous speech recognition. The system, due to its adaptive nature, is extremely effective in identifying the lips even under very difficult conditions. The system takes advantage of iterative threholding to greatly improve the lip pixels segmentation and reuse of thresholds obtained initially to improve the speed.

References

[1] McGurk, H. & MacDonald, J. Hearing lips and seeing voices. Nature, 264, 746-748, 1976.

[2] E.D. Petajan. Automatic Lipreading to Enhance Speech Recognition. PhD thesis, University of Illinois at Urbana-Champaign, 1984.

[3] D. L. Jennings and D. W. Ruck. Enhancing automatic speech recognition with an ultrasonic lip motion detector. In Proc. ICASSP ’95, pages 868-871, Detroit, 1995.

[4] K. Mase and A. Pentland. Automatic lipreading by optical flow analysis. Technical Report 117, MIT Media Lab, 1991

[5] Christoph Bregler and Yochai Konig. “EIGENLIPS” For Robust Speech Recognition. In Proc. ICASSP ’94, pages II-669-II-672, Adelaide, Australia, April 1994.

[6] A. L. Yuille, P. Hallinan, D. S. Cohen, “Feature extraction from faces using deformable templates”, Int. J. Computer Vision, Vol. 8, pp. 99-111, August 1992.

[7] Premit Patel, Karim Ouazzane, Robert Whitrow, “Automated visual feature extraction for bimodal speech recognition”, IADAT-micv2005, 118-122.

[8] Stiefelhagen, Rainer / Meier, Uwe / Yang, Jie (1997): "Real-time lip-tracking for lip reading", In EUROSPEECH-1997, 2007-2010.

[9] Rao, R. and Mersereau, R. (1994), “Lip modelling for visual speech recognition”, In 28th Annual Asimolar Conference on Signals, Systems, and Computers, volume 2. IEEE Computer Society, Pacific Grove CA.

[10] Trent W. Lewis and David M. W. Powers, “Audio-visual speech recognition using red exclusion and neural networks”, Australian Computer Society, Australia, 149 - 156, 2002

[11] Yang, J. and Waibel, A. “A real time face tracker.” In Proceedings of WACV-1996, pages 142-147

[12] Coianiz, T., Torresani, L., and Caprile, B., “2d deformable models for visual speech analysis”, Speechreading by Man and Machine: Models, System and Application. NATO Springer-Verlag, pages 391-398.

2D, 3D Scene Analysis & Visualisation

Extraction and reconstruction of shape information from a digital hologram of three-dimensional objects

C. P. Mc Elhinney, J. Maycock, B. M. Hennelly, T. J. Naughton, J. B. McDonald
Dept. of Computer Science
National University of Ireland
Maynooth
Co. Kildare , Ireland
conormce@cs.nuim.ie tom.naughton@nuim.ie johnmcd@cs.nuim.ie

B. Javidi
Electrical and Computer Engineering
University of Connecticut
371 Fairfield Road, Unit 1157
Storrs, CT 06269, USA

Abstract

We present a technique to convert a digital hologram of a three-dimensional object into a surface profile of the object. A depth-from-defocus technique is used to generate a depth map for a particular reconstructed perspective of the scene. The Fresnel transform is used to effect defocus. This depth map is then used to create an extended focused image of the object. Through the combination of the extended focused image and the depth map we are able to reconstruct a pseudo three-dimensional representation of the object. Our method produces depth maps of a significantly higher resolution than current autofocus methods. The technique could be used in registration and three-dimensional object recognition applications.

Keywords: digital holography, scene reconstruction, 3D shape measurement, three-dimensional image processing

1 Introduction

Holography [1] is an established technique for recording and reconstructing real-world three-dimensional (3D) objects. Digital holography [2, 3, 4, 5, 6, 7, 8, 9] and digital holographic image processing [9, 10, 11, 12, 13] have recently become feasible due to advances in megapixel CCD sensors with high spatial resolution and high dynamic range. A technique known as phase shift interferometry (PSI) [6, 8] is used to create our in-line digital holograms [9, 10]. The resulting digital holograms are in an appropriate form for data transmission and digital image processing.

Digital holography is one of several possible optical techniques for recovery of shape information [14]. Digital holographic microscopy [15, 16, 17] and interferometric [3] methods use a phase unwrapping approach to obtain the profile of an object. Many existing 3D imaging techniques are based on the explicit combination of several two-dimensional perspectives through digital image processing. Multiple perspectives of a 3D object can be combined optically, in parallel, and stored together as a single complex-valued digital hologram. We apply a technique known as depth-from-defocus (DFD) to create depth maps of the objects encoded in our digital holograms.

Several approaches for focusing digital holograms have been reported in the literature [18, 19, 20]. Two of these approaches attempt to focus the numerically reconstructed complex wave field [18, 19]. Liebling [18] used Fresnelets on holograms captured using digital holographic microscopy [21], to discover the focal plane for an micrometric object, but no shape measurement is attempted. Gillespie and King [19] proposed the use of the self-entropy of a hologram's quantised phase to calculate the sharpness of a numerically reconstructed complex wave field. Another approach is to try to reconstruct the 3D scene using a focus metric calculated on reconstructions of the hologram at different depths. Ma $et\ al.$ [20] used variance to calculate a depth map from a digitised analog hologram. In this paper we extend on our previous work [22] by using non-overlapping blocks to create an extended focused image and a pseudo 3D representation of the hologram.

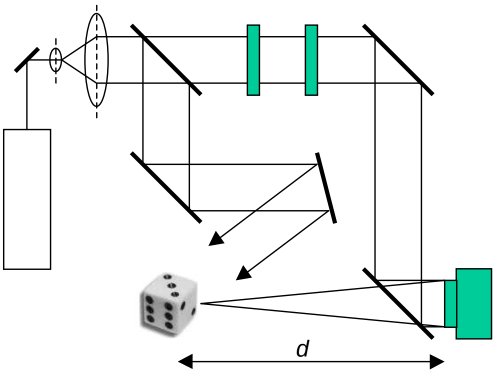


Figure 1: Experimental setup for PSI: BE, beam expander; BS, beam splitter; RP, retardation plate; M, mirror.

In Sect. 2, we describe how 3D objects are captured using phase-shift digital holography. We then describe the experiments involved in extracting a depth map from a digital hologram in Sect. 3. Section 4 details how to reconstruct a pseudo 3D representation of an object encoded in a digital hologram and, finally, conclusions are drawn in Sect. 5.

2 Phase-Shift Digital Holography

We record Fresnel fields with an optical system based on a Mach-Zehnder interferometer (see Fig. 1). A linearly polarised Argon ion (514.5 nm) laser beam is expanded and collimated, and divided into object and reference beams. The object beam illuminates a reference object placed at a distance of approximately $d = 350$ mm from a 10-bit 2028×2044 pixel Kodak Megaplus CCD camera. Let $U_0(x, y)$ be the complex amplitude distribution immediately in front of the 3D object. The linearly polarised reference beam passes through half-wave plate RP$_1$ and quarter-wave plate RP$_2$. By selectively removing the plates we can achieve four phase shift permutations of 0, $-\pi/2$, $-\pi$, and $-3\pi/2$. The reference beam combines with the light diffracted from the object and forms an interference pattern in the plane of the camera. At each of the four phase shifts we record an interrogram. We use these four real-valued images to compute the camera-plane complex field $H_0(x, y)$ by PSI [6, 8]. We call this computed field a digital hologram.

A digital hologram $H_0(x, y)$ contains sufficient amplitude and phase information to reconstruct the complex field $U(x, y, z)$ in a plane in the object beam at any distance z from the camera [4, 8, 9]. This can be calculated from the Fresnel approximation [23] as

$$U(x, y, z) = \frac{-\mathrm{i}}{z} \exp\left(\mathrm{i}\frac{2\pi}{\ }z\right) H_0(x, y) \star \exp\left[\mathrm{i}\pi\frac{(x^2 + y^2)}{z}\right] \quad , \tag{1}$$

where is the wavelength of the illumination and $\star$ denotes a convolution operation. At $z = d$, and ignoring errors in digital propagation due to discrete space (pixelation) and rounding, the discrete reconstruction $U(x, y, z)$ closely approximates the physical continuous field $U_0(x, y)$.

Furthermore, as with conventional holography [23, 24], a windowed subset of the Fresnel field can be used to reconstruct a particular view of the object. As the window explores the field a different angle of view of the object can be reconstructed. The range of viewing angles is determined by the ratio of the window size to the full CCD sensor dimensions. Our CCD sensor has approximate dimensions of 18.5×18.5 mm and so a 1024×1024 pixel window has a maximum lateral shift of 9 mm across the face of the sensor. With an object positioned $d = 350$ mm from the camera, viewing angles in the range of 1.5 are permitted. Smaller windows will permit a larger range of viewing angles at the expense of image quality at eachviewpoint.

3 Extraction of shape information

It is possible to reconstruct the object wavefield at any depth by altering the distance parameter z of Equation 1. One method for reconstructing a hologram at the most in-focus plane is to use a DFD technique. This is done by reconstructing the hologram over a range of depths and evaluating each 2D image using a focus metric, this will return the in-focus plane's depth. This technique relies on the assumption that a large majority of the scene is in focus at a certain depth. If there are multiple objects

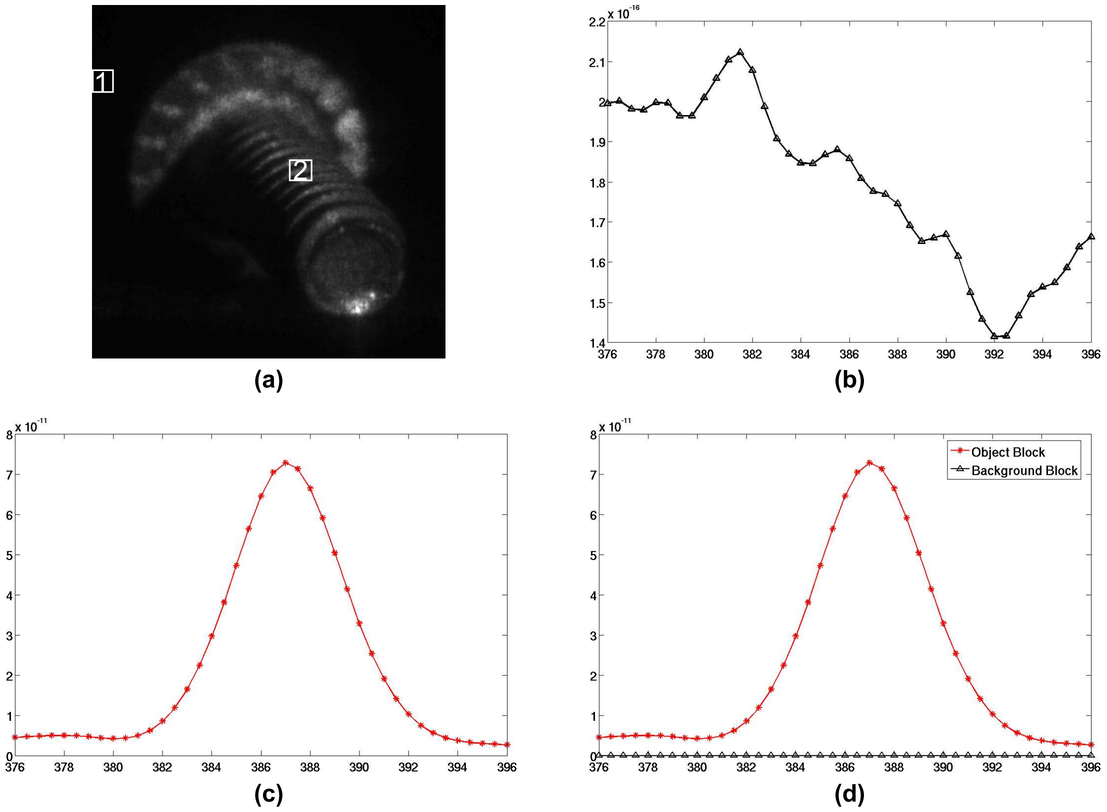


Figure 2: Reconstruction of the bolt,(a), and two trace plots of variance calculated on the amplitude of two blocks, (b) trace plot of block 1 positioned in a background region, (c) trace plot of block 2 positioned on the threads of the bolt, (d) both trace plots plotted on the same scale.

at different depths, or if the object is not relatively smooth, the scene needs to be partitioned into blocks. Each block can then be processed using a focus metric in order to gather depth information. This depth information can then be used to create a depth-map of the scene.

An approach for the recovery of shape information from digital holograms was proposed by Ma [20]. He uses variance as a focus metric to gather depth information from a digital hologram. This is defined as,

$$V(k) = \frac{1}{n \times n} \sum_{i=1}^{n} \sum_{j=1}^{n} \left[I_k(i,j) - \overline{I_k} \right]^2 \qquad (2)$$

where I_k is the k^{th} block in the image and $\overline{I_k}$ is the mean of the input block. Through block processing of the complex wave field and the calculation of variance on these blocks, depth maps have been successfully created from digitally scanned material holograms. We were the first to apply this technique to digitally captured holograms [22].

Our holograms are reconstructed at a set of different depths L using Equation 1, starting at a depth d_0. All our reconstructions have had a selected speckle reduction technique applied to them [25]. The size of each reconstruction is $M \times N$ with Δz being the interval between reconstructions. We calculate the focus at each depth by separating each reconstruction into blocks of size $n \times n$, creating $m' \times n'$ blocks. Each of these blocks are then processed using variance as a focus metric. The estimated depth for each block is evaluated by finding the depth at which variance exhibits a maximum. This gives us the shape information we need in the form of a depth map, where each value in the depth map represents the distance from the hologram plane to the corresponding block in the scene.

3.1 Non-Overlapping Blocks Approach

Depth maps created using the approach described above can suffer from noise introduced by the estimation of depths in background regions and incorrect estimation of depth in object regions. A reconstruction of the bolt hologram is shown in Fig. 2(a), the two labelled blocks are an object block and a background block. Trace plots for variance calculated on these blocks are shown for the background block in Fig. 2(b), and for the object block in Fig.2(c). In Fig. 2(d) we have plotted the two trace plots on the same scale, it is clear that variance calculated on the background block returns a lower value than

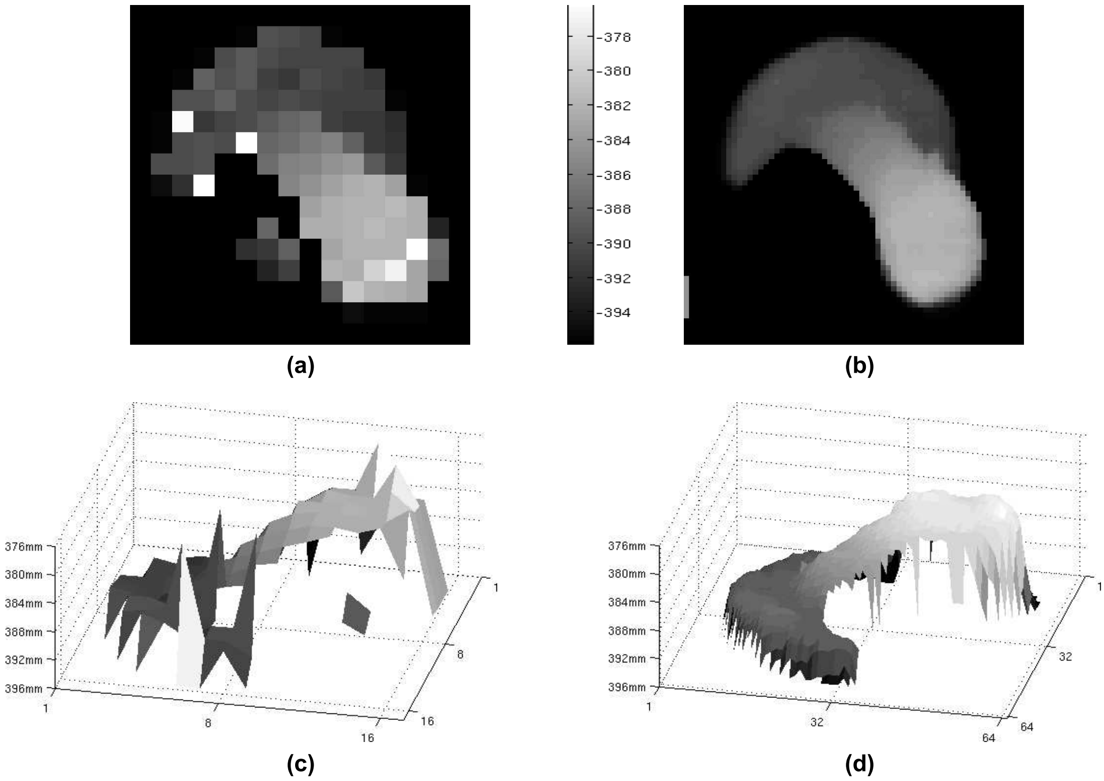


Figure 3: 2D depth maps acquired with (a) a 64×64 block size, (b) a 16×16 block size. Surface plots of the depth maps acquired with (c) a 64×64 block size, (d) a 16×16 block size.

that for the object region, in the order of $10\ ^{5}$ less. We found this to be true in the general case and used this to threshold our depth maps and remove depth estimations relating to background regions.

We first extracted shape information from a digital hologram using non-overlapping blocks [22]. This resulted in depth maps which were $\frac{M}{n} \times \frac{N}{n}$ in size. We reconstructed a hologram of a bolt forty different depths ($L = 40$), with $0.5mm$ intervals ($\Delta z = 0.5$mm) and starting at a depth of $376mm$ ($d_0 = 376$mm). The resolution of the reconstructions are 1024×1024 pixels. It was necessary to experiment with different blocks sizes to determine what the impact of larger or smaller block sizes on the depth map would be. A reduction in block size can improve the definition of features in the depth map. At some threshold point this leads to more discontinuities in the depth map. This behaviour at the threshold point is scene dependent. The discontinuities manifest themselves as both holes and spikes, which means a larger closing operation is required thus reducing the overall accuracy of the depth map. We have found that using block sizes of less than 16×16 leads to a high proportion of holes and spikes in the depth map due to the presence of speckle in the reconstructions. This high quantity of holes and spikes require more post-processing of the depth map then is required for larger block sizes and leads to an overally smoothed depth map. The depth maps acquired through calculating variance on the bolt hologram with two different block sizes are shown in Fig. 3. Figure 3(a) shows an image representation of a depth map acquired using a 64×64 block size and Fig. 3(b) shows a surface plot of this depth map. The white spots that can be seen in Fig. 3(a) are blocks whose depth has been incorrectly estimated, through median filtering these blocks are capable of being suppressed. However, no median filtering was applied to this depth map due its destructive impact on the depth map. When we reduced the block size to 16×16, it was necessary to median filter the depth map in order to suppress error caused by the lower block size. The depth map and surface plot shown in Fig. 3(b) and (c) have been median filtered with a 9×9 filtering neighbourhood. Two of the main regions in the bolt are the front of the bolt and the back of the bolt. As can be seen from Fig. 3, these are being focused at the correct depths of 390mm and 382mm, respectively.

3.2 Overlapping Blocks Approach

In order to create depth maps with an improved resolution we attempted to use variance calculated on overlapping blocks. This would increase the resolution of our depth maps from $\left(\frac{M}{n}\right) \times \left(\frac{N}{n}\right)$, using a non-overlapping approach, to $(M - n) \times (N - n)$. This increased resolution significantly improves the quality of our depth map. This improvement is most visible in the threaded region of the bolt, and also

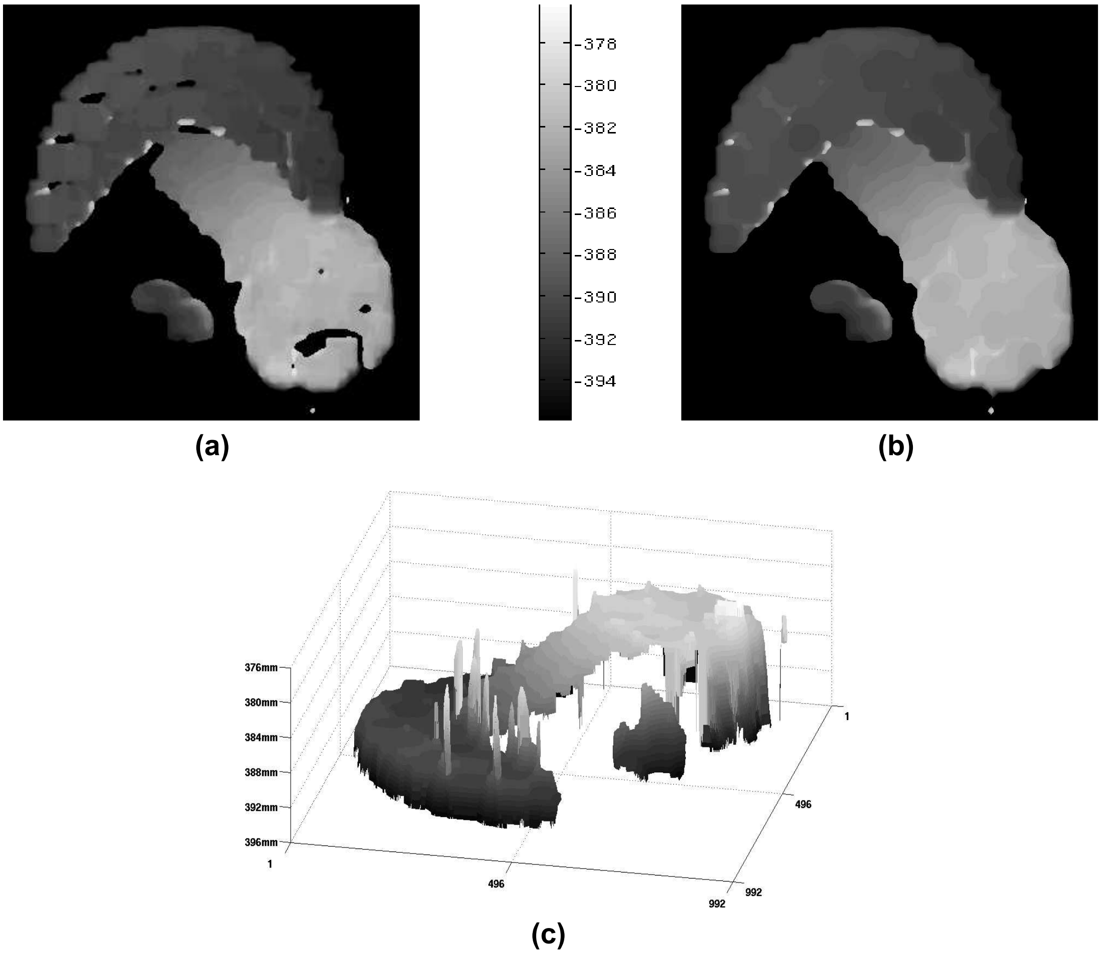


Figure 4: 2D depth maps acquired using a 64×64 block size (a) prior to closing, (b) after closing and (c) a surface plot of the depth map.

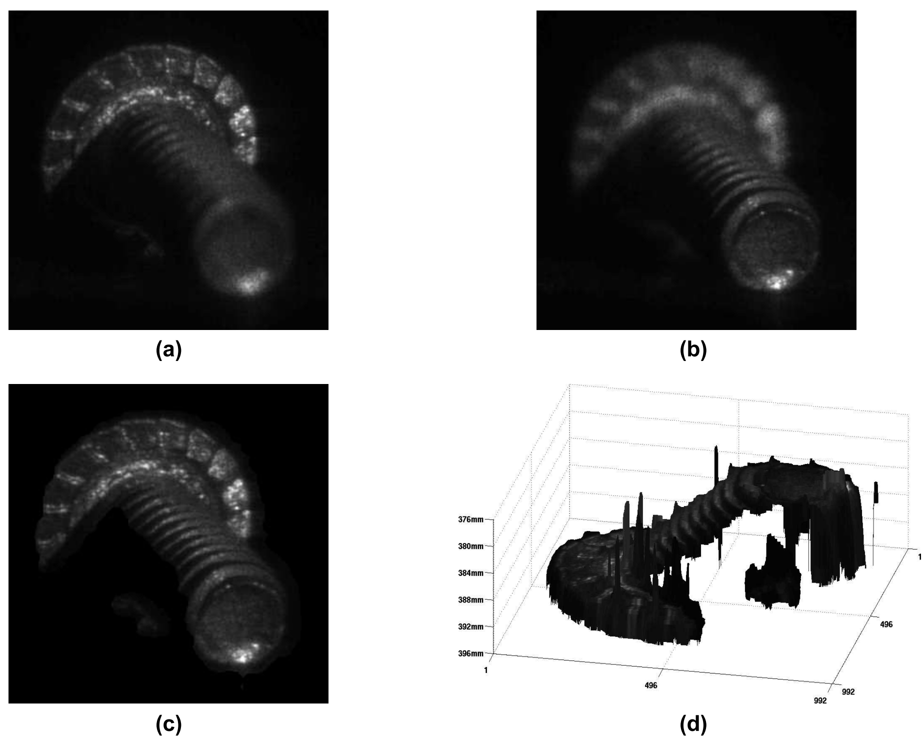

(a) (b) (c) (d)

Figure 5: Reconstructions of the bolt hologram, (a) reconstruction of the bolt with the back in focus, (b) reconstruction of the bolt with the front in focus, (c) extended focus image and (d) psueod 3D representation of the bolt.

on the back face of the bolt which is not perpendicular to the optical axis. On a 3Ghz Pentium PC with 1GB of RAM, using a 64×64 block size, it takes less than 3 hours to run this technique. The results of using variance as a focus metric with a 64×64 block size using an overlapping approach can be seen in Fig. 4.

We applied median filtering with a neighbourhood of 15×15 to the depth map, as shown in Fig. 4(a). Some object regions in the depth map were incorrectly removed after thresholding was applied. We used a mathematical morphological closing operation on the depth map using a disk structuring element with a 25 pixel diameter to fill in the holes in the image. The resultant depth map can be seen in Fig. 4(b), and the surface plot of this depth map is shown in Fig. 4(c). Some blocks have not had their depth correctly estimated, as can be seen in Fig. 4(c). It is our hope that through the evaluation of different focus metrics we will reduce the number of these blocks such that combination of median filtering and closing will suppress them. Both median filtering and the closing operation are destructive error suppression and error removal techniques. Therefore, median filtering and closing have the effect of smoothing the data, this can remove details in the original depth map. We note that the increased resolution given through using an overlapping approach produces depth maps with much greater detail.

4 Scene Reconstruction of three-dimensional objects

With the increased resolution obtained through using a overlapping block approach, we are able to use our depth map to create an image with an extended depth of field encompassing the whole object. This is an image where each pixel in the image is in-focus. This is one major advantage of using an overlapping block approach over the non-overlapping approach. Our non-overlapping block experiments produced depth maps of size 16×16 and 64×64, only allowing for an extended focused image of that size. However, using the overlapping approach the extended focused image produced was 960×960 pixels.

To create the extended focused image we use L reconstructions and a depth map. Each pixel location (i, j) in the depth map contains a depth value d . Our extended focused image copies the value from position (i, j) in the reconstruction obtained at depth d. Figure 5(a) and Fig. 5(b) show two reconstructions of the bolt hologram in which the back of the bolt and the front of the bolt are in focus, respectively. The extended focused image is shown in Fig. 5(c). This extended focused image also verifies, qualitatively, that our depth maps are calculating the correct depth. If an incorrect depth had been calculated, the object, or large regions of the object, would be blurred in the extended focused image. Figure 5(d) shows a pseudo 3D representation of the object obtained through the combination of the depth map and the extended focused image.

5 Conclusion

We have presented a technique for extracting shape information from a digital hologram. We have shown depth maps created by this technique, using two different approaches on a digital hologram of a bolt. We have demonstrated that an overlapping block approach is superior to a non-overlapping approach due to its increased resolution. This increased resolution gives us a more accurate depth map and it also, significantly, allows us to create an extended focused image of the object and create a pseudo three-dimensional representation of the object. Our technique might be applicable in situations where the object has too large a depth range, too rough of a surface, or is captured at too low a resolution for other techniques such as phase unwrapping to work adequately but may only be appropriate for objects with slow-varying depth. Using reconstructions with a smaller interval between them would result in a more accurate depth map. In the presentation we will demonstrate experimental results using other digital holograms and scenes. We intend to extend this technique through quantitative measurement of known objects to measure the accuracy of our approach, and to develop non-destructive error removal techniques.

Acknowledgements

The authors wish to thank Enrique Tajahuerce and Yann Frauel for use of their digital hologram data. Support is acknowledged from Enterprise Ireland and Science Foundation Ireland.

References

[1] D. Gabor. A new microscope principle. *Nature*, 161:77–79, 1948.

[2] J. W. Goodman and R. W. Lawrence. Digital image formation from electronically detected holograms. *Applied Physics Letters*, 11:77–79, 1967.

[3] T. Kreis. *Handbook of Holographic Interferometry*. Optical and Digital Methods, Wiley-Vch, Weinheim, 2005.

[4] L. Onural and P. D. Scott. Digital decoding of in-line holograms. *Opt. Eng.*, 26:1124–1132, 1987.

[5] L. P. Yaroslavskii and N. S. Merzlyakov. *Methods of Digital Holography*. Consultants Bureau, New York, 1980. Translated from Russian by Dave Parsons.

[6] J. H. Bruning, D. R. Herriott, J. E. Gallagher, D. P. Rosenfeld, A. D. White, and D. J. Brangaccio. Digital wavefront measuring interferometer for testing optical surfaces and lenses. *Applied Optics*, 13:2693–2703, 1974.

[7] U. Schnars and W. P. O. Jüptner. Direct recording of holograms by a ccd target and numerical reconstruction. *Applied Optics*, 33:179–181, 1994.

[8] I. Yamaguchi and T. Zhang. Phase-shifting digital holography. *Optics Letters*, 22:1268–1270, 1997.

[9] B. Javidi and E. Tajahuerce. Three-dimensional object recognition by use of digital holography. *Optics Letters*, 25:610–612, 2000.

[10] Y. Frauel, E. Tajahuerce, M.-A. Castro, and B. Javidi. Distortion-tolerant three-dimensional object recognition with digital holography. *Applied Optics*, 40:3887–3893, 2001.

[11] T. J. Naughton, Y. Frauel, B. Javidi, and E. Tajahuerce. Compression of digital holograms for three-dimensional object reconstruction and recognition. *Applied Optics*, 41:4124–4132, 2002.

[12] O. Matoba, T. J. Naughton, Y. Frauel, N. Bertaux, and B. Javidi. Real-time three-dimensional object reconstruction by use of a phase-encoded digital hologram. *Applied Optics*, 41:6187–6192, 2002.

[13] J. Maycock, C. P. McElhinney, B. M. Hennelly, T. J. Naughton, J. B. McDonald, and B. Javidi. Reconstruction of partially occluded objects encoded in three-dimensional scenes by using digital holograms. *Applied Optics*, 45:2975–2985, 2006.

[14] F. Chen, G. Brown, and M. Song. Overview of three-dimensional shape measurement using optical methods. *Optical Engineering*, 39:10–22, 2000.

[15] P. Ferraro, S. Nicola, G. Coppolla, A. Finizio, D. Alfieri, and G. Pierattini. Controlling image size as a function of distance and wavelength in fresnel-transform reconstruction of digital holograms. *Optics Letters*, 29:854–856, 2004.

[16] P. Ferraro, S. Grilli, D. Alfrieri, S. Nicola, A. Finizio, G. Pierattini, B. Javidi, G. Coppolla, and V. Striano. Extended focused image in microscopy by digital holography. *Optics Express*, 13:6738–6749, 2005.

[17] T. Zhang and I. Yamaguchi. Three-dimensional microscopy with phase-shifting digital holography. *Optics Letters*, 23:1221–1223, 1998.

[18] M. Liebling and M. Usner. Autofocus for digital fresnel holograms by use of a Fresnelet-sparsity criterion. *J. Opt. Soc Am. A*, 21:2424–2430, 2004.

[19] J. Gillespie and R. King. The use of self-entropy as a focus measure in digital holography. *Pattern Recognition Letters*, 9:19–25, 1989.

[20] L. Ma, H. Wang, Y. Li, and H. Jin. Numerical reconstruction of digital holograms for three-dimensional shape measurement. *Journal of Optics A*, 6:396–400, 2004.

[21] T. Colomb, E. Cuche, P. Dahlgen, A. Marian, F. Montfort, C. Depeursinge, P. Marquet, and P. Magistretti. 3D imaging of surfaces and cells by numerical reconstruction of wavefronts in digital holography applied to transmission and reflection microscopy. *Proc. IEEE - International Symposium on Biomedical Imaging*, pages 773–776, 2002.

[22] C. P. McElhinney, J. Maycock, T. J. Naughton, J. B. McDonald, and B. Javidi. Extraction of three-dimensional shape information from a digital hologram. *Proc. SPIE*, 5908:30–41, 2005.

[23] J. W. Goodman. *Introduction to Fourier Optics*. Roberts and Company, Englewood, Colorado, third edition, 2005.

[24] H. J. Caulfield. *Handbook of Optical Holography*. Academic Press, New York, 1979.

[25] J. Maycock, B. M. Hennelly, J. B. McDonald, T. J. Naughton, Y. Frauel, A. Castro, and B. Javidi. Reduction of speckle in digital holography by discrete fourier filtering. *submitted to Opt. Lett.*, 2006.

Reproduction of Sound Signal from Gramophone Records using 3D Scene Reconstruction

Baozhong Tian and John L. Barron
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada, N6A 5B7
{btian, barron}@csd.uwo.ca

Abstract

Preserving invaluable historic recordings has drawn some interest because the traditional record playback system wears out the record gradually. This paper presents a non-contact method to reproduce sound signal from gramophone records using 3D scene reconstruction of the micro-grooves cast on a record surface. Because of the unique shape of the microgroove, a planar assumption was made during scene reconstruction and recovered surface orientation was used to reproduce the sound signal. A robust estimation method was developed to reduce noise effects. Results from synthetic data were shown to test the technique.

Keywords: Sound Signal Reproduction, Scene Reconstruction, Gramophone Record, Robust Estimation, Surface Orientation.

1 Introduction

Reproducing sound mechanically on a record started as early as 1885 and the technology of recording and retrieving acoustic signals on gramophone records reached its peak during the 1970's, just before the digital format compact disk (CD) took over the mass marketing of music. Although the audio quality of CD is judged to be very good by most people, some audiophiles believe that the sampling rate of a CD (44.1kHz) is not high enough to reproduce the rich musical information faithfully. Today there is still some high-end record playing equipment in production. However, no matter how great a system performs, it wears out the record gradually due to the physical contact between the stylus and the record groove.

There also exists a lot of historical recordings that need to be archived. The problem with these recordings is that they have become so fragile that they can not tolerate being played back using a traditional style turntable with a mechanical stylus. This problem motivates research on non-contact record playing systems.

1.1 Traditional Method of Sound Reproduction

We will take stereo gramophone records (stereo LP) as our example, since other formats of mechanical records are similar. During the record cutting procedure, the left and right channel signals control the speed of the cutting stylus at a +45/-45 lateral manner, i.e. a composition of two orthogonal speeds perpendicular to each other, while the record rotates at a constant speed. This is called modulation of the grooves. The movement of the stylus determines the slopes in the tangential direction of the groove walls. This record cutting method keeps the left and right groove walls' modulation independent from each other. When the play back stylus has a similar setup as the cutting stylus, stereo signals can be reproduced. The electrical signal outputs are proportional to the +45/-45 lateral speeds of the stylus while riding along the groove and modulated by the groove walls.

Figure 1a illustrates the top view of the movements of record and stylus. The stylus has a tangential speed V_T relative to the groove due to the record rotation. There are also left and right lateral movements of the stylus (V_L and V_R) in +45/-45 directions. Figure 1b shows a cross section view of the compound +45/-45 lateral movement of the stylus.

The major goal of sound reproduction is to track the groove walls as precisely as possible. The conventional method uses a diamond-tip stylus to run along the V-shaped groove by applying a certain tracking force on the stylus. The problem is that the stylus has some weight, so the tracking of a

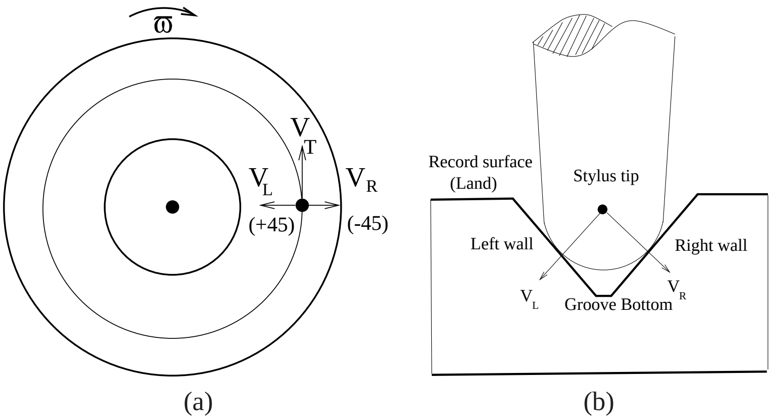


Figure 1: Illustrations of the movement of the record and the stylus: (a) top view, (b) cross section view.

high frequency signal is more difficult. The physical contact also produces a high temperature that softens the record surface and prevents it from playing well again within approximately 6 hours of time [Micrographia, 2005]. Other problems include groove damage such as scratches and small particles that result in annoying clicks, pops and degradation of sound over time and the maintenance of the correct settings of the turntable, tone arm, cartridge and stylus requires frequent adjustment.

1.2 Literature Survey of Non-Contact Record Playing Methods

Because of the problems with traditional record playback systems, people turned to the easy-to-use CDs as soon as they appeared in the early 1980s. But efforts at developing a non-contact record playback system did not cease. ELP corporation [ELP, 1997] spent ten years to develop a laser turntable (invented by Robert E. Stoddard et al. [Stoddard and Stark, 1989, Stoddard, 1989]) utilising five laser beams to track the microgroove optically. This is a pure analogue process, but it is so sensitive to foreign particles in the groove and on the record surface that it requires the record to be cleaned every time it is played. This ELP laser turntable uses two of the five beams of the laser to track the groove walls and the other three laser beams for groove tracking. This has two main advantages: the laser beams are weightless and can be made as thin as $2\mu m$ in diameter, which is much thinner than a high-end stylus (4-12μm). However, the system is very complicated and expensive and it only works well with black records because of the reflective nature of the material. Coloured records may produce unpredictable results [ELP, 1997].

Because the laser turntable is very expensive (in the price range of a small car) and because it is very sensitive to the cleanness of the record, some research has been carried out to study the feasibility of reproducing the sound signal by image processing methods. In 2002, Ofer Springer [Springer, 2002] proposed an idea he called the virtual gramophone. Springer's idea is to scan the record as an image and write a decoder to apply a "virtual needle" following the groove spiral form. However, when the authors listened to a sample decoded sound (http:www.cs.huji.ac.il/ ˜springer/), the music was judged to be barely recognisable. Inspired by Springer's idea, a group of Swedish students [Olsson et al., 2003] further developed a system to use more sophisticated digital signal processing methods such FIR Wiener filtering and spectral subtraction to reduce noise level in the reproduced sound, resulting in a better result than that of Springer's. Both systems used an off-shelf scanner, which limited the resolution of the images, to a maximum of 2400dpi or $10\mu m$ per pixel. At this resolution, the quantisation noise is quite high because the maximum lateral travel of the groove is about $150\mu m$.

[Fadeyev and Haber, 2003] developed a 2D method to reconstruct mechanically recorded sound by image processing. The resolution was greatly improved by the aid of micro-photography. Their algorithm detects the groove bottom as an edge in the image and then differentiates the bottom edge shape to reproduce sound signals. Their method uses only the groove bottom information, which was not always very well defined and may be distorted by dirt particles. The groove walls, which contain rich sound informations, were ignored. They also introduced a 3D method to reproduce the vertically modulated records such as wax cylinders. But their 3D method requires complicated 3D profile scanning, such as that provided by a laser confocal scanning probe, which is a very slow process.

Johnsen et al. [Cavaglieri et al., 2001, Stotzer et al., 2003, Stotzer et al., 2004] also proposed a 2D method they called the VisualAudio concept. A picture of the record was taken using a large format film as big as the record. The film was then scanned using a rotating scanner, which is actually a line scan camera positioned above the film while the film is being rotated on a turntable. Edges were then detected from the digitised image and then sound signals were computed from the edges. Unlike the method of [Fadeyev and Haber, 2003], Johnsen et al. used the groove and surface intersection as the edge instead of using groove bottom. This gives them the capability to reproduce the sound from stereo 33rpm recordings. Also the use of the rotating scanner eliminated the need for adjusting the sample rate as the groove turned close to the record's centre. The images are rectangular, and not circular, as scanned by a flat-bed scanner. A 10× magnifier was fitted to the rotating scanner to get the desired image resolution. A Signal to Noise Ratio (SNR) analysis showed that a satisfying SNR of 40dB can be achieved if the standard deviation (σ_n) of edge position noise was kept below 1.28 μm. However, listening to the reproduced sound clips from their web site (http://www.eif.ch/visualaudio/) indicated that the noise level needs to be further reduced.

2 Proposed Method

We propose a sound reproduction method based on Computer Vision technologies such as optical flow and surface reconstruction. The proposed method uses a microscope to obtain a sequence of magnified images of the groove walls and uses 3D scene surface reconstruction to calculate the slopes of the walls. Figure 2 shows the system diagram. The major features of the proposed method can be summarised as:

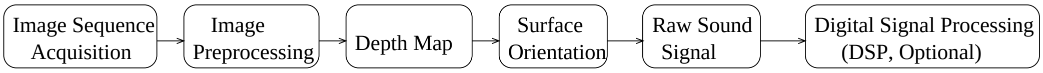


Image Sequence Acquisition
Image Preprocessing
Depth Map
Surface Orientation
Raw Sound Signal
Digital Signal Processing (DSP, Optional)

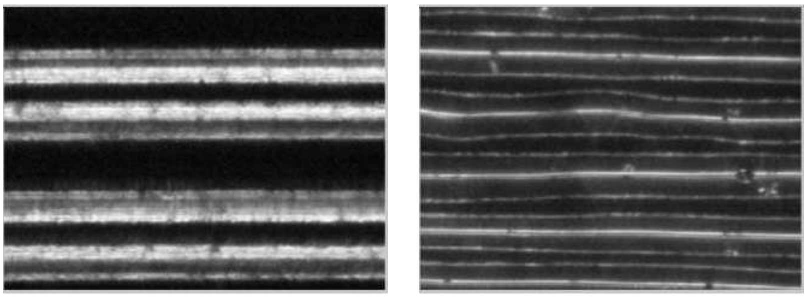
Figure 2: System diagram.

- Using as much information on the record as possible to reproduce the sound. Plenty of information is stored via the surface orientation of the groove walls, which is not used by 2D methods during their scanning/photographing processes. A 2D method only computes detectable edges such as a groove's bottom or groove-surface (land) intersections.
- Computer Vision technologies such as optical flow and depth map estimation are applied to this problem to obtain the 3D information characterising the groove, thus eliminating the requirement for a specialised 3D scanning device.
- Robust estimation techniques help choose the best areas of the groove wall for the computation and reject noisy areas which have been damaged by scratches and dirt particles, reducing the level of the noise and improving the quality of the reproduced sound.

We will discuss the individual system components below.

2.1 Image Sequence Acquisition

We need many groups of image sequence to cover the entire groove. Each group contains 36 frames of images. The two consecutive frames in same group should differ by only a few pixels. When the camera moves to the next segment of groove to capture another group of images, the last image of the current group should overlap with the last image of the previous group by a small amount so that the reconstructed groove is continuous when paired together.



(a) (b)

Figure 3: Two pieces of groove from (a) a 78rpm SP record and (b) a 33rpm LP record. The magnifying factor is 60X.

Figure 3 shows images of grooves under a microscope. The magnification factor of the microscope is set to be such that the field of view covers about 600μm in width so that for a camera with 640×480 pixels, the horizontal spatial resolution is about $1\mu m$. The illumination is set to a +45/-45 degree so that the groove walls are bright while the record surface and the groove bottom are dark. We are currently acquiring a better microscope with higher resolution and magnification camera to improve the image quality.

2.2 Image Preprocessing

The images have to be preprocessed, i.e. we need to compute the image intensity derivatives and the optical flow fields, etc. before we can compute depth maps. We experiment with implementations of two standard differential optical flow techniques, namely those of [Horn and Schunck, 1981] and [Lucas and Kanade, 1981] with differentiation by [Simoncelli, 1994].

The spatial-temporal differentiation method requires the scene to be rigid and smooth so that differentiation by convolution can be performed. Once differentiation has been performed, optical flow can be computed. Lucas and Kanade assume the flow is locally constant and use the motion constraint equation in a local least squares calculation to recover the optical flow. Horn and Schunck combined the motion constraint equation with a global smoothness term (the optical flow varies smoothly everywhere) to constrain the estimated velocity field in a regularisation (iterative) framework. We have to adapt these algorithms to our problem by imposing various constraint that arise from computing optical flow from record groove images, i.e. local surface planarity and uni-directional flow. The flow fields of Horn and Schunck look denser than that of Lucas and Kanade's but take a much longer time to compute. The computation and visualisation of depth requires a large number of such flow fields.

2.3 Depth Map Computation

We did a survey [Tian and Barron, 2005] of 4 recent algorithms for dense depth maps (from image velocities or intensity derivatives) which appeared to give good results in the literature. All of these algorithms assume known camera translation and rotation (or can be made to have this assumption). The 4 algorithms are those by [Heel, 1990], [Matthies et al., 1989], [Hung and Ho, 1999] and [Barron et al., 2003]. For a detailed description and discussions on these algorithms, please refer to [Tian and Barron, 2005]. Quantitative results show that the methods of Barron et al. is the best over all.

We report here experimental results for Barron et al.'s algorithm on synthetic record groove images and on real groove images with encouraging results. Because the groove wall orientation can be described by 2 angles, one of which is constrained and because the vertical component of image velocity is always very small (uni-direction constraint), we anticipate imposing such constraints will yield better even results. For example Barron et al.'s method could be modified to use only horizontal velocities, like Matthies et al. and effectively have only one angle of the surface orientation to track in the Kalman filter.

2.4 Robust Estimation of Surface Orientation

Surface orientation is computed from depth using least squares. Assuming local planarity, the surface orientation $\hat{\alpha}$ of a local neighbourhood is constant and satisfies the planar equation $\hat{\alpha} \cdot \vec{P} = c$, where $\vec{P} = [X, Y, Z]$ is the 3D coordinate of a pixel and c is a constant. We can solve this linear system using a robust estimation method called Local M-Estimates, recommended in Press et al. [Press et al., 1992]. This should give more accurate surface orientation as outlier data will be suppressed.

We present a robust estimation formulation of this calculation as follows. We use vector $\vec{g} = (g_1, g_2, g_3)$ to denote $\frac{\hat{\alpha}}{c} = (\frac{\alpha_x}{c}, \frac{\alpha_y}{c}, \frac{\alpha_z}{c})$. We can set up a least square system:

$$W \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & a_{N3} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = W \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}, \tag{1}$$

or

$$WA\vec{g} = WB \tag{2}$$

where W is a $N \times N$ diagonal matrix with diagonal elements acting as the weights for the N equations, and

$$a_{i1} \quad = \quad X_i \tag{3}$$

$$\begin{aligned} a_{i2} &= Y_i & (4) \\ a_{i3} &= Z_i \text{ and} & (5) \\ b_i &= 1. & (6) \end{aligned}$$

The solution is $\vec{g} = (A^T W^2 A)^{-1} A^T W^2 B$.

The weight matrix W plays a critical role in this robust estimation calculation. Initially, W is I so that a rough solution is obtained. Using this solution, we can refine W using the Lorentzian estimator, [Black and Anandan, 1996] ρ:

$$\rho(d_i, \sigma) = \log\left(1 + \frac{1}{2}\left(\frac{d_i}{\sigma}\right)^2\right) \qquad (7)$$

and the influence function ψ (which is the derivative of ρ):

$$\psi(d_i, \sigma) = \frac{2d_i}{2\sigma^2 + d_i^2}, \qquad (8)$$

where σ is a scale parameter and d_i is the residual value of each equation:

$$d_i = |a_{i1}g_1 + a_{i2}g_2 + a_{i3}g_3 - b_i|. \qquad (9)$$

Then the weight matrix elements get updated as:

$$w_i = \frac{\psi(d_i, \sigma)}{d_i}. \qquad (10)$$

We can re-calculate $\vec{g}$ again using the updated W. This procedure is repeated until one of the following stopping criteria is met:

- the total residual is smaller than some threshold: $||d_i||_2 < \tau_1$,
- the total residual begins to diverge:
  $||d_i||_{t^-} - ||d_i||_t < \tau_2$ or
- the number of iterations reaches a limit.

The second threshold, τ_2, which is a small positive number, allows the total residual to vary up and down a bit before iterations are considered to be converging or diverging.

According to Black and Anandan, tuning the scale parameter σ may work well given that the initial approximation for it is not too bad. Since in the Lorentzian estimator, a residual d_i is considered an outlier if $d_i \geq \sqrt{2}\sigma$, lowing σ after each iteration will reveal more and more outliers. Another benefit we can get from this is that the number of outliers could help us to determine whether the value of σ is properly chosen and when to terminate the iterations.

2.5 From Surface Orientation to Sound Signal

Once the surface orientations are computed, they need to be interpreted into sound signals so that they can be played. Figure 4 illustrates a piece of a groove, showing the left surface orientation $\hat{\alpha}_L$. The figure also shows the two angles (θ_{XY} and θ_{YZ}) that determine $\hat{\alpha}_L$. Due to the +45/-45 modulation of the groove walls, θ_{YZ} is approximately 45 degrees at all times. Note also that locally the surfaces are planar. To extract the left channel signal, we observe that the surface orientation lies in the plane $z = y$ because of the +45/-45 stereo modulation. Accordingly, the surface orientation corresponding to the right channel lies in the plane $z = -y$.

We define θ_L to be the modulation angle between the surface orientation $\hat{\alpha}$ and the left-channel-zero-modulation orientation $\hat{n}_L = [0, \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}]$, which is the surface orientation of the left groove wall when the signal is zero. Then the ratio of the lateral speed V_L and the tangential speed V_T of the stylus is:

$$\frac{V_L}{V_T} = \tan\theta_L \qquad (11)$$

where

$$\theta_L = \arccos(\hat{\alpha}_L \cdot \hat{n}_L). \qquad (12)$$

V_L corresponds to the left channel signal and needs to be adjusted according to $V_T = \omega R$, where R is the current distance to the record centre. A similar method can be applied to reproduce the right channel

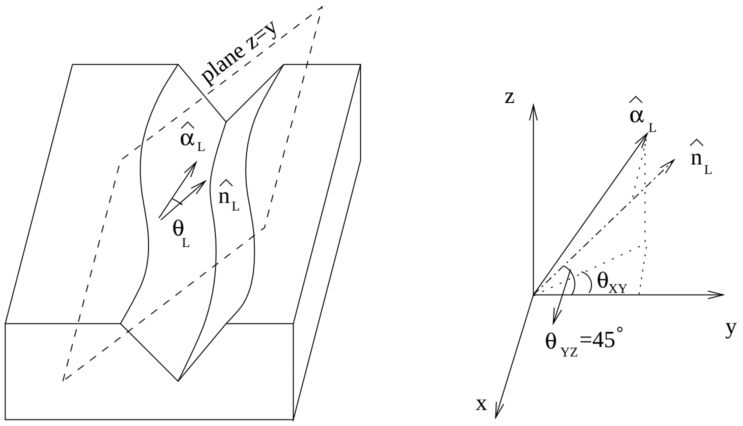


Figure 4: Illustration of a piece of groove showing the surface orientation $\hat{\alpha}_L$ of the left groove wall lies in the plane of $z = y$.

signal, V_R, using the direction $\hat{n}_R = [0, -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}]$ instead of $\hat{n}_L$. A mono recorded gramophone record with a horizontal groove modulation can be treated as a special case of the stereo groove modulation, where $V_L = V_R = V$, so either the left or right surface orientation can be used to reproduce the sound signal.

For a mono SP or a wax cylinder with vertical groove modulation, we can project the surface orientation onto the vertical $x - z$ plane, i.e. $y = 0$, and θ and V can be calculated using above equations, except now:

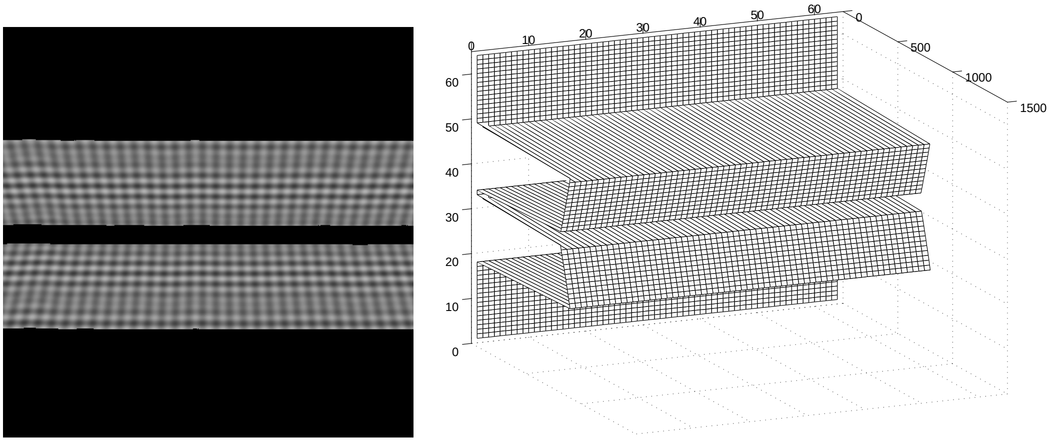
$$\hat{\alpha} = [\alpha_x, 0, \alpha_z] \text{ and} \tag{13}$$

$$\hat{n} = [0, 0, 1]. \tag{14}$$

Due to the robustness of the algorithm introduced in section 2.4, we anticipate that the algorithm will be able to reject most of the noise such as pops, clicks caused by scratches or small dirt particles, etc. However, comparing the waves in Fig 7b and Fig 7c indicate that post-processing may be needed to reduce the noise.

3 Simulation Technique

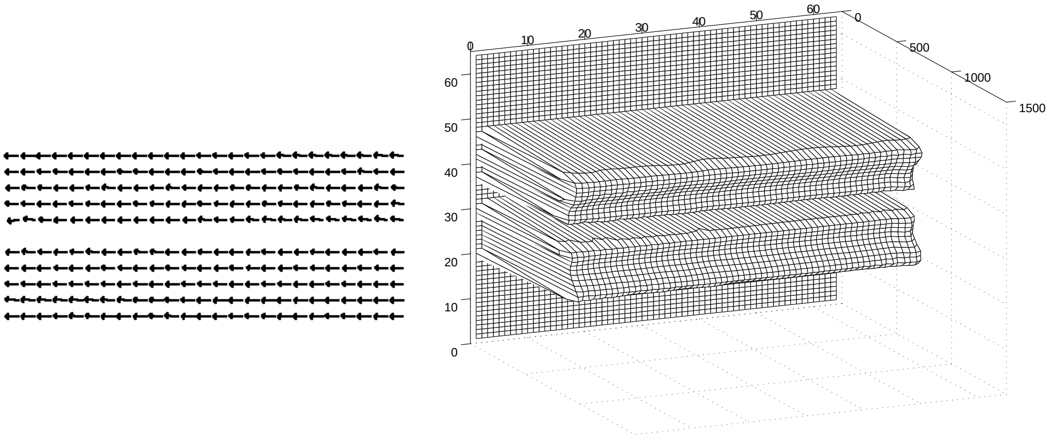
Implementation of our technique with real record data is currently underway. In this section, we report experimental results with synthetic data. We generated groups of ray-traced groove image sequences



(a) (b)

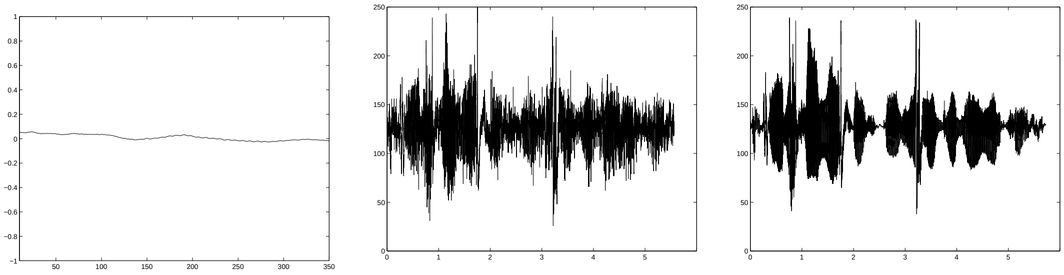
Figure 5: Synthetic test data: (a) A sinusoid-texture groove (b) The 3D perspective true depth map of the groove.

with the camera translating to the left by $(1, 0, 0)$, an example of which is shown in Figure 5. The offset of groove walls are modulated by a man’s voice. For about 2 second clip (“Computers are useless. They only give you answers” - Pablo Picasso) we generated 1390 groups of such images with each group having 36 images. From each group, we can recover a piece of the groove depth map and, hence, a piece of the sound. By ‘stitching’ these small pieces together, we obtain the complete sound recording. Optical flow was computed from these sequences of images as shown in Figure 6a. In this experiment, we used Horn and Schunck’s algorithm to obtain a smooth dense flow field. Next, we fed this flow sequence to Barron et al’s depth recovery algorithm, which incorporates a Kalman filter to compute a smooth surface reconstruction. The recovered depth map is shown in Figure 6b.



(a) (b)

Figure 6: Test results: (a) Optical flow computed from the synthetic image sequences. (b) The recovered depth map of the groove.



(a) (b) (c)

Figure 7: Sound waves: (a) The recovered sound wave from one piece of synthesised groove, (b) The recovered sound wave from groups of such images and (c) The original sound wave.

After the depth maps were computed, surface orientations of the grooves were then estimated. From the surface orientations, sound signals were computed as introduced in 2.5. The sound wave pieces such as those shown in Figure 7a were combined together to form the total sound wave as shown in Figure 7b. Compared with the original sound wave shown in Figure 7c, the reconstructed sound is very similar to the true sound, although there is some noise present, as can be seen in Figure 7b. We compute the shape envelopes of the computed and the original waveform and then compute Pearson’s product-moment correlation coefficient of them as $r = 0.848$ which indicates good correlation. Listening test confirms (available at www.csd.uwo.ca/˜btian/IMVIP2006) that the sound is recognisable in spite of the presence of noise. Further refinements in the algorithms and the use of higher resolution images may be able to attenuate the noisy components of the retrieved sound.

4 Conclusions

This paper established a framework for recovering sound from gramophone records through 3D reconstruction. This may not necessarily be a real-time system due to such limiting factors as the computation cost and camera speed, (although we believe technology advances will eventually overcome these limitations). Our algorithm has the potential of recovering sound from damaged records such as scratched or even broken records. We are investigating the feasibility of a real-time sound reproduction system, such as hardware implementations of the preprocessing steps, fast image acquisition, etc.

References

[Barron et al., 2003] Barron, J. L., Ngai, W. K. J., and Spies, H. (2003). Quantitative depth recovery from time-varying optical flow in a kalman filter framework. In T. Asano, R. K. and Ronse, C., editors, *LNCS 2616 Theoretical Foundations of Computer Vision: Geometry, Morphology, and Computational Imaging*, pages 344–355.

[Black and Anandan, 1996] Black, M. J. and Anandan, P. (1996). The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104.

[Cavaglieri et al., 2001] Cavaglieri, S., Johnsen, O., and Bapst, F. (2001). Optical retrieval and storage of analog sound recordings. In *The AES 20th International Conference*, Budapest, Hungary.

[ELP, 1997] ELP (1997). Elp laser turntable. Internet reference: www.elpj.com.

[Fadeyev and Haber, 2003] Fadeyev, V. and Haber, C. (2003). Reconstruction of mechanically recorded sound by image processing. *J. of Audio Eng. Soc.*, 51(12):1172–1185.

[Heel, 1990] Heel, J. (1990). Direct dynamic motion vision. In *Proc IEEE Conf. on Robot Automation*.

[Horn and Schunck, 1981] Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–204.

[Hung and Ho, 1999] Hung, Y. S. and Ho, H. T. (1999). A kalman filter approach to direct depth estimation incorporating surface structure. *IEEE PAMI*, pages 570–576.

[Lucas and Kanade, 1981] Lucas, B. D. and Kanade, T. (1981). An iterative image-registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130. DARPA.

[Matthies et al., 1989] Matthies, L., Szeliski, R., and Kanade, T. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *IJCV*, 3(3):209–238.

[Micrographia, 2005] Micrographia (2005). The microscopy of vinyl recordings. Internet reference, www.micrographia.com.

[Olsson et al., 2003] Olsson, P., Öhlin. R. Olofsson, D., Vaerlien, R., and Ayrault, C. (2003). The digital needle project - group light blue. Technical report, KTH Royal Institute of Technology, Stockholm, Sweden. Internet resource: www.s3.kth.se/signal/edu/projekt/students/03/lightblue/.

[Press et al., 1992] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1992). *Numerical Recipes in C*. Cambridge University Press, 2 edition.

[Simoncelli, 1994] Simoncelli, E. P. (1994). Design of multi-dimensional derivative filters. In *IEEE Int. Conf. Image Processing*, volume 1, pages 790–793.

[Springer, 2002] Springer, O. (2002). Digital needle - a virtual gramophone. Internet resource: www.cs.huji.ac.il/˜springer/.

[Stoddard, 1989] Stoddard, R. E. (1989). Optical turntable system with reflected spot position detection. United States Patent 4,870,631.

[Stoddard and Stark, 1989] Stoddard, R. E. and Stark, R. N. (1989). Dual beam optical turntable. United States Patent 4,870,631.

[Stotzer et al., 2003] Stotzer, S., Johnsen, O., Bapst, F., Milan, C., Sudan, C., Cavaglieri, S. S., and Pellizzari, P. (2003). Visualaudio: an optical technique to save the sound of phonographic records. *IASA Journal*, pages 38–47.

[Stotzer et al., 2004] Stotzer, S., Johnsen, O., Bapst, F., Sudan, C., and Ingol, R. (2004). Phonographic sound extraction using image and signal processing. In *Proc. ICASSP*, Montreal, Quebec, Canada.

[Tian and Barron, 2005] Tian, B. and Barron, J. L. (2005). A quantitative comparison of 4 algorithms for recovering dense accurate depth. In *2nd Canadian Conference on Computer and Robot Vision*, pages 498–505, Victoria, BC, Canada.

Synthetic Image Generation with Primes

Daniel C. Doolan Sabin Tabirca
Department of Computer Science
University College Cork, Ireland
{d.doolan, tabirca }@cs.ucc.ie

Abstract

The visualisation of Prime numbers has been a topic of interest for some time, the Ulam Spiral being the most famous. This paper introduces some new approaches for for the visualising of prime numbers. The method produces an image similar to that of the prime number spiral. The procedure by which this is created however, differs significantly as does the resultant image.

Keywords: Primes, Spiral, Visualise, 2D

1 Introduction

Prime Numbers has been a principle research topic of mathematicians for over 2,500 years. They were initially studied by the Greeks, Pythagoras being the most well known. By circa 300BC several important results about the nature of primes had been discovered. Euclid for example proved that the number of primes is infinite. One of the simplest and most well known algorithms for calculating primes was discovered circa 200BC by Eratosthenes and is aptly named the Sieve of Eratosthenes. However this algorithm is only of use when the number of primes to evaluate is small. Little else was discovered about primes until the 17th Century, Since then many important discoveries have been made by individuals such as Fermat, Mersenne, Euler, Gauss and Riemann.

1.1 Searching For Primes

Currently there is a search for a prime with 10 million digits or more, this is being carried out by the Great Internet Mersenne Prime Search (GIMPS) project. Since 2004 several large primes have been discovered (Table 1) and the goal of a 10 million digit prime number should be only months away.

#Primes	Rank	Prime	# Digits	Discoverer
15^{th} December 2005	43^{rd}	$2^{30,402,457}-1$	9,152,052	Dr's. C. Cooper and S. Boone
18^{th} February 2005	42^{nd}	$2^{25,964,951}-1$	7,816,230	Dr. Martin Nowak
15^{th} May 2004	41^{st}	$2^{24,036,583}-1$	7,235,733	Josh Findley

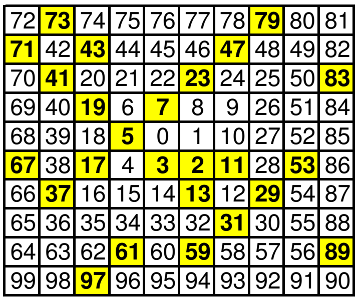
Table 1: Recently Discovered Mersenne Primes

1.2 Recent Developments in Primality

In August 2002 a paper titled "Primes in P" was published. It gave a deterministic polynomial-time algorithm that determines whether an input number n is prime or composite [Agrawal,] [M. Agrawal, 2004]. The paper was last updated in August 2005. The method produced in the 2002 paper became known as the AKS-type primality proofs. Several improvements to the algorithm were achieved during the following year by Berrizbeitia [Berrizbeitia, 2003] who was able to improve the time of the algorithm. This was improved by Cheng [Cheng,] [Cheng, 2003] and further again by Bernstein [Bernstein, 2004].

1.3 Prime Visualisations

The Ulam Spiral is the most well known method of visualising Prime Numbers graphically. The method was discovered by Stanislaw Ulam while doodling during a scientific meeting in 1963. As Ulam described the resultant image: it "appears to exhibit a strongly non-random appearance" [Stein et al., 1964]. He drew a grid of lines and then numbered the cells in a spiral pattern starting at the centre of the matrix. Circling the prime numbers (Figures 1 & 2) [Weisstein, 2006] led to a surprising result that the primes appeared to fall along a number of straight lines. The spiral appeared on the front cover of the magazine Scientific America in March 1964.



72	**73**	74	75	76	77	78	**79**	80	81
71	42	**43**	44	45	46	**47**	48	49	82
70	**41**	20	21	22	**23**	24	25	50	**83**
69	40	**19**	6	**7**	8	9	26	51	84
68	39	18	**5**	0	1	10	27	52	85
67	38	**17**	4	**3**	**2**	**11**	28	**53**	86
66	**37**	16	15	14	**13**	12	**29**	54	87
65	36	35	34	33	32	**31**	30	55	88
64	63	62	**61**	60	**59**	58	57	56	**89**
99	98	**97**	96	95	94	93	92	91	90

Figure 1: Ulam Spiral

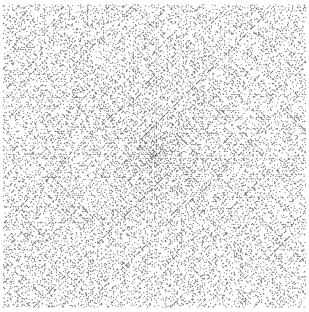


Figure 2: Ulam Spiral of 399 x 399 grid

An alternative to the Ulam Spiral is the Prime Spiral, see: numberspiral.com [Sacks,]. The procedure for creating the spiral is straight forward. Starting at the number zero, all subsequent positive numbers are arranged in an outward spiral from the centre. The primary feature of this spiral is that all the perfect squares (1, 4, 9, 16, 25 . . .) are arranged in a horizontal line on the right hand side of the graph. As with many prime visualisations highlighting the numbers that are prime produces some interesting patterns (Figures 3 & 4 [Sacks,]). The resultant image has strong links with that of the Ulam Spiral.

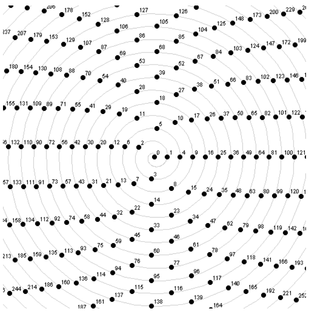


Figure 3: Labelled Number Spiral

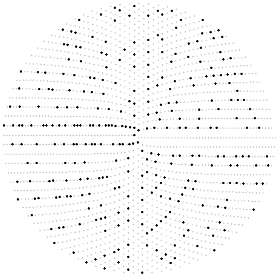


Figure 4: Number Spiral Displaying Primes

1.4 Motivation

Primes have been a topic of study for thousands of years, and yet we still are unable to determine if a number is prime, by carrying out a simple calculation. The Ulam spiral demonstrated a distinctive pattern of diagonal lines in the plotted primes. It is possible that the visualisation of primes may hold the key to this ancient mystery. The visualisation of primes has led to several important discoveries, most notable was the discovery of the Zeros by Riemann. Hence the need for further developments within the area of prime number visualisation. The primary purpose of this paper is to review some recent developments in Prime Number Fractal generation. It also details a new method of visualising primes using geometrical shapes as the corner stone.

2 Prime Number Fractal

The Prime Number Fractal (Figure 5) generates movement within the image based on the evaluation of the prime numbers. As with most prime visualisations the generation of the primes for the image bears the highest computation expense. On modern desktop systems using the Sieve of Eratosthenes it can take several seconds just to generate just a few million prime numbers. The UI of the Java application developed to generate the PNF image may be seen in Figure 6.

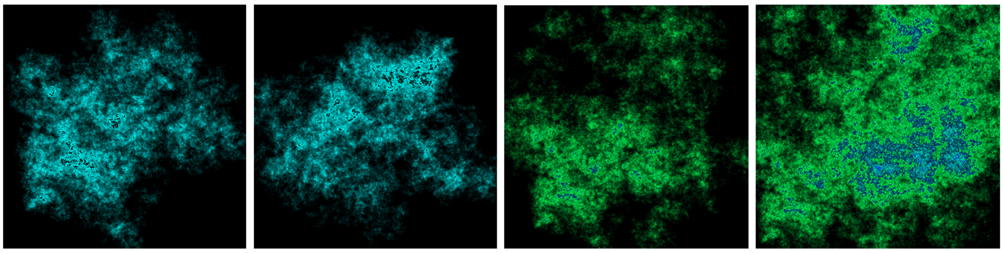
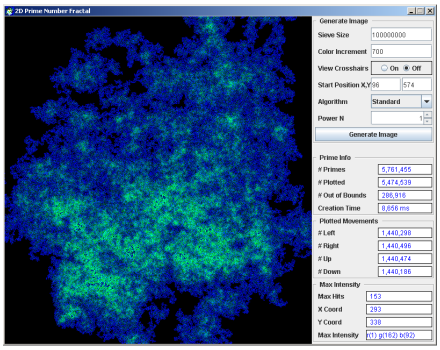


Figure 5: 2D Prime Number Fractals

```
procedure pnf_fractal
for (i=1; i<sieveSize; i++){
         if (isPrime(i)){
         dir = prime %5;
         if(dir ==1) x --
         if(dir ==2) x ++
         if(dir ==3) y --
         if(dir ==4) y ++
         incrementColour(x,y)
       }
     }
end procedure;
```

Listing 1: PNF Generation Procedure



Sieve Size 100000000
Color Increment 700
View Crosshairs ○ On ● Off
Start Position X,Y 96 574
Algorithm Standard
Power N 1

Primes 5,761,455
Plotted 5,474,539
Out of Bounds 286,916
Creation Time 8,656 ms

Left 1,440,298
Right 1,440,496
Up 1,440,474
Down 1,440,186

Max Hits 153
X Coord 293
Y Coord 338
Max Intensity r(1) g(162) b(92)

Figure 6: UI and Generated Fractal Image

#Primes	20,000	40,000	60,000	80,000	100,000
Left	4,978	10,003	14,971	19,985	24,967
Right	5,023	10,013	15,020	20,007	25,016
Up	5,011	9,997	15,018	20,031	25,007
Down	4,987	9,986	14,990	19,976	25,009

Table 2: Distribution of PNF Movements

The primary methodology behind the generation of the PNF visualisation is based on modifying the next cell that should be visited based on a calculation performed on the next prime number (Listing 1). In the case of a two dimensional image modular division of the prime number by five will yield four possible outcomes $\in \{1, 2, 3, 4\}$. Modular division by seven will yield six possible moves, yielding a 3D visualisation. By mapping these moves to a direction it is possible to navigate about the initial image based on the values of a sequence of primes. Previous work has shown that the number of moves for both 2D and 3D visualisations are asympotitically equal (Table 2) [Doolan et al., 2004].

Dirichlet's theorem assures if a and b are relatively prime then there are an infinity of primes in the set $a \cdot k + b, k > 0$. This means that the random walk has an infinity of Up, Down, Left, Right moves. If $\pi_{a,b}(x)$ denotes the number of primes of the form $a \cdot k + b$ less than x then we know from a very recent result of Weisstein [Weisstein, 2004] that $\lim_{x\to\infty} \frac{\pi_{a,b}(x)}{li(x)} = \frac{1}{\varphi(a)}$ where $li(x)$ is the logarithmic integral function and $\varphi(a)$ the Euler totient function. The particular case $a = 5$ gives $\lim_{x\to\infty} \frac{\pi_{5,k}(x)}{li(x)} = \frac{1}{\varphi(5)} = \frac{1}{4}, \forall k \in 1, 2, 3, 4$, which means that $\pi_{5,1} \approx \pi_{5,2} \approx \pi_{5,3} \approx \pi_{5,4}(x) \approx \frac{li(x)}{4}$ clearly the two dimensional prime number fractal algorithm has asymptotically the same number of Up, Down, Left and Right moves.

3 Using Geometrical Shapes to Visualise Primes

This section introduces a new way to visualise primes which is based on regular polygons. It is a new variation of spiral visualisation that is introduced in [Sacks,] in which the spiral is replaced by some other curves. For that an application for generating the new visualisation of primes via geometrical shapes is developed in Java. It provides a very simple to use User Interface (UI) to modify the parameters for the image generation process. The principle and most prevalent section of the application is the Panel for rendering the graphic visualisation (Figure 7).

Algorithm 1 The Geometrically Based Spiral Algorithm

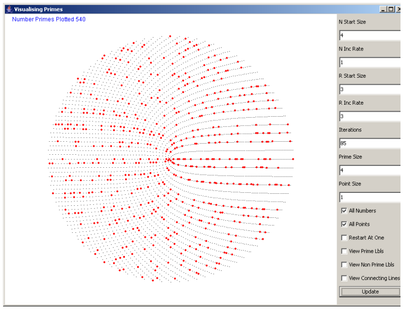
```
Inputs:
    $x0, y0$ - The central point of the image
    R - The Initial Radius
    Step - The step to increment R
    nrP - The of Polygons
Output: The Geometrically Based Spiral Image

procedure GBSA($x0$, $y0$, R, step, nrP)
    label = 1;
    for(n = 0; n< nrP; n++) begin
        for(i = 0; i< nr[n]; i++) begin
            if(isPrime(label)) begin
                $x_0$[i] = $x0 + R \times cos(2\pi \times i/nr)$;
                $y_0$[i] = $y0 + R \times sin(2\pi \times i/nr)$;
                drawPixel($x_0$[i], $y_0$[i]);
            end if
            label ++;
        end for
        R = R + step; nr++;
    end for
end procedure
```

The application parameters allow for control of the number of points that should be used in the first geometric shape at the centre of the image. The number of points in each subsequent polygon increases at a set rate defined by the "N Inc Rate" parameter which is usually 1. Similar parameters also exist for the radius into which the polygon is to be inscribed, and the rate of growth. The number of polygons to produce is controlled by the "Iteration" value. The size of both prime and non prime are also controlled by user defined parameters. Finally a set of checkboxes provide additional aids.

The "All Numbers" option allows the algorithm (Algorithm 1) to go through all numbers from 1 to n in a incremental fashion. Deselecting this option has the effect that starting at 1 only the odd numbers will be taken into account. The remaining checkboxes provide additional visual aid to the user. View or hide all points (non prime points). Restart the generation process at 1 for each new polygon. View the related numbers for each point on the image (both prime and non prime). The final option displays the connections between points with lines, allowing the geometrical shape to be discernible.

3.1 Polygon Generation

The key feature to this visualisation is the generation of polygons. Starting at the centre of the image and growing outwardly until the desired number of iterations has been completed. Both the size and number of points for the initial polygon are user defined. Initialised with a polygon of 4 points a Square or Diamond shape is produced. The next iteration will generate a Pentagon, followed by a Hexagon, Heptagon, Octagon . . . DoDecagon . . . and Hectogon after the 97^{th} iteration (Figure 8). By plotting the points (primes) of the polygons distinctive arcs may be seen in the image.



Number Primes Plotted 540

N Start Size: 4

N Inc Rate: 1

R Start Size: 3

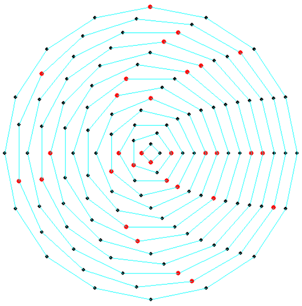
R Inc Rate: 3

Iterations: 85

Prime Size: 4

Point Size: 1

☑ All Numbers
☑ All Points
☐ Restart At One
☐ View Prime Lbls
☐ View Non Prime Lbls
☐ View Connecting Lines



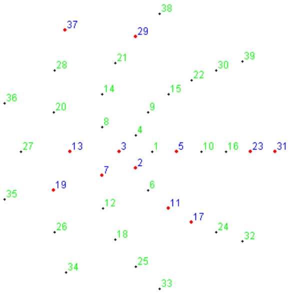


Figure 7: Visualisation Application

Figure 8: Example of Polygon Creation

Figure 9: Plotting of Polygon Points

The procedure by which the numbers are plotted, begins the generation process of each polygon at on the same horizontal axis projecting to the right of the image. From this starting position each consecutive point is generated in a clockwise manner (Figure 9). Lets suppose that the n^{th} regular polygon has nr_n points and the radius R_n. The position of the i^{th} point is calculated using

$$x_n[i] = x0 + R_n \cdot \cos\left(\frac{2 \cdot \pi \cdot i}{nr_n}\right)$$

$$y_n[i] = y0 + R_n \cdot \sin\left(\frac{2 \cdot \pi \cdot i}{nr_n}\right), i = 0, 1, ..., nr_n - 1.$$

The Java application increments the parameter R_n with a constant $step$ while nr_n increases usually by 1 so that their values are in arithmetic progression.

3.2 Some Mathematics

Some mathematical considerations are presented in the following to support this model. For that lets suppose that $nr_n = 4 + n, R_n = R_0 + n \cdot step, n \geq 0$. The curve C_i collects all the points $\{(x_n[i], y_n[i]), n + 4 > i\}$ so that the parametric equation of it is

$$C_i : \left(\left(x0 + (R_0 + t \cdot step) \cdot \cos\frac{2\pi i}{t+4}, y0 + (R_0 + t \cdot step) \cdot \sin\frac{2\pi i}{t+4}\right), t > i - 4\right). \tag{1}$$

In this case the prime numbers are not aligned on strait lines but on concave curves.

Theorem 1 *The curve C_i asymptotically converges to $y = y0 + 2 \cdot \pi \cdot i \cdot step$. The next theorem proves that the curves tend to some horizontal lines.*

Proof.

$$\lim_{t\to\infty} y_t[i] = \lim_{t\to\infty}\left(y0 + (R_0 + t \cdot step) \cdot \sin\frac{2\pi i}{t+4}\right) =$$

$$= \lim_{t\to\infty}\left(y0 + 2 \cdot \pi \cdot i \cdot \frac{\sin\frac{2\pi i}{t+4}}{\frac{2\pi i}{t+4}} \cdot \frac{R_0 + t \cdot step}{t+4}\right) = y0 + 2 \cdot \pi \cdot i \cdot step.$$

Proposition 2 *The i^{th} node of the n^{th} regular polygon is labelled with $\frac{(n+6)(n-1)}{2} + i$.*

Proof. The first $n-1$ regular polygons have the labels $4 + 5 + ... + (n+2) = \frac{(n+6)(n-1)}{2}$ so that the i^{th} node of the n^{th} regular polygon has got the label $\frac{(n+6)(n-1)}{2} + i$.

Theorem 3 *The curve C_i contains the following labels $\{\frac{(n+6)(n-1)}{2} + i,\ n > i - 4\}$.*

Practical experiments have shown that each curve contains lots of primes. Some recent developments by Einsiedler et.al. [Einsiedler et al., 2000], [Everest et al., 2006] investigate how many primes are in recursive sequences. In our particular case the curve C_i contains the labels from the sequence $q_n = \frac{(n+7)n}{2} + i, n > 0$. Obviously, this sequence satisfies the following linear recurrence $q_{n+2} = 2q_{n+1} - q_n + 1,\ q_0 = i, q_1 = i + 4$ so that similar arguments as in [Everest et al., 2006] can be applied to prove that the sequence contains an infinite number of primes.

The next result finds information about the regular polygon that contains a label.

Theorem 4 *The label l is contained by the n^{th} regular polygon on the i^{th} node where*

$$n = \left\lceil\frac{-7 + \sqrt{49 + 8 \cdot l}}{2}\right\rceil,\ \ i = l - \frac{(n+6)(n-1)}{2}.$$

Proof. The label l is contained by n^{th} regular polygon when

$$4 + 5 + ... + (n+2) < l \leq 4 + 5 + ... + (n+2) + (n+3) \Leftrightarrow$$

$$\frac{(n+6)(n-1)}{2} < l \leq \frac{(n+7)n}{2} \Leftrightarrow (n+7)n \geq 2 \cdot l \wedge (n+6)(n-1) < 2 \cdot l \Leftrightarrow$$

$$n^2 + 7n - 2l \geq 0\ \wedge\ (n-1)^2 + 7(n-1) - 2l < 0 \Leftrightarrow$$

$$n \geq \frac{-7 + \sqrt{49 + 8 \cdot l}}{2}\ \wedge\ n < \frac{-7 + \sqrt{49 + 8 \cdot l}}{2} + 1 \Leftrightarrow\ n = \left\lceil\frac{-7 + \sqrt{49 + 8 \cdot l}}{2}\right\rceil.$$

Testing primality is done by using the Sieve of Eratosthenes as we know that the labels that are prime should be found. If p is the number of regular polygons to use in this visualisation then the labels to use are $\{1, 2, ..., \frac{(p+7)p}{2}\}$. The prime labels are identified using the sieve computation which has the complexity $O(p^2 \cdot \log\log p)$ and uses $O(p^2)$ space.

3.3 Resultant Visualisations

Generating the image for all numbers between 1 and n incrementing by 1 for each point generated will generate an image such as Figure 10 (80 polygons, with 4 starting points and radius of 3 for the first polygon, displaying only the primes). Increasing the rate of increment in the number of points per polygon from one to the next will also have s significant impact on the density of the primes plotted (almost doubling)(Figure 11).

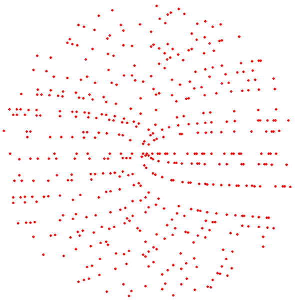


Figure 10: Prime Spiral with 80 Polygons

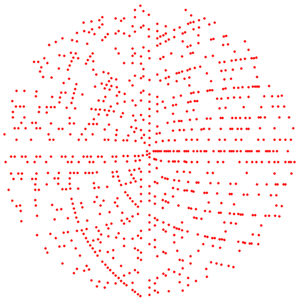


Figure 11: Prime Spiral with 80 Polygons Increment Rate of Two

An alternative to the above image is to significantly increase the start radius. This will open up the central area of the image (Figure 12) (200 polygons, with 4 starting points and radius of 80 for the first polygon, displaying only the primes).

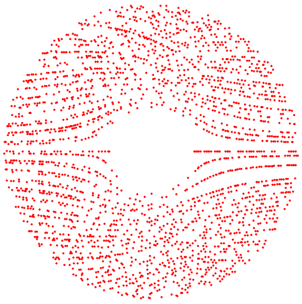


Figure 12: Prime Spiral with 200 Polygons

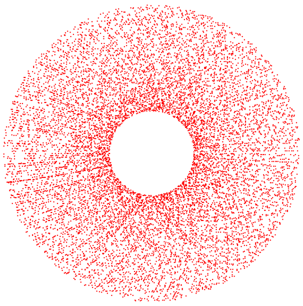


Figure 13: Prime Spiral with 500 Starting Points

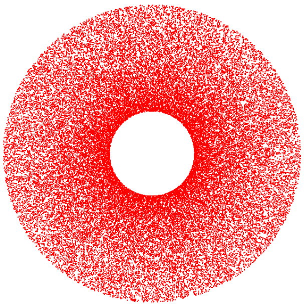


Figure 14: Prime Spiral with 2,500 Starting Points

Increasing the number of points present in the starting polygon can have a significant effect on the resultant image generated. The most prevalent outcome of this is that the number of primes (and polygon points) generated increases dramatically. This generates a far denser field of primes (Figure 13) (200 polygons, with 500 starting points and radius of 80 for the first polygon, displaying only primes). Increasing the starting number of points to 2,500 shows a huge increase in the density of primes per polygon, and the entire image itself (Figure 14).

All the previous figures showed primes being plotted for all numbers from 1 through to n. However plotting the primes for only the odd numbers, produces an image with some variations in the generated arcs, and also a doubling in density of the number of primes plotted (with 80 polygons being generated) (Figure 15).

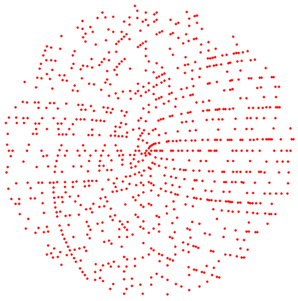


Figure 15: Prime Spiral Only Odd Values Being Used

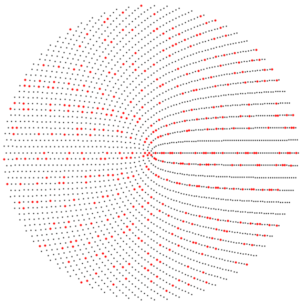


Figure 16: Displaying Both Prime and Non Prime Points

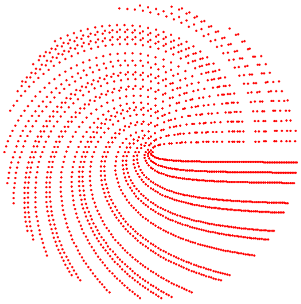


Figure 17: Restart at One

The final figure in the above series gives an example image of the plotting of both the prime and non prime points (Figure 16). The image was plotted with 70 polygons.

All prime number visualisations typically start with 1 and iterate through up to n. An alternative to this is that for each new polygon produced, the starting number is reset to 1. The effect of this is the generation of spiral sea shell like image (Figure 17).

4 Conclusion

A new method of visualising primes has been discussed. It is clear that several differing visualisation may be formed by varying the input parameters to the generation function. Initialisation of the initial polygon with a large number of points results in a highly dense field of points being plotted. This contrasts with a far more sparse distribution when the initial number of points for the starting polygon are low. The overall result still produces a pattern that follows several distinctive arcs. Several theorems and proofs have been presented that clearly identify the primes lie along specific arcs and lines.

References

[Agrawal,] Agrawal, M. Website of manindra agrawal. `http://www.cse.iitk.ac.in/users/manindra/`.

[Bernstein, 2004] Bernstein, D. J. (2004). Primality in essentially quartic random time. `http://cr.yp.to/papers.html`.

[Berrizbeitia, 2003] Berrizbeitia, P. (2003). Website of p berrizbeitia. `http://arxiv.org/abs/math.NT/0211334`.

[Cheng,] Cheng, Q. Website of qi cheng. `http://www.cs.ou.edu/~qcheng/`.

[Cheng, 2003] Cheng, Q. (2003). Primality proving via one round in ecpp and one iteration in aks. *Crypto 2003*.

[Doolan et al., 2004] Doolan, D. C., Tabirca, S., and Murphy, D. (2004). 3d prime number fractals. In *Eurographics Irish Chapter Workshop*.

[Einsiedler et al., 2000] Einsiedler, M., Everest, G., and Ward, T. (2000). Primes in sequences associated to polynomials (after lehmer). In *LMS J. Comput. Math. 3*, pages 125–139.

[Everest et al., 2006] Everest, G., Stevens, S., Tamsett, D., and Ward, T. (2006). Primes generated by recurrence sequences. In *American Math Monthly (to appear)*.

[M. Agrawal, 2004] M. Agrawal, N. Kayal, N. S. (2004). Primes in p. *Annals of Mathematics 160(2)*, pages 781–793.

[Sacks,] Sacks, R. Numberspiral.com. `http://www.numberspiral.com`.

[Stein et al., 1964] Stein, M., Ulam, S., and Wells, M. (1964). A visual display of some properties of the distribution of primes. *American Math Monthly 71*, pages 43–44.

[Weisstein, 2004] Weisstein, E. W. (2004). Arbitrarily long progressions of primes. `http://mathworld.wolfram.com/news/2004-04-12/primeprogressions/`.

[Weisstein, 2006] Weisstein, E. W. (2006). Prime spiral from mathworld - a wolfram web resource. `http://mathworld.wolfram.com/PrimeSpiral.html`.

Active Vision, Tracking & Motion Analysis

Unsupervised Camera Motion Estimation and Moving Object Detection in Videos

Rozenn Dahyot
School of Computer Science and Statistics
Trinity College Dublin, Ireland
https://www.cs.tcd.ie/Rozenn.Dahyot/
Rozenn.Dahyot@cs.tcd.ie

Abstract

In this article, we consider the robust estimation of a location parameter using M-estimators. We propose here to couple this estimation with the robust scale estimate proposed in [Dahyot and Wilson, 2006]. The resulting procedure is then completely unsupervised. It is applied to camera motion estimation and moving object detection in videos. Experimental results on different video materials show the adaptability and the accuracy of this new robust approach.

Keywords: M-estimation, camera motion, moving object detection, robust estimation, video analysis

1 Introduction

Many problems in computer vision involve the separation of a set of data into two classes, one of interest in the context of the application and the remaining one. For instance, edge detection in images requires the thresholding of the gradient magnitude to discard noisy flat areas from the edges. The challenge is then to automatically select the appropriate threshold [Rosin, 1997].

Regression problems also involve the simultaneous estimation of the variance or standard deviation of the residuals/errors. The presence of a large number of outliers makes difficult the estimation of the parameters of interest. Performance of robust estimators is highly dependent on the setting of a threshold or scale parameter, to separate the good data (inliers) that fit the model, from the gross errors (outliers) [Chen and Meer, 2003]. The scale parameter, needed in M-estimation and linked to the scale parameter of the inliers residuals, is often set a priori or estimated by the Median Absolute Deviation. In [Dahyot and Wilson, 2006], a robust non-parametric estimation for the scale parameter has been proposed and then combined with a robust RANSAC [Fischler and Bolles, 1981] for object recognition. This paper proposes to combine the robust scale parameter estimation with a M-estimation of the camera motion parameter in videos. The whole scheme is unsupervised. The estimated scale parameter is also used to detect moving objects in the sequences.

2 Robust scale estimation

2.1 Observations

The observations consist in a set of independent samples $\{x_i\}$ of a random variable X. Its probability density function can be written as a mixture:

$$\mathcal{P}_X(x|\sigma,\theta) = \mathcal{P}_X(x|\sigma,\theta,\mathcal{C}) \cdot \mathcal{P}_X(\mathcal{C}) + \mathcal{P}_X(x|\overline{\mathcal{C}}) \cdot \mathcal{P}_X(x|\overline{\mathcal{C}}) \tag{1}$$

with the *pdf* $\mathcal{P}_X(x|\sigma,\theta,\mathcal{C})$ corresponding to a particular class $\mathcal{C}$ of interest (inliers) that depends onto one scale parameter σ and possibly also on a location parameter θ. The other *pdf* $\mathcal{P}_X(x|\overline{\mathcal{C}})$ in the mixture is generated by possible outliers occurring in the observations (class $\overline{\mathcal{C}}$) and the parameters of interest σ and θ do not depend on those outlying observations. $\mathcal{P}_X(\mathcal{C})$ is the proportion of inliers and $\mathcal{P}_X(\overline{\mathcal{C}})$ is the proportion of outliers.

In this work, we assume the distribution of the inliers to be a Generalized centred Gaussian [Aiazzi et al., 1999]:

$$\begin{array}{l} \mathcal{P}_X(x|\mathcal{C},\sigma,\theta) = \frac{1}{2\Gamma(\alpha)\cdot\alpha\cdot\beta^\alpha} \exp\left[\frac{|x(\theta)|^{1/\alpha}}{\beta}\right] \\ \text{with } \beta = \sigma^{1/\alpha}\cdot\left[\frac{\Gamma(\alpha)}{\Gamma(3\alpha)}\right]^{1/(2\alpha)} \end{array} \tag{2}$$

Setting the shape parameter $\alpha = 1$ (Laplacian law) and $\alpha = 1/2$ (Gaussian law) in equation (2), are two popular hypotheses [Hasler et al., 2003, Dahyot et al., 2004]. We assume α is known and focus on the estimation of the scale σ and θ.

2.2 Robust scale estimation knowing the location parameter θ

We now assume that we have $\mathbf{n}$ independent variable X_n of the same nature of the X previously defined. Samples for each X_n can be easily obtained for instance by splitting the original sample set of samples $\{x_i\}$ into $\mathbf{n}$ sets. Depending on the applications, the $\mathbf{n}$ random variables can also be naturally defined (see section 3). We define the variables:

$$\begin{cases} Z & = \sum_{n=1}^{\mathbf{n}} |X_n|^{1/\alpha} \\ Y & = Z^\alpha \end{cases} \tag{3}$$

Inliers of Z and Y (in class $\mathcal{C}$) are the samples z_j or y_j computed with $\mathbf{n}$ independent samples of X such that $\forall i, x_i \in \mathcal{C}$. For $\mathbf{n} = 1$ and $Z = |X|^{1/\alpha}$, the *pdf* $\mathcal{P}_Z(z|\theta,\sigma,\mathcal{C})$ corresponds to the gamma distribution:

$$\mathcal{P}_Z(z|\theta,\sigma,\mathcal{C}) = \mathcal{G}_{Z|(\alpha,\beta)}(z) = \frac{z^{\alpha-1}}{\Gamma(\alpha)\cdot\beta^\alpha}\exp\left[-\frac{z}{\beta}\right], \quad z \geq 0 \tag{4}$$

When $\mathbf{n} > 1$, the *pdf* $\mathcal{P}_Z(z|\mathcal{C},\sigma)$ is the gamma function $\mathcal{G}_{Z|(\mathbf{n}\alpha,\beta)}(z)$, and the *pdf* of Y can easily be inferred [Dahyot and Wilson, 2006]. The maximum of the distributions $\mathcal{P}_Z(z|\mathcal{C},\sigma)$ and $\mathcal{P}_Y(y|\mathcal{C},\sigma)$ can be then computed:

$$\begin{cases} Z_{\max\mathcal{C}} = \beta\cdot(\mathbf{n}\alpha - 1), \ \mathbf{n}\alpha > 1 \\ Y_{\max\mathcal{C}} = [(\mathbf{n}-1)\ \alpha\ \beta]^\alpha, \ \mathbf{n} > 1 \end{cases} \tag{5}$$

Those maxima depend on the parameter σ by definition of β (cf. eq. (2)). From equation (5), the scale σ can be computed by:

$$\begin{cases} \sigma_Z = \left(\frac{Z_{\max\mathcal{C}}}{\mathbf{n}\alpha-1}\right)^\alpha \left[\frac{\Gamma(3\alpha)}{\Gamma(\alpha)}\right]^{1/2}, & \mathbf{n}\alpha > 1 \\ \sigma_Y = \frac{Y_{\max\mathcal{C}}}{(\mathbf{n}-1)^\alpha\cdot\alpha^\alpha} \left[\frac{\Gamma(3\alpha)}{\Gamma(\alpha)}\right]^{1/2}, & \mathbf{n} > 1 \end{cases} \tag{6}$$

The maximum of the distributions of Y and Z has first to be located. Depending on the proportion and the values of the outliers, the localisation of the maximum needed in the estimation gets more difficult. We assume that the relevant maximum for the estimation is the closest peak to zero in the distributions $\mathcal{P}_Y(y|\sigma,\theta)$ and $\mathcal{P}_Z(z|\sigma,\theta)$.

2.3 Computation of the scale estimate in practice

The scale estimates are computed using the meanshift procedure on the set of samples of variables Y or Z, starting from the minimum sample value (or starting from zero). More details are presented in [Dahyot and Wilson, 2006]. However, in some signal processing applications, the digitised signal is discrete with known quantized levels in a finite domain. For instance, pixel values in video data are integers in $[0; 255]$. Most of all, the variable Z (or Y) has its values in a one-dimensional space. Therefore, as an alternative to the kernel representation of the distribution and the Mean Shift algorithm to perform the estimation, standard histograms can be easily used and their derivatives easily computed using filters. This is another practical and faster way to perform the estimation when dealing with a digitised signal. Both Y and Z perform similarly a robust scale estimation (see [Dahyot and Wilson, 2006]).

3 Applications

Section 3.1 presents an experiment for unsupervised moving object detection in static sequences. The variable Z is used for the scale estimation, there is no location parameter θ (the camera motion is null) and the shape parameter is chosen $\alpha = 1$. Section 3.1 extends those preliminary results to camera motion estimation and moving object detection. The variable Y is used for the scale estimation, the location parameter θ corresponds to a 6-dimensional camera motion vector and the shape parameter is chosen $\alpha = 1/2$.

3.1 Application to moving object detection in static camera sequences

3.1.1 Using colour video data $\mathbf{n} = 3$

We are considering two different colour images $I_t = (R_t, G_t, B_t)$ and $I_{t'} = (R_{t'}, G_{t'}, B_{t'})$ from a video sequence. The samples of the random variables X_n for $n \in \{1, 2, 3\}$ are computed as the inter-frame difference on each colour band for each position i of the pixel:

$$\begin{cases} X_1 : \; x_i^{(1)} = R_{t'}(i) - R_t(i) \\ X_2 : \; x_i^{(2)} = G_{t'}(i) - G_t(i) \\ X_3 : \; x_i^{(3)} = B_{t'}(i) - B_t(i) \\ Z : \; z_i = |x_i^{(1)}| + |x_i^{(2)}| + |x_i^{(3)}| \end{cases} \tag{7}$$

The distribution of Z has been drawn using a histogram over the samples $\{z_i\}$ in figure 1(a). The estimated distribution of the inliers $\mathcal{P}_Z(z|\mathcal{C}, \sigma_Z)$ is also superimposed (with a rescaling factor to match the maxima).

3.1.2 Using grey level video data $\mathbf{n} = 2$

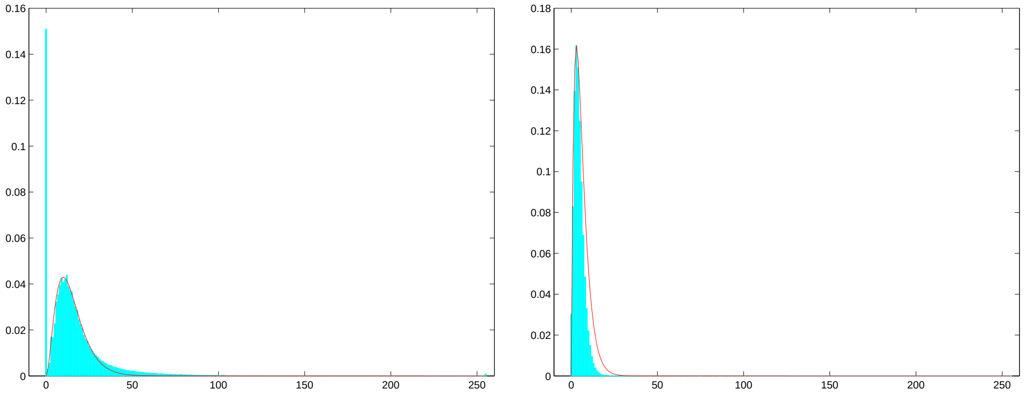
When the sequence is grey level, the samples of the variable Z can be computed using the backward and forward inter-frame differences:

$$\begin{cases} X_1 : \; x_i^{(1)} = I_{t+1}(i) - I_t(i) \\ X_2 : \; x_i^{(2)} = I_t(i) - I_{t-1}(i) \\ Z : \; z_i = |x_i^{(1)}| + |x_i^{(2)}| \end{cases} \tag{8}$$

Figure 1 (b) shows the distribution of Z in this case.

3.1.3 Results

When comparing images from a video, the interframe differences contain outliers due to camera and object motion. However in most applications, it can be assumed that a sensible proportion of pixels are matching. This proposed scheme for standard deviation estimation can be



(a) $\mathbf{n} = 3$ (b) $\mathbf{n} = 2$

Figure 1: Distribution $\mathcal{P}_Z(z)$ (blue bars) with the fitted distribution of the inliers $\mathcal{P}_Z(z|\mathcal{C}, \sigma_Z)$ on real video data for inter-frame difference analysis.

used on the inter-frame differences in order to separate or locate the outliers in the observations. We are considering in this example a video recorded from a static camera. Only moving objects in the scene generate outliers. By setting a threshold using the estimated standard deviation, the moving objects are located using the decision rule $z_i > \mathbf{n}\ 3\ \sigma$ for each pixel position i.

Figure 2 shows an example of the moving object detection process. The inter-frame difference is computed between the median frame of the video as a model of the background model of the scene, and the frame at time $t = 100$. Segmentation of the movement is not perfect as it is only based on the pixel statistics. Some pixels from the moving objects have similar values as the background ones at the same location, therefore their differences are classified as inliers (un-moving regions). However, it is a simple method to roughly locate moving regions which, in this example, are objects of interest such as pedestrians and cars for a traffic surveillance application. A result on a grey level sequence is shown in figure 3.

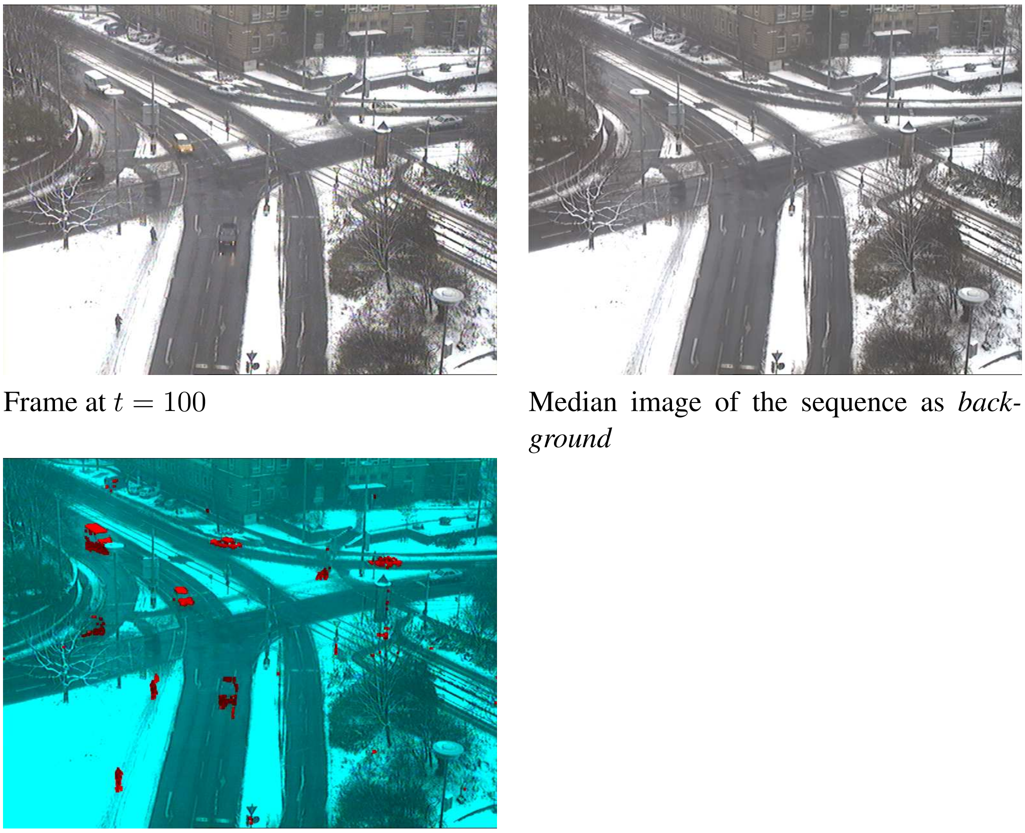
3.2 Application to unsupervised camera motion estimation and moving object detection in moving camera sequences

We consider in this section the problem of camera motion estimation from video data. Camera motion estimation has many applications such as video restoration and content analysis [Bouthémy et al., 1999, Kokaram et al., 2003, Coldefy et al., 2004]. The motion parameters are 6-dimensional to take into account zoom, rotation and translation. The residuals $\{x_i\}$, corresponding to the displaced frame difference (*dfd*), are linearly depending on the camera motion parameters θ. Figure 5 shows two successive images in a sequence and their difference, respectively before and after camera motion compensation. Those two observations correspond to the residuals observed at the first and the last iteration of our robust estimation. Images of videos used for testing are shown in fig. 4.

3.2.1 Iterative estimation of the scale and location parameters

For each iteration, we estimate the scale parameter σ_Y on the set of residuals and then perform the estimation of the motion parameters until convergence:

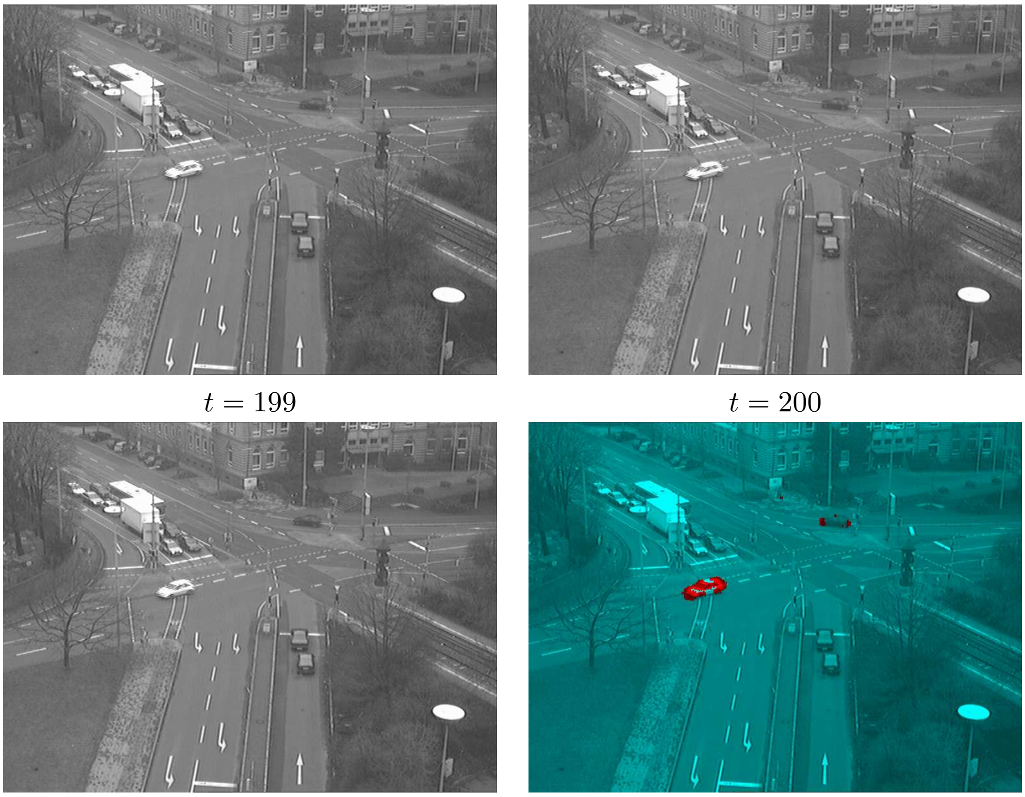
$$\begin{array}{l}
\text{Initialisation } \theta^{(0)} \\
\text{Repeat :} \\
\left| \begin{array}{l}
\sigma_\rho^{(m)} \leftarrow \text{Scale estimation on } \{y_i\} \\
\qquad \text{computed from } \{x_i(\theta^{(m))}\} \\
\\
\theta^{(m+1)} = \arg\min \left\{ \sum_i \rho \left(\frac{x_i(\theta^{(m)})}{\sigma_\rho^{(m)}} \right) \right\}
\end{array} \right. \\
\text{Until convergence } \hat{\theta}\ \hat{\sigma}_\rho
\end{array} \tag{9}$$



Frame at $t = 100$

Median image of the sequence as *background*

Figure 2: Colour sequence. Detection of moving objects (in red) based on the statistics of the difference of the colour pixels with the median image of the sequence (sequence *dtneu_winter*).



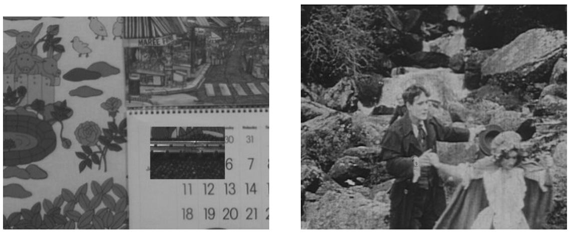
$t = 199$

$t = 200$

$t = 201$

Moving objects at $t = 200$

Figure 3: Grey-level sequence. Detection of moving objects in between successive frames, based on the statistics of the difference of grey level pixels(sequence *dt_passat03*).



(a) (b)

Figure 4: Videos used in the test. (a) is a simple video where the camera motion is known, and the dark rectangle is the only (outlier) object that moves differently from the background. (b) is a shot from an old film where several artifacts occur (blotch, flicker, sporadic and severe vertical displacement, etc).

The location parameter estimation is performed using M-estimation with a robust function ρ [Dahyot and Kokaram, 2004]. The initial guess $\theta^{(0)}$ is estimated using non-robust least squares.

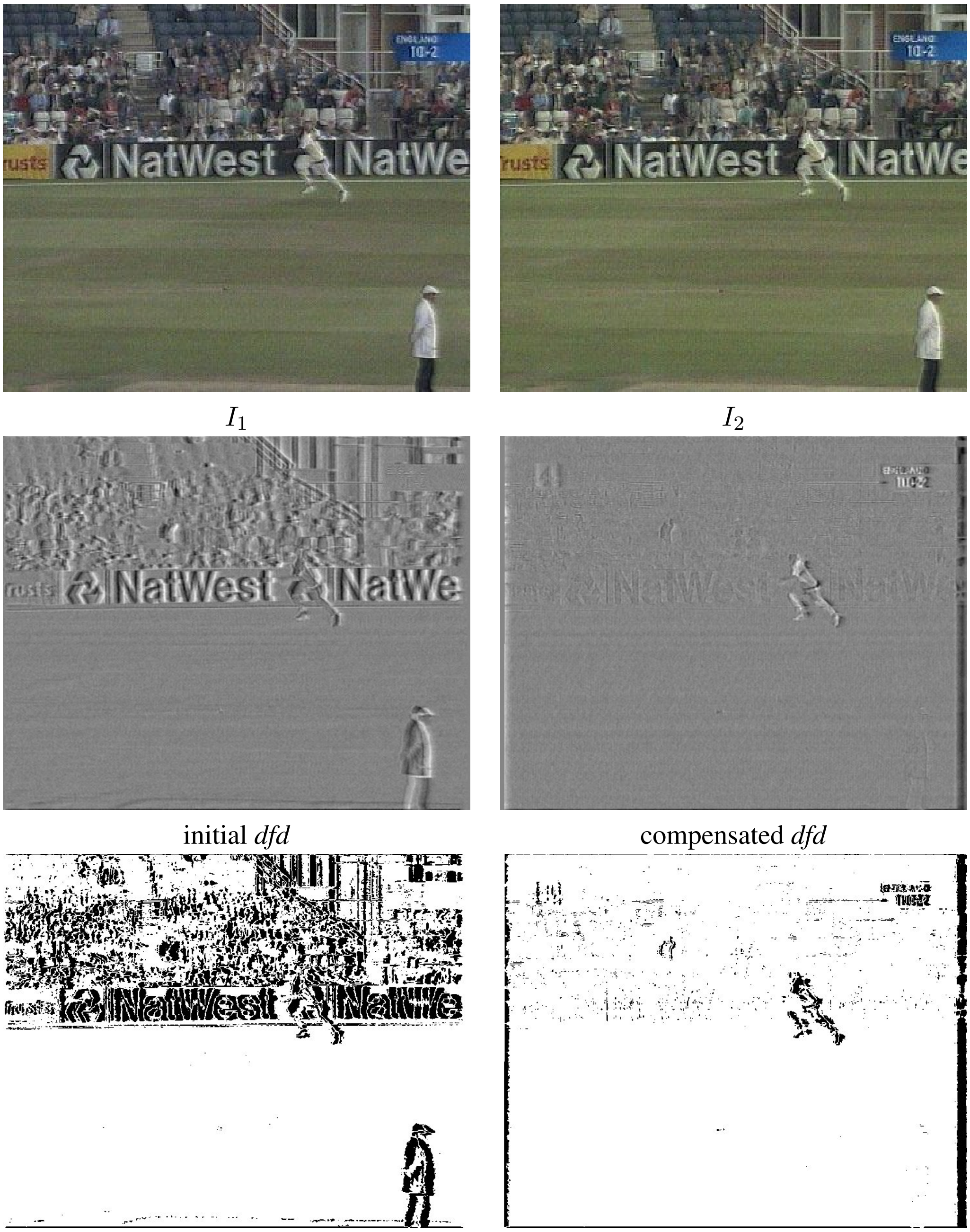
Scale parameter estimation. Using the set of residuals $\{x_i\}$ samples of the variable X, we need first to define variable Y and to compute its samples. The residuals are a mixture of inliers and outliers as being defined in this article. The proportion α of inliers is unknown and the outliers correspond to unmatched pixels due, for instance, to moving objects (different movement to the camera motion), occlusions or artifacts (specially in old films). The outliers form localized areas in the *dfd* (cf. figure 5). Using this property, we draw samples of the random variable Y such as $y_i = \sqrt{x_i^2 + x_{i+1}^2}$ where x_i and x_{i+1} are two neighbouring residuals in the *dfd*. This strategy allows to preserve a similar proportion of inliers in the observations $\{y_i\}$. Using our estimation scheme with histograms for a faster computation, the scale parameter is set to $\sigma_\rho = 3 \times \sigma_Y$. This choice insures that 99% of the inliers are kept for the estimation of the camera motion parameters.

Accuracy of the estimates. Using video (a) (cf. figure 4) where the ground truth is known, the scale parameter and the motion parameters are estimated while adding gaussian noise of variances 10 and 100. Compared to the ground truth, the motion parameters are estimated with a Mean Square Error below 0.00007 on zoom-rotation parameters and 0.05 pixels on the translation parameters. The estimated scale parameter of the class of inliers is also stable over the sequence: the standard error of the estimate σ_Y of one grey level compared to the ground truth.

Scale Adaptability. On the video (b) (cf. fig. 4), our unsupervised robust camera motion estimation is also performed. No ground truth is known, however the estimated parameters are coherent with the one with a manually tuned estimation. The estimated scale parameter of the inliers in the *dfd* remains constant over 400 frames of the sequence (b). 10 frames show a slight over-estimation of the scale. Those artifacts correspond to sudden changes in the intensity values (flicker) that increase the *dfd* [Kokaram et al., 2003]. The automatic SD estimation allows to account for those changes in the data stream. The algorithm proposed in (9) has also always converged in our experiments undertaken on different videos.

3.2.2 Moving object detection

Figure 5 shows an example with two successive images from a video of cricket. The images of the residuals is also shown before and after motion compensation. The estimated scale σ_Y allows to take a decision in between pixels being outlier residuals (in black) and pixels being inlier residuals (in white). Those binary maps allows the detection of independent moving objects in the sequence, and carry relevant information, for instance, for video understanding [Coldefy et al., 2004]. Our method provides an automatic thresholding method that does not require to manually set a threshold over the weights as in [Coldefy et al., 2004].



I_1 I_2

initial *dfd* compensated *dfd*

Figure 5: Application to camera motion estimation. Top: two successive images from a sport video. Middle: corresponding image of residuals when the motion is not compensated (left), and when it is compensated (right). Bottom: maps of the outliers (residuals above $3 \times \sigma^Y$).

4 Future work

In this article, we proposed an unsupervised method for camera motion estimation and moving object detection in video. The whole objects are not detected but only its parts that generate outliers in the *dfd*. Future work will aim at improving the segmentation of the moving objects by using mathematical morphology and/or hysteresis thresholding to take into account spatial correlation between neighbouring pixels.

Acknowledgments

This work has been funded by the European Network Of Excellence on *Multimedia Understanding through Semantics, Computation and Learning*, MUSCLE FP6-5077-52 (www.muscle-noe.org). We thank the institute KOGS/IAKS of Universität Karlsruhe for the video sequences (http://i21www.ira.uka.de/image_sequences/).

References

[Aiazzi et al., 1999] Aiazzi, B., Alparone, L., and Baronti, S. (1999). Estimation based on entropy matching for generalized gaussian pdf modeling. *IEEE Signal Processing Letters*, 6(6):138–140.

[Bouthémy et al., 1999] Bouthémy, P., Gelgon, M., and Ganansia, F. (1999). A unified approach to shot change detection and camera motion characterization. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:1030–1044.

[Chen and Meer, 2003] Chen, H. and Meer, P. (2003). Robust regression with projection based m-estimators. In *International Conference on Computer Vision*, pages 878–885, Nice, France.

[Coldefy et al., 2004] Coldefy, F., Bouthemy, P., Betser, M., and Gravier, G. (2004). Tennis video abstraction from audio and visual cues. In *Proceedings IEEE Workshop on Multimedia Signal Processing, MMSP'2004*, pages 163–166, Siena, Italy.

[Dahyot and Kokaram, 2004] Dahyot, R. and Kokaram, A. (2004). Comparison of two algorithms for robust m-estimation of global motion parameters. In *proceedings of Irish Machine Vision and Image Processing Conference*, pages 224–231, Dublin, Ireland.

[Dahyot et al., 2004] Dahyot, R., Rea, N., Kokaram, A., and Kingsbury, N. (2004). Inlier modeling for multimedia data analysis. In *IEEE International Workshop on MultiMedia Signal Processing*, pages 482–485, Siena Italy.

[Dahyot and Wilson, 2006] Dahyot, R. and Wilson, S. (2006). Robust scale estimation for the generalized gaussian probability density function. *Advances in Methodology and Statistics (Metodološki zvezki)*, 3.

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

[Hasler et al., 2003] Hasler, D., Sbaiz, L., Süsstrunk, S., and Vetterli, M. (2003). Outlier modeling in image matching. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 25(3):301–315.

[Kokaram et al., 2003] Kokaram, A. C., Dahyot, R., Pitie, F., and Denman, H. (2003). Simultaneous luminance and position stabilization for film and video. In *Visual Communications and Image Processing*, pages 688–699, San Jose, California USA.

[Rosin, 1997] Rosin, P. L. (1997). Edges: saliency measures and automatic thresholding. *Machine Vision and Applications*, 9:139–159.

A HMM Framework for Motion based parsing for video from Observational Psychology

Daire Lennon, Naomi Harte & Anil Kokaram
Electronic Engineering Dept.,
University of Dublin,
Trinity College,
Dublin, Ireland;
lennondh@tcd.ie

Erika Doyle & Ray Fuller
Dept. of Psychology,
University of Dublin,
Trinity College,
Dublin, Ireland;
edoyle4@tcd.ie

Abstract

In Psychology it is common to conduct studies involving the observation of humans undertaking some task. The sessions are typically recorded on video and used for subjective visual analysis. The subjective analysis is tedious and time consuming, not only because much useless video material is recorded but also because subjective measures of human behaviour are not necessarily repeatable. This paper presents a HMM framework for content based video analysis to facilitate the automated parsing of video from one such study involving Dyslexia. The framework relies on implicit measures of human motion that can be generalised to other applications in the domain of human observation.

Keywords: HMM, Motion Based Parsing, Rotation Estimation

1 Introduction

Visual content analysis technology applied to surveillance applications is well established [1, 2]. Less explored is a class of human observational applications related to scientific study. The field of Psychology contains many such applications which may admit solutions from the visual content analysis domain. In surveillance applications, the behaviour of the people involved is not strongly constrained. This poses a challenge to understanding in that context. In scientific applications, it is likely that the behaviour being observed is restricted in some way in order to make measurements or other inferences. In Psychology this may be in a controlled environment or a particular set of movement/behaviours are being observed in a natural setting. This implies that content analysis could bridge the semantic gap more easily in that domain.

Previous work in Psychology has asserted a connection between reflex movement in children and a Specific Learning Disability (Dyslexia) [7]. To investigate this connection Psychologists at Trinity College `www.dysvideo.org` have designed a set of experiments that attempt to quantify this relationship in a study based on 150 children. The central idea is that children with a propensity to develop Dyslexia are unable to execute particular movements without some unavoidable associated reflex. Figure 1 shows an example of one such movement. The experimenter rotates the head of a child right and left. While doing so any involuntary bend in the arms is noted. The idea is that the presence of that reflex is in some way correlated with the presence of Dyslexia.

Work presented in Joyeux et al[3] reports on the **DysVideo** project at Trinity College that was set up to observe the development of 150 children between ages 4-7 years. Video recordings are made of children observed though 3 sessions, each of 20 mins duration, and 6 months apart. Unfortunately, in each recording of 20 mins, less than 5 mins is useful material. Children may take a long time to settle down and may need to be cajoled through each session. The

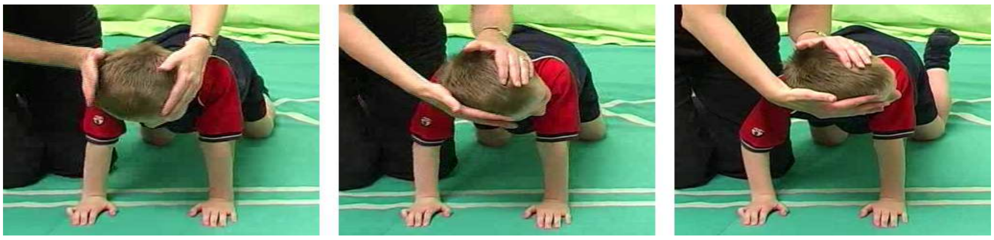


Figure 1: A demonstration of the ATNR exercise

Dysvideo project exploits automated content based audio and video analysis to allow Psychologists to index directly the useful portion of the video. Preliminary work on automated parsing was presented in Joyeux et al[3]. The focus there was the framework design and algorithms for coarse parsing and encoding of metadata by exploiting the audio stream. The parsing provided was good enough such that the estimated index points were guaranteed to contain the useful visual information. The size of the indexed portion typically contained about 20 seconds of material before the start and after the end of the actual useful motion experiment itself. This was adequate for the psychologists to browse quickly to the start of the useful recordings and perform the subjective motion measurement.

Recall that the point of the programme is to measure the presence of certain motion based reflexes in children. Currently for psychologists the only reliable way of doing this is for a human to assess the degree to which the child cannot hold a particular position. It is true that motion tracking equipment exists for this purpose, but magnetic based tracking is expensive, while visual marker based tracking would require the cooperation of the child in wearing the markered suit and not damaging the markers for future use. Quantitative visual motion measures can be designed by estimating the motion of the particular limb in the field of view. The important observation here is that this is only possible by ensuring that the motion measurement of the correct limb portion is being made and starts at the right time. Therefore for quantitative assessment of motion a finer granularity of parsing is needed. Work in this context was presented recently by Kokaram et al [4]. There the idea was to use an explicit measure of rotational motion to index measurable episodes directly. This work proposes instead a more robust framework for inference based on motion using the Hidden Markov model. The idea is to train HMMs to detect the specific type of motion required. This work also improves upon the range of motion features used and proposes the use of a novel feature for parsing: the curl of the estimated motion field.

The next two sections provide background for the reader that is useful in appreciating the context of the parsing algorithms developed. Section 5 the introduces the new features and the remaining sections present the use of the HMM framework.

2 Overview

Fig 2 shows the difference between the markers provided by audio parsing [3] and the exact start and end of a particular session, Test 10. The audio parsing is achieved by allowing the user to insert a specific audio tone into the recording using a handheld PC (PalmPilot). Postprocessing the audio signal [3] allows DTMF audio tones to be detected . The figure illustrates the importance and reliability of using audio markers to reject unwanted content. In this Test, the experimenter firmly rotates the head of the child while the child is on all fours. The hypothesis is that in children with a *retained reflex* one of the arms will bend at the elbow involuntarily during the motion of the head. To isolate the relevant content for analysis it is necessary to i) locate arms during that motion and ii) identify video portions during which the head is rotated. For some time before the start of the actual experiment, the child is coached and may undergo a few trials, in addition the child may move around and simply not be in the field of view. During the relevant video portions, the arm location will be more stable, and the rotation of the head more coherent. Therefore these features can be used to index the video with increased granularity. The starting point is an estimate of body location.

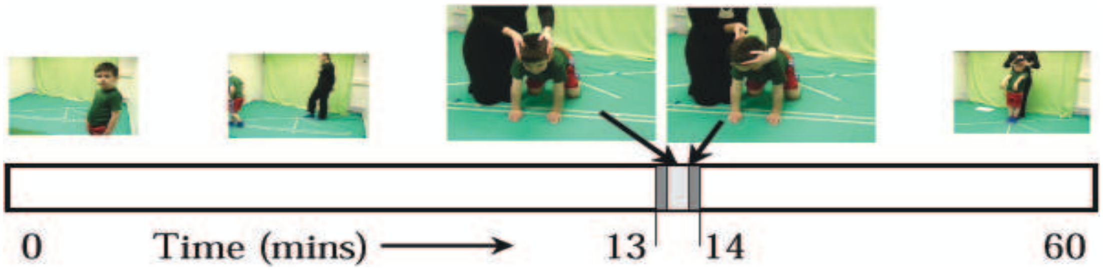


Figure 2: An example of content throughout a typical recording of 60 mins. The location of the audio markers coarsely delineates the relevant video for one Test. This is shown at mins 13 and 14. The actual useful content lies inside this period, indicated by two typical frames. The paper focuses on using motion based features for delineating this exact start and end of the experiment within the coarse audio marker period. Note that the idea of coarse indexing is to quickly reject the outlier content e.g. when the child is not in view, not cooperating, or not assuming the start position.

3 Body Localisation

Head and arm localisation is facilitated by skin detection. This is achieved by a simple colour segmentation process. The requirement is to configure a label field $l(\mathbf{x})$ that is 1 at pixel sites $\mathbf{x}$ containing skin and 0 otherwise. The algorithm is as follows.

1. Candidate pixels expressing skin ($l(\mathbf{x}) = 1$) are detected by colour thresholding (from [8]) using the following criterion

$$l(x) = \begin{cases} 1 \quad \text{if} \begin{cases} (R > 95)\&(G > 80)\&(B > 40) \\ \&(R > G - 15)\&(R > B) \\ \&((R - min(G,B)) > 10) \end{cases} \\ 0 \quad \text{Otherwise} \end{cases} \tag{1}$$

   The various parameters used in delineating the colour region were determined from the lighting used in the pictures recorded. This is the same throughout 100 hours of recording. The first two criterion delineate skin colour, while the last one rejects false alarms due to pixels that are near grey or near yellow.

2. The label field $l(\mathbf{x})$ is post-processed to smooth the surface. This is achieved using morphological closing with a dilation element of 3 pixels and a erosion element of 4 pixels.

As shown in figure 1, the arms are generally the largest area of skin exposed in the view. In addition they are near vertical. Hence a vertical sum (integration) of the label field yields modes corresponding to the horizontal position of the arms. Given the detection field $l(\mathbf{x})$ the vertical projection is defined as $p^v[h] = \sum_k l(h,k)$. Noise in $p^v[h]$ is removed by filtering with a Gaussian filter with 9 taps and variance 1.5. To detect modes in $p^v[h]$ the two most significant maxima are selected that are at least 50 pixels apart in PAL videos. This allows robustness to false alarms within a single arm segment. Figure 3 clearly shows the correlation between lobes and horizontal arm location for 2 different recordings of 2 different children. Note the false alarms due to poorly detected skin in the background (due to strangely coloured walls) are rejected with this process.

Locating the hands is achieved through the horizontal projection of the label field $p^h[k] = \sum_h l(h,k)$. The first maxima corresponds roughly to the middle of the hand position because of the orientation of the child in the view. This is shown in Figure 3. The very first non-zero projection corresponds to the start of the hand location. The hand size is estimated to be the width of the lobes corresponding to each arm, D. The wrist location is hence taken to be $1.5D$.

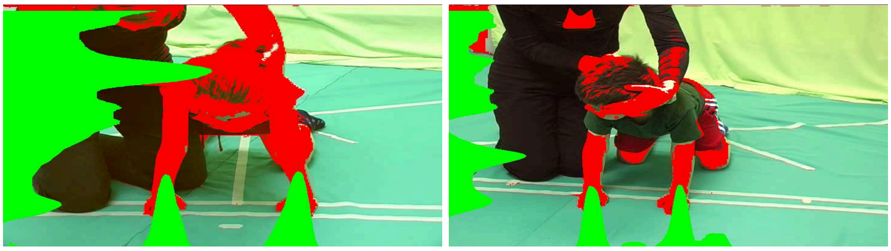


Figure 3: Example frames from different sequences showing the results of skin detection and hence body localisation. The detected skin pixels are coloured in red. The horizontal and vertical projections of the label field are shown in green along the left and bottom edges of each frame. This illustrates that the lobes in vertical projection correspond to arm location. The first mode in vertical projection corresponds to arm location.

In addition, the average forearm length is approximately 2.5 times the hand width in this view, hence the location of the elbow can be roughly delineated vertically. This enables a bounding box to be placed that contains the hand and arm locations. The process is found to be better than 99% accurate in these sequences, provided the child adopts the correct position. Typical results are shown in Figure 3. For video examples, see www.sigmedia.tv/research/indexing/dyslexia/.

The location of the arms is used to bound the head location horizontally. Therefore, head location is assumed to be contained within a column of the image bounded by the left and right arm locations. Unfortunately, detection of the head using projections is not reliable because in horizontal projection the face and arms of the experimenter can often cause ambiguity.

4 Motion Based Parsing: Features

The ultimate aim is to detect the contiguous sequence of frames showing rotation of the head. Given the delineation of a region containing the head, it is possible to estimate directly that rotation, and that can be used to attempt parsing as in [4].

For each frame, gradient based motion estimation [5] was performed where the previous frame and motion vectors were stored. Motion vectors either side of the located arms were removed for improvement in the information about the child since only those vectors were applicable. Using the motion vectors themselves and plotting there perpendicular lines in an accumulator array resulted in finding an approximate centre of rotation. This is based on the principle that a perpendicular to a tangent of a circle will always pass through the centre of the circle. The central four images in figure 4 shows a selection of accumulator arrays ranging over a head rotation sequence. The distance between the centre's of rotation from frame to frame was consistently small and stable when rotation was occurring. This was used as a feature to indicate rotation.

In this work, the use of the curl of the motion vector field is also exploited to yield an implicit measure of rotation. Given a motion vector at site $\mathbf{x}$ is specified as $\mathbf{d} = [d_1(\mathbf{x}), d_2(\mathbf{x})]$, where the horizontal and vertical displacements are d_1 and d_2 , the curl of the motion at that site $\mathcal{C}$ is given as below

$$\begin{aligned} \mathcal{C}(d_1, d_2, \mathbf{x}) &= \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{d}{dx} & \frac{d}{dy} & 0 \\ d_1 & d_2 & 0 \end{vmatrix} \\ &= \vec{k}(\tfrac{d(d_1)}{dy} + \tfrac{d(d_2)}{dx}) \end{aligned} \tag{2}$$

The curl therefore is a vector pointing out of the image plane with a length that is proportional to the amount of rotation at that site. The bottom four images in figure 4 show a selection of the curl matrices ranging over a head rotation sequence.

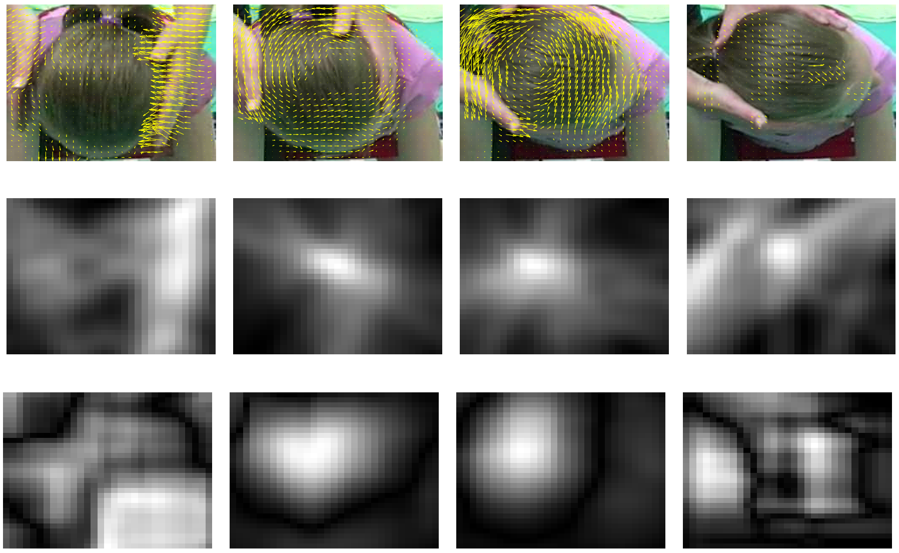


Figure 4: The top four images show a selection of frames used to demonstrate a sequence of head rotation. The central four images show the same sequence for the accumulator array and the bottom four images show the sequence for the curl matrix. All of the above images have been zoomed in on to improve clarity.

As can be seen in the central images in figure 4, the peaks in the curl correspond well to centre of rotation and there is stability in position during rotation. The distance between the maximum peaks from frame to frame indicates the level of stability and it can be observed that there is relatively little motion of the maximum peak during rotation. The distance measure between frames for the maxima was used as a feature. It is also observable that the peak value of the curl rises and falls relatively smoothly with rotation. The derivative of the graph of peak values was also used as a feature.

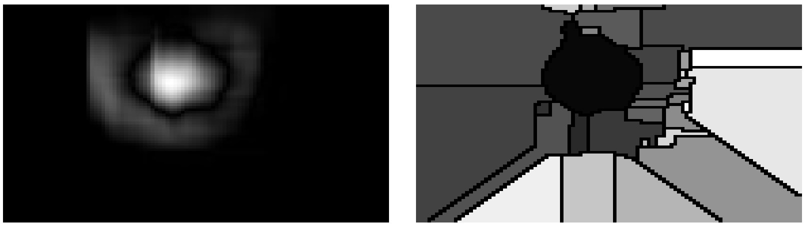


Figure 5: Example frame used to demonstrate watershed segmentation of the curl surface.

The main mass of the maximum peak during rotation are observed to be reasonably constant. To use this information, it is necessary to segment out the main mass of the curl surface. This is done with watershed segmentation [6] on that inverted curl surface, as demonstrated in figure 5. The masses of the maximum peak for each frame was used as a feature in our HMM models.

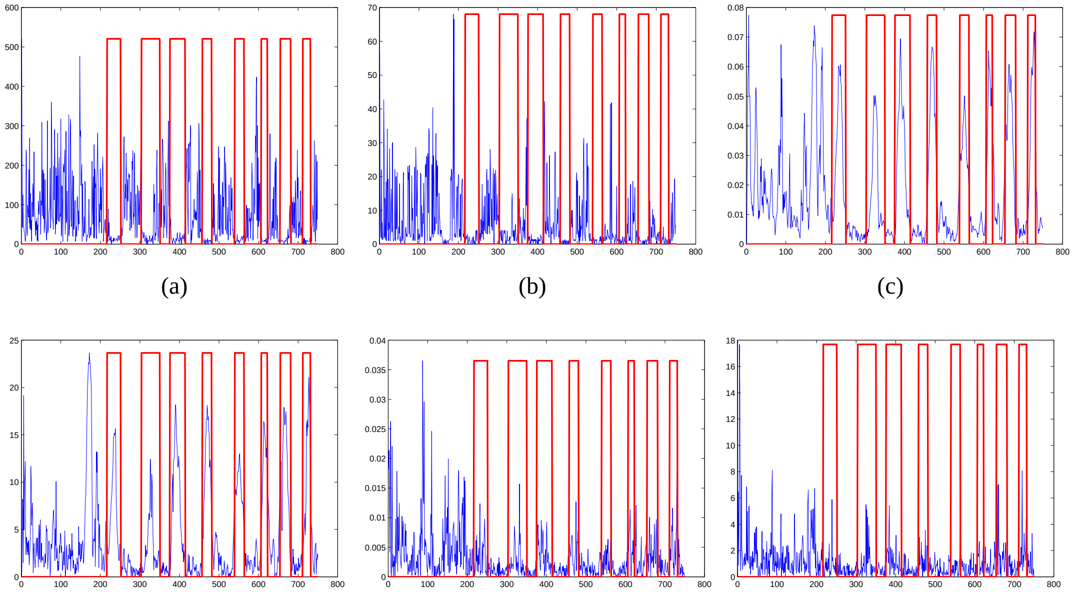
It was also found that the graph of the peak masses was consistently rising and falling during rotation. The derivative of the graph of the area under the maximum peak for the entire sequence was also considered to contain useful information and was so used as a feature vector.

The set of features used for motion based parsing can be summarised as follows:

1. The distance from maximum peak to maximum peak of the accumulator array from frame to frame. See figure 4 for illustration of process.

2. The distance from maximum peak to maximum peak of the curl surface from frame to frame.

3. The value of the maximum peak of the curl surface.



(a) (b) (c)

(d) (e) (f)

Figure 6: An example set of feature vectors to be used in the training of an HMM. The red marking represent manual segmentation markers. Figure (a) & (b) represents the distance between the maximum peaks in the accumulator array and curl matrix from frame to frame respectively. Figure (c) is the graph of the maximum peak values in the curl matrix and figure (d) is area under the peak surface. Figure (e) & (f) are the derivatives of the graphs of figure (c) & (d) respectively.

4. The area under the maximum peak surface, as segmented by the watershed algorithm on the curl surface.
5. The derivative of the graph of the maximum peaks of curl surface for the entire sequence.
6. The derivative of the graph of the area for the maximum peaks of curl surface for the entire sequence.

This feature vector $\mathbf{f}_n$, containing these 6 measures, is measured for each frame in the sequence. Figure 6 shows the evolution of each component over a an entire exercise sequence.

5 Motion Based Parsing: Modelling

The HMM is a well established framework for time series modelling and has been very successful in speech recognition [9]. The idea here is to use two HMMs to model the evolution of the multidimensional feature vector $\mathbf{f}_n$. One HMM models the non-rotation segments and the other models the rotation segments. Given a manually labelled training set, the parameters of each HMM can be established. These models are used in parsing new examples. The use of HMMs for modelling video features is a relatively recent idea, exploited successful for sports by Rea et al [10].

Fig 5 shows the structure of the HMM used in this framework. It is a four state HMM with the ability to transit from any state to any other state. Note the difference from the standard left to right models employed in speech recognition applications. The HTK Speech Recognition Toolkit was used to initialise and train the statistical parameters of our HMM models. Gaussian distributions were assumed with single mixtures per state distribution. A total of 23 videos were selected from session 1, 16 to be used for training and 7 for testing. The criterion used to select the videos was based on arm separations. An example of training data is shown in in figure 6. The performance of the resulting models is presented in section 6.

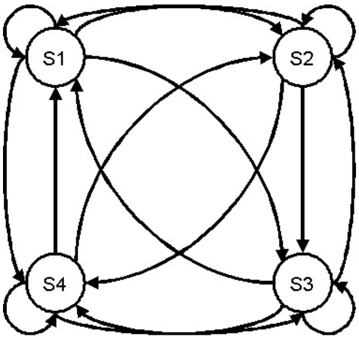


Figure 7: A 4 state HMM Model with possible transitions in all directions. This was the model used for both the rotational and non-rotational HMM's. Although with differing transition probabilities and feature statistics.

6 Results

Video selection was difficult because the quality of the child's motion was essential. The main goals of the video recordings were the ability for direct analysis by the psychologists, secondary was our analysis of the videos. As such the demonstrator focused more on just recording the exercise for human evaluation as opposed to improved conditions required for our evaluation. So if the size of the child is too small relative to the frame the motion of child will be too small relative to the larger motion of the experimenter which would be clearly visible in a shot of wide angle. Arm separations are a clear indicator as to the size of the child relative to the frame size i.e. zoom factor. The selection of 23 videos were chosen to best represent the exercise on the bases that they were the videos with the largest visible children.

The 23 videos had to be manually segmented. The frame numbers for the start and end of any rotational occurrence were noted for each sequence. These manual segmentations were compared with the outputs generated by the HMM models. The analysis of the manual segmentation markers versus the HMM markers had to take into account the human error. It can seen for individual video comparisons that there is a difference between the two markers of a couple of frames. Human observations follow the exercise start and finish more so than the rotational occurrences. Taking into account that human observations are more subjective, an error of 12 frames was deemed acceptable. The results below reflect this adaptation.

$$Recall = \frac{Correct}{Correct + Missed} \quad Precision = \frac{Correct}{Correct + False}$$

	Recall	Precision
Test Video 1	81.7109	86.8339
Test Video 2	77.7778	82.3529
Test Video 3	47.2603	50.4263
Test Video 4	62.7525	64.4617
Test Video 5	55.9361	77.044
Test Video 6	62.6935	68.1818
Test Video 7	63.6879	72.1865

Results to date have been extremely promising. By modelling the described features with the HMM framework the detection of rotational occurrences has been improved significantly over data thresholding, which was the previous method of data analysis. The performance of test video 3 is poor and may be attributable to poor motion vectors from the original data. This will be further investigated.

7 Final Comments

Future work includes trying to gain a better understanding of the contribution of the features to the recognition framework. Refining the manual segmentations to only take into account actual rotations and not other motion. Adapting the algorithm to poorer quality videos, i.e. ones where the child appears small in the frame. These refinements will hopefully improve the HMM Models and accuracy of detection.

References

[1] Edward Y. Chang and Yuan-Fang Wang. Video surveillance. In *First ACM SIGMM international workshop on Video surveillance*, November 2003.

[2] Arun Hampapur. S3-r1: The ibm smart surveillance system-release 1, June 2005.

[3] L. Joyeux, E. Doyle, H. Denman, A. C. Crawford, A. Bousseau, A.Kokaram, and R. Fuller. Content based access for a massive database of human observation video. In *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 45–62, October 15-16 2004.

[4] A. Kokaram, E. Doyle, D. Lennon, L. Joyeux, and R. Fuller. Motion based parsing for video from observational psychology. In *Proc. SPIE Vol. 6073, p. 265-274, Multimedia Content Analysis, Management, and Retrieval*, Jan 2006.

[5] A. C. Kokaram. *Motion Picture Restoration: Digital Algorithms for Artefact Suppression in Degraded Motion Picture Film and Video*. Springer Verlag, ISBN 3-540-76040-7, 1998.

[6] Vincent Luc and Pierre Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 13:583–598, June 1991.

[7] M. McPhillips, P.G. Hepper, and G. Mulhern. Effects of replicating primary-reflex movements on specific reading difficulties in children; a randomised double-blind, controlled trial. *Lancet*, 355:537–541, 2000.

[8] Peter Peer, Jure Kovac, and Franc Solina. Human skin colour clustering for face detection. In *EUROCON 2003 - International Conference on Computer as a Tool*, 2003.

[9] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[10] Niall Rea, Rozenn Dahyot, and Anil Kokaram. Modelling high level structures in sports with motion driven hmms. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2004.

Data Clustering & Texture Analysis

Processing the Multispectral Satellite Images Using RBF-based Neural Network

Rauf Kh. Sadykhov
Computer Department
Belarusian State University of Informatics and Radioelectronics
6 P. Browka Street
220013 Minsk Belarus
rsadykhov@bsuir.unibel.by

Leonid P. Podenok
System Identification Laboratory
United Institute of Informatics Problems
of National Academy of Sciences of Belarus
6 Surganov Street
220012 Minsk Belarus
podenok@lsi.bas-net.by

Yegor V. Pushkin
Computer Department
Belarusian State University of Informatics and Radioelectronics
6 P. Browka Street
220013 Minsk Belarus
egor.pushkin@gmail.com

Abstract

The method and instrumental toolkit based on radial basis function neural network to process multispectral satellite images are presented. Experimental testing of network has been carried out using images received from Landsat 7 ETM+ satellite. These images include all the bands of Enhanced Thematic Mapper: 0.45-0.52, 0.52-0.60, 0.63-0.69, 0.76-0.90, 1.55-1.75, 10.4-12.5, and 2.08-2.35 micrometers. All the layers of multispectral image were processed as aggregate. Using the histogram instead of raster representation of a multi-bands fragment to be supplied on network's inputs has allowed to increase efficiency of classification of objects via tone criterion.

Keywords: Multispectral Satellite Image, Neural Network, Machine Vision.

1 Introduction

Neural networks are widely used to process multispectral satellite images [Haykin, 1998], [Atkinson and Tatnall, 1997], [Heerman and Kjazenic, 1992], [Lee and Landgrebe, 1997], [Landgrebe and Schowengerdt, 1978], [Landgrebe, 2005], [Tilton et al., 1999], starting from segmentation and finishing forecasting of change of objects [Maier and Dandy, 2000].

The neural network to be applied consists of a layer of Radial Basis Function cells and a layer of neurons with sigmoid threshold function [Orr, 1996]. The network has been designed so that the data from all the layers of multispectral image were processed as aggregate.

The image has been broken into set of fragments during processing (they can be both overlapped and not depending on the purpose of experiment). The method of allocation of more informative area in a processable fragment has been applied due to use of a mask, applied to every image submitted on an input of a neural network.

There is a lot of methods of preliminary data processing, images enhancement, features extraction to classify the objects on multispectral satellite images [Sadykhov and Podenok, 2005]. The developed network has been modified to apply the histogram analysis instead of raster representation of a multi-bands fragment on network's inputs. It gave ability to increase efficiency of classification of objects via tone criterion. The used method of recognition is completely insensitive to a textural component of processed fragment.

2 Statement of a task

The task to be fulfilled was the evaluation of capabilities of radial basis function (RBF) based neural network to select and classify different land cover types. Experimental testing of proposed network has been carried out using images of Myadel district of Belarus received from Landsat 7 ETM+ satellite. These images include all the bands of Enhanced Thematic Mapper: 0.45-0.52, 0.52-0.60, 0.63-0.69, 0.76-0.90, 1.55-1.75, 10.4-12.5, and 2.08-2.35 micrometers. The data intended to test the classifier were prepared in the following way. Five areas of multispectral image corresponding to different land cover types have been selected. Then trainig and verifying sets were generated from these data. The training samples have been represented by 10 samples from every area using all the spectral bands. The dimension of the sample was 20×20 pixels. To verify the network classifier 2000 samples of each class were picked from different areas as shown at fig. 1.

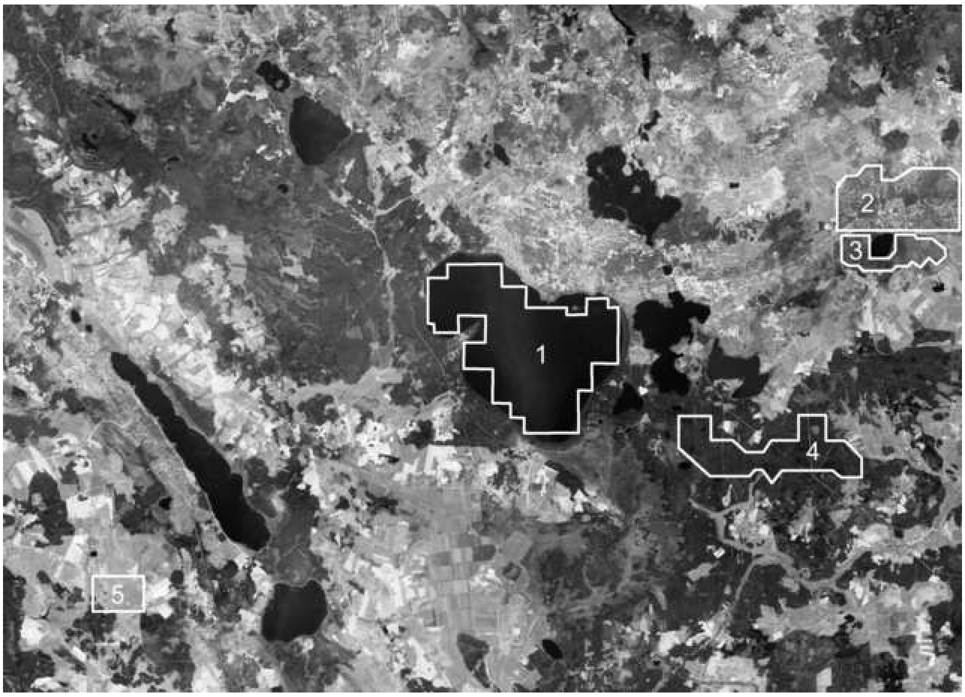


Figure 1: Region of interest. Areas picked to form verifying sets are outlined and marked.

3 The neural network: construction and training

The RBF based neural network has been chosen to implement the classifier. Each cell of a layer plays a role of a cluster center and corresponds one of the image object. Typical representatives of areas chosen for classification of multispectral image have been established as training samples.

The task of classification of objects of the stage belonging five allocated areas were put in spent experiments. It caused presence of five outputs in neural network (five neurons in output layer). Value 1 is formed on each of them in case of ranking of a fragment submitted on an input of a network to corresponding class and 0 in otherwise.

Each neuron of a hidden layer accepts 20×20 pixels fragment of multiband source image. As the result neural network (as well as each neuron of hidden layer) has $20 \times 20 \times 8 = 3200$ inputs. When the histogram instead of raw raster was used the quantity of inputs of network has been decreased down to 2048. That is resulted in reduction of expenses of time required for processing a portion of data with the use of neural network, despite of necessity of creation the histogram in real time.

Functioning of RBF cell of an input layer can be presented as follows:

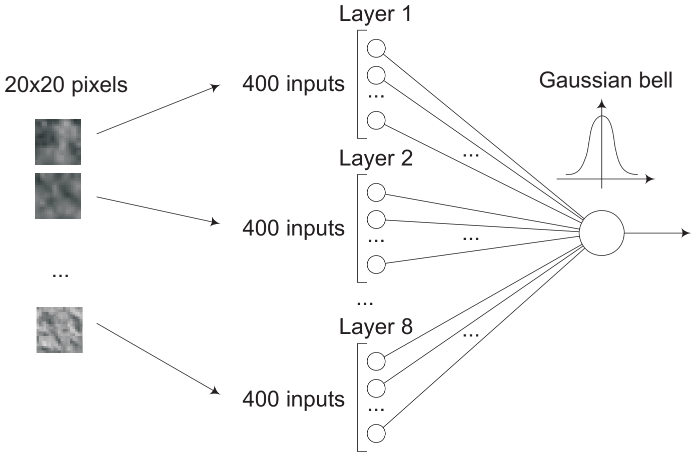


Figure 2: RBF cell functioning.

Each cell receives 3200 input values. Further the search of distance between an input fragment and the center of cluster, corresponding to a concrete cell is fulfilled. Output value in an interval $[0, 1]$ is formed as

$$NET = e^{-\frac{\|x_i - c_i\|^2}{\sigma^2}} = e^{-\frac{1}{\sigma^2}\sum_{i=0}^{n-1}(x_i - c_i)^2} \tag{1}$$

where x_i - vector, submitted on inputs of a cell, c_i - center of a cell, σ - radius of a cell, n - dimension of feature space, NET - value, formed on output of a cell.

Adjustment of centers and radiuses of RBF cells is made during training this layer. For this purpose the centers of cells were rigidly established in symbols of training set corresponding to them, and radiuses were selected experimentally.

Presence of additional layer of neurons with sigmoid transfer function is required to transform and process values formed on outputs of a layer of RBF cells. Weights of this layer should be trained to transform results of classification (clustering) by RBF layer to five target signals. Value of each of target signals should be laid in an interval $[0, 1]$. Weights of an output layer have been adjusted proceeding from the following rule

$$w_{\substack{i\in[0;4]\\ j\in[0;49]}} = \begin{cases} -2, & j < iN_z \cup j \geq (i+1)N_z; \\ 2, & iN_z \leq j < (i+1)N_z, \end{cases} \tag{2}$$

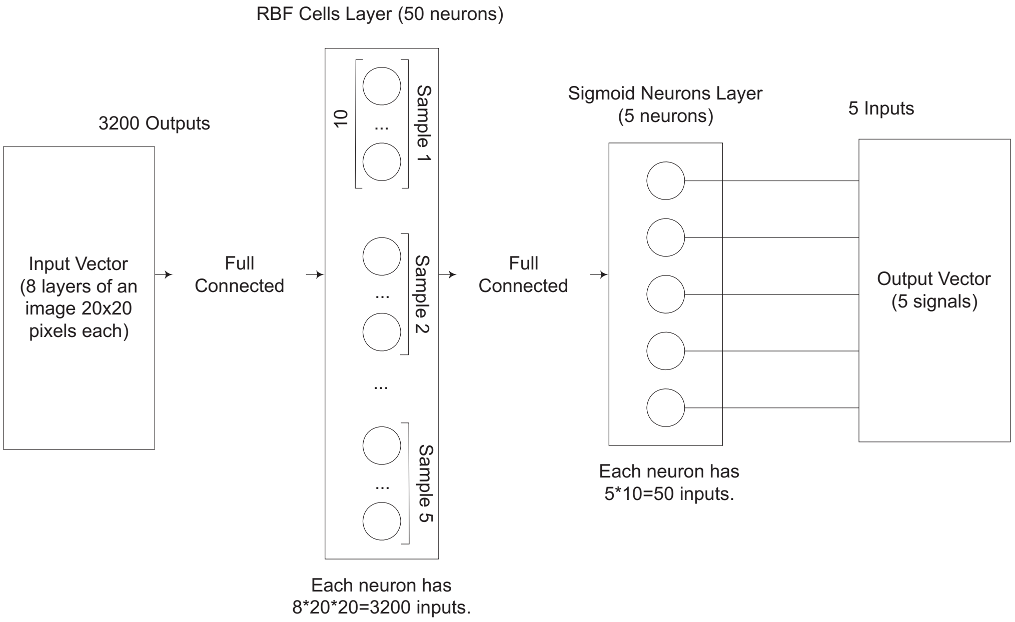
where i - index of neuron, j - index of input weight to be corrected, N_z - quantity of neurons in RBF layer responsible for each area of the initial image.

Neuron of output layer is functioned according to the following rule

$$NET = \sum_{i=0}^{N-1} w_i \cdot x_i, \qquad OUT = \frac{1}{1 + e^{-NET}}, \tag{3}$$

where x_i - vector of input values, w_i - vector of neuron weights, N - count of neurons in hidden layer (RBF cell layer), OUT - the weighted sum of values acting on inputs of neuron and it's weights, NET - value formed on an output of neuron after compression using sigmoid transfer function.

The structure of the developed network consists of two layers as shown on fig. 3



RBF Cells Layer (50 neurons)
3200 Outputs
Input Vector (8 layers of an image 20x20 pixels each)
Full Connected
10
Sample 1
Sample 2
Sample 5
Each neuron has 8*20*20=3200 inputs.
Full Connected
Sigmoid Neurons Layer (5 neurons)
Each neuron has 5*10=50 inputs.
5 Inputs
Output Vector (5 signals)

Figure 3: Complete network architecture.

The network has the following characteristics:

- 3200 inputs. The quantity of inputs has been determined by dimension of data used for training and verification of the network (fragments of eights bands of the image in the size of 20 per 20 pixels).
- 50 RBF cells in the input layer. 10 images for each class of objects have been used for training (5 classes have been allocated.
- 5 neurons with sigmoid transfer function in an output layer. The quantity of neurons in this layer has been determined by quantity of object classes. This layer is full connected with input layer of RBF cells.

To eliminate rectangular symmetry of sample data the radial weight mask α_i has been used and output of network is determined as

$$NET = e^{-\frac{1}{\sigma^2}\sum_{i=0}^{n-1}\alpha_i\left(x_i - c_i\right)^2}, \tag{4}$$

where α_i - component of a mask. The following transform has been applied to preserve functional logic of RBF cell

$$NET = e^{-\frac{1}{\sigma^2}\sum_{i=0}^{n-1}\alpha_i\left(x_i - c_i\right)^2} = e^{-\frac{1}{\sigma^2}\sum_{i=0}^{n-1}\left(\tilde{x_i} - \tilde{c_i}\right)^2}, \tag{5}$$

where $\tilde{x}$ and $\tilde{c}$ - modified input vector and center of a cell accordingly.

4 Experimental results

Training and verification data sets have been generated to carry out the experiment. Verification set has been formed from multispectral image using 2000 fragments from 5 areas of interests and consists of 10000 fragments totally. Training set has been selected from verification data set and included 10 fragments for each area, or 50 fragments totally. After training the network all the fragments from verification set including fragments used for training have been exposed to network input. As the result all training fragments have been ranked correctly to corresponding classes. Results of classification of whole verification set are represented in the following table

Table 1. Results of classification.

Area No	1	2	3	4	5
Without input mask, %	100	95.1	100	97.2	100
With input mask, %	100	92.7	100	97.6	100

The percentage ration shows, how many images from the verification set have been ranked to right class. For example, 95.1% of all fragments of area number 2 have been ranked by neural network to class corresponding to this area. Time of processing the single fragment is about 9-10 ms and time of processing whole verification set is about 180 sec when no mask has been used. When radial mask was used the time of whole classification process raised up to 280 sec. Minor alteration of results with and without mask can be explained by the specificity of a problem and sets of data used for training and running tne network.

4.1 Histogram processing

Taking into account specific of the data to be processed, histogram of samples instead of raster has been used to pass on network inputs. The histogram sample consists of eight blocks corresponding to eight bands of the multispectral image.

Process of forming the histogram is displayed on fig. 4. Histogram is presented on figure as set of functions.

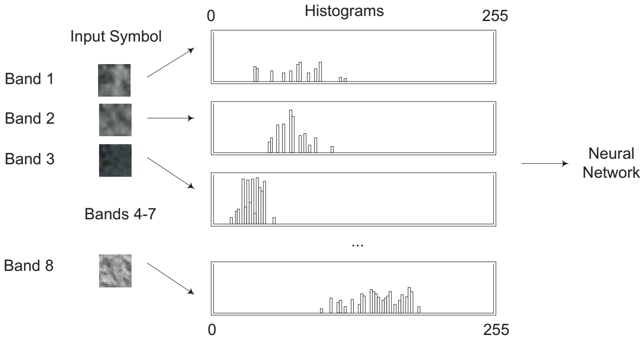


Figure 4: Histogram of sample to be passed to network

Hystograms of different objects of source multispectral image are shown on fig. 5. All the five areas are represented on the figure. The image of hystogram has inversed to provide more readable picture

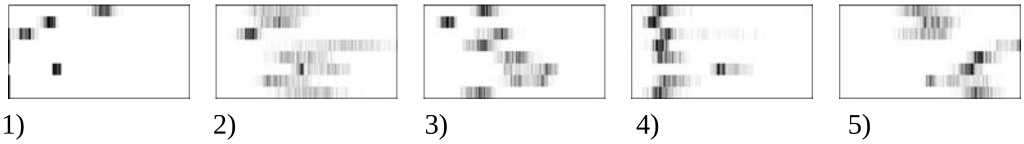


Figure 5: Histogram of objects from different areas.

Parameters of network are displayed in the table 3. Small increasing of processing performance occures because of reduction of features space dimension.

Table 3. Parameters of network when using hystogram.

Parameter	Value
Network inputs count	$8 \times 20 \times 20 = 3200$
RBF cell inputs count	$8 \times 256 = 2048$
Count of cells in hidden layer (RBF)	50
Count of neurons in output layer	5
One fragment processing benchmark, ms	9
Verification set (2000 fragments) processing benchmark, s	170

The following results of classification using hystogram instead of raw raster have been received at experiment:

Table 4. Results of classification.

Area No	1	2	3	4	5
Classification result, %	100	99.6	99.2	100	99.7

Improvement the quality of classification takes the place because of essential discrimination of processed objects in the chosen features space. It is necessary to take into account that classification was made extremely by criterion of brightness. The textural component of source fragments has been rejected at the stage of creating the histogram.

4.2 Live experiment 1

Multispectral images processing is one of the most important elements of remote sensing. A couple of experiments in this sphere have been carried out to verify quality of recognition using constructed and trained neural network model. Images of different spectral and spatial resolution have been used in these experiments.

A single band of multispectral image used in the experiment is shown at fig. 1. The neural network has been constructed and trained to perform the experiment. 10 samples of each class were picked from different areas as shown at fig. 1 to train the network classifier.

The main target of experiment is to process the whole multispectral image with the use of trained model and to build joined map of all classes which were used as source areas for picking up fragments during training neural network. Prepared network model accepts 20×20 pixels fragment of multiband source image. The image has been processed with step of 5 pixels (75% compositing between near by fragments). Total time required to process the whole image is 3 hours 27 minutes 10 seconds.

The simplest one model of all described above has been used in this experiment. Centers of RBF cells were rigidly established in symbols of training set corresponding to them, and radiuses were selected using the criterion of mean square deviation applied to the concrete cell and to the nearest cell which belongs to another class.

4.3 Live experiment 2

The next experiment has been applied to the image shown at fig. 7 which represents a fragment of Cheluskincev park (Minsk, Belarus). This image includes all the bands of visible part of spectrum: 0.4-0.5, 0.5-0.6 and 0.6-0.7 micrometers. It has a spatial resolution of 1.38 meters.

Results of classification of whole verification set (2000 sample fragments from each source area) are represented in the Table 5.

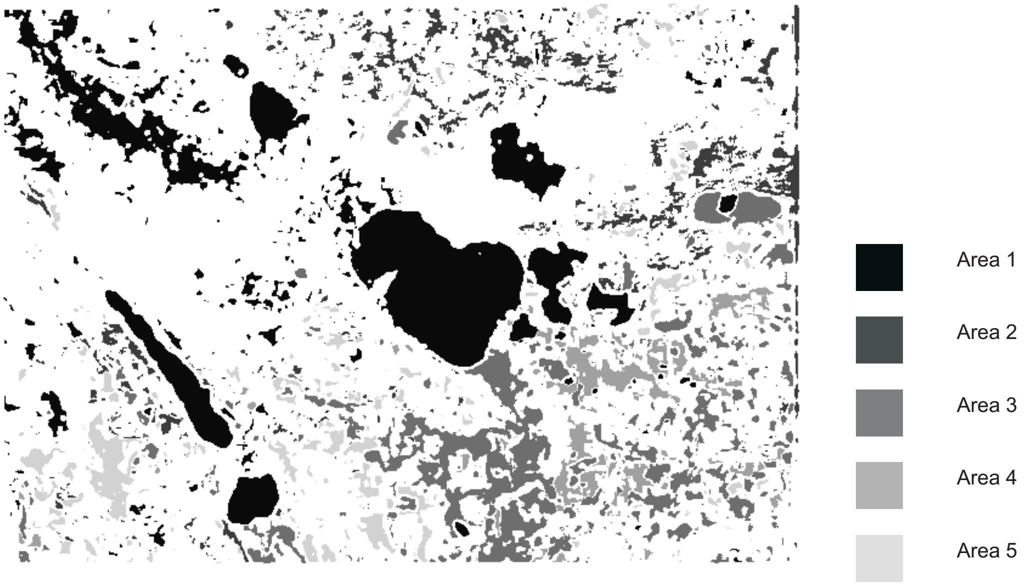


Figure 6: Result map with allocated classes displayed in gradations of gray.

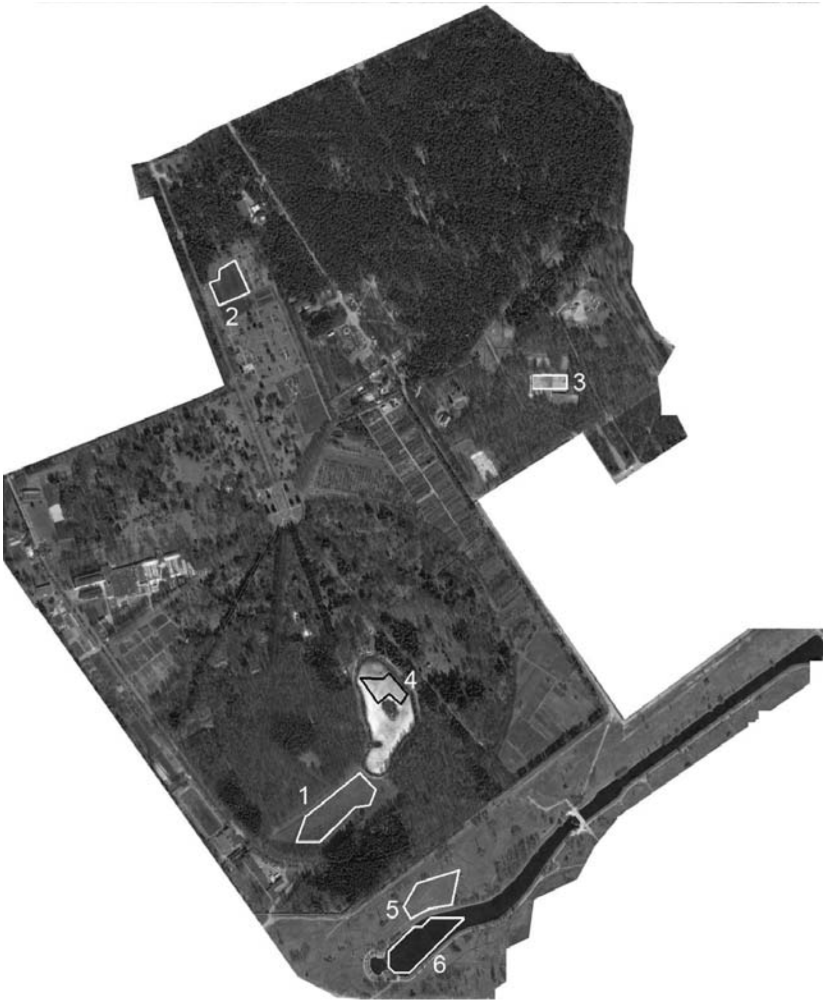


Figure 7: Region of interest. Areas picked to form verifying sets are outlined and marked.

Table 5. Results of classification.

Area No	1	2	3	4	5	6
Classification result, %	97.85	87.3	100	100	100	100

5 Conclusion

Neural Network with Radial Basis Function based layer has shown high performance and efficiency. Using the histogram instead of raster representation of a multi-bands fragment to be supplied on network's inputs has allowed to increase efficiency of classification of objects via tone criterion.

References

[Atkinson and Tatnall, 1997] Atkinson, P. M. and Tatnall, A. R. L. (1997). Neural networks in remote sensing. *Int. J. of Remote Sensing*, 18:699–709.

[Haykin, 1998] Haykin, S. (1998). *Neural Networks. A comprehensive foundation. Second Edition.* Prentice Hall.

[Heerman and Kjazenic, 1992] Heerman, H. D. and Kjazenic, N. (1992). Classification of multispectral remote sensing data using back-propagation neural network. *IEEE Transactions on Geoscience and Remote Sensing*, GE-30:81–82.

[Landgrebe, 2005] Landgrebe, D. A. (2005). Multispectral land sensing; where from, where to? *IEEE Transactions on Geoscience and Remote Sensing*, 43(3).

[Landgrebe and Schowengerdt, 1978] Landgrebe, D. A. and Schowengerdt, R. A. (1978). *Remote Sensing: The Quantitative Approach.* McGraw-Hill International Book Company.

[Lee and Landgrebe, 1997] Lee, C. and Landgrebe, D. A. (1997). Decision boundary feature extraction for neural networks. *IEEE Transactions on Neural Networks*, 8(1):75–83.

[Maier and Dandy, 2000] Maier, H. R. and Dandy, G. C. (2000). Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *ELSEVIER, Environmental Modelling & Software*, 15:101–124.

[Orr, 1996] Orr, M. J. L. (1996). *Introduction to Radial Basis Function Networks*. Centre for Cognitive Science, University of Edinburgh.

[Sadykhov and Podenok, 2005] Sadykhov, R. K. and Podenok, L. P. (2005). Raster transformation and resampling technique for geodetic applications. In *Proc. IEEE Third Int. Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS-2005), September 5-7 2005*, pages 602–605, Sofia, Bulgaria.

[Tilton et al., 1999] Tilton, J. C., Landgrebe, D. A., and Schowengerdt, R. A. (1999). *Information Processing For Remote Sensing.* Wiley.

On the importance of directional information when performing texture based unsupervised land-cover classification in urban areas.

Padraig Corcoran
National Centre for Geocomputation
National University of Ireland Maynooth,
Co. Kildare,
Ireland.
padraigc@cs.nuim.ie

Adam Winstanley
Department of Computer Science
National University of Ireland Maynooth,
Co. Kildare,
Ireland.
adamw@cs.nuim.ie

Abstract

The introduction of commercially available high resolution data from satellites such as Quickbird over the last decade has lead to a huge growth of interest into the task of deriving land-use classification for urban areas. It was quickly discovered that traditional pixel based remote sensing techniques which had previously been successfully applied to low resolution data, could no longer be used and lead to very poor results when applied to high resolution data. This lead to the emergence of a new approach to remote sensing known as Object Based Image Analysis (OBIA), where segmentation is used to derive land-cover and this serves as input to a land-use classification which is very top down driven. The research presented here focus on the first of the above steps, where texture is used to derive land-cover segmentation. Extracting useful texture features from high resolution data of urban areas is a challenging problem due to the fact that most urban land-cover is not strongly textured and is highly uniform. In previous research we have shown that incorporating traditional texture feature extraction algorithms with pixel values actually leads to a decrease in performance compared to the results achieved solely on pixel values. A post processing texture gradient diffusion technique was proposed which overcome the above, leading to an increase in performance. This paper builds on that work by evaluation the importance of directional information when performing texture feature extraction. If this information is redundant it would allow more robust estimate of our features within a smaller moving window leading to improved classification accuracy. The length of a given feature vector would also be reduced thus lessening the curse of dimensionality.

Keywords: remote sensing, urban, directional, segmentation, texture.

1 Introduction

The ever increasing use of geographical information systems (GIS) has lead to an ever greater demand for accurate and up to date land-use classifications of urban areas. Manual photo interpretation by trained operators is an expensive and time consuming process. If it was possible to automate this process it would greatly reduce cost and increase availability of such data. Traditional pixel based remote sensing techniques when applied to urban data of high spatial resolution are inaccurate due to the fact that as the spatial resolution increases so does the within class variation [1]. Many researchers in the area of remote sensing believe to overcome this failing we must move to an object orientated approach where land-use is derived in two steps [2]. The first of these steps is the generation of a land-cover (areas which are homogeneous) classification through segmentation which is used as input to a structural pattern recognition system in the second step. The structural pattern recognition system will embody knowledge representation of the relationships between different land-cover which define land-use.

Scenes in high resolution urban data are very complex and the task of deriving accurate land-cover classification is non trivial. Much effect has been invested in an attempt to derive land-cover using intensity alone but results have been disappointing due to significant within class variation leading to over and under segmentation [3, 4]. A solution to this problem would be to model this within class variation using traditional models of texture. Very high spatial resolution (less then one meter) images of urban areas are not strongly textured, boasting very different properties to images taken from texture databases so conventional methods for texture feature extraction simply cannot be *plugged* in to achieve this. Applying traditional texture feature extraction techniques produces two undesirable properties in the feature images returned and post-processing of these images is required before they can be integrated to produce a performance gain.

These two requirements have resulted in little progress being made in trying to derive unsupervised urban land-cover classification using texture. In fact most previous research in the area has focused on using texture to derive land-use classification at a higher scale (e.g. residential, industrial, commercial) using data of a lower spatial resolution [5-7]. In a recent paper [8] we proposed a texture gradient diffusion algorithm combining a technique to establish the exact location of texture boundaries [9] and vector based diffusion [10], which goes some way to overcoming the two issues mentioned above. This paper builds on this work evaluating the importance of directional information when performing texture feature extraction. The data used in this work consists of scanned aerial photography of Southampton city obtained from the Ordnance Survey Southampton. Each image has a spatial resolution of 0.25 meters.

The layout of the paper is as follows. First we discuss why it is important to evaluate the anisotropy of texture in remotely sensed data of urban areas. Section three gives details of the methodology used in this evaluation. This is followed by results, and finally the conclusions are given.

2 Isotropy/anisotropy of texture in urban areas

Directional information is obviously an important cue used by the visual system to perform texture discrimination. Julesz [11] states that directional information is an attribute which can be varied to define different textons, the fundamental building blocks of texture. Literatures from the area of remote sensing suggest a divide in opinion on the importance of such directional information. In [5, 12-14] the authors believes that such anisotropy is often immaterial for very fine or small scale textures such as those contained in remotely sensed images and thus can be ignored. Following this conclusion their geostatistical and co-occurrence texture models are only calculated for one particular direction or the average of all directions is taken. In contrast to these works there also exists literature that suggests direction is an important attribute of the texture in remotely sensed data which is required when performing texture discrimination. Lark [15] believes there are many sources of anisotropy in imagery of terrain and includes direction dependence as one feature of a working definition of texture in remotely sensed data. Carr [16] discovered that different classification accuracies where achieved using different directions in a geostatistical feature extraction algorithm suggesting an anisotropy property of texture in remotely sensed images. In [6] Conners also believed directional information to be important and extracted texture features at four directions when performing land-use classification of urban scenes.

From the above discussion it is obvious that there exists confusion on the question of whether texture in remotely sensed images is anisotropy or not. To our knowledge there exists no quantitative evaluation which eliminates this confusion in relation to the task of deriving unsupervised land-cover classification in urban areas using data of a very high spatial resolution. If direction information is redundant this would allow a more robust estimate of texture features compared to the corresponding anisotropy features within a similar size or smaller window. For example if using a geostatistical feature extraction algorithm, it is important to have a large number of observations, thus a large window size to calculate the values of the variogram robustly. According to Webster [17] a variogram based on 150 observations is satisfactory, while 225 observations is more robust. Using a window size of size 5*x*5 gives 72 observations for an anisotropy estimate for a distance of 1, while only on average 18 observations for each of the directions in an isotropy estimate also for a distance of 1. If direction is immaterial it makes sense to choose the more robust anisotropy model over the isotropy model.

Using a quantitative evaluation criterion we evaluate the performance of a texture feature extraction algorithm with and without directional information. Results suggest that in fact direction is not an important attribute of texture in urban areas.

3 Methodology

The routine used to evaluate the importance of directional information when extracting texture features is divided into four steps. First we give details of the geostatistical feature extraction algorithm used to extract anisotropy and isotropy features. Post-processing is then applied to these features to lessen the edge effect and to remove within class variation. These features are then clustered using a mean-shift clustering algorithm to produce segmentation. A relatively new quantitative evaluation technique is used. Each of these steps is now discussed in turn.

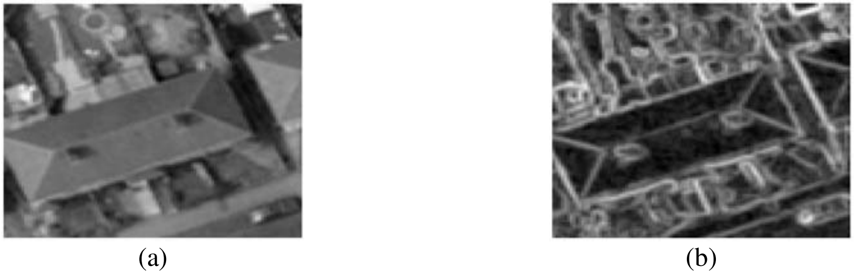
3.1 Geostatistical texture feature extraction

Geostatistics is one of most commonly used approaches to extracting texture information form remotely sensed images and has been shown to achieve promising results [18]. The centre of any geostatistics is the variogram which measure the spatial autocorrelation of a data set over different distances. Once the variogram is calculated the next step is to extract features to describe its shape and this can be achieved using two approaches. The variogram values may be used directly as features or a model may be fitted to the values and the model parameters used instead [19]. The approach of fitting a model is problematic due to the fact that one model will not fit all variograms. The process of fitting cannot be done automatically and requires manual supervision. In [20] the authors detail an approach to deriving model parameters without the need to fit a prior model avoiding the above problem. In this research it was decided to use the variogram values directly as features. This approach is computational efficient, quiet easy to implement and has been shown to achieve good results in the past [15]. To reduce the effect of outliers a robust estimation of the variogram known as the mean square-root pair difference (SRPD) is used [21]. The SRPD is calculated within a moving window of size $3x3$ with a lag of 1 and for 4 directions in the anisotropy model, these four values are averaged for the isotropy model.

3.2 Diffusion post-processing of feature images

A very useful qualitative evaluation of any feature image is how closely it resembles an accurate segmentation result. The feature image returned from a valuable feature extraction algorithm should be very close to the desired segmentation result. Visual inspection of the feature images returned from our feature extraction algorithms reveal two undesirable properties of the algorithms when applied to remotely sensed data of urban areas with high spatial resolution.

The between class variation of land-cover tends to be significant compared to the corresponding within class variation leading to a large response at land-cover boundaries [9]. This property causes the generation of unwanted segment classes along all boundaries when the feature images are clustered, thus the response to this between class variation needs to be reduced. A second problem is that the within class variation of the texture features is considerably high which leads to over segmentation. This variation needs to be removed by smoothing without losing edge localization and blurring the image which leads to mixed classes. These two unwanted properties are illustrated in figure 1.



(a) (b)

Figure 1: Image (b) shows the anisotropy feature extracted from image (a). The feature extraction algorithm responds strongly to the building boundary. There is also significant within class variation in the feature image. Ordnance Survey (c) Crown Copyright. All rights reserved.

In [8] we proposed a post-processing algorithm of the feature images which comes close to overcome the two above issues. This technique combines vector value diffusion and a technique to identify the exact location of texture boundaries which we now discuss.

3.2.1 Vector value diffusion

To smooth out the within class variation without blurring the image losing edge locations and giving mixed classes, we apply vector value diffusion to the feature images [10]. This is a non-linear smoothing technique where the amount of smoothing performed at any location is preoperational to the gradient magnitude at that point, thus less smoothing is performed at land-cover boundaries. An image *I* is smoothed/scaled by solving the partial differential equation (PDE)

$$\partial_t u_i = div\left(g\left(\sum_{k=1}^{M} |\nabla u_k|^2 \right) \nabla u_i \right) \tag{1}$$

with initial condition $u(x, y, 0) = I(x, y)$ where *g* is a decreasing function and *k* is the feature image index. In this implementation the following *g* is used

$$g(|\nabla u|) = \frac{1}{|\nabla u|}. \tag{2}$$

The result is stack of images each at a coarser scale known as a scale stack. Although this technique removes the within class variation it still retains the large response of the feature extraction algorithm to land-cover boundaries as shown in figure 2. To overcome this we propose to combine this algorithm with a procedure for establishing the exact location of texture edge and weight the diffusion amount based on these edges as apposed to the gradient magnitude image. The result is an image which has been smoothed in a nonlinear manner, where all locations apart from the locations of texture boundaries receive significant smoothing, thus the unwanted response at boundaries is reduced.

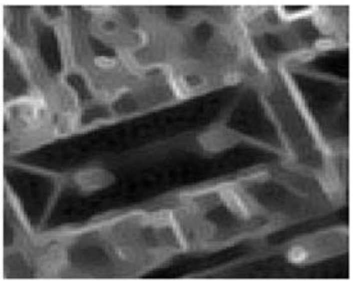


Figure 2: Within class variation is reduced but the large response to land-cover boundaries remains.

3.2.2 Detection of texture boundaries

In [9] Shao proposed a technique to detect the exact location of texture boundaries which he applied to remotely sensed data of an urban area producing very promising results. The edge strength is given by

$$a = tr\overline{\Gamma} \tag{3}$$

where

$$\overline{\Gamma} = G * (\nabla\nabla^T). \tag{4}$$

The size of the Gaussian *G* is chosen to be 5*x*5. A line of width 2*a* is detected and not two edges if the integration scale is chosen to be larger then 1.5*a*. Since a small 3*x*3 widow was used for feature extraction a 5*x*5 Gaussian is sufficiently large. To given a more exact location of the edges non-maximum suppression and hysteresis thresholding [22] is applied to *a*. In order to apply non-maximum suppression we first need to calculate the texture gradient direction. Shao proposes a measure of texture orientation but this measure applies a nonlinear operation to the gradients by squaring, thus losing the gradient sign. Using this approach texture orientation will only be returned in the range 0 to 90 degrees as opposed to the required 0 to 180 degrees, two texture orientations which are orthogonal will be assigned the same direction. We propose a method to calculate the texture orientation which overcomes this failing. Where u_x and u_y are gradient directions in the *x* and *y* directions respectively, the new texture gradient direction φ is given by

$$\varphi = G * \arctan\left({}^{u_y}\!/_{u_x} \right). \tag{5}$$

Hysteresis thresholding is applied to the image returned from the non-maximum suppression process giving a binary image of locations which have a high likelihood of being true texture boundaries. For all these locations the corresponding values in *a* are multiplied by *3/2* giving a large gradient response at the texture boundaries. Using this *a* the new diffusion algorithm is

$$\partial_t u_i = div\left(g\left(\sum_{k=1}^{M} a_k \right) \nabla u_i \right). \tag{6}$$

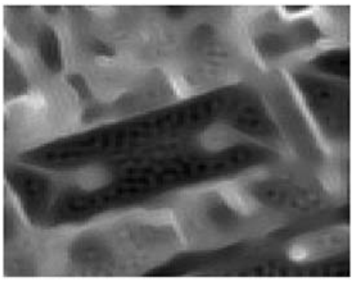
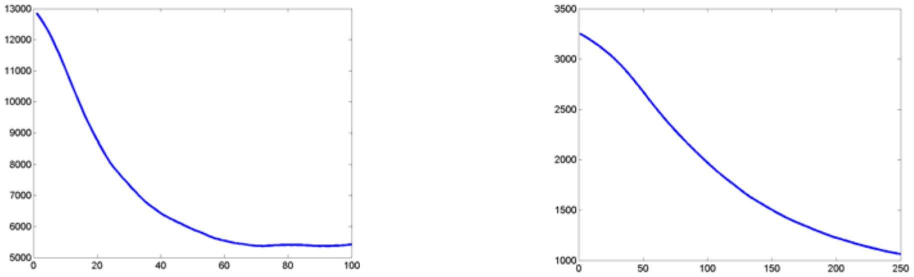


Figure 3: Result of the new texture gradient diffusion algorithm.

One of the issues when using any form of scale space is choosing at which scale the analysis should be performed. In this paper a convergence criteria is used to aid this choice. That is stopping when there is a small variation between two consecutives solutions of the evolution equations, corresponding to when the within class variation has been removed. For the anisotropy model the analysis is performed at scale 5 using a step size of 0.1, and for the isotropy the scale 15 was chosen also using a step size of 0.1.



(a)

(b)

Figure 4: Graphs used to aid the choice of scale to perform analysis (a) for the anisotropy model and (b) for the isotropy model. The x-axis represents scale and the y-axis is variation between two consecutive solutions.

3.3 Mean-shift clustering

To produce segmentation given the feature images a mean-shift clustering algorithm is applied [23]. This is a non-parametric algorithm which does not require knowledge of the number of clusters in the dataset and can detect clusters of varying shapes. The algorithm takes only one parameter *k* as input which controls the scale of clustering. The spatial locations of pixels are used as two additional features to aid the clustering giving a feature vector of length three for the anisotropy and six for the isotropy model. It was decided to give the texture and spatial features equal weight within each model by editing the distance metric used for clustering the anisotropy features. When clustering the isotropy features a traditional Euclidean distance metric is used, for the anisotropy features the following distance metric is used

$$d(x, y) = \sqrt{4(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}\ . \tag{7}$$

The anisotropy texture feature is represented by the first attribute, while attributes 2 and 3 are the spatial coordinates. All features are standardized prior to clustering. Clustering was run at a number of scales and segmentation results were evaluated. On average clustering with *k=80* produced best results.

3.4 Quantitative evaluation

Despite the significant advances in image segmentation algorithms, evaluation of such techniques tends to be largely subjective. In this study a relatively new quantitative evaluation technique known as the Normalized Probabilistic Rand (NPR) Index [24] was used. Segmentation is an inherently ill defined problem with no one ground truth to which segmentation can be compared. Rather then a comparison against a single ground truth, the comparison is made against the set of all possible perceptually consistent interpretations of the image, of which only a fraction is usually available. The algorithm quantifies the agreement of the segmentation with the inherent variation in a set of available ground truths.

For each image in the data set, five ground truths are generated by five individuals. The authors accept there is an issue of statistical significance due to the relatively small number of ground truths per image. At this moment there does not exist a public available database of remotely sensed image in urban areas and their corresponding ground truths such as the Berkeley segmentation dataset [25] for natural scenes.

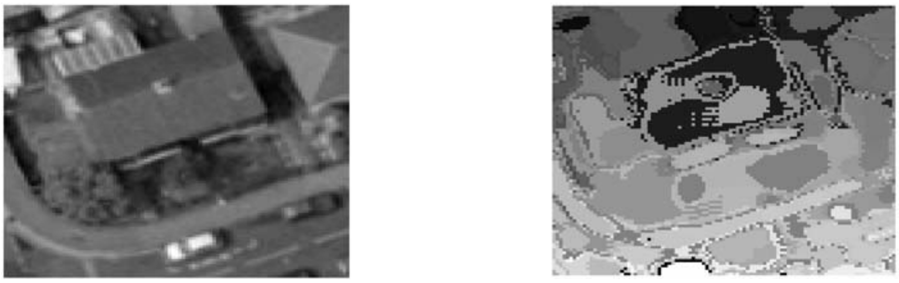
4 Results

For our data set we used five images of size 256*x*256 and for each of these five ground truths were captured. Using the methodology discussed above we evaluated anisotropy and isotropy texture feature extraction algorithms. Results are shown in table 1.

Model/Image	Image 1	Image 2	Image 3	Image 4	Image 5	Average
Isotropy	0.12	0.23	0.32	0.29	0.50	0.29
Anisotropy	0.13	0.23	0.31	0.30	0.50	0.29

Table 1: NPR Index for each model and image in dataset.

From the above we observe that the anisotropy and isotropy models both obtained almost identical results, each achieving an average index of 0.29 and a qualitative evaluation also confirmed this. Neither model returned a segmentation result of sufficient quality to be used as input to a land-use classification process as can be seen in figure 5. This suggests that accurate land-cover classification cannot be achieved using purely texture information. Intensity information must also be integrated to achieve a useful result.



(a) (b)

Figure 5: Image (a) shows an urban scene and its corresponding segmentation result achieved using the isotropy model is shown in (b). Ordnance Survey (c) Crown Copyright. All rights reserved.

5 Conclusions

Anisotropy is clearly a property of texture in images taken from texture databases. In this paper we perform an experimental evaluation to see whether this is the case for texture contained in aerial image of urban areas and results suggest that in fact it is not. No performance gain is achieved by the anisotropy model as opposed the isotropy model when performing texture feature extraction. If our results provide an accurate depiction then using an isotropy model provides a number of benefits without suffering a loss in performance. Isotropy features offer reduced dimensionality leading to faster computation time. As mentioned above isotropy features also facilitate a smaller moving window size for feature extraction, leading to a reduced edge effect at land-cover boundaries. This work also suggests that segmentation derived using texture is not sufficient and intensity features must also be integrated to achieve an accurate land-cover classification [8].

Acknowledgments: Support for this work is acknowledged to the Irish Research Council for Science Engineering and Technology (IRCSET). Ordnance Survey Southampton is acknowledged for providing data for use in this work.

References

[1] Blaschke, T. and G. Hay (2001), *What's wrong with pixels? Some recent developments interfacing remote sensing and GIS.* GIS - Zeitschrift fur Geoinformationssysteme, 14, 6, 12-17.

[2] Barnsley, M.J., L. Moller-Jensen, and S.L. Barr (2001), *Inferring urban land use by spatial and structural pattern recognition*, in *Remote sensing and urban analysis*, J.-P. Donnay, M.J. Barnsley, and P.A. Longley, Editors, Taylor & Francis.

[3] Frauman, E. and E. Wolff. (2005), *Segmentation of very high spatial resolution satellite images in urban areas for segments-based classification*, in *5th international symposium remote sensing of urban areas (URS 2005)*, Temple, AZ, USA,

[4] Blaschke, T. (2003), *Object-based contextual image classification built on image segmentation*, in *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data*, 113-119.

[5] Herold, M., L. X., and K.C. Clarke (2003), *Spatial metrics and image texture for mapping urban land use.* Photogrammetric engineering and remote sensing, 69, 9, 991-1001.

[6] Conners, R.W., M.M. Trivedi, and C.A. Harlow (1984), *Segmentation of a high-resolution urban scene using texture operators.* Computer graphics and image processing, 25, 273-310.

[7] Myint, S.W. and N. Lam (2005), *A study of lacunarity-based texture analysis approaches to improve urban image classification.* Computers, environment and urban systems, 29, 5, 501-523.

[8] Corcoran, P. and A. Winstanley. (2006), *Using Texture to Tackle the Problem of Scale in Land-Cover Classification*, in *International conference on object based image analysis*, Salzberg, In Press.

[9] Shao, J. and W. Forstner. (1994), *Gabor wavelets for texture edge extraction*, in *ISPRS commission III symposium on spatial information from digital photogrammetry and computer vision*, Munich, Germany, 745-752.

[10] Rousson, M., T. Brox, and R. Deriche. (2003), *Active unsupervised texture segmentation on a diffusion based feature space*, in *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, II-699-704.

[11] Julesz, B. (1983), *Textons, the fundamental elements in preattentive vision and perception of textures.* Bell system technical journal, 62, 1619-1645.

[12] Gamba, P., F. Dell'Acqua, and G. Trianni. (2004), *Automatic definition of scale feature for remote sensed image interpretation*, in *Proc. of the pattern recognition in remote sensing 2004 workshop*, Kingston-upon-thames,

[13] Acqua, F.D., P. Gamba, and G. Trianni (2006), *Semi-automatic choice of scale-dependent features for satellite SAR image classification.* Pattern Recognition Letters, 27, 4, 244-251.

[14] Chica-Olmo, M. and F.Abarca-Hernandez (2000), *Computing geostatistical image texture for remotely sensed data classification.* Computers and Geosciences, 26, 4, 373-383.

[15] Lark, R.M. (1996), *Geostatistical description of texture on an aerial photograph for discriminating classes of land cover.* International Journal of Remote Sensing, 17, 11, 2115-2133.

[16] Carr, J.R. and F.P.d. Miranda (1998), *The semivariogram in comparison to the co-occurrence matrix for classification of image texture.* IEEE Transactions on Geoscience and Remote Sensing, 36, 6, 1945-1952.

[17] Webster, R. and M.A. Oliver (1992), *Sample adequately to estimate variograms of soil properties.* Journal of soil science, 43, 177-192.

[18] Atkinson, P.M. and P. Aplin (2004), *Spatial variation in land cover and choice of spatial resolution for remote sensing.* International Journal of Remote Sensing, 25, 18, 3687-3702.

[19] Lloyd, C.D., P.M. Atkinson, and P. Aplin (2005), *Characterising local spatial variation in land cover imagery using geostatistical functions and the discrete wavelet transform.* GeoENV V: Geostatistics for environmental applications, in press,

[20] Chen, Q. and P. Gong (2004), *Automatic variogram parameter extraction for textural classification of the panchromatic IKONOS imagery.* IEEE Transactions on Geoscience and Remote Sensing, 42, 5, 1106-1115.

[21] Cressie, N. and D.M. Hawkins (1980), *Robust estimation of the variogram.* Mathematical Geology, 12, 115-125.

[22] Canny, J.F. (1986), *A computational approach to edge detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 8, 6, 679-698.

[23] Georgescu, B., I. Shimshoni, and P. Meer. (2003), *Mean shift based clustering in high dimensions: a texture classification example*, in *Ninth IEEE International Conference on Computer Vision Proceedings.*, 456-463.

[24] Unnikrishnan, R., C. Pantofaru, and M. Hebert. (2005), *A measure for objective evaluation of image segmentation algorithms*, in *Proceedings of the IEEE conference on computer vision and pattern recognition, workshop on empirical evaluation methods in computer vision*, San Diego,

[25] Martin, D., et al. (2001), *A database of human segmented natural images and its applications to evaluating segmentation algorithms and measuring ecological statistics*, in *International conference on computer vision*, Vancouver, 416-423.

Segmentation

Contour Based Methods for Segmentation of Low Contrast Images

Christopher Hudy,
Department of Computing, Letterkenny Institute of Technology,
Co. Donegal, Ireland;
E-mail: christopher.hudy@lyit.ie

Jonathan G. Campbell,
Department of Computing, Letterkenny Institute of Technology;

John Slater,
Department of Science, Letterkenny Institute of Technology,

29th June 2006

Abstract

In many image based object recognition activities, segmentation is a necessary preliminary step. In cases where there is low contrast between objects and background, segmentation can present significant difficulties. Occlusions are another significant problem. Microscopy images often present segmentation problems, namely low contrast (in the case of this paper, translucent objects) and occlusions. On the other hand, translucency gives some hope for solution of the occlusion problem. This paper describes experiments with a series of contour based methods aimed an segmentation of low contrast microscopy image of shellfish larvae. The methods studied are: (i) watershed; (ii) active contours; (iii) level sets; (iv) a novel boundary tracking method. We mention a possibly improved method of edge detection, namely phase congruency method of detecting edges and corners. Practical results are given.

Keywords: Machine Vision, segmentation, active contours, pattern recognition.

1 Background and Objective

The particular objective is to identify and count larvae species in microscopy images (Campbell, Slater, Gillespie, Bendezu & Murtagh 2005). Ideally the process would be fully automatic, however a more realistic but nonetheless useful outcome would be a semi-automatic method (Cutrona & Bonnet 2001). For example, via a graphical user interface a user might click on objects to be identified. In previous work (Campbell et al. 2005), segmentation was identified as a crucial step, yet one which presents considerable difficulty; in that work, segmentation required manual assistance.

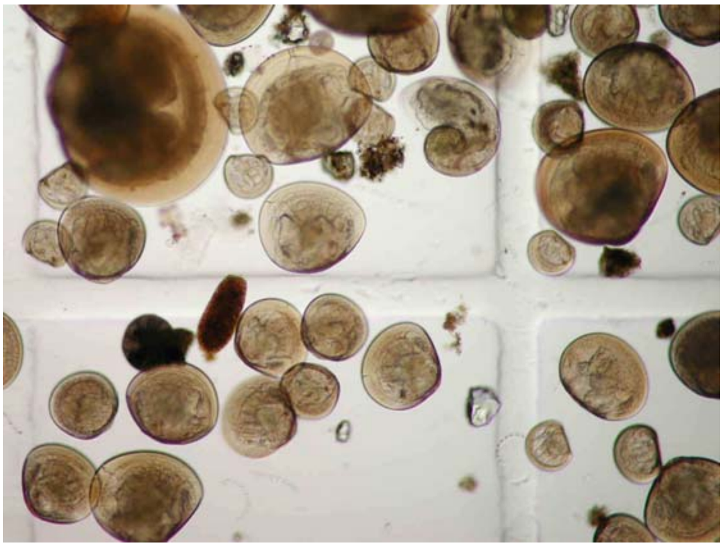


Figure 1: Raw image.

There is also a general objective to investigate segmentation and shape recognition methods applicable to microscope images. In much of the literature on microscopy image processing, for example (de Pontual, Robert & Miner 1998), it appears that the methodology is constrained by rather ineffective image processing packages. Consequently we attempt a limited survey of method suitable for this application domain.

In optical microscopy, objects are often translucent, that is, there is overlap of grey-level values of objects and the background, also, there are occlusions. An example exhibiting both translucency and occlusions shown in Fig. 2.

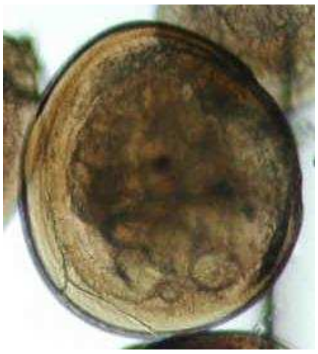


Figure 2: Occlusion example.

2 Methods

2.1 Watershed

The watershed method of Cutrona and Bonnet (Cutrona & Bonnet 2001) is based on seed points. Seed points are manually selected points, inside and outside of the object and are used to initialize the method; hence the method is only semi-automatic. The immersion of the gradient surface begins only from those points and not from all minima. The watershed segmentation process is as follows:

(i) Seed points are selected manually by the user (Fig. 3); in the case of the software used in this work; one point for each region and background. Obviously, the success of the method depends on proper selection of seed points.

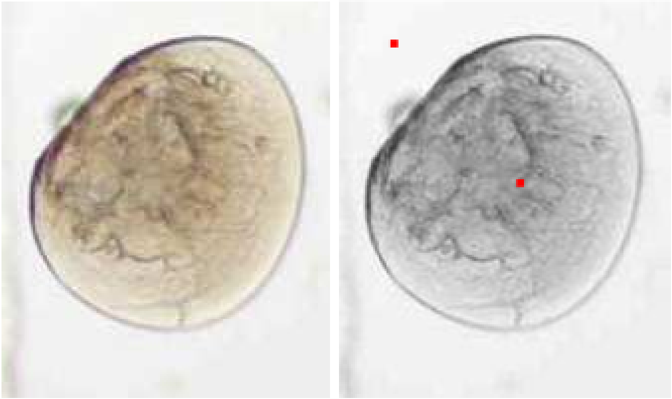


Figure 3: Watershed segmentation (i) raw data (ii) 8-bit image with seeded points (small dark rectangles).

(ii) First Gaussian smoothing is applied 4(i), then edge detection is applied Fig. 4(ii); either Shen-Castan (Shen & Castan 1992) or some standard gradient procedure. The Shen-Castan method is a development of the Canny method (Canny 1986).

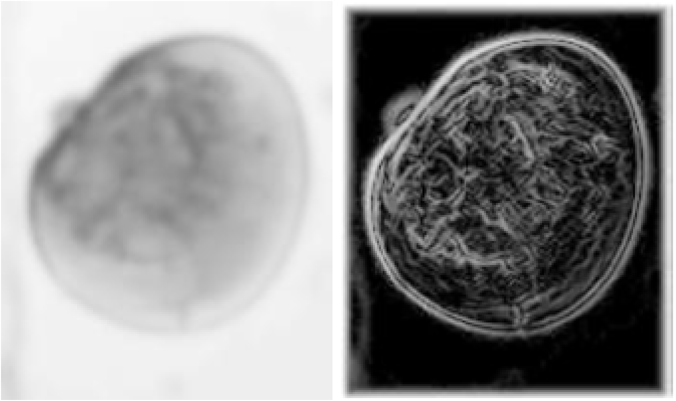


Figure 4: Watershed segmentation pre-processing (i) smoothing (ii) Shen-Castan edges.

(iii) The edge image is further processed with mathematical morphology dilation (Gonzalez & Woods 2002). The dilation effect is to gradually enlarge the boundaries of regions.

(iv) Sample results are shown in Fig. 5.

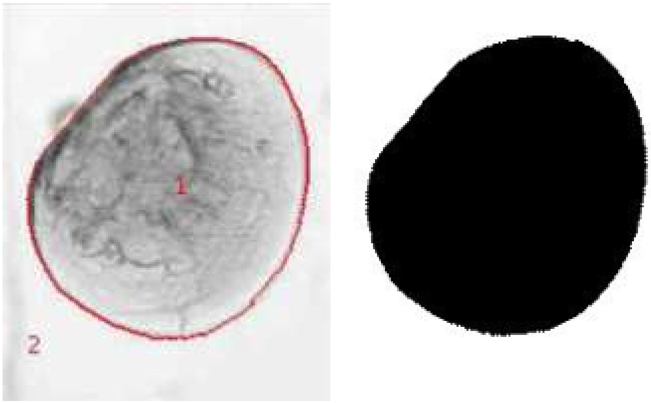


Figure 5: Watershed segmentation (i) watershed and regions (ii) segmented binary image.

2.2 Active Contours (Snakes)

Active contours (snakes) are used to locate shape features; the technique was introduced by (Kass, Witkin & Terzopoulos 1987). The physical analogy is to consider a snake as a flexible beam with energy that is composed of: an internal component which models the contour's smoothness and stiffness; and an external component which puts constrains on length and curvature.

For a contour that is defined by $c(s)$, $s \in [0,1]$, the total energy is given by:

$$E_{snake} = \underbrace{\int_0^1 \alpha(s)|c'(s)|^2 + \int_0^1 \beta(s)|c''(s)|^2}_{Internal\,Energy} - \underbrace{|\bigtriangledown I(x,y)|^2}_{External\,Energy} \tag{1}$$

where $\alpha(s)$ models lengths constrains, and $\beta(s)$ models curve smoothness.

The active contour process is shown below.

(i) An initial contour is selected (inside seems to give better results) as in Fig. 6;

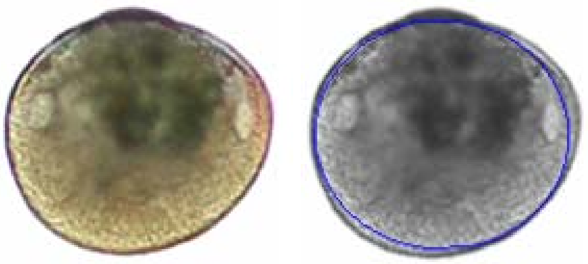


Figure 6: Active contour method: (i) raw image (ii) initialising contour

(ii) Next we select parameters: gradient threshold and regularization parameter for smoothness;

(iii) The edge detection is obtained by a Canny-Deriche filter (Canny 1986);

(iv) Results are shown in Fig. 7.

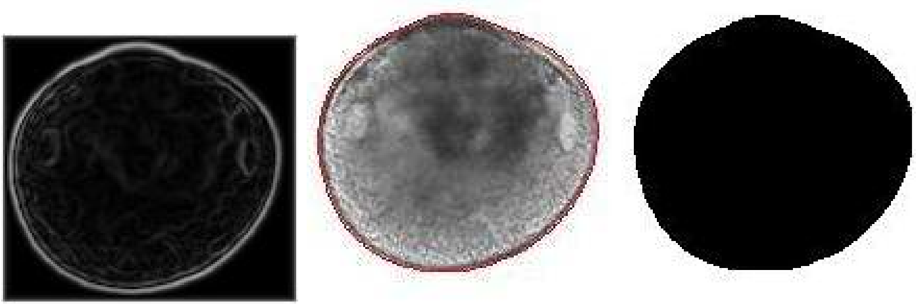


Figure 7: Active contour method: (i) edge image (ii) result image (iii) binary image

2.3 Level Set and Fast Marching Method

The level set method was introduced by Osher and Sethian (Osher & Sethian 1988) as a numerical technique for computing the position of propagating fronts and tracking the motion as the front evolves. The method embeds the curve in a higher dimension function ψ and represents the curve $C(t=0)$ as the zero level set ($\psi = 0$) of this function,

$$\psi(\mathbf{x}, t=0) = \pm d, \tag{2}$$

where d is the shortest distance from $\mathbf{x}$ to $C(t=0)$, the sign ± is to distinguish between inside and outside of the curve.

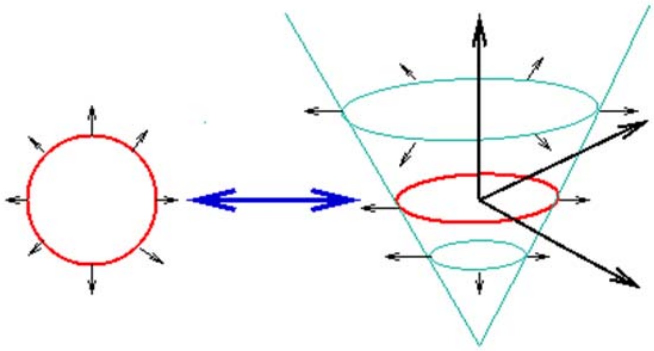


Figure 8: Propagating level set function $\psi(x,y,t)$

The evolution equation for ψ is:

$$\psi_t + F(\mathbf{x}(t))|\bigtriangledown\psi| = 0, \tag{3}$$

where $F$was defined by Osher and Sethian:

$$F(\mathbf{x}(t),t) = c + \kappa(\mathbf{x}(t),t), \tag{4}$$

and $\kappa(\mathbf{x}(t),t) = div\frac{\bigtriangledown\psi}{|\bigtriangledown\psi|}$ is the curvature of propagating front and c is a constant.

In segmentation applications the propagating front will stop when it approaches the desired object by meeting the following condition:

$$\psi_t + g(\triangledown I)(c + \kappa(\mathbf{x}(t),t))| \triangledown \psi| = 0, \tag{5}$$

where g is a smooth decreasing function. In the original paper it is:

$$g(\triangledown I) = \frac{1}{1 + | \triangledown I \star G_\sigma|},$$

where $\star$ in term: $1 + | \triangledown I \star G_\sigma|$ denotes convolution of gradient image $\triangledown I$ with a Gaussian kernel parametrised by σ.

The level set equation eqn. 5 is solved using eqn. 6.

$$c| \triangledown \psi| \approx max(c_{ij}, 0) \triangledown^+ + min(c_{ij}, 0) \triangledown^-, \tag{6}$$

where:

$$\triangledown^+ = \left[max(D_{ij}^{-x}, 0)^2 + min(D_{ij}^{+x}, 0)^2 + max(D_{ij}^{-y}, 0)^2 + min(D_{ij}^{+y}, 0)^2 \right]^{\frac{1}{2}}, \tag{7}$$

$$\triangledown^- = \left[max(D_{ij}^{+x}, 0)^2 + min(D_{ij}^{-x}, 0)^2 + max(D_{ij}^{+y}, 0)^2 + min(D_{ij}^{-y}, 0)^2 \right]^{\frac{1}{2}}. \tag{8}$$

Fig. 9 displays implementation of fast marching method for segmentation of shellfish larvae, results are after 11000, 21000, 23000 iterations respectively.

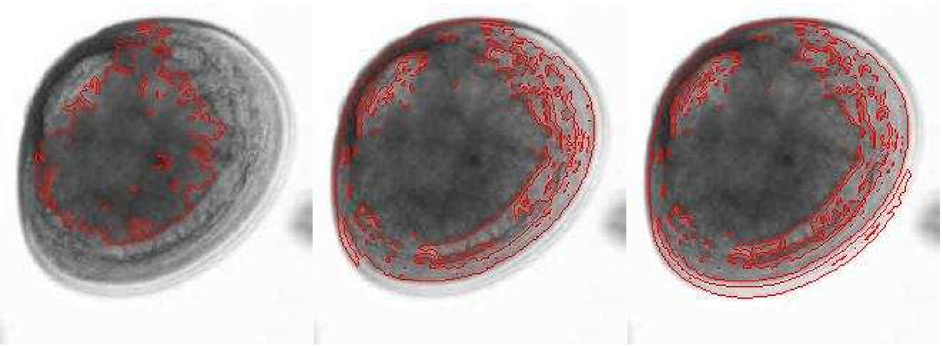


Figure 9: Fast Marching method: (i) 11000 iterations; (ii) 21000 iterations; (iii) 23000 iterations.

We can conclude that, in this form, level set methods are unsuitable for objects like our larvae; this is primarily because of the lack of homogeneity within the objects and the related problem of translucency and occlusion.

2.4 Phase Congruency

The phase congruency edge detection (Kovesi 1999) is reported to be less sensitive to problems of variable contrast and noise level and to provide better feature localisation; the latter feature is attractive because of our desire to limit the occurrence of double edges. In normal edge detection, variable edge strength (see, for example, Fig. 2) can be a significant problem because of the need to set a fixed threshold. In addition phase congruency may provide better indication of edge orientation. Detailed description of the method are given in (Kovesi 1999) and (Kovesi 2002). An example of phase congruency edge detection is shown in Fig. 10.

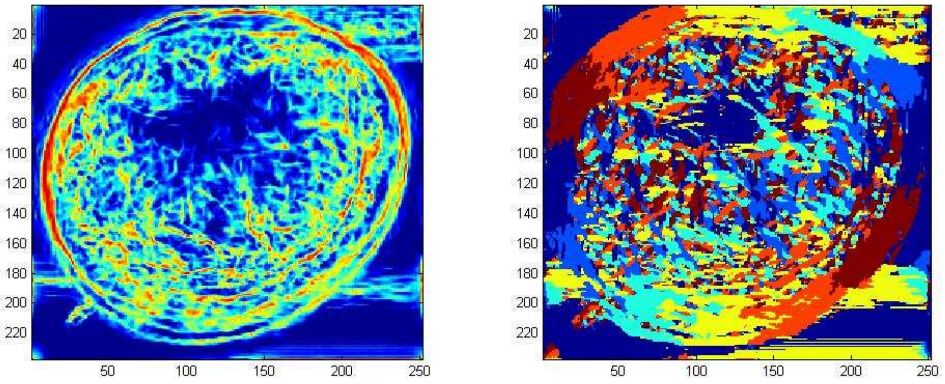


Figure 10: Phase Congruency (i) phase congruency edge image, (ii) edge orientation.

2.5 Boundary Contour Tracking on Edges

Boundary tracking is introduced to cope with: (i) breaks in edges caused by poor contrast or by occlusions and (ii) double edges.

The method tracks edge points that were obtained from the non-maxima suppressed, gradient magnitude image. Only one neighbour pixel is selected which best satisfies gradient orientation and distance criteria. The tracking process is applied in polar coordinates and is continued until angular coordinates component reaches 2π, i.e. the contour is closed. Fig. 11 shows the results of detection of object boundary using edge tracking methods (Hudy 2006).

We note that our edge tracking is dependent on the smoothness and broadly circular shapes of our objects; of course, with appropriate parameters, it could cope with any shape.

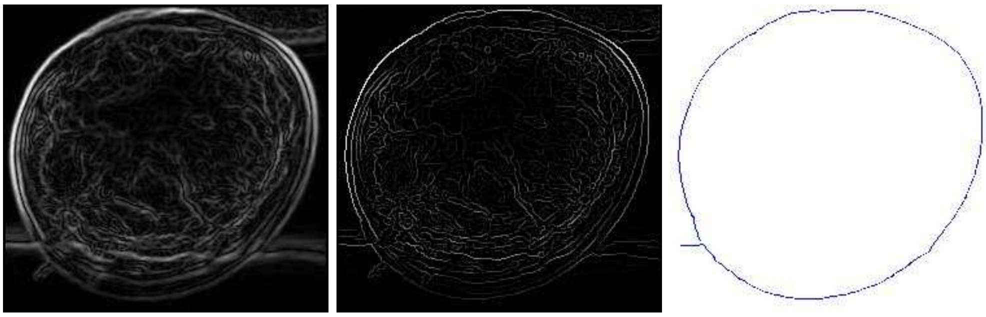


Figure 11: Boundary tracking; can cope with weak edges; also with some cases of occlusions.

3 Conclusions

This paper has described experiments on contour based methods aimed an segmentation of low contrast microscopy images. The methods studied where: (i) watershed; (ii) active contours; (iii) level sets; (iv) phase congruency edge detection; (v) a novel boundary tracking method.

Results show that the watershed method perform well on object with strong edges; the method does require initialisation of seed points but even with this requirement for manual input the method could be operationally effective.

Active contour perform similarly to watershed method, but with disadvantage of requiring more intricate initialisation

It seems that the lack of level sets method is primarily because of the lack of homogeneity within the objects and the related problem of translucency and occlusion.

Our novel edge detection method is very promising and appears capable of remedying many of the before mentioned difficulties, such as variable strength edges, double edges, and some occlusions.

Future work will adapt the edge tracking method to incorporate the supposedly superior edge orientation information provided by phase congruency.

Acknowledgements. We acknowledge the invaluable comments of two anonymous referees.

References

Campbell, J., Slater, J., Gillespie, J., Bendezu, I. & Murtagh, F. (2005). Pattern recognition methods for identification of shellfish larvae, *In Proceedings IMVIP 2005, Irish Machine Vision and Image Processing Conference 2005* pp. 97–104.

Canny, J. (1986). A computational approach to edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence* **8(6)**: 679–698.

Cutrona, J. & Bonnet, N. (2001). Two methods for semi-automatic image segmentation based on fuzzy connectedness and watersheds.

de Pontual, H., Robert, R. & Miner, P. (1998). Study of Bivalve Larval Growth using Image Processing, *Aquacultural Engineering* **17**: 85–94.

Gonzalez, R. & Woods, R. (2002). *Digital Image Processing*, 2nd edn, Prentice Hall.

Hudy, C. (2006). Segmentation of low contrast images, *Letterkenny IT internal report* .

Kass, M., Witkin, A. & Terzopoulos, D. (1987). Snakes: Active Contour Models, *International Journal of Computer Vison* **1**: 321–331.

Kovesi, P. (1999). Image feature from phase congruency, *Journal of Computer Vision Research* **1**: 1–26.

Kovesi, P. (2002). Edges are not just steps, *Asian Conference on Computer Vision* .

Osher, S. & Sethian, J. A. (1988). Front propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, *Journal of Computational Physics* **79(1)**: 12–49.

Shen, J. & Castan, S. (1992). An optimal linear operator for step edge detection, *CVGIP: Graphical Models and Image Processing* **54**: 112–133.

Color Image Segmentation Using A Spatial K-Means Clustering Algorithm

Dana Elena Ilea and Paul F. Whelan

Vision Systems Group
School of Electronic Engineering
Dublin City University
Dublin 9, Ireland
danailea@eeng.dcu.ie

Abstract

This paper details the implementation of a new adaptive technique for color-texture segmentation that is a generalization of the standard K-Means algorithm. The standard K-Means algorithm produces accurate segmentation results only when applied to images defined by homogenous regions with respect to texture and color since no local constraints are applied to impose spatial continuity. In addition, the initialization of the K-Means algorithm is problematic and usually the initial cluster centers are randomly picked. In this paper we detail the implementation of a novel technique to select the dominant colors from the input image using the information from the color histograms. The main contribution of this work is the generalization of the K-Means algorithm that includes the primary features that describe the color smoothness and texture complexity in the process of pixel assignment. The resulting color segmentation scheme has been applied to a large number of natural images and the experimental data indicates the robustness of the new developed segmentation algorithm.

Keywords: Clustering, color extraction, diffusion filtering.

1 Introduction

Image segmentation is one of the most important precursors for image processing–based applications and has a crucial impact on the overall performance of the developed systems. Robust segmentation has been the subject of research for many years, but the published work indicates that most of the developed image segmentation algorithms have been designed in conjunction with particular applications. The aim of the segmentation process consists of dividing the input image into several disjoint regions with similar characteristics such as color and texture. Robust image segmentation is a difficult task since often the scene objects are defined by image regions with non-homogenous texture and color characteristics and in order to divide the input image into semantically meaningful regions many developed algorithms either use *a priori* knowledge in regard to the scene objects or employ the parameter estimation for local texture [1]. The development of texture alone approaches proved to be limited and the use of color information in the development of joint color-texture models has led to the development of more robust and generic segmentation algorithms [2,3,4]. The area of color image analysis is one of the most active topics of research and a large number of color-driven segmentation techniques have been proposed. Most representative color segmentation techniques include histogram-based segmentation, probabilistic space partitioning and clustering [5], region growing, Markov random field and simulated annealing [6]. All these techniques have the aim to reduce the number of color components from the input image into a reduced number of components in the color segmented image that are strongly related to the image objects.

In this paper we have developed a new technique that is a generalization of the standard K-Means clustering technique. The K-Means clustering technique is a well-known approach that has been applied to solve low-level image segmentation tasks. This clustering algorithm is convergent and its aim is to optimize the partitioning decisions based on a user-defined initial set of clusters that is updated after each iteration. This procedure is computationally efficient and can be applied to multidimensional data but in general the results are meaningful only if homogenous non-textured color regions define the image data. The applications of the clustering algorithms to the segmentation of complex color-textured images is restricted by two problems. The first problem is generated by the starting condition (the initialization of the initial cluster centers), while the second is generated by the fact that no spatial (regional) cohesion is applied during the space partitioning process. In this paper we developed a new space-partitioning scheme that addresses both these limitations.

The selection of initial cluster centers is very important since this prevents the clustering algorithm to converge to local minima, hence producing erroneous decisions. The most common initialization procedure selects the initial cluster centers randomly from input data. This procedure is far from optimal because does not eliminate the problem of converging to local minima and in addition the segmentation results will be different any time the algorithm is applied. To circumvent this problem some authors applied the clustering algorithms in a nested sequence but the experimental data indicated that this solution is not any better than the random initialization procedure. In this paper we propose a different approach to select the cluster centers by extracting the dominant colors from the color histograms. The developed procedure is generic and proved to be very efficient when applied to a large number of images. There are other initialization schemes proposed in the literature and for more details the reader can refer to [7].

The second limitation associated with the K-Means (and in general clustering algorithms) is generated by the fact that during the space partitioning process the algorithm does not take into consideration the local connections between the data points (color components of each pixel) and its neighbors. This fact will restrict the application of clustering algorithms to complex color-textured images since the segmented output will be over-segmented. To address this issue we have generalized the K-Means algorithm to evaluate along with the pixel color information two more distributions that sample the local color smoothness and the local texture complexity. In this regard to sample the local color smoothness, the image is filtered with an adaptive diffusion scheme while the local texture complexity is sampled by filtering the input image with a gradient operator. Thus, during the space partitioning process, the developed algorithm attempts to optimize the fitting of the diffusion-gradient distributions in a local neighborhood around the pixels under analysis with the diffusion (color)-gradient distributions for each cluster. This process is iteratively applied until convergence is reached. We have applied the developed spatial clustering algorithm on a large selection of images with different level of texture complexity and on test data that has been artificially corrupted with noise.

2 K-Means Algorithm

In general, spatial partitioning methods are implemented using iterative frameworks that either attempt to minimize the variation within the clusters or attempt to identify the optimal partitions based on a set of Gaussian Mixture Models. In this paper we focus on the implementation of the K-Means algorithm, although the methodology detailed in this paper can be applied to other clustering schemes such as fuzzy clustering [8] or competitive agglomerative clustering [9].

The K-Means is a nonhierarchical clustering technique that follows a simple procedure to classify a given data set through a certain number of K clusters that are known *a priori*. The K-Means algorithm updates the space partition of the input data iteratively, where the elements of the data are exchanged between clusters based on a predefined metric (typically the Euclidian distance between the cluster centers and the vector under analysis) in order to satisfy the criteria of minimizing the variation within each cluster and maximizing the variation between the resulting K clusters. The algorithm is iterated until no elements are exchanged between clusters. This clustering algorithm, in its standard formulation consists mainly of four steps that are briefly described below:

Steps of the classical K-Means clustering algorithm:

1. Initialization – generate the starting condition by defining the number of clusters and randomly select the initial cluster centers.
2. Generate a new partition by assigning each data point to the nearest cluster center.
3. Recalculate the centers for clusters receiving new data points and for clusters losing data points.
4. Repeat the steps 2 and 3 until a distance convergence criterion is met.

As mentioned before, the aim of the K-Means is the minimization of an objective function that samples the closeness between the data points and the cluster centers, and is calculated as follows:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{1}$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is the distance (usually the Euclidian metric) between the data point $x_i^{(j)}$ and the cluster center c_j. As it can be easily observed in equation 1, the assignment of the data points may not be unique (a data point can be equally distanced from two or more cluster centers) a case when the K-Means algorithm doesn't find the optimal solution corresponding to the global objective function *J*. In addition, it is sensitive to the initialisation process that selects the initial cluster centers (usually randomly picked from input data). If the initial cluster centers are initialised on outliers, the algorithm will converge to local minima and this is one of the major drwbacks of this space partitioning technique. More importantly this algorithm does not produce meaningful results when applied to noisy data or to tasks such as the segmentation of complex textured images or images affected by uneven illumination. Since the pixel assignment is performed only by evaluating the color information in a certain color space, the connection between the data point under evaluation and its neighbours is not taken into account, a fact that will lead to a partition of the input data into regions that are not related to the scene objects. In the remainder of this paper we will detail a histogram based procedure used to select the dominant colors from input data and the development of a new data assignment strategy that evaluates not only the pixel's color components but also the local color-texture complexity, which allows us to obtain homogenous clusters.

3 Automatic Seed Generation

Since the random selection of the initial cluster centers from image data is not an appropriate solution, we have developed a new scheme to perform the initialization for the K-Means algorithm with the dominant color components that are extracted from the color histograms of the input image. In this regard, we have constructed the histogram for each color channel and partitioned them linearly into *R* sections (where *R* is a fixed value, *R*>*K* and n_k is the number of pixels contained in the bin *k*) and for each section of the histograms is determined the bin that has the highest number of elements:

$$P_j = \arg \max_{j \in [1,R]} (n_k) \tag{2}$$

We continue with ranking the peaks obtained from color histograms in agreement with the number of elements that are sorted in descending order. Finally, we form the color seeds (dominant colors) starting with the histogram peak that has the highest number of elements. This process can be summarized by the following pseudo-code sequence:

```
1. Construct the histograms for each color channel
2. Partition each histogram into R sections
3. Compute the peaks in each section and rank the peaks, $p_1, p_2, \ldots, p_R$ where $p_1$ has the
   highest number of elements
4. Start to form the color seeds for highest peak $p_i$
     if($p_i \rightarrow$red) mark the pixels in the red channel and calculate the $g_{mean}$ and $b_{mean}$ for
          marked pixels from green and blue channels
     if($p_i \rightarrow$green) mark the pixels in the green channel and calculate the $r_{mean}$ and $b_{mean}$ for
          marked pixels from red and blue channels
     if($p_i \rightarrow$blue) mark the pixels in the blue channel and calculate the $r_{mean}$ and $g_{mean}$ for
          marked pixels from red and green channels
5. Form the color seed and eliminate $p_i$ from the list
6. Repeat the steps 4 and 5 until the desired number of color seeds has been reached.
```

4 Diffusion-based Filtering

As it was mentioned in Section 3, one of our aims is to sample the local color smoothness. This can be achieved by filtering the data with a smoothing operator that eliminates the weak textures. The standard linear smoothing filtering schemes based on Gaussian weighted spatial operators or non-linear filters such as the median, reduce the level of noise but this advantage is obtained at the expense of poor feature preservation (i.e. suppression of narrow details in the image). To circumvent this problem we have developed an adaptive diffusion based filtering scheme that was originally developed by Perona and Malik with the purpose of implementing an optimal, feature preserving smoothing strategy [10]. In their paper, smoothing is formulated as a diffusive process and is performed within the image regions and suppressed at the regions boundaries. This non-linear smoothing procedure can be defined in terms of the derivative of the flux function that is illustrated in equation 3.

$$u_t = div(D(|\nabla u|)\nabla u) \tag{3}$$

where u is the input data, D represents the diffusion function and t indicates the iteration step. The smoothing strategy described in equation 3 can be implemented using an iterative formulation as follows:

$$I_{x,y}^{t+1} = I_{x,y}^{t} + \lambda \sum_{j=1}^{4} [D(\nabla_j I)\nabla_j I] \tag{4}$$

$$D(\nabla I) = e^{-\left(\frac{\nabla I}{d}\right)^2} \in (0,1] \tag{5}$$

where $\nabla_j I$ is the gradient operator defined in a 4-connected neighborhood, λ is the contrast operator that is set in the range $0<\lambda<0.16$ and d is the diffusion parameter that controls the smoothing level. It should be noted that in cases where the gradient has high values, the value for diffusion function $D(\nabla I) \rightarrow 0$ and the smoothing process is halted.

5 Spatial K-Means Clustering Algorithm (S-KM)

One of the main problems associated with standard clustering algorithms is the lack of using any spatial continuity with respect to the local texture and color information in the space partitioning process. Thus the application of these algorithms is restricted to input images that are defined by homogenous color regions. Our aim is to develop a space-partitioning algorithm that is able to return meaningful results even when applied to complex natural scenes that exhibit large variation in color and texture. In order to produce accurate non-fragmented segmented results, we need to sample the homogeneity of a color-texture descriptor in a given region. Nonetheless the robust evaluation of the texture is a very difficult task, since the texture is not constant within the image and this would require complex models to describe it at micro and macro level. As we are interested in evaluating the complexity of the image locally, we reformulate the problem since we do not need precise models for texture and rather the evaluation of the local image complexity in a data with a reduced number of color components. Using these assumptions, a large number of algorithms have been developed to address the problem of robust segmentation of complex images and the most representative are the mean shift [2], adaptive clustering algorithm [3] and the extension of the diffusion based algorithms [10]. Our algorithm is novel, since it attempts to minimize the errors in the assignment of the data points into clusters by evaluating the local texture complexity using two measures that constrain the region intensity (color smoothness) and the spatial continuity (texture complexity). In this regard, the clustering algorithms are perfectly suited to perform this task and the main difficulty resides in the selection of optimal features that are able to produce clusters with spatial homogeneity. In addition, these measures have to be sufficiently generic, to be able to accommodate the local variation in texture scale and the local variation in color. To address these problems we propose to use two types of descriptors along with color information. To sample the local color homogeneity, we evaluate the distribution of the data in the image resulting after the application of the diffusion filtering (see equation 4). The local texture complexity is sampled by the gradient data that is calculated using the Laplace operator that has been chosen for its low computational cost and its omni-directional properties (this assures immunity to texture rotation). Thus, the process of space partitioning is modified to accommodate these two distributions that are calculated as follows:

$$H_{(n,m)} = \bigcup_{i \in C} h(i), \quad h(i) = \sum_{y=-n}^{n} \sum_{x=-m}^{m} \delta(f(x,y), i),$$

$$\text{where } C \text{ is the color space, } \delta(i,j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{6}$$

Using the equation 6, we calculate the distributions from diffused and gradient images in the local neighborhood (n, m) for each data point. The global objective function depicted in equation 1 is modified to accommodate the diffusion-gradient distributions as follows:

$$J = \sum_{j=1}^{K} \sum_{i=1}^{n} \left[\left\| x_i^{(j)} - c_j \right\| + KS(H_{Diff_i}^{(j)}, H_{Diff}^{(j)}) + KS(H_{Grad_i}^{(j)}, H_{Grad}^{(j)}) \right] \tag{7}$$

where $H_{Diff_i}^{(j)}$ is the local color smoothness distribution (calculated from diffused image) for the data point i, $H_{Diff}^{(j)}$ is the color smoothness distribution for cluster j, $H_{Grad_i}^{(j)}$ is the local texture complexity (calculated from the gradient data) for the data point i, $H_{Grad}^{(j)}$ is the texture complexity for cluster j, and KS is the Kolmogorov-Smirnov metric that is calculated using the following expression:

$$KS(H_a, H_b) = \sum_{i \in C} \left| \frac{h_a(i)}{n_a} - \frac{h_b(i)}{n_b} \right| \tag{8}$$

where the n_a and n_b are the number of data points in the distribution H_a and H_b respectively. The KS similarity measure is bounded in the interval [0,2] and we have readjusted the metric $\left\| x_i^{(j)} - c_j \right\|$ to be also bounded (normalization with respect to the maximum value of the respective color space, i.e. 255 for RGB). The $H_{Diff}^{(j)}$ and $H_{Grad}^{(j)}$ distributions are recalculated after each iteration. It is important to note that during the space partitioning process (minimization of the objective function illustrated in equation 7) the number of clusters are reduced, since some clusters from the initial set will disappear as the clusters become more compact after each iteration. The developed algorithm is convergent and to improve the computational overhead we have applied the K-Means algorithm in the standard form for the first 5 iterations and then the spatial and color continuity constraints $H_{Diff}^{(j)}$ and $H_{Grad}^{(j)}$ are evaluated. This approach also improve the stability of the algorithm, since the $H_{Diff}^{(j)}$ and $H_{Grad}^{(j)}$ distributions can be calculated only after the algorithm executes at least one iteration.

6 Experiments and Results

In this section we examine the performance of the developed algorithm on a large number of images. The first test is performed on a synthetic color image that is corrupted with noise (standard deviation 30 grayscale values – see Figure 1). To assess the efficiency of our method, we compare the results against the segmentation results returned by the mean shift algorithm, which is widely accepted as the standard color segmentation framework. Additional results are illustrated in Figures 2 and 3, where the algorithm has been applied to a complex natural image and a natural image defined by a low signal to noise ratio. It can be observed that our algorithm produces better visual results than the mean shift algorithm. The parameters for our method are set as follows: initial number of clusters = 10 and diffusion parameter = 30 (all images). The parameters for Comaniciu-Meer algorithm (Edison implementation) are set as follows: spatial filter=8, color=50 and minimum region size=200 when applied to the noisy image illustrated in Figure 1(a), spatial filter=7, color=6.5 and minimum region size=20 when the algorithm was applied to the image depicted in Figure 2(a) and spatial filter=7, color=15 and minimum region size=20 for the image depicted in Figure 3(a). The parameters for mean-shift algorithm are selected to generate the best results.

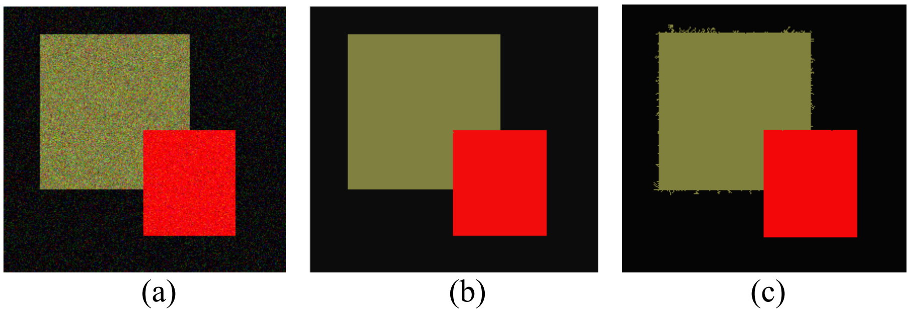


Figure 1. Segmentation results on a test image. (a) Test image corrupted with Gaussian noise (standard deviation 30 grayscales). (b) Segmented result – our algorithm (final number of regions=3). (c) Segmented result –mean shift algorithm (final number of regions=3).

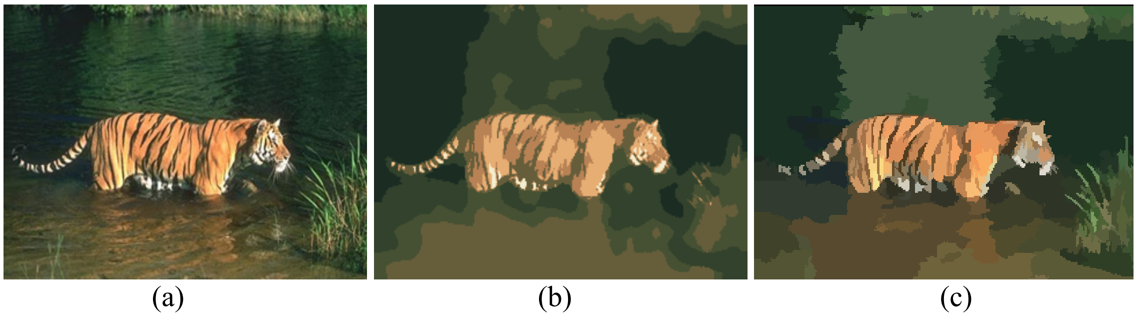


Figure 2. Segmentation results. (a) Input image. (b) Segmented result- our algorithm (final no of regions=7). (c) Segmented result – mean shift algorithm (final no of regions=223).

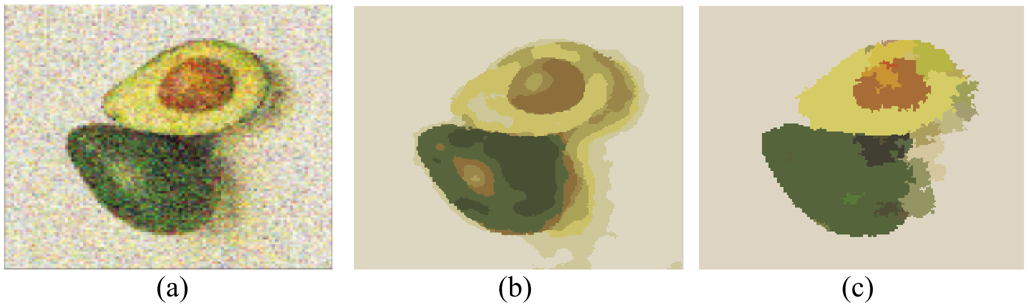
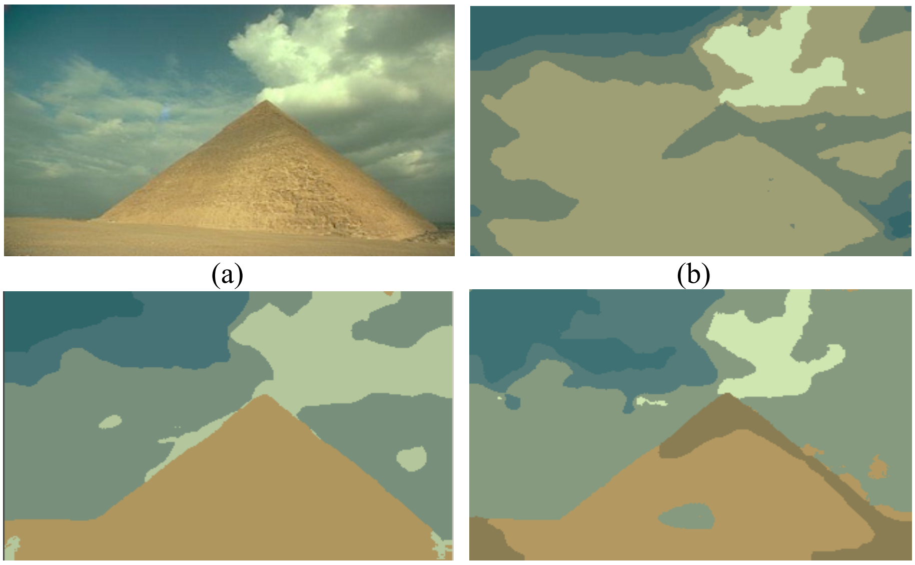


Figure 3. Segmentation results for a low-resolution natural image. (a) Input image. (b) Segmented result – our algorithm (final number of regions =6). (c) Segmented result – mean shift algorithm (final number of regions=27).

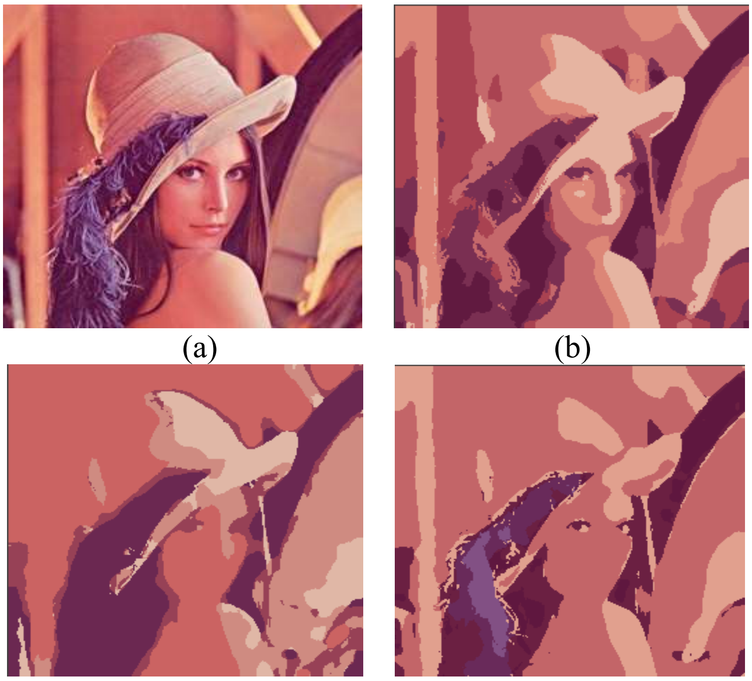
Another aim is to evaluate the behavior of our algorithm with respect to different color spaces. As expected, the RGB color space is non-linear and does not provide optimal color separation and our experiments indicate that better segmentation is achieved when the color images are converted to YIQ and HSI color spaces. Figures 4 and 5 depict the segmentation results when our algorithm has been applied to two test images. Our experiments indicate that best segmentation is obtained when the images are converted to YIQ color space. To fully evaluate the performance of our algorithm we evaluated the segmentation error for a natural image and for its version corrupted with additional noise (see Figure 6) when compared to the ground truth (manual segmentation), while the parameters of the algorithm (for this implementation the diffusion and gradient distributions are calculated within a square window, n=m) are varied. From the graphs illustrated in Figure 7, it can be noted the good stability of the developed algorithm when the parameters are varied. From these graphs it can also be observed that the segmentation error tends to be reduced with the increase of the diffusion parameter and this is generated by the fact that the smoothing is more pronounced and this translates into a more discriminative power for diffusion distribution in equation 7. The selection of the optimal window size is a difficult problem since a large window increases the discriminative power of the H_{Diff} *and* H_{Grad} distributions, but on the other hand increases the errors around the cluster borders. The optimal trade-off is achieved when the window size is set in the range [7×7, 11×11].



(a) (b)

(c) (d)

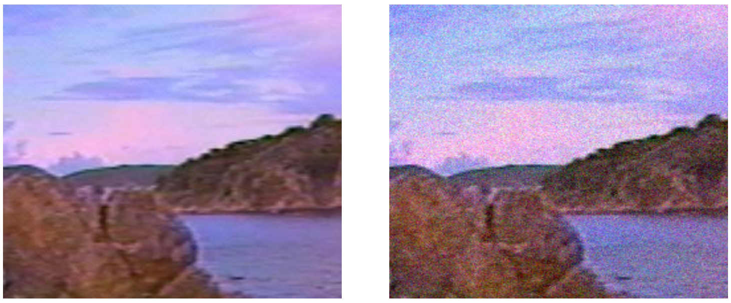
Figure 4. Segmentation results. (a) Test image. (b) Segmented result – RGB color space. (c) Segmented result – YIQ color space. (d) Segmented result – HSI color space.



(a) (b)

(c) (d)

Figure 5. Segmentation results. (a) Test image. (b) Segmented result – RGB color space. (c) Segmented result – YIQ color space. (d) Segmented result – HSI color space.



(a) (b)

Figure 6. Test images used in the evaluation of the developed algorithm. (a) Test image with low resolution. (b) Image corrupted with additional noise (standard deviation 30 grayscales).

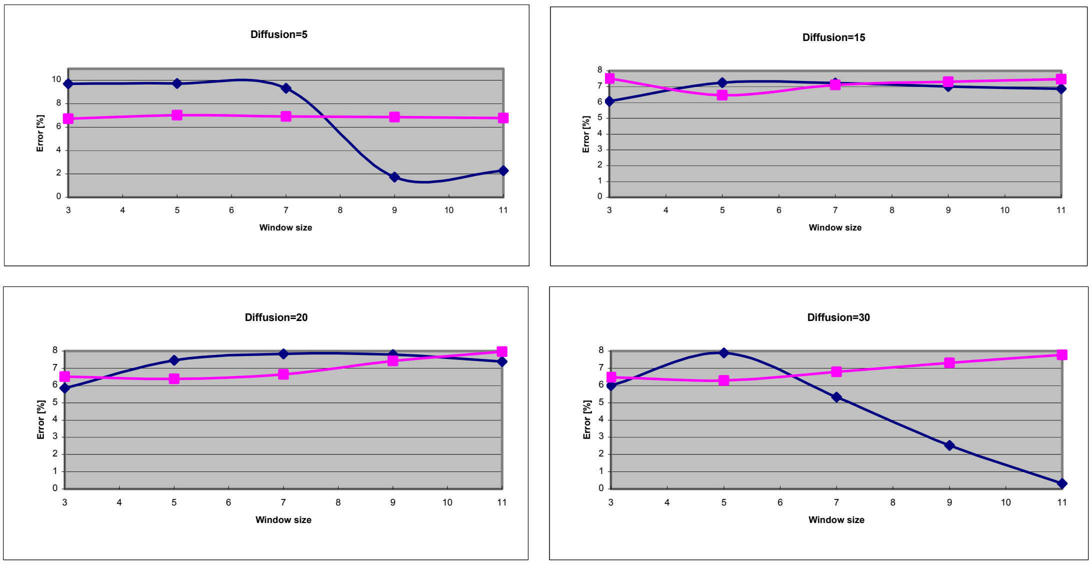


Figure 7. Performance of the developed algorithm when the diffusion and window size parameters are varied (blue dark noiseless image-Figure 6(a), purple noisy image-Figure 6(b)).

7 Conclusions

The aim of this work is the development of a new color segmentation algorithm, which is a generalization of the K-Means clustering algorithm. In its standard form the application of the K-Means algorithm to the segmentation of natural images is hindered by the fact that no constraints with respect to texture complexity or color continuity are employed during the space partitioning process. By reformulating the objective of the clustering process, we have developed a spatial constrained clustering algorithm that is found to be a powerful technique to identify continuous clusters in the color images that are strongly related with the scene objects. The developed algorithm has been tested and evaluated on a large number of natural images. The experimental data indicates that our algorithm is generic and it is robust to changes in local texture and produces accurate results even when applied to images characterized by a low signal to noise ratio.

Acknowledgments

We would like to express our gratitude to Science Foundation Ireland for supporting this research.

References

[1] Deng Y. and Manjunath B.S. (2001). Unsupervised segmentation of color-texture regions in images and video, *IEEE Trans. Pattern Analysis Machine Intell*, 23(8): 800-810.

[2] Comaniciu D. and Meer P. (2002). Mean shift: A robust approach toward feature space analysis", *IEEE Trans. Pattern Analysis Machine Intell.*, 24(5): 603-619.

[3] Pappas T. (1992). An adaptive clustering algorithm for image segmentation, *IEEE Trans. Signal Process*, 40:901-914.

[4] Drimbarean A. and Whelan P.F. (2001) Experiments in color texture analysis, *Pattern Recognition Letters*, 22: 1161-1167.

[5] Zöller H.T. and Buhmann J.M. (2002). Parametric distributional clustering for image segmentation, *7th European Conference on Computer Vision*, Denmark, pp. 577-591.

[6] Mukherjee J. (2002). MRF clustering for segmentation of color images, *Pattern Recognition Letters*, 23(8): 917-929.

[7] Khan S. and Ahmad A., (2004). Cluster center initialization algorithm for K-Means clustering, *Pattern Recognition Letters*, 25(11): 1293-1302.

[8] Jain A.K. and Dubes R.C. (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ.

[9] Frigui H. and Krishnapuram R. (1997). Clustering by competitive agglomeration, *Pattern Recognition*, 30(7): 1109-1119.

[10] Perona P. and Malik J. (1990). Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Analysis Machine Intell*, 12(7): 629-639.

Applications, Architectures & Systems Integration

A Preliminary Investigation into an Intelligent Car Headlight Dipping System

Sonya Coleman, Liam McDaid and Bryan Gardiner
School of Computing and Intelligent Systems
University of Ulster,
Northland Road, Londonderry, BT48 7JL
{sa.coleman, lj.mcdaid, gardiner-B}@ulster.ac.uk

Abstract

Current vision systems for driver assistance are now a huge focus within car manufacturing companies. Such systems use sensor techniques to detect moving objects, such as pedestrians, or cameras where 360^{o} vision is possible using front and rear cameras. The main advantage of using cameras is that visual data is crucial to the detection of moving objects within lanes, the recognition of traffic signs, pedestrians and of particular interest here, car headlights. However, a reliable vision based driver assistance system using even the most sophisticated vision system is extremely difficult as vehicles and objects vary in shape, size and colour and outdoor environments can be very complex. An essential feature of car vision systems is real time recognition of objects and environments. This paper presents a preliminary study into the development of a real-time automatic car headlight dimming using an appropriate combination of image analysis, neural networks topologies and training paradigms.

Keywords: Car Vision, image analysis, neural networks

1 Introduction

The long term goal of research into driver assistance is the development of robust and general purpose vision systems with human level visual capabilities [3, 22]. The majority of the issues or problems related to attempts to build general purpose vision systems mimicking the functionality of our own sense arise due to the inherent complexities of human vision which has evolved over millions of years to such an extent that the eye and the brain have become partly hard wired through evolution. This fact, complicated further by the inherent parallel processing capabilities of the brain, has meant that the majority of human visual and mental processing tasks are carried out in a seemingly effortless fashion, at performance levels far beyond the capabilities of present hardware/software configurations and, to a greater extent, in a manner that is beyond our current understanding [6]. As a direct consequence of this it is now apparent that this research area is intellectually a difficult task to undertake. Nevertheless, research in the area of developing vision systems for driver assistance is attracting huge attention and is motivated by statistics which show that the human and financial cost resulting from vehicle accidents are enormous [12].

Current vision systems for driver assistance use both active and passive sensors techniques. Although active sensors such as radar [13, 16] and lasers [8] are the most popular because they require minimal computational effort, they are severely limited in terms of scanning speed, and interference across sensors in heavy traffic is a major problem. An alternative is to use passive sensors such as cameras [4, 18] where 360^{o} vision is possible using front and rear cameras. The main advantage of using cameras is that visual data is crucial to the detection of moving objects within lanes and the recognition of traffic signs, pedestrians, vehicles [21] and other objects is greatly enhanced using this type of sensor. However, a reliable vision based driver assistance system using even the most sophisticated system is extremely difficult as vehicles vary in shape, size and colour and outdoor environments can be very complex.

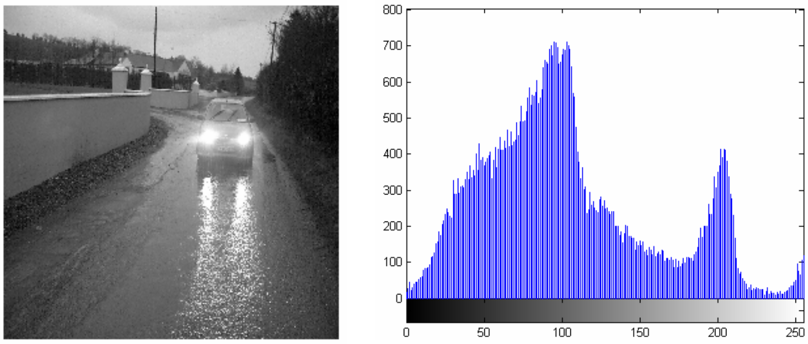
For effective use of car safety vision systems, it is essential that associated vision algorithms must operate in real time. However, image analysis can and usually is a computationally intensive task and current approaches [7, 9] therefore use a two step technique where initially the location of a vehicle in an image is hypothesized (Hypothesis Generation) and a subsequent test is performed to verify the presence of a vehicle (Hypothesis Verification). This technique has been applied to problems such as vehicle detection under poor environmental conditions [15] and vehicle position within lanes [23] to prevent vehicles departing these lanes. Also the technique has found use in perception based systems for collision avoidance. Although this and others approaches [3] have resulted in a large number of prototype vehicles, including Kanwal Jeet Seth's automatic headlight dimming switch, a highly robust and reliable real time vision system for driver support is yet to be developed.

Common to all image processing techniques is the computational effort required which prohibits solutions in real time. To enhance this performance metric research is now beginning to focus on a hybrid of image processing, neural networks and sensory fusion [24]. The motivation for this approach is based on the ability of these networks to learn and therefore classify by example [2, 20]. Neural Networks are an information processing paradigm that is inspired by the way the biological nervous systems process information. Key to this processing paradigm is the large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Neural networks have the ability to derive meaning from complicated or imprecise data, and therefore can be used to extract patterns and detect trends that are too complex or ill defined for traditional analytical methods, empirical rules or other computational techniques [2]. In 1993 Pal and Pal [17] commented that neural networks would soon be extensively used in image processing tasks and this has indeed become the case. Neural networks can readily be found in tasks such as image reconstruction [1], image enhancement [18], feature extraction [10] and image segmentation [14]. A full range of uses of neural networks in image processing tasks are outlined in [11]

This paper presents a preliminary study into the use of image analysis techniques combined with neural networks in order to automatically dip car headlights based on the scene or image ahead. Images are captured with a digital camera at dusk and night time, image histograms are generated and these are fed to the neural network. Preliminary results are promising and indicate that such an automated system could be developed. The neural networks topology, activation function and training algorithm will be described in Section 3 and Section 4 will present the results obtained using a sample of test images. A summary of our findings will be given in Section 5.

2 Image Analysis

The images currently under investigation were captured as still colour images, taken at night-time and at dusk with either a vehicle or no vehicle present, and then converted to 256×256 greyscale images for simplicity as the aim here is to determine if the neural network will readily train. The image test set includes different scenarios such as a vehicle moving away from the camera, a vehicle approaching the camera, and a vehicle situated in different areas of an image as the car may be close or far away.



(a) (b)

Figure 1. (a) Original greyscale image; (b) Corresponding histogram

Instead of feeding the entire 256×256 greyscale image to the neural network, a 1×256 vector representing the image is applied to the neural network for training and testing. As discussed further in Section 3, in one scenario the 1×256 vector is the result of image column summing, in the other scenarios the 1×256 vector represents the image histogram, as illustrated in Figure 1.

The aim of this research is to initially train the neural network to recognize the peaks in the histogram that correspond to the brightness of the car lights in the images and recognize the rest of the histogram as background information.

3 Using Neural Networks

3.1 Back Propagation Neural Network

The back propagation training algorithm [5] applied to a feed forward neural network is the most common neural network based computation mechanism. They are simple and effective, and have found use in a wide assortment of machine learning applications, such as character recognition. Typically these networks are arranged in three layers: the input, hidden, and output layers as illustrated in Figure 2. It should be noted that at any layer L_x in the neural network, $z = \sum_{i=1}^{i=N} wL_{x,i}$ where N corresponds to the number of nodes in that layer and w is a random weight. Before any data has been applied to the network, the weights are initialised randomly and the network is trained until a successful outcome is achieved: here we define a successful outcome as $\theta > 0.7 =$ dipped headlights and $\theta < 0.3 =$ headlights, where θ is the output of the neural network.

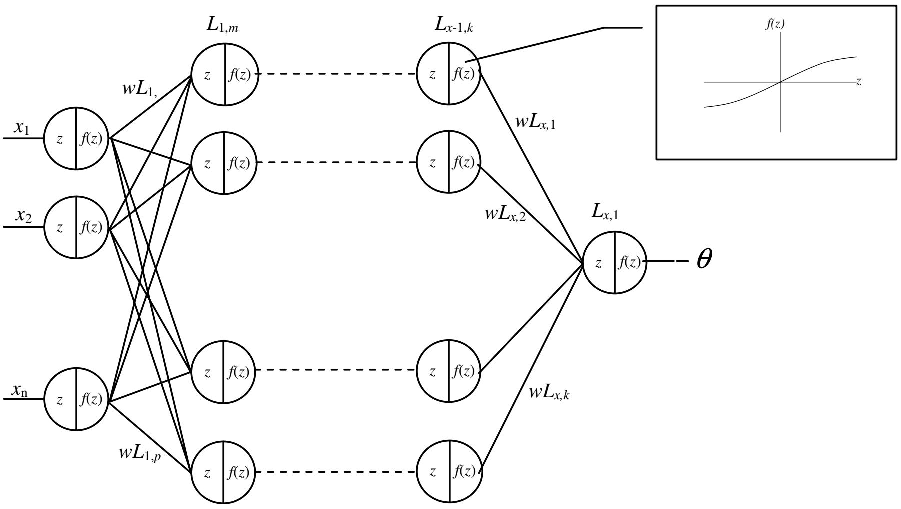


Figure 2. Neural network topology: all neurons are identical with tan-sigmoid activation function

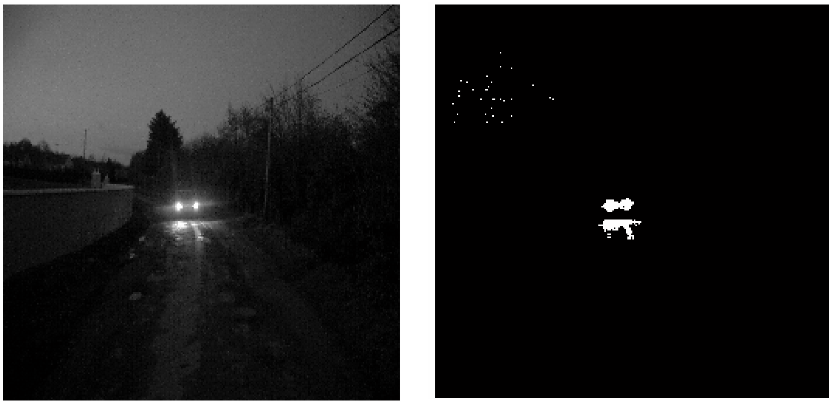
Here, we use a back propagation neural network with supervised learning and a tan-sigmoid activation function, $f(z)$. Due to its amplitude range, the use of a tan-sigmoid function enables easier determination of the output θ using the z-axis as a threshold of the output function, see inset in Figure 2. The back propagation training algorithm compares actual outputs with expected outputs to calculate an error signal. The error is then back propagated from the outputs to the inputs in order to appropriately adjust the weights in each layer of the network.

As the back propagation training algorithm compares actual outputs with expected outputs these expected output must be selected by the user. For the purpose of our application of intelligent car headlight dipping, for images input into the system that contained a car and the decision to automatically dip the car headlight is expected, the expected output is set at 0.7 and likewise, for an image input into the system that contained no car and hence the decision to maintain full headlight is expected, the expected output is set at 0.3. These expected outputs were arbitrarily chosen to enable an error band around each required output within the range 0 to 1. During the training phase there were two thresholds set on the neural network for the computed values, 0.4 and 0.6. Hence, if the network after training computed an output > 0.6 the system would dip and < 0.4 the system would not dip the headlights. These thresholds where set in order to improve the success rate of the network.

The neural network was trained for two classifications (dipped headlights or full headlights) using a set of 46 training images containing cars approaching the camera, moving away the camera and without cars. It should be noted here that there was only one car in any image for the purpose of this preliminary investigation.

3.2 Night-time Images

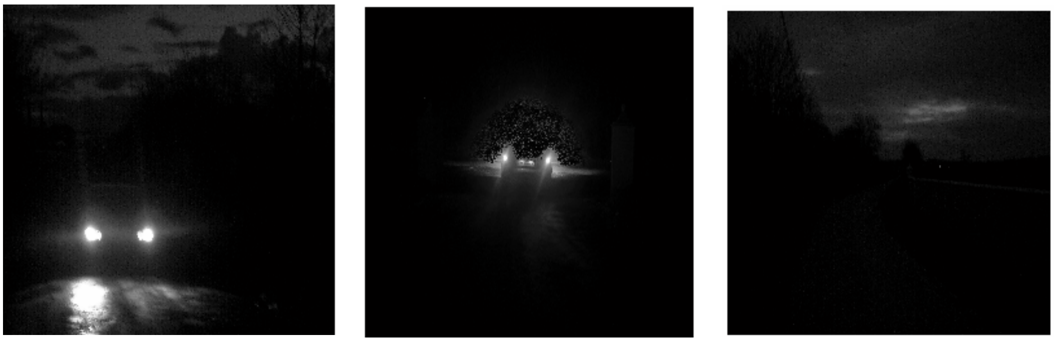
The initial investigation tested our neural network approach using only night-time images and therefore the background was dark. Greyscale images were converted to binary images using an empirically chosen threshold value of 240, as illustrated in Figure 3 and hence only the headlights were visible. Each column of the image was summed to obtain a 1×256 vector that contained numerical peaks corresponding to the presence of car headlights anywhere within the image. Subsequently a neural network was trained using these vectors and a 70% success rate was obtained.



(a) (b)

Figure 3 (a) Greyscale image; (b) Corresponding binary image

On obtaining an encouraging success rate using the binary image vector input, we altered the process to use the original greyscale images, illustrated in Figure 4, rather than the binary image.



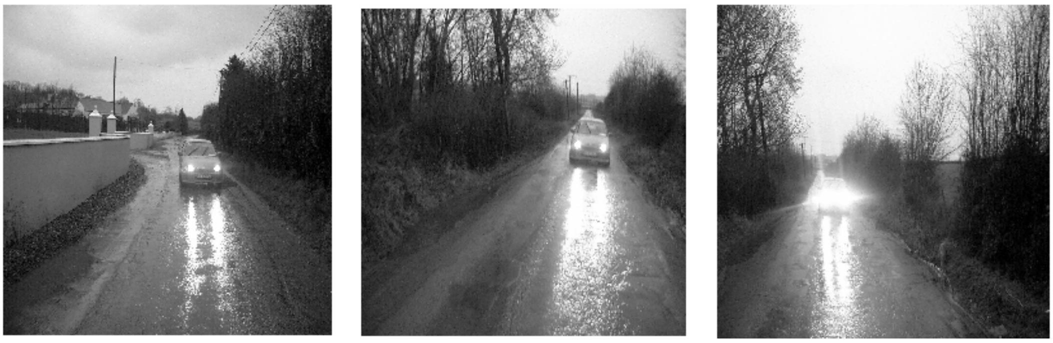
(a) Oncoming car headlights (b) Car tail-lights (c) Image containing no car

Figure 4. Sample of the greyscale night-time images

Instead of using a basic column summing approach, we applied the histogram data, corresponding to the greyscale image, to the neural network. Hence, the neural network was still receiving 256 inputs with the inputs corresponding to the true image values. We trained the network several times using a variety of topologies and achieved an 80% success rate, hence 80% of the time the neural network classified a correct response when determining whether the car headlights should be dipped.

3.3 Night-time and Dusk Images

Due to the success of this application using only night-time images, we extended the image data set to include images captured a dusk, where cars would have their headlights on but the background would not be completely dark and typical countryside scenery was visible, as can be seen in Figure 5. Again 256 inputs were applied to the neural network, corresponding to the greyscale image histogram. Throughout the experimental training phase a number of different network topologies were implemented and the topology yielding the highest percentage success rate, when trying to make a decision as to whether the car headlights should be dipped, was selected. The neural network that provided the highest rate of success of approximately 85% is a four layered topology as illustrated in Figure 2.



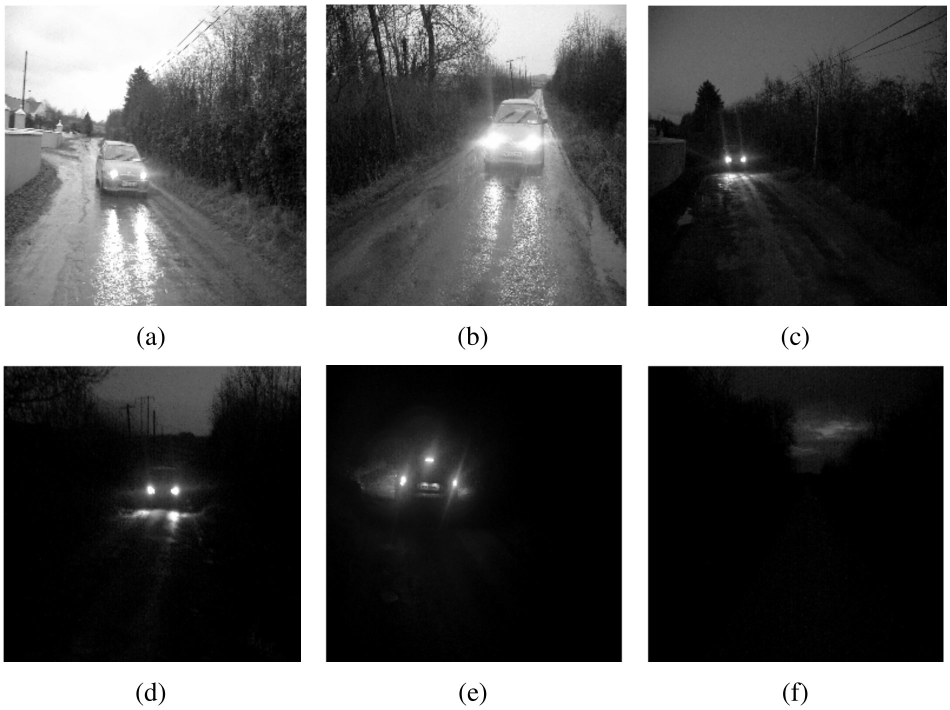
(a) (b) (c)

Figure 5. Sample of the greyscale image captured at dusk

4 Preliminary Results

In order to test the neural network presented here, we used a set of 20 test images, a sample of which is presented in Figure 6. This test set included a combination of dusk images, night-time images and images containing no cars. Each of the 20 images in the test image set was applied to the trained neural network described in Section 3.3.

As described in Section 3.1, there were two threshold values set on the neural network during the training phase. However, this leaves an undefined region between 0.4 and 0.6 where no decision is made. This clearly is not appropriate as the aim of this project is to make an intelligent decision as to whether to dip car headlights. Therefore, for the testing phase, the threshold was altered to 0.5, and hence if the neural network output is >0.5, the system will dip the car headlights and if the output is <0.5, the system will not dip the car headlights. Table 1 shows the actual outputs computed for each of the 20 test images with the corresponding expected output, decision made by the intelligent system and whether the decision is correct. As illustrated in Table 1, the system made all of the decisions correctly except for 2 images. These images contained no cars but were captured at dusk and hence the background information in the image has driven the neural network output to be above threshold rather than below it. However, overall 90% of the decisions made by the system are correct and these results are encouraging at this early stage of the work.



(a) (b) (c)

(d) (e) (f)

Figure 6. Sample of test images capture at dusk and night-time

Image	Actual Output	Expected Output	Decision	Correct Decision?
Test Image 1	0.6397	0.7	Dip	Yes
Test Image 2	0.2912	0.3	Full headlights	Yes
Test Image 3	0.5909	0.7	Dip	Yes
Test Image 4	0.6131	0.7	Dip	Yes
Test Image 5	0.5876	0.3	Dip	No
Test Image 6	0.6131	0.7	Dip	Yes
Test Image 7	0.2863	0.3	Full headlights	Yes
Test Image 8	0.5331	0.7	Dip	Yes
Test Image 9	0.3087	0.3	Full headlights	Yes
Test Image 10	0.6030	0.7	Dip	Yes
Test Image 11	0.6661	0.7	Dip	Yes
Test Image 12	0.7025	0.7	Dip	Yes
Test Image 13	0.5079	0.3	Dip	No
Test Image 14	0.2945	0.3	Full headlights	Yes
Test Image 15	0.7189	0.7	Dip	Yes
Test Image 16	0.2893	0.3	Full headlights	Yes
Test Image 17	0.2950	0.3	Full headlights	Yes
Test Image 18	0.6661	0.7	Dip	Yes
Test Image 19	0.2840	0.3	Full headlights	Yes
Test Image 20	0.2745	0.3	Full headlights	Yes

Table 1.

In Table 1, the two images that have been incorrectly classified by the neural network, test image 5 and test image 13, were both captured at dusk, with no oncoming vehicle in the scene.

5 Conclusion

We have demonstrated here that a neural network can be trained with approximately 85% accuracy to interrupt a greyscale image histogram and make an intelligent decision whether the system should dip the car highlights. This preliminary study uses only simple images containing one vehicle but the preliminary results are encouraging and indicate that such an intelligent system could be fully developed. Further work in this area will entail more detailed image analysis, for example the current system may interrupt an image containing two street lights as an oncoming car also we will consider images that contain more than one car, further image processing such as feature extraction would distinguish between a vehicle and street lights. We will also consider using both intensity and range data in order to determine if the light source is near or in the distance. Ultimately the system will be developed to deal with a variety of environmental conditions, backgrounds and multiple vehicles and will use image sequences captured in real-time rather than simple still images.

6. References

[1] Adler A., and Guardo, R., A neural network image reconstruction technique for electrical impedance tomography. *IEEE Trans. Med. Imaging* Vol. 13 No. 4 pp. 594–600, 1994.

[2] Belatreche A., Maguire L.P., McGinnity T.M., “An evolutionary strategy for supervised training of biological plausible neural networks”, *JCIS* pp.1524-1527, 2003.

[3] Bertozzi M., Broggi A., Cellario M., Fascioli A., Lombardi P., and Porta M., “Artificial vision in road vehicles”, *IEEE Special issue on Technology and Tools: Visual Perception*, vol. 90, no. 7, pp. 1258-12171, 2002.

[4] Broggi A., Bertozzi M., Fascioli A., and Sechi M., “Shape-based pedestrian detection”, *Proc. of IEEE Intelligent Vehicle Symposium*, 2004.

[5] Daelemans W, Van den Bosch A, Drossaers M and Nijholt A, “Generalization performance of backpropagation learning on a syllabification task”, *Connectionism and Natural Language Processing, Proc 3rd Workshop on Language Technology*, 27-38, 1992.

[6] Davies E.R., *Machine Vision, Theory, Algorithims and Practicalities*, Academic Press 1997

[7] Egmont-Petersen M., de Ridder D., and Handels H., “Image processing with neural networks - a review” *Pattern Recognition,* Vol 35, No. 10, Pages 2279-2301 October 2002.

[8] Franke U., and Heinrich S., “Fast obstacle detection for urban traffic situations”, *IEEE Trans. on Intelligent Transportation Systems*, Vol. 3, pp. 173-181, 2002.

[9] Fuerstenberg K., and Dietmayer K., “Object tracking and classification for multiple active safety and comfort applications using a multiplayer Laserscanner”, *IEEE Intelligent Vehicle Symposium*, pp.802-807, 2004.

[10] Gavrila D., and Philomin V., “Real-time object detection for “SMART” vehicles”, *Proc. of IEEE Int. Conference on computer vision*, pp.87-93, 1999.

[11] Glass J.O., and Reddick W.E., Hybrid artificial neural network segmentation and classification of dynamic contrast-enhanced MR imaging (DEMRI) of osteosarcoma. *Magn. Resonance Imaging* Vol. 16 No. 9, pp. 1075–1083, 1998.

[12] Jones W., “Building safer cars”, *IEEE Spectrum*, vol. 39, no. 1, pp. 82-85, 2002.

[13] Knoll P., Schaefer B., Guettler H., Bunse M., and Kallenbach R., “Predictive Safety System – Steps towards collision mitigation”, *SAE Technical Paper*, 2004-01-1111, 2004.

[14] Kotropoulos, C., Magnisalis, X., Pitas I., *et al.*, Nonlinear ultrasonic image processing based on signal-adaptive filters and self-organizing neural networks. *IEEE Trans. Image Processing* Vol. 3, No.1 pp. 65–77, 1994.

[15] Nakayama O., Shiohara M., Sasaki S., Takashima T., and Ueno D., “Robust vehicle detection under poor environmental conditions for rear and side surveillance”, *IEICE transactions on information and systems* Vol. E87D, No. 1, Jan. 2004.

[16] Noyce D., and Dharmaraju R., “An evaluation of technologies for automated detection and classification of pedestrians and bicyclists”, *Massachusetts Highway Department report*, 2002.

[17] Pal N.R., and Pal S.K., “A review on image segmentation techniques.” *Pattern Recognition* Vol. 26 No. 9 pp. 1277–1294, 1993.

[18] Pugmire R.H., Hodgson R.M. and Chaplin R.I., The properties and training of a neural network based universal window filter developed for image processing tasks. In: S. Amari and N. Kasabov, Editors, *Brain-like computing and intelligent information systems*, Springer-Verlag, Singapore pp. 49–77, 1998.

[19] Shashua A., Gdalyahu Y., and Hayun G., “Pedestrian detection for driving assistance systems: Single-frame classification and system level performance”, *Proc. of IEEE Intelligent Vehicle Symposium*, 2004.

[20] Strain T, McDaid L.J., Maguire L.P., McGinnity T.M., “A Novel Mixed Supervised-Unsupervised Training Approach For A Spiking Neural Network Classifier”, *IEEE SMC UK-RI Chapter Conference 2004 on Intelligent Cybernetic Systems*, pp.202-206,September 7-8, 2004.

[21] Taktak, R., DuFaut, M., Wolf, D., Husson, R., “Analysis and Inspection of Road Traffic using Image Processing” *Mathematics and Computers in Simulation*, Vol. 41, pp. 273-283, 1996.

[22] Thorpe C., Carlson T., Duggins D., Gowdy J., MacLachlan R., Mertz C., Suppe A., and Wan C., “Safe robot driving in cluttered environments”, *11th International Symposium of Robotics Research*, 2003.

[23] Xu L., Weigong Z., and Xiaodong B., “Research on detection of lane based on machine vision”, *Journal of Southeast University (English Edition),* vol. 20, No. 2, June 2004.

[24] Zhao L., “Stereo-and Neural network-based pedestrian detection”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, No. 3, pp. 148-154, 2000.

Object Recognition and Real-Time Tracking in Microscope Imaging

J. Wedekind, M. Boissenin, B.P. Amavasai, F. Caparrelli, J. Travis
MMVL, Materials and Engineering Research Institute
Sheffield Hallam University,
Pond Street,
Sheffield S1 1WB
{J.Wedekind,B.P.Amavasai,F.Caparrelli,J.R.Travis}@shu.ac.uk,
Manuel.Boissenin@student.shu.ac.uk

30.6.2006

Abstract

As the fields of micro- and nano-technology mature, there is going to be an increased need for industrial tools that *enable the assembly and manipulation of micro-parts*. The feedback mechanism in a future micro-factory will require computer vision.

Within the **EU IST MiCRoN** project, a computer vision software based on *Geometric Hashing* and the *Bounded Hough Transform* to achieve **recognition of multiple micro-objects** was implemented and successfully demonstrated. In this environment, the micro-objects will be of variable distance to the camera. Novel automated procedures in biology and micro-technology are thus conceivable.

This paper presents an approach to estimate the pose of multiple micro-objects with up to four degrees-of-freedom by using focus stacks as models. The paper also presents a formal definition for Geometric Hashing and the Bounded Hough Transform.

Keywords: object recognition, tracking, Geometric Hashing, Bounded Hough Transform, microscope

1 Introduction

Under the auspices of the European *MiCRoN*[MiCRoN consortium, 2006] project a system of multiple micro-robots for transporting and assembling μm-sized objects was developed. The micro-robots are about 1 cm^3 in size and are fitted with an interchangeable toolset that allows them to perform manipulation and assembly. The project has developed various subsystems for powering, locomotion, positioning, gripping, injecting, and actuating. The task of the *Microsystems & Machine Vision Lab* was to develop a real-time vision system, which provides feedback information to the control system and hence forms part of the control loop.

Although there are various methods for object recognition in the area of computer vision, most techniques which have been developed for microscope imaging so far do not address the issue of real-time. Most current work in the area of micro-object recognition employs 2-D recognition methods (see *e.g.* [Begelman et al., 2004]) sometimes in combination with an auto-focussing system, which ensures that the object to be recognised always stays in focus.

This paper presents an algorithm for object recognition and tracking in a microscope environment with the following objectives: Objects can be recognised with *up to 4 degrees-of-freedom*, *refocussing is not required*, *tracking is performed in real-time*.

The following sections of this paper will provide a formalism to *Geometric Hashing* and the *Bounded Hough Transform*, how they can be applied to a pre-stored focus stack, and how this focus stack can be used to recognise and track objects, the results, and finally we draw some conclusions.

2 Formalism

In applying Geometric Hashing and the Bounded Hough Transform to micro-objects, recognition and tracking with three degrees-of-freedom is first developed. Later this is expanded to four degrees-of-freedom.

2.1 Geometric Hashing

[Forsyth and Ponce, 2003] provides a complete description of the Geometric Hashing algorithm first introduced in [Lamdan and Wolfson, 1988]. Geometric Hashing is an algorithm that uses geometric invariants to vote for feature correspondences.

2.1.1 Preprocessing-Stage

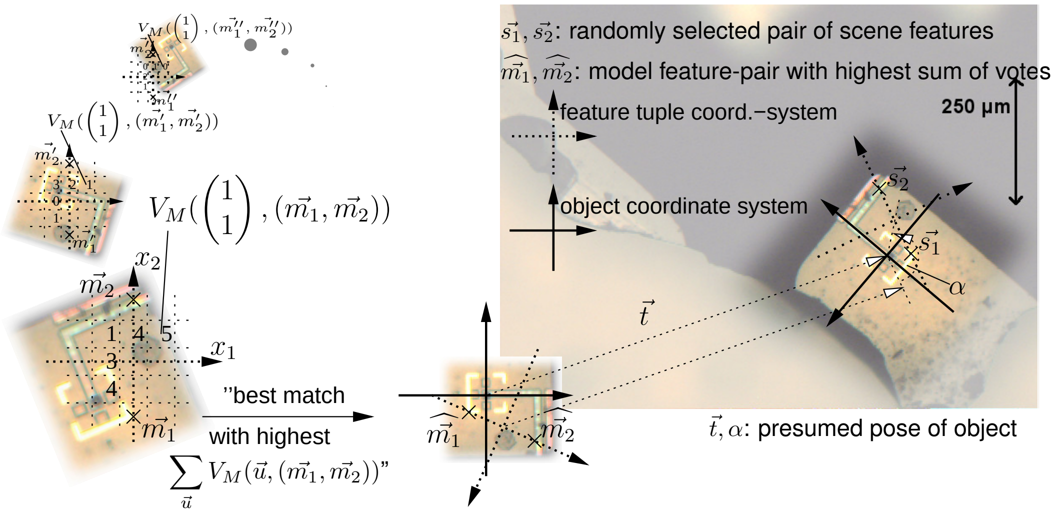
Let the triplet $\vec{p} := (t_1, t_2, \theta)^\top \in P$ be the pose of an object and $P := \mathbb{R}^3$ the pose-space. $\dim(P) = 3$ is the number of degrees-of-freedom. The object can rotate around one axis and translate in two directions. It can be found using a set $M \subset \mathbb{R}^{d+1}$ of homogeneous coordinates denoting 2-D ($d = 2$) feature points (here: edge-points acquired with a Sobel edge-detector) as a model

$$M := \left\{\vec{m_i} = (m_{i,1}, m_{i,2}, 1)^\top \middle| i \in \{1, 2, \ldots\}\right\} \tag{1}$$

First the set of geometric invariants $L(M)$ has to be identified. A geometric invariant of the model is a feature or a (minimal) sequence of features such that the pose of the object can be deduced if the location of the corresponding feature/the sequence of features in the scene is known.

Consider the example in fig. 1. In this case the correspondence between a two feature points $\vec{s_1}, \vec{s_2} \in S$ in the scene $S \subset \mathbb{R}^{d+1}$ and two feature points $\vec{m_1}, \vec{m_2} \in M$ of the model M would reveal the pose $\vec{p}$ of the object. Therefore feature tuples can serve as geometric invariants.

In practice only a small number of feature tuples can be considered. A subset of $M \times M$ is selected by applying a minimum- and a maximum-constraint on the distance between the two feature points of a tuple $(\vec{l_1}, \vec{l_2})$. Hence in this case, L is defined as $L(A) = \left\{(\vec{l_1}, \vec{l_2}) \in A \times A \middle| g_u \leq ||\vec{l_1} - \vec{l_2}|| \leq g_o\right\}$ for any set of features $A \subseteq \mathbb{R}^{d+1}$.



$V_M(\begin{pmatrix}1\\1\end{pmatrix}, (\vec{m_1''}, \vec{m_2''}))$
$\vec{m_2''}$
0 1 0
2
1
$\vec{m_1''}$
$V_M(\begin{pmatrix}1\\1\end{pmatrix}, (\vec{m_1'}, \vec{m_2'}))$
$\vec{m_2'}$
3 2 1
0
1
$\vec{m_1'}$
$V_M(\begin{pmatrix}1\\1\end{pmatrix}, (\vec{m_1}, \vec{m_2}))$
x_2
$\vec{m_2}$
1 4 5
3
4
x_1
$\vec{m_1}$
"best match with highest $\sum_{\vec{u}} V_M(\vec{u}, (\vec{m_1}, \vec{m_2}))$"
$\widehat{\vec{m_1}}$
$\widehat{\vec{m_2}}$
$\vec{s_1}, \vec{s_2}$: randomly selected pair of scene features
$\widehat{\vec{m_1}}, \widehat{\vec{m_2}}$: model feature-pair with highest sum of votes
feature tuple coord.−system
250 µm
object coordinate system
$\vec{s_2}$
$\vec{s_1}$
α
$\vec{t}$
$\vec{t}, \alpha$: presumed pose of object

Figure 1: Geometric Hashing to locate a syringe-chip (courtesy of IBMT, St. Ingbert) in a microscope image (reflected light) allowing three degrees-of-freedom

Geometric Hashing provides a technique to establish the correspondence between the geometric invariants $(\vec{m_1}, \vec{m_2}) \in L(M)$ and $(\vec{s_1}, \vec{s_2}) \in L(S)$ where $S, M \subset \mathbb{R}^{d+1}$.

To apply Geometric Hashing a function

$$t : \left\{ \begin{array}{ccc} L(\mathbb{R}^{d+1}) & \rightarrow & \mathbb{R}^{(d+1)\times(d+1)} \\ (\vec{l_n}) & \mapsto & T\big((\vec{l_n})\big) \end{array} \right. \tag{2}$$

is chosen which assigns an affine transformation matrix $T\big((\vec{l_n})\big)$ to a geometric invariant $(\vec{l_n}) := (\vec{l_1}, \vec{l_2}, \ldots \vec{l_n})$ in $L(M) \subset L(\mathbb{R}^{d+1})$ or $L(S) \subset L(\mathbb{R}^{d+1})$. The affine transformation inverses the transformation which is designated by the sequence of features $(\vec{l_1}, \vec{l_2}, \ldots \vec{l_n})$. *E.g.* in this case t must fulfil

$$\forall \vec{p} \in P, (\vec{l_1}, \vec{l_2}) \in \mathbb{R}^3 \times \mathbb{R}^3 : t\big((\mathcal{R}(\vec{p}) \cdot \vec{l_1}, \mathcal{R}(\vec{p}) \cdot \vec{l_2})\big) = \mathcal{R}(\vec{p}) \cdot t\Big((\vec{l_1}, \vec{l_2})\Big)$$

$$\text{where } \mathcal{R}(\begin{pmatrix} t_1 \\ t_2 \\ \theta \end{pmatrix}) := \begin{pmatrix} \cos(\theta) & -\sin(\theta) & t_1 \\ \sin(\theta) & \cos(\theta) & t_2 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

Furthermore t must conserve the pose-information, *i.e.* $\dim\big(\operatorname{aff}(t(L(\mathbb{R}^{d+1}))\big) = \dim(P)$ where $\operatorname{aff}(X)$ is the affine hull of X.

Note that $\vec{l}$ and $\vec{l'}$ are homogeneous coordinates of points, and for simplification we can use $l_3 = l'_3 = 1$. Using this, a possible choice for t is given by

$$t\big((\vec{l},\vec{l'})\big) = \frac{1}{\sqrt{(l'_1 - l_1)^2 + (l'_2 - l_2)^2}} \begin{pmatrix} l'_2 - l_2 & l_1 - l'_1 & 0 \\ l'_1 - l_1 & l'_2 - l_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -\frac{1}{2}\,(l_1 + l'_1) \\ 0 & 1 & -\frac{1}{2}\,(l_2 + l'_2) \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

The choosen transformation $t\big((\vec{m_1}, \vec{m_2})\big)$ maps the two feature points $\vec{l}$ and $\vec{l'}$ on the x_2-axis as shown in figure 1.

Let h be a quantising function for mapping real homogeneous coordinates of feature positions to whole-numbered indices of voting table bins of discrete size Δs:

$$h : \left\{ \begin{array}{ccl} \mathbb{R}^{d+1} & \to & \mathbb{Z}^d \\ \vec{x} & \mapsto & \vec{u} \text{ where } u_i = \left\lfloor \dfrac{x_i}{x_{d+1}\,\Delta s} + \dfrac{1}{2} \right\rfloor, i \in \{1, 2, \ldots, d\} \end{array} \right. \tag{5}$$

[Blayvas et al., 2003] offers more information on how to choose the bin size Δs properly. Note that $x_{d+1} = 1$ since h is going to be applied to homogeneous coordinates of points only.

First a voting table $V_M : \mathbb{Z}^d \times L(M) \to \mathbb{N}_0$ for the model M is computed (see alg. 1)[1]. In practice V_M only needs to be defined on a finite subset of $\mathbb{Z}^d$, while $L(M)$ is finite if M is.

Algorithm 1: Creating a voting table offline, before doing recognition with the Geometric Hashing algorithm[Forsyth and Ponce, 2003]

```
Input: Model $M \subset \mathbb{R}^{d+1}$
Output: Voting table $V_M : \mathbb{Z}^d \times L(M) \to \mathbb{N}_0$
/* Set all elements of $V_M$ to zero                                  */
$V_M(\cdot,\cdot) \mapsto 0$;
foreach geometric invariant $(\vec{m_n}) = (\vec{m_1}, \vec{m_2}, \ldots, \vec{m_n}) \in L(M)$ do
    foreach feature point $\vec{m'} \in M$ do
        /* Compute index of voting table bin                           */
        $\vec{u} := h(t\big((\vec{m_n})\big) \cdot \vec{m'})$;
        /* Add one vote for the sequence of features $(\vec{m_n})$         */
        $V_M\big(\vec{u}, (\vec{m_n})\big) \mapsto V_M\big(\vec{u}, (\vec{m_n})\big) + 1$;
    end
end
```

$(t\big((\vec{m_1}, \vec{m_2})\big) \cdot \vec{m'})$ is the position of $\vec{m'}$ relatively to the geometric invariant $(\vec{m_1}, \vec{m_2}) \in L(M)$. This relative position is quantised by h and assigned to $\vec{u}$. $V_M\big(\vec{u}, (\vec{m_1}, \vec{m_2})\big)$ is the number of features residing in the bin of the voting table with the quantised position $\vec{u}$ relative to the geometric invariant $\vec{m} \in L(M)$.

2.1.2 Recognition-Stage

A random pair of features $(\vec{s_1}, \vec{s_2})$ is picked from the Sobel-edges of the scene-image. All other features of the scene are mapped using the transform $t\big((\vec{s_1}, \vec{s_2})\big)$ (see alg. 2). The accumulator a is used to decide where both features are located on the object and whether they are residing on the same object at all.

On success, sufficient information to calculate the pose of the object is available. The pose $\vec{p} = (t_1, t_2, \theta)^\top$ of the object can be calculated using:

$$\mathcal{R}(\vec{p}) = t\big((\vec{s_1}, \vec{s_2})\big)^{-1}\, t\big((\widehat{\vec{m_1}}, \widehat{\vec{m_2}})\big) \tag{6}$$

2.2 Bounded Hough Transform

As Geometric Hashing alone is too slow to achieve real-time vision, a tracking algorithm based on the Bounded Hough Transform[Greenspan et al., 2004] was employed. Thus after a micro-object has been located, it can be tracked in consecutive frames with much lower computational cost.

[1] $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$

Algorithm 2: The Geometric Hashing algorithm for doing object recognition[Forsyth and Ponce, 2003]

```
Input: Set of scene features $S \subset \mathbb{R}^{d+1}$
Output: Pose $\vec{p}$ of object or failure
Initialise accumulator $a : L(M) \to \mathbb{N}_0$;
Randomly select a geometric invariant $(\vec{s_n}) = (\vec{s_1}, \vec{s_2}, \dots, \vec{s_n})$ from $L(S)$;
foreach feature points $\vec{s'} \in S$ do
    /* Compute index $\vec{u}$ of voting table bin                        */
    $\vec{u} := h\big(t\big((\vec{s_n})\big) \cdot \vec{s'}\big)$;
    foreach $(\vec{m_n}) \in L(M)$ do
        /* Increase the accumulator using the voting table               */
        $a\big((\vec{m_n})\big) \mapsto a\big((\vec{m_n})\big) + V_M(\vec{u}, (\vec{m_n}))$;
    end
end
/* Find accumulator bin with maximum value                                 */
$(\widehat{\vec{m_n}}) := \underset{(\vec{m_n}) \in L(M)}{\operatorname{argmax}} \Big(a\big((\vec{m_n})\big)\Big)$;
if $a\big((\widehat{\vec{m_n}})\big)$ is bigger than a certain threshold then
    /* $t\big((\vec{s_n})\big)^{-1} t\big((\widehat{\vec{m_n}})\big)$ contains suggested pose of object.
       Back-project and verify before accepting the
       hypothesis[Forsyth and Ponce, 2003]                                  */
else
    /* Retry by restarting algorithm or report failure                     */
end
```

2.2.1 Preprocessing-Stage

The basic idea of the Bounded Hough Transform is to transform the positions of all features $\vec{s} \in S$ to the coordinate-system defined by the object's previous pose $\vec{p}$. If the speed of the object is restricted by $r_1, r_2, \dots$ (*i.e.* $|p'_i - p_i| \leq r_i$, $\vec{r} \in \mathbb{R}^{\dim(P)}$) and the change of pose is quantised by $q_1, q_2, \dots$ (*i.e.* $\exists k \in \mathbb{Z} : p'_i - p_i = k\,q_i$, $\vec{q} \in \mathbb{R}^{\dim(P)}$), the problem of determining the new pose $\vec{p'} \in P$ of the object is reduced to selecting an element $\widehat{\vec{d}} := \vec{p'} - \vec{p}$ from the finite set $D \subset P$ of pose-changes

$$D := \big\{\vec{d} \in P \big| \forall i \in \{1, \dots, \dim(P)\} : |d_i| \leq r_i \wedge \exists k \in \mathbb{Z} : d_i = k\,q_i\big\} \tag{7}$$

Fig. 2 illustrates how the Bounded Hough Transform works in the case of two degrees-of-freedom ($\vec{p} = (t_1, t_2)^\top$). The hough-space of pose-changes D is limited and therefore only the features residing within a small local area of M can correspond to the scene-feature $\vec{s} \in S$. Each possible correspondence votes for one pose-change (in the general case it may vote for several different pose-changes). As one can see in fig. 2, accumulating the votes of two scene-features already can reveal the pose-change of the object.

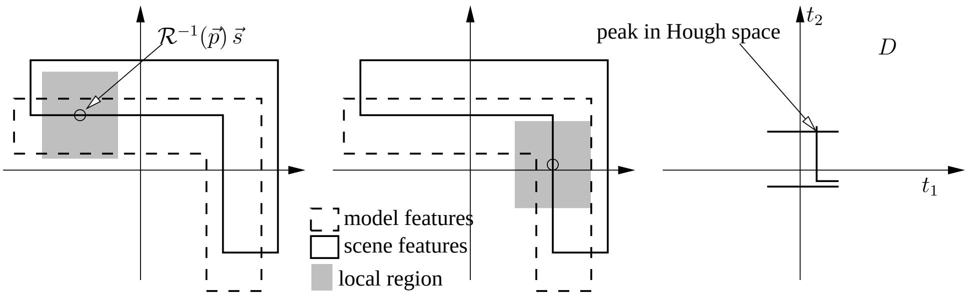


Figure 2: Bounded Hough Transform with 2 degrees-of-freedom

First a voting table H_M is computed as shown in alg. 3. In practice H_M only needs to be defined on a finite subset of $\mathbb{Z}^d$ while D is finite.

The functions $C : \mathbb{R}^{d+1} \to P$ and $W : \mathbb{R}^{d+1} \to \mathbb{R}_0^+$ are required to cover H_M properly. In the case of two degrees-of-freedom one can simply use $C(\vec{m}) = \{\vec{0}\}$ and $W(\vec{m}) = 1$ if the quantisation of the

Algorithm 3: Initialising voting table offline, before doing tracking using the Bounded Hough Transform algorithm

```
Input: Model $M \subset \mathbb{R}^{d+1}$, ranges $\vec{r}$, quantisation $\vec{q}$
Output: Voting table $H_M : \mathbb{Z}^d \times D \rightarrow \mathbb{N}_0$
/* Set all elements of $H_M$ to zero                              */
$H_M(\cdot,\cdot) \mapsto 0$;
foreach pose difference vector $\vec{d} \in D$ do
    foreach feature point $\vec{m} \in M$ do
        foreach pose difference vector $\vec{c} \in C(\vec{m})$ do
            /* Compute index of voting table bin                    */
            $\vec{u} := h(\mathcal{R}(\vec{d}+\vec{c}) \cdot \vec{m})$;
            /* Update votes for pose-change $\vec{d}$                 */
            $H_M(\vec{u},\vec{d}) \mapsto H_M(\vec{u},\vec{d}) + W(\vec{m})$;
        end
    end
end
```

translation in D does not exceed the bin-size (*i.e.* $q_i \leq \Delta s$).

In the case of three degrees-of-freedom ($\vec{p} = (t_1, t_2, \theta)^\top$) the *density* of the votes depends on the features distance from the origin (radius). If the radius is large, several bins of H_M may have to be increased. If the radius is very small, the weight of the vote should be lower than 1 as the feature cannot define the amount of rotation unambiguously. Therefore in the general case C and W are defined as follows

$$C(\vec{m}) := \big\{\vec{c} \in P \big| \forall i \in \{1,\ldots,\dim(P)\} : |c_i| \leq \frac{q_i}{2}\Big|\Big|\frac{\delta\mathcal{R}(\vec{x})}{\delta x_i}\vec{m}\Big|\Big| \wedge \exists k \in \mathbb{Z} : c_i = k\,\Delta s\big\} \quad (8)$$

$$W(\vec{m}) = \prod_{i=1}^{\dim(P)} \min\big(1, \Big|\Big|\frac{\delta\mathcal{R}(\vec{x})}{\delta x_i}\vec{m}\Big|\Big|\big) \quad (9)$$

In the case of three degrees-of-freedom C and W are defined using

$$\left(\Big|\Big|\frac{\delta\mathcal{R}\big((t_1,t_2,\theta)^\top\big)}{\delta t_1}\vec{m}\Big|\Big|, \Big|\Big|\frac{\delta\mathcal{R}\big((t_1,t_2,\theta)^\top\big)}{\delta t_2}\vec{m}\Big|\Big|, \Big|\Big|\frac{\delta\mathcal{R}\big((t_1,t_2,\theta)^\top\big)}{\delta\theta}\vec{m}\Big|\Big|\right)^\top = \begin{pmatrix} m_3 \\ m_3 \\ \sqrt{m_1^2+m_2^2} \end{pmatrix} \quad (10)$$

Note that $\vec{m}$ is a homogeneous coordinate of a point and therefore $m_3 = 1$.

2.2.2 Tracking-Stage

The tracking-stage of the Bounded Hough Transform algorithm is fairly straightforward. All features of the scene are mapped using the transform $\mathcal{R}(\vec{p})^{-1}$ defined by the previous pose $\vec{p}$ of the object (see alg. 4). The accumulator b is used to decide where the object has moved or whether it was lost.

2.3 Four Degrees-of-Freedom

In practice the depth information contained in microscopy images can be used to achieve object recognition and tracking with four degrees-of-freedom. Recognition and tracking with four degrees-of-freedom is achieved by using two sets of competing voting tables $\{V_{M_1}, V_{M_2}, \ldots\}$ and $\{H_{M_1}, H_{M_2}, \ldots\}$, which have been generated from a focus stack of the object. Figure 3 shows an artificial focus stack of the text-object “Mimas”, which is being compared against an artificial image, which contains two text-objects.

The voting tables for recognition can be stored in a single voting table V_M^* if an additional index for the depth is introduced. Furthermore during tracking only a subset of $\{H_{M_1}, H_{M_2}, \ldots\}$ needs to be considered as the depth of the object can only change by a limited amount. In practice an additional index for change of depth is introduced, and a set of voting tables $\{H_{M_{1,2}}, H_{M_{1,2,3}}, H_{M_{2,3,4}}, \ldots\}$ is created from images of neighbouring focus-layers. During tracking only a single voting table in this set needs to be considered.

Algorithm 4: The Bounded Hough Transform algorithm for tracking objects

```
Input: Set of scene features $S \subset \mathbb{R}^{d+1}$, previous $\vec{p}$ of object
Output: Updated pose $\vec{p'}$ of object or failure
Initialise accumulator $b : D \to \mathbb{N}_0$;
foreach feature point $\vec{s} \in S$ do
    /* Compute index $\vec{u}$ of voting table bin */
    $\vec{u} := h\big(\mathcal{R}(\vec{p})^{-1}\,\vec{s}\big)$;
    foreach vector of pose-change $\vec{d} \in D$ do
        /* Increase the accumulator using the voting table */
        $b(\vec{d}) \mapsto b(\vec{d}) + H_M(\vec{u}, \vec{d})$;
    end
end
/* Find accumulator bin with maximum value */
$\widehat{\vec{d}} = \underset{\vec{d} \in D}{\operatorname{argmax}} \big(b(\vec{d})\big)$;
if $b(\widehat{\vec{d}})$ is bigger than a certain threshold then
    /* $\vec{p'} = \vec{p} + \widehat{\vec{d}}$ is the suggested pose of the object */
else
    /* Report failure */
end
```

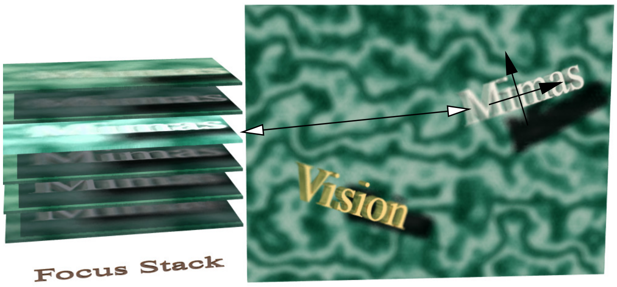


Figure 3: Geometric Hashing with four degrees-of-freedom

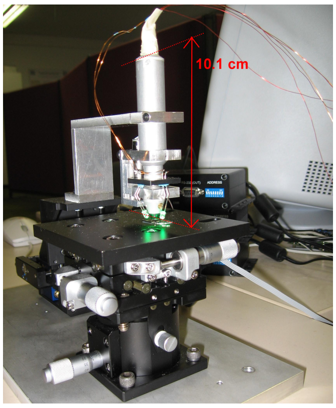


Figure 4: Test environment

3 Results

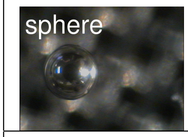
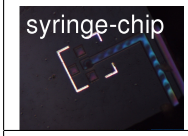
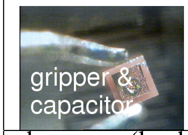
In order to observe the tools and micro-objects, a custom built micro-camera was developed and mounted on a motorised stage (see fig. 4). The micro-camera has an integrated lens system and a built-in focus drive that allows the lens position to be adjusted. The field of view is similar to that obtained from a microscope with low magnification (about 0.8 mm×0.5 mm field of view).

The test environment (see fig. 4) allows the user to displace a micro-object using the manual translation stage. The task of the vision-system is to keep the micro-object in the centre of the image and in focus using the motorised stage.

Figure 5 shows a list of results acquired on a 64-bit AMD processor with 2.2 GHz. The initialisation time for the voting-tables has not been included as they are computed offline. First recognition using geometric hashing was run on 1000 frames. The *recognition rate* indicates the percentage of frames, when the object was recognised successfully. In a second test tracking was applied to 1000 frames. The last column in the table shows the corresponding improved frame-rate. In both tests the graphical visualisation was disabled (which saves 0.013 seconds per frame). To require less memory for V_M and H_M, recognition and tracking are performed on down-sampled images. The disadvantage is that the resulting pose-estimate for the micro objects is coarser.

The recognition rate can be increased at the expense of allowing more processing time. However in reality a low recognition rate is much more tolerable than a low frame-rate. Furthermore recognition is only required for initial pose-estimates, when new objects are entering the scene. As the tracking-

Figure 5: Results for object recognition with Geometric Hashing in a variety of environments

video	resolution (downsampled)	time per frame (recognition)	stack size	degrees-of-freedom	recognition-rate	time per frame (tracking)
	384×288	0.20 s	7	(x, y, z)	88%	0.020 s
	160×120	0.042 s	10	(x, y, z, θ)	87%	0.016 s
	384×288	0.27 s	16	(x, y, z, θ)	88%	0.025 s
	384×288	0.072 s	14	(x, y, z, θ)	88%	0.018 s
	192×144	0.32 s	9 1	(x, y, z, θ) (x, y, θ)	35% 45%	0.022 s
dry run (load frames only)	384×288	0.0081 s	-	-	-	-

rate (the complement of the recognition-rate) always is near 100%, a low recognition rate does not necessarily affect the overall performance of the system.

The recognition rates are particularly low when the object is small, when the object has few features, when there is too much clutter in the scene image, or when multiple objects are present in the scene. The reason is that the final feature (or feature-tuple), which leads to a successful recognition of the object, needs to reside on the object. Furthermore both features of a feature-tuple need to reside on the same object. If all corresponding features to the features of the model M are present in the scene S, the probability of randomly selecting a suitable sequence of features is $(|M|/|S|)^n$.

The focus stack must not be self similar. For example the depth of the micro-capacitor in fig. 7 cannot be estimated independently because a planar object which is aligned with the focused plane will have the same appearance regardless whether it is moving upwards or downwards.

The grippers displayed in fig. 6 and 7 show a rough surface due to the etching step in the gripper's manufacturing process. From a manufacturing point of view it would be desirable to smooth out this "unwanted" texture. This surface texture however led to the best of all results because it is rich with features.

As both recognition and tracking are purely combinatorial approaches, the memory requirements for the algorithms are high. In the case of the video showing the micro-gripper and the micro-capacitor, 130 MByte of memory was required for the tracking- and 90 MByte for the recognition-algorithm. State-of-the-art algorithms like *RANSAC* (see [Fischler and Bolles, 1981]) use local feature context so that less features are required. RANSAC in combination with Linear Model Hashing also scales better with number of objects[Shan et al., 2004].

In Geometric Hashing, it is only feasible to compute V_M from a small subset of $M \times M$. By considering only a part of M, one can reduce the size of H_M in a similar fashion. Experimentally H_M was initialised only from features fulfilling $||\vec{m}||\, q_3 \geq 1$ without affecting the tracking performance.

4 Conclusion

The presented algorithm was applied successfully in a variety of environments: the micro camera environment as shown in figure 4, reflected light microscope environment, and transmitted light microscope

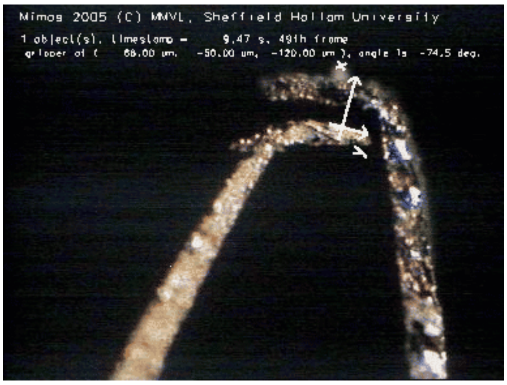

Mimas 2005 (C) MMVL, Sheffield Hallam University
1 object(s), timestamp = 9.47 s, 49th frame
gripper at (68.00 um, -50.00 um, -120.00 um), angle is -74.5 deg.
x

Figure 6: Micro camera image of gripper with uniform background with superimposed pose estimate

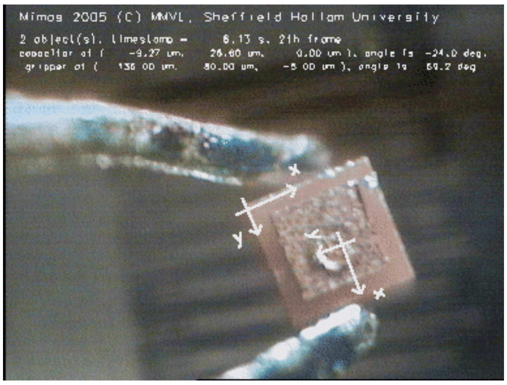

Mimas 2005 (C) MMVL, Sheffield Hallam University
2 object(s), timestamp = 6.13 s, 21h frame
capacitor at (-9.27 um, 26.60 um, 0.00 um), angle is -24.0 deg.
gripper at (136.00 um, 80.00 um, -5.00 um), angle is 69.2 deg
x
y
x

Figure 7: Gripper placing a capacitor (courtesy of SSSA, Sant' Anna) with superimposed pose estimates for gripper and capacitor

environment.

According to [Breguet and Bergander, 2001] the future micro-factory will most probably require automated assembly of micro-parts. The feedback mechanism for the robotic manipulators could be based on computer vision. A robust computer vision system which allows real-time recognition of micro-objects with 4 or more degrees-of-freedom would be desirable.

The algorithm presented in this paper has been implemented using the computer vision library of the *Microsystems & Machine Vision Lab* called **Mimas**, which has been under development and refinement for many years. The library and the original software employed in the MiCRoN-project are available for free at `http://vision.eng.shu.ac.uk/mediawiki/` under the terms of the LGPL.

References

[Begelman et al., 2004] Begelman, G., Lifshits, M., and Rivlin, E. (2004). Map-based microscope positioning. In *BMVC 2004*. Technion, Israel.

[Blayvas et al., 2003] Blayvas, I., Goldenberg, R., Lifshits, M., Rudzsky, M., and Rivlin, E. (2003). Geometric hashing: Rehashing for bayesian voting. Technical report, Computer Science Department, Technion Israel Institute of Technology.

[Breguet and Bergander, 2001] Breguet, J. M. and Bergander, A. (2001). Toward the personal factory? In *Microrobotics and Microassembly III, 29-30 Oct. 2001*, Proc. SPIE - Int. Soc. Opt. Eng. (USA), pages 293–303.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–95.

[Forsyth and Ponce, 2003] Forsyth, D. A. and Ponce, J. (2003). *Computer Vision: A modern Approach*. Prentice Hall series in artificial intelligence.

[Greenspan et al., 2004] Greenspan, M., Shang, L., and Jasiobedzki, P. (2004). Efficient tracking with the bounded hough transform. In *CVPR'04: Computer Vision and Pattern Recognition*.

[Lamdan and Wolfson, 1988] Lamdan, Y. and Wolfson, H. J. (1988). Geometric hashing: A general and efficient model-based recognition scheme. In *Second International Conference on Computer Vision, Dec 5-8 1988*, pages 238–249. Publ by IEEE, New York, NY, USA.

[MiCRoN consortium, 2006] MiCRoN consortium (2006). Micron public report. Technical report, EU IST-2001-33567. `http://wwwipr.ira.uka.de/~seyfried/MiCRoN/PublicReport_Final.pdf`.

[Shan et al., 2004] Shan, Y., Matei, B., Sawhney, H. S., Kumar, R., Huber, D., and Hebert, M. (2004). Linear model hashing and batch ransac for rapid and accurate object recognition. volume 2 of *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 121–8.

Survey and Pareto Analysis Method for Coding Efficiency Assessment of Low Complexity H.264 Algorithms

Y. V. Ivanov, C.J. Bleakley
School of Computer Science and Informatics
University College Dublin
Belfield, Dublin 4
{yury.ivanov, chris.bleakley}@ucd.ie

Abstract

A large number of algorithms have been proposed by researchers to reduce H.264 computational complexity. Currently there is no method for reliably comparing the effectiveness of these algorithms. This paper proposes a method that allows direct comparison of the results obtained for various previously published low complexity H.264 encoding algorithms. The method is based on a new coding efficiency metric for unified bit rate and quality assessment. Pareto analysis is used to derive an optimal reference efficiency complexity curve using standard H.264 encoding tools and parameters. The paper demonstrates application of the method to the assessment of recently published low-complexity algorithms. The method shows that some published low complexity algorithms can be outperformed by simply adjusting the standard video encoder parameters.

Keywords: Video Compression, H.264, Complexity Scaling, Pareto Analysis.

1 Introduction

The H.264 standard [1] was developed by the Joint Video Team (JVT) for video compression in applications where bandwidth or storage capacity is limited. Experimental results show that H.264 provides better coding efficiency than MPEG-4 and H.263 at lower bit rates [2, 3] at the cost of significantly increased computational complexity.

A large number of algorithms have been proposed by researchers to reduce MPEG-4 and H.264 complexity [10-15]. Most of these algorithms are focused on new methods for the most computationally complex components of the video encoder, i.e. Motion Estimation (ME) [8], Discrete Cosine Transform (DCT) coding and Mode Decision (MD).

Unfortunately, there is no method for comparing the effectiveness of the various encoding algorithms. Most papers quote the percentage reduction in computational complexity relative to an arbitrary configuration of the JM reference encoder [5]. Variations in bit rate and perceptual quality are calculated in percentages relative to the reference encoder. Since different encoder configurations are used, the results described in different papers are not directly comparable and can be misleading. The impact of reduced complexity encoding schemes on both visual quality and bit rate is often not clearly stated.

This paper proposes a single metric for accessing the effectiveness of video encoding. This metric allows direct comparison of the results obtained for various previously published low complexity H.264 encoding schemes.

It is well known that the computational complexity of the H.264 encoder can be scaled by simply adjusting the encoding parameter configuration, for example the search range and number of modes to be searched. However, to the author's knowledge, there have been no publications which provide an analysis of the optimal encoding parameters for a required complexity point. This paper presents the results of a Pareto analysis which identifies the optimum operating points for the H.264 video encoder obtained by scaling complexity via modification of the parameter configuration. The analysis is important for two reasons. Firstly, it allows system designers to select the best encoder operating point for a given processor. Secondly, it allows researchers to assess the performance of novel low complexity encoding algorithms relative to that obtained by simply modifying the parameter settings of the encoder. We believe that the methodology applied is generally applicable to other complexity scaling problems.

The effectiveness of a number of published low complexity H.264 video encoding schemes is assessed by projecting their findings onto the Pareto curve. It was found that a number of papers use a sub-optimal reference encoder. While some ambiguity may arise from utilization of different software and hardware platforms, the results suggest that some published methods do not perform as well as simply scaling the video encoding parameters in an optimal fashion.

The paper is structured as follows. Section 2 gives a definition of complexity and introduces a coding efficiency metric for bit rate and quality assessment. Section 3 describes the method itself consisting of complexity analysis for the H.264 reference encoder and development of a Pareto-optimized H.264 complexity curve. Section 4 demonstrates application of our metric to recently published low complexity encoding algorithms. Finally, conclusions are given in Section 5.

2 Theory

There are two major components of computational complexity – time complexity and storage complexity. Time complexity is the number of computational operations required to execute a specific implementation of an algorithm. Storage complexity is the amount of memory required to perform the algorithm. In software implementation these two quantities determine the computational complexity of the algorithm on a specific hardware platform. This paper focuses on execution time, as derived from a reference software implementation of the H.264 encoder.

Complexity scaling of a video encoder, such as H.264, is a trade-off between complexity, bit rate and visual quality. When reducing complexity C it is expected that there will be some increase in bit rate R and in distortion D:

$$R = f_r(C), \quad D = f_d(C) \tag{1}$$

In general, the functions f_r and f_d have different characteristics. The computational complexity of the H.264 encoder for different encoding tool combinations has been partially evaluated in [2, 3]. A brief summary is given in Table 1.

Table 1. Summary of Tool-By-Tool Encoder Complexity Analysis

Encoding tool	*Impact on the PSNR and bit rate*	*Impact on the complexity*
1/4th Sub-pixel accuracy	At low rates PSNR +0.07 dB, +4.7% bit rate. At high rates PSNR +0.04 dB, –12 % bit rate.	Memory access frequency (MAF) +15%.
Variable Block Sizes (VBS)	PSNR +0.07dB to +0.02dB, bit rate –5% to –17%.	Complexity increases linearly: +2.5% for each additional mode. Most of the bit rate reduction (75%–85%) is achieved using 4 modes.
Hadamard Transform	+0.02 to +0.12 dB PSNR, +2.4% bit rate.	MAF +20%.
CABAC	Up to 16% bit reduction.	MAF +25 to 30%.
Search Range Size	Negligible impact on PSNR and bit-rate.	MAF is higher up to 60 times.

It can be clearly seen from these results that the complexity of the video encoder can vary widely with different settings for the tools. Obviously not every combination is optimal in the rate-distortion sense. Unfortunately, the results do not provide information regarding the best combination of tools required to set the H.264 encoder complexity C to a specific operating point C_i. These measurements only provide a general picture of how H.264 complexity can be scaled.

Furthermore, since different low complexity algorithms impact bit rate and distortion in different ways, it is often difficult to determine which algorithm is in fact more efficient at a given operating points. Since changing complexity effects both bit rate and distortion, the need arises to unify both quantities into a single metric. At present, the rate-distortion model is widely used in video coding for making optimal decisions where both bit rate and distortion are important. The model is based on the following Lagrange formula [7]:

$$\min\{J\}, \quad \text{where} \;\; J = D + \lambda \cdot R \tag{2}$$

where D is a distortion measure (usually Sum of Absolute Differences) and R represents bit rate. During video encoding, the Lagrange rate-distortion function J is minimized for a particular value of the Lagrange multiplier, λ.

Based on this, we introduce a coding efficiency metric, W, which is dependent on visual quality loss, ΔD, and bit rate change, ΔR, relative to a reference full complexity encoder:

$$W = \Delta R + \mu \Delta D \tag{3}$$

where ΔR is a percentage, ΔD is PSNR in dB and μ is a constant relating bit rate loss and distortion increase. Thus, for any given computation complexity, C_i, the most efficient encoder can be identified as that providing minimum W.

The constant μ can be interpreted as the percentage increase in bit rate equivalent to 1 dB loss in PSNR. Previous work reported in [16] determined that a 10% decrease in bit rate is roughly equivalent to a loss of 0.5 dB in PSNR. In our work, μ was determined experimentally. Bit rate and PSNR were measured for four video sequences at QP settings of 26, 28, 30 and 32. CIF and QCIF sequences with high and low bit rates were selected for the analysis. The results were plotted, as shown in Figure 1, and the gradient determined by fitting a linear model. The gradient was found to vary between 3.62 and 29.6 with a mean of 12.9. Hence, μ was set to 13 for calculation of W. To check the robustness of the method, extreme values for μ were also tested in the Pareto analysis and were determined to have little impact on the findings.

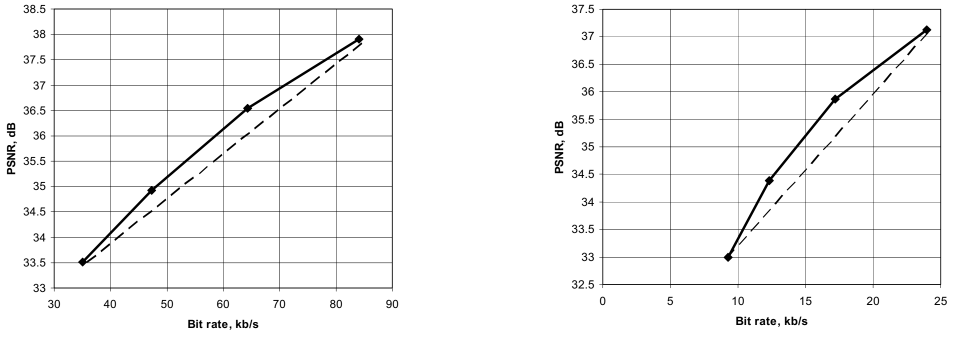


Figure 1. Example of Rate-Distortion graphs for Carphone, QCIF (left) and Container, CIF (right). Dashed lines indicate a gradient of R-D function. Calculated μ values are 11.3 and 3.62 respectively.

Given the coding efficiency metric, W, it is now possible to compare the performance of various encoder configurations. This can be considered as an optimization problem. Given a particular computational complexity requirement, what is the optimum encoder parameter configuration? Given that the encoder parameter configurations form a discrete set, we chose to solve the problem by Pareto analysis [9]. The efficiency of the encoder is assessed across a range of parameter configurations. These results are a projected on to a graph relating coder efficiency to computational complexity. The optimum encoder parameters can then be identified as those points leading to the points (C_i, W_i) which form the Convex Hull of the Individual Minima (CHIM). Parameter configurations corresponding to points inside the CHIM are sub-optimal.

3 Method of Coding Efficiency Assessment

The complexity distribution across all tools for the full complexity mode was profiled, i.e. full search, full VBS, search range 8, CABAC, full Hadamard, sub-pel accuracy on and de-blocking filter on. A wide range of different QCIF and CIF video sequences (298 frames each) with variable content were encoded at different QP settings using the JM 9.5 reference encoder, running on 3GHz Pentium IV with 1Gb RAM. The average results are shown in Table 2.

Table 2. Complexity Distribution Between Different Encoding Tools in H.264

Encoding tool	*% of complexity*
Motion Estimation (including Hadamard Transform)	66
Mode Decision and CABAC	22
Deblocking Filter	5
Transform Coding	4
Other (including I/O)	3

It can be observed that the computational complexity of Motion Estimation dominates and therefore provides most potential of complexity scaling (i.e. search range size, Hadamard on/off, varying of sub-pixel accuracy), which is consistent with [8]. However, since the variation of VBS modes has a high impact on MD complexity, which is 22% with CABAC, VBS can also be considered as an important tool for the purposes of complexity scaling. Switching off the deblocking filter reduces complexity further by 5%. Transform coding in H.264 has negligible impact on complexity (only 4%) since the H.264 standard adopts a separable integer transform with properties similar to DCT instead of the actual DCT [4].

In the next experiments in order to scale the complexity of the encoder from the full mode, search range size and number of VBS modes were varied. The limit of scaling is when the encoder operates in the lowest VBS mode with the search range size is equal to one. Additional simulations show that switching off sub-pixel accuracy in ME reduces complexity further at the cost of a significant bit rate increase and noticeable quality degradation. Utilization of UVLC instead of CABAC for low VBS also has a high impact on the bit rate and leads to an increase in complexity. Switching off Hadamard scales complexity to 28% with minimal bit rate increase and quality degradation.

After encoder profiling it can be concluded that the computational complexity of the standard H.264 encoder can be scaled to 28% by adjusting only the standard encoding tools. The simulation results provide information on the combination of encoding tools and parameters that are optimal for setting H.264 encoder complexity to a specific operating point C_i.

Using the results obtained during the previous experiments, W_i was calculated for each complexity point C_i as given in equation (3) with μ=13. For each (C_i, W_i) calculated, (C_j, W_j) where $C_i \leq C_j$ and $W_i \leq W_j$ we leave only (C_i, W_i) and discard (C_j, W_j). Therefore the CHIM of the Pareto surface is isolated. The variation of W with complexity averaged across all sequences is plotted in Figure 2 with optimal points marked as crossed squares and non-optimal points as hollow diamonds. Values for ΔR, ΔD and W from Figure 2 and H.264 parameter settings are given in Tables 3 and 4, respectively.

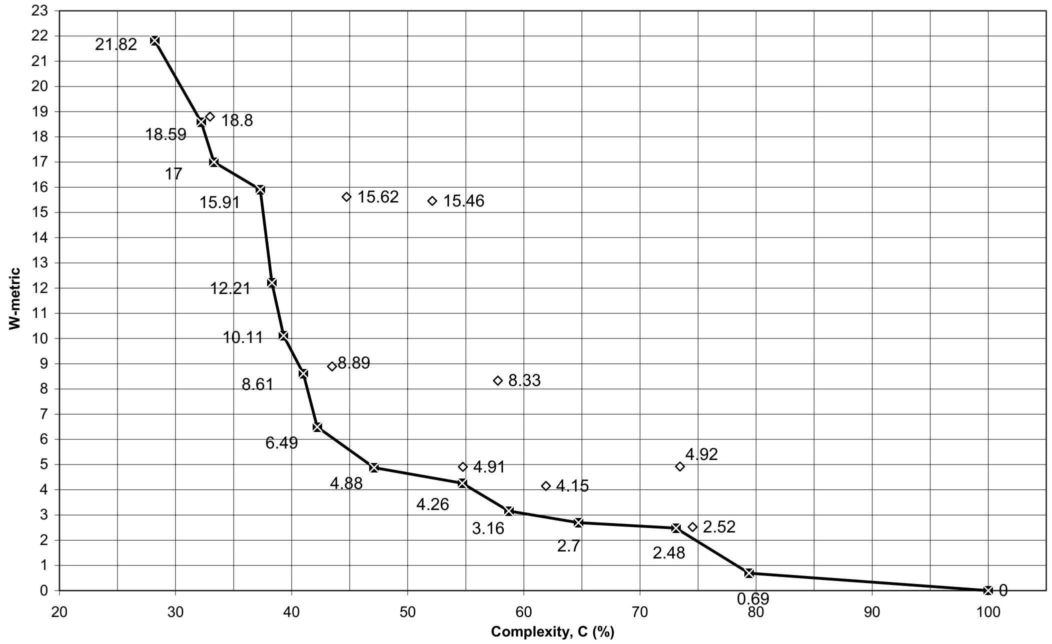


Figure 2. Variation of W-metric with Complexity of Encoder

As can be observed from Table 3, the function f_d has ΔD_{max} = 0.37 dB and, at the same time, complexity is gradually scaled down to 28% from its highest settings, resulting in up to a 17% of a bit rate increase ΔR. Perceptual evaluation of visual quality reveals no anomalies in the encoded sequences. The slight quality degradation measured between full and minimal complexity modes only can be noticed on still images.

Table 3. Optimal Points on Complexity Curve

Value	*Complexity, %*													
	79	*73*	*64*	*58*	*54*	*47*	*42*	*41*	*39*	*38*	*37*	*33*	*32*	*28*
ΔR, %	0.5	0.8	1	1.4	2.4	2.9	4.3	6.4	7.4	9.3	12.3	13.4	14.9	16.9
ΔD, dB	0.01	0.12	0.12	0.13	0.14	0.14	0.15	0.16	0.2	0.21	0.27	0.27	0.28	0.37
W	0.69	2.48	2.7	3.16	4.26	4.88	6.49	8.61	10.1	12.2	15.9	17	18.6	21.8

Table 4. H.264 Encoding Tools Settings for Reference Curve

Encoding tool	*Complexity, %*														
	100	*79*	*73*	*64*	*58*	*54*	*47*	*42*	*41*	*39*	*38*	*37*	*33*	*32*	*28*
VBS	7	7	4	4	4	3	3	3	3	2	2	1	1	1	1
Search range	8	4	8	6	4	6	4	2	1	2	1	4	2	1	1
Hadamard	on	on	on	on	on	on	On	on	on	on	on	on	on	on	off

In order to investigate the behavior of the Pareto curve further from the ‘knee’, additional experiments were performed with the same configuration as for the 100% complexity point, except search range size was increased to 32. W averaged across all sequences was –0.07 while complexity increase is 345% from reference point (or about 4.5 times). Thus, increasing the search range further than 8 for QCIF and CIF images provides little PSNR or bit rate advantage, but does lead to a significant complexity increase. This result is consistent with [2, 3]. It is recommended to avoid large search ranges for low video resolutions.

The Rate-Distortion Optimization (RDO) tool [6] was also tested for the same reason. When switched on, it gives around a 295% complexity increase, while *W* drops to –2.83. Since this complexity point is so far from the Pareto 'knee', again, it is not included in the graphs. However, when working on optimizing of RDO algorithm (e.g. [12]), developers may want to extend the complexity curve with this point for assessment purposes.

Finally, it can be concluded that the optimal set of H.264 parameters given in Table 4 is generally suitable for the purposes of H.264 complexity scaling. The complexity curve can be refined further by running more experiments in the desired range.

4 Assessment of Published Algorithms

For the purpose of illustration, several recently published algorithms were investigated [10-15]. The method proposed in [10] utilizes motion vector cost and previous frame information for adaptive threshold cost based selection of macroblock (MB) mode. Fast multi-block selection [11] is based on the idea of detecting fast and slow moving areas of the frame and processing them differently. The algorithm proposed in [12] utilizes a special block matching order combined with SAD pre-calculation for reducing ME complexity and for skipping spatial predictive coding. The method [13] is based on the correlation of motion vectors across the various MB partitions. The block mode selection algorithm in [14] relies on two factors – complexity of macroblock and MB mode from previous frame. The low complexity encoding scheme described in [15] uses VBS prediction from the surrounding MBs.

Based on the simulation results provided by the authors, the W-metric was calculated for each algorithm and plotted along with the reference complexity curve, as shown in the Figure 3.

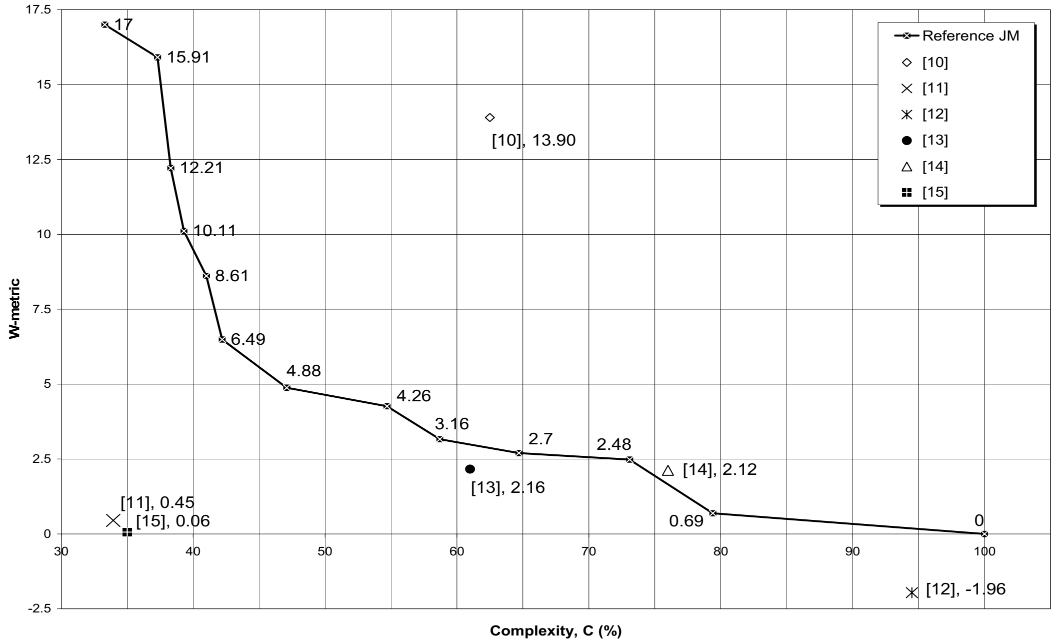


Figure 3. Comparison of Published Algorithms with Reference Complexity Curve

It can be seen that algorithms [10] and [13] provide complexity reduction around 40%, but [10] at the cost of significant bit rate increase (resulted high *W*). Better results than [10] can be achieved by running H.264 with the parameters given at Table 4 for the complexity point of 58%.

Alternatively, to achieve W around 13.9 as in [10], the 38% complexity point can be chosen, thus, the same bit rate and quality results will be achieved by simply reducing VBS number and search range size. The algorithm in [14] is also not optimal – with only 25% of computational reduction the resulting W lies almost on the Pareto convex hull. It is preferable to use the configuration associated with the 79% complexity point.

In contrast, algorithms in [11] and [15] perform much better than the scaled reference encoder. Both provide quite significant complexity reduction of around 65% and have relatively low W. This can only be obtained by scaling the reference encoder to a complexity point of 70–80%. Thus, these algorithms are about 2.3 times more efficient than reference JM encoder.

The algorithm in [12] uses an improved rate-distortion technique and reports a significant complexity reduction, but calculated relative to a reference encoder with RDO on (i.e. relative to 295% instead of 100%). Plotted on the curve, it results of $W = -1.96$ without reducing complexity significantly relative to the reference encoder with RDO off (only 10%). However, since similar W can only be achieved by running JM with RDO tool, the algorithm provides better bit rate than the reference encoding configuration.

5 Conclusions

In this paper a method for accessing the effectiveness of low complexity H.264 video encoding algorithms was proposed. The method allows direct comparison of the results obtained for various previously published low complexity H.264 encoding schemes. It has been demonstrated that by introducing a coding efficiency metric as a single measure, the assessment of bit rate and perceptual quality can be unified.

To the author's knowledge there have been no publications providing an analysis of the optimal encoding parameters for a required complexity point. The computational complexity of the H.264 encoder was scaled by adjusting the encoding parameter configuration. Pareto analysis was introduced for identifying the optimum operating points. The obtained results not only demonstrate the general picture of H.264 complexity scaling, which was found to be consistent with other publications, but, more important, they allow systems designers to select the optimum encoder operating point for a given processor.

The effectiveness of a number of published low complexity H.264 video encoding schemes was assessed by projecting results provided by the authors onto the Pareto curve. It was found that a number of papers use sub optimal configurations for the reference encoder (i.e. [10], [14]). While some ambiguity may arise from utilization of different software and hardware platforms, the results suggest that these methods do not outperform a reference encoder with encoding parameters scaled in an optimal fashion.

Acknowledgements

This research project is sponsored by University College Dublin, Ireland.

References

[1] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. (2003). *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H264|ISO/IEC 14496-10 AVC).* document JVT-G050d35.doc, 7th Meeting: Pattaya, Thailand.

[2] Implementation Studies Group of ISO/IEC. (2002). *Main Results of the AVC Complexity Analysis*. ISO/IEC JTC1/SC29/WG11, Klagenfurt.

[3] Saponara, S., Blanch, C., Denolf, K. and Bormans, J. (2002). *Data Transfer and Storage Complexity Analysis of the AVC Codec on a Tool-By-Tool Basis.* ISO/IEC JCT1/SC29/WG11, M8547, Klagenfurt.

[4] Ostermann, J., Bormans, J., List, P., Marpe, D., Narroschke, M., Pereira, F., Stockhammer, T. and Wedi, T. (2004). Video Coding with H.264/AVC: Tools, Performance, and Complexity. *IEEE Circuits and Systems Magazine*, 4.1: 7-28.

[5] JVT reference software JM 9.5, on the Web: http://iphome.hhi.de/suehring/tml.

[6] Sullivan, G. J. and Wiegand, T. (1998). Rate-Distortion Optimization for Video Compression. *IEEE Signal Processing Magazine*, 15.6: 74-90.

[7] Everett III, H. (1963). Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources. *Operations Research*, 11: 399-417.

[8] Kuhn P. (1999). *Algorithms, Complexity Analysis and VLSI Architecture for MPEG-4 Motion Estimation*, Kluwer Academic Publishers: Boston.

[9] Das I. (1999) On characterizing the 'knee' of the Pareto curve based on Normal-Boundary Intersection, *Structural and Multidisciplinary Optimization*, 18.3:107-115.

[10] Ahmad A., Khan N., Masud S. and Maud M.A. (2004). *Selection of Variable Block Sizes in H.264*, in Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'04), 3:173-176.

[11] Chang A., Wong P.H.W., Yeung Y.M. and Au O.C. (2004). *Fast Multi-Block Selection for H.264 Video Coding*, in Proc. of IEEE Int. Symp. on Circuits and Systems (ISCAS'04), 3:817-820.

[12] Han K. and Y. Lee Y. (2004). *Fast Macroblock Mode Decision in H.264*, in Proc. of IEEE Region 10 Conf. (TENCON'04), 1:347 – 350.

[13] Kuo T. Y. and Chan C.H. (2004). *Fast Macroblock Partition Prediction for H.264/AVC*, in Proc. of IEEE Int. Conf. on Multimedia and Expo (ICME'04), 1: 675–678.

[14] Yu A.C. (2004). *Efficient Block-Size Selection Algorithm for Inter-Frame Coding in H.264/MPEG-4 AVC.*, in Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'04), 3: 169-172.

[15] Li G. L., Chen M. J., Li H. J. and Hsu C. T. (2005). *Efficient Motion Search and Mode Prediction Algorithms for Motion Estimation in H.264/AVC*, in Proc. of IEEE Int. Symp. on Circuits and Systems (ISCAS'05), 6: 5481 – 5484.

[16] Bjontegaard G. (2001). *Calculation of average PSNR differences between RD curves*, ITU-T VCEG Meeting, Austin: VCEG-M33.

Posters

A High Performance Distributed System for Astronomical Image and Data Analysis.

Adrian Collins†
Astronomy & Instrumentation Group,
Department of Applied Physics & Instrumentation,
Cork Institute of Technology, Cork, Ireland.
adrian.collins@cit.ie

Stephen O'Driscoll
Astronomy & Instrumentation Group,
Department of Applied Physics & Instrumentation,
Cork Institute of Technology, Cork, Ireland.
stephen.odriscoll@cit.ie

Niall Smith
Astronomy & Instrumentation Group,
Department of Applied Physics & Instrumentation,
Cork Institute of Technology, Cork, Ireland.
niall.smith@cit.ie

Abstract

Modern astronomical imaging systems are generating enormous amounts of data. These data volumes present significant challenges in terms of storage and analysis. A new photometric imager currently being developed at CIT has the capacity to acquire 1.2 million images (1 TByte) per night. Processing these data in an efficient way will require novel analysis procedures to be developed. One approach - distributed processing, offers a potential solution to this problem that is both cost effective and highly efficient.

In this paper we present the design of a distributed system currently being developed to manage and process data from this new instrument. The system called Quick:silver, has been developed using Microsoft's .NET Framework and utilizes MATLAB as its processing engine. We describe the motivation behind its development and an overview of its design. A preliminary characterization of the system indicates that a distributed processing approach to dealing with data from this instrument can provide significant improvement in the rates at which image processing procedures can be applied.

Keywords: Astronomy, Distributed Computing, MATLAB, Image Processing

† Corresponding Author.

1 Introduction

A number of instrument projects in progress around the world highlight the significant role astronomy plays in the development of high performance imaging systems. The design of these instruments is driven by specific scientific objectives which often require the development of highly novel systems possessing impressive performance characteristics. A key feature of these projects is that their innovative nature offers a unique window for studying complex phenomena currently unexplored by more conventional systems. The challenges presented by the development of such systems do not only apply to their design and manufacture but also extends to the process of dealing with the data these instruments generate. In particular, two directions of instrument development highlight the data generation challenges currently faced by astronomy; these movements are aimed at i) increasing the imaging area of the sensor used and ii) increasing the rate at which images are acquired. Increasing the area of a sensor allows a telescope to image larger areas of the sky at a given resolution and is useful for conducting surveys or detection programmes (such as searching for near-earth objects). The *Quest* Large Area Camera at Palomar Observatory, for example, is constructed using 112, 600×2400 pixel charge coupled devices (CCDs), giving it a total of 161×10^6 pixels [1]. Another instrument intended for survey applications is *OmegaCAM* which uses 32 CCDs (2048×4096 pixel), giving a total of 268×10^6 pixels [2]. While these instruments maximize sensor size for their intended application, others maximize speed. Perhaps the best known instrument in this category within the astronomical community is *ULTRACAM* [3]. ULTRACAM is a 3-channel instrument which images in the ultraviolet, visible and red wavebands. For each channel it employs a single 1024×1024 pixel CCD which can be operated in an optimized "windowed mode" to provide frame rates in excess of 500Hz (500 images/s). Clearly, from the description of these projects, they have the potential to produce huge amounts of data. These data not only need to be stored but must also be processed and analyzed in acceptable time. Of concern is the fact that this data trend does not show signs of abatement, in fact, despite the impressive data generation potential of these projects, a new astronomical instrument called TOφCAM (**T**wo-Channel **O**ptical ***Ph***otometric ***I***maging **Cam**era – pronounced "*toffee- cam*") stands at the threshold of generating data at a rate of 1TByte per night. The prospect of dealing with enormous data sets of this kind requires that existing approaches to astronomical image and data processing change significantly, not only in the manner in which processing is conducted but also in relation to how applications are developed and deployed. In response to the processing requirements of this instrument, a distributed image and data analysis pipeline called Quick:silver is being developed using the Microsoft .NET Framework. At its current stage of development Quick:silver supports the deployment of image processing and data analysis applications developed in MATLAB to multiple networked computers which have MATLAB installed, providing a novel method for processing astronomical images in a more efficient way.

In this work, we present the motivation, design and the initial results of performance tests on Quick:silver, a distributed computing "backbone" to support the use of MATLAB as a delivery vector of scientific and numeric processing applications during the development and commissioning of the TOφCAM instrument. In the remainder of this section we briefly outline the scientific motivation and characteristics of TOφCAM. We illustrate the significant impact a distributed computing approach can have in dealing with the enormous amounts of image data associated with the instrument and we outline our rationale for choosing MATLAB as the development and deployment method for Quick:silver. In Section 2, we describe the design and implementation of the system, illustrating how the system supports deployment from MATLAB. In Section 3, we present initial results on Quick:silver performance in applying a MATLAB image processing routine to a set of test images. Finally, Section 4 contains our concluding remarks on the system and an outline of our planned future work.

1.1 High-Time Resolution Imaging and TOφCAM

High-time resolution imaging (HTRI) is one example of the use of highly specialized instrumentation which offers a unique opportunity to study specific physical phenomena. An emerging field in astronomy, HTRI is applied where accurate measurements of light intensity are required over very short time-scales, as short as milliseconds. This type of imaging can be used in the study of objects exhibiting either rapid or transient brightness variation characteristics, the fastest examples being millisecond *Pulsars*. HTRI has also been applied to enhance ground based telescope observations by overcoming atmospheric turbulence limitations on the resolution of telescopes. TOφCAM is a high performance imager currently being developed by the Astronomy & Instrumentation Group (AIG), in the Department of Applied Physics & Instrumentation at Cork Institute of Technology. TOφCAM is a two-channel photometric imager which splits an incoming light beam into two channels (or wavebands) using a dichroic beam splitter. Each channel image is then sensed using a 512×512 Andor iXon CCD camera capable of operating in full-frame mode at 34Hz (and a "windowed mode" up to 440Hz). With each channel being acquired at this frame rate and using for example, 14-bit pixel digitization, the data storage rate is expected to be ~29MBytes/s. If we assume that the instrument operates continuously over a 10 hour observing run, it will acquire ~1.2 million raw images occupying ~1TByte.

1.2 Distributed Processing for Image and Data Analysis

Processing 1.2 million images per night would be extremely inefficient if conducted by a single computer. For example, if the computer could fully analyze each image in 1 sec, the entire data set would require 2 weeks to process. This latency limits the number of scientific applications the instrument can be applied to, impacting on the overall scientific return possible from the system. Ideally, in the study of phenomena with rapidly varying characteristics, we would like to be able to monitor in real-time. Many astronomical objects, such as *Pulsars* exhibit characteristic variability on time-scales of minutes or less, which, if we could detect quickly enough, would allow us to adapt an observing schedule to study these changes in greater detail and facilitate coordinated multi-location observing. To reach this level of observational flexibility we would require processing times of 0.1 seconds per image or better on a single computer. This would reduce the full data set process time to just over 1 day. If an in-line processing system was used, most of the data could then be processed before the nights observation was completed and facilitate the possibility of dynamic scheduling. Alternatively, a different paradigm for processing could be employed, utilizing not one but a group of computers processing sections of the data set in a coordinated, parallel fashion. This paradigm, known as distributed processing, represents a very real possibility for solving this type of problem offering both an efficient and cost effective solution. Distributed processing involves the division of a linear process which runs on a single computer, into a number of independent processes that can be executed on several computers in parallel [4]. The attractiveness of a distributed approach lies in the relationship between process speedup and the number of computers (or nodes) used. Ideally, if N nodes are used then the expected process speedup is N; however, in practice this will apply only to a point, after which the addition of nodes will not result in any significant process speedup and may in fact reduce overall process speed. The degree to which a system adheres to this linear relationship is called *scalability*. Now, even if an image were processed in 1 sec on a single computer, based on the expected behavior of a distributed system an effective image processing rate of 0.1 seconds could be achieved through the use of $N = 10$ nodes. There is however a special case in which the expected speedup for a given number of nodes can be exceeded, suggesting that fewer nodes could achieve an equivalent or faster processing rate. This behavior, known as *superlinear speedup*, is achievable if the distributed process is very efficient, having low communication overhead, efficient algorithm implementation or optimized memory usage, but does have implications for the scalability of the system.

1.3 MATLAB

"MATLAB has established itself as the de-facto standard for engineering and scientific computing" [5]. Among the features which have led to the widespread acceptance of MATLAB are its ease of use, high level of abstraction which provides scientist's and engineers with a powerful rapid prototyping and development resource, and its high quality numerical routines. Coupled with this are its many toolboxes and functions which extend its application potential to many different areas such as image processing. These factors make MATLAB ideal for the development of an instrument *Data Reduction Pipeline (DRP)*. A *DRP* can be defined as a *set of logically contiguous data* *processing operations designed to transform data from one functional level to another* [6]. For TOφCAM data, these processing operations include instrument correction, determination of objects of interest in each image and the generation of instrumental brightness data for each of the objects under examination.

2 System Implementation

Quick:silver has been developed using the Microsoft .NET Framework and controls all aspects of process distribution and communication. The .NET Framework provides high level functionality allowing scientists and engineers to rapidly develop applications and allows easy integration into future technologies [7].

MATLAB is used as the processing engine located at each node of the system. A clear advantage of using MATLAB in this way is that existing resources on a network can be utilised and any scripts already developed can be adapted for use with the system. While other efforts have been made to provide distributed MATLAB operation [8][9], Quick:silver's primary advantage over these approaches are its simplicity and flexibility. Quick:silver will not require the user to have any knowledge of the distributed system, it will be transparent, being presented as a simple function that can be included in a MATLAB script. Transparency is an important factor in the acceptance of a distributed computing application [10].

At this stage of development Quick:silver supports the distribution of MATLAB scripts only, but it is planned in future work to allow executable programs to be distributed also. This means, for example, that image processing applications developed perhaps in C/C++, which use the OpenCV library, can also be used with the system. A general system schematic is shown below illustrating how Quick:silver currently interacts with the user and nodes (Figure 1).

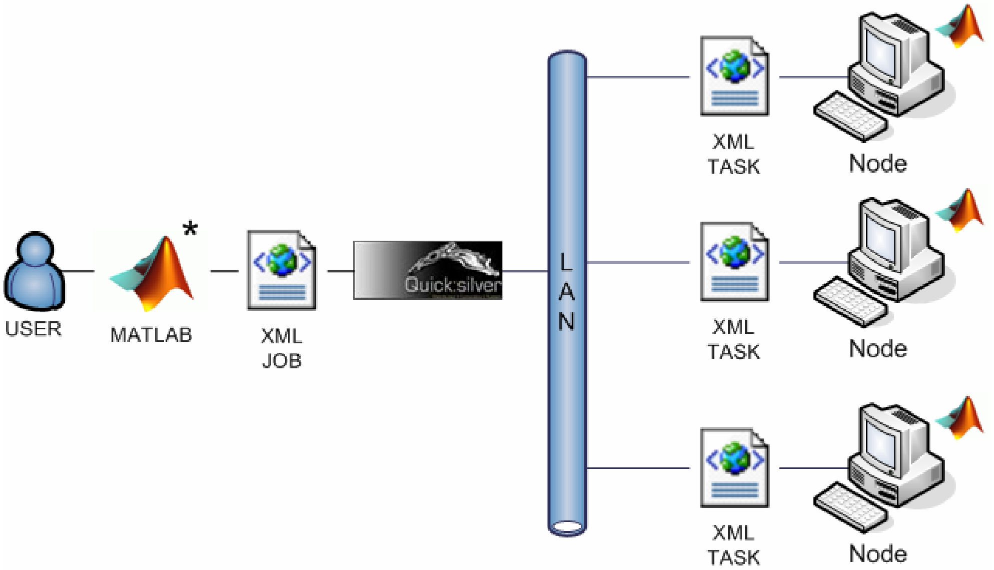


Figure 1. Schematic representation of Quick:silver system. The user interacts with the system by calling a function within MATLAB. This function will allow the user to specify a job in an XML document which is sent to Quick:silver for processing. (* This stage of the process is not yet implemented).

2.1 Process Description

Quick:silver uses a client–server process model. Each node is required to have a version of MATLAB (with additional toolboxes where necessary) and the Quick:silver client program installed. Currently, to use the system, the user must compose an XML (eXtensible Markup Language) formatted document detailing; i) a list of images to process and ii) the MATLAB script(s) to be used on each image. The document in which this information is specified is called a JOB and defines which node will process an image, and also facilitates the use of different MATLAB scripts on different images. A list of participating nodes must also be provided but this is currently not specified in XML format. Once started, the Quick:silver server then manages all aspects of the distributed process for the user which includes splitting the JOB into TASKs for processing by a single node. Later versions will automatically generate the XML JOB description using high level information provided by the user. XML was used to specify JOBs and TASKs within the system because it is human readable, supporting traceability and validation, and is extendible allowing for the inclusion of new process parameters as the system develops. These XML files are very small in size and require very short processing (parsing and validation) times by a node. It is planned that the user will be able to integrate script distribution transparently into existing analysis procedures through the use of a simple Matlab function call. Interacting through a simple Matlab function makes code more readable and allows an existing script to be modified for distribution with little modification to the original code. A process overview is provided below (Figure 2).

- User starts system and specifies:
  - XML JOB description to use,
  - list of potential nodes.
- Quick:silver queries nodes to confirm participation in process.
- Quick:silver splits JOB into smaller TASKs and sends each node a TASK description.
- Node reads TASK and downloads image (and script if needed) from the server.
- On completion of current TASK, node requests next TASK.
- When all TASKs are completed, each node returns its TASK results.
- Results are collated and presented to the user.

Figure 2. Overview of Process Distribution.

3 Performance Characterisation

The system was assessed by applying a simple image pre-processing procedure to a group of images and determining a performance metric based on the observed results. A test image dataset was composed of four groups, each group consisting of 100 images. Each group corresponded to a size category of 256×256, 426×438, 852×876 and 1704×1752 pixels. The time taken for the distributed system to apply the pre-processing procedure to all the images in a group was recorded. This process was repeated a total of ten times, each time incrementing the number of nodes in the system. The times recorded for a particular processing run on a group were then used to determine a performance metric for the system.

3.1 Image Pre-Processing Procedure

To illustrate the application of the system to image processing problems we composed a Matlab script, using functions from the Matlab image processing toolbox, to perform a pre-processing procedure which might be used to locate stars within an image of a sparse star field. While this procedure is simple, it is nonetheless effective and easy to verify which supports direct comparison of performance with other distributed computing systems. The script convolves a 5×5 low pass (or smoothing) filter with an image to remove noise artefacts (which generally exhibit rapid spatial variation), then the edges within the image are detected using a Sobel edge detector. Matlab's edge detection function with Sobel option produces a binary image. The resulting binary image is then clustered to identify localized spatial features. Finally, the centroid of each of these features (which are assumed to be stars) is determined. The source code for this procedure is shown below (Figure 3).

```
F = [1 4 6 4 1];
F2 = F'*F;
F2n = F2./sum(sum(F2));               % Normalized 2D low pass filter
Filtered_I = filter2(F2n,I);          % Apply filter to image I
Edge_I = edge(Filtered_I,'sobel');    % Edge detection
L = bwlabel(Edge_I);                  % Cluster binary image
xy = regionprops(L,'centroid');       % Find centroid of each cluster
```

Figure 3. Matlab code for test image pre-processing routine.

3.2 Characterization Results

A clear distinction is observed between the processing times (measured by elapsed time) required for each image size category which can be attributed to the additional processing times associated with larger image sizes (Figure 4). This is evident in the separation between each set of image category processing times. Within each set of category results we observe that the addition of nodes reduces the processing time associated with that category, highlighting an advantage of processing in a distributed fashion. In order to determine more accurately the effect of additional nodes, a procedure of normalizing each set of image category results is applied. This procedure allows us to observe the relative improvement in rate of processing, or *speedup* [11], provided by the addition of nodes to the distributed system. The *speedup*, S_N , provided by a distributed system comprising *N* nodes is defined by:

$$S_N = \frac{T_1}{T_N}, \qquad \text{Eqn. 1}$$

where T_1 is the time to complete a process on a single node and T_N is the time to complete the process on a distributed system comprising *N* nodes.

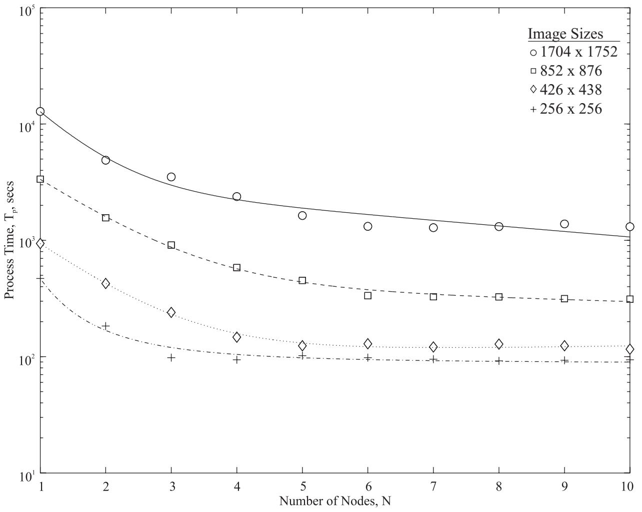


Figure 4. Process Time Comparison. As nodes are added to the distributed system the time to process each image category reduces. The results for each category are shown on the same axes for comparison.

Analyzing the speedup of a system is important in order to assess its performance scalability. Distributed systems exhibiting linear speedup achieve predictable changes in processing rates with the addition of extra nodes, which is a direct result of the processing rate following a 1/*N* relationship. The results of our analysis suggest that the relationship between speedup and node number exceeds linear behavior and is thus *superlinear* in nature (Figure 5).

There is a lack of clear consensus in the scientific literature regarding the cause of superlinear speedup. The explanations most commonly cited refer to situations where the communications overhead may be very low, the processing algorithm being used is very efficient or the data sets are small enough to fit in the computers' cache memory. Of these, the third possibility is most frequently cited as being the most probable [12]. The cause of superlinear speedup in Quick:silver may be attributed to some of these explanations but requires further investigation.

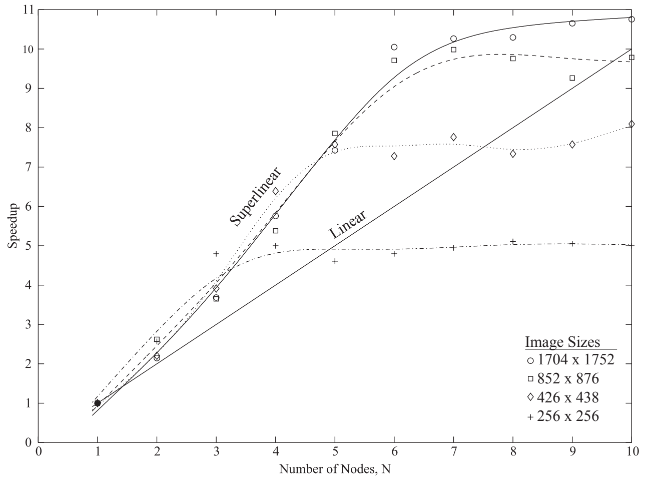


Figure 5. Speedup Comparison. The process speedup observed as the number of nodes suggests a superlinear behaviour for range of image sizes tested.

For each image category we observe that the (superlinear) speedup breaks down after a certain number of nodes is exceeded. This breakdown is seen as a flattening of the speedup behavior, indicating that the addition of nodes brings no further benefit to system performance. This can be explained by considering the frequency at which task requests are made to the server. The frequency of task requests will depend on the number of nodes and the size of the images being processed. If the images are small, then each node will be able to process them quickly, as nodes are added, the frequency of task request to the server will increase. Larger image sizes will result in longer node processing times which will initially offset increases in the frequency of request to the server, allowing speedup behavior to be maintained while more nodes are added. Eventually however, a number of nodes will be reached after which the server is unable to meet the increased task demand. The performance of the system will therefore saturate, the speedup at which this occurs being governed by how quickly the server responds to requests. We would expect for this reason the saturation value to be reached at lower node numbers when the image size is small and higher node numbers when the image size is large. We would also expect the speedup rate to remain consistent between image category tests if this explanation were true (Figure 5).

4 Conclusion and Future Work

In this paper we have presented a distributed image and data analysis system designed for use with CIT's TOφCAM astronomical imager. This system, called Quick:silver, was designed to address the problem of processing extremely large sets of data generated by the instrument in a reasonable time. A requirement of this design was that the system could be integrated into existing MATLAB procedures without expecting the user to have any knowledge of the distributed system architecture. Initial tests of the system's performance provide encouraging results, providing evidence that the use of distributed processing significantly reduces data set processing times. A characterisation of the system's performance indicates a speedup behaviour that is superlinear in nature. While this is appealing at one level, allowing us to achieve higher expected processing power for a given number of nodes, its presence ultimately limits the scalability of the system.

Future work involves investigating how the scalability of the system can be regulated, which includes more resolution of the origin of the superlinear behaviour observed in the performance of the system. Further work must be conducted on the integration of the system with MATLAB, allowing process distribution through interaction with a single MATLAB function. Additional work on the automated composition of XML JOB description files is needed to implement this feature. Due to the cost associated with requiring each node to have a MATLAB installation, we intend to explore the feasibility of alternatives to MATLAB such as Scilab, Octave and RLab while also developing the system to support executable applications. The concept of dynamic reallocation of tasks is also a feature that will be investigated as this has applications in coordinated telescope observations where nodes are distributed on the internet at different telescopes around the world. Dynamic reallocation of tasks is useful when contact to a node is lost for some reason and would allow tasks to be reassigned to active nodes.

References

[1] Graham, M. et al. (2004). Palomar-QUEST: A case study in designing sky surveys in the VO era. *ADASS XIII*, ASP Conference Series 314.

[2] Deul, E.R. et al. (2002). OmegaCAM: The 16k x 16k Survet Camera for the VLT Survey Telescope. *Proc. SPIE Survey and Other Telescope Technologies and Discoveries*.

[3] Dhillon, V. et al. (2002). ULTRACAM – an ultra-fast, triple beam CCD camera. *The Physics of Cataclysmic Variables and Related Objects*, ASP Conference Series, 261.

[4] Grosu, D. et al. (2005). Noncooperative load balancing in distributed systems. *Journal of Parallel and Distributed Computing*, 65:1022-1034.

[5] Fernandez, J. et al. (2003). Performance of Message-Passing MATLAB Toolboxes. *Lecture Notes in Computer Science*. Springer.

[6] O'Driscoll, S., Smith, N., (2004). On the realization of an automated data reduction pipeline in IRAF – the PhotMate system, *Proc. of IOP/ASGI Astronomy and Astrophysics in Ireland*.

[7] Champlain, M., Patrick, B. (2005). C# 2.0 Practical Guide for Programmers. Morgan Kaufmann

[8] Kepner, J., Ahalt, S. (2004). MatlabMPI. *Journal of Parallel and Distributed Computing*, 64:997-1005.

[9] Chen, Y., Tan, S. (). MATLAB*G: A Grid-Based Parallel MATLAB. http://citeseer.ifi.unizh.ch/649021.html.

[10] Cunha, J. et al. (2005). Future trends in distributed applications and problem-solving environments. *Future Generation Computer Systems*, 21:843 – 855.

[11] Bevilacqua, E., Piccolomini, E. (2000). Parallel image restoration on parallel and distributed computers. *Parallel Computing*, 26:495-506.

[12] Coddington, P. (2000). Parallel Programming Models and Performing Analysis. http://www.dhpc.adelaide.edu.au/education/dhpc/2000/performance.pdf

Stereo Vision for the Detection of Road Signs in Dusk and Nighttime Traffic Sequences

Simon McLoughlin[1], Catherine Deegan[2], Ciara Mulvihill[2], Stephen Foy[2], Conor Fitzgerald[3], Charles Markham[1]

[1]Dept. of Computer Science, National University Ireland Maynooth, Co. Kildare, Ireland
[2]Dept. of Engineering, Institute of Technology Blanchardstown, Dublin 15, Ireland
[3]Tramore House Regional Design Office, Tramore House, Tramore, Co. Waterford, Ireland

Abstract

This paper presents a mobile stereo vision system designed for the assessment of road signage and delineation (lines and reflective pavement markers or “cat’s eyes”). Using the system it has been shown that retro-reflectors, and in particular road signs, can be identified by the nature of their reflective properties. Any objects examined can also be accurately positioned on a National grid through the fusion of stereo vision with GPS technology.

Keywords: Stereo Vision, GPS, Mobile Mapping, Road Sign Detection

1 Introduction

Accurate terrestrial mobile mapping systems have been in operation for over two decades. Through advances in Global Positioning System (GPS) technology and photogrammetry, features can be extracted from an image automatically and positioned in a global reference frame to an accuracy of less than one metre. The early 1990’s in particular, saw two major developments in mobile mapping through the production of the GPSVan [1, 2] by the Ohio state University in the USA and the VISAT van [3, 4] developed at the University of Calgary in Canada. Both systems have been modernised through the years and are now commercial concerns. Such systems boast feature accuracy levels of the of the order of 0.3 metres or better, although this seems to be a theoretical limit based on specifically designed target objects. The challenges that remain in this area today can be split into two domains. The development of algorithms and techniques to automate feature extraction from the vast amounts of data acquired by such systems and the construction of more accurate, more robust, more portable and less expensive mapping systems for object positioning. This paper describes a technique that exploits the surface properties of retro-reflective materials for the automated extraction and localization of road signs in dusk and nighttime road settings.

2 Road Sign Detection

The principal cue used in the detection of road signs is different to the conventional techniques that use either colour or shape as the main cues like in [5] and [6]. By carrying out experimental analysis of the material properties composing signs, interesting insights are noted. Road signs are retro-reflectors meaning they reflect significant amounts of incident light back to the source. They are purposely engineered as imperfect retro-reflectors, meaning they actually reflect light back toward the source in a non-uniform cone, see figure 1.

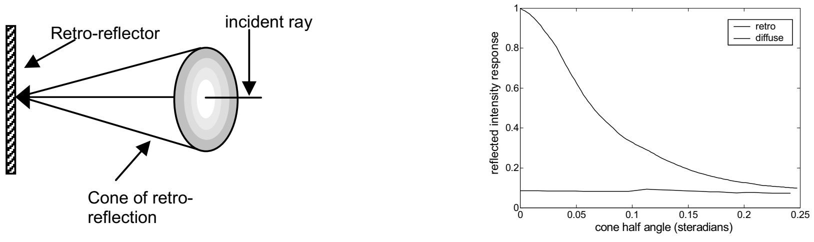


Figure 1 - (a) Cone of retro-reflection as produced by a road sign and (b) Reflected light intensity response inside the cone of retro-reflection for a typical road sign

Hence, the luminance levels inside the cone of retro-reflection increase as the observer moves in towards the centre. An observer at the cone centre will notice a significantly greater signal than one at the cone edge. It is this property that allows retro-reflectors to be identified in a stereo pair of correctly positioned cameras. One camera is placed very close to the cone centre where luminance levels are high and the other is placed toward the cone edge where luminance levels are lower. For actual data acquisition, one camera is located beside a powerful 500 watt infrared source (20 cm from source) and acts as the cone centre sensor, where the signal observed from retro-reflectors is significantly greater than the one observed by the cone edge sensor, which is located on the opposite side of the roof-box (1 metre from source) acquisition system. A comparative analysis of the light intensity information in the cone centre camera and the cone edge camera is used as the main cue to identify retro-reflective surfaces.

3 Results

Figure 2 below shows the Easting and Northing locations of road signs detected on a 6.5 kilometre GPS trail. The data was acquired in a dusk setting and in an urban environment. 57 detections of different objects were made by the technique. Of this figure, 47 objects were road signs and the remainder were false positives. 59 road signs in all are actually present on this road segment, which implies a detection rate of 80% and a false positive rate of 17%. False positives were due to specular reflections and incorrect correspondence matches. Undetected signs included those that were poorly illuminated by the source.

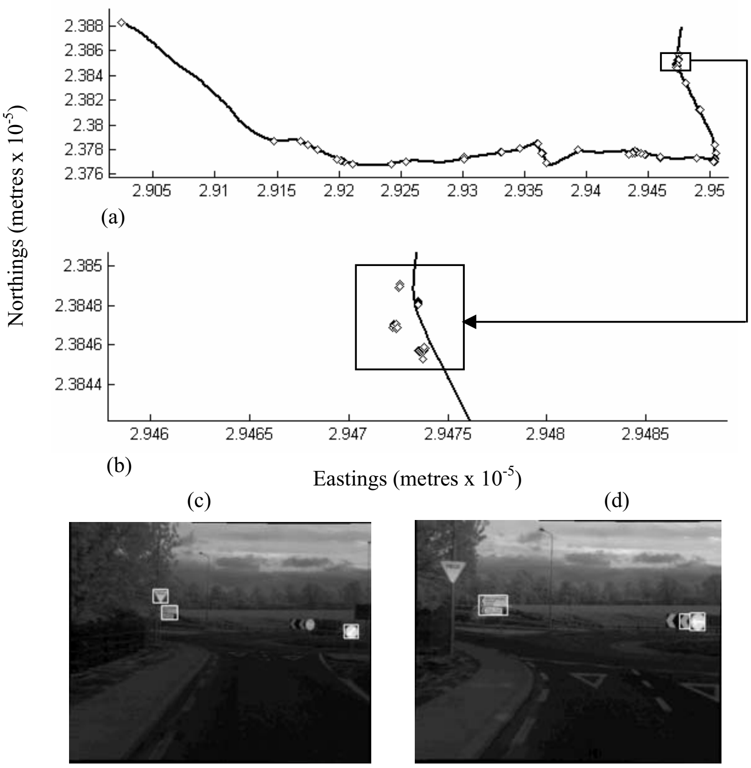


Figure 2 - (a) The locations of the detected features on a 5.5 kilometre GPS trail heading west (b) A closer view of part of the road segment showing 5 features detected and their Northing and Easting locations (c, d) Two images showing the detection of the road signs on the GPS trail in (b)

References

[1] Bossler, J., C. Goad, P. Johnson and K. Novak, 1991. GPS and GIS Map the Nations Highways, *GeoInfo Systems Magizine*, pp. 26-37, March 1991.

[2] Bossler, J., 1992. GPS Van – Input to GIS, *Proceeding of ION GPS*, 1992.

[3] El-Sheimy, N., Schwarz, K.P., 1993. Kinematic Positioning in Three Dimensions Using CCD Technology, in *Proceedings of IEEE Vehicle Navigation and Infromation Systems Conference (VNIS)*, Ottawa, 1993.

[4] Schwarz, K.P., Martell, H.E., El-Sheimy, N., Li, R., Chapman, M.A., Cosandier, D., 1993. VIASAT – A Mobile Highway Survey System of High Accuracy, in *Proceedings of IEEE Vehicle Navigation and Infromation Systems Conference (VNIS)*, Ottawa, 1993.

[5] Habib, F., Uebbing, R., Novak, K., 1999. Automatic Extraction of Road Signs from Terrestrial Color Imagery, *Photogrammetric Engineering and Remote Sensing*, Vol. 65, No. 5, pp. 597-601, May 1999.

[6] De la Escalera, A., Armingol, J.M., Mata, M., 2003. Traffic Sign recognition and analysis for intelligent vehicles. *Image and Vision Computing*, vol. 21 pp. 247-258, 2003.

Dictionary Based Lip Reading Classification

Dahai Yu, Alistair Sutherland, Paul F. Whelan, Ovidiu Ghita
School of Computing
DCU, dahai.yu2@mail.dcu.ie

Abstract

Visual lip reading recognition is an essential stage in many multimedia systems. The use of lip visual features to help the audio or sign/hands recognition is appropriate because this information is invariant to acoustic noise perturbation. In this paper, we describe our work towards the development of a robust method for lip reading feature classification that extracts the lips in color images by using EM-PCA feature extraction and K-Nearest-Neighbor classification.

1. Introduction

In this paper we attempt to evaluate whether the lip motion can be used as an additional cue to improve the performance of the systems designed to recognize the sign language. The identification of the words based on the lip movement is a difficult task and to solve this problem we developed a method to cluster the words using a three-template model that is employed to capture the distribution of these states in the image sequence. To evaluate this approach we create a dictionary-based database that consists of a number of image sequences associated with different words.

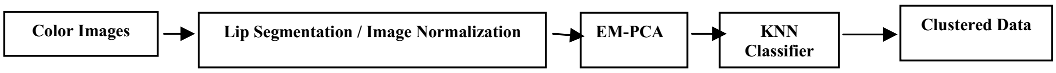
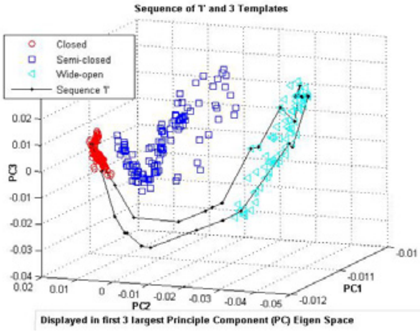

Color Images → Lip Segmentation / Image Normalization → EM-PCA → KNN Classifier → Clustered Data

Fig. 1. Overview of the lip reading system.

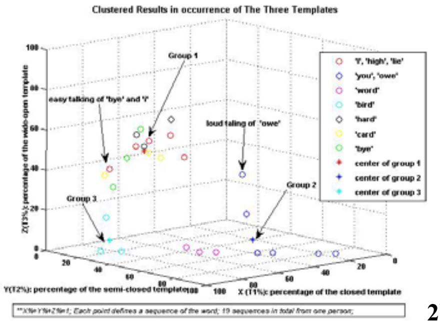
2. Work in Progress

The first component of the system performs lip segmentation by analyzing the hue component of the color image. To cluster the words with similar attributes we evaluate the image sequence based on three fundamental templates: **lips closed (T1)**, **semi-closed (T2)** and **wide open (T3)** (fig 2). We employed the **Expectation-Maximization (**EM) **PCA** approach to identify the distribution of the three states in the image sequences that define the words to be analyzed. Based on the distribution (percentage) of the three models in the image sequence the system groups the words in different clusters.

3. Experimental Results



2a



2b

Figure 2a: **3 Templates Model with one sequence of 'I' (black lines)**
The data used to construct the three templates is manually generated.
Figure 2b: **Cluster Results.** (***X***-Axis: T1; ***Y***-Axis: T2; ***Z*** -Axis: T3)

The KNN is applied to classify each image frame (see the black points in Figure 2a) with respect to the three templates. Some frames are not classified to any template, so those frames are assigned as Not Classified Frames (**NCF**). Based on the distribution of the three templates in the image sequence, we can group the words in different clusters as illustrated in Figure 2b*. Each point in Figure 2b defines the plot of the words contained in the database based on the occurrence of the three templates in the image sequence.

Table 1: Training Results and Classification Error of Templates Model based Clustering

Group Words	T1	T2	T3	Classification Error
Group 1: 'I', 'High', 'Lie', 'Bye', 'Hard', 'Card'	41-65%	5-10%	35-47%	17%
Group 2: 'You', 'Owe', 'Word', 'Old'	15-38%	51-85%	0-30%	33%
Group 3: 'Bird'	75-84%	11-22%	0-5%	0%

Over 7500 frames of 47 sequences from 3 different people are used for training and testing. (*: T1+T2+T3+NCF=100%)

Reference:

[1] Roweis S. (1998), "EM Algorithms for PCA and SPCA", Advances in Neural Information Processing Systems, 10: 626-632.

[2] Eveno N., A.Caplier A. and Coulon P.(2004), "Accurate and Quasi-Automatic Lip Tracking", *IEEE Trans. Circuits Syst. Video Techn.* 14(5): 706-715.

Feature Detection in Range Images

S. Suganthan, S.A. Coleman
School of Computing and Intelligent Systems, University of Ulster, Northland Road, Londonderry
s.suganthan@ulster.ac.uk, sa.coleman@ulster.ac.uk

Abstract

We present a general approach to the development of image processing gradient operators that can be applied directly to range data. This approach is demonstrated for first order derivative operators.

Keywords: Feature detection; Range images; Gradient operators

1 Introduction

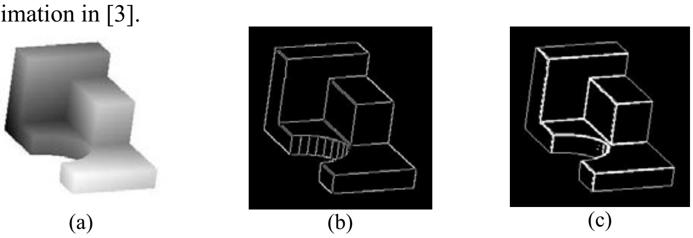
In recent years computer vision applications have increasingly began to use range image data instead of, or conjunction with, intensity image data. This is largely because range imagery can be used to obtain reliable descriptions of 3-D scenes [1]. Due to the locational irregularity of range image data, multiscale feature detection on range images is a significantly different problem than that on intensity images. A number of feature extraction algorithms for use on range data have used a scan line approach which tends to be time consuming and hence inappropriate for real-time image processing. In recent work, scalable and adaptive first and second order derivative operators have been developed via a finite element (FE) framework for use on intensity images; such operators have been proven to perform successfully when compared with well-known intensity image feature detection operators [2]. This research extends the work in [2] by developing FE based gradient operators for use on irregular quadrilateral meshes that can be used directly on range image data with pre-processing with the ultimate aim of achieving real-time range data processing for robotic vision.

2 Work in progress

We have initially focused attention on the design and implementation of irregular 3×3 first order derivative operators for direct use on range images. The first order gradient operators correspond to weak forms of operators in the finite element method and can be defined as $E_i^\sigma(U)=\int_\Omega \underline{b}_i\cdot\underline{\nabla}U\psi_i^\sigma d\Omega$. Here, ψ_i^σ is a Gaussian test functions, embracing a scale parameter, σ, enabling the operator to be readily scaled across the image plane. On implementing the 3×3 gradient operator is was found that standard thresholding that would be applied to an intensity image feature map was not appropriate for a range image edge map, in fact it led to detecting surfaces rather than edges. Hence rather than specifying a threshold above which all feature points were selected, we had to determine significant changes in the operator responses which correspond to edges in a range image.

3 Results

We present results using a range image from [4] as illustrated in Figure 1(a). Figure 1(b) shows the feature map obtained using our finite element based approach and Figure 1(c) shows the feature map obtained using the scan line approximation in [3].



(a) (b) (c)

Figure 1. Range data feature maps (a) Original image; (b) 3×3 FE operator; (c) Scan line approximation [3]

Acknowledgements

This work was funded by EPSRC under the research grant number EP/C006283/1.

References

[1] Besl, P.J., “Active, optical range imaging sensors”, *Machine Vision and Apps,* Vol.1, pp127-152, 1988

[2] Coleman, S.A., “Scalable Operators for Adaptive Processing of Digital Images” *PhD thesis,* University of Ulster, 2003

[3] Jiang, X., et al., “Edge Detection in Range Images Based on Scan line Approximation”, Computer Vision and Image Understanding, Vol.73 ,No.2, February, pp183-199,1999

[4] http://sampl.ece.ohio-state.edu/data/3DDB/RID/index.htm

Shape Classification of Colorectal Polyps at CT Colonography using Support Vector Machines

Abhilash A. Miranda, Tarik A. Chowdhury, Ovidiu Ghita, Paul F. Whelan
Vision Systems Group, Dublin City University, Ireland

Abstract

We present a novel colorectal polyp shape classification scheme for Computer Aided Diagnosis at CT Colonography (CAD-CTC) that attains a very low false positive (FP) rate with a high sensitivity for clinically significant polyp sizes. The technique applies a simple and reliable orientation-invariant feature-set obtained from candidate shapes to an SVM classifier trained using phantom polyps.

Introduction

Studies have shown that there could be a considerable reduction in the number of over half-million fatalities per year caused due to colorectal cancer if growths in the colon called polyps are removed before they become cancerous. The aim of the CAD-CTC is to perform fast and early detection of colorectal polyps by classifying complex colorectal shapes with high sensitivity and low False Positives (FP) rate. Many researchers[1][2] have addressed the problem of colorectal shape classification based on morphological features of the polyps with varying degrees of performance.

Work in Progress

The focus of the presented CAD-CTC is in classification stage of the candidate convex colonic surfaces obtained as a result of an efficient colon segmentation and surface extraction process[2]. From each candidate surface, we extract surface features using an orientation-invariant shape distribution function[3] (SDF) and the Gaussian distribution of the surface voxels[1][2]. The histogram of the distances of the candidate surface voxels from the Gaussian center[2] is chosen as the SDF. We use the feature-set $\{f_{dB}, d_G\}$, where f_{dB} is the -9dB attenuation frequency of the power spectral density of the SDF and d_G is the sum of the weighted Gaussian distances normalized by the number of surface voxels constituting the candidate shape.

Using medium and large sized polyps from a phantom as a training set $\mathcal{X} = \mathcal{X}_p$, we find that no linear decision boundary can classify polyps from non-polyps in the feature-space. In addition, we observe that feature-space is very sparse and $\mathcal{X}_p$ requires further population. We tested a polynomial kernel SVM classifier for varying degrees k and artificial populations $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$ on 63 real patient datasets to ascertain the suitability of a nonlinear decision boundary.

Results

		SVM1	*SVM2*	*SVM3*	*SVM4*
k		3	3	5	5
$\mathcal{X}$		$\mathcal{X}_p$	$\mathcal{X}_{SVM2} = \mathcal{X}_p+\mathcal{X}_1$	$\mathcal{X}_{SVM3} = \mathcal{X}_{SVM2}+\mathcal{X}_2$	$\mathcal{X}_{SVM3}+\mathcal{X}_3$
Sensitivity (%)	⩾10 mm	90	90	90	90
	[5,10) mm	72	78	75	81
	< 5mm	60	63	62	60
Total Sensitivity ⩾5 mm (%)		76	81	79	83
FP per dataset		2.54	5.78	4.42	6.45

Acknowledgments

We acknowledge the contributions of Dr. Helen Fenlon and Dr. Padraic MacMathuna of the Mater Hospital, Dublin. This work was supported under an Investigator Programme Grant by Science Foundation Ireland.

References

[1] Kiss, G., Cleynenbreugel, J., Thomeer, M., Suetens, P., Marchal, G., "Computer Aided Diagnosis for Virtual Colonography", *MICCAI* (2001) 621-628

[2] Chowdhury, T A., Ghita O., Whelan, P. F., Miranda, A. A., "A Note on Feature Selection for Polyp Detection in CT Colonography", *ICPR* (2006)

[3] Miranda, A. A., Chowdhury, T. A., Ghita, O., Whelan P.F., "Shape Filtering for False Positive Reduction at Computed Tomography Colonography", *MICCAI* (2006)

Facial Expression Classification using LLE and SVMs

Jane Reilly, John Ghent, John McDonald.
National University of Ireland, Maynooth
jreilly@cs.nuim.ie, jghent@cs.nuim.ie, johnmcd@cs.nuim.ie

Abstract

In this paper we present results of applying LLE to facial expression classification. This technique can accurately classify and differentiate between subtle and similar expressions, involving the lower face.

Keywords: Locally Linear Embedding, Facial Expression.

1 Introduction

Since the importance of facial expressions was first established in 1872 [1], many studies have been carried out attempting to interpret their meaning. In 1978, the *Facial Action Coding System* (FACS) was created by Ekman and Friesen and is the most comprehensive method for describing facial movement [2]. The FACS provides an unambiguous quantitative means of describing all movements of the face in terms of 46 *Action Units* (AU's). Although the FACS provides a good foundation for the coding of face images by human observers, the automatic recognition of AUs by computers remains a difficult challenge. Many different techniques have been applied to the problems of automatic FACS classification – see [3] for an overview of the current techniques.

2 Work in progress

To date our group has proposed a computational model for the classification of facial expression. This model, which is based on *Principal Component Analysis* (PCA), can accurately classify extreme expression changes, but as PCA is a linear technique, it can't accurately classify subtle changes in appearance [4]. In this paper we apply *Locally Linear Embedding* (LLE) to classify subtle changes in expression. LLE is a non-linear dimensionality reduction technique that computes low-dimensional neighborhood preserving embeddings of high-dimensional data be unfolding the underlying manifold [5].

3 Results

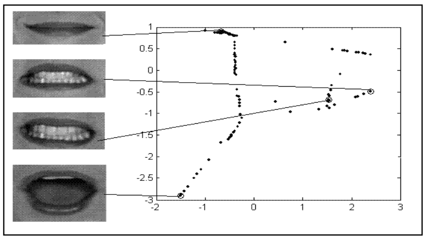


Figure 1: The AUs portrayed from top to bottom are, AU12, AU20+AU25, AU10+AU20+AU25, AU25+AU27.

Table 1: In the table above , A = AU12, B = AU20+AU25, C = AU10+AU20+AU25, D = AU25+AU27.
E is the soft margin, ***σ*** is the kernel parameter, ***Ts*** is the percentage accuracy

AU	*E*	σ	Ts
A-v-all	8.0	0.5	98.7
B-v-all	6.0	0.3	83.8
C-v-all	7.0	0.8	98.0
D-v-all	6.1	0.1	84.6
Total			**91.3**

In this experiment four lower facial expressions are classified (see figure 1). A LLE shape space was developed using 141 images of a subject performing the four expressions at varying levels of intensity. Test images of various subjects portraying these expressions at varying degrees of intensity are then projected into the LLE space, these outputs are then used as inputs to the SVM classifier. As there are four expressions to be separated, 4 SVM classifiers are used. As can be seen from Table 1, this technique accurately classifies similar and subtle expressions with an average accuracy of 91.3%. This improves on our previous work as our based PCA approach could only classify extreme appearance changes [4].

Acknowledgements: This publication has emanated from research conducted with the financial support of Science Foundation Ireland.

References

[1] Darwin, C., Ekman, P.: (1889 - 1998) *The Expression of the emotions in man and animals.* The University of Chicago Press.
[2] Ekman P et al. (1978) Facial Action Coding System. Consulting Psychologists Press.
[3] Fasel B. et al. (2003) Automatic facial expression analysis : A Survey. Pattern Recognition, Vol 36(1) (2003) 259-275
[4] Ghent J. (2005) A computational Model of Facial Expression. PhD thesis, NUI, Maynooth.
[5] Saul L.K. Roweis S.T. (2002) Think Globally Fit Locally unsupervised learning of low dimensional manifolds, Journal of Machine Learning Research.

A Reader Training System for CT Colonography

V. Luauté, R.J.T. Sadleir, H.M. Fenlon and P.F. Whelan
Vision Systems Group, School of Electronic Engineering, Dublin City University.
luautev@eeng.dcu.ie

Abstract

Radiologists using computed tomography colonography (CTC) need appropriate training in order to become proficient in this technique. We have developed a remote access, web based, training system that allows the user to flag polyp candidates in a range of CTC datasets. Upon completion of their work, the trainee can run an automatic evaluation and monitor their progress in real time. This remote access system allows a trainee radiologist to gain proficiency in an emerging colorectal cancer screening technique over the Internet.

Keywords: Remote imaging, colonography, colon cancer, Java Servlets, Java Applets.

1 Introduction

Introduced by Vining et al. in 1994 [1], computed tomography colonography (CTC) is demonstrated to have similar sensitivity to conventional colonoscopy (CC) for the detection of significant colorectal polyps [2]. However, CTC is not yet in widespread use and the literature highlights the lack of suitably trained radiologists [3] as a possible reason for this. To address this problem, we have developed a novel remote access system to train radiologists for colorectal cancer screening using CTC. Accessibility is another key issue for this project. To ensure radiologists gain the relevant experience without any computer-related issues, the system should be operating system (OS) independent and operate without the need for specific software packages. This can be achieved by using client-server architecture over the Internet that is implemented using Java.

2 Methods

To grant the fastest response time via the interface, the CT dataset is stored on the client's hard disk (using a signed Applet). To reduce the transfer time, we have developed a loss-less compression algorithm. The trainee is requested to highlight polyp candidates by flagging locations via circles superimposed on 2-D axial images. Also, the trainee is required to define the size (in mm) and the type of the polyp e.g. "sessile" or "pedunculated". The interface provides the appropriate tools that are required to facilitate the identification and measurement of polyps. These features include windowing, polyp measurement and zooming. The navigation through the 2-D slices of a dataset can be performed by dragging a slider or by the rotation of the mouse wheel. In addition to axial images, the user can display coronal and sagittal reformat views of a dataset. A 3-D view can also be taken into consideration. A volume rendering technique called "ray-casting" has been implemented in the system. Each user has an account in order to allow monitoring of their training. They can also run an automatic evaluation of their work based on gold standard information previously gathered from specialists [4].

3 Results

Our compression algorithm insures a 50% lossless compression rate. Transfer time of a 150MB CTC dataset from the server to the client is approximately 40 seconds over a local network. The viewer Applet gives the user the ability to start working on the initial images while the rest of the dataset is being transferred. Once the dataset has been received, the client's interface is able to offer a fast response time. An iteration of lumen tracking using the system takes approximately 45 seconds. The use of our system can also help to determine the learning curve associated with interpreting CTC findings [5].

References

[1] D.J. Vining, D.W. Gelfand, R.E. Bechtold et al. (1994) "Technical Feasibility of Colon Imaging with Helical CT and Virtual Reality" American Journal of Roentgenology 162(Suppl):104.

[2] P.J. Pickhardt, J.R. Choi, I. Hwang et al. (2003) "Computed tomographic virtual colonoscopy to screen for colorectal neoplasia in asymptomatic adults" New England Journal of Medicine 349(23):2191-2200.

[3] P.B. Cotton, V.L. Durkalski, B.C. Pineau et al. (2004) "Computer tomographic colonography (virtual colonoscopy): A multicenter comparison with standard colonoscopy for detection of colorectal neoplasia" The Journal of the American Medical Association 291(14)1713-1719.

[4] V. Luauté, R.J.T. Sadleir and P.F. Whelan (2006) "An automatic evaluation strategy for a remote access CT colonogaphy training system" Biosignal 2006 - The 18th biennial International EURASIP Conference , Brno, Czech Republic, June 28th to 30th.

[5] Jorge A. Soto, MD, Matthew A. Barish, MD and Judy Yee, MD (2005) "Reader Training in CT Colonography: How Much Is Enough ?" Radiology 2005;237:26-27.

Active Surface Meshes for Intuitive 3-D Computer Graphics Shape Deformation and Modelling

Patricia Moore and Derek Molloy
Vision Systems Group
School of Electronic Engineering, Dublin City University, Dublin 9.
patricia.moore@eeng.dcu.ie

Abstract

This paper presents ongoing research into the generation of a novel approach to 3-D computer graphics modelling that will allow intuitive deformation of models in 3-D Space. Currently being investigated is a reformulation of Active Meshes for use in the 3-D modelling domain.

Keywords: Modelling, Deformation, Virtual Sculpting, 3-D Shape.

1 Introduction

The modelling of 3-D objects is an important and challenging problem that has long been a fundamental part of computer graphics. Although computer graphics and computer aided design tools have evolved rapidly in the past few years, 3-D graphics designers still rely on non-intuitive modelling procedures to create 3-D objects with complex freeform shapes. Indeed, traditional methods of generating and deforming 3-D models often require skilled labour and large time investments on the part of the designer, as current surface and solid modelling tools often require the user to manually manipulate numerous control points via a keyboard/mouse while monitoring modifications on a 2-D visual display. A novel and more intuitive approach would not only greatly reduce the time investment made by designers, but would also open up the domain of 3-D computer graphics design to the lay person.

2 Work in progress

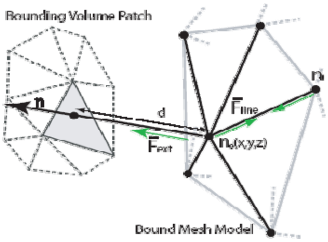


Fig 1: Active Mesh Deformation

Active Contour Models (ACMs) are energy based tools, used in Machine Vision to extract image features. In work carried out by the Vision Systems Group in Dublin City University, ACMs have been reformulated as active-meshes [3] which were initially applied to the problem of tracking motion in 2-D images. However, it has been demonstrated that this approach can be extended to allow 3-D human models to be fitted with an underlying generic model, containing skeletal and animation information, by converting the model to a 3-D active mesh [1]. Figure 1 illustrates this approach. Work is currently being carried out to develop Active Surface Meshes with various constraints to control and manipulate their response to applied forces.

3 Results

Introduced by Kass et al [2] for the 2-D case, as explicit deformable contours, and generalised to the 3-D case by Terzopoulos et al [4], deformable models have provided an extensible framework for the construction of virtual 3-D objects. As many different 3-D reconstruction problems have different requirements, various modelling approaches have been proposed. A review of these approaches was carried out in order to determine the state of the art and to assess the viability of this project. Overall, ACMs and Active Meshes seem to offer the most suitable approach. Although classed as non physics based, these models are physically motivated and deform under applied forces while preserving their structure. These models offer the geometric and computational efficiency of non physics based models, while incorporating some of the realism achieved using those that are physics based. This research will extend the application of Active Meshes to the 3-D modelling realm, by reformulating them for generic use in 3-D modelling applications.

Acknowledgements

I would like to thank the Irish Research Council for Science, Engineering and Technology for the financial support that has made this research possible.

References

[1] Boyle, E. (2006). Active Modelling of Virtual Humans. PhD thesis, Dublin City University.

[2] Kass, M., Watkin, A., Terzopoulos, D. (1987) Snakes: Active Contour Models. *International Journal of Computer Vision 1*, 4, 231-331.

[3] Molloy, D. (2000). Active Meshes for Motion Tracking. PhD thesis, Dublin City University.

[4] Terzopoulos, D., Witkin, A., Kass, M. (1988) Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion. *Artificial Intelligence,* 36:91-123.

Controlling Character Animation with Hand Gestures

J.V. Condell[a], **G. Moore**[b]

[a]SCIS, Faculty of Engineering, University of Ulster at Magee, Northland Road, Londonderry

[a]SCM, Faculty of Engineering, University of Ulster at Jordanstown, Shore Road, Newtownabbey

Abstract

This abstract describes current research carried out at UU in collaboration with an animation studio to employ computer vision techniques to develop a prototype camera based desktop system and associated animation process. This will allow an animator to control 3D character animation through the capture and interpretation of hand gestures.

Keywords: Animation, Hand tracking and analysis, Motion capture.

1 Introduction

As the demand and range of outlets for 3D computer animation grows, animators need tools to help them meet this rise in demand. This is especially true for small animation studios which require short production cycles yet need to maintain high quality output to ensure their products do not look out of place when juxtaposed to productions from larger studios. While the initial goal of this research is focused on a literal translation of hand gestures to hand models, the eventual goal of the project accommodates a more generally applicable gesture grammar. This would support existing common practice with studios by providing a softer, more intuitive user interface for the animator that improves the productivity of the animation workflow and the quality of the resulting animations, helping studios to compete better in an increasingly competitive industry.

2 Work in progress

Internally the system design has two key software components: a *real-time vision capture system* and an *animation control system* that is embedded into the target 3D animation software. The *vision system* will make a literal interpretation of the hand movements. While this may appear unambitious it serves the purpose of acting as a proof of concept. Animators agree that such a tool which did not need intrusive gloves or markers would improve on the current system. The vision system is responsible for capturing, processing and translating a video feed of a hand in order to generate a gesture data feed describing hand movements to the animation control system. The capture software will employ a recently developed optical flow technique [1] to track finger and hand movements. The *animation control system* applies the hand gesture data to the 3D model in order to produce an animated sequence. Most of the interaction and processing extraneous to interacting with and controlling the scene hierarchy will be passed to the plugin in the form of coordinate and transform data. Likewise, the plugin will, as far as possible, be independent of the modeling and animation processes. This latter point has the added benefit that models to be animated will not require any special setup, e.g. custom rigging, in order for the plugin to animate them. This is key to realizing the goal of providing an integrated yet transparent tool.

3 Results

The software infrastructure that provides a platform within which to process images has been completed, which will ease further experimentation and development of the computer vision algorithms required to complete this subsystem. Initial data import has been successful. Work has begun implementing a preprocessing stage to prepare the video data for processing by the motion estimation software. The adaptation of existing research on the motion tracking system is ongoing. It is anticipated that this is where the projects major contribution to computer vision research will reside. The software infrastructure, in the form of a utility plugin for 3DS Max is nearing completion. Once this is in place research into how to make the approach as independent as possible from the target platform will be possible. Work has already been completed on coding a MaxScript utility for 3DS Max that allows data typical of that which will be generated to control a 3D model in 3DS Max. Imported data is being applied to the scene hierarchy.

While the project is still at an early stage it is felt that a working solution is fully realizable and that the innovation required in how the real-time vision system will process the video feed will provide a valuable contribution to computer vision research.

Acknowledgements

The authors wish to thank Banjax Studios and The Midas Initiative for support and funding for this project.

References

[1] Condell, J.V., Scotney, B.W., Morrow, P.J. (2005) Adaptive Grid Refinement Procedures for Efficient Optical Flow Computation. *International Journal of Computer Vision*, 61:1, 31-54.

Automated Image Similarity and 3-D Visualisation of Large Digital Content Sets

Dave Barry, Andrew Donnellan, Derek Molloy[1]
Department of Electronic Engineering, ITT Dublin
Tallaght, Dublin 24, Ireland.
[1] School of Electronic Engineering, Dublin City University.
Andrew.Donnellan@ittdublin.ie, Derek.Molloy@dcu.ie

Abstract

There have been many developments in the area of image retrieval systems, from textual based to automatic image classification and retrieval. This poster presents the beginnings of a scheme for the automated analysis and retrieval of home everyday digital content utilizing an immersive 3-D environment.

Keywords: Image Retrieval

1. Introduction

In the past number of years the usage and archives of digital media being generated by the home user is larger than ever before as a result of the cost of digital camera and camcorder technology becoming more affordable The digital nature of this content allows individuals to share personal image and video content with friends, relatives and colleagues thus compounding the management and visualization problems, such as locating a particular image in a very large collection. This project will aim to develop a similarity measure which will enable the automated analysis and categorization of the database content. The incorporation of a 3-D visualisation retrieval environment will allow the user to add significant value to the collection by incorporating human visual abilities with the automated organization of the content.

Creating a system which incorporates an automatic analysis measure removes the use of annotation. Annotation of images can be a time consuming arduous task which will leave images described inadequately based on the subjective nature of the images.

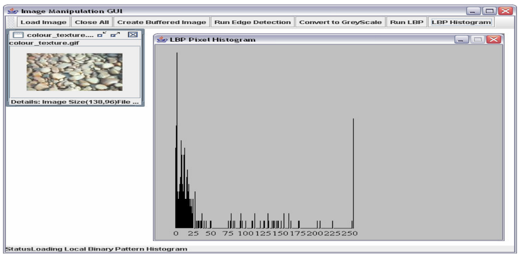
With an ever increasing interest in to image retrieval, systems are now relying on content rather then on textual descriptions. Examples of such systems are QBIC System [1], Four Eyes System [2], NeTra System [3] and CANDID System [4].

2. Work in progress

Thus far an investigation into previous Content Based Image Retrieval (CBIR) systems has been carried out. Emphasis was placed on the methods used within these systems for texture description and colour segmentation which lead to a need for better understanding of co – occurrence matrices, Gabor filters, Gaussian – Markov Random Fields, Local Binary Patterns and histograms.

A system for implementing in part some of these techniques has been developed and a sample output can be seen in the results section below.

3. Results



Details: Image Size(138,96)File ...

StatusLoading Local Binary Pattern Histogram

Figure 1: LBP Histogram

References

[1] M. Flickner, et al., Query by image and video content: the QBIC system IEEE Comput. 28 (1995) 23 – 32.

[2] T. Minka, An image database browser that learns from user interaction, Technical Report, MIT Media Lab Perceptual Computing Section, 1995

[3] T. Minka, R. Picard, Interactive learning with a society of models, Pattern Recognition 30 (1997) 565 – 582.

[4] P.M. Kelly et al., Query by image example: the CANDID approach, Proceedings of the SPIE Storage and Retrieval of Image and Video Databases, 1995, pp. 238 – 248.

Remote Access CT Colonography using a Java Enabled Mobile Device

Julien Le Colloec, Robert J. T. Sadleir and Paul F. Whelan
Vision Systems Group, School of Electronic Engineering, Dublin City University, Ireland.
julien.lecolloec2@mail.dcu.ie

Abstract

We present a novel remote access medical imaging system that has been developed for use with Java enabled mobile devices. The system allows a remote user to examine computed tomography colonography (CTC) data using various 2-D and 3-D visualisation techniques. A data reduction scheme incorporating segmentation, cropping and compression is utilised to reduce the amount of data sent from the server to the client. All aspects of the system are implemented using the Java programming language from Sun Microsystems.

Keywords: Remote access medical imaging, Segmentation, Computed tomography colonography

1. Introduction

The use of mobile devices for general applications in remote access medical imaging has been investigated [1, 2]. The aim of this research is to investigate the use of mobile devices for a specific examination. This examination, computed tomography colonography (CTC), is used for colorectal cancer screening and involves screening the large intestine for polypoid growths which are precursors to colorectal cancer.

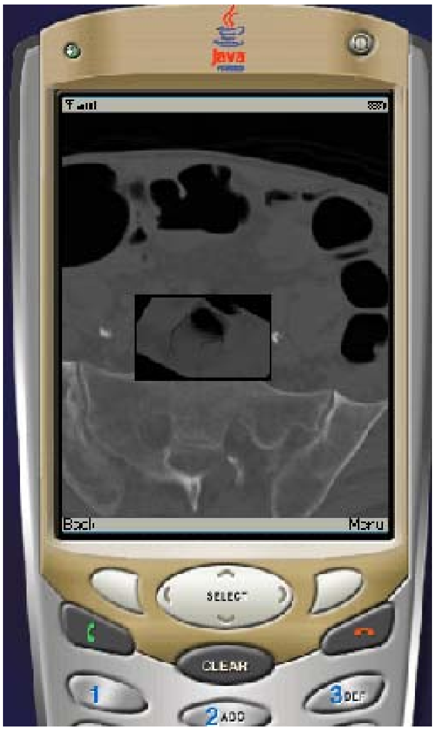


Figure 1: A 3-D volume rendering of a section of the large intestine displayed using the Java wireless toolkit emulation environment.

2. Work in progress

The client software is implemented using the J2ME wireless toolkit. The user can interact with a CTC data set using a custom interface. The interface enables the user to examine the individual slices of the data set. In addition, it is possible to generate a 3-D volume rendering of a user defined region of the data set (see Figure 1). This feature is useful for discriminating between polyps and naturally occurring colonic features. The client stores a total of 10 successive slices in a buffer to facilitate localised volume rendering and to minimise the effects of the latency associated with the network connection. The amount of data sent between the server and the client is reduced by segmenting, cropping and compressing the original CTC data sets.

3. Results

Tests have been carried out using a Dell Inspiron 2200 with 512 RAM and a 1.4 Ghz Celeron processor. It takes 108 seconds to read an entire CTC dataset and a further 18 seconds to complete the region growing based segmentation stage. The time required for the transmission of 10 images to the client is 14.047 seconds. Sequencing between successive images requires 0.344 seconds. The time required to generate the 3-D volume rendered image is 2.231 seconds and full rotation of the resulting volume requires 2.231 seconds.

References

[1] Flanders, A.E., Wiggins III, R.H., Gozum, M.E. (2003) “Handheld Computers in Radiology”, RadioGraphics, 23 (4) 1035-1047.
[2] McGonigle, E. (2006) “Medical Imaging using a Java Enabled Mobile Phone” Masters thesis, Dublin City University.

Motion Distillation

M. Sugrue and E. R. Davies
Machine Vision and Signal Processing
Royal Holloway, University of London
TW20 0EX, UK
{m.sugrue, e.r.davies}@rhul.ac.uk

Abstract

Here we present a tracking paradigm modelled on the dual channel form-motion architecture of the human visual system. The scheme assumes no prior knowledge of the objects to be tracked or any characteristics of the scene while achieving integration of object detection and tracking steps. Our research focus is on the difficult case of pedestrian tracking in uncontrolled outdoor environments. Results show the scheme to be highly robust, as well as extremely computationally efficient.

Keywords: Spatio-temporal, Motion detection and Tracking

1 Introduction

One of the most important aspects of pedestrian tracking is that, unlike the rigid translational movements of traffic monitoring, pedestrians actually change their shape and appearance *in order to move*. Thus any tracking scheme which is dependant on frame-to-frame appearance matching is fundamentally weak. Neuroscience tells us that the human brain behaves more subtly, utilising a dual channel form-motion tracking scheme.

2 Work in progress

This work aims to achieve fast and robust results using a dual channel architecture, similar to the Human Brain. A dedicated motion channel is calculated using spatio-temporal Haar wavelet decomposition, while the appearance of the target is recorded in the form channel and used to resolve ambiguities and detect when targets have become stationary or have left the scene.

The filters we use result in a wealth of sub-target level speed information can be derived, allowing discrimination of slow and fast moving regions of the target, a result we have termed *Motion Distillation*. Our current work is focused on utilising this non-binary segmentation information for the task of automatic behaviour recognition in CCTV video.

3 Results

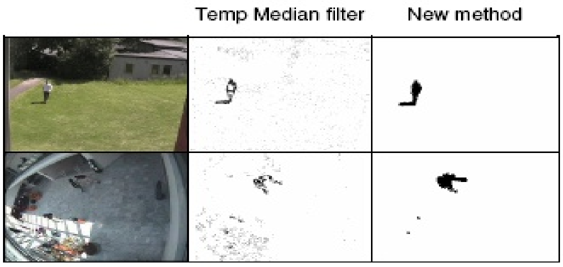
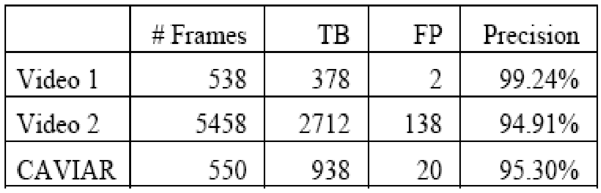


Fig 1. (Top) Pedestrian walking slowly towards the camera. (Bottom) Fight sequence from CAVIAR database.

Table 1. Performance of motion channel with three videos



	# Frames	TB	FP	Precision
Video 1	538	378	2	99.24%
Video 2	5458	2712	138	94.91%
CAVIAR	550	938	20	95.30%

Tracking results show Motion Distillation to be computationally extremely cheap. A maximum of 1.14 operations per pixel can provide for motion channel calculation (compared with ~256 for temporal median filtering, for example), translating to 62 fps on a P4 processor. The segmentation results are also more robust, as can be seen in the comparisons with temporal median filtering in figure 1, and the motion detection results show promisingly low false positive rate (FP – see table 1) and a near zero false negative rate (not shown)

References

[1] M. Sugrue, E.R. Davies, "Tracking in CCTV Video Using Human Visual System Inspired Algorithm", Visual Information Engineering 2005, University of Glasgow, 4–6 April 2005. pp.417–423

[2] M. Sugrue, E. R. Davies, “Motion Distillation for Pedestrian Surveillance”, Visual Surveillance 2006, pp 105-112

Modeling the Dynamics of Facial Expression using LLE

Jane Reilly, John Ghent, John McDonald.
National University of Ireland, Maynooth
jreilly@cs.nuim.ie, jghent@cs.nuim.ie, johnmcd@cs.nuim.ie

Abstract

In this paper we outline results from ongoing research into modeling the dynamics of facial expression. The dynamics of facial expression can be described as the intensity and timing of a facial expression as it forms.

Keywords: Locally Linear Embedding, Facial Expression Dynamics.

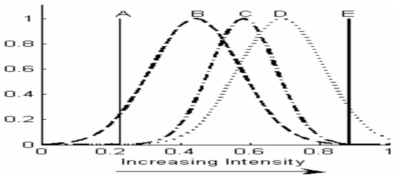
1 Introduction

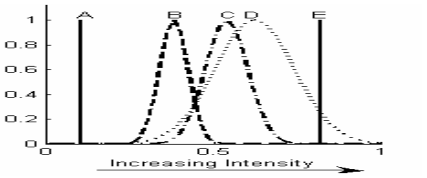
Researchers have been studying facial expressions since their importance was first established in 1872 [1]. During the subsequent years various different techniques were developed that attempt to accurately classify facial expressions. The Facial Action Coding System (FACS), created in 1978 is the most comprehensive of these and is widely used in research [2]. The FACS provides an unambiguous means of quantitatively describing all movements of the face in terms of Action units (AU's). While the FACS provides a foundation for the analysis of facial expression, according to Ambadar et al. only a few investigators have examined the impact of dynamics in deciphering faces and these studies were largely unsuccessful. Results of the research carried out by Ambadar et al. found that facial expressions are frequently subtle, and that subtle expressions that were not identifiable in static presentations suddenly became apparent in dynamic display [3].

2 Work in progress

To date our group is working on a technique to classify subtle and similar facial expressions as they form. In an extension of this technique we estimate the dynamics of facial expression formation in terms of intensity and timing. To achieve this we use *Locally Linear Embedding* (LLE). LLE was originally proposed in 2001 as a non-linear dimensionality reduction technique that computes low-dimensional neighborhood preserving embeddings of high-dimensional data be unfolding the underlying manifold [4]. We show that this non-linear dimensionality reduction technique provides a means for the analysis of the dynamics of facial expression.

3 Results





Figures 1 & 2: LLE intensity coding results
Left: single AU intensity coding using AU25
Right: composite AU intensity coding using AU20+AU25

In this experiment information about the dynamics of facial expression is extracted. The intensity over time of a composite AU lower facial expression and a single AU lower facial expression are estimated according to the FACS intensity coding [2]. The FACS intensity coding ranges from A to E, with A representing a minor change in appearance, and E an extreme change in appearance. For AU20+AU25 a total of 10 subjects (60 frames) were landmarked and for AU25 a total of 24 subjects (144 frames) were landmarked. After preprocessing, LLE was applied to the datasets, reducing the dimensionality to one dimension. Gaussians were fitted to the samples for intensities B, C and D, while for A and E, the mean is calculated. As can be seen from Figures 1 and 2, our technique accurately characterizes the intensity of the facial expressions over time and hence provides a means for modeling their dynamics.

Acknowledgements: This publication has emanated from research conducted with the financial support of Science Foundation Ireland.

References

[1] Darwin, C., Ekman, P.: (1889 - 1998) *The Expression of the emotions in man and animals.* The University of Chicago Press.

[2] Ekman P et al. (1978) Facial Action Coding System. Consulting Psychologists Press.

[3] Ambadar et al. (2005) Deciphering the enigmatic face: The importance of facial dynamics to interpreting subtle facial expressions. Psychological Science.

[4] Saul L.K. Roweis S.T. (2002) Think Globally Fit Locally unsupervised learning of low dimensional manifolds, Journal of Machine Learning Research.

Notes

The Irish Machine Vision and Image Processing (IMVIP) Conference 2006 took place from the 30th of August to the 1st of September. IMVIP is the main conference of the Irish Pattern Recognition and Classification Society (IPRCS), a member body of the International Association for Pattern Recognition (IAPR).The 2006 Irish Machine Vision and Image Processing conference was hosted by the Vision Systems Group, RINCE, Dublin City University. It brought together theoreticians and practitioners, industrialists and academics, from the numerous related disciplines involved in the processing and analysis of image-based information. IMVIP is a single-track conference consisting of high quality previously unpublished contributed papers. The conference emphasises both theoretical research results and practical engineering experience in all areas. Full papers were subject to a double-blind review process by the Programme Committee. In addition to the contributed programme, there were also talks by invited speakers.

http://www.rince.ie/imvip2006/

ISBN: 0-9553885-0-3

Sponsored By: DCU RINCE | Research Institute for Networks and Communications Engineering Vision Systems Group