ADAPTIVE OBJECT SEGMENTATION AND TRACKING

by

Nagachetan Bangalore Manjunathamurthy

SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
AT THE UNIVERSITY OF SUSSEX

School of Engineering and Design

University of Sussex

Brighton

January 2012

Declaration

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree.

Signature

Nagachetan Bangalore Manjunathamurthy
Dated: 12 January 2012

ADAPTIVE OBJECT SEGMENTATION AND TRACKING

Summary

Efficient tracking of deformable objects moving with variable velocities is an important current research problem. In this thesis a robust tracking model is proposed for the automatic detection, recognition and tracking of target objects which are subject to variable orientations and velocities and are viewed under variable ambient lighting conditions. The tracking model can be applied to efficiently track fast moving vehicles and other objects in various complex scenarios. The tracking model is evaluated on both colour visible band and infra-red band video sequences acquired from the air by the Sussex police helicopter and other collaborators. The observations made validate the improved performance of the model over existing methods.

The thesis is divided in three major sections. The first section details the development of an enhanced active contour for object segmentation. The second section describes an implementation of a global active contour orientation model. The third section describes the tracking model and assesses it performance on the aerial video sequences.

In the first part of the thesis an enhanced active contour snake model using the difference of Gaussian (DoG) filter is reported and discussed in detail. An acquisition method based on the enhanced active contour method developed that can assist the proposed tracking system is tested. The active contour model is further enhanced by the use of a disambiguation framework designed to assist multiple object segmentation which is used to demonstrate that the enhanced active contour model can be used for robust multiple object segmentation and tracking. The active contour model developed not only facilitates the efficient update of the tracking filter but also decreases the latency involved in tracking targets in real-time. As far as computational effort is concerned, the active contour model presented improves the computational cost by 85% compared to existing active contour models.

The second part of the thesis introduces the global active contour orientation (GACO) technique for statistical measurement of contoured object orientation. It is an overall object orientation measurement method which uses the proposed active contour model along with statistical measurement techniques. The use of the GACO technique, incorporating the active contour model, to measure object orientation angle is discussed in detail. A real-time door surveillance application based on the GACO technique is developed and evaluated on the *i-LIDS* door surveillance dataset provided by the UK Home Office. The performance results demonstrate the use of GACO to evaluate the door surveillance dataset gives a success rate of 92%.

Finally, a combined approach involving the proposed active contour model and an optimal trade-off maximum average correlation height (OT-MACH) filter for tracking is presented. The implementation of methods for controlling the area of support of the OT-MACH filter is discussed in detail. The proposed active contour method as the area of support for the OT-MACH filter is shown to significantly improve the performance of the OT-MACH filter's ability to track vehicles moving within highly cluttered visible and infra-red band video sequences.

Acknowledgements

With the deepest gratitude I wish to thank every person who has come into my life and inspired, touched and illuminated me through their presence. I would like to express my gratitude to all who consciously or unconsciously have made contributions to my research pursuit. I would like to start with special thanks to my main supervisor, Dr. Rupert Young. Rupert is an embodiment of an affable and charming personality. He bears a deep sense of understanding and knowledge in optics and computer vision methods. It is due to the thought provoking discussions with him over a wide range of topics, I have realised that a deep and thorough understanding of a problem helps in arriving at a solution in an efficient way. I thank him for generously sharing his wisdom and for the kind cooperation that he bestowed upon me in all matters for the asking. The way he greets me with a pleasant smile on his face is greatly alleviating and dispels any hesitation on my part for approaching him with any problem. I owe a deep sense of gratitude for his keen support, patience and guidance without which my research wouldn't have been possible.

I would also like to take this opportunity to acknowledge my gratitude to my co-supervisor, Prof. Chris Chatwin. Chris is the director of our research group and has an exuberant experience in research on various fields with numerous publications to his credit. I thank him for providing me various opportunities to work on real-time projects that gave me a broader sense of understanding and approach in solving several practical research problems. I am pleased to have had the honour of working under him and receive valuable advice and directions regarding strategies to be adopted in facing challenging research problems. I owe a deep sense of gratitude to him for making my research work truly productive and stimulating.

I would like to thank Dr. Philip Birch, for all his support and advice that helped me gain a deeper understanding on practical research issues which improved my programming skills. He helped me learn a great deal of skills involved in innovating research into a practical cutting edge solutions for various real-time applications. I am thankful to the wonderful administration and support staff at the School office. Thanks also goes to my research colleagues and students with whom I have spent a wonderful time at Sussex.

Finally, I would like to thank my family and friends for their support and for always standing by me during the ups and downs of my life. I am very grateful to the supreme light that drove and drives me in all walks of my life for making my research a plain sail.

Nagachetan Bangalore Manjunathamurthy

University of Sussex

January 2012.

List of Acronyms

AC	Active contours
ACE	Average correlation energy
ACH	Average correlation height
AWGN	Additive white Gaussian noise
ATR	Automatic target recognition
ASM	Average similarity measure
CM	Computational model
DoG	Difference of Gaussian filter
DS	Door surveillance system
DSTL	Defence science and technology laboratory
EACS	Enhanced active contour snakes
EKF	Extended Kalman filter
FFT	Fast Fourier transforms
FFTW	Fastest Fourier transform in the West
GACO	Global active contour orientation
GHT	Generalised Hough transforms
HW	Harris window

IFFT	Inverse fast Fourier transforms
i-LIDS	Imagery library for intelligent detection systems
LoG	Laplacian of a Gaussian filter
MACE	Minimum average correlation energy filter
MACH	Maximum average correlation height filter
MVSDF	Minimum variance synthetic discriminant function
MoG	Mixture of Gaussian
OT-MACH	Optimal trade-off maximum correlation height filter
ONV	Output noise variance
PF	Particle filter
SDF	Synthetic discriminant function
SIFT	Scale invariant feature transform
SURF	Speeded up robust features
SW	Surveillance window

List of Symbols

α	Coefficient of power spectral density of additive input noise
β	Coefficient of average power spectral density of training images
γ	Coefficient of similarity matrix of the training images
σ	Standard deviation
$\forall$	Logical condition ‘for each’
ϵ	Notation for signifying ‘belongs to’
a	Active contour snake parameter emphasising the continuity energy
b	Active contour snake parameter emphasising the curvature energy
c	Active contour snake parameter emphasising the image energy
N	Total number of snaxels in the contour vector
$F^{-1}(x)$, $F(x)$	Fourier transform and inverse Fourier transform
θ_c	Global active contour orientation angle
$\theta_i(x, y)$	Orientation angle at each contour point in a contour vector
R_{Harris}	Harris response function
S_x	Similarity matrix of the training image set
D_x	Average power spectral density of the training image set

Table of Contents

List of figures

List of Tables

List of Publications

1. Philip Birch, Bhargav Mitra, **Nagachetan Bangalore**, Saad Rehman, Rupert Young, Chris Chatwin, "Approximate bandpass and frequency response models of the difference of Gaussian filter", *Optics Communications*, vol. 283, no. 24, pp. 4942-4948, December 2010.

2. **Nagachetan Bangalore**, Rupert Young, Philip Birch, Chris Chatwin, "Tracking moving objects using bandpass filter enhanced localisation and automatic initialisation of active contour snakes", *ICGST International Journal of Graphics, Vision and Image Processing GVIP*, Vol. 10, no. 4, pp. 1-8, October 2010.

3. **Nagachetan Bangalore**, Waqas Hassan, Bhargav Mitra, Philip Birch, Rupert Young, Chris Chatwin, "Door surveillance using edge map-based Harris corner detector and active contour orientation", in *Proc. of Visual Information processing XX, SPIE Defense, Security + Sensing*, vol. 8056, Orlando, Florida, USA, April 2011, pp. 805608-1:10.

4. Waqas Hassan, **Nagachetan Bangalore**, Philip Birch, Rupert Young, Chris Chatwin, "Object Tracking in a Multi Camera Environment," in Proc. of *IEEE International Conference on Signal and Image Processing Applications (ICSIPA 2011)*, Kuala Lumpur, November 2011.

5. Ahmad Alkandri, Akber Gardezi, **Nagachetan Bangalore**, Philip Birch, Rupert Young, Chris Chatwin, "Automatic parameter adjustment of difference of Gaussian (DoG) filter to improve OT-MACH filter performance for target recognition applications", in Proc. of *Electro-Optical and Infrared Systems: Technology and Applications VIII, SPIE Europe Security+Defense*, vol. 8185, Prague, Czech Republic, September 2011, pp. 81850M-1:10.

6. Waqas Hassan, **Nagachetan Bangalore**, Bhargav Mitra, Philip Birch, Rupert Young, Chris Chatwin, "Robust human intrusion detection technique using hue-saturation histograms", in *Proc. of Optical Pattern Recognition XXII, SPIE*

Defense, Security + Sensing, vol. 8055, Orlando, Florida, USA, April 2011, pp. 80550J-1:12.

7. Bhargav Mitra, Waqas Hassan, **Nagachetan Bangalore**, Philip Birch, Rupert Young, Chris Chatwin, "Tracking illegally parked vehicles using correlation of multi-scale difference of Gaussian filtered patches", in *Proc. of Optical Pattern Recognition XXII, SPIE Defense, Security+Sensing*, vol. 8055, Orlando, Florida, USA, April 2011, pp. 805503-1:9.

8. Bhargav Mitra, **Nagachetan Bangalore**, Waqas Hassan, Philip Birch, Rupert Young, Chris Chatwin, "Change of illumination tolerant scene surveillance using a multi-stage edge detector", *Asian Journal of Physics*, Vol 19, No. 1, pp. 87-97, 2010.

9. Akber Gardezi, **Nagachetan Bangalore**, Ahmad Alkandri, Philip Birch, Rupert Young, Chris Chatwin, "Application of speed-enhanced spatial domain correlation filters for real-time security monitoring", in *Proc. of Optics and Photonics for Counterterrorism and Crime Fighting VII, SPIE Europe Security+Defense*, vol. 8189, Prague, Czech Republic, September 2011, pp. 81890R-1:12.

10. Waqas Hassan, Bhargav Mitra, **Nagachetan Bangalore**, Philip Birch, Rupert Young, Chris Chatwin, "Image processing methods for event detection from video surveillance sequences", *Information Technologies, System and Networks (ITSN-2010)*, Moldova, May 2010.

11. Philip Birch, Waqas Hassan, **Nagachetan Bangalore**, Rupert Young, Chris Chatwin, "Stationary Traffic Monitor," *4th International Conference on Imaging for Crime Detection and Prevention (ICDP-11)*, London, November 2011.

12. Chris Chatwin, Rupert Young, Philip Birch, Waqas Hassan, Bhargav Mitra, **Nagachetan Bangalore**, Ioannis Kypraios, "Global Panopticon", *IET*, Invited Presentation, The Hawth-Spotlight, Crawley, Sussex, 8th October, 2009.

Technical Reports

1. **Nagachetan Bangalore**, Bhargav Mitra, Philip Birch, Rupert Young, Chris Chatwin, "Filter Design, Target Reference & Clutter Models for Recognition and Prioritisation", Report 1, *Centre for Defence Enterprise - TR1/US/C4ISTAR/RT/COM/7/101/100301CCRYPB*, 1st March 2010.

2. **Nagachetan Bangalore**, Bhargav Mitra, Philip Birch, Rupert Young, Chris Chatwin, "Design of Optimal trade-off (OT) MACH filter", Report 2, *Centre for Defence Enterprise – TR2/US/C4ISTAR/RT/COM/7/101/100301CCRYPB*, 10th May 2010.

3. **Nagachetan Bangalore**, Bhargav Mitra, Philip Birch, Rupert Young, Chris Chatwin, "Tracking of motor vehicles from the Sussex police helicopter colour camera using the OT-MACH correlation filter", Report 3A, *Centre for Defence Enterprise – TR3A/US/C4ISTAR/RT/COM/7/101/100301CCRYPB*, 5th July 2010.

4. **Nagachetan Bangalore**, Bhargav Mitra, Philip Birch, Rupert Young, Chris Chatwin, "Tracking of vehicles from the Sussex police helicopter colour camera using the OT-MACH correlation filter II", Report 4A, *Centre for Defence Enterprise – TR4A/US/C4ISTAR/RT/COM/7/101/100301CCRYPB*, 25th September 2010.

5. **Nagachetan Bangalore**, Bhargav Mitra, Philip Birch, Rupert Young, Chris Chatwin, "Reducing the Operational Burden On Imagery Analysts -Target Detection, Recognition, Identification and Prioritisation - Sussex", Report Final, *Centre for Defence Enterprise – TR5F/US/C4ISTAR/RT/COM/7/101/110125CCRYP*, 25th January 2011.

Chapter 1

INTRODUCTION

1.1 Motivation

Image is an information carrier according to a computer vision scientist [1], [2]. The information contained in an image may not always be perceivable with the human eye. The information may be corrupted by noise or simply be combined with information that is of no interest creating a confusion. The first and most important step in image processing or analysis involves segmentation of the objects in an image. Segmentation divides an image into its constituent parts. Segmentation algorithms are generally based on discontinuities and similarities in an image. It is often difficult to identify and classify similarities such as edges, especially if they are spurious. Hence, high level segmentation methods involve techniques utilising *a priori* knowledge about the object's shape, texture, colour or position which are included in the search procedures.

Object tracking involves the detection of moving objects over multiple frames captured from a video source. The objects must be associated with the same object observed or segmented in the previous frames in order to successfully track an object. With the tracking information of an object it is possible to ascertain a great deal of information regarding the nature of the object that is considered for tracking. The task of tracking an object becomes particularly challenging when there are multiple moving objects involved, when the objects are moving at high velocities or when the camera is constantly in motion making the whole scene dynamic.

The tracking process can be divided into the processes of target representation, localisation, filtering and association. The target representation and localisation process

attempts to recognise objects in the scene and represent them in a simplified way. Blob detection is one the common methods for achieving this, where the regions of the image are segmented from the background using a known characteristic of the object such as its relative speed, its intensity or colour. Moving regions can be extracted using a number of methods such as optical flow [14], [6], which continually updates the velocity vectors across the image. Multiple cameras can be used to acquire the position and movement information in three dimensions. The use of multiple cameras is by far the most robust method to extract moving objects in three dimensions but it is not practical for most applications. The most common method for extracting moving regions from a single static camera is by using the background subtraction method to distinguish between the area of movement and the areas of the background.

Objects can also be represented by the shape of their boundaries and can be tracked across consecutive frames using active contour methods [2], [3], [11], [18]. Active contour methods attempt to identify the target boundary points using random and active contour segmentation and hence allow tracking between frames. These methods are useful in tracking deformable dynamic objects based on several image characteristics such as intensity, colour or edges [48], [32], [29], [28], [26] and can therefore track objects in moving backgrounds thus making them suitable for most real-time object segmentation and tracking scenarios. Chapter 2 of the thesis discusses an adaptive segmentation and tracking technique using the enhanced active contour snake (EACS).

When a particular type of target is to be tracked in a scene containing similar moving objects, it is often necessary to use correlation pattern recognition methods, such as maximum average correlation height (MACH) filters [75], [67], [77], [98], to classify, locate and track the objects in the scene. This can be used in conjunction with enhanced active contour techniques to reduce the computational cost involved in the training of

the correlation filters for pattern recognition and tracking purposes. An adaptive area of support based on the enhanced active contour method, together with the optimal trade-off maximum average correlation height (OT-MACH) filter, can be used for robust object segmentation and tracking in challenging scenarios. This method is described and demonstrated in Chapter 5 of the thesis.

1.2 Deformable templates

Deformable templates are an important approach for object estimation for segmentation and tracking. The theory of deformable templates can be related to the shape class description based on pattern theory [32], [29], [24], [12], [10], [8].

Deformable templates utilise prior knowledge of the shape of the object specified in the form of a sketch, binary template or parametric prototype. The *a priori* knowledge is encoded either in the form of the edge information computed from the binary template or the parameter vector. The difference between snakes and deformable templates is that snakes are form-free energy minimising functions [3], [13], [15], [17], [20]. In snake models, there is no global structure of the curves except for the general regularisation constraints such as continuity and smoothness of the boundary to be contoured. The parametric deformable templates control deformations using a pre-defined set of parameters capable of encoding a specific shape to initiate deformation. Hence, the deformable templates are used when more specific shape information is available than can be described either by a binary template or a set of parameters [35], [32], [27].

The prototype template describes the most prominent instance of the object boundary defining the shape of the object to be contoured. A parametric transformation is applied to the prototype to deform its boundaries varying the deformation parameters in order to capture a large variety of possible instances of the object. Several variations may be

captured by making random deformations to the prototype so that the deformed template matches the object of interest. However, the object of interest may be noise corrupted or degraded causing the original shape to be lost. In such a circumstance, a deformed template may match the object better than the original prototype defined.

By using an appropriate edge detector [14], [64], [46], the object boundaries can be extracted from the image. The deformable templates can be matched for all objects found using the template base. A potential energy function can be used to check for similar objects by aligning the templates from the database with the image being used.

The detection can be simplified and improved by imposing a probability distribution on the images in the database. It is assumed that the prototype template defined is the most likely *a priori* shape of the object. One of the earliest approaches to deformable template analysis was aimed at finding facial features for human recognition purposes [8], [7], [17], [62]. The deformable template techniques depend on a number of parameters which is cumbersome for optimisation purposes. Thus an alternative is to investigate techniques that use fewer parameters. Snakes are a more popular approach, that evolve a set of points (a contour) to match the object in the image data rather than evolving a set of shapes to match the object [24] [29].

1.3 Active contours (Snakes)

Active contours or Snakes are a completely distinct approach to feature extraction [3]. The active contour is featured as a set of points enclosing a target feature to be extracted. It can be visualised like a balloon that is placed outside the object, enclosing it to find the shape of the object in an image [2]. Similarly, the active contours arrange a set of points in a way to describe the shape of the target object. Active contour snakes

were originally designed for interactive extraction of the object shape, though they are now usually deployed for automatic feature extraction [6], [11], [5], [29].

Active contours are described as an energy minimisation process. The target object feature is a minimum of a suitably formulated and balanced energy functional. The energy functional includes more information than just the edge information of the object under consideration. The energy functional includes properties that control the snake or contour's progression.

A snake represents a compromise between its own properties, known as intrinsic properties, and the image properties, also known as extrinsic properties. Based on this the active contour snake energy functional is the addition of a function of the contour's internal energy, constraint energy and the image energy. These are denoted as E_{int}, E_{image} and E_{con}, respectively. The energy terms are the function of the set of points that make up the snake, represented as $v(s)$ which is a set of the x and y co-ordinates of the points in the snake.

The energy functional is defined as the integral of all the energy functional of the snake, given that s is a member of the set [0,1] is the normalised length around the snake. The energy functional of the snake E_{snake} is given by equation 1.1:

$$E_{snake} = \int_{s=0}^{1} E_{\text{int}}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))ds \qquad (1.1)$$

where the internal energy E_{int} controls the natural behaviour of the snake and the arrangement of the snake points in a contour and the image energy E_{image} controls the emphasis of the snake in choosing low-level features of the image such as the edges of the object [25], [32], [2]. The constraint energy allows the higher level information to control the snake evolution. The criterion used to make the active contour evolve is to

minimise the energy in equation 1.1 for each point in the contour. A set of new contour points are chosen with lower energy that is a better match to the target object shape based on the values of the energy terms around it. The active contours seek a set if points *v(s)* to choose a lower energy than the original set of points. This can be expressed mathematically as in equation 1.2:

$$\frac{dE_{snake}}{dv} = 0 \tag{1.2}$$

The energy functional is expressed in terms of functions of the snake and the image in which the object is contoured. These functions form the snake energy according to the values chosen for the weighting coefficients. The internal energy of the snake is defined to be a weighted sum of first order and second order derivatives of the contour *v(s)* [10] [3]. This can be expressed as shown in equation 1.3:

$$E_{\text{int}} = a(s)\left|\frac{dv(s)}{ds}\right|^2 + b(s)\left|\frac{d^2v(s)}{ds^2}\right|^2 \tag{1.3}$$

The first order differential measures the energy due to stretching, also referred to as elastic energy of the snake. A higher value of this differential implies a high rate of change in the region of the contour. The second order differential measures the energy due to bending, also known as the curvature energy. The first order differential is weighted by *a(s)* controlling the rate of the contribution of the elastic energy due to point spacing in the contour. The second order differential is weighted by *b(s)* controlling the rate of contribution of the curvature energy due to point variation. The parameters *a* and *b* control the shape of the snake that is needed to contour an object. Low values of *a* imply that the points can change in spacing flexibly, whereas higher values imply that the snake is formed by evenly spaced contour points which are known

as snaxels. Lower values of *b* imply that the curvature is minimised and the contour can form corners in its length, whereas higher values of *b* imply that the contours are smooth. The image energy attracts the contour to low-level features such as brightness or edge data. For instance, the edges and intensity values in the image can be used to contribute to the energy functional. Each of these energies can be controlled by weighting factors. The image energy function can be expressed as in equation 1.4:

$$E_{image} = W_{Intensity}.E_{\text{intensity}} + W_{gradient}.E_{gradient} \tag{1.4}$$

where the intensity based energy is denoted as $E_{intensity}$ and the corresponding weighting parameter is denoted as $W_{intensity}$. The energy based on the gradient value is represented as $E_{gradient}$ and the corresponding weighting coefficient controlling the emphasis of the gradient energy is denoted as $W_{gradient}$. Several combinations of low-level features can be emphasised and extracted using them as the image energy in the snake energy equation [31], [32], [35]. The work in the thesis discusses the use of the difference of Gaussian (DoG) filtered image term in the energy equation so emphasising the negative polarity around the zero-crossing edge detected by the application of this band pass filter to the image. Figure 1.1 illustrates an active contour snake deformed on to a hand image; note that the contour is an open contour, meaning that the first snaxel and the last snaxel of the contour are not connected to each other to form a closed contour.

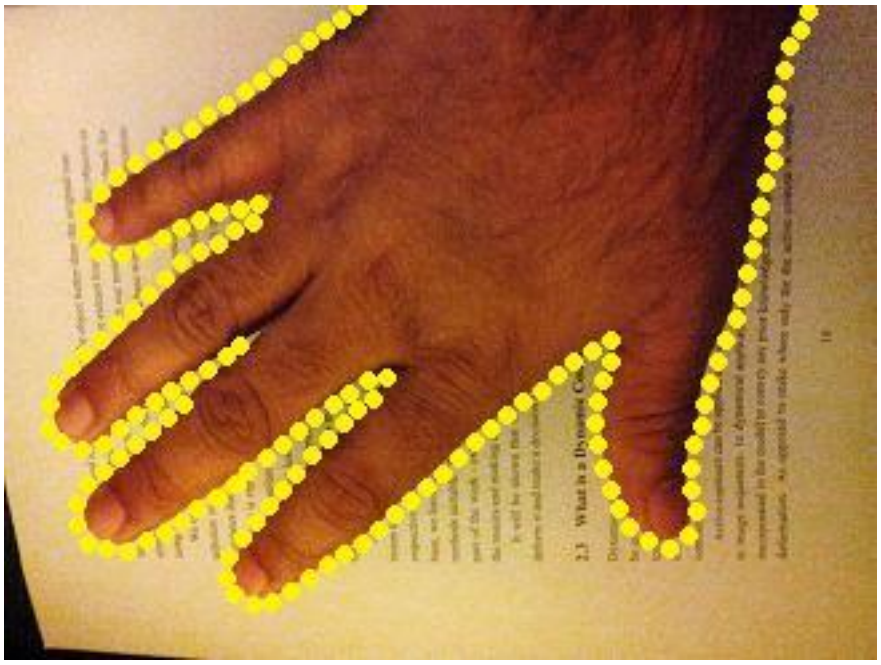

Figure 1.1: Active contour snake contouring the shape of the hand in the image (drawn in yellow)

The active contours can be finite element based or finite difference based. The most common and easier way of realising the active contours was proposed by Shah *et al* [19], [16], [25]. It is known as the Greedy algorithm for snakes. It implements the energy minimisation process purely as a discrete algorithm. The greedy process starts by a user defined specification of an initial contour. More recent work suggests several automatic methods of specifying an initial contour of the snake [3], [8], [27]. The greedy algorithm evolves the snake in an iteration based local neighbourhood scan for the lowest energy snaxel points. The new set of points are those neighbourhood points which have the overall lowest energy points computed for each snaxel based on the energy minimisation expression for the snake. The technique is called 'greedy' by virtue of the fact that the search propagates over the contour for each snaxel around its local neighbourhood [21]. The enhanced active contour snake technique discussed and designed in this thesis uses the greedy search technique for the energy computations.

More detailed discussion of the enhanced active contour snakes (EACS) is given in Chapter 2 of this thesis.

1.4 Correlation pattern recognition

Correlation pattern matching or recognition provides a mechanism for comparing two image signals. It has the benefit of being shift invariant so it can also provide tracking, if implemented via the Fourier domain, at no extra computational cost. If *f*(.) and *g*(.) are two continuous functions, the correlation between the two signals *c(x)* is given by:

$$c(x) = f(x) \otimes g^*(x) \tag{1.5}$$

Equation 1.5 is in fact the integral equation shown in equation 1.6;

$$c(x) = \int_{-\infty}^{\infty} f(\tau) g^*(x+\tau) d\tau \tag{1.6}$$

where τ is a dummy variable. The function *c(x)* gives a measure of the similarity of the function *f(.)* and the function *g(.)* at position *x*. The function *f (.)* is now referred to as the input signal and the function *g(.)* as the target signal [76], [77]. However, when dealing with discrete digitised images, it is more common to modify the integral equation 1.6 so that it is expressed as a discrete summation, as shown in equation 1.7:

$$c_{(m,n)} = \sum_k \sum_l f_{(k,l)} g_{(k+m,l+n)} \tag{1.7}$$

where *m* and *n* are valid 2-D pixel co-ordinates. For larger images this can be rather computationally expensive so it is common to use the Fourier transform relationship which can be expressed as in equation 1.8:

$$c(m,n) = F^{-1}\left(F[f(m,n)].F[g^*(m,n)]\right) \tag{1.8}$$

where the $F(.)$ indicates a Fourier transform operation and $F^{-1}(.)$ denotes the inverse Fourier transform of the function contained in the brackets. In a discrete image, equation 1.7 is therefore replaced by two forward fast Fourier transforms (FFTs) and a complex multiply of the conjugate Fourier spectrum of the target function with the Fourier spectrum of the input signal function, the result of which is then inverse fast Fourier transformed (IFFT) [93], [92], [99]. Since it is convenient for the answer to be real, the modulus squared of $c(m,n)$ is used by convention.

The component $F[g^{*}(m,n)]$ of equation 1.8 can be treated as a filter function. This is the most basic form of correlation filter and is also known as the matched filter. However, correlation filters have a few shortcomings [91], [92], [89], namely:

1. The filter has a very large bandwidth; this means that it is poor at discriminating against similar targets
2. Only one template image can, in practice, be used to train the filter due to the broad correlation response produced.
3. Knowledge of image noise, out of class targets, and background clutter is neglected
4. The filter has some scale or rotation invariance but less than that of an appropriately multiplexed filter.

The drive behind correlation pattern recognition research is to overcome these limitations by modifying the filter function. Hester and Casasent made a major contribution to the field in 1980 with the formulation of the synthetic discriminant function (SDF) [89], [76] that allows multiple training images to be included in a single filter design. A plethora of synthetic discriminant function (SDF) based filters have been proposed since then to accommodate various kinds of arbitrary distortions in the

filter training images caused by changes in the viewing angle, scale and rotation of the target object [84], [87].

The primary issues that are to be borne in mind while designing a correlation filter are [93]:

1. the filter should be tolerant to distortions

2. it should be able to suppress clutter and/or noise

3. the correlation peak should be easily detectable

It has been reported in the literature that early SDFs, by and large, failed to meet any of the above three design criteria [75], [77], [76], [87]. Progress in the design process was achieved with the rigorous development of the minimum variance synthetic discriminant function (MVSDF) by Kumar in 1986 [96], [97]. Kumar and co-workers proved that the minimum variance synthetic discriminant function (MVSDF) should be treated as the optimum filter that minimises the effects of additive noise. However, the MVSDF has some serious drawbacks. The downsides of implementing the filter include the inversion of a large covariance matrix making the process computationally intensive. Moreover, performance of the filter suffers because of the fact that clutter noise is not additive white Gaussian noise (AWGN). Another filter improvement in the SDF family that has received some attention from the automatic target recognition (ATR) community is the minimum average correlation energy (MACE) filter by Mahalanobis *et al* [90], [89]. Although the MACE filter is capable of generating detectable correlation peaks by suppressing the image clutter, it is too sensitive to image distortions.

The maximum average correlation height filter (MACH) was also proposed by Mahalanobis *et al* [93], [98], [99]. One of the hard constraints involved in the SDF based design method is to have the correlation peaks pre-specified and of constant value. This requirement is absolutely unnecessary, not only because it limits the number of possible SDF solutions [88], [89], but also because of the fact that in practice there is very low probability of correlating the filter function with a scene where a target will have exactly the same orientation as that of any view included in the training set. The MACH filter theory thus allows a better filter solution by removing the peak-height constraint from the SDF based design techniques. The MACH filter design includes the minimisation of an average similarity measure (ASM) that leads to a compact set of correlation planes that resemble each other and exhibit the least possible variation [93]. This statistical approach to the correlation filter design process makes the filter more robust as compared to other designs based on SDFs with hard constraints.

Since the motivation of the work in the thesis is to enable adaptive segmentation and tracking of moving objects of a given type, embedded in clutter, an Optimal trade-off maximum average correlation height (OT-MACH) filter, assisted by the enhanced active contour snake (EACS) in the tracking research presented in this thesis, has been found to outperform any other traditional SDF based filter design.

The MACH filter maximises the relative height of the average correlation peak with respect to the expected distortions. An improved version of this can be designed to cope with specific clutter noise. The Optimal trade-off (OT) MACH filter transfer function in the frequency domain can be expressed as [79], [88] in equation 1.9:

$$h = \frac{m_x^*}{\alpha C + \beta D_x + \gamma S_x} \tag{1.9}$$

where, α, β and γ are non-negative OT parameters, m_x is the frequency domain average of the training vector and C is related to the power spectral density matrix of the additive clutter. The quantity D_x is the average power spectral density of the training images. S_x denotes a similarity matrix of the training image set. The different values of α, β and γ control the OT-MACH filter's behaviour to match different application requirements.

As an example, a tank dataset used to train the OT-MACH filter, is considered. The training images are rendered for different rotation angles (between 0 to 360 degrees) and for different look-down angles and scales (view range). One such dataset is depicted as shown in Figure 1.2.

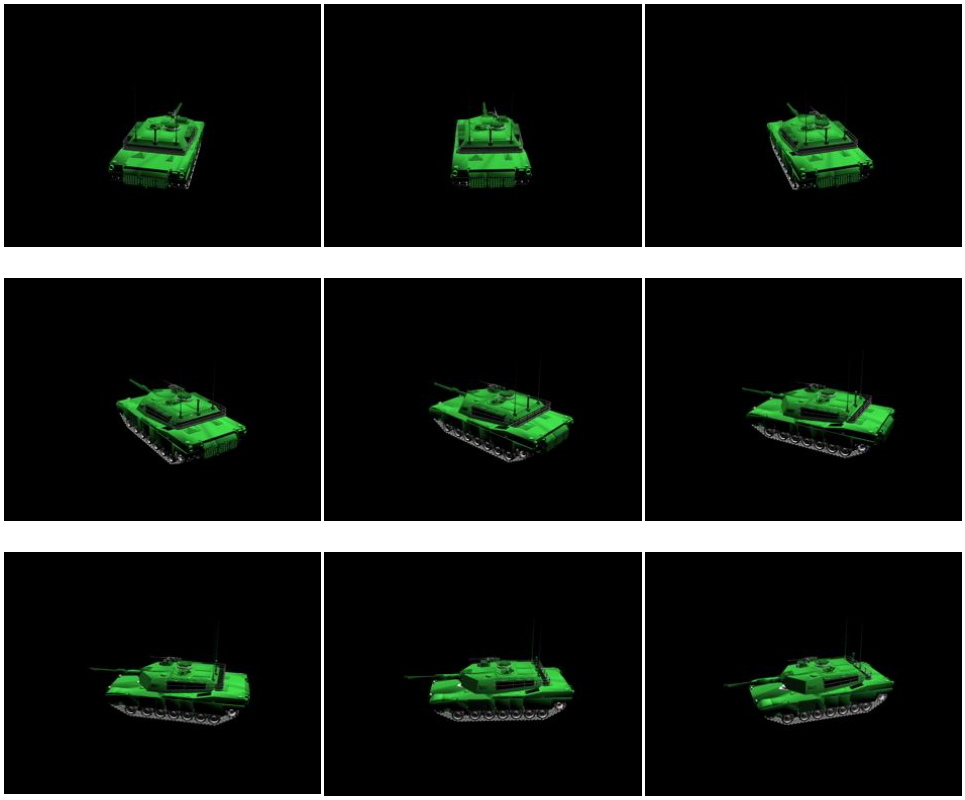

Figure 1.2: An example tank training dataset (angle of rotation between 0 and 180 degrees)

A sub-set of the training images can be used to compute the OT-MACH filter transfer function as expressed in equation 1.5. The correlation plane obtained by correlating the filter function with an in-class tank image can be represented as a correlation mesh plot as shown in Figure 1.3.

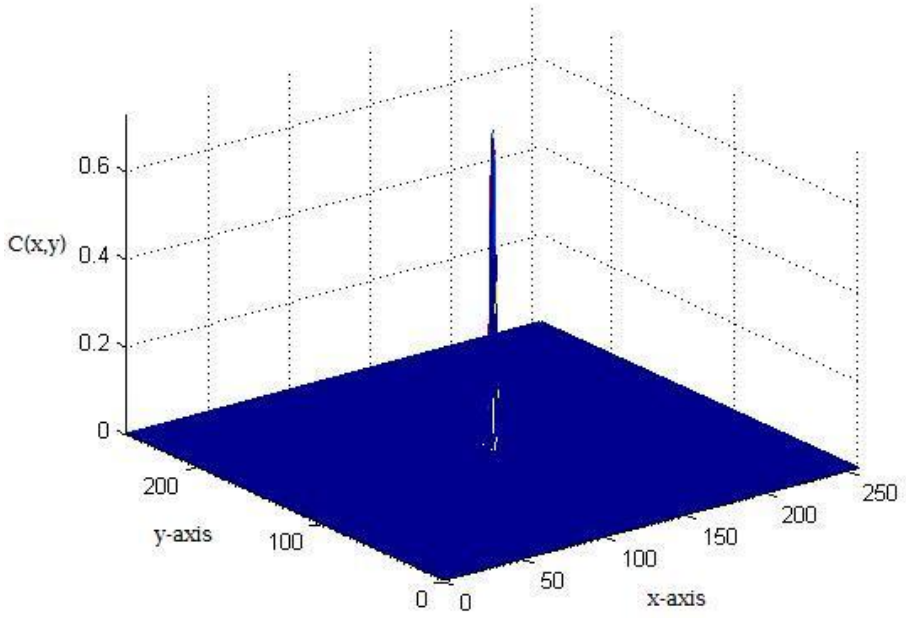

Figure 1.3: Correlation plot for an in-class target image correlated with an OT-MACH filter function

The location of the matched target is computed from the correlation plane by scanning for the maximum peak height in the two dimensional correlation plane thus obtained [74], [85], [84], [79].

1.5 A brief overview of the thesis chapters

Chapter 2: An enhanced active contour snake (EACS) computational model to track dynamic a moving object in a scene has been developed and discussed in this chapter. It is demonstrated how the active contour snakes can be enhanced to perform accurately in real-time, with the use of the difference of Gaussian (DoG) filter. The DoG filter energy

term is incorporated in the enhanced active contour expression to increase the model's ability to contour the exterior edges of a moving target. Automatic initial contour extraction along with the snake deformation and progression techniques is discussed in detail. Finally, the chapter presents several example results of the enhanced active contour technique. Evaluations of the computational model are discussed and summarised.

Chapter 3: An extension to the enhanced active contour snake (EACS) technique to perform multiple object tracking is proposed in this chapter. The contour splitting, merging and selection criteria used for multiple object contouring are described along with a few examples demonstrating the contour splitting and multiple object contouring process. The statistical parameters of the EACS, such as the active contour boundary rectangle and the active contour centroid, are discussed in detail. An application to perform upper body humanoid skeletal modelling is designed to demonstrate the importance of computing accurate active contour vectors using the EACS framework. Results determining the skeletal stick model for the upper body of a humanoid are presented and summarised.

Chapter 4: The overall enhanced active contour vector orientation, called the global active contour orientation (GACO) measure, of the EACS is proposed in this chapter. It is demonstrated how the GACO statistical measurement for each active contour vector is computed using simple geometric calculations and assumptions. A robust algorithm for door surveillance (DS) is developed and presented in detail in order to evaluate the performance of the proposed GACO measure computation. The door surveillance application developed is evaluated on the *i-LIDS* door surveillance dataset provided by the UK Home office. The performance results demonstrating the use of GACO to evaluate the *i-LIDS* dataset is discussed and summarised.

Chapter 5: The enhanced active contour snake (EACS) based optimal trade-off maximum correlation height (OT-MACH) filter tracker is designed and demonstrated in this chapter. The dynamic filter update and rotational multiplexing techniques are proposed for robust tracking. The results obtained are discussed and illustrated to evaluate the OT-MACH tracker on Sussex police and DSTL videos.

Chapter 6: In the final chapter the entire thesis is discussed. Conclusions are drawn and future work is outlined.

Chapter 2

ENHANCED ACTIVE CONTOUR SNAKES INCORPORATING A DIFFERENCE OF A GAUSSIAN FILTER

2.1 Introduction

Contour extraction is one of the basic tasks in image and video processing. A snake can be described as an energy minimisation spline or a curve that represents several salient features of the object being contoured. Active contours are extensively used in computer vision particularly to identify object boundaries [19], shape modelling [21], segmentation [25] and motion tracking [59]. Active contours or snakes are curves defined within an image that move based on the influence of internal forces inside the curve and the external forces derived from the object data.

Snakes are a more general technique of matching a deformable model to an image by means of energy minimisation. The basic snake model is a controlled continuity curve under the association of image forces, external constraint forces and internal forces. The external image forces propel the snake towards the salient image features. The internal snake force serves to improve the smoothness constraint of the contour and the external constraint forces drive the snake to near the desired local minimum points, in this case exterior edges [48], [47], [39], [70].

The position of the snake can be parametrically represented by:

$$v(s) = [x(s), y(s)], s \in [0,1] \qquad (2.1)$$

The energy functional can be expressed in the continuous domain as:

$$E^*_{snake} = \int_0^1 \left[a.E_{\text{int}}\left(v(s)\right) + b.E_{ext}\left(v(s)\right) \right].ds \qquad (2.2)$$

where E_{int} and E_{ext} are the internal and external energies. The internal energy signifies the energy due to bending and the external energy signifies the image energy and the constraint energies to contour the required point on an object. The parameters a and b are the constants controlling the influence of different energies giving importance to the object data to be contoured [3], [2].

There are two key issues while contouring an object using the active contour techniques in real-time [28], [32]. Firstly, the initial contour must be close enough to the boundary of the object being contoured. If the initial contour is near to the boundary of the object, the active contours require less iteration to dynamically contour the object boundary. The second issue is that the active contours require an external energy term to distinguish between the internal and external edges of the object. An external energy force is also required for disambiguation of two or more objects or their edges joined together forming a cluster.

The first issue of having an initial contour close enough to the boundary of the object is solved in the following sections making use of a Difference of Gaussian (DoG) filter term in the overall snake expression for energy minimization. The following sections discuss a new technique of incorporating a term derived from a thresholded Difference of Gaussian (DoG) filtered image in the equation to allow quick and accurate contouring of the edges of the object being tracked. A technique for initialising snake points at the first iteration using the thresholded DoG filtered images to improve accuracy is introduced. Methods to decrease the number of snake points in the contour

is discussed to improve the robustness in real-time tracking of an object in a sequence. Unlike most other techniques for finding salient contours, the model is active and fast.

2.2 Chapter Organisation

The chapter is organised as follows: the design of the difference of Gaussian filter is discussed in detail in the next section. The following sections describe how the DoG filter term is implemented and incorporated into the snake expression to assist in the initialisation of the active contours. Section 2.4 discusses in detail the enhanced active contour expression using the DoG filter energy term. In Section 2.5 an automatic initial contour extraction process is discussed in detail along with a few results. Snake deformation and progression along with a full flowchart depicting the design of the enhanced active contour extraction technique is discussed in the Section 2.6. Finally, Section 2.7 displays several results of the enhanced active contour technique discussed. Conclusions are drawn in Section 2.8.

2.3 Design of the Difference of Gaussian filter

A Difference of Gaussian (DoG) filter is implemented in the spatial frequency domain of the image as a band-pass filter. The Difference of Gaussian (DoG) function is calculated as a convenient good approximation to the scale normalized Laplacian of Gaussian (LoG) filter as it is accurately and readily implementable in the spatial frequency domain as a band-pass filter [53], [58]. In standard practice the first order or second order derivative functions are applied to an image in order to detect edge locations or zero-crossings. The major disadvantage of using a first order function to detect edges is that there is a need to preselect the thresholds. The use of a second order derivative function implemented using a Laplacian kernel is also disadvantageous due to its acute sensitivity to noise [33], [41]. Hence, the use of the Laplacian kernel to detect edges may result in high frequency noise which disrupts the process of finding

edges. In order to remove the over-sensitiveness of the Laplacian kernel, the image may be low pass filtered using a Gaussian kernel to remove any high frequency noise. An appropriate way of applying the entire scheme to an image is to combine both the kernels into a single Laplacian of Gaussian (LoG) kernel [41] as shown below:

$$I_{LoG}(u,v) = \nabla^2 \left[G(u,v) * I(u,v) \right]$$

$$= \left[\nabla^2 G(u,v) * I(u,v) \right] \qquad (2.3)$$

where in equation (2.3) $I(:,:)$ is the intensity of the image, $\nabla^2 G(:,:)$ is the combined operator also called as LoG operator and $I_{LoG}(:,:)$ is the output image obtained after applying the LoG operator. The Gaussian kernel $G(u,v)$ is applied to the image as a low pass filter to smoothen the high frequency noisy edges. The Gaussian kernel is dependent on the standard deviation σ and is given as:

$$G(u,v) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(u^2+v^2)}{2\sigma^2}} \qquad (2.4)$$

where, the standard deviation is σ, and the value of *u* and *v* lie between -∞ and +∞ [58]. A two dimensional Gaussian function mesh plot is shown in Figure 2.1, for illustration

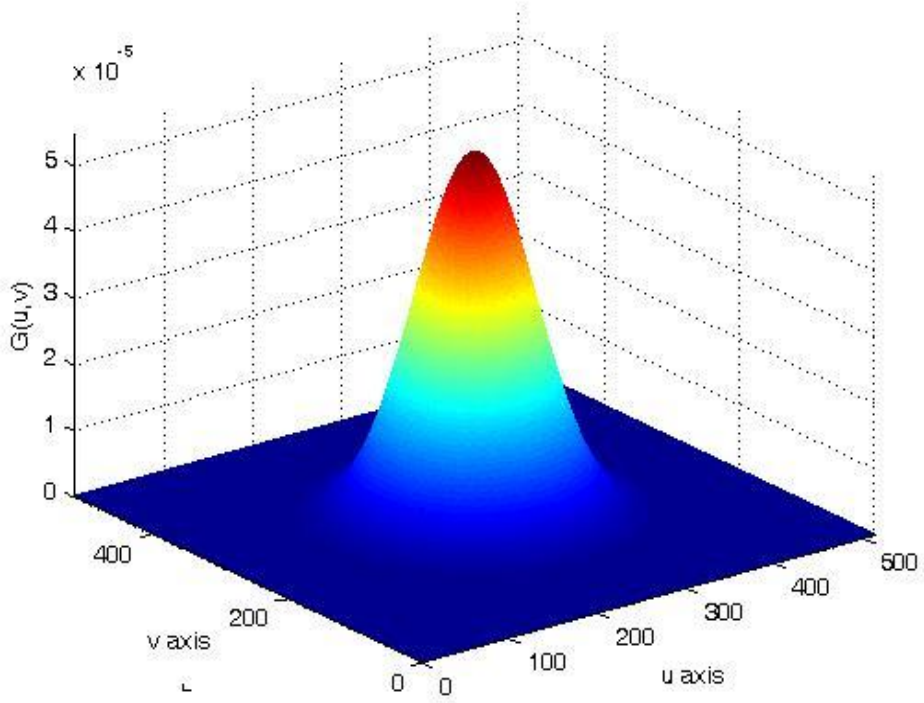

Figure 2.1: Gaussian filter function mesh plot, with σ = 92 pixels with a mesh size of 512 x 512 pixels.

The problem associated with the second order derivative function is the fact that the operator results in multiple pixel thick edges. In order to avoid this problem and to localise the edges, zero-crossings in the image are detected after applying the combined operator [48], [33], [32].

The DoG operator has the advantage of being a better fit numerically when implemented for small kernel sizes. In addition to this, the spatial domain numerical approximation of the LoG operator has a high tendency to have a ringing effect. The ringing effect results in the appearance of spurious or ring like edges near the sharp transitions at the edges of an object in the image. The DoG filter is implemented in the frequency domain by using the σ value for the inhibitory Gaussian as 1.6 times to that of the excitatory Gaussian ensuring a good approximation of LoG filter. It is important

to note that the output obtained after DoG filtering is distortion or illumination-change sensitive depending on the pass band scale selected. Hence, the pass band or, equivalently, the patch size of the filter are appropriately selected based on the object to be contoured in the image. The DoG filter plotted as a band pass filter in the spatial frequency domain is shown in Figure 2.2.

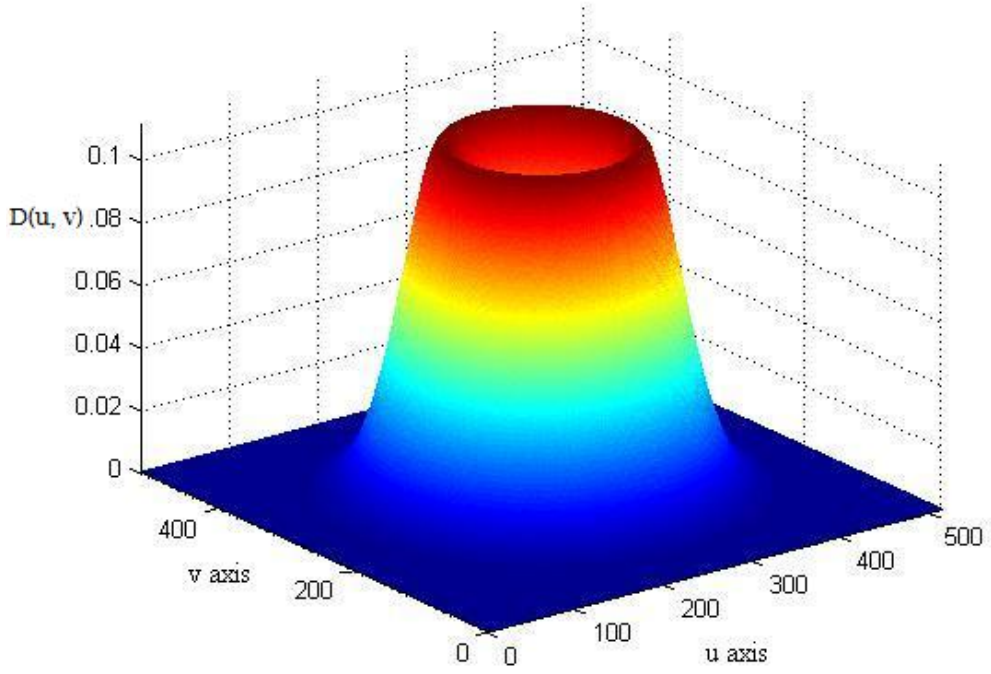

Figure 2.2: DoG filter shown as a band-pass filter in the spatial frequency domain

From equation 2.4 for Gaussian filter function, the DoG filter function can be described as the difference of excitatory and inhibitory Gaussian filter functions:

$$D(u,v) = \frac{1}{2\pi\sigma_2^2}.e^{\left[-\frac{u^2+v^2}{2\pi\sigma_2^2}\right]} - \frac{1}{2\pi\sigma_1^2}.e^{\left[-\frac{u^2+v^2}{2\pi\sigma_1^2}\right]} \qquad (2.5)$$

where, σ_2 and σ_1 are the two standard deviations of the excitatory and inhibitory Gaussian functions. In order to have an optimum performance as a bandpass filter and

to allow accurate zero-crossing detection at the edge locations in the image, the ratio between the two standard deviations is maintained at 1.6 [48], [55], [58].

2.4 Enhanced active contour expression using the DoG filter energy term

Consider an example test frame as shown in Figure 2.3 with a DoG filtered output of the frame as shown in Figure 2.4 with an appropriate bias added so the bi-polar filtered image can be displayed as an intensity image.

Figure 2.3: Test frame from a colour video

Figure 2.4: DoG band-pass filtered image with an appropriate bias added for display

From Figure 2.4, it can be seen that the edge locations of the object under consideration are detected by applying a DoG band-pass filter to the image in Figure 2.3. Due to the second differential operation performed by the DoG filter, the pixel values at the image edges have zero-crossings between positive and negative values. The negative pixel values at the location of the edges can be utilized for the addition of a new parameter into the Snake equation.

Considering Equation 2.2, this can be expressed at any point v_i as:

$$E^*_{snake} = a.E_{\text{int}}(v_i) + b.E_{ext}(v_i) \qquad (2.6)$$

where, i = 1 to n, n being the number of points in the active contour (AC) Snake and the * signifies that this is a continuously updated snake energy [2], [10].

The internal energy function is intended to enforce a shape on the deformable contour and to maintain a constant distance between the points in the contour. It consists of continuity energy denoted as E_{cont} and curvature energy denoted E_{curv}.

An average distance between the contour points is computed and denoted as $D_{average}$ [31], [34]. The variable *a* is initialized and the continuity term is calculated. This can be computed according to:

$$E_{cont}(i) = \left| D_{average} - \left\| \left(P_i - P(:,:) \right) \right\| \right| \tag{2.7}$$

where, P_i denotes the current contour position and $P(:,:)$ represents the initial contour point vector. The mathematical symbols $\left| \right|$ and $\left\| \right\|$ represent the absolute and norm of the terms contained in them, respectively.

The constraint energy can be of many kinds and here the curvature energy is considered. The curvature energy is computed using the initial Contour point vector [31], [19]. The curvature energy, denoted as E_{curv}, controls the curve or bending of the contour and can be computed according to:

$$E_{curv}(i) = \left\| P_{i-1} - 2P_i + P_{i+1} \right\|^2 \tag{2.8}$$

where, P_i denotes the current contour position and P_{i-1} and P_{i+1} represent the previous and next contour position in the contour point vector.

The external energy functional attracts the deformable contour to interesting features, such as object boundaries, in the image. The image energy is computed for each neighbourhood pixel. The image energy can be manipulated to contain the DoG filtered image energy denoted as $E_{DoG,}$ along with the sum of intensity energy and the gradient energy. The magnitude of the intensity of a corresponding image point in the

neighbourhood gives the image intensity energy, E_{in} , and the gradient magnitude of the same point gives the image gradient energy, E_{grad} [2], [1]. The bipolar value of the DoG filtered image at the point in consideration gives the DoG filtered energy term, E_{DoG}. This can be expressed as follows:

$$E_{image} = c.\left(E_{in} + E_{grad} + E_{DoG}\right) \tag{2.9}$$

Here, the constant c is the term controlling the rate of importance given to the image energy E_{image} while computing the overall AC energy at any given point. The DoG filtered image pixel values are added to the equation for each pixel external energy computation. If the value of each DoG filtered output pixel is considered as the energy of the DoG filtered image at that pixel then the energy term E_{DoG} contains a positive or a negative value due to the bipolar characteristics of the DoG filtered resultant image. This contributes towards reducing the overall snake energy at the point in consideration. Equation 2.9 can be modified to contain three different constant values controlling the emphasis given to each energy term while computing the overall energy of the Snake. This can be expressed as:

$$E_{image} = c_1 E_{in} + c_2 E_{grad} + c_3 E_{DoG} \tag{2.10}$$

The constant c_3 gives a measure of the influence of the DoG filtered image energy at the pixels being considered. For full derivation of the enhanced active contour snake expression please refer to Appendix 1. Thus, by adhering to the energy minimization rule of the Snake algorithm, an improved selection of the correct edge pixel for the contouring process is achieved leading to a greater accuracy in the choice of contour edges for the object [2], [3].

2.5 Automatic initial contour extraction

In classical methods, an initial contour point is selected by the user, near to the contour of the target object. Various methods use circular or rectangular initial contour points around the object of interest [34], [35], [28], [27]. The minimum energy functional, its co-ordinates and the weighting parameters are then initialized. The energy functional is calculated, as explained in the previous section, by considering several feature parameters to calculate the overall snake energy. By giving importance to the image energy term in the overall snake equation, the active contour can be made to stick to the zero-crossing edge of the object under consideration.

The new method, designed to initialize the contour on the given object, is based on locating a moving object in the scene and employing the DoG filter image boundary extraction technique. For any given frame (say frame 1) DoG filtering is performed, followed by threshold and morphological operations. The morphological operation, binary erosion, is employed in order to remove unconnected noisy edges. The basic effect of binary erosion is to erode away the boundary regions of foreground pixels so as to remove isolated and discontinuous pixels in the binary image. A subsequent frame (say frame 2) is also processed in a similar manner to obtain two different edge images. This step is illustrated (after adding an appropriate bias to allow display of the images) in Figures 2.5 to Figure 2.8 below.

Figure 2.5: Frame 1 of a colour video to be DoG filtered

Figure 2.6: Frame 2 of the colour video to be DoG filtered

The images shown in Figure 2.5 and Figure 2.6 are DoG filtered at user selected standard deviations with a pre-condition that their ratio is maintained at 1.6. The images are converted from colour to greyscale before applying the DoG filter. The colour images are converted to greyscale by performing a weighted summation of the colour space components of the image to obtain the single channel greyscale image. The DoG filtered and appropriately biased results of both the frames under consideration are shown in Figure 2.7 and Figure 2.8.

Figure 2.7: DoG filtered frame 1 with an appropriate bias added for display

Figure 2.8: DoG filtered frame 2 with an appropriate bias added for display

An absolute subtraction of the DoG filtered output of the frames 1 and 2, shown in Figure 2.7 and Figure 2.8, is performed after a threshold operation. An optimal thresholding technique by Otsu *et al* [54] is used to perform thresholding. The threshold level is automatically selected by Otsu's technique for thresholding. This gives only the moving object edges, comprising both edges in the position of the object in frame 1 and frame 2. Binary erosion is performed on the resultant image to get rid of the residual noisy edges resulting from the subtraction. The resultant image is shown in Figure 2.9.

Figure 2.9: Resultant image after performing absolute subtraction of the thresholded DoG filtered images

The resultant image is now further processed; the coordinates of the first non-zero pixel chain in the image are located by scanning the entire image starting from the top left corner of the image. The non-zero pixel is selected, after verifying that the pixel is not an isolated pixel, to perform robust boundary tracing of the object being contoured. From the first non-zero and non-isolated pixel of the pixel chain, a boundary tracing operation is initiated to obtain a boundary traced image, as shown in Figure 2.10.

Figure 2.10: Boundary traced resultant image (boundary drawn in green for display)

The boundary thus extracted is placed in a coordinate vector array containing the coordinates of all pixels in the boundary. The number of boundary pixels is too large for fast and efficient calculation of the Snake contours. The coordinate vector is reduced so as to achieve fixed pixel spacing between each snake point element, so-called ‘snaxels’. For instance, in this case this is done by selecting every 5^{th} coordinate point in the array to produce a reduced resolution contour vector with fixed pixel spacing of five pixels thus reducing the number of snaxels, *n*, in the contour. This forms the initial contour vector of the snake and is shown below in Figure 2.11. The Table 2.1 describes each energy term included in the Snake energy minimization equation for reference.

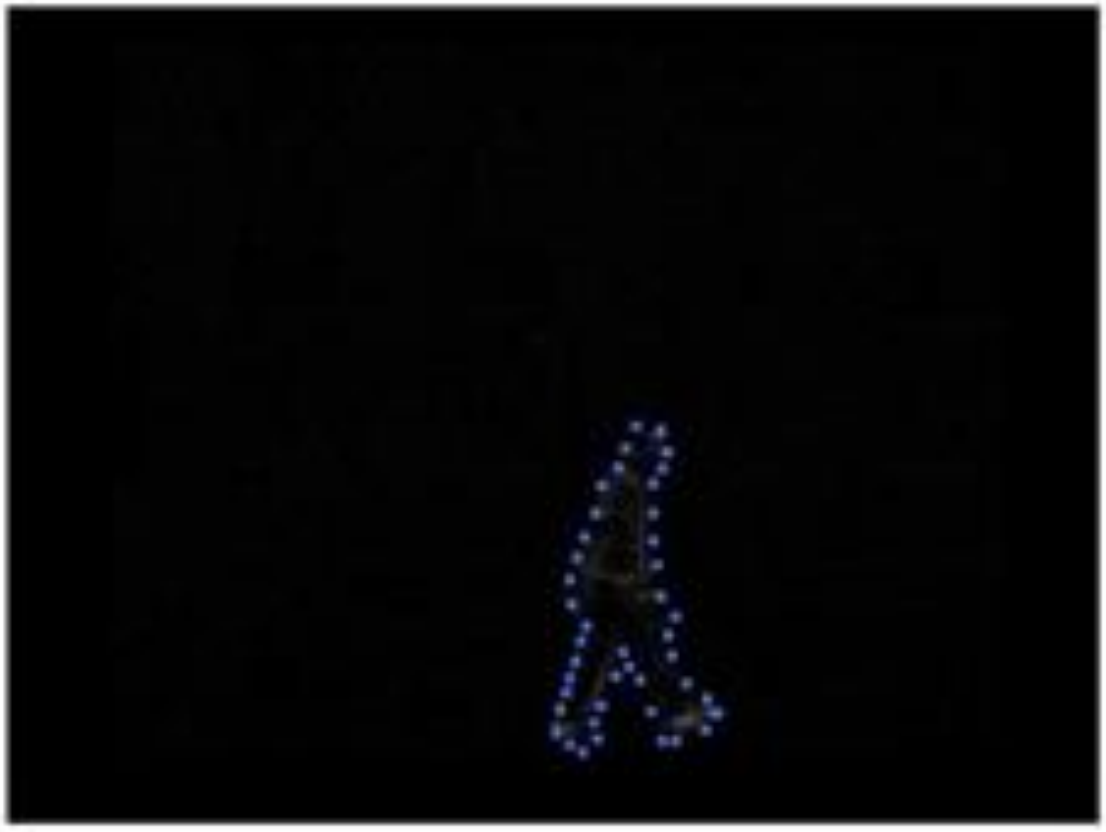

Figure 2.11: Initial reduced contour vector of snaxels plotted as blue dots for display

	Energy term in the Snake equation	**Description**
1	E_{cont}	Energy computed based on the continuity of the Snaxel vector
2	E_{curv}	Energy due to the effect of curvature in the Snaxel vector
3	E_{int}	External energy of the snake based on the image intensity at each Snaxel in the contour
4	E_{grad}	External energy of the snake derived from as the gradient value at each Snaxel in the gradient image
5	E_{DoG}	External energy of the snake deduced from the DoG filtered image at each Snaxel location on the contour vector

Table 2.1 Energy terms of the snake listed and described

2.6 Snake deformation, progression and algorithm design flowchart

2.6.1 Snake deformation

The energy functional and the initial contour are computed as explained in the previous sections considering several feature parameters to calculate the energy (see equations 2.6, 2.7, 2.8, 2.9 and 2.10). Eight point neighbourhood pixels are traversed and the energy functional of each of these points is computed. The total energy of each point calculated is compared with the minimum energy and the new points are appended in the contour point vector, based on the minimum energy criterion. Thus, by iterating for all contour points in the initial contour, the active contour (AC) fits around the moving object under consideration [18], [19].

The active contour of the next frame is based on the active contour vector of the current frame. Thus the current frame contour vector becomes the initial contour vector of the next frame. After setting the initial contour, the snake deformation takes place in an iterative manner, as explained earlier, in order to fit on to the object being contoured. Contouring the same object under motion in subsequent frames, the snake acts as a dynamic deforming contour. This then allows tracking of the moving object. In the frames in which the object becomes static, the Snake remains on the final boundary locations of the object thus locking onto the object.

2.6.2 Snake progression

Due to the dynamic change in shape of the object tracked, there can be an addition or removal of extra snake points into the contour vector. When the length of any segment between two coordinate points in the active contour array goes higher than the average distance (denoted as $D_{average}$) an addition of a coordinate point takes place to compensate for the changes in the object. Similarly, when the length of any segment

between two coordinate points in the array goes lower than the average distance $D_{average}$ a coordinate point is removed from the contour array to compensate for the shrinkage in size of the object being traced. This allows the snake to efficiently define the object shape even if there is a change in the scale or rotation of the object during a tracked sequence in the scene [62], [61], [67]. Figures 2.12 and 2.13 illustrate the snake deformation and progression in an example sequence. Figure 2.13 shows an example of the progressive snake tracking the object moved from its initial location in Frame 3 as shown in Figure 2.12.

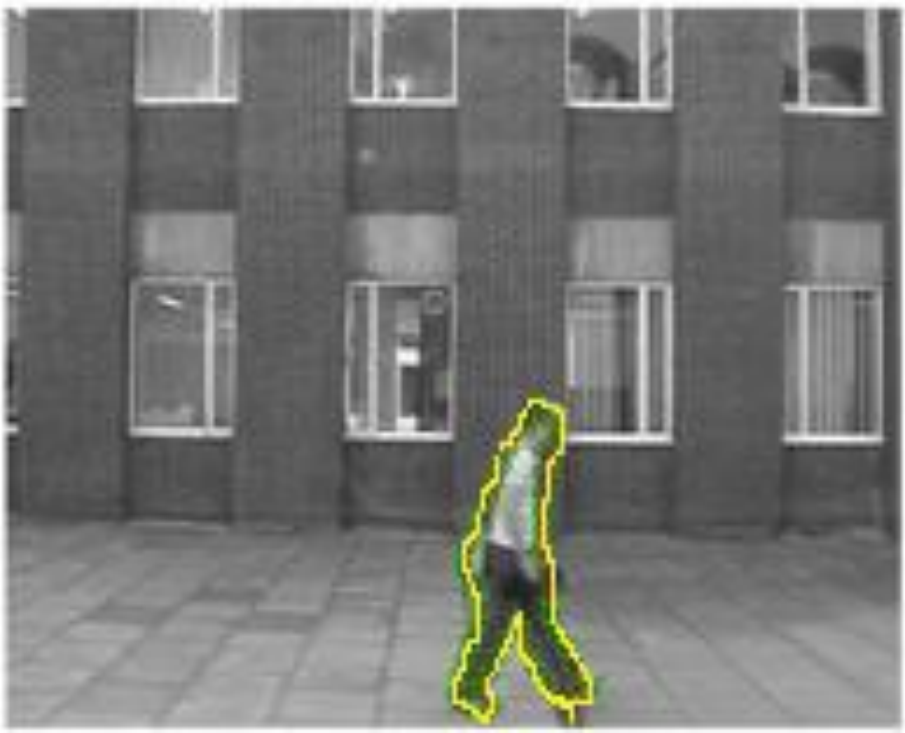

Figure 2.12: Snake deformation onto moving object at Frame 3 in a video sequence

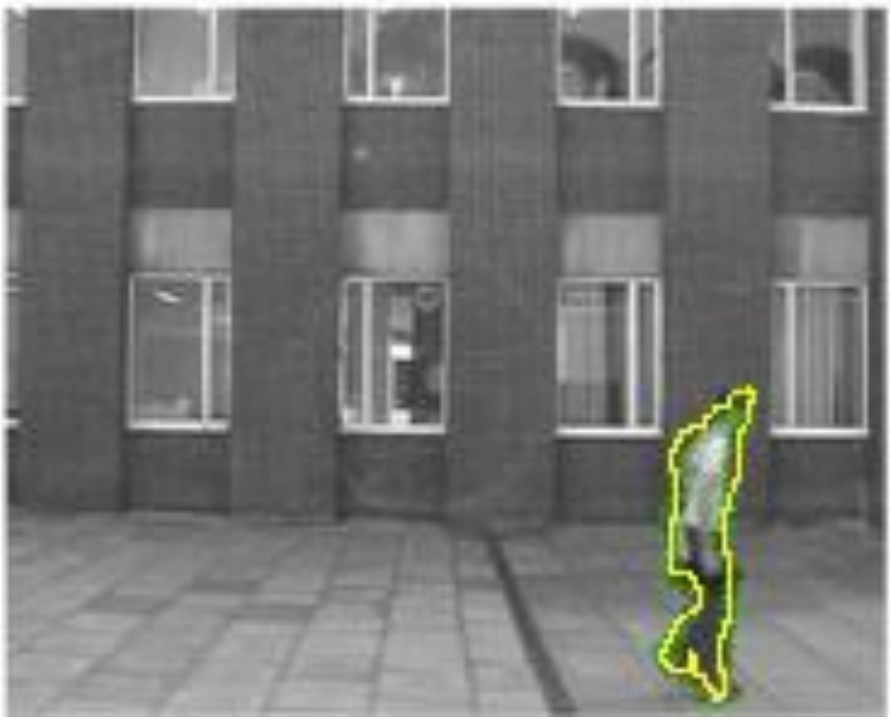

Figure 2.13: Snake progression on moving object at Frame 30 in a video sequence

Thus, the modified Snake progresses to follow the object deformation and motion. It then stops and appends onto the same points when the object ceases to move.

2.6.3 Overall design flowchart of the enhanced active contour framework

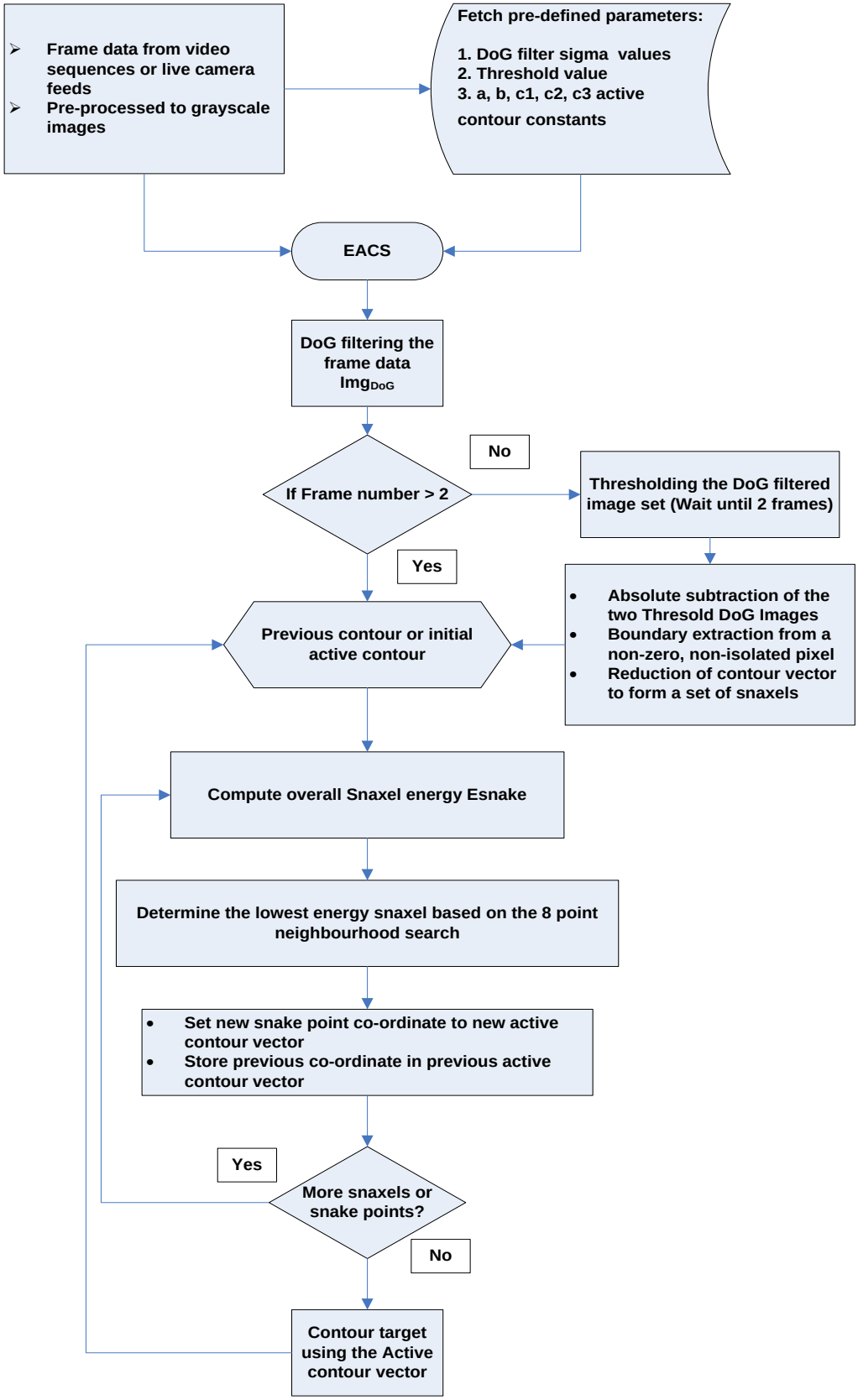

Figure 2.14: Flowchart of the enhanced active contour snake (EACS) framework

Figure 2.14 depicts the overall flow diagram of the enhanced active contour snake technique discussed so far in the previous sections. The algorithm progresses by fetching the user defined active contour parameters and greyscale frame data from a video source. For the first two frames, the initial contour is extracted by using the automatic initial contour extraction technique discussed in Section 2.5. After the initial active contour vector is obtained the snake deformation and progression takes places based on the energy minimisation condition by computing the energy for each of the snaxels or snake points in the contour vector according to the equation 2.6. The previous active contour vector is then utilised in the next frame as the starting contour vector for snake deformation and progression in order to lock on to the moving target. Due to the automatic initial contour extraction and the use of DoG filtered bipolar images for energy minimisation, the active contour technique is enhanced and so can be used in real time robust target tracking applications.

2.7 Results and discussion

A few more examples of colour video sequences are shown below in Figures 2.15 to Figure 2.18 to describe the enhanced active contour snake technique discussed so far. For colour video sequences it is observed that a higher threshold value ranging between 140 and 255 is suitable for accurate initial contour extraction. The patch size of the DoG filter designed for the colour band videos is considerably smaller as compared to infra-red or near infra-red band video sequences. The colour frames [44] are converted to greyscale data before the application of the enhanced active contour technique. Figure 2.15 shows the frame before the snake deformation is performed. A subsequent frame showing the deformed snake on the moving target is shown in Figure 2.16. The snake progression and deformation along with the active contour vector shrinkage due to scale change of the object is shown in Figure 2.17 and Figure 2.18.

Figure 2.15: Frame 1 of the colour video sequence (before deformation)[44]

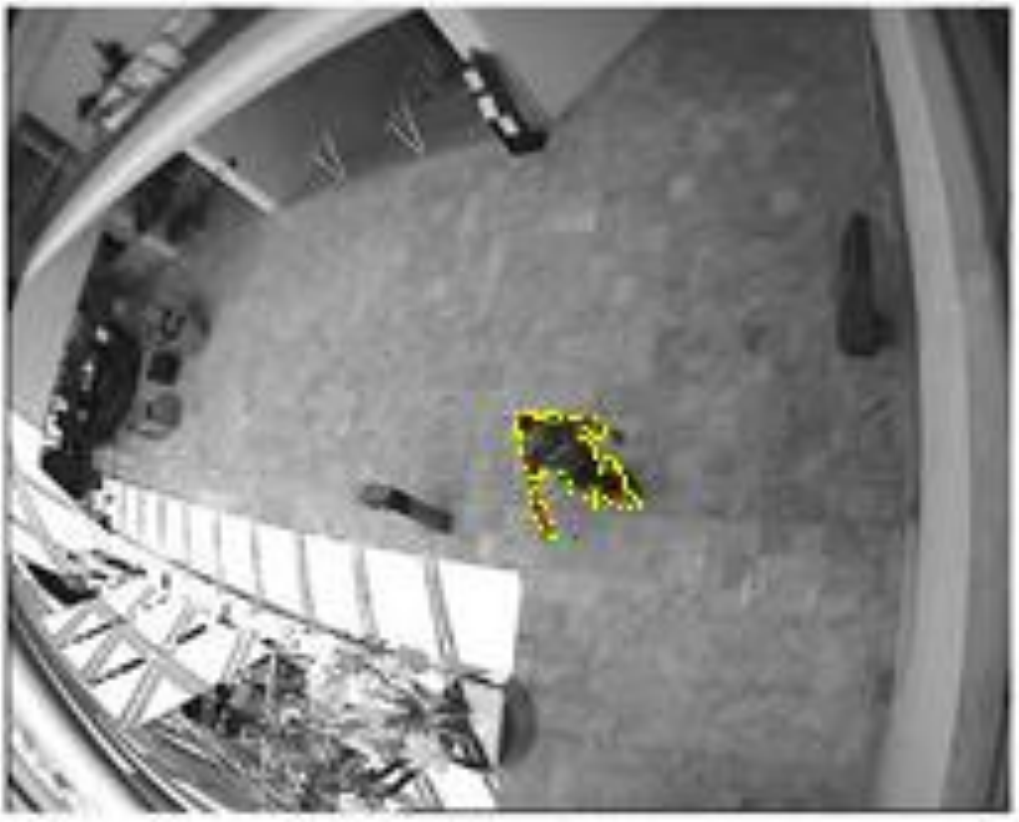

Figure 2.16: frame number 3 after deformation of the enhanced AC

Figure 2.17: Frame 60 of the video sequence after deformation and progression

Figure 2.18: Frame 103 of the colour video sequence continuously tracked even after the object scale changes

In a further example, frames from an infrared traffic video are shown in Figures 2.19 and 2.20, illustrating the tracking of a motor vehicle. The threshold value pre-defined for the initial active contour computation is considerably low (in the range of 50 to 150) for infrared video sequences. Figure 2.19 shows the snake deformation on target at the 10th frame and Figure 2.20 shows the snake deformation and continuous tracking of the target at the 22nd frame in the video sequence.

Figure 2.19: Enhanced active contour snake on infrared video at the 10th frame showing the snake deformation on target

Figure 2.20: Enhanced active contour on target in infrared video sequence, depicting continuous tracking and progression in the 22nd frame

From Figure 2.19 and Figure 2.20 it can be observed that the size of the contour vector and the total number of snaxels change dynamically for the infrared band video sequences. This is due to the change in intensity values because of varying temperatures in the moving vehicle being tracked. Note that the images in Figure 2.19 and Figure 2.20 are from a static camera with a constant view angle.

2.8 Summary

This chapter introduces a modified model for active contours which is fast, robust and accurate as compared to other available methods for Active Contours or Snake models. The described method is based on DoG filtered images thus giving more emphasis to the edges of the object under study. The DoG filter is designed using a pre-defined set of parameters such as the standard deviation values, obeying the rule that their ratio be

equal to 1.6, as explained in Section 2.3. This method can either be used in an image sequence or on frame data from a continuous video source. The active contours are set as soon as the object in a video is in motion and the contour follows the object by deforming onto the object in each frame, thus providing robust tracking of the acquired object in the video. The introduction of the new energy functional term derived from the DoG filtered image, as explained in Section 2.4, reduces the snake energy to give emphasis to the object edges, thus making the snake deformation faster and more robust.

The method of automatic contour initialization makes it possible for the algorithm to work on video sequences with limited user intervention, unlike previously presented methods [9], [11], [19], [15], [13]. The initial contour set, based on moving objects and the boundary data of the object, makes the snake fitting and deformation quicker than the other methods previously reported in the literature. Due to the initialization on the edges of the object, the snake requires less iteration for the first frame the object is acquired, thus reducing the processing time. This not only reduces processing time but also makes it accurate, as the initial contour fits readily onto the object. The progression of the snake onto future frames is also made accurate and fast due to the presence of the DoG energy parameter in the minimization equation. The method also works when an object being tracked stops moving and becomes static in the scene, as discussed in Section 2.6. Addition or removal of contour coordinates into the AC allows the contour to fit onto the moving object precisely even when the scale and orientation of the object changes subsequently. The enhanced active contour discussed in this chapter works on a single moving object and can be further enhanced to work for multiple object tracking and contouring applications.

The results and discussion in Section 2.7 shows that the enhanced active contour snake (EACS) framework performs accurately in real-time for variable scale and orientations of the object being contoured. It is also evident that the enhanced active contour technique not only works on colour video sequences but also on infrared band sequences. The computational model for single object contouring can be modified and improved for multiple objects contouring and tracking. This is discussed in detail in Chapter 3 where it will be explained how multiple contours can be constructed out of a single contour vector deformed on to the moving objects in video sequences.

Chapter 3

ENHANCED ACTIVE CONTOUR SNAKES FOR MULTIPLE OBJECT TRACKING AND ITS APPLICATIONS

3.1 Introduction

Many advanced computer vision tasks such as recognition, tracking and scene understanding require accurate tracking of the complete contour of the objects. It is often observed in real-time scenarios that the objects undergo dynamic deformations and occlusions. It is common for the deformable objects to encounter crossing trajectories. The non-rigid objects often get occluded causing a merger of their contours into a single contour vector. Classical active contour methods fail to track multiple objects at once [2], [3], [7], [8], [22]. Tracking non-rigid objects in dynamic scenarios is a challenging task as the objects split and merge when they come closer and move apart. The extended computational model proposed in this chapter enables topographical change to the contour vector contouring the target objects. Contour merging and splitting operations are performed when the objects come closer and move apart, respectively. The proposed topographical change computational model also increases the tolerance to noise due to high frequency edge intensity changes in dynamic scenes.

Enhanced active contour snake based statistical measurements such as the contour bounding rectangle and contour centroid are discussed in detail. Contour bounding rectangle is a measure of overall space occupied by the object in the frame. Contour boundary rectangle measure assists in computing the centroid of the contour. The

centroid of the active contour can be utilised to integrate the enhanced active contour system with several target tracking techniques to improve the efficiency of tracking in difficult vision conditions. This has been discussed in Chapter 5 along with promising results.

Based on the active contour centroid measure a real-time upper body skeletal modelling application has been designed. An upper body stick skeleton model is derived from the enhanced active contour vector and its centroid. This can be utilised in estimating the pose of the human body and to track salient body parts of the humanoid being contoured.

3.2 Chapter organisation

The entire chapter is organised in the following way: the extension to the enhanced active contour snake (EACS) model for multiple object tracking is discussed in detail along with a few results in the following section. The Section 3.3 also describes the snake termination, separation, merger and constraint conditions using a detailed mathematical model. The statistical measures of the enhanced active contour model are detailed in Section 3.4. The active contour bounding rectangle and the AC centroid computations are discussed and demonstrated using several examples. The upper body humanoid skeletal modelling application is described in Section 3.5 along with some results. Conclusions are drawn in the summary Section 3.6.

3.3 Multiple object contouring

The active contour model described in the previous chapter is only able to track a single moving object. There is a need to extend the enhanced active contour technique to work for multiple moving objects in dynamic scenarios. In order to accomplish multiple object tracking, the enhanced active contour needs to be improved to differentiate

between multiple objects interacting and moving apart from each other. Multiple object tracking can only be possible if the contour can be split into multiple contours or merged into a single contour depending on the interaction between the objects in motion [29], [18].

Extension to the enhanced active contour model is proposed to solve the problems described above. A contour process termination step is introduced to stop the snaxels from moving after a stable lower minimum energy contour vector is attained. The contour splitting step is also introduced in the following section describing the conditions and procedure to divide the snake into several contours. A contour selection step is also described to eliminate irrelevant contour vectors after the contour is split into separate contour vectors. These extensions to the EACS assist in tracking multiple moving objects at the same time. These extension steps are only performed after the final active contour vector is obtained using the EACS model [18], [20], [19].

3.3.1 Contour termination

The active contour deformation process dynamically changes the position of each snaxel contained in the contour until the lowest energy point for all the snaxels is obtained. This process is often computationally intensive. The constraint criteria for stopping the snake deformation can be included to improve the efficiency by decreasing the computational cost. This can be achieved by adhering to the criterion of maintaining a constant distance equal to the average distance $D_{average}$ between each neighbouring snaxel in a contour vector [26], [34], [18].

3.3.2 Contour splitting step

The active contour can be divided into several contours by evaluating the length of every segment v_i-v_{i+1} connecting two successive snaxel points v_i and v_{i+1} contained in

the contour vector. If the objects move apart from each other, the length of the segment increases. A segment length threshold can be computed as the product of a fixed coefficient with the average distance $D_{average}$ between each neighbouring snaxel pair forming a segment. If the length of a segment in the contour vector is greater than the threshold, the segment is divided by introducing new contour points (snaxels) between the two consecutive snaxels. The segments whose lengths are greater than the threshold is divided into three equal segments by introducing two intermediary points between them (say v_i and v_{i+1}) [18]. An initial contour considered for splitting is depicted as shown in Figure 3.1.

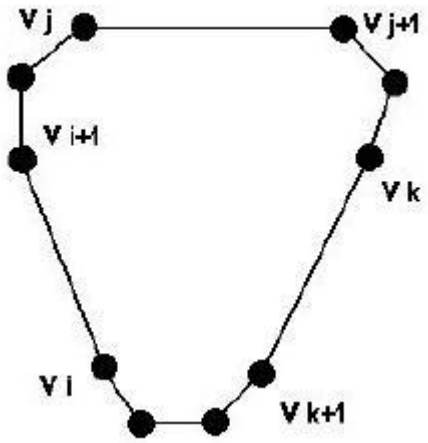

Figure 3.1 Contour '*v*' analysed for segments before splitting [18]

The segments to be divided, due to their lengths being greater than the threshold, are identified as shown in Figure 3.2.

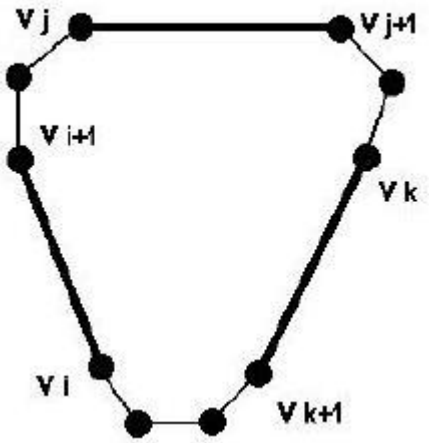

Figure 3.2 Identified contour segments to be divided are highlighted [18]

Points p_i^1 and p_i^2 denote the two intermediate points between the snaxels v_i and v_{i+1}. Three new segments, namely v_i- p_i^1, p_i^1- p_i^2 and p_i^2- v_{i+1} are appended into the contour *'v'*. This can be illustrated as shown in Figure 3.3.

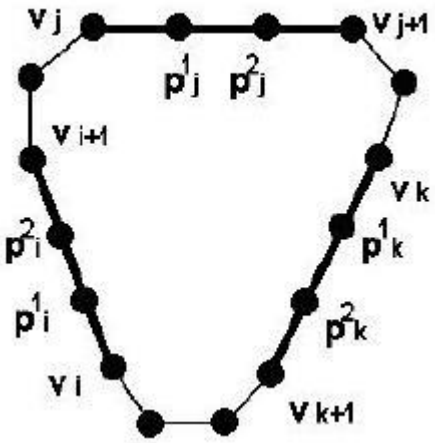

Figure 3.3. The segments identified that need splitting; two new points are appended to create three new segments

If the lengths of segments keep increasing over the defined threshold, the contour vector can then be divided into separate closed contours at the new segments created. This is illustrated in Figure 3.4 [19], [18].

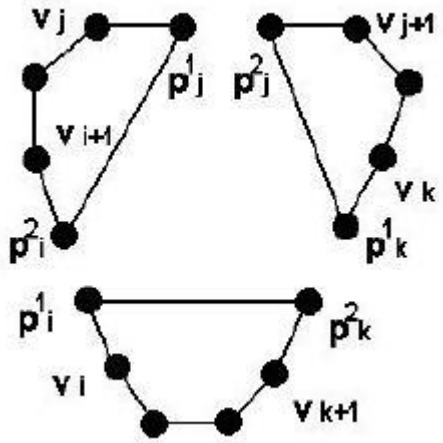

Figure 3.4 New set of segments reorganised to form three separate active contours

The active contour snake is split into three separate contours and will now contain the additional set of segments as described in the contour vector segment equations 3.1, 3.2 and 3.3:

$$\mathrm{X}_1 = \left[v_i \, p_i^1 - p_i^1 \, p_k^2 - p_k^2 v_{k+1} \right] \tag{3.1}$$

$$\mathrm{X}_2 = \left[v_j \, p_j^1 - p_j^1 \, p_i^2 - p_i^2 v_{i+1} \right] \tag{3.2}$$

$$\mathrm{X}_3 = \left[v_k \, p_k^1 - p_k^1 \, p_j^2 - p_j^2 v_{j+1} \right] \tag{3.3}$$

The three separate active contour vectors, X_1, X_2 and X_3 are formed due to the contour splitting operation when the segment lengths calculated are higher than the defined threshold. Figure 3.4 illustrates the three separate contours produced by rearranging the new set of segments created from the original active contour vector '*v*' [22], [19].

3.3.3 Contour selection step

Distinct active contour vectors are obtained by performing contour splitting on a single active contour vector. This is followed by a contour selection step to identify the

relevant active contour vectors from the new set of contour vectors created during the splitting process, as explained in the previous section. The new sets of contours are size filtered to decrease the number of false contour vectors fitting into the noisy background pixels. Inter-snaxel coherence is verified for each sub-set of contour vectors, isolated snaxels forming a separate contour vector containing a point or a line being discarded. The resultant sub-set of contour vectors so obtained are maintained in a separate contour array for each moving object being contoured [21].

Figure 3.5 and Figure 3.6 illustrate an example of contour vector splitting and the forming two distinct active contour vectors. An active contour vector that locked on to a human being carrying a box is as shown in Figure 3.5.

Figure 3.5 Contour vector locked onto the human holding a box (drawn in green, signifying the original contour vector considered for subsequent splitting).

The original contour vector shown in Figure 3.5 is split due to the objects (human and box) separating themselves by a distance in subsequent frames. After the contour splitting and relevant contour selection process, the two distinct contour vectors contouring the human and the box, respectively, are depicted as shown in Figure 3.6.

Figure 3.6. Two distinct contour vectors created from the single contour present in previous frames as shown in Figure 3.5

In Figure 3.6 note that the contour drawn in green signifies the original contour and the contour drawn in yellow signifies the new contour vector created. The yellow coloured contour is affixed onto the static box and will remain contouring the box until it is displaced due to the contour process termination criteria, as discussed in Section 3.3.1.

Another example of contour splitting, where the objects continue their motion in the scene, is illustrated in Figure 3.7 and Figure 3.8. The objects being considered here are

two human targets, occluding each other and forming a single active contour vector as shown in Figure 3.7.

Figure 3.7. Single contour vector drawn in green covering two people occluding each other in a scene

The contoured objects subsequently move apart and continue moving along their respective trajectories. The single contour is now split into two distinct contours when the two people contoured move apart. It is followed by the contour selection process to minimise the possibility of assigning false contours to track the people. After splitting into two distinct contour vectors, each person being contoured moves in his own direction. The enhanced active contour snake (EACS) continues to track the objects separately, maintaining a distinct contour vector for each moving object in the scene. The split contours of each person in the image in Figure 3.7 are shown in Figure 3.8.

Figure 3.8. Two distinct contour vectors created from the single contour present in the previous frames as shown in Figure 3.7

The new contour vector created is marked in yellow to signify that the contour will now track a new target moving in a different direction.

3.4 Statistical measures of the enhanced active contour snakes (EACS)

3.4.1 Active contour bounding rectangle

The enhanced active contour vector obtained from an object being tracked can be used to compute a minimum area bounding rectangle covering the circumference of the contour vector. Mathematically, the active contour vector can be scanned to find the highest and lowest x-axis and y-axis coordinate values. This can be done by considering one axis point vector at a time. Thus, by considering the x-axis coordinate point set of the snaxel array, the minimum and maximum x-axis value can be computed as described in equations 3.4 and 3.5. A set of snaxels S_i is considered where 'i' ranges

between 1 and *N*; *N* being the total number of snaxels in the contour vector. The maximum x-axis coordinate X_{max} and the minimum x-axis coordinate X_{min} can be derived as the shown in equations 3.4 and 3.5, respectively:

$$X_{\max} = \max(S_i(X_i)), \forall (i = 1:N) \tag{3.4}$$

$$X_{\min} = \min(S_i(X_i)), \forall (i = 1:N) \tag{3.5}$$

where, *max(.)* is the function scanning for the maximum value between all the x-axis coordinates of the snaxel array. The *min(.)* function scans for the minimum value out of all the x-axis coordinates of the snaxel array [44], [32].

Note that the symbol '$\forall$' is used to signify a 'for each' condition, which means the maximum or minimum value is searched throughout the contour vector. Similarly, the maximum y-axis coordinate Y_{max} and the minimum y-axis coordinate Y_{min} are computed using equations 3.6 and 3.7:

$$Y_{\max} = \max(S_i(Y_i)), \forall (i = 1:N) \tag{3.6}$$

$$Y_{\min} = \min(S_i(Y_i)), \forall (i = 1:N) \tag{3.7}$$

Using the values obtained from equations 3.4 to 3.7, the bounding rectangle coordinates can be obtained as described by equation 3.8. The minimum x-axis and y-axis coordinate point is considered as the top-left corner of the bounding rectangle. The bottom-right corner of the bounding rectangle is the point described by the maximum x-axis and y-axis coordinate points. If the bounding rectangle is denoted as *B*, then the top-left and bottom-right coordinate of the bounding rectangle can be computed from equations 3.8 and 3.9, respectively:

$$B_{top-left}(x,y) = (X_{\min}, Y_{\min}) \tag{3.8}$$

$$B_{bottom-right}(x,y) = (X_{\max}, Y_{\max}) \tag{3.9}$$

From equation 3.4 to 3.7, the width and height of the bounding rectangle can be computed as described in equations 3.10 and 3.11:

$$B_{width} = (X_{\max} - X_{\min}) \tag{3.10}$$

$$B_{height} = (Y_{\max} - Y_{\min}) \tag{3.11}$$

The minimum up-right bounding rectangle of an active contour vector can be deduced using the above equations. The EACS contouring a person is shown in Figure 3.9.

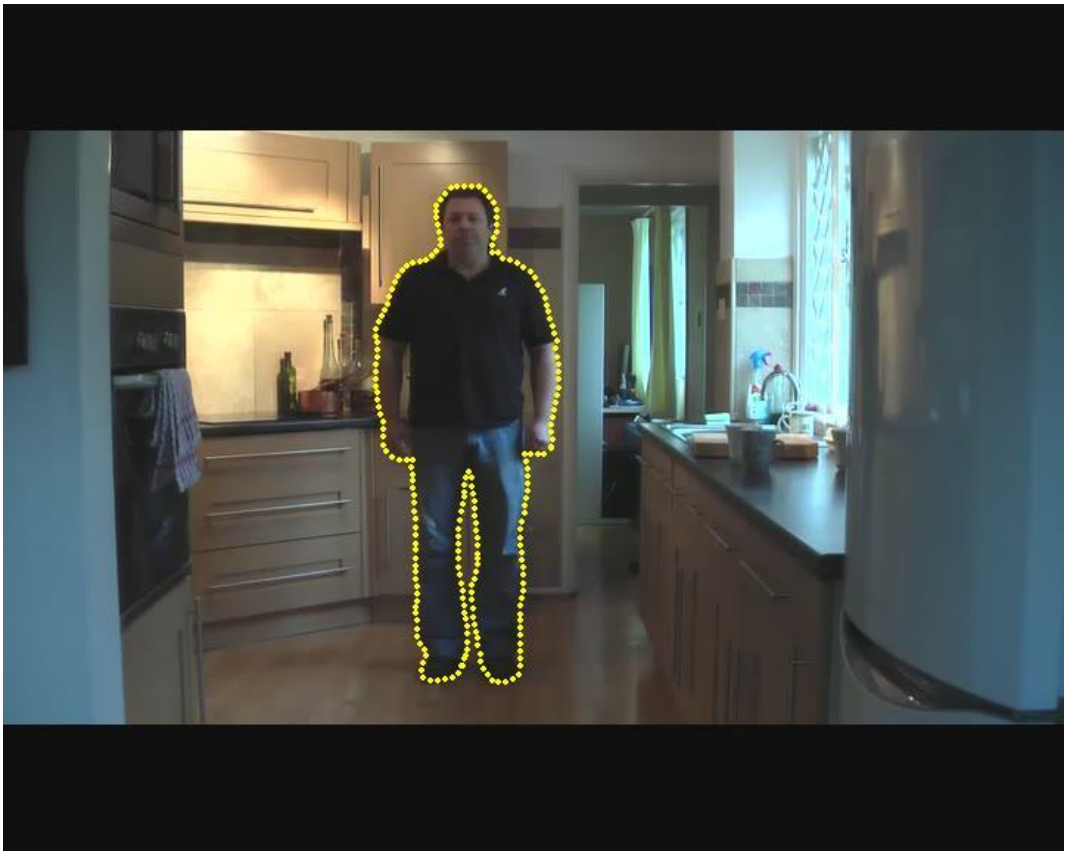

Figure 3.9. The active contour vector plotted on the object being contoured

The minimum bounding rectangle of the contour vector is computed using equations 3.8 and 3.10 and it is drawn in red as illustrated in Figure 3.10.

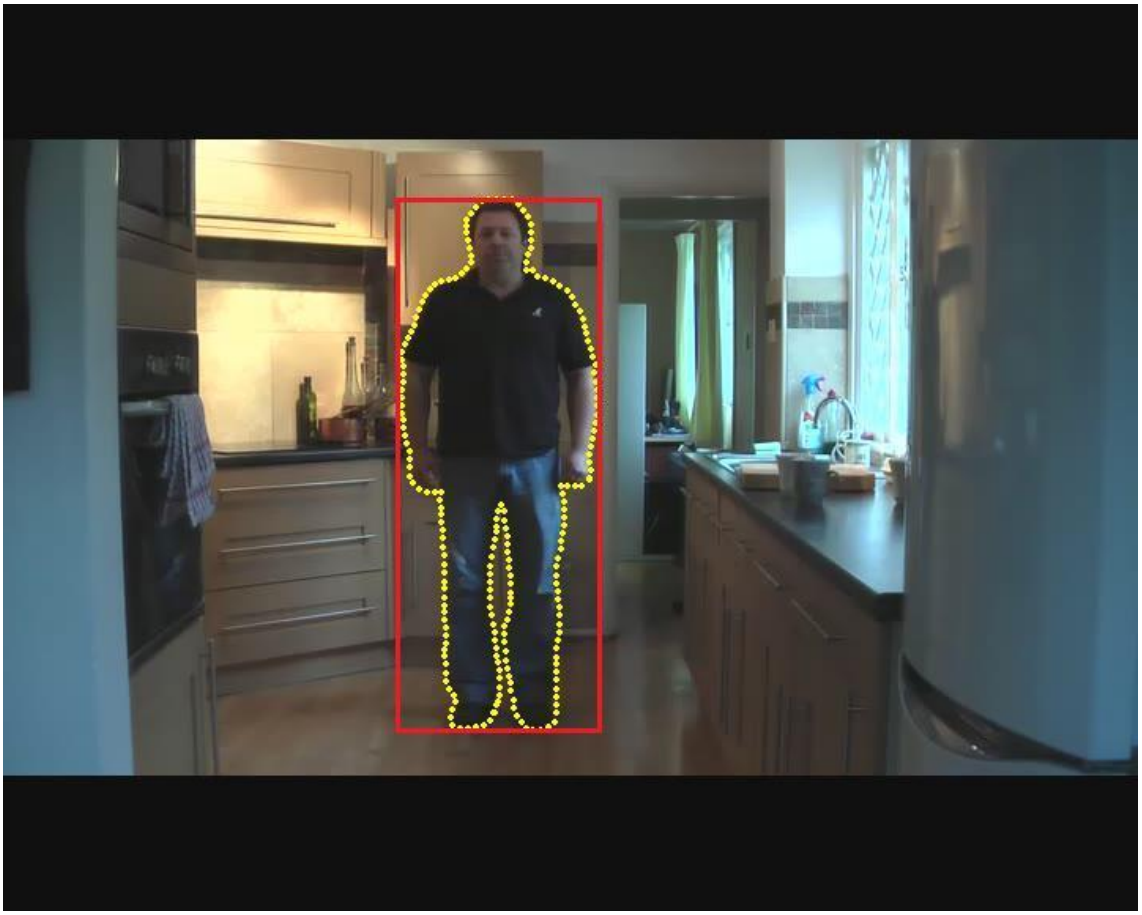

Figure 3.10 The minimum bounding rectangle bounding the contour vector

3.4.2 Active contour centroid

The coordinate of the active contour centroid can be computed using the AC bounding rectangle coordinate values as computed in equations 3.8 to 3.11. If the centroid of the active contour vector is denoted by C_{AC} it can expressed as in equation 3.12:

$$C_{AC}(x,y)=\left[\left(B_{top-left}(x)+\frac{B_{width}}{2}\right),\ \left(B_{top-left}(y)+\frac{B_{height}}{2}\right)\right] \tag{3.12}$$

The centroid is computed for the bounding rectangle as depicted in Figure 3.10 and plotted in green to display the computed centroid point from the active contour vector

drawn in yellow, as illustrated in Figure 3.11. The centroid point plotted in green is used as a centre to draw a small circle in yellow for display purposes.

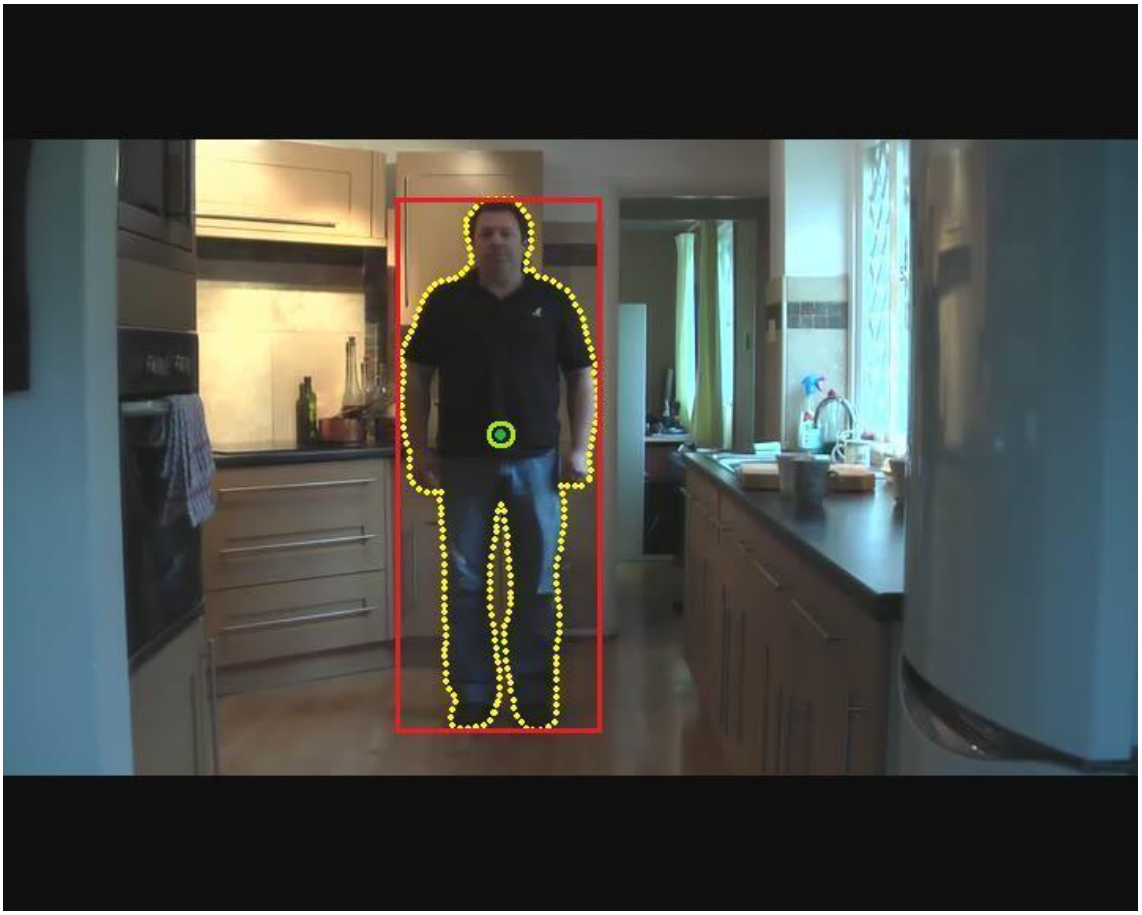

Figure 3.11 Active contour centroid point plotted in green surrounded by a yellow circle for display

3.5 Upper body humanoid skeletal modelling application

Human pose estimation has been an important research problem in computer vision applications. The active contour statistical parameters discussed in Section 3.4 can be utilised in constructing a skeletal stick model for the humanoid being contoured. A simple humanoid skeletal modelling application for pose estimation of the frontal pose upper body is designed and discussed in detail in this section [66].

The active contour centroid is the important parameter considered while deducing the skeletal model for the contoured humanoid. The EACS contour vector computed on the humanoid is illustrated in Figure 3.12.

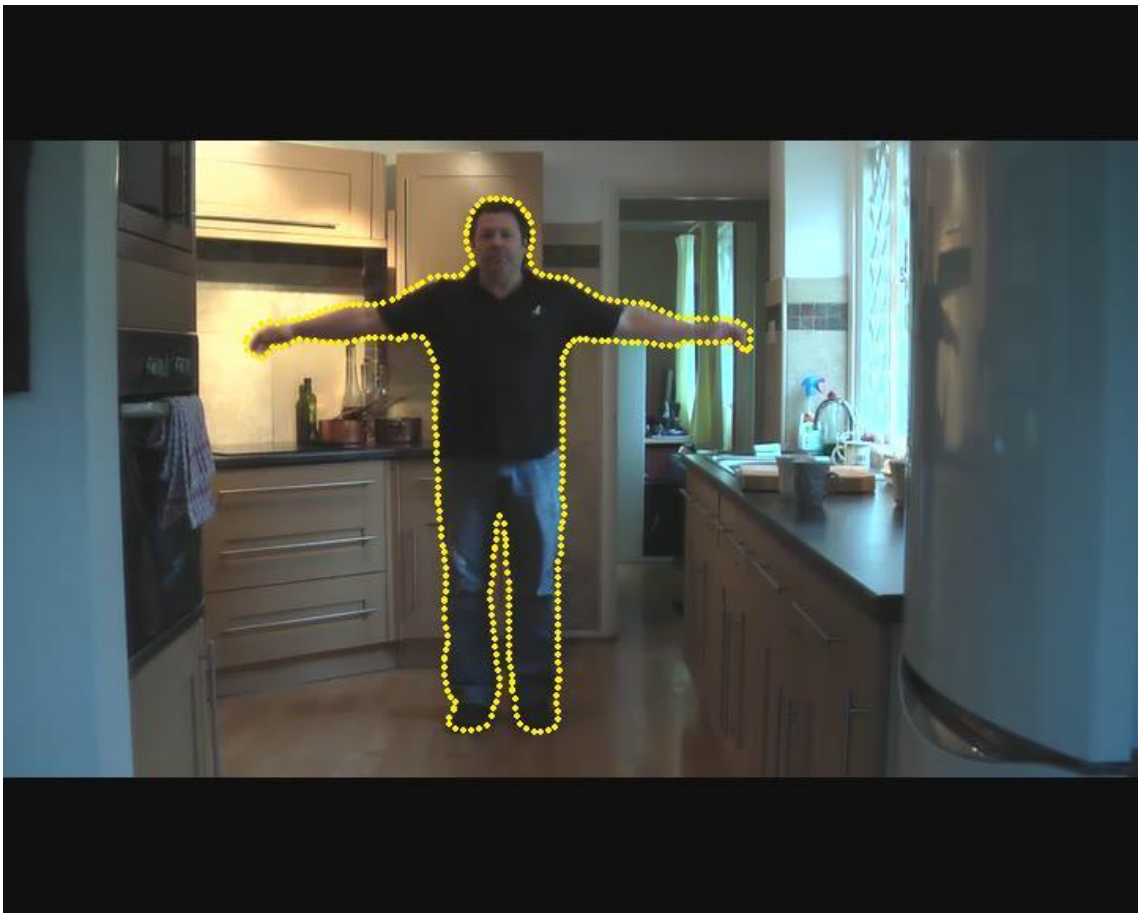

Figure 3.12 The EACS contour vector drawn in yellow for the humanoid being contoured

The active contour bounding rectangle and the centroid are computed for the contour vector shown in Figure 3.12. This is illustrated in Figure 3.13 with the centroid point plotted in green with a yellow circle around it for display purposes. The bounding rectangle is drawn in red as shown in Figure 3.13.

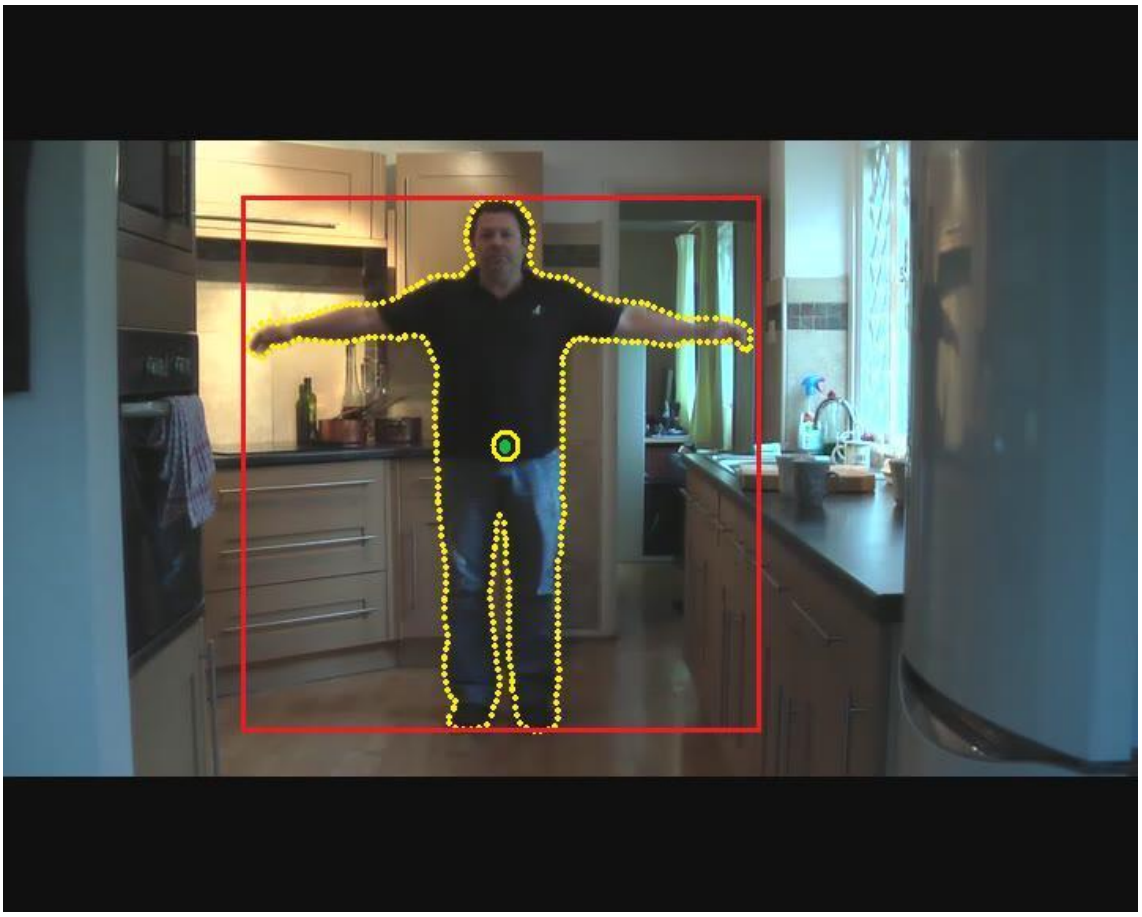

Figure 3.13 Active contour centroid point computed from the EACS contour, plotted in green surrounded by a yellow circle for display

The active contour centroid is one of the salient skeletal points of the humanoid. The location of the AC centroid can be used to deduce a stick model for the frontal pose estimation of the humanoid under consideration.

Using the active contour bounding rectangle and the centroid data, it is possible to compute the top most point on the head of the humanoid denoted as $Skel_{head}$. This can be computed as expressed in equation 3.13 (considering also equations 3.8 and 3.12):

$$Skel_{head}(x, y) = \{C_{AC}(x), B_{top-left}(y)\} \tag{3.13}$$

A line is drawn between the active contour centroid to the $Skel_{head}$ (top-most point of the head) forming the upper body skeletal stick model. This is depicted in Figure 3.14.

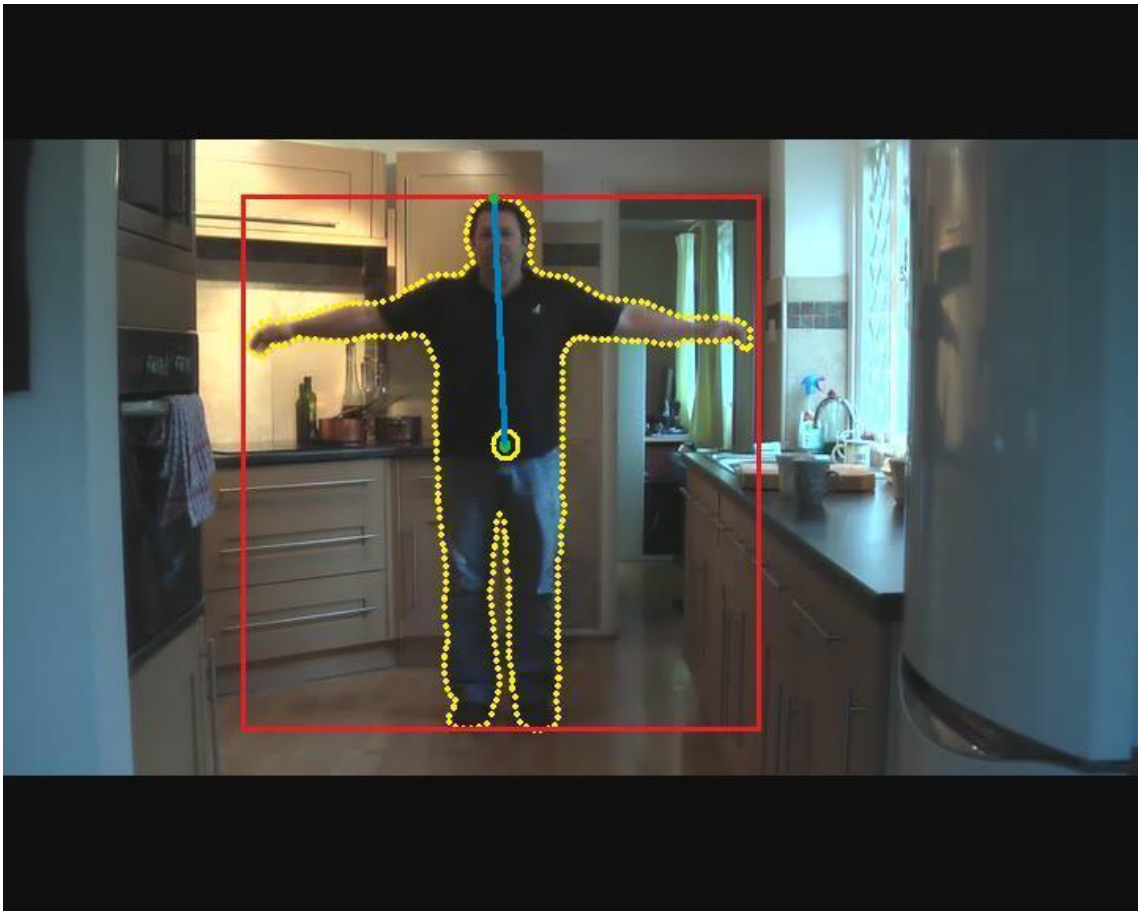

Figure 3.14 The upper body skeletal stick drawn between the centroid and the top most point of the head determined using equation 3.13

The anthropometric data from NASA's anthropometric data source book [65],[66] used to estimate the size of body parts is utilised to compute the neck (or the centre of the shoulder) point of the humanoid. According to the data the neck point of the humanoid should lie at a point at one-third the distance from the top of the head to the centroid of the body. The distance calculated here is the Euclidian distance [62] (denoted as $E_{euclidian}$) that can be computed between any two points given by the expression in equation 3.14:

$$D_{euclidian} = \sqrt{\left(C_{AC}(x) - Skel_{head}(x)\right)^2 + \left(C_{AC}(y) - Skel_{head}(y)\right)^2} \quad (3.14)$$

where the active contour centroid coordinate C_{AC} and $Skel_{head}$ are given by equation 3.12 and 3.13, respectively. The Euclidian distance computed is used to find the neck point at one-third the distance from the top most point of the head. This has been illustrated using a yellow circle at the computed neck point for display as shown in Figure 3.15.

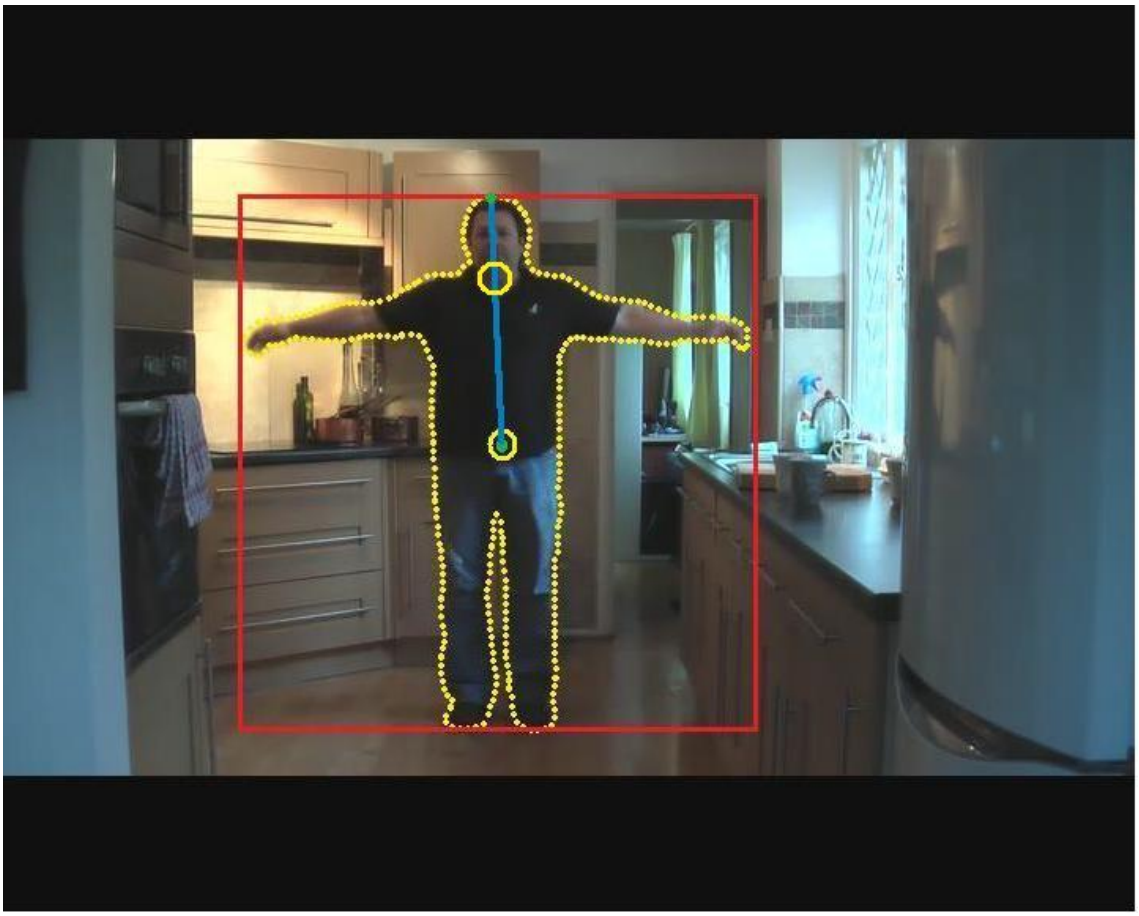

Figure 3.15 The center of the neck point deduced using the Euclidian distance between the centroid and the top of the head in the humanoid being contoured

Using the centre of the neck point the left most and right most points of the arms can be derived by simple linear scanning inside the active contour vector surrounded by the bounding rectangle. The linear scanning is performed by calculating the left most and right most points of the active contour vector touching the bounding rectangle. The lines

drawn between the neck point to the tip of both the arms represent the upper body skeletal stick model of the humanoid being contoured. The computed left most and right most points on the left and right arms connected to the neck point form the upper body humanoid skeletal stick diagram as illustrated in Figure 3.16.

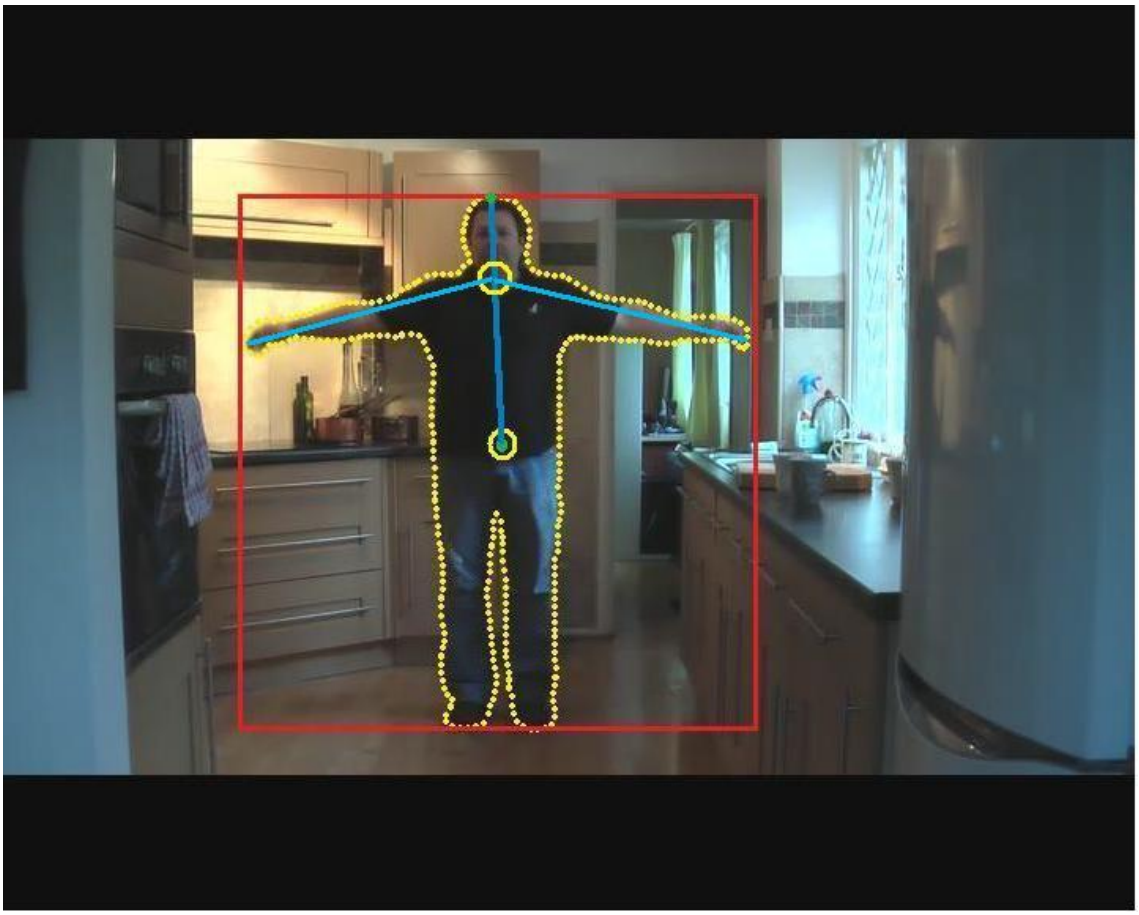

Figure 3.16 The upper body humanoid skeletal stick model derived from the active contour centroid and AC bounding rectangle parameters

The upper body humanoid skeletal modelling application is only designed to demonstrate the use of active contour statistical parameters. It is not a complete human skeletal tracking solution. The upper body humanoid skeletal modelling application can be further enhanced to perform as a full body skeletal tracking system by utilising

several geometric computations. The future work involved in improving the skeletal modelling application is discussed in Chapter 6 of the thesis.

3.6 Summary

An extension to the enhanced active contour snake (EACS) technique to perform multiple object tracking has been proposed in this chapter. The contouring process termination criterion has been described in order to enhance the computational efficiency. It is demonstrated that the active contour vector tracking a group of occluded moving objects can be divided into distinct contour vectors that can maintain a separate track of each moving object in the scene. The contour splitting and selection criteria responsible for causing the active contour to split and then maintain distinct contour vectors for multiple objects are discussed in Sections 3.3.2 and 3.3.3. Several results incorporating the contour splitting and selection process for tracking multiple objects have been discussed. From the results documented, it is clear that the enhanced active contour model is also suitable for multiple object tracking in real-time applications.

EACS statistical parameters, such as the active contour boundary rectangle and the active contour centroid, have been described in detail in Section 3.4. Several results have been discussed to demonstrate the use of these statistical measures. A simple application to perform upper body humanoid skeletal modelling has been designed and discussed to demonstrate the importance of computing an accurate active contour vector using the EACS. Several results determining the skeletal stick model for the upper body of a humanoid have been presented and discussed. How the enhanced active contour framework can be utilised to derive an important statistical measure known as the global active contour orientation (GACO) is discussed in the next chapter and a description of an application for a robust door surveillance system is presented.

Chapter 4

GLOBAL ACTIVE CONTOUR ORIENTATION AND ITS APPLICATIONS

4.1 Introduction

A popular approach for estimating the direction of motion of moving objects in a scene is the application of optical flow techniques [14], [16]. These focus on the observation of the differential changes in pixel values. The optical flow approach has been used in automatic gait recognition amongst many other applications with a static background and a single object in motion [6]. In real time applications such as estimating the direction of flow of a selected object in motion as opposed to the overall flow of motion in a dynamic scene, the optical flow approach becomes unsuitable. This is partly due to the fact that optical flow methods consider the differential changes in the pixel values of the whole frame.

Enhanced active contour techniques discussed so far can be utilised for locking on to a moving object of interest. The active contour vector comprising a finite number of snake points or snaxels for estimating the direction of motion is a better approach for computing the localised motion and orientation of an object in a frame. The orientation of the active contour vector can be computed for each contour co-ordinate or snaxels using simple geometry. This chapter introduces the global active contour orientation (GACO) technique for statistical measurement of contoured object orientation. It is an overall active contour orientation measurement method which uses the proposed

enhanced active contour model along with statistical measurement techniques. The computation of global active contour orientation angle is discussed in detail.

A real-time door surveillance application based on the global active contour orientation has been developed and discussed to evaluate the technique for its application in determining the contoured object orientation in a localised region of the frame. The door surveillance application developed is evaluated on the i-LIDS (an acronym for the 'Imagery library for intelligent detection systems') door surveillance dataset provided by the UK Home Office [43]. The door surveillance application uses an edge-map based Harris corner detector in a localised area of the door for generating an alarm. Furthermore, the system accuracy is enhanced by using a Hough transform based error correction technique. The performance results demonstrate the use of GACO to evaluate the door surveillance dataset gives an overall success rate of 92%, thus proving the proposed GACO technique to be useful in real-time applications.

4.2 Chapter organisation

The entire chapter is organised in the following way: the global active contour orientation (GACO) and the statistical measurement of the contoured object orientation is discussed in detail in the following section. The Section 4.3 also describes the radian angle computations for each defined quadrant of the image. The following Section 4.4 describes the overall design flowchart for the GACO technique along with a few results. The door surveillance system developed is introduced in Section 4.5.1. The following sub-section describes the edge-map based Harris corner detector for primary surveillance alarm generation. Section 4.5.3 describes the door surveillance system algorithm design along with an overall flowchart of the system. A system to perform error correction using Hough transforms is discussed in the Section 4.5.4. Finally, Section 4.6 presents several results of the GACO based door surveillance system. The

section also discusses the i-LIDS dataset evaluation results along with a result table estimating the detection rate and the probability of genuine alarms. Conclusions are drawn in the summary Section 4.7.

4.3 Global active contour orientation measurement

The enhanced active contour snake model discussed in Chapter 2 is utilised in contouring an object in motion in order to compute its global active contour orientation. The energy of each coordinate point is calculated based on the 8- point neighbourhood pixels at each snaxel. A difference of Gaussian (DoG) filtered image of the frame is computed to emphasise the exterior edges of the object being contoured. The process is maintained and controlled by the number of iterations throughout the contour vector. The enhanced active contour vector determined is stored in a snaxel vector array after the (EACS) snake deformation. The total number of snaxels, *N*, contained in the snaxel vector is also stored for geometric computations.

The orientation of the contour vector can be computed for each contour coordinate point or snaxel using simple geometric calculations. The overall contour orientation is then computed as the mean of all the snaxel or vector point orientations. The calculations are carried out based on the defined quadrant of the image in which the coordinate point lies. The orientation of each point in the contour denoted as $\theta_i(x, y)$ can be calculated by equation 4.1, if $x_i \neq x^*_i$:

$$\theta_i(x,y) = \tan^{-1}\left(\frac{y_i - y^*_i}{x_i - x^*_i}\right) \tag{4.1}$$

where x^*_i and y^*_i denote the previous coordinate position of the ith coordinate point. The x_i and y_i denote the current coordinate of the ith snaxel.

If $x_i = x^*_i$, $\theta_i(x, y)$ is set to '$\pi/2$' or '$3\pi/2$' radians depending on the y-coordinate points. This can be expressed:

$$\theta_i(x, y) = \frac{\pi}{2} \subset \left(y^*_i > y_i\right) \tag{4.2}$$

$$\theta_i(x, y) = \frac{3\pi}{2} \subset \left(y^*_i < y_i\right) \tag{4.3}$$

If $y_i = y^*_i$, $\theta_i(x, y)$ is set to '0' or 'π' radians depending on the x-coordinate points. This can be expressed as:

$$\theta_i(x, y) = \pi \subset \left(x^*_i > x_i\right) \tag{4.4}$$

$$\theta_i(x, y) = 0 \subset \left(x^*_i < x_i\right) \tag{4.5}$$

When $x^*_i > x_i$ the angle $\theta_i(x, y)$ is computed based on equation 4.1 with additional calculations given by;

$$\theta_i(x, y) = \left[\frac{\pi}{2.0 - \theta_i(x, y)}\right] \subset \left(x^*_i > x_i\right) \tag{4.6}$$

If $x^*_i < x_i$ the angle $\theta_i(x, y)$ is computed based on equation 4.1 with additional calculation given by:

$$\theta_i(x, y) = \left[(\pi \times 1.5) - \theta_i(x, y)\right] \subset \left(x^*_i < x_i\right) \tag{4.7}$$

Note that the symbol $\subset$ in equations 4.2 to equation 4.7 denotes the 'if' condition described within the bracket.

The overall orientation of the contour, also known as the active contour global orientation, is then computed as the mean of all the angles $\theta_i(x, y)$ where 'i' ranges between 1 to *N*.

This can be described by the expression:

$$\theta_c = \frac{1}{N}\sum_{i=1}^{N}\theta_i(x, y) \tag{4.8}$$

Here, *N* denotes the total number of snaxels or contour points contained in the active contour of the object being considered [38], [45].

4.4 Global active contour orientation design flowchart

The radian angle can be converted to degrees using equation 4.9. Figure 4.1 shows a typical example of the global active contour orientated at 345^0, computed on a contoured moving object.

$$\theta_c{}^0 = \theta_c \times \frac{180}{\pi} \tag{4.9}$$

Figure 4.1: Global active contour of the object contoured in blue with AC vector (345^0)

The contour vector of the previous frame is stored and used for computing the global AC orientation in the current frame. The summation of all the angles calculated for each snaxel divided by the total number of snaxels, *N*, gives the overall GACO of the object considered. From Figure 4.2 it is evident that for the initial frame the GACO cannot be computed due to its dependency on a previous AC vector for its calculation. (Note that the angles calculated are in radians which can be converted to degrees using the equation 4.9).

The overall flowchart describing the computation of GACO is as shown in Figure 4.2.

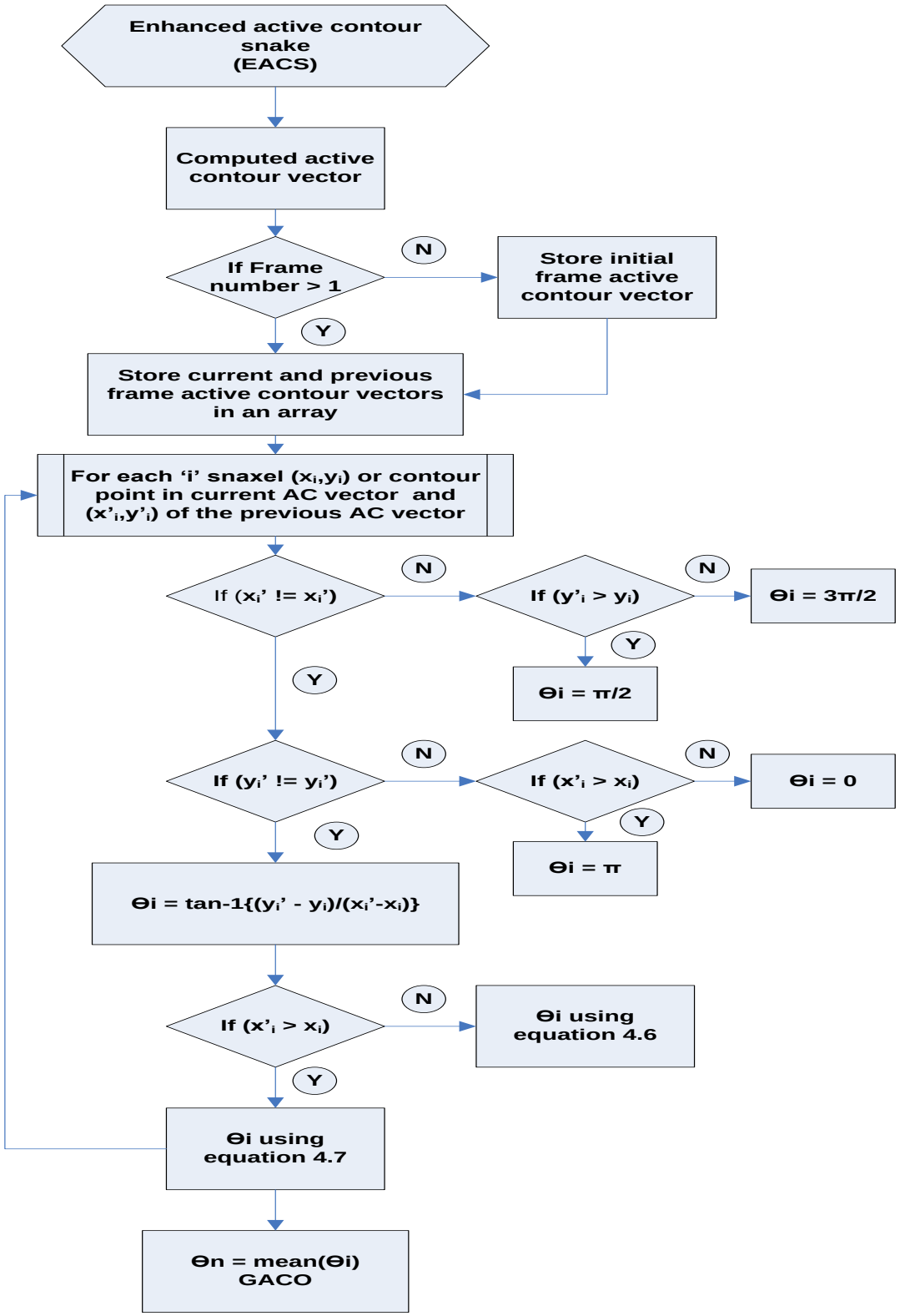

Figure 4.2 Overall flowchart for computing the global active contour orientation (GACO)

Figure 4.3 shows an example of the GACO at 162^0 being calculated in the 2^{nd} quadrant (i.e. 90^0 to 180^0).

Figure 4.3: Global active contour of the object contoured in blue with AC vector computed as 162^0

4.5 Real-time door surveillance application

4.5.1 Door surveillance introduction

Some of the prominent surveillance solutions are centered on an entrance or an exit door. Doors pose difficult problems to most surveillance systems as they usually violate the basic assumption about the nature of the background. The movements involved with doors are relatively infrequent compared to camera noise or jittering clutter objects. People or objects dynamically occluding the moving door add in to the foreground and

hence create ambiguity in detecting the activity at the door. Generating an alarm when a door event occurs during an obscured entrance at the door is challenging. The detection of door activity is even more challenging with the presence of varying illumination conditions. A number of door surveillance techniques can be found in the literature. A mixture of Gaussians (MoG) background model technique is very popular due to its flexibility and stability in complex scenes [80] [57], [62], [61]. A technique of modelling scene appearance by storing Gaussian pixel distributions corresponding to a discrete sample of the door's range of motion has also been implemented [81], [60]. An algorithm to detect doorways in a scene by observing vertical edge features from camera images fused with laser range information has been developed [82], [45]. A technique of using a generative probabilistic model of the scene to generate a maximum-likelihood map of the walls and doors from visual range data has been presented in [83], [45]. The biggest disadvantage of such methods is either that the camera view is just above the door or they address scenarios devoid of complex doors, moving cameras, occlusions and varying illumination conditions.

The proposed door surveillance application uses edge-map information to detect moving Harris corners in a small localised window at the top of the door. An assumption is made that the top end of the door is normally not occluded due to human activity. The window in which the Harris corner detector is applied is called the Harris window (HW). The window around the door is selected for activity detection when the door is displaced. This window is called a surveillance window (SW). The proposed overall method also generates an alarm to signify whether the door was displaced to provide entry or exit. To do this the activity at the surveillance window is classified based on the global active contour orientation (GACO). An overall active contour object orientation is computed within the surveillance window to classify the activity. The

algorithm also aims at compensating for the effects due to camera motion. A Hough transform is applied to a distinct region in the frame to detect a line. In case of camera movement the detected line coordinates are used to readjust the Harris and surveillance window for accurate surveillance. The novel contribution of this work is the application of these methods along with the GACO to a new problem involving doors and the use of problem-specific constraints to improve the accuracy of the surveillance.

4.5.2 Edge-map based Harris corner detector

In order to detect the door event, the frames of the 'Door-Zone' videos are pre-processed with an average-estimating kernel and a Gaussian kernel. The presence of multi-directional lighting effects from artificial lighting sources and different illumination effects originating from transparent windows in the doors and elsewhere in the scene, complicate the processing considerably. An omni-directional edge detector [55] is used to estimate the door edges as it helps to remove the effect of both impulse and Gaussian noise present in each frame. It is particularly effective in picking up the weak edges created due to the movement of the door during an event. It has been observed that the edge detector is also efficient in being invariant in the presence of burst noise due to variation in the bulb filament lighting.

The application of the multi-stage edge detector on each frame can be demonstrated by choosing a typical frame from the i-Lids Door Zone single door videos [55]. Figure 4.4 shows a typical Single Door Zone video frame from the dataset. Figure 4.5 (a) shows the greyscale intensity image of the top-left rectangle (Harris window) of the single door and Figure 4.5 (b) shows the edge detected at the top-left corner of the door using the edge detector.

Figure 4.4. The single door video frame from i-LIDS door surveillance dataset to be processed

(a) (b)

Figure 4.5. (a) Left: The top-left corner of the door frame converted to greyscale intensity image; (b) Right: the edge detector used on the left image to find Omni-directional edges for further processing.

The search of local maxima of the Harris response function gives the Harris corner detected in the specified window. It is suggested in [64] a local maxima search of the Harris response function is used to obtain corners in an image.

$$Corners_{[n,idx]} = \max_{local}\left(R_{Harris}\right) \qquad (4.10)$$

In the equation 4.10, the Harris response function is denoted as R_{Harris} and the local maxima search function is denoted as $\max_{local}(\ )$. The process works by finding the maximum value point in the Eigen value vector derived by Taylor series expansion of the pixel intensities in a particular area of the image. The edges detected in the Harris window of each frame are used to detect the Harris corners of the door by employing a Harris corner detector [64]. The most significant corners in the Harris window (HW) of the door are detected for each frame. The changes in count of the corners and displacement of the detected corners as the event progresses are estimated and used for deciding whether the event occurred or not. The algorithm checks for any change in the displacement or the count of the corners and gives out an alarm. The number of corners, *n*, and their indices 'idx' can be obtained as described in equation 4.10. A statistical decision algorithm is also implemented to decide whether the door is opened or closed based on the fact that the detected corners displace back into the same position after the door closes.

The Harris corner detected in the Harris window for the two kinds of doors namely: single and double door, are shown in Figure 4.6 and 4.7, respectively.

Figure 4.6. The single door video frame with Harris corner detector applied to the Harris window showing the displaced corner at the top when the door opens

The corners detected in the Harris window for each kind of door displace when the door opens and return back into the same position when the door closes.

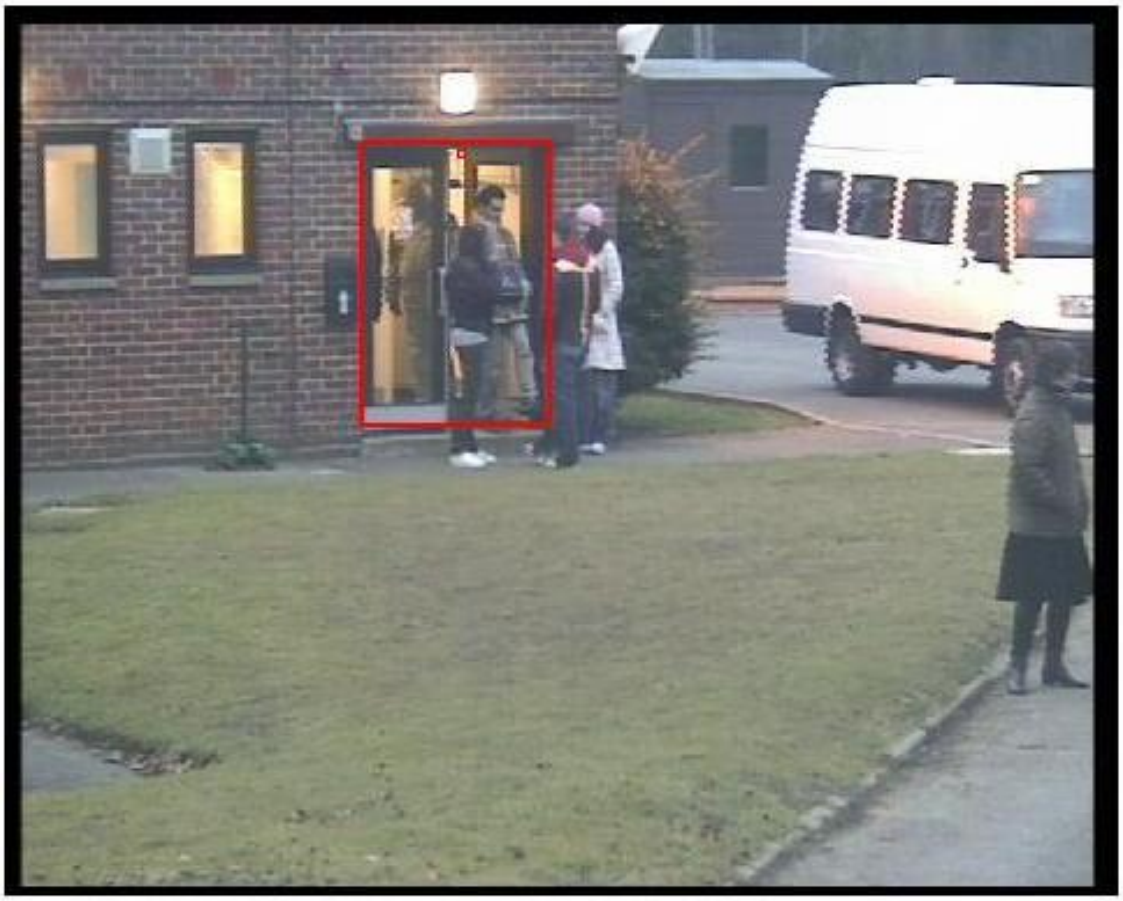

Figure 4.7. The double door video frame with the displaced corner at the top

Thus the capability to reliably detect the door corners based on the edge detector [55] is exploited in the decision algorithm to aid in reliable Door-Zone surveillance. Further, due to the effective estimation of the door activity, the decision algorithm is enhanced using the global active contour orientation (GACO) technique to indicate if people exit or enter the door.

4.5.3 Door surveillance algorithm design

The initial set up of the algorithm comprises the user interface designed to initiate surveillance. The interface prompts the user to signify if the door being used is a single or a double door. A designed mouse interface is then activated to signify the Harris window and the surveillance window for the algorithm to actively monitor the specified

regions of the frame. The location of the Harris and surveillance window is stored for segmenting the region. The flowchart of the algorithm is depicted in Figure 4.8. The user interface part of the algorithm is used only at the first frame or during the initial set up of the system.

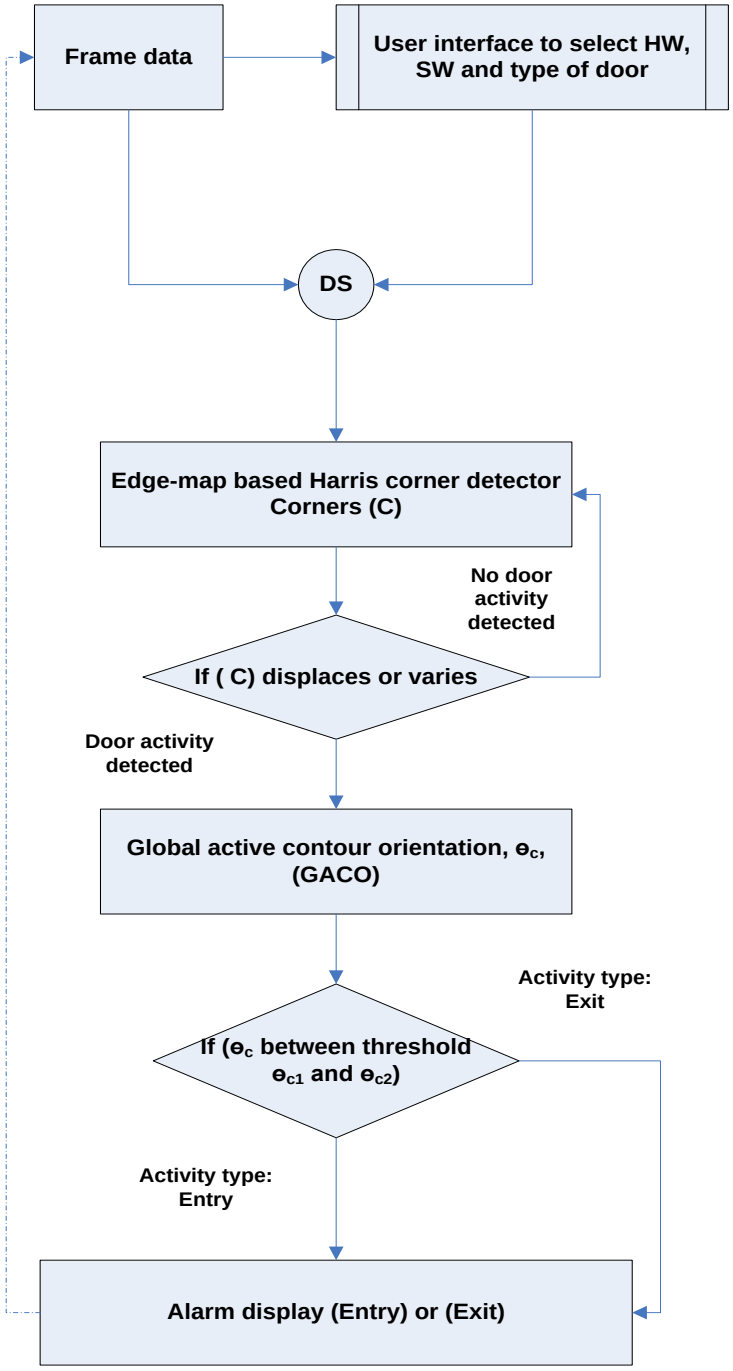

Figure 4.8. The Door surveillance system designed to evaluate the i-LIDS door surveillance dataset

Note that DS signifies the proposed door surveillance system, Corners(C) contains both the total number of corners and the index and threshold (θ_{c1}) and threshold (θ_{c2}) are the range of angles through which objects/people can enter the door. The Harris (HW) and surveillance windows (SW) are initialised at the first frame and stored for further processing.

4.5.4 Hough transform based error correction

The Hough transform is a feature extraction method used to analyse images and has been used in several computer vision applications [38], [39]. The Hough transform is concerned with the identification of lines, circles, ellipses and arbitrary shapes. The generalised Hough transform (GHT) is the modification of classical Hough transform using the principle of template matching [39]. Although the 'i-Lids Door Zone' dataset contains videos from fixed cameras, it has been observed that the cameras are often subject to movement due to wind and other environmental changes. Thus an additional Hough transform error correction module has been used in the door surveillance system designed.

The proposed technique uses a specific region in the frame containing a line-like object near to the door to extract a fixed length line using Hough transforms. The position (x, y) of the line is stored for correction of the Harris and surveillance window positions. If there is a camera movement due to environmental changes, the position of the line detected in the specified region changes. The change in the line position is used to update the position of the Harris window (HW) and the surveillance window (SW) for further processing. The line and region selected for the two types of doors, namely single and double doors, are shown in Figure 4.9 and Figure 4.10, respectively.

Figure 4.9: Hough transform region and the line detected for Single door i-LIDS video

Figure 4.10: Hough transform region and the line detected for the double door i-LIDS video

It has been observed that the error correction implemented avoids most of the false alarms due to the camera motion and enhances the accuracy of the proposed surveillance system. The error correction is performed every 10 minutes of the video to minimize computational cost thus making the surveillance system robust and real-time.

4.6 Results and discussion

The proposed technique is tested on all the videos provided with the i-LIDs dataset, [43] with encouraging results. The HW for the single door is located at the top-left corner of the door. During an activity the Harris corners in the HW displace, generating the primary alarm. The primary alarm activates the global active contour orientation module to determine if the door was displaced to make an entry or an exit. Figure 4.11 below depicts the continuous surveillance at the Harris window.

Figure 4.11: Active surveillance on single door videos HW on top-left corner

The door activity primary alarm is displayed in Figure 4.12 showing the primary alarm.

Figure 4.12: Primary alarm on single door videos with 2 corners detected in HW

The secondary alarm signifying entry or exit at the door is shown in Figure 4.13.

Figure 4.13: Secondary alarm on single door videos showing ‘exit’ at the SW

The active contour global orientation module decides whether the door is displaced for ‘entry’ or an ‘exit’ to activate a secondary alarm around the surveillance window (SW). Robust surveillance during surveillance window (SW) occlusions is shown in Figure 4.14.

Figure 4.14: Robust surveillance during SW occlusion due to external activity (no false alarm)

During occlusions and people moving in front of the door the system produces no false alarms, thus improving the accuracy of the system.

The top-centre window containing the two edges of the door is used as the Harris window for the double door scenarios. The double door is continuously monitored at the HW as shown in Figure 4.15.

Figure 4.15: Active surveillance on double door videos HW at top-center

During door activity the two corners of the door displace and initiate a primary alarm as shown in Figure 4.16

Figure 4.16: Primary alarm on double door videos with 2 corners displaced in the HW

The primary alarm triggers the global active contour orientation (GACO) module to detect the type of activity at the door as shown in Figure 4.17.

Figure 4.17: Secondary alarm on double door videos when 2 corners move in the HW

Figure 4.18: Robust surveillance at the double door during occlusions in the SW

The robust surveillance during SW occlusions is shown in Figure 4.18. The door is subject to activities around the surveillance window. The false alarms due to the movements around the door are avoided due to the selection of the Harris window. The system's tolerance to false alarms is as shown in Figure 4.18.

The reason for computing an overall active contour orientation is due to the fact that the door is frequently occluded and the surveillance window is subject to continuous human activity during busy hours of the day. Hence, the use of the centroid of the active contour for computing the exit or entry angle produces inefficient and false secondary alarms. Thus, computing the global active contour orientation (GACO) of the object is of paramount importance in obtaining accurate secondary alarms and assists in deciding if the door activity occurred for entry or exit.

The proposed method was also tested on several difficult scenarios containing sequences where the camera goes out of focus and the image quality is deteriorated. The surveillance system designed is found to be tolerant to camera noise and yields accurate alarms during door activity. The Hough transform based error correction is applied every 10 minutes of the surveillance time. The differential distance error calculated between the previous Hough line coordinate and the current Hough line coordinate is used in compensating for the camera movement due to wind or other environmental factors. The error calculated is utilised in moving the Harris window (HW) and the surveillance (SW) in order to carry out robust surveillance at all times. The system's accuracy with deteriorated frame quality is shown in Figure 4.19 and Figure 4.20. Figure 4.19 shows the primary alarm due to door activity. Figure 4.20 shows the Hough transform error corrected and occlusion tolerant secondary alarm calculated using the proposed GACO technique to signify an exit at the door.

Figure 4.19. Primary alarm as the double door displaces in poor quality frame

Figure 4.20. Secondary alarm signifying an exit at the door after Hough transform error correction

The i-LIDS door surveillance dataset was evaluated using the techniques discussed in the previous sections. The performance results on the video sequences demonstrates that the use of GACO to evaluate the door surveillance dataset gives a detection rate of 91.2% and a probability of genuine alarms of about 92%. Overall performance of the proposed door surveillance algorithm on a set of i-LIDS door surveillance videos is tabulated in Table 4.1.

Stage	Total Alarms	True positive Alarms	False positive Alarms	False negative Alarms	Detection Rate 'r'	Probability of Genuine Alarm 'p'
DSTEA101a	22	21	2	1	95.45%	91.3%
DSTEA101b	18	15	2	3	83.33%	88.24%
DSTEA105a	30	29	2	1	96.67%	93.55%
DSTEA202b	27	24	1	3	88.89%	92.31%
DSTEA202a	36	33	2	3	91.67%	94.29%
Combined Results	**133**	**122**	**9**	**11**	**91.20%**	**91.94%**

Table 4.1: Overall performance of the proposed door surveillance application using the GACO secondary alarm approach

In the Table 4.1, the true positive alarm denoted as 'a' signifies a genuine alarm event and the false positive alarm denoted as 'b' signifies that the system alarmed without the presence of a genuine alarm event. The false negative alarm denoted as 'c' is the number of times the system missed a genuine alarm event during surveillance. The detection rate 'r' is computed using equation 4.11.

$$r = \frac{a}{a + c} \tag{4.11}$$

The precision or the probability of an alarm being genuine ‘p’ can be calculated using equation 4.12. The precision ‘p’ is calculated by dividing the number of true positive alarms by the sum of true positive and true negative alarms as shown in equation 4.12.

$$p = \frac{a}{a+b} \tag{4.12}$$

From Table 4.1, it is evident that a high number of true positive alarms are generated for each video sequence giving an overall detection rate of 91.2%, thus proving that the system is a robust surveillance system. The probability of a genuine alarm of almost 92% signifies that the surveillance system is reliable under varying conditions[43].

4.7 Summary

The overall enhanced active contour vector orientation method known as the global active contour orientation (GACO) measure has been discussed in detail in this chapter. It is found that by using simple geometric calculations and assumptions, it is possible to determine the direction of motion of a selected object using the GACO technique as discussed in Section 4.3. The GACO statistical measurement calculations have been discussed by choosing an appropriate quadrant in which the movement occurs. The expressions determining the local snaxel orientation have been described in detail. GACO is the mathematical average of the complete set of local snaxel orientations contained in a contour. It is computed as the total sum of the orientation angles of each snaxel divided by the total number of snaxels. The GACO statistical measurement system is further described using a design flowchart along with a few examples in Section 4.4.

A robust algorithm for door surveillance has been presented in this chapter in order to evaluate the performance of the proposed GACO measure. The edge-map based Harris

corner detector has been used to provide a primary alarm of door activity. The Harris corner detection is found to be more accurate due to the pre-processing steps involving Gaussian smoothing and multi-stage edge detection as discussed in Section 4.5.2. The algorithm is designed to use the Harris corner data to decide whether the door is displaced to produce a primary alarm. The primary alarm triggers the global active contour orientation (GACO) module as described in Section 4.5.3, to compute the object's angle of movement at the door. The calculated global orientation angle is used to decide whether the objects entered or moved out of the door, producing a secondary alarm. The system is improved by the addition of a Hough transform error correction module, to append the Harris and surveillance window locations based on the error calculated due to camera motion. The system is found to be accurate even when the frame quality is reduced due to the camera movements. The selection of the Harris window at the top of the door assists in making the system robust and efficient even in the presence of occlusions at the door. The overall performance of the algorithm is found to be accurate and robust on several i-LIDS door-zone dataset sequences as discussed in Section 4.6. The Table 4.1 shows that the proposed GACO based door surveillance system has a success rate of 91.94%. The enhanced active contour model can be modified and used in several applications. A robust moving object tracking system using an optimal trade-off maximum average correlation height (MACH) filter assisted by the EACS is discussed in detail in Chapter 5 of the thesis, together with several challenging target tracking scenarios.

Chapter 5

ENHANCED ACTIVE CONTOUR ASSISTED OT-MACH FILTER TRACKER

5.1 Introduction

Accurately tracking moving targets in a complex scene involving moving cameras, occlusions and targets embedded in noise is a very active research area in computer vision. This chapter presents methods on implementing and enhancing an optimal trade-off maximum average correlation height filter (OT-MACH) as a tracker. The enhanced active contour technique discussed in the previous chapters has been utilised to assist an OT-MACH filter to provide robust detection and tracking. Several techniques involving the area of support to the filter has been discussed with results to conclude that the aforementioned active contour technique gives consistent accurate tracking. The tracker has been tested both on colour visible band as well as infra-red band video sequences acquired from the air by the Sussex police helicopter and on the videos provided by the DSTL.

An interrupt based user interface which is active for the whole duration of tracking in the videos or live feed has been developed. The user defines a circular area around the target using an interrupt to start tracking the selected target. The user interface is designed so as to allow the user to select a different target for tracking at any time. The filter is updated at a frequency selected by the user. The filter parameters are initialised and amended using an initialisation text file. The initialisation file is used to fetch the frequency of up-date, i.e. rate of correlation, and filter parameter values provided by the

user. The design makes the filter more resistant to progressive changes in the object's orientation and scale.

The Kalman and Particle filter used as a tracker are implemented and utilised to trace the target in multi-target environments. This allows the tracker to continuously track the selected target by ignoring adjacent identical non-target objects in the scene. The OT-MACH tracker is also compared with the Kalman and particle filter [68], [67]tracking for its accuracy and efficiency in tracking selected targets in real-time scenarios; results are compared with [71], [73], [75] and [67].

5.2 Chapter Organisation

The entire chapter is organised in the following way: the design of OT-MACH filter is discussed in the next section. The following sections describe how the OT-MACH filter has been implemented as a robust target tracker in real-time situations. Section 5.4 describes the design of a unique user interface developed to enhance the usability of OT-MACH filter as a tracker. In Section 5.5 several techniques on reference image extraction for the OT-MACH filter have been discussed in detail with flowcharts depicting the procedures involved. Sections 5.6 and 5.7 discuss the issue of real-time implementation of the OT-MACH tracker. Comparison of the OT-MACH tracker with an extended Kalman filter and a colour based Particle filter have been discussed in Section 5.8. Multiple target-like object confusion problems and their solution have been discussed in the Section 5.9. Finally Section 5.10 displays several results of the OT-MACH acting as a robust tracker. Conclusions are drawn in Section 5.11.

5.3 Design of Optimal trade-off (OT) MACH filter

The maximum average correlation height filter (MACH) has been proposed by Mahalanobis *et al.* [93], [92]. The MACH filter theory allows a better filter solution by removing the peak-height constraint from the SDF based design methods [96], [89], [76]. The MACH filter design includes the minimisation of an average similarity measure (ASM) that leads to a compact set of correlation planes that resemble each other and exhibit the least possible variation [93].

To understand the MACH and related OT-MACH filter it is useful to review its derivation given by Vijaya Kumar *et al* [93], [91], [97]. A variation of the MACH filter that includes an additional output noise variance (ONV) term to offer better clutter resistance [99], [98] is known as the optimum trade-off maximum average correlation height filter (OT-MACH). It is assumed that the training set consists of N true-class images and M images from the false class. The i^{th} training image is denoted by $\varphi_i(m,n)$ and is represented in the frequency domain by a vector q_i, obtained by lexicographically reordering its 2-D FFT. The filter transfer function is denoted by the vector h.

The ONV (denoted as C) is defined as:

$$C = h^{+} P h \tag{5.1}$$

where P is the zero mean stationary noise power spectral density arranged into the diagonal of a matrix, h is the filter transfer function. The average correlation height (ACH denoted as W) is defined as:

$$W = \left| \frac{1}{N} \sum h^{T} q_i \right| = \left| h^{T} \bar{q} \right| \tag{5.2}$$

The OT-MACH seeks to minimise the energy function expressed as equation 5.3:

$$\begin{aligned} E(h) &= \alpha(ONV) + \beta(ACE) + \gamma(ASM) - \delta(ACH) \\ &= \alpha(h^{+}Ph) + \beta(hD_x h) + \gamma(h^{+}S_x h) - \delta\left(h^{T}\bar{q}\right) \end{aligned} \tag{5.3}$$

where α, β, δ, and γ are variables whose sum equals one, D_x is the diagonal average power spectral density of the training images and the average correlation energy (ACE) now refers to the true class. The S_x denotes the similarity matrix of the training image set [93]. Rearranging the equation 5.3 gives us:

$$h = \frac{\bar{q}}{\alpha(C) + \beta(D_x) + \gamma(S_x)} \tag{5.4}$$

where, C is the diagonal power spectral density matrix of additive noise. This version of the filter is used throughout this chapter.

5.4 Interrupt based user interface

A user interface has been developed to select the target from the source. The user interface is activated once the user finds a visual target in the scene by pressing any key from the keyboard. The interface creates a mouse interaction protocol on the current frame. The interface takes a snap shot of the current frame and displays the image for user target selection. With the assistance of the mouse, the user can draw and drag a circle over the target to be selected for tracking. The initialised parameters such as the x and y co-ordinates of the centre and the diameter of the circle are stored for further use to allow real-time fabrication of the filter. The circular area around the target is traced to find the coordinate vector at the circumference of the circle. These coordinate points are used to extract the reference image from the target. The extracted circular target is then passed on to one of the supporting methods namely: rectangular, circular and active

contour based extraction, to process the reference image selected from the target. The processed target is automatically cropped and used for training the filter. Figure 5.1 below illustrates an original frame in which the initialised circle has been drawn over the target using the developed interface.

Figure 5.1: Initialisation circle over the target vehicle to be tracked

5.5 Reference Image extraction

5.5.1 Rectangular target extraction

The rectangle coordinate is obtained from the initialisation step by computing a bounding rectangle around the selected circle. The rectangle parameters (x_1, y_1) and (x_2, y_2) are computed and stored in the configuration file. Width and height are calculated as the difference of the parameters stored. A blank reference template image of frame size is created. The centre of the frame is computed as frame_width/2 and frame_height/2 for x and y coordinates respectively. The current frame is cropped at the initialised

rectangle and copied into the blank image at the centre. The starting point for copying the target to the centre is straightforwardly calculated as:

$$(\mathrm{X}_c, Y_c) = \left(\frac{\text{Frame\_width}}{2}, \frac{\text{Frame\_height}}{2} \right) \tag{5.5}$$

where the centre of the blank frame is denoted as (X_c, Y_c). The co-ordinates used to copy the cropped target into a blank frame are, clearly, then given as:

$$(X, Y)_{tocopy} = \left(X_c - \frac{rect\_width}{2}, Y_c - \frac{rect\_height}{2} \right) \tag{5.6}$$

The windowed target, shown in Figure 5.2, can then be used to train the filter. If 'theta' from the user defined initialisation file is 2 degrees, the windowed reference is rotated in 2 degree increments to +6 and -6 degrees, thus obtaining 7 reference images to multiplex into the filter function. This ensures some degree of in-plane rotation tolerance of the filter and facilitates its ability to maintain a track on the vehicle for n frames, after which the filter is up-dated.

Figure 5.2: Target reference image used to train the filter

The flowchart depicting the rectangular target extraction method is shown in Figure 5.3.

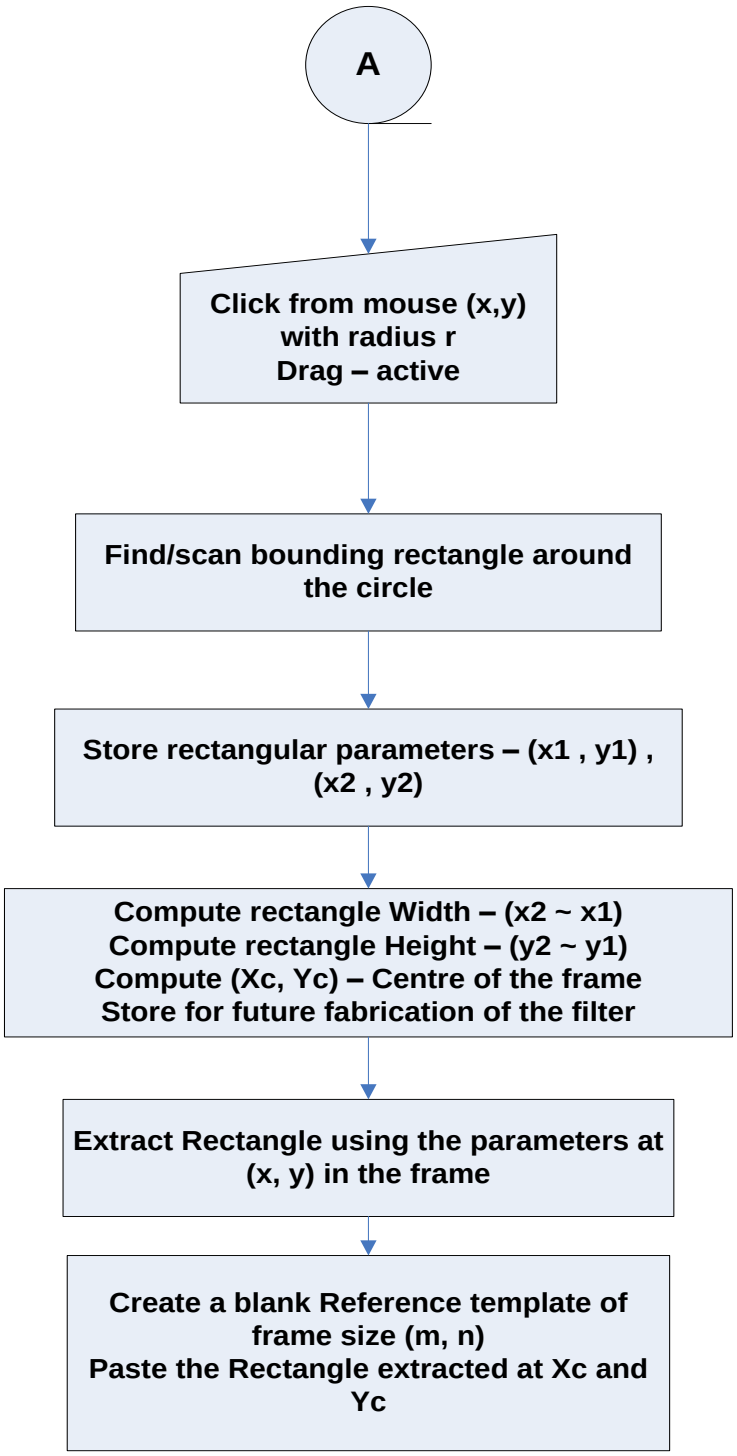

Figure 5.3: Flowchart for rectangular target extraction (A)

5.5.2 Circular target extraction

The circle coordinates are computed from the circle selected on the target. The centre of the circle (x, y) is obtained from the initialisation step. The circumference (C) is obtained by tracing the circle. The centre of the frame is computed as frame_width/2 and frame_height/2 for x and y coordinates, respectively. The current frame is cropped at the initialised circle and copied into the blank image at the centre. The starting point for copying the target to the centre is calculated according to equation (1). The flowchart depicting the Circular target extraction method is shown in Figure 5.4.

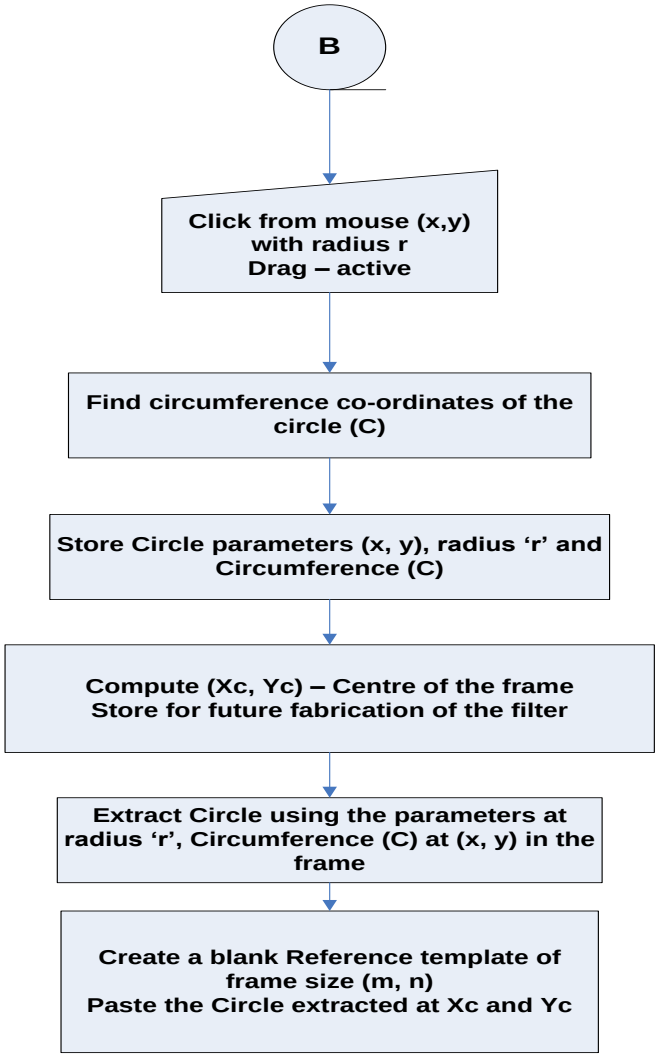

Figure 5.4: Circular target extraction (B)

5.5.3 Active contour based target extraction

The Active contour work discussed in previous chapters employs the principle of energy minimization. The energy of each coordinate point is calculated based on the neighbourhood pixels of each point. A difference of Gaussian (DoG) filtered image of the circular area is computed to emphasise the exterior edges of the target. The energy minimisation process is executed until the exterior edge of the target is contoured. The process is maintained and controlled by the number of iterations throughout the contour vector points. The number of iterations required and the snake parameters are also included in the configuration file for standardised usage of the algorithm in software. The energy functional computed and iterated for each coordinate point is described by the expression in Equation 5.7 [31], (for the full derivation of the snake equation please refer to Appendix 1):

$$E^*_{snake}(s) = E_{\text{int}}(v_s) + E_{image}(v_s) \tag{5.7}$$

where * means that this is a continuously updating snake energy.

This can be expressed as:

$$E^*_{snake} = a(s).\left|\frac{dv_s}{ds}\right|^2 + b(s).\left|\frac{d^2v_s}{ds^2}\right|^2 + c(s)E_{Edge} \tag{5.8}$$

where the first-order and second order differentials are approximated for each point that is searched in the local neighbourhood of the currently selected coordinate point. The weighting parameters *a*, *b* and *c* are all functions of the contour. E^*_{snake} is thus the overall Snake energy term and E_{edge} the computed edge energy. A complete derivation of the Snake energy minimisation expression is given in Appendix 1. By calibrating the Snake, the exterior edge of the target is contoured. The edge contour gives the

coordinate vector of the target in the frame. The shape vector thus obtained is used to segment and extract the target object from the scene, Figure 5.5. It is placed at the centre of a blank zero background image, as shown in Figure 5.6 to create training images for filter initialisation and computation [73], [88].

The contour's coordinate vector of the exterior edge of the target is used to extract the target from the frame interrupted. A blank reference template image of frame size is created. The centre of the frame is computed as frame_width/2 and frame_height/2 for x and y coordinates, respectively. The current frame in which the target is contoured is used to extract the contoured object and is copied to the centre of the blank frame.

The figures below show the steps for generating the reference images.

Figure 5.5: Initialisation by selecting circular area around the target

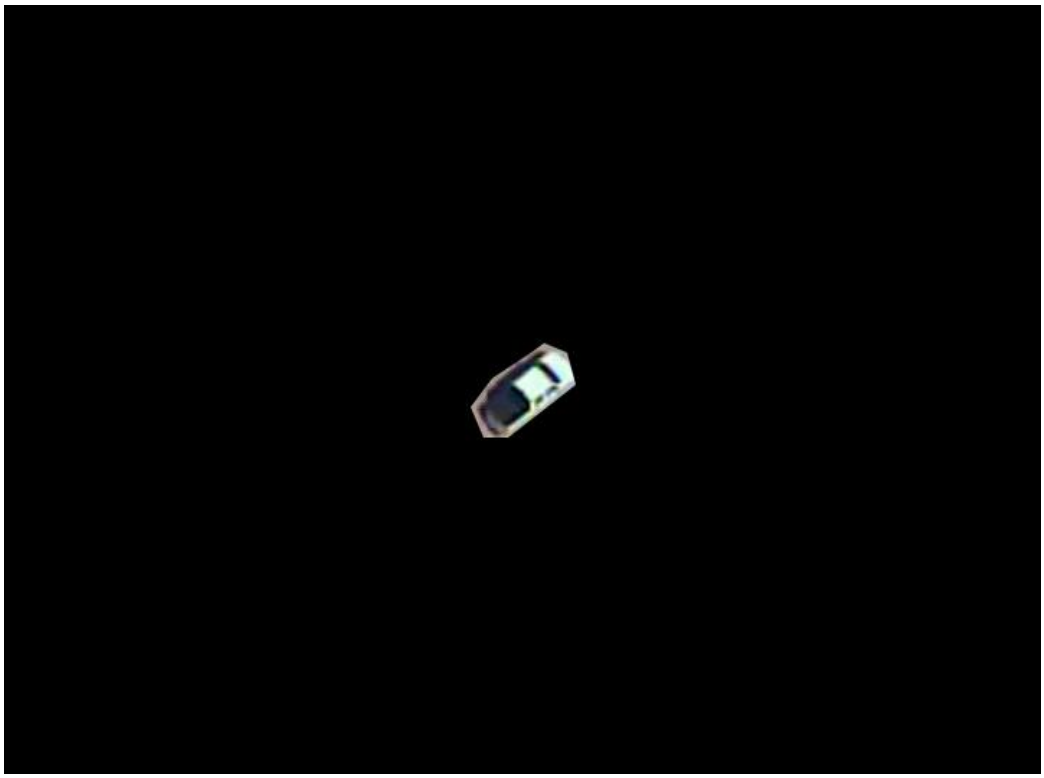

Figure 5.6: The reference image generated for training the filter

The selected target, shown in Figure 5.6, can then be used to train the filter. If ‘theta’ is 2 degrees, the reference image is rotated by 2 degree increments to +6 and -6 degrees, thus obtaining 7 reference images to multiplex into the filter function. This ensures some degree of in-plane rotation tolerance of the filter and facilitates its ability to maintain a track on the vehicle for n frames, n being the frequency of the filter upgrade parameter fetched from the configuration file. Unlike rectangular and circular target extraction techniques, the active contour based target extraction is found to produce accurate correlation peaks to distinguish between the target and the background. A flowchart depicting the active contour extraction is given in Figure 5.7 [31].

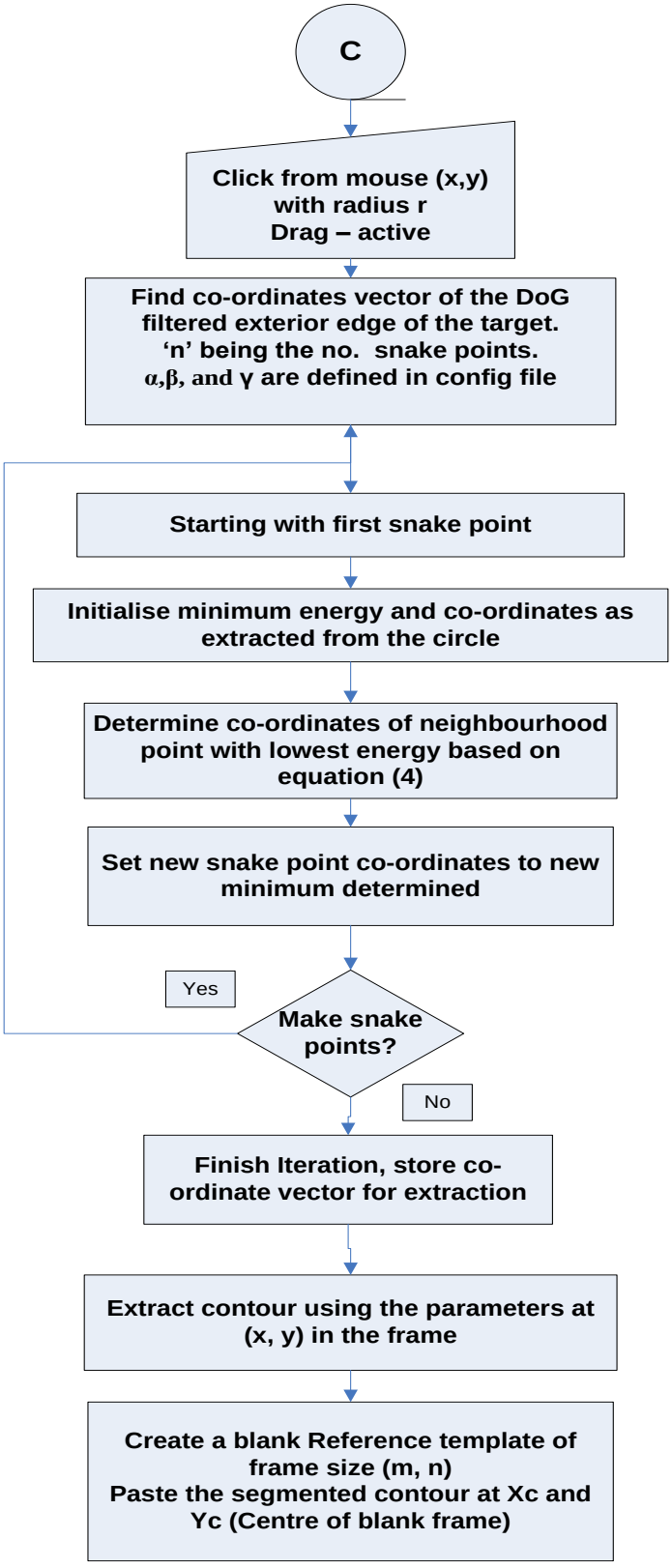

Figure 5.7: Active contour extraction method [31]

5.6 Computing rotationally multiplexed reference image

A double precision addition of several rotated reference templates is performed to compute a rotationally multiplexed reference image. The reference image obtained from the processing, as explained in the previous sections, is rotated in increments and decrements of 'theta' degrees. The 'theta' and the level of rotational multiplexing are fetched from the configuration file and are user defined. For example if the 'theta' value is 2 degrees and the level is '3', then the reference image is rotated in increments and decrements of 2 degrees between -6 and +6 degrees to obtain 7 rotated reference images. The reference images are of double precision and are added to obtain a multiplexed reference image, which is further used in fabricating the OT-MACH filter. A typical example of a rotationally multiplexed reference image is shown in Figure 5.8 [97], [99].

Figure 5.8: Rotationally multiplexed reference image

5.7 Real-time implementation of the OT-MACH filter

The OT-MACH filter is designed by passing the set of reference images to the filter design function. The filter is applied to every m^{th} input video frame to generate a correlation peak, the location of which indicates the position of the target vehicle in the video frame. The rate of cross-correlation peak generation is controlled by parameter 'm' initialised in the configuration text file by the user. The target location is then displayed using cross-hair markers as shown in the example of Figure 5.9 below. The filter can be updated in real-time or changed, based on the user's requirement, by initialising the update frequency in the configuration file. The rotational multiplexing increases the tolerance of the filter to changes of vehicle rotation angle between filter up-dates. There are also progressive changes in scale, due to variations in distance of the camera from the target vehicle, but these are sufficiently small that they can be accommodated by the filter up-date process described below.

Figure 5.9: Cross-hair on target

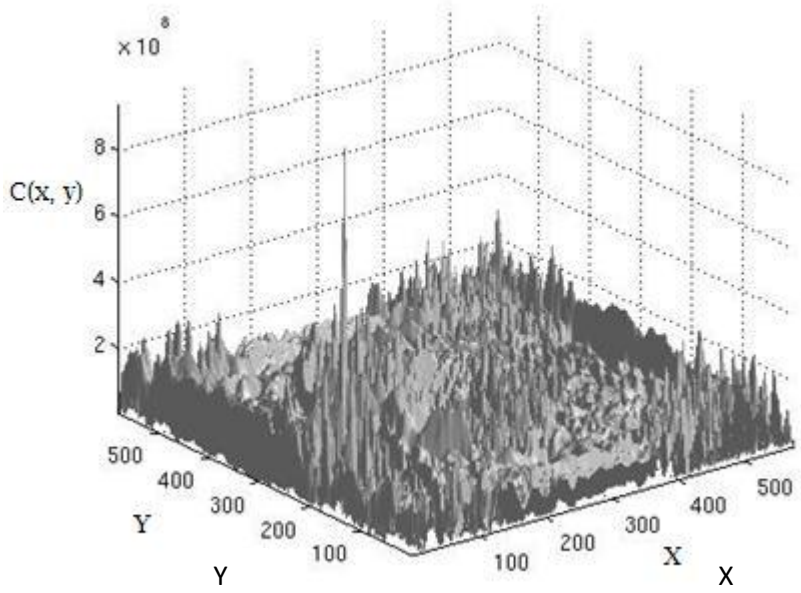

Figure 5.10: Correlation plane with peak location at (X, Y) = (195,342)

At every m^{th} frame (for example if m =25 in the initialisation file, then every second of the video sequence) the filter is updated. To do this, a correlation of the filter and target frame takes place and a measurement of the height of the correlation peak at the target location is made. To ensure the filter function has correctly identified the target vehicle, the correlation peak height obtained is compared to the average of the previous values at the rate of correlation set by the user in the configuration file. This allows the algorithm to take account of those cases in which the target may not be visible during the m^{th} frame cycle of measurement. Once a correlation peak of sufficient height has been obtained, the filter is up-dated using the current image. The reference image for the current frame is obtained, and a new rotationally multiplexed filter is created, as described previously. If the user decides to change the target at anytime, a keyboard interrupt can be used to switch 'on' the user interface to select a new target. This captures the current interrupted frame and allows the user to retrain the filter with a new target. The updating of the filter takes place as explained in the previous sections. The

correlation of every frame at the rate interval is again carried out to maintain the track for the next m frame sequence. The flowchart describing the OT-MACH tracker is shown in Figure 5.11.

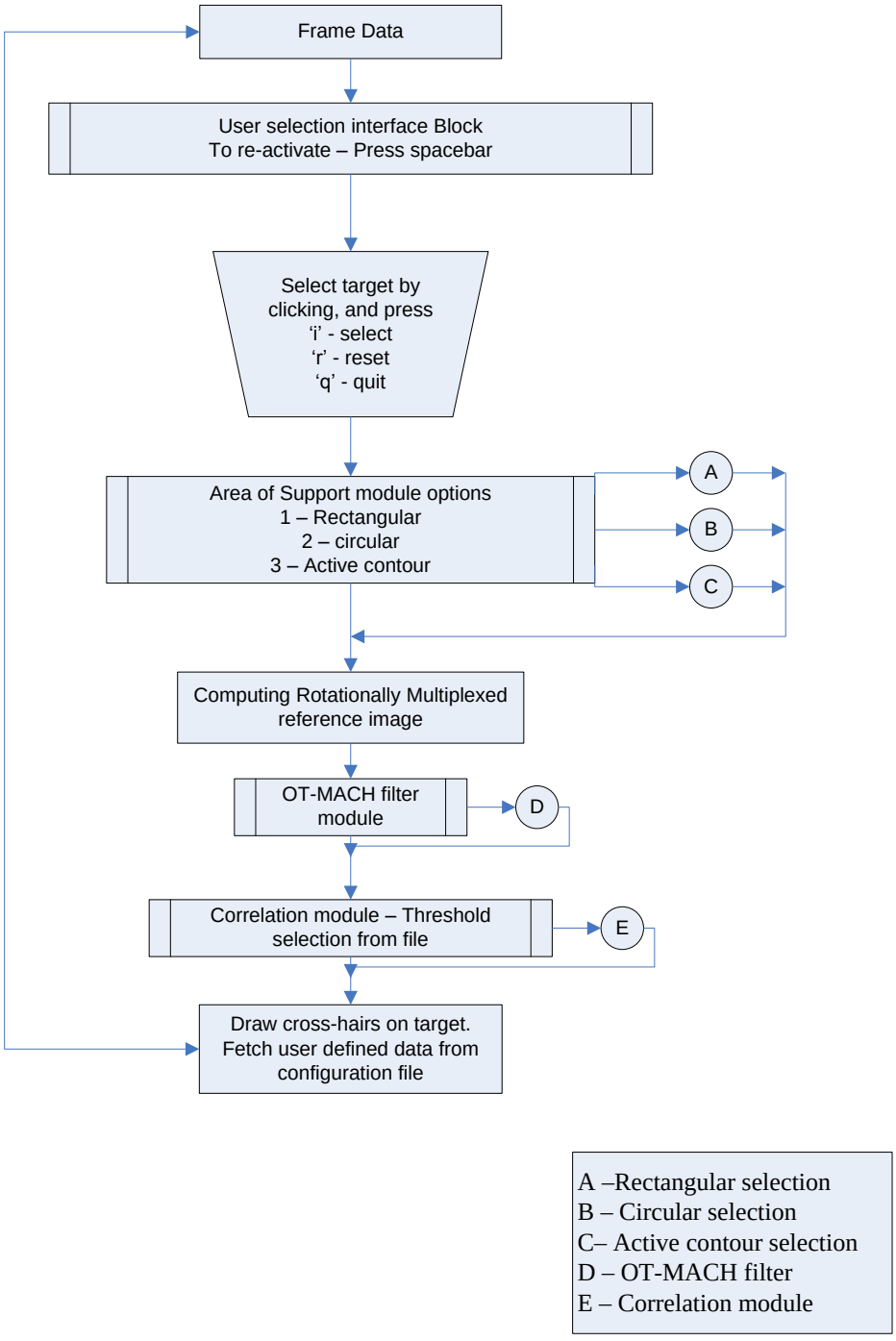

Figure 5.11: The OT-MACH tracker

Here ‘D’ signifies the OT-MACH filter module and ‘E’ signifies the correlation module. The Fastest Fourier transform in the West (FFTW) [94] is a C subroutine library for computing the discrete Fourier Transforms (DFT) in one or more dimensions. Using C library linkers the FFTW has been interfaced to the C program developed for the filter design. The output from the FFTW is not a shifted FFT. Shifting the zero component of the fast Fourier transform (FFT) to the centre of the spectrum is performed using the function FFTWSHIFT() implemented in C. The FFTWSHIFT() function rearranges the output obtained from the FFTW by swapping the first quadrant with the third quadrant and the second quadrant with the fourth quadrant. The swapping operation is depicted Figure 5.12 below.

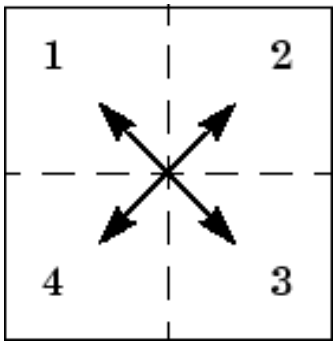

Figure 5.12: FFT Shift operation

The Fourier transformed output from the FFTW routine is converted to an image in the IPL_IMAGE format of the OpenCV library. Several OpenCV library functions were used to perform the swap operations to obtain the Shifted FFT (FFTSHIFT). The flowcharts for ‘D’ and ‘E’ are shown below in Figure 5.13 and 5.14.

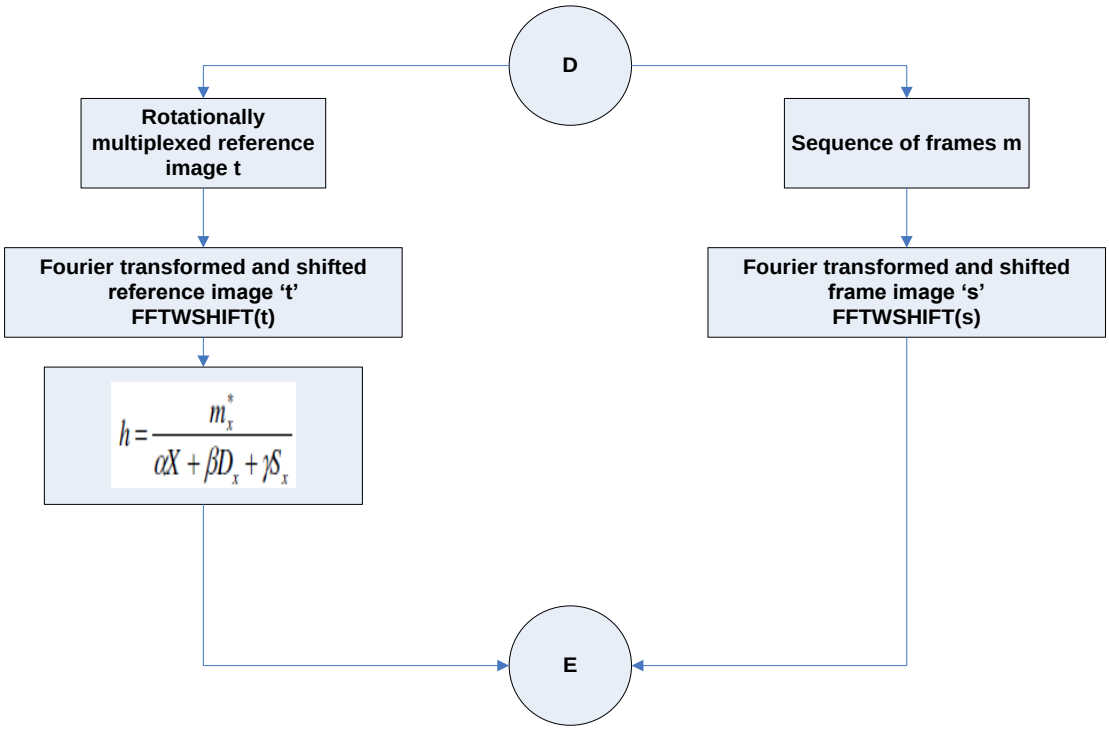

Figure 5.13: OT-MACH filter module

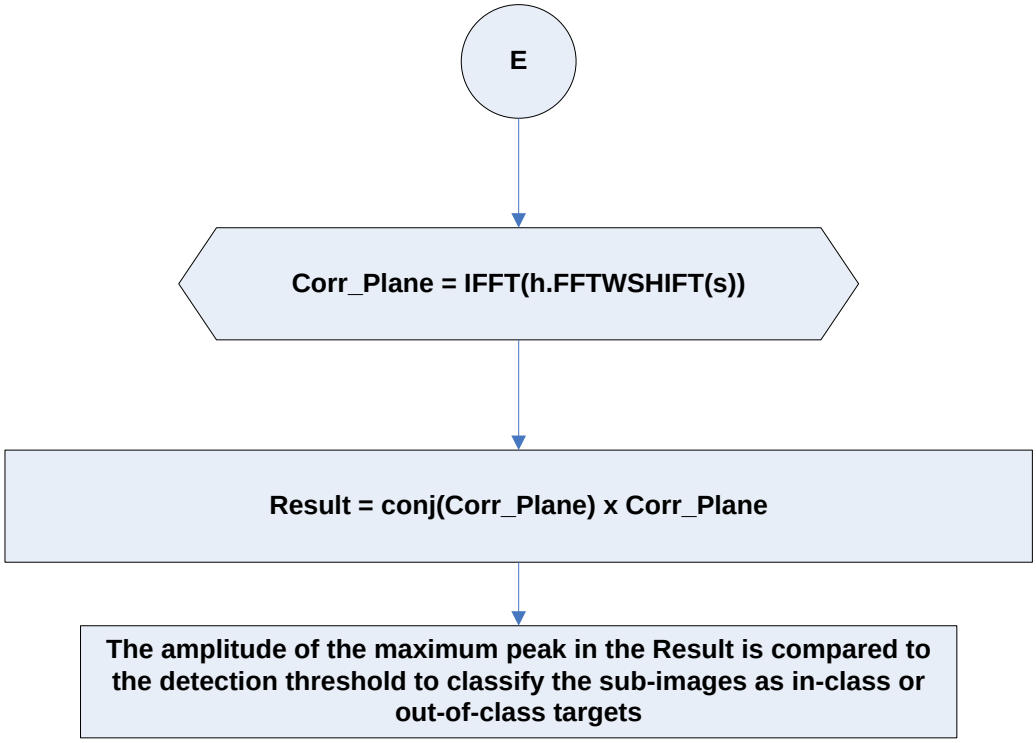

Figure 5.14: Correlation and decision module

5.8 Comparison of OT-MACH tracker with Kalman and Particle filters

An extended Kalman filter and colour based particle filter tracker have been tested to compare with the accuracy of the OT-MACH filter tracker discussed in the previous sections. The results of the comparison have clearly proven the OT-MACH filter tracker as being robust and efficient in non-linear, noisy and dynamic environments. The particle filter tracker tested is colour based and hence has not been used to verify the accuracy of OT-MACH filter tracker on infra-red video sequences [68], [69].

In the video sequences (snapshot figures shown below) the Kalman filter result, signifying the predicted position of the target, is indicated with a red cross-hair marker while the OT-MACH tracker result is marked with a yellow cross-hair. The particle filter result uses blue markers to signify the particles composing the filter along with a red track marker to trace the trail of the target. The noisy, dynamic and non-linear target scenes that have been used as test sequences have resulted in the inability of the Kalman and particle filters to perform as accurate trackers. The OT-MACH filter tracker has some degree of in-plane rotation and scale invariance as well as tolerance to orientation changes of the target object. Due to dynamic and frequent filter updates it has a higher track accuracy with varying velocity and other non-linear target real-time alterations. Figure 5.15 and 5.16 show examples of tracking failure using the Kalman and particle filter in situations where the OT-MACH filter is able to maintain successful tracking. The particle filter lags behind due to complex computation time required to decide on the location of the target as opposed to the OT-MACH filter tracker as shown in Figure 5.16. The Kalman filter gets diverted in a linear path loosing the target when the target changes its direction of motion dynamically, this is shown in the Figure 5.15 below.

Figure 5.15: Kalman filter (Red) and OT-MACH filter tracker (Yellow)

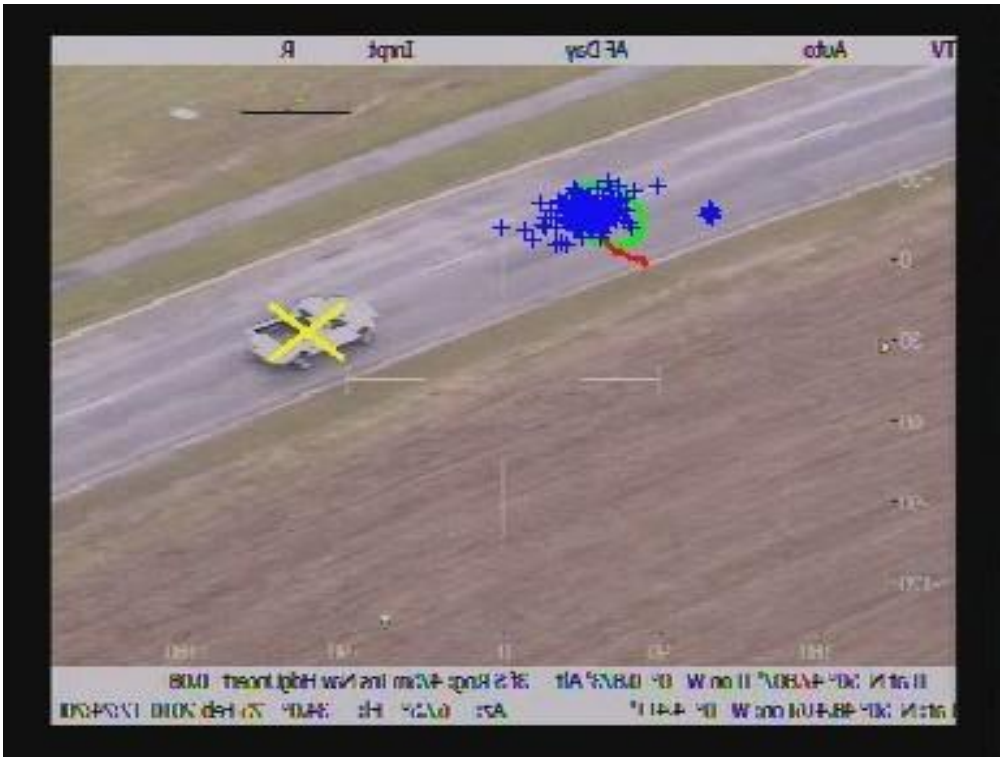

Figure 5.16: Particle filter (blue, green and red) and OT-MACH filter tracker (Yellow)

5.9 Use of an Extended Kalman filter to distinguish between target and non-target objects when in close proximity

There may be situations when multiple target-like objects are found in a scene. If the objects are identical, then the OT-MACH filter will detect multiple objects as targets. In order to help avoid loss of the selected target and to differentiate between targets and non-targets (in situations, for instance, in which very similar vehicles pass close to each other on a road), an extended Kalman filter has been developed; this allows a predictive track to be applied to the target object. It is implemented by estimating the uncertainty of the predicted position. A weighted average of the predicted value is then computed and most weight given to the value with least uncertainty. The maximum correlation peak location of the target is used in the filter as the original location. After obtaining about 10 coordinates of the target from the correlation result, the Kalman filter is trained to predict approximate future locations. The Kalman filter estimated coordinate is then used as a reference point to retrain the OT-MACH filter.

The extended Kalman filter, although a non-linear filter, it is not an optimal estimator when dealing with noisy frames, varying velocity targets and extreme scale variations. Encouragingly, the OT-MACH tracker has been found to work accurately with varying scale, orientation and velocity of the target object. In contrast, the original Kalman filter, as a target predictor, is feasible only for constant speed target scenes. The colour based Particle filter has been found to fail to precisely locate the target in noisy sequences and in infra-red band videos [70], [71], [73], [75].

The results of the partially successful Kalman filter are shown in Figures 5.17 and 5.18. The Kalman filter coordinates are indicated with red cross-hairs and the OT-MACH tracker coordinates with yellow cross-hairs.

Figure 5.17: Kalman filter and OT-MACH tracker outputs (frame 104)

Figure 5.18: Kalman filter and OT-MACH tracker outputs (frame 122)

5.10 Representative examples of tracking results

Several tests were conducted and several examples of tracking targets at different scales and orientations in DSTL videos are shown in Figures 5.19-5.22 below.

Figure 5.19: DSTL Video (1) frame 10 target acquired

Figure 5.20: DSTL Video (1) continuous tracking at frame 30

The OT-MACH tracker's ability to retain track of the target acquired during camera motion is demonstrated as shown in Figure 5.19 and 5.20. The Figure 5.21 and 5.22, demonstrate the tracker's efficiency to continuously track while the target undergoes continuous orientation changes. The DSTL videos shown in Figure 5.19 – Figure 5.22, are de-interlaced and low-pass Gaussian filtered as pre-processing steps before the OT-MACH tracker is applied in order to remove camera noise and distortions due to video compression and digitisation.

Figure 5.21: DSTL Video (2) frame 100 target acquired

Figure 5.22: DSTL Video (2) continuous tracking, scale and orientation changed at frame 130

Figure 5.23 to 5.28, shows the result of the OT-MACH tracker on Sussex police helicoptor videos. The frames displayed in Figure 5.23 and 5.24, contain multiple false objects. Due to rotational multiplexing and robust target template acquisition based on enhanced active contour snakes, the tracker displays good discriminability during the presence of false objects in the scenarios. The tracker efficiently discriminating the target and the false objects is demonstrated as shown in Figure 5.23 and 5.24.

Figure 5.23: Sussex Police video (1) Frame 5 target acquired in the presence of false objects

Figure 5.24: Sussex Police video (1) Frame 15 with multiple false objects

The tracking results shown in Figure 5.25 to Figure 5.27, demonstrate the tracker's ability to track the moving target undergoing considerable dynamic scale changes.

Figure 5.25: Sussex Police video (1) Frame 100, dynamic scale changes

Figure 5.26: Sussex Police video (1) Frame 120, scale changed false object present in close proximity

Figure 5.27: Sussex Police video (1) Frame 240, scale changed

Figure 5.28, depicts continuous tracking of the acquired target during noisy frames due to changes in the camera focus and movement.

Figure 5.28: Sussex Police video (1) scale changed and noisy frame at Frame 250

Another example of the OT-MACH tracker tracking a different coloured car drastically changing scale and lighting, is shown in Figure 5.29 and Figure 5.30. It has been demonstrated in the Figure 5.30 that the tracker is not only robust and tolerant to scale changes but also to varying lighting conditions. The Figure 5.31 is a typical example of the tracker performing accurate tracking on Sussex Police infrared noisy videos. The future frames as shown in Figure 5.32, Figure 5.33 and Figure 5.34, demonstrates that the OT-MACH tracker can efficiently tracking during dynamic scale change combined with several orientation changes.

Figure 5.29: Sussex Police video (2) Frame 50, scale changed

Figure 5.30: Sussex Police video (2) Frame 65, scale changed, varying lighting conditions

Figure 5.31: Sussex Police infra-red video frame 65, scale changed

Figure 5.32: Sussex Police infra-red video frame 104, scale changed with Gaussian noise

Figure 5.33: Sussex Police infra-red video frame 265, dynamic scale change

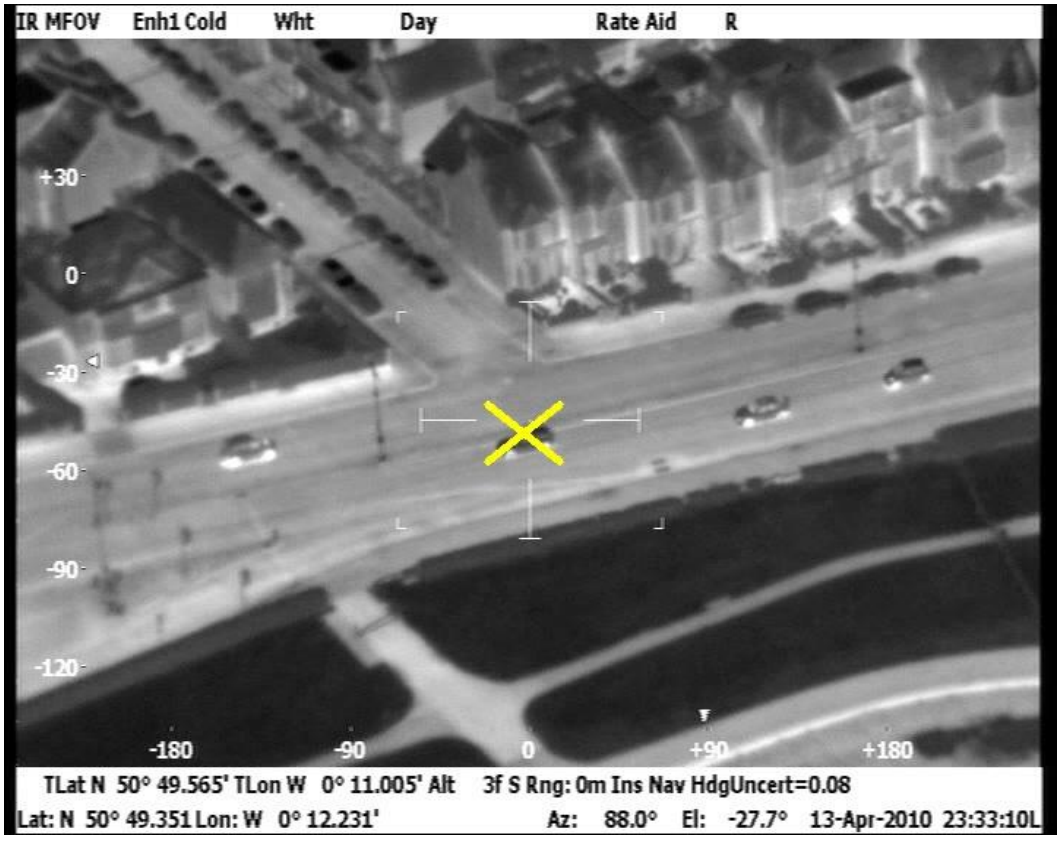

Figure 5.34: Sussex Police infra-red video frame 435, dynamic scale and orientation changes

The Figure 5.35 shows a green coloured car with dynamic scale changes being tracked efficiently for several orientation changes.

Figure 5.35: Sussex Police video (3) frame 35 with a difficult coloured car being tracked

Figure 5.36 depicts a scenarios where, the target is being tracked despite evident change in the direction of motion along with dynamic scale changes as the car turns into a side road.

Figure 5.36: Sussex Police video (3) frame 548, scale and orientation changed

Figure 5.37 and 5.38 shows a typical tracking scenarios provided by the DSTL for testing the ability of the tracker to track targets during noisy frames combined with orientation and scale changes. The OT-MACH tracker shows considerable promise and has the capability to perform accurately in cluttered and noisy sequences as demonstrated in the Figure 5.37 and Figure 5.38 below.

Figure 5.37: Noisy DSTL video (3) frame 15, scale and orientation changed

Figure 5.38: Noisy DSTL video (3) frame 75, continuous tracking during scale and orientation changes

5.11 Summary

In this chapter an enhanced active contour based OT-MACH tracker has been optimised and implemented as a robust vehicle tracker. The filter is rotation multiplexed and applied at a frame rate initialised by the user on the video sequences, with a filter up-date being implemented every m^{th} frame. On the video sequences for which it has been tested, a few typical examples have been displayed in previous sections. The algorithm can switch from using rectangular, circular or active contour based extraction methods. Compared to rectangular and circular extraction methods the active contour snake is found to allow the maintenance of a strong and accurate correlation peak at the target location. The user interface is designed so as to allow the user to interrupt and select a new target from a desired current frame. The OT-MACH filter is dynamically and frequently updated by retraining with rotationally multiplexed reference images extracted and processed during an interval period chosen by the user. An extended Kalman filter predictor is implemented and utilised to conditionally solve the problem of multiple targets in the scene. The Kalman filter as described in section 5.8 is useful for disambiguation when multiple targets are in close proximity and in application to constant velocity targets; hence running the two algorithms in parallel is a useful approach to solve the problem of tracking multiple target-like objects. The results obtained have been discussed and illustrated in section 5.9. From the tests performed on Sussex police and DSTL videos, the OT-MACH tracker shows considerable promise and has the capability to perform accurately in cluttered and noisy sequences. It is found to be accurate in recognising and tracking the target, out-performing an extended Kalman filter (EKF) [101] and colour based Particle filter (PF) [69] approach in noisy, velocity variant and dynamic sequences.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this thesis adaptive object segmentation and tracking models and methods based on the enhanced active contour snake (EACS) technique and the optimal trade-off maximum average correlation height (OT-MACH) filter have been studied, designed and developed to address the need for robust object tracking in complex scenarios. It has been demonstrated that active contour techniques can be exploited to perform the task of deformable object segmentation and tracking to fulfil typical moving deformable object segmentation and tracking system requirements. In challenging circumstances, such as in the presence of similar moving deformable objects in dynamic scenarios, the enhanced active contour techniques are used to assist an optimal trade-off maximum average correlation height (OT-MACH) filter to accomplish the realisation of a robust correlation pattern recognition based tracking system.

In the first part of this thesis, a computational model (CM) has been developed that can be used to adaptively segment and track deformable objects in motion using enhanced active contour snake (EACS) techniques. The EACS computational model is extended for multiple moving object segmentation and tracking. The EACS based statistical measurement parameters are also discussed in detail, along with real-time applications demonstrating the use of the active contour statistical measures.

In the second part of this thesis the EACS computational model is employed to assist an optimal trade-off maximum average correlation height (OT-MACH) filter to perform as a robust and variable speed object tracker. Several other methods for defining the area

of support for the OT-MACH filter, such as rectangular and circular target extraction methods, have been discussed, also. The OT-MACH based tracker has been evaluated on several video sequences from the Sussex police and the DSTL fast moving vehicle video datasets for its accuracy and performance in challenging scenarios.

Chapter 2 of this thesis introduces a modified computational model for active contours which is fast, robust and accurate, as compared to other available methods for active contour snake models. The developed computational model is based on a difference of Gaussian (DoG) filtered image thus emphasising on the exterior edges of the object under consideration. The DoG filter is designed using a pre-defined set of standard deviation values adhering to the rule that their ratio is equal to 1.6 so as to produce a bandpass filter. It is demonstrated that the active contours are initialised as soon as the object in a sequence is in motion and the contour thus obtained follows the object by deforming onto it during each subsequent frame. The introduction of the new energy term into the energy minimisation equation of the active contours is to emphasise the object edges and has been described in detail. The method of automatic contour initialisation has been introduced for automatic segmentation and tracking of moving objects in video sequences without user intervention. It has been demonstrated that due to the automatic initialisation of the contour, the snake requires fewer iterations to acquire the boundary of the object being tracked. It is also found that the method not only reduces computational time but also makes it accurate, as the initial contour fits readily into the object exteriors. It is reported that the presence of the DoG energy term in the energy minimisation equation makes the snake progression accurate and fast as compared to the classical active contour methods. It is also discussed that the designed computational model works when the object being tracked stops moving and becomes static. It is concluded that the enhanced active contour snake model discussed in this

chapter works on a single moving object. The supporting results presented demonstrate that the enhanced active contour snake framework performs accurately in real-time for variable scale and orientations of the object being contoured.

In Chapter 3 of this thesis, an extension to the enhanced active contour snake technique to perform multiple object segmentation and tracking is developed. The termination criteria for the contouring process is described and found to enhance the computational efficiency. It is demonstrated that the active contour vector tracking a group of occluded moving objects can be sub-divided into distinct contour vectors that can maintain a separate track of each moving object in the scene. The contour splitting and selection criteria responsible for allowing active contours to split and maintain distinct contour vectors for multiple objects are described in detail. The results documented demonstrate that the enhanced active contour model is suitable for robust multiple object tracking in real-time situations. The statistical parameters of the EACS model, such as the active contour boundary rectangle and the active contour centroid, are discussed in detail. Several results are discussed to demonstrate the use of these statistical measures. In order to evaluate and demonstrate the importance of computing accurate active contour vectors using the EACS model, a simple application to perform upper body humanoid skeletal modelling has been designed and described in detail. Several results showing the skeletal stick model for the upper body of a humanoid are presented and discussed.

In Chapter 4 of the thesis, the overall enhanced active contour vector orientation estimation method known as the global active contour orientation (GACO) measure is discussed in detail. It is demonstrated that by using simple geometric calculations and assumptions, it is possible to determine the direction of motion of a selected contoured object using the GACO measure technique. The mathematical calculations of the GACO statistical measure are described by choosing appropriate quadrants in which the

snaxel movements occur. The local snaxel orientations are computed using the detailed expressions. It is concluded that the GACO is the mathematical average of the local snaxel orientations and can be computed using these expressions. The GACO measure is further described by the use of a detailed design flowchart together with a several results. In order to evaluate the performance and importance of the proposed GACO measure, a robust algorithm for door surveillance is presented in this chapter. It is shown that an omni-directional edge-map based Harris corner detector can be used to provide the primary alarm during a door activity. The algorithm is designed to use the Harris corner data to decide whether the door is displaced to produce a primary alarm. The secondary alarm determining whether objects entered or moved when the door is displaced is designed and demonstrated using the global active contour orientation module. The improvement made to the door surveillance system by the addition of a Hough transform error correction module to append the Harris and surveillance window locations is developed and described in detail. The overall performance of the door surveillance application is evaluated on several i-LIDS door-zone dataset sequences. The results obtained are reported to evaluate the door surveillance system based on a GACO measure to show a success rate of almost 92% on several i-LIDS door-zone dataset scenarios.

In Chapter 5, an enhanced active contour assisted OT-MACH tracker is optimised and implemented as a robust moving vehicle tracker. The rotational multiplexing and dynamic filter update techniques are discussed in detail. The flexibility of the designed algorithm to switch between rectangular, circular and active contour based target extraction for filter training is described in detail. It is concluded that, compared to the rectangular and circular target extraction methods, the active contour based extraction is able to assist in maintaining a strong and accurate correlation peak at the target locations

recognised. It is demonstrated that the OT-MACH filter can be dynamically and frequently updated by retraining with rotationally multiplexed reference images extracted and processed during an interval period defined by the user. An extended Kalman filter predictor is found to conditionally solve the problem of confusion due to multiple targets in the scene. It is demonstrated that the Kalman filter applied in parallel with the OT-MACH tracker is useful in disambiguation when multiple targets are in close proximity at constant velocity thus solving the problem of tracking multiple target-like objects in the scene. The results obtained are discussed and illustrated. Several tests performed on Sussex police and DSTL (an acronym for the Defence Science and Technology Laboratory) videos are documented to conclude that the OT-MACH tracker shows considerable promise and has the capability to perform accurately in cluttered and noisy sequences. Thus the OT-MACH tracker is capable of accurately tracking a target object, out-performing an extended Kalman filter and colour based Particle filter approach in noisy, velocity variant and dynamic scenarios.

6.2 Future work

The design and implementation of an enhanced active contour assisted OT-MACH filter tracker has yielded promising results for several challenging videos provided by the Sussex police and DSTL tracking video dataset. It has been observed that the tracker is found to be dependant on several parameters as described in Table 6.1.

Type of videos	Changes to parameters	Ideal parameters
Ideal colour videos	• Lower DoG filter patch size. • Threshold value higher than 100 below 255. • α-OT parameter lower than 10^{-4}	• DoG filter patch size 8 • Threshold value 125-145 • α = 0.00005
Noisy colour videos	• Lower DoG filter patch size. • Threshold value higher than 150 below 255. • α-OT parameter lower than $5x10^{-4}$ • Needs low pass Gaussian filter to remove noise	• DoG filter patch size 10-15 • Threshold value higher than 180. • α= 0.0005 • Low pass Gaussian filter is applied to the target frame before filtering.
IR videos	• Higher DoG filter patch size. • Threshold value higher than 50 below 100. • α-OT parameter lower than 10^{-5} • Needs low pass Gaussian filter to remove noise	• Higher DoG filter patch size 20-30 • Threshold value 50-90 • α = 0.00001 • Low pass Gaussian filter is applied tn the target frame before filtering.
Noisy IR videos	• Higher DoG filter patch size. • Threshold value higher than 50 below 100. • α-OT parameter lower than 10^{-5} • Needs low pass Gaussian filter to remove noise • Quicker filter update and correlation rate	• Higher DoG filter patch size 20-30 • Threshold value 50-90 • α = 0.00001 • Low pass Gaussian filter is applied to the target frame before filtering. • Quicker filter update and correlation rate

Table 6.1: OT-MACH filter tracker ideal parameters and changes required for different scenarios

From the Table 6.1, it can be concluded that the tracker parameters, such as the difference of Gaussian filter scale and threshold values, need changing for different

kinds of scenarios. Hence, there is a need to further develop the tracker into a completely automatic tracking system without the requirement for user intervention in setting the filter parameters. One possible approach to accomplish this can be the use of a multi-layer perceptron based neural network prediction system. A multi-layer perceptron (MLP) can be trained for a set of videos to predict possible parameter values in order to accomplish robust tracking. Several weighting conditions for the neural network algorithm can be derived by analysing the object being considered for tracking along with the characteristics of the environment in which the target is to be tracked. The OT-MACH filter parameters, such as α, can be estimated by analysing the correlation plane data obtained for previous frames.

Further work could consider modifications in the area of support for more robustly resolving multiple target confusion. Current techniques such as the SIFT (scale invariant feature transform) [102] and SURF (speeded up robust feature detector) [103] algorithms for scale invariant feature recognition could be considered for enhancing the performance of the OT-MACH tracker. Combinations of these methods may allow the target to be continuously tracked even in the presence of close proximity multiple target-like objects in the scene and will help provide a predictive track to assist the re-acquisition of the target vehicle if the track is interrupted. A new approach to train a Particle filter using the correlation plane result obtained from the OT-MACH tracker could also be investigated to improve the accuracy and the ability to track in multi-target scenarios. Further study could also be conducted in the area of low pixel count object detection and tracking in low signal-to-noise ratio conditions to assist in real-time target tracking applications from an extended range.

Appendix 1

A complete snake implementation of Kass Snake [2],[3] solves for all snake points in one step to ensure that the snake moves to the best local energy minimum.

The snake points chosen are in such a manner so as to make sure that energy is minimised. The snake points chosen can be represented as:

$$v(s) \equiv (x(s), y(s)) \tag{A.1}$$

By the calculus of variations and considering an admissible solution $\hat{v}(s)$ perturbed by a small amount, $\varepsilon\delta v(s)$, a minimum energy can be achieved:

$$\frac{dE_{snake}\left(\hat{v}(s) + \varepsilon\delta v(s)\right)}{d\varepsilon} = 0 \tag{A.2}$$

The perturbation is spatial and affects the x and y co-ordinates of a snake point as:

$$\delta v(s) = \left(\delta_x(s), \delta_y(s)\right) \tag{A.3}$$

Equation (A.3) gives the perturbed snake solution as:

$$\hat{v}(s) + \varepsilon\delta v(s) = \left(\hat{x}(s) + \varepsilon\delta_x(s), y(s) + \hat{\varepsilon\delta}_y(s)\right) \tag{A.4}$$

According to Kass [1,2], the energy functional is an integral; s ϵ [0, 1] is the normalised length around the snake. The energy functional is then given as:

$$E_{snake} = \int_{s=0}^{1} E_{\text{int}}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))ds \tag{A.5}$$

where, E_{int} is the internal energy, E_{image} is the image energy or the external energy and E_{con} is the constraint energy [3]. By substituting the constraint energy E_{con} and forcing to zero, the snake energy becomes:

$$E_{snake}(v(s)) = \int_{s=0}^{1} \left\{ E_{\text{int}}(v(s)) + E_{image}(v(s)) \right\} ds \qquad \text{(A.6)}$$

Edge magnitude information is often used to make the snake attracted to the edges in the image. Here, we consider a DoG filtered edge image:

$$E_{image} = c.\left(E_{in} + E_{grad} + E_{DoG} \right) \qquad \text{(A.7)}$$

where, E_{DoG} is the DoG filtered energy term, E_{in} and E_{grad} are the energy due to the intensity and gradient at that pixel point. Substituting the same for a perturbed snake point we get:

$$E_{snake}(v(s)) = \int_{s=0}^{1} \left\{ E_{\text{int}}(v(s)) + E_{image}(v(s)) \right\} ds \qquad \text{(A.8)}$$

By expanding E_{image} at the perturbed solution by a Taylor series, we obtain for E_{DoG},

$$E_{DoG}\left(\hat{v}(s) + \varepsilon\delta v(s) \right) = E_{DoG}\left(\hat{x}(s), \hat{y}(s) \right) + \varepsilon\delta_x(s) \left. \frac{\partial E_{DoG}}{\partial x} \right|_{\hat{x},\hat{y}} + \varepsilon\delta_y(s) \left. \frac{\partial E_{DoG}}{\partial y} \right|_{\hat{x},\hat{y}} + O(\varepsilon^2) \qquad \text{(A.9)}$$

This implies that the image information must be twice differentiable which holds for edge information, but not for other forms of image energy (ignoring higher order terms since ϵ is small). Since the perturbed solution is at a minimum, the integration terms must be identically zero. By integrating with constants we get:

$$E_{snake}(s) = a(s)\left| \frac{dv_s}{ds} \right|^2 + b(s)\left| \frac{d^2 v_s}{ds^2} \right|^2 + c(s)E_{\text{Im}age} \qquad \text{(A.10)}$$

Equation (A.10) is in the general form of the Greedy algorithm for snakes, and is used to accurately contour the exterior edges of deformable objects or targets (in this particular application). The constant $a(s)$ controls the contribution of the elastic energy due to point spacing. The second order differential is weighted by constant $b(s)$ which controls the contribution of the curvature energy due to point variation. The image energy E_{image} is weighted by constant $c(s)$, controlling the external image energy emphasising on the exterior edges of the object while contouring.

The equation (A.7) can be modified to contain three different constant values controlling the emphasis given to each energy term while computing the overall energy of the Snake. This can be expressed as:

$$E_{image} = c_1 E_{in} + c_2 E_{grad} + c_3 E_{DoG} \qquad \text{(A.11)}$$

The DoG filtered image energy, E_{DoG}, is weighted by constant c_3 controlling the emphasis on exterior edges or the negative value around the zero-crossing located at the edge of the object being contoured. Several variations of the Snake are possible by adjusting the energy terms and the constants [2].

References

[1] R.C. Gonzalez and R.E. Woods, *Digital Image Processing.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992.

[2] M. Nixon and A.S. Aquado, *Feature Extraction and Image Processing*, 2nd ed. Hungary: Academic Press, 2007

[3] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, "Snakes: Active Contour Models", *International Journal of Computer Vision,* Volume 1, pp. 321-331, 1988.

[4] A.L. Yuille, "Deformable templates for face recognition", *Journal of Cognitive Neuroscience,* Volume 3, pp. 59-70, 1991.

[5] D. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning,* Addison-Wesley, 1988.

[6] I. Cohen, L.D. Cohen and N. Ayache, "Using deformable surfaces to segment 3D images and inter differential structures", *CVGIP: Image Understanding*, Volume 56, Issue 2, pp. 242-263, 1992.

[7] A.K. Jain, Y. Zhong and M-P. Dubuisson-Jolly, "Deformable template models: a review", *Signal Processing*, Volume 71, pp. 109-129, 1998.

[8] K.F. Lai and R.T. Chin, "Deformable contours – Modelling and extraction", *IEEE Transaction on PAMI,* Volume 17, Issue 11, pp. 1084-1090, 1995.

[9] A. Lanitis, C.J. Taylor and T. Cootes, "Automatic interpretation and coding of face images using flexible models", *IEEE transaction on PAMI,* Volume 19, Issue 7, pp. 743-755, 1997.

[10] T.F. Cootes, A. Hill, C.J. Taylor, D.H. Cooper and J. Graham, “Active shape Models – their training and application”, *Computer Vision and Image Understanding- CVIU,* Volume 61, Issue 1, pp. 38-59, 1995.

[11] D. Geiger, A. Gupta, L.A. Costa and J. Vlontsos, “Dynamical Programming for Detecting, Tracking and Matching Deformable contours”, *IEEE transactions on PAMI,* Volume 17, Issue 3, pp. 294-302, 1995.

[12] T. McInerney and D. Terzopolous, “Deformable models in Medical Image Analysis, a survey”, *Medical Image Analysis,* Volume 1, Issue 2, pp. 91-108, 1996.

[13] R. Ronfard, “Region-based Strategies for Active Contour Models”, *International Journal of Computer Vision,* Volume 13, Issue 2, pp. 229-251, 1994.

[14] J.R. Parker, *Practical Computer Vision using C,* Wiley and Sons Inc., NY, USA, 1994.

[15] Chenyang Xu, Jerry L. Prince, “Snakes, Shapes, and Gradient Vector flow”, *IEEE Transactions on Image processing,* Volume 7, Issue 3, pp. 359-369, March 1998.

[16] F. Mohanna and F. Mokhtarian, "Improved curvature estimation for accurate localization of active contours," in *Proc. International Conference on Image Processing (ICIP'01),* Volume 2, pp. 781-784, Thessaloniki, Greece, October 2001.

[17] Min hu, Xijian Ping, Yihong Ding, “Automated Cell Nucleus Segmentation using Improved Snake”, *International Conference on Image Processing,* Volume 4, pp. 2737 - 2740 (ICIP), October 2004.

[18] S. Lefevre, J.G.Gerard, A.Piron, N.Vincent,"An Extended Snake model for Real-time multiple Object tracking", *RFAI publication: International workshop on Advanced concepts for Intelligent vision systems*, Ghent, Belgium, pp. 268-275, September 2002.

[19] D. Williams, M. Shah, "A Fast Algorithm for Active Contours and Curvature Estimation", *Computer Vision Graphics and Image Processing: Image Understanding*, Volume 55, pp. 14-26, 1992.

[20] J. Hernandez, F. Prieto, T. Redarce, "Fast Active Contours for Sampling", *CERMA'06 Conference, IEEE Computer Society*, Volume 2, pp. 9-13, September 2006.

[21] G. Xu, E. Segawa, and S. Tsuji, "Robust Active Contours using Insensitive Parameters", *Pattern Recognition*, Volume 27, Issue 7, pp. 879-884, 1994.

[22] R. Ronfard, "Region-based strategies for active contour models", *International Journal of Computer Vision*, Volume 13, Issue 2, pp. 229-251, 1994.

[23] M. Writh, D. Nikitenko, J. Lyon, "Segmentation of the Breast region in Mammograms using a rule based fuzzy reasoning algorithm", *ICGST International Journal of graphics, vision and image processing*, Volume 5, Issue 2, January 2005.

[24] W.E.S. David, M. Honea and G. Bilbro, "Active Contours using a potential filed", *ICPR02: International Conference on Pattern Recognition*, Volume 2, pp.20757, August 2002.

[25] H. Yang, Y. Tan, J. Tian, J. Liu, "Accurate Dynamic Scene Model for Moving Object Detection", *ICIP 2007, IEEE*, Volume 6, pp. 157-160, September 2007.

[26] C. P. Lee, W. Snyder, C. Wang, " Supervised Multispectral Image Segmentation using Active Contours", *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 4242-4247, April 2005.

[27] Farag Cao, Y. Rose, D.M. Delp, E. J, "On empirical estimation of the parameters of edge enhancement filters", *International Conference on System, Man and Cybernetics*, Volume 1, pp. 346-350, October 1992.

[28] Rajeev Ratan, Sanjeev Sharma, S.K. Sharma, "Brain tumor detection based on multi-parameter MRI image analysis", *ICGST International Journal of Graphics, vision and Image processing*, Volume 9, issue 3, June 2009.

[29] S.R. Gunn and M.S. Nixon, "A Robust Snake Implementation: a Dual Active contour", *IEEE transactions on PAMI*, Volume 19, Issue 1, pp. 63.68, 1997.

[30] Philip Birch, Bhargav Mitra, Nagachetan Bangalore, Saad Rehman, Rupert Young, Chris Chatwin, "Approximate bandpass and frequency response models of the difference of Gaussian filter", *Optics Communications*, vol. 283, no. 24, pp. 4942-4948, December 2010.

[31] Nagachetan Bangalore, Rupert Young, Philip Birch, Chris Chatwin, "Tracking moving objects using bandpass filter enhanced localisation and automatic initialisation of active contour snakes", *ICGST International Journal of Graphics, Vision and Image Processing GVIP*, Vol. 10, no. 4, pp. 1-8, October 2010.

[32] M. O. Berger, "Towards dynamic adaption of Snake contours", *Proceeding of 6th International conference on Image Analysis and Processing*, Como, Italy, pp. 47-54, 1991.

[33] B. McCane, "*Edge Detection",* Lecture-Notes, Department of Computer Science, University of Otago, Dunedin, New Zealand, February 20, 2001.

[34] J. Ivins and J. Porrill, "Active Region models for segmenting Textures and colours", *Image and Vision Computing,* Volume 13, Issue 5, pp. 431-437, 1995.

[35] K. F. Lai and R.T. Chin, "On regularisation, extraction and initialisation of the Active Contour model (Snakes)", *Proceedings of First Asian Conference on Computer Vision,* pp. 542-545, 1994.

[36] S. Zhang and M. A. Karim, "A new impulse detector for switching median filters", *IEEE Signal Processing Letters,* Volume 9, pp. 360 – 363, 2002.

[37] B. K. Mitra, W. Hassan, P. Birch, A. Gardezi, R. Young, C.R. Chatwin, "A Two-Stage Approach to Detect Abandoned Baggage in Public Places", *Visual Information Processing XIX, SPIE Defense, Security + Sensing,* Orlando, Florida, USA, 2010.

[38] Duda, R.O., Hart, P.E., "Use of Hough Transformation to Detect Lines and Curves in Pictures*," Communications of ACM,* Volume 15, pp. 11-15, 1972.

[39] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", *Pattern Recognition,* Volume 13, Issue 2, p. 111-122, 1981.

[40] B. K. Mitra, P. Birch, I. Kypraios, R.Young. C. R. Chatwin,"On a Method to Eliminate Moving Shadows from Video Sequences", in *Proc. of SPIE Photonics Europe- Optical and Digital Image Processing*, Volume 7000, pp. 700012-1:9, Strasbourg, France April 7 – 10, 2008.

[41] D. Marr, E. Hildreth, "Theory of Edge detection", *Proceedings of the Royal Society of London. Series B, Biological Sciences*, Volume 207, Issue 1167, pp.187-217, 1980.

[42] B. K. Mitra, R. Young, C. R. Chatwin, "On shadow elimination after moving region segmentation based on different threshold selection strategies", *Optics and Lasers in Engineering*, Volume 45, pp. 1088 – 1093, November 2007.

[43] Home Office, Scientific Development branch in partnership with Security Services, UK, *Imagery library for intelligent detection systems (i-LIDS)*.

[44] Context Aware Vision using Image-based Active Recognition - 'CAVIAR' test case scenarios, July 2003 & Jan 2004.

[45] Chris Harris, Mike Stephens, "A combined corner and edge detector", *Plessey Research Roke Manor*, UK, The Plessey Company Plc, 1988.

[46] Miroslav Trajkovi, Mark Hedley, "Fast Corner Detection", *Image and Vision Computing*, Volume 16, Issue 2, pp. 75-97, February 1998

[47] Robert Laganiere, "A morphological operation for corner detection", *Pattern Recognition*, Volume 31, Issue 11, pp 1643-1652, November 1998

[48] Nagachetan Bangalore, Waqas Hassan, Bhargav Mitra, Philip Birch, Rupert Young, Chris Chatwin, "Door surveillance using edge map-based Harris corner detector and active contour orientation", in *Proc. of Visual Information processing XX, SPIE Defense, Security + Sensing,* vol. 8056, Orlando, Florida, USA, April 2011, pp. 805608-1:10.

[49] Waqas Hassan, Nagachetan Bangalore, Philip Birch, Rupert Young, Chris Chatwin, "Object Tracking in a Multi Camera Environment," in Proc. of *IEEE International Conference on Signal and Image Processing Applications (ICSIPA 2011),* Kuala Lumpur, November 2011.

[50] U. Muehlmann, P. Lang, and A. Pinz, ”A new high speed CMOS camera for real-time tracking applications”, *Robotics and Automation, Proceedings, ICRA*, Volume 5, pp. 5195-5200, April 2004.

[51] Ahmad Alkandri, Akber Gardezi, Nagachetan Bangalore, Philip Birch, Rupert Young, Chris Chatwin, “Automatic parameter adjustment of difference of Gaussian (DoG) filter to improve OT-MACH filter performance for target recognition applications”, in Proc. of *Electro-Optical and Infrared Systems: Technology and Applications VIII, SPIE Europe Security+Defense*, vol. 8185, Prague, Czech Republic, September 2011, pp. 81850M-1:10.

[52] Waqas Hassan, Nagachetan Bangalore, Bhargav Mitra, Philip Birch, Rupert Young, Chris Chatwin, “Robust human intrusion detection technique using hue-saturation histograms”, in *Proc. of Optical Pattern Recognition XXII, SPIE Defense, Security + Sensing*, vol. 8055, Orlando, Florida, USA, April 2011, pp. 80550J-1:12.

[53] Bhargav Mitra, Waqas Hassan, Nagachetan Bangalore, Philip Birch, Rupert Young, Chris Chatwin, “Tracking illegally parked vehicles using correlation of multi-scale difference of Gaussian filtered patches”, in *Proc. of Optical Pattern Recognition XXII, SPIE Defense, Security+Sensing*, vol. 8055, Orlando, Florida, USA, April 2011, pp. 805503-1:9.

[54] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms”, *IEEE Transactions on Systems, Man and Cybernetics*, Volume 9, Issue 1, pp. 62-66, 1979.

[55] Bhargav Mitra, Nagachetan Bangalore, Waqas Hassan, Philip Birch, Rupert Young, Chris Chatwin, “Change of illumination tolerant scene surveillance using

a multi-stage edge detector", *Asian Journal of Physics*, Vol 19, No. 1, pp. 87-97, 2010.

[56] Akber Gardezi, Nagachetan Bangalore, Ahmad Alkandri, Philip Birch, Rupert Young, Chris Chatwin, "Application of speed-enhanced spatial domain correlation filters for real-time security monitoring", in *Proc. of Optics and Photonics for Counterterrorism and Crime Fighting VII, SPIE Europe Security+Defense*, vol. 8189, Prague, Czech Republic, September 2011, pp. 81890R-1:12.

[57] Waqas Hassan, Bhargav Mitra, Nagachetan Bangalore, Philip Birch, Rupert Young, Chris Chatwin, "Image processing methods for event detection from video surveillance sequences", *Information Technologies, System and Networks (ITSN-2010)*, Moldova, May 2010.

[58] Philip Birch, Waqas Hassan, Nagachetan Bangalore, Rupert Young, Chris Chatwin, "Stationary Traffic Monitor," *4th International Conference on Imaging for Crime Detection and Prevention (ICDP-11)*, London, November 2011.

[59] C. Dong and Y. Dong "A Novel Algorithm for Object Tracking with Particle Filtering and GVF-Snake," *Intelligent Systems Design and Applications*, pp. 8-13, 2008.

[60] Horprasert, T., Harwood, D., Davis, L. S., 'A statistical approach for real-time robust background subtraction and shadow detection,' in *Proc. IEEE International Conference on Computer Vision,* '99 FRAME-RATE Workshop, 1999.

[61] A. Aksel, S.T. Acton, "Target tracking using the snake particle filter," *Image Analysis & Interpretation (SSIAI)*, pp. 33-36, 2010.

[62] Y. Rathi, N. Vaswani, A.Tannenbaum, A. Yezzi, "Tracking Deforming Objects Using Particle Filtering for Geometric Active Contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 29, pp. 1470-1475, 2007.

[63] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, Volume 50, pp. 174-188, 2002.

[64] C. Harris and M.J. Stephens, "A combined corner and edge detector", in *Alvey Vision conference*, pp. 147-152, 1988.

[65] Churchille, E., McConville, J. T., Laubach, L. L. Churchill, T., Erskine, P., & Downing, K., *Anthropometric Source Book - Volume II: A Handbook of Anthropometric Data. (NASA Reference Publication 1024).* NASA, Scientific and Technical Information Office, 1978.

[66] A. Kar, "Skeletal Tracking using Microsoft Kinect The Microsoft Kinect sensor", *Methodology*, pp. 1-11, 2010.

[67] Y. Yang, C. Yang; X. Wang. 'A Detection and Tracing Algorithm of Moving Vehicles', *2nd International Conference on Pervasive Computing and Applications*, 2007. ICPCA 2007

[68] A. Dearden and Y. Demiris, "Tracking football player movement from a single moving camera using particle filters," *In Proceedings of the 3rd European Conference on Visual Media Production*, pp. 29–37, 2006.

[69] Y. Movshovitz-Attias, S. Peleg, "Bacteria-Filters: Persistent particle filters for background subtraction," *17th IEEE International Conference on Image Processing (ICIP)*, pp. 677-680, 2010.

[70] Z. Yun-feng, S. Gan-lin, J. Bing, 'A dynamic-template-library based method to measure the pose of manoeuvring target', *4th IEEE Conference on Industrial Electronics and Applications,* ICIEA 2009.

[71] Erhan Bas, A. Murat Tekalp, F. Sibel Salman, "Automatic Vehicle Counting from Video for Traffic Flow Analysis", *IEEE Intelligent Vehicles Symposium*, pp. 392 – 397, June 2007.

[72] P. M. Crisóstomo-Romero, "Vehicle counting system using real-time video processing", *Proceedings of SPIE, Real-Time Image Processing*, Volume 6063, pp. 66-73, February 2006.

[73] M. M. Zadeh, T. Kasvand, C. Y. Suen, "Localization and Recognition of Traffic Signs for Automated Vehicle Control Systems", *Proceedings of SPIE, Intelligent Transportation Systems*, Volume 3207, pp. 272-282, January 1998.

[74] I. Kypraios, R. C. D. Young, P. Birch, C. Chatwin, "Object recognition within cluttered scenes employing a Hybrid Optical Neural Network (HONN) filter", *Optical Engineering, Special Issue on Trends in Pattern Recognition*, Volume 43, Issue 8, pp. 1839-1850, 2004.

[75] Ph. Refregier, "Optimal trade-off filters for noise robustness, sharpness of the correlation peak and Horner efficiency", *Optics Letters*, Volume 16, Issue 11, pp. 829-831, 1991.

[76] L. S. Jamal-Aldin, R. C. D. Young, and C. R. Chatwin, "Synthetic discriminant function filter employing nonlinear space-domain pre-processing on bandpass-filtered images", *Applied Optics*, Volume 37, Issue 11, pp. 2051-2062, 1998.

[77] L. S. Jamal-Aldin, R. C. D. Yound, and C. R. Chatwin, "In-class distortion tolerance, out-of-class discrimination and clutter resistance of correlation filters

that employ a space domain non-linearity applied to wavelet filtered input images", *Proc. of SPIE,* Volume 3386, pp. 111-122, 1998.

[78] C. Wren, A. Azabayejani, T. Darrel, and A. Pentland, Pfnder: Real-time tracking of the human body," IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 780-785, 1997.

[79] Ph. Refregier, "Filter design for optical pattern recognition: multicriteria optimization approach", *Optics Letters,* Volume 15, Issue 15, pp. 854-856, 1990.

[80] Miller, A., Shah, M., "Foreground segmentation in surveillance scenes containing a door," *IEEE International conference on Multimedia and Expo,* pp. 1822-1825, August 2007

[81] Power, W.P., Schoonees, J.A., "Understanding background mixture models for foreground segmentation," *Proceedings of Image and Vision Computing, Newzealand,* 2002.

[82] Stoeter, S.A. et al, "Real-time door detection in cluttered environments," *IEEE International symposium on Intelligent Control,* July 2000.

[83] Anguelov, D. et al, "Detecting and modelling doors with mobile robots," *Proceedings on ICRA,* 2004.

[84] E. L. Schwartz, D. Greve, G. Bonmasser, "Space-variant active vision: definition, overview and examples," *Neural Networks,* Volume 8, Issue 7, pp. 1297-1308, 1995.

[85] P. Bone, R. C. D. Young, C. Chatwin, "Position-, rotation-, scale- and orientation-invariant multiple object recognition from cluttered scenes", *Optical Engineering,* Volume 45, Issue 7, pp. 077203-1:8, July 2006.

[86] C. G. Ho, R.C.D. Young, C.R. Chatwin, “Sensor Geometry and Sampling Methods for Space-Variant Image Processing”, *Journal of Pattern Analysis and Applications*, Volume 5, pp. 69-384, 2002.

[87] B. Javidi, Ph. Refregier, J. Wang, P. Willett, “Pattern recognition with spatially disjoint target and scene noise”, *Real-Time Optical Information Processing, Academic*, New York, pp. 3-38, 1994.

[88] O. Gualdron, H. H. Arsenault, “Optimum rotation-invariant filter for disjoint noise scenes”, *Applied Optics*, Volume 35, Issue 14, pp. 2507-2513, May 1996.

[89] C.F. Hester and D. Casasent, “Multivariant technique for multiclass pattern recognition”, *Applied Optics*, Volume 19, pp. 1758-1761, 1980.

[90] A. Mahalanobis, B.V.K. Vijaya Kumar and D. Casasent, “Minimum average correlation energy filters”, *Applied Optics*, Volume 26, pp. 3363-3640, 1986.

[91] R. Young, C.R. Chatwin, and S.B., “High speed hybrid optical/digital correlator system”, *Optical Engineering*, Volume 32, pp. 2608-2615, 1993.

[92] A. Mahalanobis, B.V.K. Vijaya Kumar, S. Song, S.R.F. Sims, and J.F. Epperson, “Unconstrained correlation filters”, *Applied Optics*, Volume 33, Issue 17, pp. 3751-3759, 1994.

[93] B.V.K. Vijaya Kumar, A. Mahalanobis, and R. D. Juday, *Correlation Pattern Recognition*, United Kingdom, Cambridge University Press, 2005.

[94] Frigo M, Johnson SG, "The design and implementation of FFTW3", FFTW – *Fastest Fourier Transform in the West*, Proceedings of the IEEE, Volume 93, Issue 2, pp. 216–231, 2005.

[95] R.A. Kerekes and B.V.K. Vijaya Kumar, “Selecting a composite correlation filter design: a survey and comparative study”, *Optical Engineering*, Volume 47, Issue 6, pp. 067202-1:10, 2008.

[96] B.V.K. Vijaya Kumar, “Minimum variance synthetic discriminant functions”, *Journal of Optical Society of America A*, Volume 3, pp. 1579-1584, 1986.

[97] G.F. Schils and D.W. Sweeney, “Rotationally invariant correlation filtering”, *Journal of Optical Society of America A*, Volume 2, pp.1411-1418, 1985.

[98] B.V.K. Vijaya Kumar, A. Mahalanobis and A. Takessian, “Optimal trade-off circular harmonic function correlation filter methods providing controlled in-plane rotation response”, *IEEE Transations on Image Processing*, Volume 9, pp. 1025-1034, 2000.

[99] R. Wu and H. Stark, “Rotation-invariant pattern recognition using a vector reference”, *Applied Optics*, Volume 23, pp. 838-843, 1984.

[100] P.A. Molley and B.A. Kast, “Automatic Target Recognition and Tracking using an acousto-optic image correlator”, *Optical Engineering*, Volume 31, pp. 956-962, 1992.

[101] S.J. Julier, J.K. Uhlmann, “Unscented filtering and nonlinear estimation”, *Proceedings of the IEEE*, Volume 92, Issue 3, pp. 401-422, 2004.

[102] David Lowe, "Object recognition from local scale-invariant features", *Proceedings of the International Conference on Computer Vision*, Volume 2, pp. 1150-1157, 1999.

[103] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Volume 110, Issue 3, pp. 346-359, 2008