A HIGHLY ADAPTABLE MODEL BASED – METHOD FOR COLOUR IMAGE INTERPRETATION

by

MALIK SHEHADEH BRAIK

A thesis submitted to the University of Birmingham

for the degree of

DOCTOR OF PHILOSOPHY

SCHOOL OF ELECTRONIC, ELECTRICAL AND SYSTEMS ENGINEERING

UNIVERSITY OF BIRMINGHAM

November 2015

Abstract

This Thesis presents a model-based interpretation of images that can vary greatly in appearance. Rather than seek characteristic landmarks to model objects we sample points at regular intervals on the boundary to model objects with a smooth boundary. A statistical model of form in the exponent domain of an extended superellipse is created using sampled points and appearance by sampling inside objects.

A colour Maximum Likelihood Ratio criterion (MLR) was used to detect cues to the location of potential pedestrians. The adaptability and specificity of this cue detector was evaluated using over 700 images. A True Positive Rate (TPR) of 0.95 and a False Positive Rate (FPR) of 0.20 were obtained. To detect objects with axes at various orientations a variant method using an interpolated colour MLR has been developed. This had a TPR of 0.94 and an FPR of 0.21 when tested over 700 images of pedestrians.

Interpretation was evaluated using over 220 video sequences (640 x 480 pixels per frame) and 1000 images of people alone and people associated with other objects. The objective was not so much to evaluate pedestrian detection but the precision and reliability of object delineation. More than 94% of pedestrians were correctly interpreted.

Acknowledgements

My first debt of gratitude must go to my supervisor Mr. Pycock for leading me to this project and his time spent in providing me with ideas on this project and in reviewing this thesis. Patiently, he provided me with every bit of guidance; valuable feedback and assistance that I much needed throughout to complete this research. His delight and enthusiasm shown in the meetings during the research inspired me. He has taught me consciously how research can be achieved. I would like to thank my supervisor Mr. Pycock for teaching me how to capture video sequences of people walking and how to chat with them when seeking permission to photograph them.

I wish to express my deep thanks to my assessor Dr. Spann for his time, insightful questions and helpful comments. This valuable guidance has served me well. I am very grateful to the examiner Prof. Orwell for reading this Thesis and who was willing to help and give his best suggestions.

I gratefully acknowledge the international development bank and the Al-Balqa' Applied University for funding me to study Ph.D. My greatest gratitude must go to my friend Mr. Al-Adwan who patiently helped me to take the photographs for the participants and printed this Thesis.

I have kept this last word of acknowledgment to my wife Heba, who has been with me throughout my Ph.D. study. I would like to thank her for her support, love and great patience at all times. Thank you with my heart and soul.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AAAM	Adaptive Active Appearance Model
AAM	Active Appearance Model
AP	Average Precision
BP	Back-Propagation
BPNN	Back-Propagation Neural Network
CBIT	Context-Based Image Transmission
CMAT	Concordance-based Medial Axis Transform
CMG	Colour Morphological Gradient
CMU PIE	Carnegie Mellon University Pose, Illumination, and Expression
CNN	Convolutional Neural Network
DoG	Difference of Gaussian
DOT	Dominant Orientation Templates
DPM	Deformable Part-based Model
EOF	Error Of Fit
ESAM	Extended Superellipse Appearance Model
ESGM	Extended Superellipse Geometric Model
ESE	Extended SuperEllipse
ESQs	Extended SuperQuadrics
F	F-score rate
f_n	false negative
FNR	False Negative Rate
FOM	Figure Of Merit
f_p	false positive
FPPI	False Positives Per Image
FPPW	False Positives Per Window
FPR	False Positive Rate
fps	frames per sec
GLCM	Grey-Level Co-occurrence Matrix
HF	Hough-based Forest
HOG	Histograms of Oriented Gradients
HT	Hough Transform
IAS	Intensity Axis of Symmetry
KLT	Karhunen-Loeve Transform
LR	Likelihood Ratio
LV	left ventricular
MAP	Maximum A Posteriori
MAT	Medial Axis Transform
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimate
MLR	Maximum Likelihood Ratio
MMA	Multi-scale Medial Axis
MR	Magnetic Resonance

MRF	Markov Random Field
MSE	Mean Square Error
MVD	Minimum Vector Dispersion
NN	Nearest Neighbour
P	Precision rate
PCA	Principal Component Analysis
PDF	Probability Density Function
PDM	Point Distribution Model
PR	Precision-Recall
R	Recall rate
RANSAC	RANdom SAmple Consensus
RCMG	Robust Colour Morphological Gradient
RF	Random Forest
ROC	Receiver Operating Characteristic
SAT	Symmetry Axis Transform
SD	standard deviation
SE	standard error
SE	SuperEllipse
S3F	Simultaneous Segmentation and Superquadric Fitting
SNR	Signal to Noise Ratio
SQs	SuperQuadrics
TA-CNN	Task Assistant - Convolutional Neural Network
t_n	true negative
t_p	true positive
TPR	True Positive Rate
SVM	Support Vector Machine

Chapter 1 INTRODUCTION

1.1 The Importance of Image Interpretation

Images are important in image interpretation and understanding of the world. The interpretation of images helps to locate and recognise objects and people. To interpret an image that is a 2D projection of a 3D scene either assumptions must be made about the projection or a 3D calibration is needed. Non-model-based methods of image interpretation involve a series of steps: pre-processing, segmentation, feature extraction and classification. These approaches lead to a fast and reliable processing scheme when the image structure is simple, and the result of each step is distinct. However, such a sequential set of operations cannot be guaranteed to produce the desired result when the image structure is not well-defined. The simple sequential set of operations is fragile because it works as an open - loop system with no checks or corrections to the outcome of each step. Each program in a procedural method must be designed with care, and cannot readily be adapted from one application to another. It is not currently possible to accommodate, within one procedure, the ability to identify objects that vary greatly in appearance.

Traditional non-model-based approaches to image interpretation fail with modest changes in scene content and illumination. This was addressed in part, in early model-based methods where models were used to define the geometry and topology of the objects considered. Previous research has shown that model-based methods offer a great improvement in reliability over non-model-based methods [AIX03]. Early model-based methods were used to recognise pedestrians and a wide range of objects using models that often had to be customised to each object such as a model reported by Baumberg [BAU95] for pedestrian detection.

Moses and Ullman [MOS92] argued that non-model-based vision systems cannot correctly recognize objects in a consistent manner and constructed a mathematical proof of this based on a definition of consistent recognition functions invariant to viewing position and illumination conditions. The essential premise is that because different objects can produce similar looking images or image features, it is not possible to distinguish these objects without prior knowledge of how the images were formed. For example, to recognize an object of several views, a non-model-based system should be trained on all possible perspective views. On the other hand, a model-based system would not have to be trained on all possible view orientations. A model-based system might reasonably adapt to accomodate differents projections into 2D of a 3D object.

A model-based method was well-established as a powerful approach to recognising examples of known objects in new images in the presence of clutter, noise and occlusion. It is problematic to apply model-based methods to images of objects whose appearance can vary greatly. Often the structures to be located can vary in shape, either because they are flexible, articulated or because natural variation is present. A problem with existing methods is that they sacrifice model specificity in order to accommodate variability, thereby compromising robustness during image interpretation. A model-based system can learn patterns of considerable variability from a suitably annotated training set of typical images and should only be able to deform in ways characteristic of the class of objects it represents. In many practical situations objects of the same class might exhibit variation in shape. In such cases flexible models which allow for some degree of variability in the shapes of imaged objects, are often appropriate. Many applications of image interpretation typically require an automated system to understand the images with which it is presented. These images may provide noisy and typically deal with complex and variable struc-

ture. This necessarily involves the use of models which describe the expected structure of the images. Model-based methods offer potential solutions to these difficulties and they provide a compact representation of allowable variation, but are specific enough not to allow arbitrary variation different from that seen in the training set.

There are three key issues to be addressed in the design of models for image interpretation: generality, specificity and compactness. Generality concerns the ability of a model to accommodate variations in appearance of an object. Specificity concerns the ability to differentiate between objects and compactness is concerned with the use of the smallest number of parameters necessary.

1.2 Limitations of Non-Model-Based Methods

Many non-model based methods of image interpretation are ad hoc solutions to well-defined applications [CAN86a] [WON89]. Such non-model based methods have limited flexibility and rely on a pre-defined representation of the form of the object to be identified. Also, it is assumed that the objects appear the same always and that they are always in a similar context. These methods are fragile and involve a series of steps as described above. If any one step fails then those the whole process fails. Each class of object to be identified requires a customised interpretation programme.

1.3 Model Objectives

To extend model-based image interpretation an approach is needed that will accommodate variations in pose, illumination and topology of the objects. There is a need for a single model that can be used to represent a wide range of objects. There is a need to be able to identify many different

objects in a scene and to reason about which model should be used to identify each object. This requires a method with a great level of flexibility and an ability to differentiate between objects.

The effectiveness with which an object is represented can be illustrated when the mean model and its variations are replayed to synthesise images that can be seen to be representative of the class of objects modelled. This was demonstrated for the geometry and appearance of constrained models with early and more recent methods of model-based interpretation [COO92] [COO98].

Having a flexible method of interpretation with one model or the ability to select and apply multiple models means that it might be possible to reason about strategies for interpreting complex scenes and more readily adapt interpretation strategies to new contexts. However, the model presented in this thesis does not describe a method that works with multiple models for general scene interpretation.

1.4 The Approach Taken

In this Thesis we are concerned with a model-based method of flexible and adaptable 2D image interpretation that can adapt to variations of pose. To achieve greater flexibility and adaptability the model is formed in the exponential domain of an Extended SuperEllipse (ESE) to model object form. Using this representation sampled key points can be used rather than landmark points which are required to mark the same features at the same positions from all views of the object, as in the geometric domain of a PDM or AAM. The use of key points and an ESE to model a curve distinguishes the approach presented here from that in PDM and AAM methods that use landmarks.

A model-based system requires a user to be able to mark landmarks on each of a set of training images in such a way that each landmark represents a distinguishable point present on every example image. Landmarks are not a problem if the shape of objects is highly variable. Landmarks are a problem if the same point cannot be identified as a landmark in each form taken by an object. This is a particular problem for the creation of a model for images of pedestrians in various poses and contexts. Interpretation of images of pedestrians demands a degree of flexibility beyond that which is possible with current model-based methods of image interpretation [COO92] [COO92a]. Here this issue is addressed with key points, which are required only to sample the shape of the object being modelled. This allows points sampled at regular intervals around an object to be used to create a model. To achieve a model representation with substantial adaptability, the model is formed in the parametric domain of an ESE. Here we introduce the Extended Superellipse Appearance Model (ESAM), a statistical model that uses key points to represent a shape. The ESAM using key points in the exponent space has the potential to create a representation for a large range of shapes.

To determine where model interpretation may be applied a reliable method for detecting the location of all instances of the object sought in each image is required. These locations are customarily referred to as cues and the process as cue detection. A cue detection method based on regional symmetry is presented. A variant method of cue detection is presented to identify the principal component axes for complex and articulated objects by identifying the major component axes at any orientation and linking them together.

Cue points or axis points are used as the basis for a search for edge points. A selected set of well-defined search paths are used to generate the edge points sampled at regular intervals along the boundary of an object. In this Thesis an edge detector is presented to detect the edge points on systematically defined paths generated around cue reference points or perpendicular to axis points. These edge points along the boundary of an object are the key points that are the basis for the geometric representation of the model.

1.5 Challenges in Pedestrian Detection

The main difficulties that are presented to pedestrian detection systems are:

(1) Variability in the appearance of pedestrians

- People are non-rigid objects with a variety of gaits.
- People are individually clothed and often carry, wear or use accessories such as bags, hats, sticks and umbrellas. They may also be pushing a pushchair, pushing or riding a bicycle.
- People vary in size, ranging from children to adults and vary with distance from the camera.
- People may be viewed in a range of poses, such as, standing, walking and running.
- People can be viewed from many possible angles and are often occluded in varying degrees when they pass behind one another or other objects.

(2) Environmental variability

- The environments in which pedestrian images are captured vary significantly from natural scenes with trees, bushes to man-made scenes with roads and street furniture (such as traffic

signs and lampposts), billboards, walls, buildings and vehicles present. It can be difficult to distinguish pedestrians from scene clutter, such as trees and posts with pedestrians.

- There are two situations where lighting may impair the acquisition of a satisfactory image:
  - Over exposure: Resulting in saturation and loss of image content.
  - Under exposure: Resulting in insufficient signal to detect any image content.

(3) Camera movement

Movement of the camera during image capture further compromises pedestrian detection. Whilst it is necessary to be able to obtain a good quality image from a vehicle moving at speed and on an uneven surface these issues are beyond the scope of this study and methods that deal with these issues already exist [GAV00] [GAV07] [TUO11].

1.6 A Hierarchy of the Proposed Model

Where complex objects or a combination of objects with an obvious structure are concerned, a model and an interpretation scheme are formulated by identifying the component parts, each of which is a simple object. The ESAM presented here models and interprets simple objects or complex objects that are composed of simple objects such as the pedestrians which vary in form as the walk and, push pushchairs and push or ride bicycles. A filter at a variety of angles is used to identify primary and component axes of a combination of objects. In interpreting a complex object composed of simple object components, the parameters of each sub-model are matched to the extracted parameters of each respective object component of the complex object. In this The-

sis it is shown that simple objects, such as people alone and vehicles can be interpreted with a non-hierarchal image interpretation scheme.

1.7 Overview of the Thesis

In this chapter issues related to the research presented in this Thesis have been introduced.

Chapter 2 provides a review of cue detection to determine the location of the object sought and where model interpretation may be applied in an image. An overview is also given for pedestrian detection, important pedestrian detection systems and methods of image interpretation. The key methods of image interpretation reviewed are co-occurrence matrices, symmetry and axis of symmetry detection, alternative methods of object detection and recognition. The use of an axis detector to identify the major component axes for articulated objects is described. The concept of edge detection, which is important to sample boundary points and form a geometric model is described using two edge detection methods; the difference of Gaussians and the maximum likelihood ratio. The mathematical representation for each method is introduced and statistical model-based image interpretation schemes based on geometry and appearance, such as the point distribution model, active appearance model and the Eigenface method are introduced. Parametric curve representation methods such as the B-Spline, superellipse, superquadrics, extended superellipse and the extended superquadric are reviewed. A set of applications that illustrates the strengths and weakness of these methods is reviewed.

In Chapter 3 a cue detection method for locating potential cue points is described. This method is based on a method for edge detection that is used to locate boundary key points. The geometry of

the cue detector mask, the search strategy, the method of applying a threshold to identify potential axis cues is described. The method by which potential axis cues are clustered and selected to form the required axis cues is described. A variant approach to identify augmented cues for vehicles is also presented. A key point detection method using an edge detector method to locate potential boundary points along the boundary of the objects for forming a model and interpreting an image is also presented.

In Chapter 4, a brief review of the steerable filter transform is introduced. A description is given of how the maximum-likelihood edge detection method described in chapter 3 is adapted to make use of concepts used with steerable filters. A detailed description of an axis detection method for locating component axes of local symmetry as a basis for detecting pedestrians alone and associated with pushchairs or bicycles is presented. A set of procedures for locating and refining local points of symmetry along the major component axes is described. The augmentation of the detected cues to characterise the objects detected is also described. A key point selection algorithm is then introduced to identify the key points which sample the boundary of the objects and on which the formulation of a model and its interpretation are based.

Chapter 5 describes an adaptable model based on an extended superellipse representation, derived from boundary key points. The Extended Superellipse Geometric Model (ESGM) based on a parametric extended superellipse integrated with a distribution representation for the representation of the boundary shape of objects is introduced. This chapter further describes how the geometric and textural representations are combined to form a single appearance model. A way to identify

the most appropriate models for a given object or variation of an object is described. A limited description of log likelihood function and naive Bayesian classifier are given in this chapter.

In Chapter 6, the interpretation methods for the geometric and appearance models are presented along with an elaboration of how to train and match the geometric and appearance models to unfamiliar images. How the model parameters guide the interpretation process using a Jacobian matrix of the residual vector between the texture of the current image and the texture of the synthetic image is described.

Chapter 7 describes the experiments conducted on cue detection, local axis of symmetry detection, the creation and application of geometric and appearance models. The results obtained are presented, interpreted and evaluated. The training and test datasets, described in this chapter, are each contains images representing pedestrians, pushchairs, bicycles and vehicles; people are alone and in groups, pushchairs are being pushed, bicycles are being ridden or pushed. Experimental settings, cue detector results and the performance metrics for the cue detectors are presented in this chapter. The cue detection results for vehicles and the axes detection results for pedestrians alone and pedestrians associated with pushchairs and bicycles are presented and evaluated. The results of interpretation, the evaluation criteria for the geometric and appearance models and also a set of modes of failure for the appearance model are presented and interpreted.

Finally, in Chapter 8, conclusions and some further suggestions for further research are presented, including the further investigation for pedestrian cue detection, axis detection to help

locate pedestrians with pushchairs and bicycles, key point detection and an extended superellipse appearance model representation.

Fig. 1.1 shows a summary of key point selection algorithm which is used to identify the key points which sample the boundary of images of pedestrians. Pedestrian cue detector is used to identify a single reference point for each pedestrian sought in an image of pedestrians. These key points are used to create a model on which image interpretation is based. Detailed descriptions of these issues are given in Sections 3.3 and 3.5.

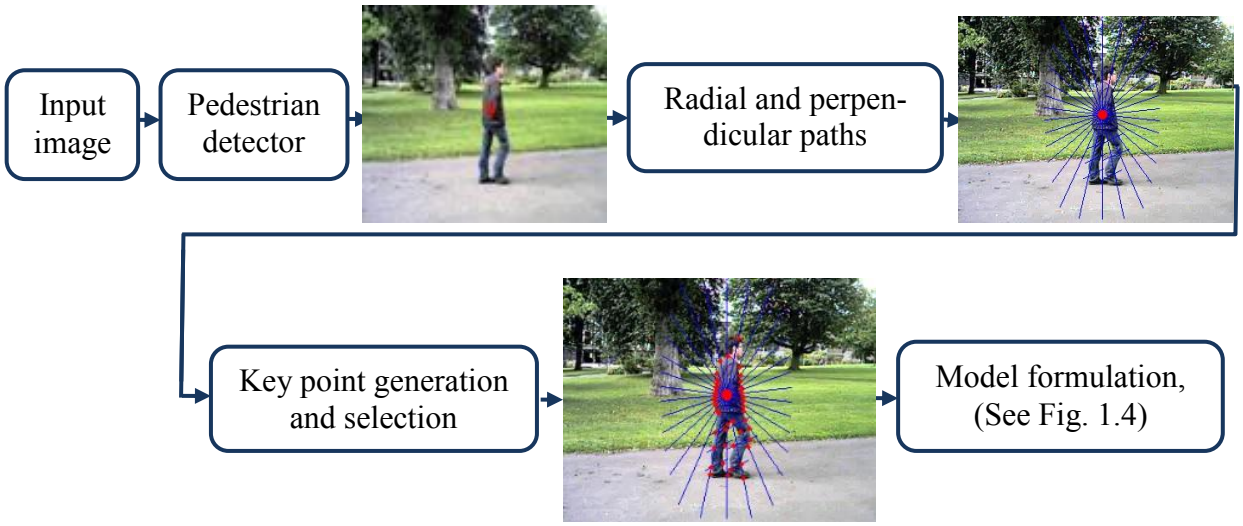


Fig. 1.1. A summary of key point detection algorithm on a pedestrian to construct a model.

Fig. 1.2 shows a summary of how to generate and select the key points along the boundary of an image of vehicles. Vehicle cue detector is a variant to the pedestrian cue detection algorithm which is used to determine the locations of the vehicles sought in an image of vehicles. The key points which sample the boundary of vehicles are required to create a model for vehicles and perform an interpretation. Detailed descriptions of these issues are given in Sections 3.4 and 3.5.

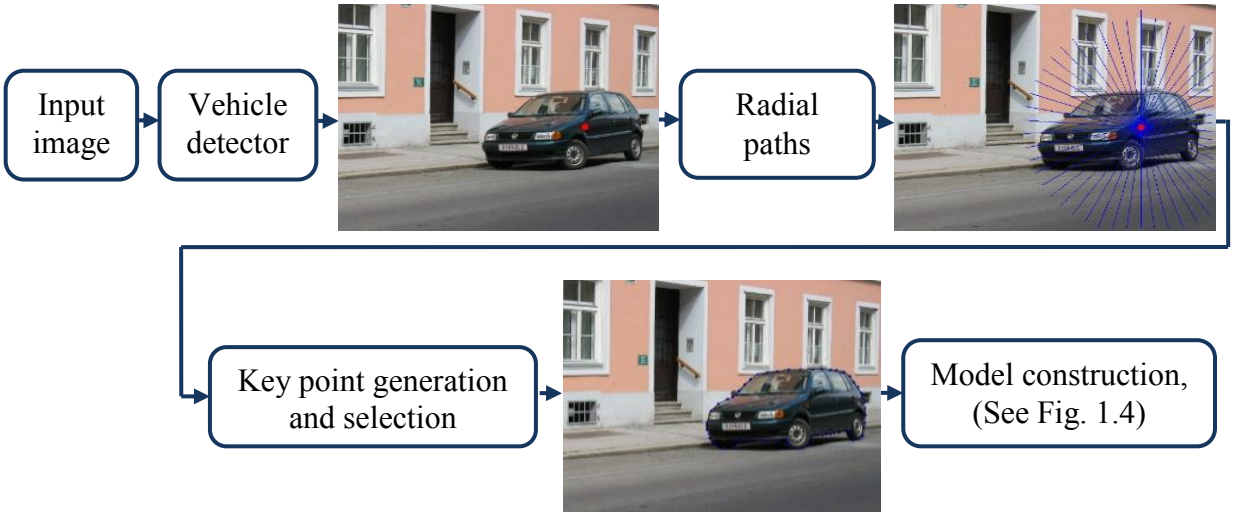


Fig. 1.2. A summary of key point detection algorithm on a vehicle to construct a model.

Fig. 1.3 presents a diagram showing an overview of axis detection method for generating the axes as a basis to characterise the objects to help locate pedestrians alone and pedestrians associated with pushchairs or bicycles. Further, this diagram shows the value of using component axes to identify key points on the boundary for pedestrians alone and associated with a pushchair or a bicycle on which the creation of a model and image interpretation using that model are based. Detailed descriptions of these issues are introduced in Sections 3.3 to 3.6.

Fig. 1.4 presents a diagram which shows an overview of the main procedures adopted for the proposed model-based system, including creation of geometric and texture models and how the geometric and textural representations are combined to form the appearance model. A key point selection algorithm to identify the key points along the boundary of the objects and on which the creation of a model is based is described in Figs. 1.1, 1.2 and 1.3 for pedestrians not associated with pushchairs or bicycles, vehicles,.and pedestrians associated with pushchairs or bicycles, respectively. The model representation is described in detail in Sections 5.4 to 5.6.

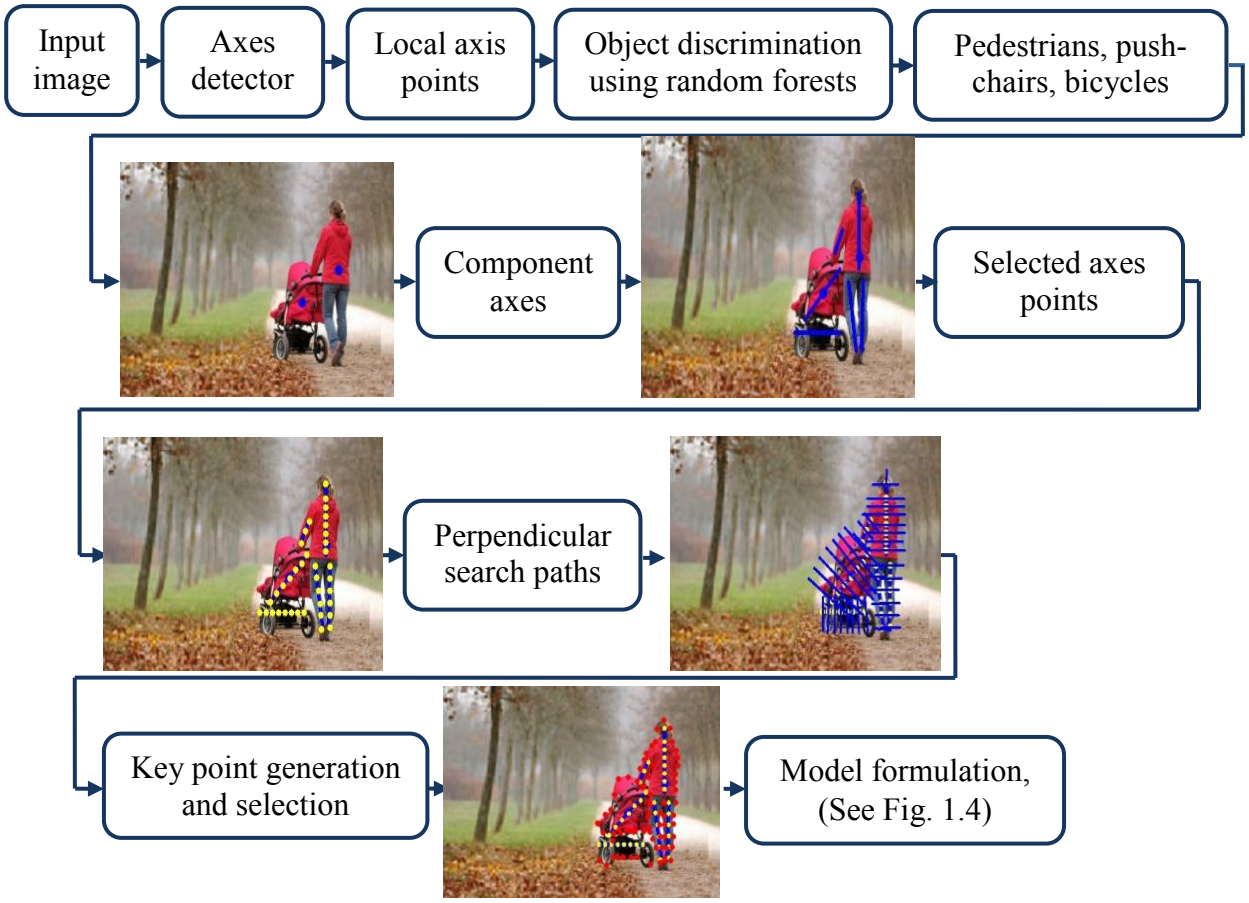


Fig. 1.3. A diagram showing an overview of detection of component axes and key point on a pedestrian pushing a pushchair.

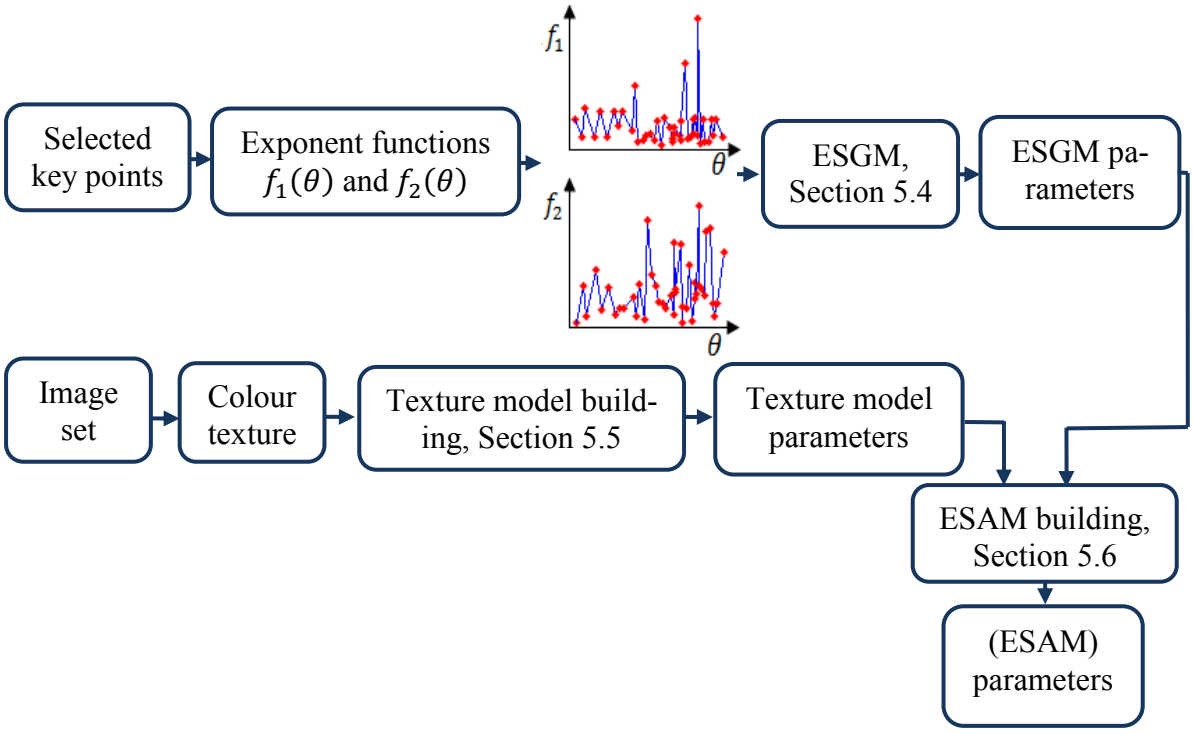


Fig. 1.4. An overview of the proposed model-based system.

Fig. 1.5 shows the key point detection method using the extended superellipse geometric model-based interpretation. A detailed description of the key point matching method is described in sub-section 6.2.2.

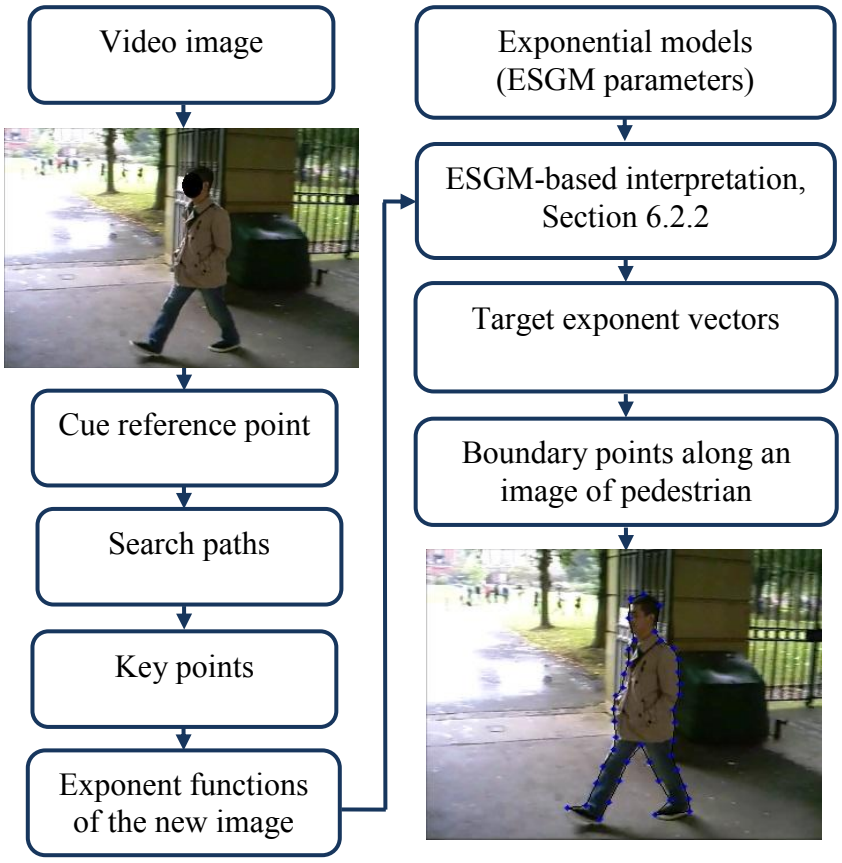


Fig. 1.5 Extended superellipse geometric model – based image interpretation.

A diagram showing a summary of image interpretation using extended superellipse appearance model-based image interpretation is presented in Fig. 1.6. A detailed description of ESAM interpretation method is described in subsection 6.3.3.

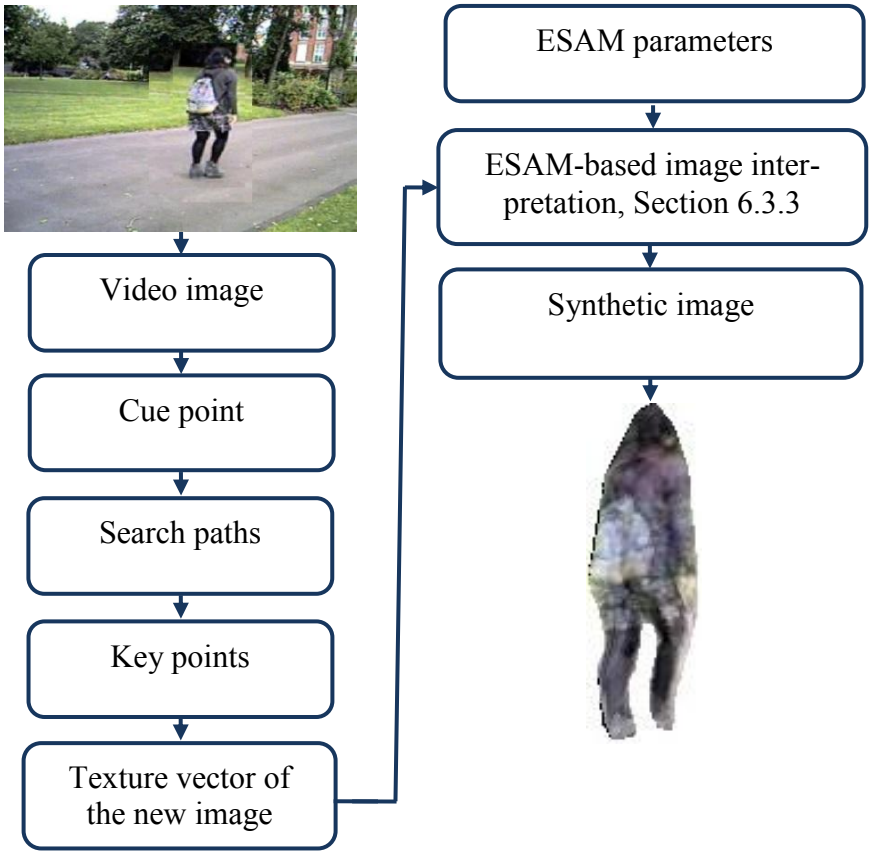


Fig. 1.6 Extended superellipse appearance model – based image interpretation.

Chapter 2 LITERATURE REVIEW

2.1 Introduction

Model-based image interpretation methods are commonly used to interpret image scenes where objects vary in appearance objects [COO01a]. A model is used to improve the reliability of interpretation [AIX03], over that provided by non-model-based [MCI96]. Specifically, model-based schemes have helped to identify objects and their boundaries with greater precision than was previously possible [TIL00]. A model-based method of image interpretation can be used as a mean of dealing with the intrinsic complexity in the appearance of objects and can accommodate a fair degree of flexibility. Model-based methods have been valuable in medical image interpretation [COO01a], in industrial inspection [AIX03] and in security surveillance [DEM05].

2.1.1 Model-based Image Interpretation

Statistical model-based image interpretation schemes commonly use multivariate statistics to summarise the patterns in images that are characteristic of the objects being modelled. Model-based methods may use statistical scene descriptions or mathematical models to represent the patterns of the shape and appearance of objects in images [COO98]. Point Distribution Model (PDM) and Active Appearance Model (AAM) are customised to an object by selecting characteristic points and, in the case of the AAM, image values associated with each landmark. The average values of landmark points represent the mean shape and appearance of the training set. The variability of the shape and appearance about the mean is parameterised by computing the eigenvectors of the covariance matrix of landmark coordinates and appearance values. Interpretation is performed by varying the model until a variation that is a good match to the features in an image

is found. New cue detection, model and interpretation strategies are often required for each application domain or if the variation in appearance within the domain is substantial.

2.2 Scope of the Literature Review

Although much research has been conducted into image interpretation, most results of significance have been reported using methods tailored for specific applications. There remains a need to develop methods of image interpretation that can be used with complex images and which can be adapted to new contexts. There is a need to find a way to obtain a precise, reliable and computationally efficient method of interpretation. Further, there is a need to develop adaptable methods that can be used for a wide range of applications.

This chapter introduces a review of: 1) cue and pedestrian detection, 2) axes symmetry detection, 3) edge detection, 4) a statistical PDM that models geometry [COO92], the AAM that models geometry and appearance [COO98] and the Eigenface method that models facial appearance [TUR91], 5) parametric representation methods such as the SuperEllipse (SE) that models symmetric shapes, the Extended SuperEllipse (ESE) that models shapes, which are not necessarily symmetric and the B-spline that models the curved outline of objects.

Cue, symmetry axis and edge detection methods are reviewed in Sections 2.3, 2.4, and 2.5, respectively. Cue and symmetry axis detection methods are necessary to determine where an object is likely to be and where it is appropriate to try to fit a model in an image. Edge detection is important to find the edge points on the boundary of the object being the. Point distribution, active appearance and Eigenface representation methods are reviewed in Sections 2.6, 2.7 and 2.8 re-

spectively. The B-spline, superellipse and extended superellipse curve representation methods are reviewed in Sections 2.9, 2.10 and 2.11, respectively. The Eigenface method is reviewed as an appearance method and the B-spline, superellipse and extended superellipse representations are reviewed as flexible methods for modelling form. A cue point or a series of points forming axes are used to guide the search for object boundaries, the creation of a model and image interpretation using that model. Finally we finished with a summary in Section 2.12.

2.3 Cue Detection

A cue is a point that determines where model interpretation may be applied. A cue can be a single point or a series of points forming an axis. Where complex objects are concerned cues might be required for each component so that each component and their articulation can be modelled. In model-based interpretation a cue detector is required to be fast and reliable, should not require arbitrary critical parameters to be set and should be largely invariant to changes in illumination and variations of target object scale. A cue detector should generate cues for a wide range of objects, identifying each object of interest and a small number of cues not related to objects of interest. It is that cues for all objects of interest are detected. The expectation is that model fitting will eliminate from further consideration detected points not associated with objects of interest. Therefore, the detection of cues that are not required reduces efficiency.

Cue detection is an ill-posed problem because it is impossible to fully define what should and should not be detected.

2.3.1 Pedestrian Detection

Pedestrian detection plays an important part in driver assistance [NAN02], monitoring human activity in indoor and outdoor scenes [VIO05], such as at a bus stop [MOH0] and monitoring footfall in shopping malls [SCH09].

2.3.1.1 A Review of Selected Pedestrian Detection Systems

It is not practical to review all reported pedestrian detection systems and it is difficult to quantitatively compare results between different pedestrian detectors because the degree of variation in pose, camera position and the motion of the camera often vary and often are not quantified. Below is a review of selected popular pedestrian detection systems.

Mohan et al. [MOH01] investigated a component-based recognition system for pedestrian detection in cluttered static images. A Haar wavelet transform was used to represent the head, legs and arms as components of the human body. A Support Vector Machine (SVM) was employed to classify the components of the body. These components were combined by another SVM to identify the full body of the person. The system was able to detect a person, even if one of its trained components was not detected or occluded. However, the features used were sensitive to changes in appearance and illumination. The positive examples of the training dataset were taken from a database captured in Boston and Cambridge, Massachusetts with different cameras, lighting and weather conditions. The negative examples were images of building and scenery including lamp posts and trees. The system was run over 123 images of people to determine the positive detection rate. The false alarm rate was obtained by running the system over a set of 50 images of

scenery and buildings. The system offered a positive detection rate of 67.3% and a false detection rate per frame of 4.6%.

Nanda and Davis [NAN02] presented a real time pedestrian detection system for operation in cluttered scenes, which works on low quality infrared videos, viewed from a moving platform. They developed a Bayesian probabilistic template matching scheme that operated at 3 scales using infrared images. The template was matched at all locations in an image by searching from coarse to fine resolution to increase the speed of operation. Pedestrian detection was performed at a low resolution with a relatively low threshold. The pixels that belong to objects that do not emit heat were given the value 0 and the pixels that correspond to objects emitting heat such as lamps, cars and humans were given the value 1. Only the regions that passed the threshold at a lower resolution level were passed to the next finer level of resolution for recognition. Three probability maps were created and then thresholded using a Bayesian classification of the probability maps as defined by Nanda and Davis [NAN02]. Local maximas were found on each probability map and declared to be pedestrians. The system ran at 11 frames per sec (fps) on 320 x 240 pixel images and at 3 fps on 640 x 480 pixel images. The images of people might also contain street lamps and cars. The system was resilient to noise and a good degree of occlusion. The true detection rate ranged from 75% to 90% on a dataset of 6 videos with one false positive per frame on an average.

The Histograms of Oriented Gradients (HOG) method is based on evaluating normalized local HOG in a dense overlapping descriptor [DAL05]. The image window was divided into a grid of "cells". The detector window was tiled with a grid of blocks in which HOG feature vectors were

extracted. For each cell a local histogram of edge orientations from the cell was accumulated. The local histograms for block of cells "a group of neighbouring cells" were accumulated and each cell histograms normalized. The presence of people was detected using the HOG feature vectors and a linear SVM. The detection window was scanned across the image at all positions and scales and non-maximum suppression was applied to the pyramid output to detect pedestrians. Each detection window needs a large number of pixels so that each cell can have sufficient pixels to a significant sample and so that a block of cells can be formed. This method required a high resolution image. They considered the impact of each step of the computation of the detector and found that a large bin size was required to detect pedestrians well. They evaluated the method on the MIT pedestrian dataset [MOH01] which contains 509 training and 200 test images of pedestrians with a relatively limited range of poses and the INRIA [DAL05] data set which contains 1805 human images with size 64×128 cropped from a diverse set of photographs of people. The performance of the pedestrian detector on MIT and INRIA pedestrian datasets was evaluated by plotting the miss rate, defined as the proportion of negative tests among people present in a scene, against the False Positive Per Window (FPPW) rate. In this plot, curves that lie lower on the figure represent better performance (because they have a lower miss rate for a given FPPW rate). Overall system performance depends on how many windows are presented to the detector in an average image. The FPPW statistic is attractive for evaluating the behaviour of the classifiers, but less useful for evaluating the whole systems. A higher FPPW rate may be tolerable at fewer windows, though looking at fewer windows might affect the detect rate. Dalal and Triggs [DAL05] demonstrated that the HOG performs better than other methods based on predefined feature measurements for person detection. The method gave an impressive detection rate of 89% with 4^{-10} FPPW. The overall image resolution was not given.

Viola et al. [VIO05] reported a pedestrian detection system using a series of Haar-like edge detection masks and an AdaBoost classifier. The AdaBoost classifier selected a set of features and formed a classification tree. In each round the learning algorithm chose from a heterogeneous set of filters, including motion direction, motion shear, motion magnitude and appearance filters. The AdaBoost algorithm picked the optimal threshold for each feature. With AdaBoost the features with the lowest weighted error on the training examples was selected at each stage. The authors created eight video sequences of pedestrians in street scenes. Six video sequences were used to create a training set for both a dynamic and a static pedestrian detector. The remaining two sequences were used for evaluation. The dynamic and static detectors were trained on consecutive frame pairs and static patterns, respectively. Each classifier in the cascade was trained using 2,250 positive and 2,250 negative samples. This system operated at about 4 fps to detect pedestrians at very low resolution (20 x 15 pixels). The detection rate for both dynamic and static pedestrian image sequences was 80% and the False Positive Rate (FPR) was 1/400,000 which corresponds to 1 FPR every 2 frames for image frames of 360 x 240 pixels. On another evaluation test, the detection rate was 80% for moving pedestrians with an FPR of 1/400,000. For static pedestrian images the detection rate was 80% with an FPR of 1/15,000. The method was not effective for partially occluded pedestrians or pedestrians in crowded scenes. A large training dataset (2,250 images) was needed for reliable operation because of the large number of features and the nature of the classifier. The pedestrian detection system scaled the training image set to a resolution of 20 x 15 pixels and worked well with images taken in snow and rain. The method was robust and computationally efficient making it suitable for real time operation.

Schwartz et al. [SCH09] sought to improve on the method of [DAL05] by using the partial least square analysis to model the features extracted using HOG [DAL05] and co-occurrence matrices created at 18 × 36 pixels resolutions using the DaimlerChrysler dataset [MUN06]. A SVM was used to identify pedestrians. This method was invariant to small changes of rotation and scale and was able to detect isolated pedestrians of varied size in real time. A close inspection of the results shows that at low light levels, and in crowded scenes the false positive and false negative rates were higher. The method was able to process 2,929 detection windows per second. This method was tested on grey-level 64 x 128 pixel INRIA person set [DAL05], the DaimlerChrysler person set with images of 18 x 36 pixel resolutions [MUN06] and the ETHZ crowded pedestrian video scenes of 640 x 480 pixels at 15 fps [ESS07]. On the INRIA person set the recall rate was about 60% and a miss rate of 5.8% was reported at 10^{-5} FPPW and 7.9% at 10^{-6} FPPW. The detection rate on the DaimlerChrysler pedestrian dataset was approximately 85% with a false positive rate of 0.15 per image. The false positive rate measures the proportion of negatives that are incorrectly identified. On the ETHZ dataset, the method was evaluated on three test sequences of 64 x 128 pixels at 15 fps where Schwartz et al. used the False Positives Per Image (FPPI) [DOL09] as the evaluation metric, in which the miss rate was plotted against the FPPI. On the first sequence, the recall rate was about 70% with 4.5 FPPI. On the second sequence, the recall rate was about 60% with 2.5 FPPI and on the third test sequence the recall rate was 78% with 2.3 FPPI. The recognition rates on the INRIA person dataset and the three test sequences of the ETHZ dataset are low compared to other methods. FPPI is a measure that takes into account the number of windows presented to the classifier. Curves that lie lower are better. Dollar et al. [DOL09] have conducted FPPI as a systematic evaluation of pedestrian detectors on a large dataset built for that purpose. However, the ranking of pedestrian detection methods changes depending on whether

one plots FPPW or FPPI; generally, FPPI is expected to be more predictive of performance of pedestrian detection approaches than the FPPW. Table 2.1 shows a comparison of pedestrian detection performance of [SCH09] for different datasets in terms of the image size, frame rate, recognition rate and the evaluation measures of FPPW, FPR and FPPI.

Table 2.1. A comparison of detection performance of the pedestrian detection method [SCH09] with different data sets over different evaluation measures.

Data set	Image size	Frame rate	Recognition rate	Evaluation measure
INRIA	64 x 128	Not given	60%	5.8% miss rate at 10^{-5} FPPW 7.9% miss rate at 10^{-6} FPPW
DaimlerChrysler	18 x 36	Not given	85%	85% detection rate at 0.15 FPR
ETHZ	640 x 480	15 fps	70% 60% 78%	70% recall rate at 4.5 FPPI (dataset 1) 60% recall rate at 2.5 FPPI (dataset 2) 78% recall rate at 2.3 FPPI (dataset 3)

Tang et al. [TAN12] combined a Random Forest (RF) classifier and Dominant Orientation Templates (DOT) for pedestrian detection. The DOT was a binary feature version of the HOG adopted to improve the speed of computation by down-sampling the search space. The detection method was represented in a 2-level cascade architecture. First, a holistic RF detector was trained to identify the interest points and classify the patches centred at those points to identify potential regions. A patch-based RF with a Hough-based Forest (HF) [GAL13] was performed on the identified regions to accelerate the detection of pedestrians. HF exploited all the feature vectors extracted from the training data for registering a vote during testing. HF was performed to identify possible object centres. This system operated at 5 fps for 24 scales on images of 640 x 480 pixels at the base level. This system was able to detect pedestrians in the foreground and background. This was probably due to the use of 24 scale images. The detection rate was 90%. However, the system is computationally complex.

A joint deep model was formulated for pedestrian detection, with a combined Convolutional Neural Network (CNN) architecture with a Deformable Part-based Model (DPM) [OUY13]. Four processes (feature extraction, component deformation, occlusion and classification) operate in cooperation in a learning model for pedestrian detection. A learning process maximizes the strength of each process. The feature values that describe body parts of a pedestrian, their visibility and occlusion are learnt. The component parts of a pedestrian body are the head, torso, shoulder, hands and the legs. The deep model organized these components into layers and jointly optimized them through Back-Propagation (BP). The filters in the second convolutional layer were designed with variable sizes since the parts of pedestrians have different sizes. In this model filtered data maps were obtained from the first convolutional layer. Pixel values at each resolution and edge values were input to the first convolutional layer. This layer convolved the 3-channel YUV input image data with $9 \times 9 \times 3$ filters and outputs 64 maps. Feature maps were obtained by averaging the 64 filtered data maps using 4×4 boxcar filters with a 4×4 sub-sampling. Part detection maps were obtained from the second convolutional layer. This layer was formed by convolving the feature maps with 20 part filters of different sizes producing 20 part detection maps. A deformation handling layer generates 20 part scores from the 20 part detection maps. The visibility reasoning model was used to label each window that encloses a pedestrian. The detection windows were extracted into images with height of 84 and width of 28, in which pedestrians have height 60 and width 20. The interaction between deformation, visibility, and feature learning improved the detection ability of the model. It was assumed that a pedestrian only has one instance of a body part. A good performance was achieved on the Caltech dataset. However, the deep convolutional learning classifiers [TAN12] [OUY13] have not achieved impressive results for pedestrian detection. The log-average miss rate computed by averaging the

miss rate at nine FPPI rates on the Caltech pedestrian test was used to summarize the performance of the pedestrian detection deep convolutional learning model. FPPI rates were evenly spaced in the log-space in the range from 10^{-2} to 1. An average miss rate of 39%, a false positive per image of 0.01 and a detection rate of 92% were achieved. At the testing stage, the execution time required by the pedestrian detector model was less than 10% of the whole time.

Lim et al. [LIM13] proposed mid-level features to both feature learning and detecting local edge structures. The features, called sketch tokens, were learned using supervised mid-level information in the form of hand drawn contours. The supervised mid-level information was obtained from human labelled edges in natural images [ARB11]. A set of sketch token classes representing a wide variety of local edge structures in an image was defined. These classes include structure shapes such as straight lines, junctions, corners, curves and parallel lines. A set of images with corresponding binary images representing the hand drawn contours were generated. Patches of human centred on contours were extracted from the hand drawn sketches and clustered to form token classes. The sketches were generated by dividing each image into pieces, where each piece represents a shape in the image. The sketch patches were clustered using the K-means algorithm and only the patches that contain a labelled contour at the centre pixel were used. A data driven approach inspired by Dollar et al. [DOL09] was presented to classify each image patch with a token label using a set of low-level features such as oriented gradient channels, colour image channels and self-similarity features [SHE07]. The colour channels were computed using the CIE-LUV colour space that composed of three colour channels, eight oriented gradient information and three gradient magnitude channels. The self-similarity features captured the image patches that contain similar textures based on gradient information. The ground truth class labels

were obtained from the token classes produced from clustering the patches of hand drawn contours. Random decision forests [BRI01] were employed to predict the probability that an image patch belongs to each token class. The pixels token labelling of low-level contours was computed in less than 1 second per an image of 480 x 320 pixel resolution. Contour detection results were explored on the Berkeley segmentation dataset [ARB11] and achieved a detection rate of 95%. The approach was also explored on the INRIA person [DAL05] and PASCAL 2007 object recognition datasets [EVE10]. The approach achieved log-average miss rates of 19.5% using 150 sketch tokens and 14.7% when token features were combined with 10 low-level features.

Tian et al. [TIA15] proposed a deep model for pedestrian detection that learns high level features from multiple tasks and multiple data sources. Tian et al. jointly optimized pedestrian detection with auxiliary semantic tasks to eliminate the hard negative proposals in the background. These semantic tasks include pedestrians with backpacks, bags and hats and background instances such as vehicles, trees and lamp posts. The pedestrians and background instances were jointly learnt using a single Task-Assistant-CNN (TA-CNN). Tian et al. used a binary label to indicate whether an image patch contained a pedestrian or not. The TA-CNN labelled an input image patch as containing a pedestrian, or not, by stacking four convolutional layers, four max-pooling layers, and two fully-connected layers. This structure was inspired by the AlexNet [KRI12] for large-scale general object categorization. The training set was constructed by combining patches cropped from both image regions containing pedestrians and not. Image patches containing pedestrians with backpacks, bags or hats were manually labelled. The existing background image patches were transferred to the pedestrian dataset without annotating them manually when the number of negatives is significantly larger than the number of positives. Image patches with trees or other

non-pedestrian objects and image patches with such objects were separately analysed. The former one facilitated the learning of shared representation among backgrounds, whilst the latter one increased diversities of semantic tasks. The TA-CNN was trained and evaluated on the Caltech [DOL12] and ETH train and test person datasets [ESS07]. The log-average miss rate over nine points was ranged from 10^{-2} to 1 FPPI on Caltech and ETH test datasets. The TA-CNN achieved a 25.64% average miss rate, 0.001 false positive per image and a detection rate of 93% on the Caltech test and 91% on ETH test person datasets. Additional datasets for training and additional labels for pedestrians in the Caltech dataset were used. The relatively high detection rate depends primarily on the choice of dataset as a qualification of confidence in the qualitative estimates of performance. Also, they generally ignored the critical issue caused by various scales of pedestrians in an image which is considerably affecting the performance of pedestrian detection.

A Markov Random Field (MRF)-poselet model [NGU15] constructed from poselets, a notion of parts, which represent the appearance and the structure or pose of the human body parts was proposed. MRF was presented for modelling the spatial and structural relationship of the human body structure. There is a high degree of articulation of people as pose and viewpoint is varied. The observation nodes in the MRF- model were the detected poselets and hence, for each poselet type such as head, torso or legs, more than one poselet instance can be detected in an object hypothesis. A poselet was defined to represent the pose using an appearance model and a set of key points which captures the structure of the poselet was used to represent the boundary of the parts of the pedestrian such as for the body joints, eyes, ears, and nose. The appearance of poselets was represented by features determined by HOG [DAL05]. Person detection was formulated as a Maximum A Posteriori (MAP) estimation in the MRF model. This was efficiently solved using

variational mean field inference. Detecting poselets in an image was performed by scanning the image at various scales and locations with each poselet detector using deep neural networks as part detectors. For each candidate poselet the features were extracted and classified against a threshold. Non-maximal suppression was used to merge nearby poselet candidates of the same poselet type to form a set of poselet detections. The MRF-poselet model was shown to be flexible and robust to a wide range of deformation modes of deformation of the human body, provided a sufficient number of poselet types was defined. Poselets were not required for all body components such as the head, arms and legs to be represented; the number of detected parts was not fixed enabling the model to deal with occlusion. The MRF-poselet model was evaluated on torso detection, person detection and key point prediction using the H3D [BOU09] and PASCAL VOC 2007-2009 [EVE10] datasets. Key points were predicted as follows: For each human hypothesis, an MRF-poselets model was constructed and the inference algorithm was applied. Then, each key point in the human object could be predicted by more than one poselet detection and only the poselet detection with the maximum of the variational distributions was used to compute the key points. The detection performance was measured using Precision-Recall (PR) and Average Precision (AP) metrics. True detections and false alarms were determined using the PASCAL VOC criterion [EVE10]. The MRF-poselet model significantly outperformed the original work of poselets [BOU10] [BOU09] and increased the AP approximately by 4% compared with that reported in [BOU10], by 9% compared with that reported in [BOU09], and by 12% with that reported for the DPM in [FEL08]. On the PASCAL VOC 2009 dataset, this model achieved a performance 4% higher than with the AP model presented in [GKI14]. The person detection rates for the H3D dataset are 94% and 93% on the PASCAL VOC 2007 with a false alarm of 0.0001 per window

on PASCAL VOC 2007 dataset. The MRF-poselets model processed an image for human detection in the PASCAL VOC 2007 dataset in about 15 seconds.

Table 2.2 summarises pedestrian detection performance in terms of the detection rate, false detection rate, processing rate and resolution.

Table 2.2. Performance comparison for selected pedestrian detection methods using the dataset presented by each detector unless otherwise stated.

	Detection rate (%)	Log-average miss rate	Frame rate (frames per sec.)	Pixel resolution
Mohan et al. [MOH01]	67.3	96%	0.2	128 x 64
Nanda and Davis [NAN02]	75-90	86%	0.33	640 x 480
Dalal and Triggs [DAL05]	89	68%	1	64 x 128
Viola et al. [VIO05]	80	95%	0.25	64×128
Schwartz et al. [SCH09], INRIA set	92.1-94.2	62%	0.77	64 x 128
Tang et al. [TAN12]	90	57%	5	640 x 480
Lim et al. [LIM13]	95	14.7%	1	480 x 320
Ouyang and Wang [OUY13]	92	39%	0.025	640 x 480
Tian et al. [TIA15]	93	25.6%	30	640 x 480
Nguyen et al. [NGU15]	94	67%	1 frame per 15 sec.	1024 x 768

The state of the art in pedestrian detectors is advancing and considerable progress has been made in recent investigations. Each of the reviewed pedestrian detection methods performed well in the situations considered and each fell short of what was required in one aspect or other, such as resilience to variation in illumination, pose and severe occlusion. They each have limited adaptability with cluttered scenes. A need remains for further improvement on pedestrian detection, especially where people are clustered together, spread across a complex scene, occluded at a high level and present in both the foreground and background.

Dollar et al. [DOL12] evaluated the state of art of pedestrian detection in a unified framework. In this paper the authors: 1) integrated a large, well-annotated, monocular person dataset and examined performance as a function of scale, location and occlusion, 2) reported a new refined per-frame evaluation method to analyse detection rates under diverse levels of occlusion and scale, localization accuracy and execution speed and 3) investigated the performance of 16 pedestrian detectors on 6 datasets. The pedestrian detectors reported in [DOL12] typically follow a sliding window pattern which involves feature extraction, classification, and a multi-scale scanning of detection windows followed by a non-maximum suppression scheme. The statistical importance of the results was assessed leading to the suggestion that pedestrian detection requires further improvement, particularly to deal with occlusion, small scale and motion.

Table 2.3 shows a summary of the different pedestrian datasets described in [DOL12]. The data presented in Table 2.3 covers the use of photographs [DAL05], surveillance video [WUB05] and images captured from a mobile recorder [ESS07] [WOJ09] [ENZ09]. The number and type of data in each dataset is categorized by: 1) the number of pedestrian windows, 2) the number of images without pedestrians, and 3) the number of un-cropped images with at least one pedestrian present. The 10^{th} percentile, median and 90^{th} percentile pedestrian pixel heights are also listed.

Table 2.3. Pedestrian detection datasets [DOL12].

		Training			Testing			Height		
	imaging setup	Number pedestrians	Number neg. images	Number pos. images	Number pedestrians	Number neg. images	Number pos. images	10% quantile in pixels	Median in pixels	90% quantile in pixels
MIT [PAP00]	Photo	924	-	-	-	-	-	128	128	128
USC-A [WUB05]	Photo	-	-	-	313	-	205	70	98	133
USC-B [WUB05]	Surv.	-	-	-	271	-	54	63	90	127
CVC [GER05]	Mobile	1000	6175	400	-	-	-	46	83	164
TUD-det [AND08]	Mobile	400	-	-	311	-	250	133	218	278
Daimler-CB [MUN06]	Mobile	2.4k	15k	-	1.6k	10k	-	36	36	36
NICTA [OVE08]	Mobile	18.7k	5.2k	614	6.9k	50k	-	72	72	72
INRIA [DAL05]	Photo	1208	1218	499	566	453	288	139	279	456
ETH [ESS07]	Mobile	2388	-	1092	12k	-	1804	50	90	189
TUD-Brussels [WOJ09]	Mobile	1776	218	-	1498	-	508	40	66	112
Daimler-DB [ENZ09]	Mobile	15.6k	6.7k	67k	56.5k	-	21.8k	21	47	84
Caltech [DOL09]	Mobile	192k	61k		155k	56k	65k	27	48	97

The INRIA pedestrian dataset [DAL05] is widely used, having featured in the evaluation of recent advances in pedestrian detection. It contains relatively high resolution pedestrian images as indicated (but was not specified). The TUD-Brussels [WOJ09] and Daimler-DB [ENZ09] datasets were captured in urban contexts using a camera mounted on a vehicle, whilst the ETH dataset [ESS07] was captured in urban settings using a camera mounted on a pedestrian. The Caltech dataset [DOL09] consists of video sequences with occlusion that is only concerned with the detail of the pedestrian, pedestrians with a wide range of scales and a high degree of scene variation.

Dollar et al. demonstrated that the detection of pedestrians in images must often be performed using low resolution images and partial occlusion. They have commented that this was likely to be a reason for the failure to detect pedestrians.

2.3.1.2 Co-occurrence Matrices

The Grey-Level Co-occurrence Matrix (GLCM) [HAR73] is a square matrix whose elements corresponding to the relative frequency of occurrence of the pairs of grey-level of image pixels separated by a certain distance in a given direction [ELE11]. The GLCM, initially proposed by Haralick et al. [HAR73], estimates texture feature properties of an image related to second-order texture classification method. The GLCM represents the distributions of the intensities and the information about relative positions of neighbouring pixels combination in an image.

Haralick et al. [HAR73] proposed fourteen statistical features extracted from the GLCMs. These features characterized the spatial relationship between the grey-levels of pixels in a neighbourhood. The features were defined at orientations between neighbour pixels of $0^{\circ}, 45^{\circ}, 90^{\circ}$ and 135° and averaged across the four orientations. The co-occurrence matrices were computed for all the images in the normalized database. This is to overcome the effects of monotonic transformations of the true image grey-levels caused by variations of lightning. To normalize GLCM, its values are divided by the total number of increments. The elements of a grey-level co-occurrence matrix, *P*, were defined as [ELE11]:

$$P(i,j) = \sum_{x=1}^{N_g} \sum_{y=1}^{N_g} \begin{cases} 1, & if\ I(x,y) = i \text{ and } I(x+\Delta_x, y+\Delta_y) = j \\ 0, & otherwise \end{cases} \tag{2.1}$$

Where:

$I(x, y)$*:* denotes an image of N_g pixels in the horizontal and vertical directions, respectively.

i and *j:* denote the horizontal and vertical co-ordinates of the image, respectively.

(Δ_x, Δ_y): is the distance between the pixel-of-interest and its neighbour.

$P(i,j)$ is a normalized entry of the co-occurrence matrices. That is $P(i,j) = P(i,j)/R$, where R is a normalizing constant, refers to the total number of pixel pairs (i,j). Each entry (i,j) in $P(i,j)$ corresponds to the number of occurrences of the pair of grey-levels *i* and *j* which are at a displacement distance (Δ_x, Δ_y) apart in the original image. The marginal probabilities matrices were defined as [ELE11]:

$$P_x(i) = \sum_{j=1}^{N_g} P(i,j) \tag{2.2}$$

$$P_y(j) = \sum_{j=1}^{N_g} P(i,j) \tag{2.3}$$

Where:

$P_x(i)$: is the i^{th} entry in the probability matrix obtained by summing the rows of $P(i,j)/R$.

$P_y(j)$: is the j^{th} entry in the marginal matrix obtained by summing the columns of $P(i,j)/R$.

The fourteen texture features extracted from the GLCMs were presented below [HAR73]:

(1) Angular second moment: $\text{ASM} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P(i,j)^2$ (2.4)

Angular second moment or Energy is a measure of textural uniformity of an image, which is a measure of the homogeneity of an image.

(2) Contrast: $$\text{CON} = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \begin{matrix} P(i,j) \\ |i-j| = n \end{matrix} \right\} \quad (2.5)$$

Contrast measures the local grey-level variations in the GLCM of an image.

(3) Correlation: $$\text{CORR} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{ijP(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (2.6)$$

Correlation is a measure of grey-level linear dependencies in the image that represents the correlation of a reference pixel to its neighbour over an image.

Where: μ_x, μ_y, σ_x and σ_y are the means and standard deviations of p_x and p_y, respectively. They were defined as [HAR73]:

$$\mu_x = \sum_{i=1}^{N_g} i P_x(i) \quad (2.7)$$

$$\mu_y = \sum_{i=1}^{N_g} i P_y(i) \quad (2.8)$$

$$\sigma_x = \left(\sum_{i=1}^{N_g} P_x(i) \, (i - \mu_x)^2 \right)^{\frac{1}{2}} \quad (2.9)$$

$$\sigma_y = \left(\sum_{i=1}^{N_g} P_y(i) \left(i - \mu_y\right)^2 \right)^{\frac{1}{2}} \quad (2.10)$$

(4) Sum of squares: variance: $$\text{SOS} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu)^2 P(i,j) \quad (2.11)$$

(5) Inverse difference moment: $\text{IDM} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{1}{1+(i-j)^2}P(i,j)$ (2.12)

Inverse difference moment measures the local homogeneity of grey-levels in the spatial distribution of an image.

(6) Sum average: $\text{SA} = \sum_{k=2}^{2N_g} kP_{x+y}(k)$ (2.13)

Where:

$$P_{x+y}(k) = \sum_{i=1}^{N_g}\sum_{\substack{j=1 \\ i+j=k}}^{N_g} P(i,j); \qquad k = 2,3,\cdots,2N_g \tag{2.14}$$

(7) Sum variance: $\text{SV} = \sum_{k=2}^{2N_g}(k-SA)^2 P_{x+y}(k)$ (2.15)

(8) Sum entropy: $\text{SE} = -\sum_{k=2}^{2N_g} P_{x+y}(k)\log\{P_{x+y}(k)\}$ (2.16)

(9) Entropy: $\text{ENT} = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} P(i,j)\log\{P(i,j)\}$ (2.17)

Entropy represents the randomness or disorder of grey-level distribution of an image and it achieves its largest value when all elements in $P(i,j)$ are equal [ELE11].

(10) Difference variance: $\text{DV} = \sum_{k=0}^{N_g-1}\left[k - \sum_{l=0}^{N_g-1} lP_{x-y}(l)\right]^2 P_{x-y}(k)$ (2.18)

Where:

$$P_{x-y}(k) = \sum_{i=1}^{N_g}\sum_{\substack{j=1 \\ |i-j|=k}}^{N_g} P(i,j); \quad k = 0,1,\cdots,N_g-1 \tag{2.19}$$

(11) Difference entropy: $\text{DE} = -\sum_{k=0}^{N_g-1} P_{x-y}(k)\log\{P_{x-y}(k)\}$ (2.20)

(12) Information measure of correlation 1: $\text{IMC1} = (ENT - HXY1)/max(HX, HY)$ (2.21)

(13) Information measure of correlation 2: $\text{IMC2} = (1 - exp[-2(HXY2 - ENT)])^{\frac{1}{2}}$ (2.22)

Where:

$$HX = -\sum_{i=1}^{N_g} P_x(i) \log\{P_x(i)\} \tag{2.23}$$

$$HY = -\sum_{j=1}^{N_g} P_y(j) \log\{P_y(j)\} \tag{2.24}$$

$$HXY1 = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} P(i,j) \log\{P_x(i)P_y(j)\} \tag{2.25}$$

$$HXY2 = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} P_x(i)P_y(j) \log\{P_x(i)P_y(j)\} \tag{2.26}$$

Where: HX and HY are the entropies of p_x and p_y, respectively.

(14) Maximal correlation coefficient: $\text{MCC} = \left(\text{second largest eigenvalue of } Q(i,j)\right)^{\frac{1}{2}}$ (2.27)

Where:

$$Q(i,j) = \sum_{k=1}^{N_g} \frac{P(i,j)P(j,k)}{P_x(i)P_y(j)} \tag{2.28}$$

2.4 Symmetry

Symmetry is a common feature of natural shapes and man-made objects which can, therefore be effective for object detection and recognition. Symmetry is a complex property used by many species in visual interpretation [PAR08]. The nature of symmetry used in these cases is often imperfect. Symmetry detection can be considered to be a task of global optimization concerned with the determination of location, scale and orientation, determined from edges, contours or sets of points [FAR10] [MAS93].

2.4.1 Axes of Symmetry Analysis

An axis of symmetry is a line about which one side of an object is a reflection of the other side. Symmetry can also be found about a point. Detecting an axis of symmetry for an object is a complex task that can be simplified if prior knowledge of image structure is known. The ideal of object symmetry may be compromised due to imperfect illumination, image noise and occlusion making it difficult to observe perfect symmetry is natural images. Humans have an innate capability to recognize image symmetry. The symmetry recognised by people is often that of a component and imperfect.

The medial axis was first proposed by Blum [BLU67] to capture the global shape properties of an object. It is the locus of centres of the maximal disks that fit within an object. However, a single scale definition of the medial axis is sensitive to minor variations in the boundary of an object.

Gauch and Pizer proposed the Intensity Axis of Symmetry (IAS) method for describing the shape of structures in grey-level images [GAU93]. This method identified figural symmetry which describes the spatial and intensity variations of an image. Detecting points and axes of symmetry in an image can help to identify an object in an image. The IAS was derived by accumulating the points that belong to the medial axis representation of each level curve of an image having a level curve of the image as a boundary. The curvature extremes of level curves form connected curves in the image. The method depended on minimizing an active surface model to calculate the Symmetry Axis Transform (SAT) also known as the Medial Axis Transform (MAT). They developed active surface models to express changes in shape topology and axis segments across scales. This method was applied to shape-based image segmentation where it was possible to

identify parts of the object corresponding to individual components of the IAS. This method has contributed to the identification of axes of pedestrians and applied axes of symmetry detection to object detection and recognition. The method is computationally costly. It is also not obvious how this method can cope with the positions where symmetry surfaces cross over each other.

The interpretation of fine-scale detail was detached from the interpretation of larger-scale shape properties. Xu and Pycock [XUM98] presented a Concordance-based MAT (CMAT) which was a refinement of the Multi-scale Medial Axis (MMA) transform that used the symmetry of both boundary position and strength to avoid false medial responses, provide improved localisation of the medial responses and better identify the scale of symmetry. A key property of MMA computation is that it treats scale as a metric property such as the work of Gauch and Pizer [GAU93]. The MMA is important in that any object can be detected at a blurring scale proportional to the size of the object. The CMAT medialness responses were illustrated using a radiograph of a hand.

An appearance-based method for tracking human body parts was presented by Farenzena et al. [FAR10]. A set of local features that model three forms of appearance were extracted: the chromatic content, the nature of recurrent informative patches and the spatial arrangement of the coloured regions. The salient parts of the body were selected based on the localization of perceptual relevant human body parts using symmetry and perceptual principles. The horizontal axes that separate the torso, head and legs regions were identified and the vertical axis of symmetry estimated. The complementary aspects of the human body appearance were identified on each part that highlighting: the chromatic content, the region colour displacement and the presence of the structured patches. Performance was evaluated on ViPER [GRA07] and ETHZ [ESS07] data-

bases that have aspects of occlusions and illumination changes. Robustness to occlusion, pose, low resolution and illumination variation were verified to a number of single images and bunch of frames for individuals that varied continuously. The method was functioning in both single images of people and video sequence images. The evaluation rates and the computational cost were not given. The ability to detect axes of symmetry for other objects was not reported.

Edge features, contours and boundary points can be derived to represent sets of points or lines of symmetry in an image. Masuda et al. [MAS93] adopted an image similarity measure based on a directional correlation of edge features. They detected rotational and reflectional symmetry in an image using a search for rotation, translation and reflection transformations to identify the transformations to which parts of the image were nearly invariant to some congruent transformation which consists of translation, reflection and rotation. The symmetry of edges is difficult to be distinguished when the objects are small or the background is complex.

An effective approach to improve the identification of symmetry for component parts of an object in an image when the background is complex or the image has a high level of noise was proposed by Hu et al. [HUW06]. In this approach a composite view is constructed to identify the axes of symmetry for people in multiple camera views to improve the speed of computation. This method reflects the importance of axes of symmetry in detecting the major components of a person. The axes of symmetry were constructed as a result of matching the detected edges. Matching correspondence between multiple cameras was based on minimising the sum of distances between the detected ground-points of a principal axis of people in a view and the intersection of the principal axis of a person in another view. Detecting the axes of symmetry for people in each single cam-

era view was correctly detected even when the people are under a degree of occlusion. The method was evaluated effectively on a number of video sequences of people from the PETS 2001 and NLPR datasets [HUW06]. People standing close together were often identified as a single person. The computational cost of the axes of symmetry computation is complex. The method had a limited ability to identify the axes of symmetry for people and was not shown to adapt to other objects. The computational cost for the axis of symmetry computation in the above studies was not stated. Also, each reported method had a limited ability to identify the axes of symmetry for people and was not shown to adapt to other objects. The principal axis-based method [HUW06] matched people across multiple cameras based on principal axes of people while the CMAT [XUM98] considered the symmetry of both boundary position and strength and accurately estimated the position of the medial axis across scale. The CMAT provided a clearer description of shapes than the results of the method reported by Hu et al. [HUW06]. Moreover, the CMAT has a better estimate of medial axis position than the detection of principal axes of people using the principal axis-based method [HUW06].

It is concluded that the identification of axes of symmetry can form a part of a process for identifying composite axes. Further, the axis of symmetry detection has a role in the identification and recognition of people and other objects but symmetry detection for people was not shown to adapt to other objects. The method adopted in this thesis is able to identify the axes for people and readily adapt to other objects, with relatively low computational cost.

2.5 Edge Detection

Edge detection is the process by which the discontinuities between homogenous regions in an image are detected. These discontinuities may be due to changes in image intensity, colour or texture from one region to another. To follow image edges accurately an edge detector must be precise, respond to true edges only and be relatively insensitive to noise and artefacts. To be effective it must be reliable and computationally efficient.

The performance of an edge detector is assessed in terms of sensitivity, accuracy and precision of edge localization [PRA91] [PYC01].

Sensitivity: is a measure of the ability of an operator to detect an edge. It may be assessed as the ratio of the difference between the magnitude of the response at the detected edge and the average background response to the standard deviation of the background response of the image.

Accuracy: is the correctness of the detected edge position. It may be assessed as the mean of the absolute Euclidean distance between the detected and true edge position.

Precision: is the repeatability with which an edge position is detected. It may be assessed as the standard deviation of the absolute Euclidean distance between the detected and true edge position.

The above criteria are fundamental and well accepted criteria for assessing edge detection methods. The accuracy of any new edge detector is often assessed by comparison with the edge loca-

tions detected by the Canny edge detector [CAN86], described below, and by comparison with other edge detectors.

The Canny algorithm [CAN86] overcomes noise sensitivity by taking the difference between regions and processing across scale, identifying the steepest slope of an edge and using a threshold with hysteresis to track the boundary. The Canny operator adapts the size of support regions to the scale and structure of the object considered. However, Canny edges are often unsuitable in aesthetic appeal for stylistic depiction applications without further processing because edges representing traces or outlines are commonly expected to exhibit a certain amount of width and width-variability. In comparison, the Difference of Gaussian (DoG) [WIN11] operator, introduced in subsection 2.5.1, provides a good compromise between computational efficiency and stylistic versatility and achieves aesthetically pleasing edge lines without post-processing, particularly when synthesizing line drawings [KYP08].

The Maximum Likelihood Ratio (MLR) [ZHO97] criterion, introduced in subsection 2.5.2, was adopted to detect the boundary edge points of epithelial cells of a wide range of appearance in grey-level images. Zhou and Pycock [ZHO97] demonstrated that the MLR has high potentiality in extracting weak edges, high resilience and relatively low computational cost. Further, it was shown that the MLR has good localisation. A large support may limit resolution but it will enhance localisation. The effectiveness of the MLR edge detector was evaluated on synthetic images with defined levels of Signal to Noise Ratio (SNR). The performance of the MLR was evaluated in terms of sensitivity, accuracy of edge localization and precision of edge localization as shown in Table 2.4 [ZHO97].

Table 2.4. Evaluation of edge detection results using the MLR edge detector [ZHO97].

Criteria	SNR	
	0.2	1.0
Sensitivity	300	-
Accuracy	6	0
Precision	10	0

Table 2.4 presents a high selectivity for the MLR operator. It presents that the selectivity is preserved at all SNRs and shows that the MLR has a high sensitivity, whilst keeping a high degree of accuracy and precision.

Edge detection is normally based on a computation with scalar values. Colour is a vector quantity but edge detection in colour is normally defined in terms of scalar quatities. This ignores differences in the direction of a colour vector. Edge detection in colour images is more challenging than in grey-level images, given that colour is triple of values that might be represented as a vector. Colour edge detection methods have the potential to detect discontinuities that would not be apparent in a grey-level image. Novak and Shafer [NOV87] found that approximately 90% of edges in a colour image can be identified in the equivalent grey-level image, although the precise location of the edges may vary. They also commented that 10% of the remaining undetected edges might be important for certain applications.

Dutta and Chaudhuri [DUT09] reduced the influence of noise on colour edge detection by using an adaptive median filter. The average of the maximum colour difference at orientations of, 0^o, 45^o, 90^o and 135^o was computed by summing the RGB values at each pixel. A single automatically computed threshold was applied to generate an edge map in which the location, direction

and strength of the edge was recorded. However, some edges were absent from the output edge map. Masks oriented with the *x*- and *y*-axes were applied to create thin edge responses. The performance of this scheme was demonstrated by comparing results with Canny and Sobel edge detection methods for the location of edges using synthetic and natural images. The complexity of this algorithm is similar to the complexity of Canny and Sobel operators [DUT09]. The average execution time for processing a set of ten images of 816 x 616 pixels on a Dual core CPU of 1.73 GHz was 9.3 mS using this method [DUT09] compared to 3.2 and 5.3 mS for the Canny and Sobel operators, respectively. This algorithm [DUT09] was not appropriate for video sequences. Further the processing of colour images as three channel scalar images can lead to the generation of false colours and false edge responses.

Evans and Liu [EVA06] computed the Colour Morphological Gradient (CMG) as the difference between the dilation and erosion of the image. To reduce noise sensitivity a robust version of this algorithm (RCMG) was developed. RCMG employed a pair-wise pixel rejection scheme to provide a better estimate of the true gradient in the presence of noise. An estimate of the edge direction was used to enable non-maximal suppression in the CMG operator. Different parameter values on the RCMG behaviour in the vicinity of step edges were considered. The RCMG technique was evaluated both quantitatively and qualitatively. The quantitative evaluation of the RCMG edge detector was assessed using Pratt's Figure Of Merit (FOM) [ABD79] on synthetic and natural images using different levels of noise. It was shown to have good localization, noise immunity properties and low computational cost. Examination of the FOM figures for both independent and correlated Gaussian noise with a correlation factor of 0.5 presented that the FOM result was more than 92% for noise levels between 6 and 8. The FOM results for impulsive noise provided a simi-

lar result for both independent and correlated noise with a FOM of more than 95% for noise levels between 6 and 8. The performance of RCMG edge detector was compared with the Minimum Vector Dispersion (MVD) [TRA96] and compass edge detector [RUZ01] on a simulated colour image. This was illustrated by the edge detection results with a correlated factor of 0.5. The FOM for the MVD was 78%, reflecting that many true edge points were not detected. The RCMG reported a FOM result of 98%, while the compass edge detector produced a 97% which presented more noise responses than that of the RCMG. The performance of the RCMG is better than the performance of MVD edge detector and comparable to the compass edge detector. However, the RCMG is computationally complex and restricted to certain applications.

Edge detection is often used as an early step in a bottom up interpretation strategy. In model-based interpretation it can be used to find the edge points that sample the boundary of the objects for forming a model and interpreting an image. The edge points are often required to represent the cues in a model-based interpretation as the basis of a geometric model.

2.5.1 The Difference of Gaussians

The DoG was introduced by Rodieck [ROD65] to describe the spatial sensitivity of a dot of light of retinal ganglion cells. The DoG operator was used to detect edges of an input image without being affected by noise [DAV06]. The DoG function was obtained by talking the difference of two Gaussian functions with different spatial constants. When luminance values of pixels in an input image are given to the DoG filter, the edged image is obtained by convoluting input images over all pixels with the DoG function. That is, the edged images are produced by taking the difference between two Gaussian-smoothed images of the input image. Noise in the image can be

eliminated by selecting an appropriate value of the standard deviation of the first Gaussian function, σ_1, while the standard deviation of the second function, σ_2, determines the spatial resolving power in detecting the positions of edges with respect to the luminance values of σ_1 and σ_2. The luminance values of the edges become small and the resultant edged image becomes noisy. In contrast, when large σ_1 and σ_2 are used, the luminance values of the edges become large and noises will be attenuated. The edged image becomes blurry and it would be difficult to determine the position of the edges [DAV06]. The DoG can be used to increase the visibility of edges and other details present in an image [SAT08]. It involves the subtraction of a blurred version of an original grey-level image from a less blurred version of the image. The blurred images are obtained by convolving the original images with Gaussian kernels having varied standard deviations. The final image is calculated by replacing each pixel with the difference between the two blurred images and detecting when the values cross zero. The resulting zero crossings focuses at edges or areas of pixels that have some variation in their surrounding neighbourhood.

The Gaussian function in one dimension is the probability density function of the normal distribution, defined as [AME14]:

$$f_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} \tag{2.29}$$

Where: x is a parameter of the Gaussian function and σ is the standard deviation of the distribution.

The two–dimensional Gaussian function is as follows [AME14]:

$$f_\sigma(x,y) = f_\sigma(x).f_\sigma(y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.30)$$

Where: (x, y) is a two-valued coordinates of the Gaussian function.

The mask elements of the Gaussian filter can be calculated from the following formula [ALH91]:

$$f_\sigma(x,y) = \frac{1}{\sigma} e^{\{-[(x-x_0)^2+(y-y_0)^2]/2\sigma^2\}} \quad (2.31)$$

Where: (x, y) is a two-valued coordinates of the central mask element and (x_0, y_0) is the centre coordinates of the central mask.

The standard deviation σ controls the width of the function in the mask. If σ is varied in pixels and two Gaussian functions with two different values of σ are subtracted, the features whose scales between the two σ values are enhanced.

The DoG filter was constructed from the difference of two Gaussian functions as shown in Equation 2.32 [WAN12]:

$$f_{\sigma_1,\sigma_2}(x,y) = k_1 e^{(x^2+y^2)/(2\sigma_1^2)} - k_2 e^{(x^2+y^2)/(2\sigma_2^2)} \quad (2.32)$$

Where: x and y are the coordinates of a pixel in an image, k_1 and k_2 are the height factors of the Gaussian functions and σ_1 and σ_2 are the standard deviations of the two Gaussian filters.

The two smoothening Gaussian filters of the DoG must have different variances. The DoG functions were designed to allow some low spatial frequencies to pass, while attenuating high

spatial frequencies that often include random noise and further to obtain an image with only the desired frequency range. This is to facilitate the interpretation of the filtered images. The DoG of an image I is a function obtained by subtracting the image I convolved with the Gaussian of variance σ_2^2 from the image I convolved with a Gaussian of narrower variance σ_1^2, with $\sigma_2 > \sigma_1$. In one dimension, DoG was defined as [WAN12]:

$$f_{\sigma_1,\sigma_2}(x) = I * \frac{1}{\sigma_1\sqrt{2\pi}} e^{-(x^2)/(2\sigma_1^2)} - I * \frac{1}{\sigma_2\sqrt{2\pi}} e^{-(x^2)/(2\sigma_2^2)} \tag{2.33}$$

The DOG filter of 2-dimensional Gaussian function was defined as [WAN12]:

$$f_{\sigma_1,\sigma_2}(x,y) = \frac{1}{2\pi{\sigma_1}^2} e^{-(x^2+y^2)/(2{\sigma_1}^2)} - \frac{1}{2\pi{\sigma_2}^2} e^{-(x^2+y^2)/(2{\sigma_2}^2)} \tag{2.34}$$

2.5.2 Maximum Likelihood Ratio

The maximum likelihood principle [DUD12] can be used to estimate the likelihood that data values are derived from a particular distribution. The theoretical basis for the MLR formulation is drawn from a statistical analysis to determine that two data samples are drawn from similar or dissimilar populations. This criterion assumes a Gaussian distribution. This might not be valid in all situations but the normalising effect of taking the ratio of the likelihood between one sub population and a common population reduces the impact of any deviation from a Gaussian distribution. Assuming that both populations have a Gaussian Probability Density Function (PDF), as defined in [DUD12]:

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{\frac{1}{2}}} e^{\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]} \tag{2.35}$$

Where:

$P(\mathbf{x})$: is the PDF of drawing a particular value for a data vector, **x**.

d: is the dimension of the data vectors.

$\mathbf{x}$: is a sample d-component column vector from which a particular value is drawn.

$\boldsymbol{\mu}$: is the computed d-component mean vector of the population.

Σ: is d-by-d covariance matrix of the population.

$|\Sigma|$: is the determinant of the covariance matrix Σ.

Σ^{-1}: is the inverse of the covariance matrix Σ.

$(\mathbf{x}-\boldsymbol{\mu})^T$: is the transpose of $(\mathbf{x}-\boldsymbol{\mu})$.

The likelihood L_1 that a data sample can be split into two sub-populations, A and B of a region under investigation is evaluated as:

$$L_1(\mathbf{x}_A\mathbf{x}_B|AB) = \prod_{i=1}^{m_A} p_A(x_i|A) \prod_{j=1}^{m_B} p_B(x_j|B) \tag{2.36}$$

Where:

m_A and m_B: denote the size of regions A and B, respectively.

$\mathbf{x}_A$ and $\mathbf{x}_B$: are two sample vectors from the populations A and B, respectively.

A and B: are the two sub-populations of regions A and B, respectively.

x_i and x_j: are two data values drawn from the populations A and B, respectively.

$p_A(x_i|A)$: is the PDF for the sample vector, $\mathbf{x}_A$, that parameterise the populations of A.

$p_B(x_i|B)$: is the PDF for the sample vector, $\mathbf{x}_B$, that parameterise the populations of B.

The likelihood L_2 that a certain value drawn from AB, the combined populations, is defined as:

$$L_2(\mathbf{x}_{AB}|AB) = \prod_{i=1}^{m_{AB}} p_{AB}(x_i|AB) \tag{2.37}$$

Where:

m_{AB}: denotes the size of the combined region, *AB*.

x_i: is the i^{th} sample of $\mathbf{x}_{AB}$.

$\mathbf{x}_{AB}$: is the grey-level sample vector derived from the population, *AB*.

AB: is the parameter for the grey-level sample of the combined populations *A* and *B*.

$p_{AB}(x_i|AB)$: is the PDF of the grey-level sample vector, $\mathbf{x}_{AB}$, parameterized by *AB*.

Take the ratio of likelihoods $\left(\frac{L_1}{L_2}\right)$ and substituting the likelihood estimates of Equation 2.35 in Equations 2.36 and 2.37, gives the Likelihood Ratio (LR) as [DUD12]:

$$\text{LR} = \frac{L_1(\mathbf{x}_A\mathbf{x}_B|AB)}{L_2(\mathbf{x}_{AB}|AB)} = \frac{P(\mathbf{x}_A)P(\mathbf{x}_B)}{P(\mathbf{x}_{AB})} \tag{2.38}$$

Where:

$$P(\mathbf{x}_A) = \prod_{i=1}^{m_A} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_A|^{\frac{1}{2}}} e\left[-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_\text{A})^T\Sigma_A^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_\text{A})\right]$$

$$= \frac{1}{(2\pi)^{\frac{m_A d}{2}}|\Sigma_A|^{\frac{m_A}{2}}} e\left[\boldsymbol{\mu}_\text{A}{}^T\Sigma_A^{-1}\left[-\frac{m_A}{2}\boldsymbol{\mu}_\text{A} + \sum_{i=1}^{m_A}\boldsymbol{x}_i\right]\right] e\left[-\frac{1}{2}\sum_{i=1}^{m_A}\boldsymbol{x}_i{}^T\,\Sigma_A^{-1}\boldsymbol{x}_i\right] \tag{2.39}$$

$$P(\mathbf{x}_B) = \prod_{j=1}^{m_B} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_B|^{\frac{1}{2}}} e\left[-\frac{1}{2}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_\text{B}\right)^T\boldsymbol{\Sigma}_B^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_\text{B}\right)\right]$$

$$= \frac{1}{(2\pi)^{\frac{m_B d}{2}} |\Sigma_B|^{\frac{m_B}{2}}} e\left[\boldsymbol{\mu}_{\mathrm{B}}{}^T \Sigma_B^{-1}\left[-\frac{m_B}{2}\boldsymbol{\mu}_{\mathrm{B}} + \sum_{j=1}^{m_B} \boldsymbol{x}_j\right]\right] e\left[-\frac{1}{2}\sum_{j=1}^{m_B} \boldsymbol{x}_j{}^T \Sigma_B^{-1} \boldsymbol{x}_j\right] \tag{2.40}$$

$$P(\mathbf{x}_{AB}) = \prod_{i=1}^{m_{AB}} \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_{AB}|^{\frac{1}{2}}} e\left[-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_{\mathrm{AB}})^T \boldsymbol{\Sigma}_{AB}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_{\mathrm{AB}})\right]$$

$$= \frac{1}{(2\pi)^{\frac{m_{AB} d}{2}} |\Sigma_{AB}|^{\frac{m_{AB}}{2}}} e\left[\boldsymbol{\mu}_{\mathrm{AB}}{}^T \Sigma_{AB}^{-1}\left[-\frac{m_{AB}}{2}\boldsymbol{\mu}_{\mathrm{AB}} + \sum_{i=1}^{m_{AB}} \boldsymbol{x}_i\right]\right] e\left[-\frac{1}{2}\sum_{i=1}^{m_{AB}} \boldsymbol{x}_i{}^T \Sigma_{AB}^{-1} \boldsymbol{x}_i\right] \tag{2.41}$$

In Equation 2.38, $\frac{1}{m_A}\sum_{i=1}^{m_A} \boldsymbol{x}_i$, $\frac{1}{m_B}\sum_{i=1}^{m_B} \boldsymbol{x}_i$ and $\frac{1}{m_{AB}}\sum_{i=1}^{m_{AB}} \boldsymbol{x}_i$ are sufficient for $\boldsymbol{\mu}_{\mathrm{A}}$, $\boldsymbol{\mu}_{\mathrm{B}}$ and $\boldsymbol{\mu}_{\mathrm{AB}}$; respectively, in particular the sample vector means:

$$\left\{\widehat{\boldsymbol{\mu}}_{\mathrm{A}} = \frac{1}{m_A}\sum_{i=1}^{m_A} \boldsymbol{x}_i; \qquad \widehat{\boldsymbol{\mu}}_{\mathrm{B}} = \frac{1}{m_B}\sum_{i=1}^{m_B} \boldsymbol{x}_i; \qquad \widehat{\boldsymbol{\mu}}_{\mathrm{AB}} = \frac{1}{m_{AB}}\sum_{i=1}^{m_{AB}} \boldsymbol{x}_i\right\} \tag{2.42}$$

are also sufficient for $\boldsymbol{\mu}_{\mathrm{A}}$, $\boldsymbol{\mu}_{\mathrm{B}}$ and $\boldsymbol{\mu}_{\mathrm{AB}}$, respectively. Using these statistics:

$$\mathrm{LR} = \frac{\dfrac{e\left[-(\boldsymbol{\mu}_{\mathrm{A}} - \widehat{\boldsymbol{\mu}}_{\mathrm{A}})^T \left(\frac{A}{m_A}\right)^{-1}\left(\frac{\boldsymbol{\mu}_{\mathrm{A}} - \widehat{\boldsymbol{\mu}}_{\mathrm{A}}}{2}\right)\right]}{(2\pi)^{\frac{m_A d}{2}} |\ _A|^{\frac{m_A}{2}}} \dfrac{e\left[-(\boldsymbol{\mu}_{\mathrm{B}} - \widehat{\boldsymbol{\mu}}_{\mathrm{B}})^T \left(\frac{B}{m_B}\right)^{-1} \frac{(\boldsymbol{\mu}_{\mathrm{B}} - \widehat{\boldsymbol{\mu}}_{\mathrm{B}})}{2}\right]}{(2\pi)^{\frac{m_B d}{2}} |\ _B|^{\frac{m_B}{2}}}}{e\left[-\frac{(\boldsymbol{\mu}_{\mathrm{AB}} - \widehat{\boldsymbol{\mu}}_{\mathrm{AB}})^T}{2}\left(\frac{AB}{m_{AB}}\right)^{-1}(\boldsymbol{\mu}_{\mathrm{AB}} - \widehat{\boldsymbol{\mu}}_{\mathrm{AB}})\right] / (2\pi)^{\frac{m_{AB} d}{2}} |\ _{AB}|^{\frac{m_{AB}}{2}}} \tag{2.43}$$

$\widehat{\boldsymbol{\mu}}_{\mathrm{A}}$, $\widehat{\boldsymbol{\mu}}_{\mathrm{B}}$ and $\widehat{\boldsymbol{\mu}}_{\mathrm{AB}}$ are the maximum likelihood estimates for $\boldsymbol{\mu}_{\mathrm{A}}$, $\boldsymbol{\mu}_{\mathrm{B}}$ and $\boldsymbol{\mu}_{\mathrm{AB}}$, respectively. The maximum likelihood estimate for $\boldsymbol{\mu}_A$, $\boldsymbol{\mu}_B$ and $\boldsymbol{\mu}_{AB}$ must satisfy:

$$\left\{\begin{array}{l} \sum_{i=1}^{m_A} \Sigma_A^{-1}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_{\mathrm{A}}) = (\boldsymbol{\mu}_{\mathrm{A}} - \widehat{\boldsymbol{\mu}}_{\mathrm{A}}) = 0 \\ \sum_{j=1}^{m_B} \Sigma_B^{-1}\left(\boldsymbol{x}_j - \widehat{\boldsymbol{\mu}}_{\mathrm{B}}\right) = (\boldsymbol{\mu}_{\mathbf{B}} - \widehat{\boldsymbol{\mu}}_{\mathrm{B}}) = 0 \\ \sum_{i=1}^{m_{AB}} \Sigma_{AB}^{-1}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_{\mathrm{AB}}) = (\boldsymbol{\mu}_{\mathrm{AB}} - \widehat{\boldsymbol{\mu}}_{\mathrm{AB}}) = 0 \end{array}\right\} \tag{2.44}$$

That is, the maximum likelihood estimates for the unknown population means are just the arithmetic average of the sample means, $\widehat{\boldsymbol{\mu}}_{\mathrm{A}}$, $\widehat{\boldsymbol{\mu}}_{\mathrm{B}}$ and $\widehat{\boldsymbol{\mu}}_{\mathrm{AB}}$. Thus, Equation 2.43 reduces to:

$$\mathrm{LR} = \frac{\frac{1}{(2\pi)^{\frac{m_A d}{2}}|\Sigma_A|^{\frac{m_A}{2}}}\frac{1}{(2\pi)^{\frac{m_B d}{2}}|\Sigma_B|^{\frac{m_B}{2}}}}{\frac{1}{(2\pi)^{\frac{m_{AB} d}{2}}|\Sigma_{AB}|^{\frac{m_{AB}}{2}}}} = \frac{[(2\pi)^d|\Sigma_{AB}|]^{\frac{m_{AB}}{2}}}{[(2\pi)^d|\Sigma_A|]^{\frac{m_A}{2}}[(2\pi)^d|\Sigma_B|]^{\frac{m_B}{2}}} \tag{2.45}$$

The larger the sample regions of the LR operator the greater is the confidence in the likelihood estimates and the risk that detailed changes will be lost. Therefore, a balance is required between having a statistically adequate number of samples and suitable region sizes to resolve detailed structure. Assuming in Equation 2.45 that the size of regions $m_A = m_B = \frac{m_{AB}}{2}$ to simplify the computation of the likelihood ratio; then the ratio of MLR becomes:

$$\mathrm{MLR} = |\Sigma_{AB}|^2/|\Sigma_A||\Sigma_B| \tag{2.46}$$

Where: Σ_A , Σ_B and Σ_{AB} are the estimated covariance between the colour components of the respective regions *A*, *B* and *AB*, the combined region.

For a univariate distribution equation 2.46 simplifies to:

$$\mathrm{MLR} = \sigma_{AB}^2/\sigma_A\sigma_B \tag{2.47}$$

Where: σ_A , σ_B and σ_{AB} are the standard deviations of the respective regions *A*, *B* and *AB*.

The MLR operator was used in a resilient manner to detect weak edges in images [ZHO97], as described in Section 2.5, and segments in signal interpretation [PYC01]. In that work [PYC01] a model-based scheme for feature extraction and signal identification which uses MLR criteria for edge detection, in a sensitive and resilient manner, has been described and evaluated in [PYC01]. Signal models presume a parametric pattern for the underlying representation. New unseen signals can be classified by comparing the parameters extracted from the signals with the parameters of the signal model derived from a representative set of signals [PYC01]. It is difficult to derive a representative model and therefore to reliably interpret signals that vary greatly in form. Likelihood measures from the feature identification process were shown to provide a well behaved measure of signal interpretation confidence. It had been shown that complex, transient signals, from one of six classes, can reliably be identified at signal to noise ratio of two and that signal identification does not fail until the signal to noise ratio has reached one [PYC01]. The evaluation results presented that the loss in identification performance was produced from the use of a heuristic, rather than an exhaustive, search strategy is minimal.

2.6 Point Distribution Model

A PDM [COO92a] is a linear statistical representation of the geometric form of an object. An object in PDM is defined in terms of landmarks positioned on various object features, and at regular intervals in between. The PDM has been used to represent a wide range of objects including the shape of a resistor and of a hand, see Fig. 2.1. The primary landmarks are at the points of high boundary curvatures as highlighted by red dots. Those in between are secondary landmarks.

(a) (b)

Fig. 2.1. A contour with possible landmarks representing: (a) a resistor, (b) a hand [COO92a].

The PDM models object shape statistically by training with example shapes. The characterization of this model relies on the ability to label a consistent set of landmarks representing the shape of the object in an image.

2.6.1 Construction of PDM

The landmark points are the data from which the PDM is created. The landmarks are described by an ordered vector list of x and y coordinates. The i^{th} shape of an object in 2D is a vector of K landmark coordinates expressed as $= [x_{i0}, x_{i1}, \cdots, x_{iK-1}, y_{i0}, y_{i1}, \cdots, y_{iK-1}]^T$. A list of coordinates is collected for a number of objects and the coordinates for the training set registered to eliminate systematic variations in rotation, translation and scale prior to statistical characterization [COO92]. Variation is modelled by generating a covariance matrix of the aligned shapes and performing a Principal Component Analysis (PCA) to identify the principal eigenvectors and eigenvalues. The covariance matrix S of the training data is defined as:

$$= \frac{1}{N}\sum_{i=1}^{N}(_{\mathbf{i}} - ^{-})(_{\mathbf{i}} - ^{-})^T \tag{2.48}$$

Where: N is a set of shape vectors, $^{-}$ is the mean shape vector and $_{\mathbf{i}}$ is an example shape vector. The shapes represented by the model are obtained by a linear combination of the mean shape and the basis vectors [COO92]:

$$x = ^{-} + P_s \ _{\mathbf{s}} \tag{2.49}$$

Where: P_s is a matrix whose columns are the eigenvectors for the first *v* modes of variation found in the training set and is a vector of weights that controls the modes of shape variation.

The model is the mean and variance of the parameters. Each eigenvector describes a mode of variation of the PDM. The *v* unit principle vectors are used to vary the model through ±3 standard deviations of the mean and match it to landmarks in an unfamiliar image. The *v* unit principle vectors with the largest eigenvalues are used to represent the variation of the model [COO92]:

$$\sum_{k=1}^{v} \lambda_k \Big/ \sum_{i=1}^{2K} \lambda_i \geq p_v \tag{2.50}$$

Where: λ_k is the k^{th} largest eigenvalue corresponding to the k^{th} eigenvector, $\sum_{i=1}^{2K} \lambda_i$ the total variance of the training set and p_v is the fractional variation expressed by the first *v* unit eigenvectors.

Vary the model projection to obtain the best match of the model to the object in the image being interpreted. A variant shape vector of the model, ´, can be constructed by finding the weighting parameter, such that:

$$\bar{} \; (\acute{} - \bar{\mathbf{x}}) \tag{2.51}$$

An instance of a model is described by the parameter vector, , combined with the transformation from the model coordinate to the image coordinate. This can be a similarity transformation function identifying the position,(x_z, y_z), scale, *s*, and rotation, γ, of the model in the image. The locations of the model points in image, ´, are described by:

$$\acute{} \quad T_{x_z,y_z,s,\gamma}(\bar{} + P_s \; \mathbf{s}) \tag{2.52}$$

Where: $T_{x_z,y_z,s,\gamma}$ denotes a translation by (x_z, y_z), a scaling by *s* and a rotation by γ.

2.6.2 Non-Linearity in the PDM

The PDM is based on linear statistics; for any particular mode of variation, such that the pattern of variation is linear as opposed to quadratic. The problem with the linear model matching is that the best match for the model may be at local optima that lie at some distance from the true optimal match. It was shown that by using alternative metrics for matching local optima can be avoided [HEA96]. Heap and Hogg [HEA96] presented a classification method which discovered the pivotal deformation present via statistical analysis of the training data, and automatically used pivot points on training data when required. The method was tested on natural and synthetic data.

The initial selection of landmarks may require manual intervention to identify suitable landmarks for each shape in the training set. The PDM is effective when a shape is well-defined with reliable landmark points at homologous positions on each shape [KOT97]. The process of landmark detection may presume the topology of the shape characterized [KOT97]. This is a significant limitation requiring programme changes for each family of objects. Landmark points must be selected with care to minimize non-linearity in the shape space. Landmarks were selected for inclusion manually and are a property of the object [KOT97]. To automatic the model construction they defined a non-linear objective function in terms of shape symmetries that handle the model construction as an optimisation problem. This objective function was defined to measure the properties of pose and parameterization of each shape to produce a compact and a specific model. The objective was constrained to be linear and optimised using a genetic algorithm [KOT97]. The generated models were better than hand built models but the computational cost was high.

Representing variations in shape that are represented by non-linear patterns in shape space the PDM provides a non-ideal representation [HEA97]. With polynomial regression the PDM [SOZ94] allows landmarks to move along combinations of polynomial paths varying shape parameters and capturing non-linearity in the model space. Although this compensates for some of the curvature represented within the training set, it does not adequately compensate for high order non-linearity. The fitting process is time consuming.

A Cartesian-Polar Hybrid PDM [HEA96] was used to model human hands because the standard PDM was not suitable for modelling the non-linearity presents when, for example, fingers were bent. This allows a non-linear training set to be projected into a linear space where PCA was used to represent the deformation. The landmarks had to be identified by hand and classified as either Cartesian or polar. The polar points required a pivot and axis reference which was chosen manually. Landmark points that do not rotate about a pivot point between examples were modelled in Cartesian coordinates. Experiments were conducted on natural and synthetic data from a simple jointed object. The specificity of the Cartesian-Polar Hybrid PDM was better than that of the PDM for objects with significant bending. It was commented [HEA96] that specificity and compactness of the PDM can be improved by performing a non-linear mapping of the shape space reducing the number of modes of variation needed and leading to a more compact model. Bregler and Omohundro [BRE94] suggested modelling the non-linear datasets of human lips using a shape space constraint surface for improving the specificity of the PDM. Surface constraints were introduced to the model by separating the space surface into linear patches using cluster analysis. However the dimensionality of these lip shape spaces is low and had low non-linearity.

Baumberg [BAU95] solved non-linearity in deformable models using a cubic B-spline representation. The control points of the B-spline represented as a PDM [COO92] were used to model the person's outline. More details about this method are presented in Section 2.9.

These methods improve the ability of the PDM to model shapes, but the use of a linearising projection of the shape space might not always be appropriate, as when the distribution of shapes forms a hollow region in the new space [HEA96] and therefore does not provide a solution for the heavier non-linearity. The method in [SOZ94] models a limited selection of non-linear deformations. The performance of the method [BRE94] is poor in situations where there is more than one degree of deformation.

A more generic solution to model non-linear representation is needed, but with the simplicity and speed of the linear model.

2.6.3 A Review of Selected Applications of PDM

Despite the limitations cited above the PDM has proven its usefulness in medical image analysis for locating the outline of the abdomen [COO94], in identifying bones in X-ray images [ZHE06], in industrial inspection [AIX03], tracking automobile trajectories [DEM05] and characterising the form of fish [TIL00].

Cootes et al. [COO94] applied the PDM to identify the outline of the abdomen and the prostate in Magnetic Resonance (MR) images and the left ventricle (LV) of the heart in echocardiogram images. In each case the model consists of a shape template describing how the points of each ob-

ject can vary within a valid configuration. To extract the LV from echocardiograms, they chose points around the ventricle boundary, the nearby edge of the right ventricle, and the top of the left atrium. The PDM was matched to the image by examining a region around each model point to calculate the displacement required to move it towards the boundary. These displacements were used to update the shape parameter weights. The shape model allows for considerable variability, but is specific to the class of structures it represents. The limitations of the approach are its inability to handle large changes of scale and orientation and sensitivity to partial occlusion.

The potential of the PDM was also demonstrated for building a model to reconstruct 2D and 3D representations of bones in X-ray images of the surface of the proximal femurs [ZHE06]. The model was established by an iterated image-to-model correspondence between the X-ray image and instances of the PDM model. The matching process identified a fraction of the best matching 2D point pairs between features identified from fluoroscopic images and those extracted from the 3D model. Eleven images of each bone were used to show the accuracy of this method. The images were from cadavers is matter for the application but is not of major importance. A mean reconstruction error of 1.2 mm was obtained when two fluoroscopic images of the same bone in the same body were used. It dropped to 1.0 mm when three fluoroscopic images were employed. Two or three images are insufficient to reach a conclusion regarding precision. The image-to-model correspondence and the reconstruction algorithm were computationally complex. The convergence of the 2D/3D reconstruction method depended on the initialization of pose and scale of the mean surface model.

In a system for classifying crystal stones according to their shape in the manufacture of chain [AIX03] the PDM was extended by defining a set of acceptable deformations and adding a constraint to the amount of deformation permitted by the PDM to improve the model. This constraint was based on statistical control with Hotelling's T^2 chart [JOO05] as used in industrial inspection to reject outliers. Thus, PDM accepts some deformation on the contour of the shape. The results have shown that different objects can be matched independently of their orientation, location and size. The results on new images were judged visually.

A PDM was employed to track automobile trajectories in closed circuit TV images [DEM05]. The models created represented object paths as an average trajectory and a set of deformation modes. Evaluation was performed on motion data extracted from a vision system that tracked radio-guided cars running inside a circuit. The deformation coefficients were adjusted to detect outlier trajectories and large deformations were controlled using Hotelling's T^2 statistic [JOO05]. The deformation modes were each normalized to unit variance. The results show that PDM was able to interpret trajectories, by considering them to be a complex shape.

A 3D PDM was developed to estimate the dimensions of Atlantic salmon from stereoscopic images [TIL00]. The model was matched to stereo images of a fish by minimizing an energy function based on probability distributions of shape, grey level, direction, distance and the magnitude. This function was implemented in an iterated method in which edges were selected by magnitude, direction, and proximity to the model. The selection of edge points to be matched to the model was based on a combination of edge strength, at the position of the model landmark, and distance from the model. The model was tested on two image sets. The training images were tak-

en from image set 1. This set was divided into 18 images for training and 18 images for an independent test set. In the first set of 18 images of 26 fish 73% of the fish were correctly identified in the presence of shadows. The modes of variation suggested that the most significant mode was due to the swimming motion of the fish. The key points were identified manually on the fish. This is not an ideal solution. In the second set of 11 images the average error in length estimation was 5%, with a standard deviation of ±2.8%. Some of failures were caused by large variations in scale and rotation and the overlapping of fish. The size of training and test sets were inadequate to draw a strong conclusion. Model matching failed to converge correctly in few cases. This might be because the orientation of the fish in the test image was very different from the initial.

The above applications were reviewed to show the potential effectiveness and efficiency of the PDM for a wide range of applications. The methods presented in [TIL00] and [ZHE06] demonstrate the potential of the PDM to model different objects in 3D with potentially small errors.

2.7 Active Appearance Model

The AAM, first proposed by Cootes et al. [COO98], is a statistical deformable method that combines both object shape and appearance.

2.7.1 AAM Shape of Form

A list of coordinates is collected for a number of objects, and the coordinates aligned to a common mean coordinate structure to normalise variations in translation, scale and rotation. A covariance matrix is computed, and PCA performed to identify the principal eigenvectors and eigenvalues as described in subsection 2.6.1.

2.7.2 AAM Texture Model

The notion of texture refers to the grey-level or colour pattern of pixel values. The sample of pixel values used to represent texture vector is denoted by:

$$\mathbf{g} = [g_1, g_2, \cdots, g_M]^T \tag{2.53}$$

Where: *M* denotes the number of pixel samples over the object surface.

To obtain consistent pixel values, images must be warped and interpolated within regions, as determined by the placement of landmarks points, in the mean model and between landmarks.

2.7.2.1 Image Warping

Image warping [COO04] was used to collect the texture values between the landmarks. The texture warping was performed by partitioning the mean model using a piece-wise affine warp based on the Delaunay triangulation [COO04]. This algorithm was employed to warp the key points and intermediate points by connecting three key points as a triangle. The pixels within each triangle were warped to correspond to define values for equivalent pixels in the geometric reference image. An affine transformation is computed between the control points in the image and the vertices of the triangulation in the mean geometric model. Each pixel in each image inside a particular triangle was mapped to a point inside the corresponding triangle in the geometrical reference image using the barycentric coordinate criteria and bilinear interpolation correction [MAR08].

2.7.2.2 Photometric Normalization

A photometric normalization was proposed by Cootes et al. [COO98] to reduce the impact of global illuminations amongst the colour training images for the AAM. Each vector, $\mathbf{g_k}$, is iteratively aligned to the mean vector by adjusting the scale γ_k and offset β_k:

$$\mathbf{g_k} = (\acute{\mathbf{g}}_\mathbf{k} - \beta_k \mathbf{1})/\gamma_k \tag{2.54}$$

Where: $\acute{\mathbf{g}}_\mathbf{k}$ denotes the actual vector pixel values sampled in the image and $\mathbf{1}$ is a vector of ones with the same length as $\acute{\mathbf{g}}_\mathbf{k}$.

The values of γ_k and β_k were selected to best match the vector to the normalised mean; that to obtain zero mean and unit variance. The values γ_k and β_k were calculated using:

$$\gamma_k = \acute{\mathbf{g}}_\mathbf{k}\bar{\mathbf{g}} \tag{2.55}$$

$$\beta_k = (\acute{\mathbf{g}}_\mathbf{k}\mathbf{1})/(3 \times M) \tag{2.56}$$

Where: $\bar{\mathbf{g}}$ is the mean texture vector.

The texture vectors are aligned to the mean texture vector which is recalculated inside each iteration loop until convergence. The search for convergence is stopped when the maximum number of iterations was reached or the newly estimated mean has converged, as defined by a threshold on the difference between the newly estimated mean vector at each iteration and the mean of all the aligned texture vectors.

2.7.2.3 Modelling Texture Variation

PCA was performed on the normalised textures to obtain a compact linear statistical model of texture [COO98]:

$$g = \bar{\mathbf{g}} + \quad_{g}\ \mathbf{g} \tag{2.57}$$

Where: g is the synthesised texture, $\bar{\mathbf{g}}$ is the mean texture vector, is a set of orthogonal modes of texture variations of the covariance matrix that contains the eigenvectors corresponding to the largest eigenvalues and $_{\mathbf{g}}$ are the texture deformation parameters.

The texture parameters for a given sample image can be retrieved using [COO98]:

$$\mathbf{b_g} = P_g^{-1}(\acute{g} - \bar{\mathbf{g}}) \tag{2.58}$$

Where: $\acute{g}$ is a texture instance from the model.

2.7.3 Combined Model of Shape and Texture

The shape and texture of any example were described by the vectors, and $_{\mathbf{g}}$, respectively. These vectors were concatenated so that any correlation between shape and texture variations can be considered using a common parameter vector. The property of correlation was used to build a combined statistical appearance model. A concatenated vector, **b**, was generated for each example image in the training set as [COO98]:

$$\mathbf{b} = \begin{pmatrix} \\ \mathbf{g} \end{pmatrix} = \begin{pmatrix} (\ \ ^{-}) \\ _{g}(g\ \bar{\mathbf{g}}) \end{pmatrix} \tag{2.59}$$

Where: is a diagonal matrix of weights between shape and texture vectors, which measures the unit difference between the texture and shape parameters.

PCA was applied to these vectors giving the further model:

$$\mathbf{b} = Q\mathbf{c} \tag{2.60}$$

Where: **c** is a common parameter vector that controls both shape and texture information, Q identifies the eigenvectors of **b** and **b** denotes a vector of appearance parameters.

The linearity of the model allows the shape and texture to be defined using the combined model:

$$x = \bar{\mathbf{x}} + P\,W_s^{-1}Q_s\mathbf{c} \tag{2.61}$$

$$g = \bar{\mathbf{g}} + P_g Q_g \mathbf{c} \tag{2.62}$$

Where: and are matrices describing the variation modes of shape and texture, respectively.

2.7.4 Interpretation with the AAM

AAM interpretation procedures were treated as an optimisation problem in which the texture residual vector was minimized by updating the model parameters [COO98] [COO01b]. To find the best match between a model and an image, it is important to minimise the magnitude of the difference vector between the previously unseen image and one synthesized from the appearance model. The AAM search algorithm varies model parameters to minimise the difference between a previously unseen image and the model instance in order to generate a synthetic image that matches as close as possible to the unseen image. The AAM search algorithm proposed by Cootes et al. [COO98] was executed iteratively and the quality of fit optimized using least squares criteria. Cootes et al. [COO98] described an AAM matching algorithm that alternates between matching shape and appearance. This approach demonstrated a rapid convergence. The

mapping from error images to AAM parameters was modelled by linear regression. A good match to an unseen image was rapidly achieved, even when the starting position was poor.

In a later approach [COO01b] the regression estimates were replaced by a simplified Gauss-Newton procedure, where a Jacobian matrix was evaluated from the training data. The error surface for both approaches was approximated by a quadratic function. This has the advantage that, during training, not all difference images need be held in memory.

Despite the great success of AAMs in many domains, there are some limitations; it is impossible to capture a complex non-linear shape and appearance variations for a large image set with a single PCA. Also it is assumed that the appearance and shape parameters are linearly related around the optimum match between the image and the model.

2.7.5 AAM Enhancement

The basic application of the AAM [COO98] does not demonstrate the full flexibility of a statistical appearance model; this is better demonstrated by [BAT05] where the Jacobian matrix was linearly adjusted according to the texture configuration of the target image, in the generalised AAM [SAU11] which uses non-linear regression models and in the work of [GAL06] where distance maps were used in the place of texture pixels to adjust the texture vectors.

Batur and Hayes [BAT05] proposed an Adapted AAM (AAM), in which the gradient matrix was adapted linearly at each iteration. The contributions of the texture eigenvectors to the gradient matrix were added to the fixed gradient matrix according to the composition of the target image.

A high degree of flexibility in the texture component of the model was demonstrated in the recognition of facial expressions under large texture and lighting variations. They examined the AAM and AAAM on four subsets of test images of faces with varying lighting angles, shadows and illumination. They concluded that using the median values of the shape and texture error distributions that the AAAM performed better than the AAM in all the situations considered. They presented the percentages of the runs for the AAM and AAAM with five forced iterations for each subset of the test images. The final shape mean error of AAM was 0.30 pixels. There were few cases where the AAM failed to converge, with no such cases for the AAAM. This is probably due to the greater reliability of the AAAM. They also demonstrated that with AAAM the average shape and texture errors were reduced when the number of modes was increased. AAAM provided a great performance increase over the fixed gradient matrix approach in the expense of an increase in computational cost.

Generalisation within the AAM framework was addressed by Sauer et al. [SAU11] using two non-linear regression models: boosting [FRI00] and Random Forest (RF) [BRE01]. Each stage in the sequence consisted of a shape model and a corresponding regression model. The trees of the RF in 1D regression were built recursively and at each node the training data was split by selecting a threshold on a feature variable selected randomly from a subset of all features to decrease the of sum of squared errors. The trees were constructed until each node contained a single sample. The Boosting method additively combined a set of weak learners into a strong regression function. A Haar-like feature [VIO01] was computed on the pixel sample vectors to update the model parameters. The Haar-like features provided a superior generalisation performance and a lighting-insensitive feature for capturing facial features. The Boosting and RF algorithms were

also combined with Haar-like features [VIO01] and illumination-independent features derived from an integral image [CAL10] to handle a large degree of variation in lighting and appearance. The relative difference between pixels values across images were captured by these features, making them invariant to monotonic transformations. The RF and Boosting algorithms were assessed on the XM2VTS [MES99] and BioID [JES01] face datasets by cross-comparing the performance of the resulting algorithms. Both datasets consists of frontal face images. The XM2VTS dataset contains a little pose and lighting variation between instances in high-resolution images. The BioID dataset contains a large proportion of low quality images of faces with a significant degree of variation in pose and illumination. The results demonstrated that the RF generalises well. This was confirmed by the percentage of successfully converged images particularly when training on the XM2VTS dataset. However, this method has a high computational complexity.

The Random Forest (RF) [BRE01] [BRE07] is a collection of decision trees that vote on the correct classification of the input data. Each decision tree is trained on a subset of data. Each decision tree is used individually to vote for one class using its features and the RF predicts the class that has the most of votes. Each tree is grown with a randomized subset of predictors. In constructing each tree of a RF, a bagged training sample is selected by drawing a random subset of n instances from the n-member training set, with replacement from the data that replaces missing data using the median of the non-missing data. The same number of vectors as in the original set are randomly selected. When vectors are chosen by replacement that some vectors will be repeated and some will be absent. At each node of each trained tree a new random subset of parameters are created and used. The size of each node and tree is fixed. In bagging, multiple training sets are selected with replacement trees fit to these samples [BRE07]. Bagging improves

model stability, avoiding overfitting. At each splitting node in a classification tree, a random subset of predictor variables is used to define the best split. A random number of predictor variables are used at each node. The RF can rank predictor variables by importance so that the less important can be pruned.

Modelling high resolution 2D and 3D images in AAM is relatively slow due to the high storage and computational demands. Gallou et al. [GAL06] used maps of distance from the boundary instead of texture pixels in the basic AAM to improve matching to facial features in images with variable illumination. The method was evaluated on several images of faces from the Carnegie Mellon University Pose, Illumination, and Expression (CMU PIE) database [SIM02]. This approach demonstrated less sensitivity to illumination change. The basic AAM and the AAM with distance maps were compared over two sets of face images under the same lighting variations. Errors were expressed as a percentage of the distance between the ground truth points and model points. The mean error for the two sets of face images using AAM with distance maps were 0.1 compared to mean errors of 0.2 and 0.3 using the basic AAM showing that the use of distance maps enhances robustness against lighting variations.

There is scope to improve on the basic AAM to better accommodate the variations in appearance for complex situations where objects and lighting vary greatly.

2.8 Eigenface Method

The Eigenface method for facial representation and recognition [TUR91] is based on a Karhunen-Loeve Transform (KLT) [KIR90]. In the Eigenface method the KLT approximates a set of im-

ages by a low dimensionality subspace from which features are extracted and the Mean Square Error (MSE) between the model and the feature vectors on the subspace used to reduce the computational cost of matching. The Eigenface method [TUR91] used a nearest mean classifier on a dataset of 2,500 face images of 16 people, digitized at 3 head orientations, with 3 head sizes and 3 lighting conditions. The system was invariant to changes in illumination, but performance degraded with changes in head size, head orientation and scale. Faces that varied in scale by more than those in the original dataset were not readily recognized. The authors suggested a multi-resolution method to overcome this problem. Performance decreased when the face orientation was not fixed.

The benefits of the Eigenface method are that it is simple, computationally efficient, uses raw image data, has a modest memory requirement and does not require prior knowledge about the geometry of faces. However, the method is sensitive to changes of scale. Further, changes in appearance such as created by the wearing of spectacles, pose, illumination, occlusion, and facial expression reduce the rate of correct classification. Enhancements made to overcome such limitations are described by Pentland [PEN94] and Murase [MUR95].

In [PEN94] a view-based eigenspace method and extension to 3D with greater resilience to changes of pose and illumination was reported. The recognition of human faces was tested under unconstrained viewing conditions, using multiple sets of eigenvectors, one for each face orientation. The residual error for each view space was calculated to identify the orientation of the test face and to select the eigenspace that best described the input image. Whilst the view-based

method gave a more accurate representation of faces, it was more computationally intensive than the standard Eigenface method [TUR91].

Object learning requires the acquisition of large image sets and a computationally complex process to generate eigenvectors. A compact Eigenspace representation parameterised by pose and illumination was reported by [MUR95]. This approach allows the 3D appearance of objects to be learnt from 2D images. Each object was represented as a parametric manifold in two eigenspaces; the universal eigenspace and the object's eigenspace. The eigenspace for the image set was constructed by computing the most prominent eigenvectors of the set based on the KLT. The eigenvectors (individual images) were projected to the eigenspace to obtain a set of points parameterised by pose and illumination. A set of experiments were conducted using objects with complex appearance characteristics. The recognition and pose estimation were studied using over a thousand input images of sample objects. The images were automatically normalized in scale and brightness. Each normalized image was 128 x 128 pixels in size. For each object they used 5 different light source directions and 90 poses for each direction. The reported results suggest that real-time appearance recognition would be possible. This approach requires a re-computation of the entire eigenspace to add new objects incurring a high computational cost. Appearance representation for the objects in eigenspaces with 20 dimensions produced precise recognition results with an average pose estimation error of 1.0 degree. The recognition rate was not specified. The computational cost increases with image resolution and the number of images in the training set.

The papers reviewed show that the Eigenface method is a reliable method for face recognition.

2.9 B-Spline Curve Modelling

The B-spline curve representation is a piece-wise approximate model, with an individual curve for each segment constructed between points and defined about separate control points. The control points determine the path and the shape of the curve. The generation of a B-spline curve is started by creating the parameters of the data points, generating a knot vector and finished by solving a system of parametric B-spline functions for curves as defined in [WAN90]:

$$\mathbf{s}(\mathbf{q}) = \sum_{i=0}^{n} p_i B_i(q) \tag{2.63}$$

Where: p_i are the control polygon vertices, $B_i(q)$ are the normalised B-spline basis functions defined on the knot sequence for the curve, n is the number of data points. The position of the curve **s(q, h)** in a surface with parameters q and h can be defined as [WAN90]:

$$\mathbf{s}(\mathbf{q}, \mathbf{h}) = \sum_{i=0}^{n} \sum_{j=0}^{m} p_{i,j} K_i(q) M_j(h) \tag{2.64}$$

Where: $p_{i,j}$ is the control polyhdren, $K_i(q)$ and $M_j(h)$ are the B-spline basis functions in q and h directions on the surface, n and m represent the size of data in directions q and h, respectively.

A B-spline approximation is generated from the control polygon vertices. The B-spline curve does not generally pass through all the control points. For the B-spline pass through all the control points, a careful selection control points and of linear equations is necessary, as defined in Equation 2.64 [WAN90].

The parameterization of the B-spline leads to a representation for a wide range of shapes from a small set of sampled points, where the knot vector specifies the parameter interval for the seg-

ments that make up the B-spline. Closed curves are formed by connecting the first and last points of the B-spline, although continuity will not be maintained automatically.

Methods for B-spline matching are based on: 1) minimising the MSE from the data curve to find the best number of points for the B-spline [COH94] or 2) minimizing the MSE for coarse-matching at the corners of object boundaries [GUY00]. These methods have a low computational cost but are sensitive to noise and do not benefit from the continuity of B-spline curve because the MSE method does not protect against outliers.

Notable B-spline methods include curve matching for deformable shapes using sparse spline 2D knot points [LEE03] as applied to object detection and the use of a cubic B-spline for pedestrian tracking [BAU95]. These methods are discussed below.

Lee et al. [LEE03] reported a B-spline curve matching method for deformable shapes. The strain differences of the spline approximations and the deformation energy of thin plate spline mapping were calculated between knot points and normalized local curvature to derive the mapping parameters between two sets of corresponding points. Point-correspondence for sparse knot points was achieved by matrix matching. An example image of blobs with different shapes was used to test shape detection. Each blob in the input image was extracted and its boundary was approximated with a spline curve. The cost of matching was low. They showed examples of spline curve fitting where a few knot points in the matching process, allows the splines to approximate object boundaries. Only a few knot points were used in the matching process. Therefore, the algorithm was fast and applicable to real-time tasks such as industrial robot vision and target detection.

Baumberg [BAU95] used the control points of a cubic B-spline representation in a PDM [COO92] to model the outline of a pedestrian. The curvature of the B-spline took on some of the non-linearity of the model and therefore reduced the problems presented with using a linear PDM to represent non-linear models. The B-spline training shapes were aligned by scaling and translation, and eigenvectors computed for the covariance matrix of the B-spline control points. The model was composed of a mean shape and an orthogonal basis extracted by analyzing the variances of the shapes. The initial shape estimate was provided by the trained mean pedestrian shape. The shape was parameterized from one fixed point and the length around the contour. The fixed point used was the upper most point at which the principal axis crossed the object boundary. The current position and shape estimates were adjusted in response to measurements made on the image. A point by point difference between the current image and its pre-computed background was applied. The contour of the person was found by the measurements being made at various points around the current shape. The search was carried out along lines aligned with the Mahalanobis distance measure. In this fitting process, the system allows for a certain percentage of the control points not to be matched in the current image. This improves tracking when parts of the person are occluded. The PCA parameters and the position of the person in the image were used to aid tracking. A Kalman filter was used to model the speed of movement and to predict position in the current frame. The initial positional estimates and the current shape parameters were used as a starting point for model fitting. Repeated measurements made along the Mahalanobis search direction at the control points of the B-spline were used to find the new position and the person's outline in the current image. This is a complex system for tracking a pedestrian alone and often fails if the people are not amongst the objects modelled. It is effective if the pedestrians are well separated and can support a good degree of occlusion for images of high con-

trast. The results were good, suggesting that the parameterization process was consistent. This method has been applied to real-time interpretation.

The above studies illustrates that the B-spline was used to generate a curve that is flexible and controllable when applied to interpret deformable shapes [LEE03] such as the outline of a pedestrian [BAU95]. However, some curves generated with B-spline might be difficult to control and can be unstable. The reviewed previous research shows that the B-spline method is a flexible method for modelling form.

2.10 Superellipses

A SuperEllipse (SE) is a closed curve that may be represented in a canonical implicit form by the contour surface, in a generalisation of an ellipsoid as [DUR08]:

$$f(x,y) = \left(\frac{x}{a}\right)^{\frac{2}{\alpha}} + \left(\frac{y}{b}\right)^{\frac{2}{\alpha}} - 1 = 0 \qquad (2.65)$$

Where: *a* and *b* define the size of the SE along the major and minor axes, respectively and α, the shape coefficient of the SE, is the angle of orientation in the x - y plane, expressed in the object centred coordinate frame. They are each non-zero positive real values.

An SE can represent a wide range of shapes from a circle to a rectangle to star like shape with the adjustment of a small number of parameters. The parametric form of SE defines real values that can be plotted as $\alpha$is varied. The implicit equation of SE was popularized by Piet Hein [GAR65] who described shapes in terms of the relative distance from a given 2D point to a SE surface. The solution of Equation 2.65, in parametric (explicit) form is:

$$\begin{bmatrix} x = a\text{sign}(\cos\theta)|\cos\theta|^{\alpha} \\ y = b\text{sign}(\sin\theta)|\sin\theta|^{\alpha} \end{bmatrix} \tag{2.66}$$

Where: the exponent, α, is called the squareness parameter of the SE that defines the angle of orientation in the x - y plane and θ is an independent parameter in the parametric form of the SE in the range $[-\pi, \pi]$, which is used along other parameters to allow the parametric equations to generate the SE shapes.

Fig. 2.2 shows the pseudo-code for computing the spatial coordinate parameters *x* and *y* in the x - y plane using the SE in parametric form, given the scaling and exponent parameters. The angular parameter takes the values in the range of $-\pi$ to π.

$\|.\|$	calculates an absolute value.
sign	returns 1 if input is positive, -1 if it is negative and 0 if it is 0
a and *b*	are scale parameters that define the size of the SE in the x and y-axes.
α	the exponent coefficient of the SE in the x - y plane.
θ	an independent angular parameter of the SE in the range $[-\pi, \pi]$.
x and *y*	are 2D spatial coordinate points

1. Read α
2. Set the values of *a* and *b*
3. For θ $-\pi$ to π : $\pi/16$
   3.1 $x(\theta) = a\ \text{sign}(\cos\ (\theta))|\cos\ (\theta)|^{\alpha}$
   3.2 $y(\theta) = b\ \text{sign}(\sin\ (\theta))|\sin\ (\theta)|^{\alpha}$
   End For

Fig. 2.2. A pseudo-code description of the superellipse to compute the coordinate parameters *x* and *y*.

The SE curve in Equations 2.65 and 2.66 can represent a wide range of symmetrical shapes such as rectangle, circle and an ellipse by varying the scales and angular parameters of the SE. For example, with α = 0, the curve takes a form of rectangle ($a \neq b$) or square (a=b). As α increases but

is still less than 1 a series of rectangular shapes with rounded corners are generated until an ellipse or circle is obtained when $\alpha = 1$. A diamond shape is obtained when $\alpha \approx 2$ and a pinched shape when $\alpha > 2$, see Fig. 2.3.

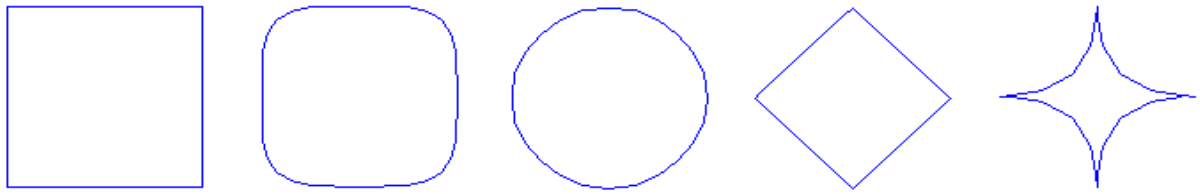


Fig. 2.3. The SE shapes generated for constant values of *a* and *b* and as α is varied, the shape progression is: (a) square; (b) square with rounded corners; (c) circle; (d) flat bevelled and (e) pinched.

The intrinsic symmetry of the SE arises because the exponents are the same for x and y-axis terms and constant. The compact SE is a compact representation.

SuperQuadrics (SQs) are defined as the spherical product of two 2-D superellipses [BAR81]; SQs form a family of parametric surfaces in three dimensions which can model a wide variety of objects.

2.10.1 Superquadrics Literature Survey

SQs were widely adopted in a wide range of applications because their mathematical representation is simple and they use a relatively small number of parameters. Barr [BAR81] introduced the SQ representation to computer graphics and formulated the deformation of SQs for linear axial twisting, tapering and bending. SQs were introduced to computer vision by [PEN86]. He used SQs with parameterised global deformations as a model for object recovery. The use of the pa-

rameterised implicit function of an SQ simplifies the optimization involved in curve fitting. Pentland argued that SQs represent natural objects better than simple primitive shapes such as circles. His initial formulation of the SQ to recover object form proved unstable and computationally complex. He later developed a stable method for both segmentation and fitting deformable SQs using the minimum description length in a segment-and-fit paradigm [PEN87]. He recovered SQ models by searching over the entire SQ space and integrated segmentation with the SQ model. An exhaustive search was used to allocate the best initial values for the model. The computational burden of this process was high. Pentland [PEN90] later proposed a segmentation method which matched 2D projections of 3D SQs of different shapes, orientations and scales to the image. After this segmentation, 3D SQ models were fit to range points for individual components.

A deformable SQ model incorporating both global and local deformations in a physics-based representation was developed by Metaxas and Terzopoulos [MET91]. Global deformations captured the salient structure of object components and local deformations structural detail. Local and global deformations were induced by converting data into forces. The physically based models were controlled by equations of motion to adjust the rotational, translational and deformational degrees of freedom of the models. The equations allowed the deformable SQs to react to externally applied forces drawn from image to range data so that the model could be adapted to match the visual data. The model was helpful in reconstructing 3D objects and extracting global shape features. These techniques provided a robust framework for fitting, and presented the possibility for a natural extension to dynamic scenes. These models have the benefit of modelling non-symmetric objects and simultaneously satisfying the requirements of shape reconstruction and recognition. However, the transition between local and global shape deformation was not smooth.

SQs have been used to identify shape primitives or "geons" in single view range images [RAJ92]. Geons were classified on the basis of axis shape, cross-section shape, cross-section edges and cross-section symmetry. The SQ was applied to determine the principal axis of the geon and shape discrimination. The deformation parameters of a SQ were determined for synthetic and natural range images, acquired from a number of perspectives, of models belonging to 12 shape classes corresponding to a set of 36 different geons. Five features were derived from the estimated SQ parameters to differentiate between these 12 shape classes. They investigated the recognition of geons from SQs fitted to range data, but not the identification of objects composed of assembled geons. Classification error rates were estimated for binary tree and k-nearest-neighbour classifiers. The results indicated that the shape attributes can reliably be inferred from SQ parameters, with a simple choice of features. The qualitative shape properties were about 80% reliable for range images of objects with smooth surfaces using the features derived from the estimated parameters of SQ and a binary tree classifier.

Park et al. [PAR94] proposed the use of SQs to represent local and global shape deformation. The object's shape detail was represented by the local deformation parameters. The global parameter functions improved the accuracy of representation of an object's shape for the investigated applications in terms of a few intuitive parameters such as functional twisting and bending, where their values varied across the primitive's shape. The diversified global parameters were independent of the underlying shape. The global parameter functions made it easy to find a compact representation for complex shapes, which would support shape reconstruction and recognition. Evaluation was performed on MR images of the LV of the heart for normal and abnormal heart movement during systole. The root mean square error between the interpretation and the manu-

ally defined boundary was low at less than 5%. The model was applied to abnormal and normal hearts to consider how well it could interpret what is normal LV motion and in particular the effects to the LV motion of the various LV diseases.

SQs were used in a Context-Based Image Transmission (CBIT) scheme to optimise bandwidth utilisation when large volumes of medical image data are transmitted, especially over low-bandwidth channels [SAL99]. This was achieved by providing a context for the transmission process and a mechanism to update key image regions. An approximate iconic image was used to identify the gross structure. The iconic image was constructed using a combination of shape-tree and SQ representation [SAL99]. A wide range of complex shapes were represented compactly with a small set of parameters.

A model in which a complex shape is decomposed into simpler components, each of which is then modelled by a SQ was presented by Krivic and Solina [KRI04]. They described a SQ model which represented articulated objects by their parts. The system was based on a tree representation that modelled the flexible articulated objects of a human figurine. A symmetric shape was easily constructed by a SQ and modified by the addition of sub-regions to form an asymmetric shape. A recognition system of articulated objects was used to search for matches between parts and their rough shape in a scene and parts of the modelled objects using interpretation trees. They argued that the configuration of simple sub-parts and their rough shape should provide sufficient constraints for successful matching. The object shapes were constructed from range images using their recover-and select Segmentor program. The representation with the recover-and select Seg-

mentor program is more compact if regions are added and the objects are represented clearly by the SQ. However, this system was computational costly and unstable on some shapes.

An SQ alone was shown to model a wide range of symmetric shapes [PAR94] [MET91]. An SQ model with a shape tree has been shown to be effective for representing a wide range of asymmetric shapes [KRI04] [SAL99]. However, the precision of object shape representation using an SQ is limited and the interpretation is largely procedural refining the interpretation until the error is sufficiently low. For a SQ model to be applied to model a wide range of objects in images this model needs adapting to a model-based strategy.

2.10.2 Superquadrics and Superellipses Curve Fitting

SQ curve fitting is commonly performed by a least square minimization to match points on the SQ to point in an image, as the SQ parameters are varied. SQs have been applied to segment range images with a non-linear least squares minimization method [GRO88]. A Euclidean Error Of Fit (EOF) function was suggested by Gross [GRO88] but it introduced a high-curvature bias that produced counter-intuitive results. Rosin and West [ROS95] presented the use of Powell's optimization technique to fit the superellipses to minimize an error metric based on Euclidean distance by segmenting curves into a series of super elliptical arcs. The various shapes in the images were well represented by the superellipses.

Pilu [PIL99] investigated an approach for fitting an SE curve with a statistical PDM. A SE model was used to generate a large random synthetic set of deformable SE shapes in order to train a PDM, from which linear deformable SEs were be generated. Pilu commented that the SE mathe-

matical model could be substituted with other representations to generate more appropriate shapes. The fitting was carried out as in [COO92] and a PDM initialized by fitting ellipses to pixels belonging to a small set of seed points as described in [PIL97]. The PDM was used to represent the variability of the SE model in terms of size, bending, tapering and squareness. They observed that a PDM initialized using ellipses with geometric parameters [PIL97] geometric parameters [PIL99] were able to similarly represent shapes. This algorithm [PIL99] is versatile, efficiently and can readily be adapted to new applications. It converged well and did not require a good initial set of parameter values.

Fitting an SQ to partial data is difficult because reliable parameter estimates cannot be made with partial data. Iterative methods can sometimes minimise predefined objective functions [ZHA03]. Fitting an SQ to partial data using a conventional algebraic distance did not provide a sufficiently dense distance map to guide the optimisation process because it generates a series of discrete distance values [ZHA03]. The approach described by Zhang and Rosin [ZHA03a] improved the fitting process to partial data by augmenting the algebraic distance with gradient and curvature information. An SQ was fit by finding a set of model parameters that minimised the sum of the squares of the distances between the model curve and a given set of pixel data [ZHA03]. To find the Euclidean distance between the data and the SQ curve a quadratic equation must be solved.

An iterative gradient descent method to recover deformed SQ models from range data, by a least-squares minimization was introduced by Solina and Bajcsy [SOL90]. An SQ was fit to range images to enable a robot to grasp and manipulate objects [CIP03]. In their Simultaneous Segmentation and Superquadric Fitting (S3F) method range data was matched to an intrinsic model. They

proposed a solution for self-occlusion and reported numerical experiments on 2D and 3D data that improved the fit of SQs to laser-range data collected from moving ground-robotic platforms. SQs provided a well-developed mathematical foundation for the recovery of surfaces from range data and for a concise shape description. However, they reported that recovering a SQ from range data is sensitive to noise and outliers, and that stability is difficult to achieve.

Selecting an appropriate objective function for SQ fitting was addressed in [GRO88] [ZHA03]. Experiments were performed to characterize the behaviour of objective functions used to fit 3D SQs with curves [GRO88] [ZHA03]. Dense synthetic range data was considered in [GRO88] and natural data in [ZHA03]. The concavity of the objective function towards the minimum and the accuracy of the recovered parameters were analyzed. Points from complete and partial laser scans were fit. The analysis presented in [GRO88] showed that a radial objective function outperformed other objective functions in terms of precision and sensitivity to parameter changes. The method of fitting accommodated points occluded from the view of the laser scanner. They concluded that objective functions based solely on distance and volume performed well in fitting an SQ to scanned points with one self-occluded side along a segment.

A Euclidean objective error function [GRO88] was employed to fit the SQ by minimizing the distance between the data points on the shadow contour and the SQ. Further, metrics similar to those used for the ellipse fitting [ROS93] and polynomial fitting [TAU91] were investigated with an algebraic distance measure, defined in Equation 2.65. They chose to minimize the Euclidean dis-

tance d_i from a point (x_p, y_p) on the contour to the point (x_s, y_s) on the SQ along the line that passes through (x_p, y_p) and the centre of the SQ:

$$d_i = \sqrt{(x_p - x_s)^2 + (y_p - y_s)^2} \tag{2.67}$$

Where:

$$x_s = \left| 1 / \left(\left| 1/a^{\frac{2}{\alpha}} \right| + \left(y_b / x_p b \right)^{\frac{2}{\alpha}} \right) \right|^{\frac{2}{\alpha}} \tag{2.68}$$

$$y_s = x_s y_b / x_p \tag{2.69}$$

To allow a rotation θ and translation of the centre of the SQ to a point (x_c, y_c), Equation 2.65 was modified to:

$$f(x,y) = \left(\frac{(x - x_c)\cos\theta - (y - y_c)\sin\theta}{a} \right)^{\frac{2}{\alpha}} + \left(\frac{(x - x_c)\sin\theta - (y - y_c)\cos\theta}{b} \right)^{\frac{2}{\alpha}} - 1 \tag{2.70}$$

Results showed that there is a high curvature bias in which the algebraic distance from a point to the SQ is underestimated [DUR08].

2.11 Extended Superellipse Modelling

The Extended SuperEllipse (ESE) is formed by expressing the exponent of the SE as a function of angular orientation [ZHO99] so that a wide range of shapes, which are not necessarily symmetric, can be represented. The implicit form of a 2D ESE is defined as:

$$\left(\frac{x}{a} \right)^{\frac{2}{f_1(\theta)}} + \left(\frac{y}{b} \right)^{\frac{2}{f_2(\theta)}} = 1 \tag{2.71}$$

Where: *a* and *b* are scale parameters that define the size of the ESE in the *x* and *y* directions, respectively, *x* and *y* are two dimensional points sampled on the ESE surface, $f_1(\theta)$ and $f_2(\theta)$ are

relative shape exponent functions in the x - y plane that vary with the angle of orientation θ to control the shape of the ESE and θ is the angular index parameter of the exponent functions.

2.11.1 Extended Superquadrics in Graphics and Image Interpretation

The Extended SuperQuadrics (ESQs) in 3D have exponents changing according to the latitude and longitude angles respectively in the object centered spherical coordinate system. ESQs were adopted in several applications for modelling natural objects. Zhou and Kambhamettu [ZH099] sought to create a deformable model that is better able to model the variability of data with a small number of parameters. They demonstrated that the ESQ is a powerful method for describing quasi-algebraic surfaces in 3D using range data. They presented experiments on both the ESQ fitting and realistic modelling which showed that the ESQ generated good 3D representations in computer graphics and that the ESQ is a promising approach for object recovery and interpretation of range data in computer vision. They used Bezier curve functions for the exponent of an SQ to create the ESQ that was able to model complex non-symmetric shapes determined by adjusting control points that are inputs to the ESQ. They observed that it is not necessary for the control points to be evenly distributed, that the number of control points required depends on the detail to be represented. The ESQ representation is both compact and rich. Zhou and Kambhamettu demonstrated the generation of non-symmetric objects such as a spoon and a duck from 3D data. However, this process was computationally expensive. In [ZHO00] they utilized the ESQ in visualization and motion analysis. They modelled human faces with a hierarchy of geometric ESQ models which integrated both local patch analysis and global shape descriptions to recover the structure and non-rigid motion. The non-rigid object in a 2D monocular image was

segmented into many small areas and local analysis performed to recover the detail for each small area. A recursive algorithm and a global shape model were used to guide the local analysis. The results indicated that this method was effective in capturing both the global and local deformations of flexible objects. Later, Zhou and Kambhamettu [ZHO02] related all the thirty six geon models with an ESQ surface model. Thirty six geons were represented and recognised using the ESQ. They performed a set of experiments on both hand-carved and simulated geon models. The results indicated that the parameters of the ESQ contain enough information to identify each geon model. Thirteen features were derived from the ESQ parameters to distinguish each geon class. The fitting process was tested on 3D simulated and natural geons.

The Nearest Neighbour (NN) and Back-Propagation Neural Network (BPNN) classifiers were employed for objects' classification. The classification error rates for objects with uneven surfaces in hand-carved models created from soap and a radish were 25% using NN classifier and 8.3% using BPNN classifier. An error measure of 8.3% between the interpreted surface and the ground truth of the object shape is relatively small demonstrating that an accurate detailed representation was achieved. However, no attempt was made to realize a smooth and even surface for the natural geon models. Top and side views of natural geon models were not recognized by SQs due to their asymmetric cross-sections. The classification error rates for noise-free simulated and natural geon models were 2.8% and 8.3%, respectively when using a BPNN. The classification error rates when adding zero-mean additive Gaussian noise with a standard deviation of 1.0 and of 2.0 to 3D data of the simulated geon models were respectively 6.9% and 16.7% using the NN classifier and 4.2% and 7.8% using the BPNN classifier showing a good level of accuracy. The high error rates for the simulated data with high noise might result from the ESQ fitting to the

noise in the surface data points. Using the ESQ fitted parameters as input to the BPNN, the output class shape may be incorrect, thus increasing the classification error rate.

The notion is that the Extended SuperEllipse (ESE) is a parametric representation of a curve in the same way that the Hough Transform (HT) is a parametric model of a shape such as a line, circle or ellipse [BAL81] and which can be generalized to any parametric shape. The ESE does not lead to the plotting of parameters to create a model. In the Hough transform the parametric form leads to a representation in a different space that allows the parameters of the line, circle or ellipse to be determined [BAL81]. Therefore it is acceptable to suppose that the ESE could be used as a flexible model to represent objects. As 2D spline curves were employed to construct a flexible model describing the outline of a person [BAU95] it is therefore reasonable to suppose that the ESE parametric curve could be structured as a flexible model to represent objects in an image with curves. The ESE representation can define curves and geometric shapes and, further, has the potential to form a suitable model.

2.12 Chapter Summary

From this literature review we draw the following requirements:

1) A new cue detection approach that will allow figural axes to be computed for a low computational cost as a cue detector.
2) A method of object identification that is adaptable to a wide range of objects and which can deal with changes in pose and variations of form.
3) A model that is able to identify many different objects in a scene and to reason about which model should be used to identify each object.

To realise these key ideas is to develop an improved model-based method for image interpretation that can identify a greater range of objects. The development of a parametric representation for the ESE combined with a statistical model will be described in the following chapters. This approach seeks to build on the strengths of the ESE and statistical modelling of form and appearance using points sampled at regular intervals. The first process is to identify cues to determine where it would be appropriate to try to fit a model in an image**.** This process is described in the next chapter. A variant method of cue detection to identify the axes of local symmetry for articulated objects is described after the following chapter and in the following chapters the ESE will be established to create a model and perform image interpretation using that model.

Chapter 3 CUE AND EDGE DETECTION

3.1 Introduction

In model-based interpretation it is necessary to determine where it is appropriate to try to fit a model at the appropriate place in the image. The model is fit to points that represent the potential boundary of the object to verify that the object represented by the model is present. One approach is to search from cue points to locate key features such as edges that sample the boundary of an object. However, edges are often not good model key points, because they do not provide good localisation; they do not uniquely identify a point as a corner does. There are also many edges in an image that will not be part of the object sought; edges are less specific than corners.

Here the cue detection method uses the Maximum Likelihood Ratio (MLR) criteria [ZHO97] to identify object cues based on regional symmetry. The MLR can also be formulated to identify edge points on the potential boundary of the objects. The edge points are sought on a systematically selected set of search paths around the cue point or axis. The MLR is used to detect edge points using a pair of regions that straddle the edge sought. The direction of search for edges is chosen so that the search path is close to perpendicular to the boundary. Therefore a pair of regions considered for edge detection may be extended along the search path.

The RGB colour space is used in the MLR for cue detection. The chief justification for the use of RGB is speed of computation to avoid the need to perform a colour space conversion. In addition cue detection seeks to locate major changes and they are apparent in all colour spaces. A perceptually uniform colour space is not required because major differences are sought; the sacrifice of

using RGB not being a perceptually uniform colour space is small, especially given the high level of redundancy in colour.

3.2 Maximum Likelihood Ratio in Grey-Level Images

The maximum likelihood ratio was reviewed in subsection 2.5.2. Edge detection in grey-level images can be performed by identifying regions where the population of grey-level values differ. The unbounded upper range of LR value makes it difficult to determine when a significant change arises. This issue can be resolved if the inverse of LRs is employed because the inverse of the LR is self-normalising over the range [0, 1]. The MLR in grey-level was given by Zhou and Pycock [ZHO97] as defined in Equation 2.47. Fig. 3.1 shows the MLR operator windows *A*, *B* and *AB*, the combined window, where the number of pixels in regions *A* and *B* are denoted by m_A and m_B, respectively.

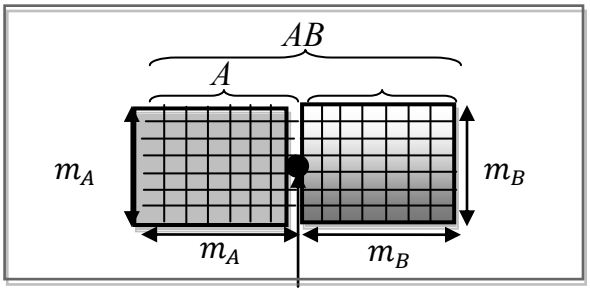


Fig. 3.1. MLR regions, *A*, *B* and the whole region, *AB*.

If the sub-populations of regions *A* and *B* are Gaussian distributed and from the same distribution then the combined population *AB* will also be Gaussian distributed. However, if the populations of regions *A* and *B* are Gaussian distributed but from different populations, then the distributions of the region *AB* will not be Gaussian distributed. That is, the likelihood that a particular value of

x drawn from the total population of *AB* will be inaccurate, but still a systematic characterisation of the population reduces as the differentiation of the populations increases. Furthermore, the likelihood estimate for the joint population will be reduced for values drawn from either population *A* or population *B*, decreasing the MLR value and emphasising the difference. Thus any deviation of population *A* or *B* and *AB* from Gaussian distributions is unlikely to have a marked impact on the computed MLR value. Deviation from Gaussian distributions will increase the likelihood of falsely detecting a difference and is therefore any error is in the direction of a preferred or safe outcome. Therefore, any deviation from Gaussian in the population being analysed has a small impact because it is the change in likelihood ratio that is important, not the absolute value.

The full expression for multi-channel data was given in Equations 2.38, 2.42 and 2.45. A simple way to extend the MLR in Equation 2.46 to colour is to assume that each colour channel of an RGB image is independent. This would give an expression in which the MLR responses for each colour channel are multiplied together. The colour MLR in this case is defined as:

$$\mathrm{MLR} = \frac{|\sum \mathrm{AB_R}|^2 + |\sum \mathrm{AB_G}|^2 + |\sum \mathrm{AB_B}|^2}{\sqrt{|\sum \mathrm{A_R}|^2 + |\sum \mathrm{A_G}|^2 + |\sum \mathrm{A_B}|^2}\sqrt{|\sum \mathrm{B_R}|^2 + |\sum \mathrm{B_G}|^2 + |\sum \mathrm{B_B}|^2}} \tag{3.1}$$

Where: $\sum A_R$, $\sum A_G$,$\sum A_B$, $\sum B_R$, $\sum B_G$, $\sum B_B$, $\sum AB_R$, $\sum AB_G$ and $\sum AB_B$ are the computed covariance of each population in windows *A*, *B* and *AB*, for the *R*, *G* and *B* channels respectively.

A simplification of Equation 3.1 results if it is assumed that the colour channels of an image in RGB colour space are highly correlated [TSA05]. The MLR is then given by:

$$\text{MLR} = \left(\sigma^2_{AB_R} + \sigma^2_{AB_G} + \sigma^2_{AB_B}\right) \Big/ \left(\sqrt{\sigma^2_{A_R} + \sigma^2_{A_G} + \sigma^2_{A_B}}\sqrt{\sigma^2_{B_R} + \sigma^2_{B_G} + \sigma^2_{B_B}}\right) \quad (3.2)$$

Where: $\sigma^2_{A_R}$, $\sigma^2_{A_G}$, $\sigma^2_{A_B}$, $\sigma^2_{B_R}$, $\sigma^2_{B_G}$, $\sigma^2_{B_B}$, $\sigma^2_{AB_R}$, $\sigma^2_{AB_G}$and $\sigma^2_{AB_B}$ represent the variance of each population in windows *A*, *B* and *AB*, for the *R*, *G* and *B* channels respectively.

3.3 MLR Cue Detection

Cue detection is based on the colour version of the MLR criteria. The goal is to identify a single cue for each object.

3.3.1 Design of Cue Detector Mask

The geometry of the MLR cue detector, shown in Fig. 3.2, was designed to accommodate the symmetry in the appearances of a variety of objects. The detector looks for pedestrians of different sizes within the image. Recall that distance to the pedestrian determines the size of the pedestrian in the image: pedestrians that are farther away appear smaller in the image; while closer pedestrians appear bigger. The detection system searches the image at set of scales and because the pedestrians are bound to the ground, this can be used to limit the search range. To detect a pedestrian it must accommodate a pedestrian in various poses, and be sufficiently specific to avoid identifying other objects. The mask is designed to detect image regions with a central patch of pixel values that differ from two regions alongside that have a similar distribution of pixel values. The vertical extent of the central region is substantially greater than that of the flanking regions and responses in the vertical axis are accumulated to allow the mask to detect pedestrians of varying heights. The width of the central region has been selected to be narrower than the average target region to accommodate pedestrians of different widths. The size of the central region and

the two flanking regions is a compromise. Small regions provide a localised estimate of response but are more prone to the limitations of a small population size and variability of the likelihood estimates. Larger regions can lead to small structures being missed and take longer to compute. The computation is simplified if the number of pixels in the central region is the same as the number of pixels in the combined flanking regions. A mask with the flanking regions spaced apart from the central region, as shown Fig. 3.2 (a), offers the potential of accommodating more variation in the form of the objects detected. The alternative mask as shown in Fig. 3.2 (b) is also considered.

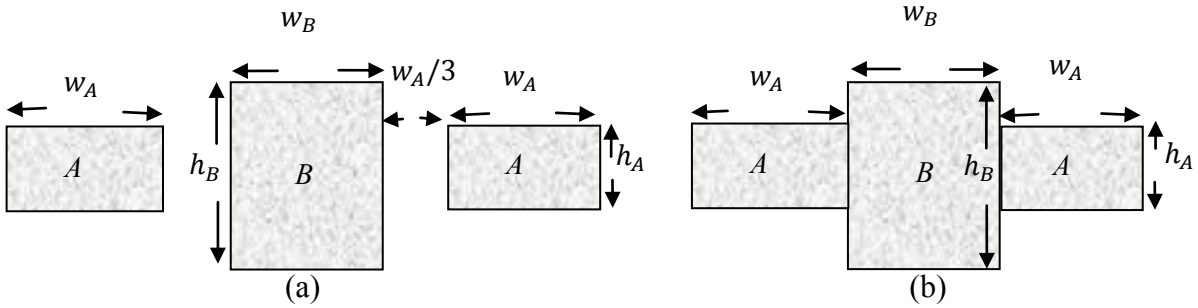


Fig. 3.2. Alternative mask patterns (a) with a gap and (b) no gap between the central and flanking regions.

In Fig. 3.2 (a) and (b), w_A, w_B, h_A and h_B denote the width and height, respectively, of regions *A* and *B*. The heights h_A and h_B are related such that $h_A = h_B/2$. The gap between windows *A* and *B* for mask 1 is $w_A/3$. The masks presume that the pedestrian is upright.

3.3.2 Search Strategy

A search for object cues is performed at four scales to accommodate changes in the size of objects with distance and natural variation in their size. The mask pattern is moved along a selected sparse set of columns in the image. The sparse sampling of columns is sufficiently dense to detect pedestrians such that the separation between columns was selected to be *1/3 rd* of the width of the

central region *B* of the mask, the smallest width of a pedestrian. This distance between sampling columns was selected so that at least one column will be located close to the centre of any pedestrian in the image. The step size along columns (between the rows) is the same as the height of region *A*. As the mask is moved along a vertical path of the image in turn, MLR response is computed at each position and associated with the centre point of region *B* (see Fig. 3.2). When moving to a new column position the operation required to re-compute the MLR response is the same for a large or a single pixel step. The flanking regions do not overlap and only half the pixels overlap from the central region during the movement along columns. With movement along columns it would take as long to update the statistics for the central region as to re-compute for the whole region. The operator is applied across scale and the responses at each scale were combined at the position of the response in the finest scale image. The responses were combined across scale after the hysteresis threshold is applied. This combination of responses helps to identify the target object from other background objects. At each change of scale the operator size is reduced by a factor of 2. The cue detection algorithm is summarised in Fig. 3.3.

w_A: The width of region *A*, h_A: The height of region *A*
i: denotes the rows of the image, *j*: denotes the columns of the image

1. For each of four scales from fine to coarse
    1.1 For each $i = 0$ to end of row by w_A
        1.1.1 For each $j = 0$ to end of column by h_A
            1.1.1.1 Compute MLR using Equation 3.2
        1.1.2 Apply Hysteresis Threshold
        End
    1.2 Combine all detected responses at each scale
    End
2. Select a cue that best identifies a pedestrian.

Fig. 3.3. Cue detection summary.

The search path of the cue detection algorithm is shown in Fig. 3.4 (a) and a mask of central and flanking regions is shown in Fig. 3.4 (b).

Fig. 3.4 (a) shows the path and the positions of the mask at an image column. The widths of the central region and the flanking regions are w_B and w_A, respectively and the heights of the central region and the flanking regions are h_A. The blue flanking regions on the left and the right are spaced from the grey central region by a gap of width $w_A/3$. The stack of blue and grey coloured regions shows the mask and the movement of the mask down a column with vertical distance equal to the height of the flanking regions. This scanning process is shown by a black arrow where the large gray dot shows the centre of a central region to which a response obtained is associated. The distance between columns was selected to be $w_B/3$ so that at least one column will be located near to the centre of a pedestrian in the image.

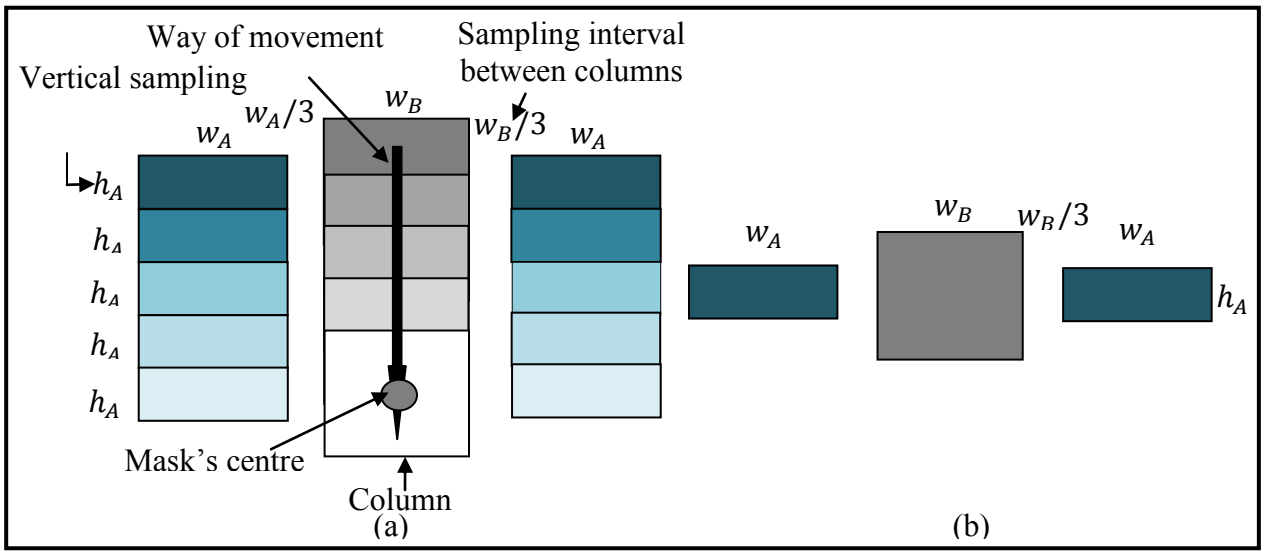


Fig. 3.4. Cue detection search path.

3.3.3 Hysteresis Threshold

Thresholding with hysteresis is applied to the generated likelihood responses to select bands of peak MLR responses for each column before combining the responses across scale. Detection is initiated when the high threshold is exceeded and continues until the response falls below the low

threshold. The response at the middle of this range is reported as the response of the operator. Hysteresis thresholding tracks along the detected responses to remove weak responses and maintain connected strong responses.

3.3.4 Clustering of Responses and Selection of Cues

Given the MLR responses detected by the cue detector; a clustering method of three successive phases was applied to identify the positions of potential pedestrians. The generated MLR responses were scanned to identify a local peak of MLR response. A distance of w_A pixels of one another of MLR responses were identified into a group. The distance w_A is the same as the width of the central region of the mask. That this distance is chosen to associate points across the width of a pedestrian together without incorporating to many points that do not correspond to a pedestrian. The validity of this assumption is evaluated in subsections 7.3.2 and 7.3.3.

In the scanning and clustering procedures the class of each set of points can be described as: 1) clustered points at the centre of the pedestrian, 2) clustered points not centred on the pedestrian and 3) single points. These classes and clusters of MLR responses along with the cluster points are illustrated, by a set of experimental examples, in Section 7.3.

A single point closest to the centre of each cluster of responses is computed. This was performed using a weighted squared mean distance, defined as:

$$\varepsilon_k = \sum_{i=1}^{m_k} \sum_{k=1}^{K} z_{ki} \|p_i - \bar{q}_k\|^2 \quad , p_i \in C_k \tag{3.3}$$

Where:

m_k: is the total number of responses in the k^{th} cluster.

K: is the number of clusters.

C_k: is a cluster k.

z_{ki}: is a weight that defines the appropriateness that the i^{th} response, p_i, is close to the centre of the k^{th} cluster. The value of z_{ki} was defined to be *1* or *0*.

p_i: is a response in cluster k at index i, defined by its spatial position, $p_i = (x_i, y_i)$.

$\bar{q}_k$: is the centre of cluster k, C_k, defined by $\bar{q}_k = (\bar{x}_k, \bar{y}_k)$.

$\|p_i - \bar{q}_k\|$: is the Euclidean distance between p_i and $\bar{q}_k$.

The cluster centre identified by the mean position of the points in cluster C_k was computed as:

$$\bar{q}_k = \frac{1}{m_k} \sum_{i=1}^{m_k} p_i \tag{3.4}$$

A weight to define a point and the distance from that point to each cue point were used to determine which points are included in the cluster. This process was recalculated until each cluster reduced to a single point. The process is stopped when each cluster reduced to a single point or a threshold defined by the distance between the centres of the clusters is less than w_A.

All clusters with the cue points were scanned using a similarity measure [SHA95] [WAN02] to determine if the candidate cue points belonging to the same object or not. This measure was computed between each pair of clusters of cues. The cue points for the same object were combined. A measure of similarity using the MLR response values for clusters of cues was defined between two clusters, as defined by [SHA95] [WAN02]:

$$SIM = \frac{\sum_i (r_i - \bar{r}) \sum_j (s_j - \bar{s})}{\sqrt{\sum_i (r_i - \bar{r})^2 \sum_j (s_j - \bar{s})^2}} \tag{3.5}$$

Where: *SIM* defines a measure of similarity between a pair of clusters *R* and *S*, $\bar{r}$ and $\bar{s}$ are the mean of the response values in cluster sets *R* and *S*, respectively, r_i and s_j are the i^{th} and j^{th} MLR response values in clusters *R* and *S*, respectively where *i* and *j* represent the indexes of the responses in clusters *R* and *S*, respectively.

All combinations of the points of the clusters are considered to measure the similarity between the clusters. This formula of similarity does not consider corresponding points between the clusters but all combination of the responses. Equation 3.5 measures the product of the sum of the differences between the value of each MLR response and the mean value for a pair of clusters divided by the square root of the sum of the squared difference between the values of MLR responses and the value of the mean for a pair of clusters. Thus, this formula measures the correlation between clusters *R* and *S* with respect to all response values [WAN02]. The correlation measure defines the coherence between the clusters of cues as evaluated by Wang et al. [WAN02] and Shardanand and Maes [SHA95]. A large positive value signifies a strong correlation and a large negative value signifies a lack of correlation. Two clusters belong to an object if they exhibit a large coherent value. A threshold was applied to the similarity value to determine if the responses in a cluster belong to a single object so that it is appropriate to combine the cues of those clusters. An appropriate threshold was experimentally determined have a value of 3. The choice of value was shown in Section 7.3 not to be critical. The experimental settings and performance of this clustering method along with the MLR profile responses on a set of experiments

are presented in Section 7.3. The response of a rubbish bin or a tree to the pedestrian detector is illustrated in subsection 7.3.3.

A Support Vector Machine (SVM) classifier, as an alternative method to the similarity and selection of cues criteria, was combined with the clustering procedure to identify pedestrians. Here the feature set produced by the MLR criteria and a new feature set of co-occurrence matrices were combined with the SVM classifier to detect pedestrians. The cue detector algorithm based on MLR was trained to identify the interest points and classify the patches centred at those points to identify the potential regions of interest using the clustering method. The SVM [SCH09] classifier introduced in subsection 2.3.1.1 was trained on those potential regions of responses for pedestrians and non-pedestrians. The SVM classifier used the set of features of MLR and a set of features extracted using the co-occurrence matrices as described in [HAR73] and [SCH09]. For each cluster and detection window in the image, texture features extracted using co-occurrence matrices were concatenated with the MLR feature set and classified by the SVM classifier as a pedestrian or non- pedestrian. A set of five descriptors was used: the entropy, angular second-moment, sum of squares, contrast and inverse difference moment of the co-occurrence matrix [HAR73] SCH09]. Co-occurrence features were functional in pedestrian detection since they provide information regarding homogeneity and directionality of patches. Block sizes of $w_A/2\times w_A/2$ with shifts of $w_A/3$ pixels were used for feature set computed by co-occurrence matrices. The RGB colour space was used and each colour channel was quantized into 16 bins. The features of the co-occurrence matrices and their descriptions and mathematical calculations were introduced in the literature review in subsection 2.3.1.2. The SVM used here is implemented in

OpenCV based on the library LibSVM [CHA11]. Details of the configuration of the SVM are introduced in the experimental subsection 7.3.3. The clustering procedure of the proposed cue detector follows the sequence of steps summarized in Fig.3.5.

1. For each MLR response
   1.1 Cluster all responses within a distance of w_A pixels.
2. For each cluster of responses
   2.1 Identify a single candidate cue point closest to the centre of each cluster using the formulas defined in Equations 3.3 and 3.4.
   2.2 Repeat with a new weight until each cluster is reduced to a single point.
3. Scan through the identified clusters of the candidate cue points
   3.1 Combine cues that share a degree of similarity between their clusters of responses using the measure of similarity criterion defined in Equation 3.5.
4. Use the MLR feature set and a set of measured features using the co-occurrence matrices to train the SVM to the responses of pedestrians and non-pedestrians.
5. Repeat steps 1 to 3 and use the feature sets of MLR and co-occurrence matrices to evaluate the SVM and the clustering procedure of the cue detector method to identify the pedestrians.

Fig. 3.5. A summary of clustering and selection of cues procedure.

3.3.5 Computational Complexity

The MLR values were computed using the sum of values and sum of squares for computational efficiency in a single pass of the image data at each position of the mask. Processing the cluster responses requires each cue to be visited and the points are re-examined for each cluster. Therefore, the method has a complexity of $O(nmp^2)$ where n is the number of points in the image at which the mask is applied, m is the number of points in the detector mask and p is the number of candidate cue points. The width of the mask is about twice the width of the pedestrian and the height of the mask is about the same height of the pedestrian. However, the width of the image is several times that of the pedestrian and the height of the image several times that of the pedestrian. Therefore, the number of points in the mask is less than the number of points in the image

but the number of points in the image that lay under the mask is the same as the number of points in the image. The order of the complexity issue of MLR computation is similar to the order of image size. The factor *m* is similar in magnitude to *n* and *p* does form an important factor of the complexity assessment. Then, the complexity can be reduced to $O(n^2p^2)$.

3.4 Variations of MLR Cue Detector

A variant to the pedestrian cue detection algorithm for detecting vehicles is described here. The hysteresis threshold and the clustering methods were not changed. The geometry of the mask to detect pedestrians, shown in Fig. 3.2, was modified to detect vehicles by changing the width and height of the regions of the mask. This is a minor adaptation. The search strategy to detect vehicles scanned first along horizontal paths. The values used for the width, w_B, height, h_B and the gap between the two regions for vehicle detection, at the first scale, are shown in Table 3.1.

Vehicles do not respond to a pedestrian detector because the dimensions of vehicles are very different to those for pedestrian detection. Therefore, the dimensions of the vehicle detector mask are very different to those of the pedestrian detection mask and, therefore, it is unlikely that a pedestrian would be detected using the vehicle mask. To detect both pedestrians and vehicles in one image each detector must be applied in turn. The responses of vehicles to the pedestrian detector and of pedestrians to the vehicle detector are illustrated in subsection 7.3.5.

Table 3.1. Parameter values used for the masks of the pedestrian and vehicle detectors.

Detector	Mask dimension (pixel)		
	w_B	h_B	gap
Pedestrian	48	72	16
Vehicle	108	54	-

The adopted cues are points at which it is appropriate to try to fit a model. That model fitting process, as described here requires "key points" that identify potential object boundary points.

3.5 Key Point Generation

Sampled points on the boundary of the object, key points, are required to build a model and to interpret an image. A set of systematic search paths are used to generate the key points (edge responses) sampled at regular intervals around an object. They are detected using a variant of the MLR operator with a pair of regions *A* and *B,* as shown in Fig. 3.1, designed to identify edge discontinuities as they are moved along each search path.

3.5.1 Search Paths

To identify the key points we first search on radial paths from a cue point or perpendicular to an axis. This form of search strategy is used to increase the likelihood that the search path crosses the boundary at angle close to 90 degrees and thereby improve the edge response. When a radial path is close to parallel with the boundary then a secondary path, perpendicular to the initial search path is adopted. For vehicles it is only necessary to search along paths that are radial to the cue points. The initial search path is radial from the object location cue, as shown in Fig. 3.6 (a) - (b). If the initial search path is not close to perpendicular to the boundary a series of secondary search paths are constructed perpendicular to the chord between adjacent radial paths, as shown in Fig. 3.6 (c) – (d). The key point detection process is described in subsection 3.5.2, and Fig. 3.7.

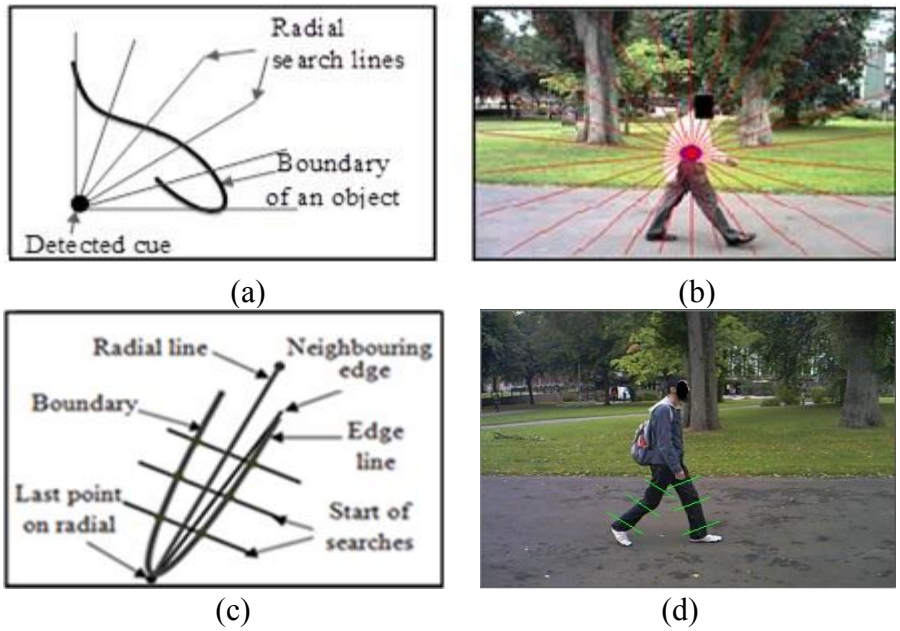


Fig. 3.6. Key point search paths: (a) Initial radial search path (b) initial radial search paths on an image of a pedestrian, (c) Secondary search path perpendicular to initial radial search path and (d) secondary search paths on an image of a pedestrian.

1. For angular increments in orientation of the radial search path from $\alpha = 0$ to 2π
   - 1.1 Generate a radial line.
   - 1.2 From start to end of radial line
     - 1.2.1 Compute MLR edge response
     - 1.2.2 Select *N* strongest responses as potential key points.
2. For each radial line *i* = 0 to *last but one*
   - 2.1 For key point *j* =*1* to *N*
     - 2.1.1 Compute distance, d_i, from start of radial line to key point *j*
       - 2.1.1.1 If $\left((d_i/d_{i+1}) < 0.5\ OR\ (d_i/d_{i+1}) > 2.0\right)$ Then
         For line from key point *N* on radial line *i* to key point *N* on radial line *i+1*.
         - 2.1.1.1.1 Search perpendicular to line for key points.
         - 2.1.1.1.2 Add point with largest edge point to selected point set.

Fig. 3.7. Key point detection algorithm.

3.5.2 Edge Detection

The windows *A* and *B* of the MLR edge detector of Fig. 3.1 were each set to a size of 7×7 pixels. The generation of edge points on both radial and perpendicular search paths to form a set of candidate boundary edge points is described in Fig. 3.7. The need for a secondary search path was

identified from the ratio of the distance of the key points from start of radial line to the key points on successive radial search paths. The radial distance to the corresponding first and third points on successive radial lines was compared to establish a secondary search path. Pairs of points were taken in sequence on each search path such as the i^{th} point on one radial line was selected to correspond to the i^{th} point on the next radial line. The criterion for determining when a secondary search path was required was determined empirically.

A set of values in the range of 0.1-1.0 and 1.7-3.5 pixels were applied to the distances between the key points on consecutive radial lines. How the ratio of the distance between the key points on successive radial search paths was calculated to determine the need for a secondary search path is described in subsection 7.4.6. The strategy can be observed in Fig. 3.8 with two different set of values used to determine when a secondary search path should be used.

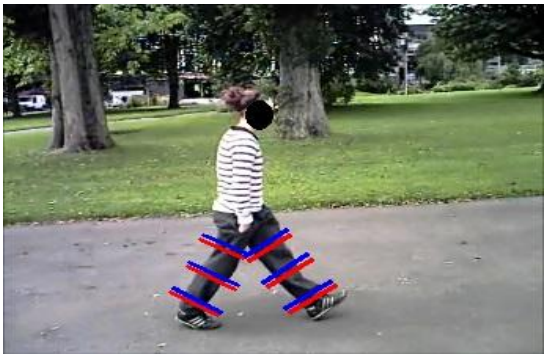


Fig. 3.8. Secondary search paths using two sets of distances.

The blue search paths in Fig. 3.8 were identified using the distances of 0.9 and 2.5 while the red search paths were identified using the distances of 0.5 and 2.0. These distances evaluated to correct results in key point selection algorithm.

The edge responses detected on the radial and secondary search paths were analysed using the peak detector, described below, to locate the key points required to form a model.

3.5.2.1 Peak Detector

Peak detection with hysteresis is applied to the MLR edge responses to detect the position of the key points required to create a model and perform an interpretation. This is to ensure that only significant peaks and valleys are detected and to avoid the use of smoothing which can distort edge position. The parameters width, *w,* and height, *h,* of the candidate peaks (valleys) define the peak (valley) response, as shown in Fig. 3.9. To be a peak (valley) the value at point, *X*, must be greater (less) than the value at positions *Y* and *Z* by at least *h*. Peak detection with hysteresis is fast and avoids local minima, which would arise at weak boundaries between regions.

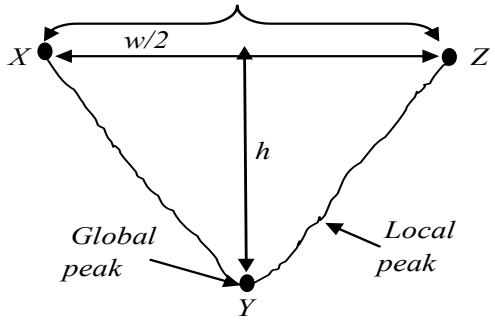


Fig. 3.9. Peak and valley detection: peak width (*w*) and peak height (*h*).

Peak width and peak height for peak and valley detection were determined as described in Section 7.4.6. This is illustrated in Fig. 3.10. Only the five most significant edge responses identified by light blue dots were selected along each search path. This was to constrain the search space for the model whilst ensuring that the correct edge position was preserved for model interpretation.

3.5.2.2 Key Point Selection

One boundary key point was selected along each search path. The key points on the boundary of an object were selected so that a well-defined model can be created. A set number of potential key points per search path should suffice to represent a shape. Key points are important for fitting an ESE curve to represent a variety of poses with one model. The key point selection process is illustrated in Fig. 3.10 with selected frames from a sequence of frames showing a pedestrian in slightly different poses. In each frame 43 key points were identified to form the boundary. The contour and the radial lines are shown in red. The perpendicular lines are shown in green and the key points used to form a model are shown as blue dots and the other detected key points are shown by light blue dots, which in some cases hide the selected key points.

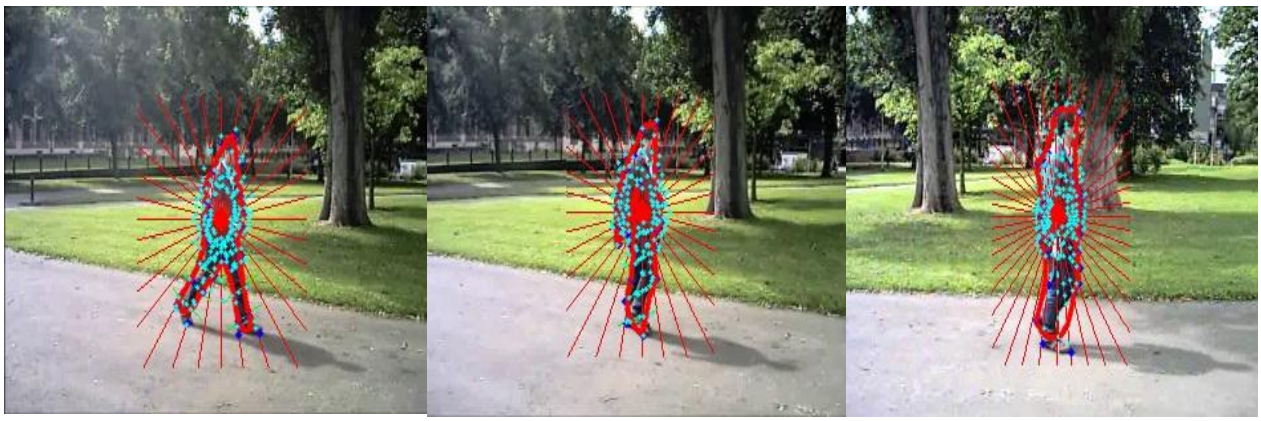


Fig. 3.10. A human in different poses with radial, red, and perpendicular, green, search lines: three contour examples with key points identified with blue dots.

Initially key points were manually selected from the identified edge responses along each search path to collect the data and form a model. The radial search paths are spaced at equal angles. The spacing of key points varies when a secondary search path is used. The perpendicular, secondary search paths are introduced as necessary when a radial path is close to parallel with the boundary

as explained in subsections 3.5.1 and 3.5.2. It is observed from Fig. 3.10 that the points sampled are not all at equal angles given the nature of the sampling interval with the secondary search path; they are mainly equally spaced and in places approximately equally spaced. The total number of radial and perpendicular search paths is the same regardless of the pose to simplify model generation and matching, as illustrated in Fig. 3.10.

Many edge detection methods were reviewed in Section 2.5. These edge operators are not appropriate to find the edge points along the search paths described in subsection 3.5.1 because these edge detection methods are sensitive to noisy images, since both the noise and the edges contain high frequency content and there are problems of false edge detection and missing true edges in these previously reported edge detection methods. Also, the edges produced by these methods do not identify well defined edge points. Here the MLR has a relatively high computational burden and the positions of edge points were used to create a model rather than to define the boundary of an object as in [YEN03]. Here, a peak hysteresis algorithm was employed to find the most significant edge points "peaks" on the search paths that can sample the boundary of an object.

3.6 Summary

A generic cue detector based on MLR computation that can be used to detect cues for a wide range of objects has been described. How this cue could be used as a basis for detecting edge responses, again using an MLR edge detection operator was also described. In addition a process for selecting the boundary key points has been described.

Chapter 4 COMPONENT AXES DETECTION

4.1 Introduction

This chapter introduces the development of an axis detector at various orientations to identify axes as cues for complex and articulated objects. These axes are used to guide the search for object boundaries, the creation of a model and image interpretation using that model. Any axis detector must be able to identify axes in a variety of contexts to provide effective cues for model creation and image interpretation. It is difficult to define an axis detector that is effective and robust in all situations.

Computing two orthogonal Maximum Likelihood Ratio (MLR) operations allows the response for a filter at a variety of orientations to be synthesised. The initial cues are generated and augmented with features to identify the objects for which they are most likely to be a cue. The axial cues are linked to form a single composite axial cue. Search paths analogous to the radial paths in subsection 3.5.1 are constructed perpendicular to the axes. The edge detection procedure described earlier in subsection 3.5.2 is used to search for the points on the potential boundary of the object. Edge points around the object boundary are selected as potential key points to form a model.

4.2 Steerable Filter Transform

The steerable filter was introduced by Freeman and Adelson [FRE91] to provide a linear analysis across scale and resolution. The filter is steered by rotation and translated. The filter was formulated for a continuous range of orientations and scales, and can be synthesized with a linear com-

bination of the result of applying a set of basis filters at fixed orientations. The basis filters can be applied to an image and the responses interpolated. A traditional steerable and scalable filter is performed in three phases:

1. Decomposition: to produce a set of steerable and scalable kernels.
2. Convolution: to produce a set of basis responses of an input image with the kernels.
3. Reconstruction: to combine the basis responses with appropriate weights to produce the responses for a particular orientation band and scale.

The basis filters, at each orientation, provide discrete responses that are interpolated. The result is a filter *f(x, y)* at orientation θ defined as [FRE91]:

$$f^{\theta}(x,y) = \sum_{q=1}^{Q} k_q(\theta) f^{\theta_q}(x,y) \tag{4.1}$$

Where: $f^{\theta}(x,y)$ is the effective filter at angle θ, $f^{\theta_q}(x,y)$ is the basis filter at q and θ_q, $k_q(\theta)$ are the interpolation functions for the q^{th} basis filter and Q is the number of basis functions required to steer $f^{\theta}(x,y)$.

4.3 Axis Point Detection

4.3.1 Interpolated Function of the MLR

The MLR steerable (interpolated) function, F_{mlr_n}, for an input image *(I)* of an arbitrary orientation, θ, and scale, *n*, is given by:

$$F^{\theta}_{mlr_n}(I) = \sum_{q=1}^{2} k_q(\theta)\, F^{\theta_q}_{mlr_n}(I) \tag{4.2}$$

Where: $F^{\theta}_{mlr_n}$ is the effective filter at angle, θ, and scale n, $k_q(\theta)$ is the interpolation weight for the q^{th} basis filter, $F^{\theta_q}_{mlr_n}$ is the basis function at n and q and 2 is the number of basis functions.

With the colour MLR of Equation 3.2 as the basis function in Equation 4.2, the axes at a range of orientations can be identified using:

$$F^{\theta_2}_{mlr_n}(I) = cos(\theta)F^{0}_{mlr_n}(I) + sin(\theta)F^{90}_{mlr_n}(I) \tag{4.3}$$

Where: $cos(\theta)$ and $sin(\theta)$ are the weights of the function.

It is not appropriate to linearly synthesize the MLR responses because the MLR operation is not linear. Therefore, the standard deviation of two adjacent 7 x 7 regions was considered. Also, the standard deviation of the joint 14 x 7 region is not the linear combination of the standard deviation of the two regions. Therefore, the standard deviation of each region is squared, added together and then the square root taken. It is not possible to synthesise the output of the MLR filter at various orientations but to approximate the output response of the filter at various orientations by interpolation. This process encompasses a class of filters at a set of predefined orientation with interpolation to synthesise response at other orientations. This interpolates the filter using many versions of the same filter, each different from the others by an orientational angle. The output responses were identified at any orientation with the correct filter set, correct interpolation rule and the number of versions of the filter equal to the number of angles.

A set of experiments, to approximate the output responses of the MLR operator, was applied, by interpolation, at various orientations to identify axis points for pedestrians alone or pedestrians

associated with a pushchair or a bicycle. These axis points were defined by the symmetry of the objects and their component parts in order to determine the major component axes.

To show the validity of interpolation and to assess the appropriateness of interpolation to identify the axes points, a set of experiments are presented in subsection 7.4.4 to see the difference between the result of interpolation and the result of the actual MLR operator at various orientations. Further experiments of interpolating the outputs of MLR filter oriented in the correct direction, are illustrated in the results subsection 7.4.4.

Illustrative results of interpolation to identify the axes points are shown in Figs. 4.5 and 4.7. Fig. 4.5 shows the detected axis point locations for two pedestrians in two difference images with red crosses mark axis points. Fig. 4.7 shows the axis point locations for a pedestrian pushing a push-chair. The red bullets mark axis points. A set of subsequent procedures were used after the interpolation process to refine the output responses to identify the component axes.

4.3.2 Detection of Initial Local Axes Points

The interpolated MLR operator was scanned across the image, a hysteresis threshold and clustering procedures similar to those described for the non-interpolated filter (see subsections 3.3.2 - 3.3.4). The difference is that here the MLR response is interpolated at a variety of orientation angles and a range of scales to generate local axis points on an object by superposition. The colour MLR values were computed using the square root of the squared variance of the combined region divided by the squared variances of the flanking regions. The local axes points, defined by local patterns of symmetry, were refined on the components of an object as described in the following

sections. These axes points were linked to identify the axes for the components of an object as described in Section 4.5. These component axes show the local symmetry on the objects. These axes were based on the symmetry of the objects and their component parts. The use of these axes to define models and interpret complex objects and combination of objects in a variety of poses as in images of people with pushchairs or bicycles is also described in Section 4.5. The detected component axes form a set of starting points from which to search for the potential boundary "key points", as described in Section 4.6. The initial local axial responses were generated at a range of orientations as summarised in Fig. 4.1.

I is the input image, *n* is the scales, $n = (0.5,1,2,4)$
1. For each of four scales from fine to coarse
   1.1 For each $i = 0$ to end of column by $h_B/2$
      1.1.1 For each $j = 0$ to end of row by w_B
         1.1.1.1 Compute $F^0_{imlr_n}(I)$ and $F^{90}_{imlr_n}(I)$ as shown in Equations 3.2 and 4.2.
         1.1.1.2 For each orientation θ
            1.1.1.2.1 Compute $F^{\theta}_{imlr_n}$ using Equations 4.1 and 4.2
            1.1.1.2.2 Assign the axes points to the centre point of region *B*.
            End
         End
      End
   1.2 Apply hysteresis thresholding.
   1.3 Combine all detected responses at each scale.
   End

Fig. 4.1. Algorithm summary for finding the initial local axes responses.

4.3.3 Clustering and Selection of Cue Responses

A clustering procedure similar in some aspects to that described in subsection 3.3.4 was applied to the initial local axis points to refine the detected responses on the objects of interest and to guide the detection of these objects with the procedures described below. The responses and the initial candidate cue points detected within a distance of $5w_A/2$ of one another were combined and the centre of each cluster of responses returned as a single response. These procedures are de-

scribed in Equations 3.3 and 3.4 to find the closest points at the mean positions of the cluster of points and (3.5) to measure the similarity of responses between the cluster sets. A SVM implemented in OpenCV based on LibSVM [CHA11] was used as an alternative method to the selection of cues to identify pedestrians which vary in form as the walk, push pushchairs and push or ride bicycles. The SVM classifier used the features introduced in subsection 3.3.4 and described in detail in subsection 2.3.1.Details of the configuration of the SVM classifier are introduced in the experimental subsection 7.4.2. The SVM was trained and evaluated on the the same training and evaluation datasets as those used with the FT classifier as described in Section 4.4. This SVM was trained and evaluated on pedestrians alone, pedestrians associated with pushchairs and bicycles and non-pedestrians objects.

Every pair-wise combination of candidate local axis points was checked to identify single axes. The points were then checked by the proximity of the neighbours. In particular, the magnitude of the responses and the distance between the local axes points were used as:

$$F\left(r_i(x,y), r_j(x,y)\right) = \left(\left|d_{i,j}\right| \cdot r_i(x,y)\right) \cdot \left(\left|d_{j,i}\right| \cdot r_j(x,y)\right) \tag{4.4}$$

Given that $d_{i,j} = d_{j,i}$ represents the distance between the position of responses $r_i(x,y) = (x_i, y_i)$ and $r_j(x,y) = \left(x_j, y_j\right)$, defined as:

$$d_{i,j} = \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2}; \qquad \left(w_A \leq \left\|d_{i,j}\right\| < 2w_A\right) \tag{4.5}$$

Where: $r_i(x,y)$ and $r_j(x,y)$ are local axes responses at positions *i* and *j*, respectively.

The criterion $F\left(r_i(x,y), r_j(x,y)\right)$ in Equation 4.4 is the product of the moment of the responses r_i and r_j by which points are selected. The nature of the distance $d_{i,j}$ in Equation 4.5 between each two neighbour axes responses was restricted to be between w_A and $2w_A$, which is proportional to width of the operator used. This is to limit the influence of neighbouring response to the scale of the object mask. This is a basis by which a single point is retained for each of two neighbouring responses if $F\left(r_i(x,y), r_j(x,y)\right)$ is greater than the multiplication of w_A and $2w_A$. This process requires a small number of comparisons because the responses were initially clustered and axes points within a distance of $5w_A/2$ of one another were combined and the centre of each cluster returned before the hysteresis threshold. Finally, the axis cues that result from the clustering procedure and the strategies described above were combined to form a composite cue and to identify the objects for which they are most likely to be a cue, as described in Sections 4.4 and 4.5. These axes were used to locate the boundary points of objects as shown in Section 4.6.

4.4 Object Identification

In the images considered here the objects of interest, people, pushchairs and bicycles appear separately, together and in groups. The determination of object type from the output of the cue detector is important because it allows an appropriate interpretation model to be selected. Energy and entropy attributes were computed for the local axes points identified to identify the extent of object and the type of the object and a model for each cue. The data for the energy and entropy measures was normalised by dividing each element of the data by sum of all the elements in the data. The energy attribute is the sum of the squared normalised frequencies of occurrence, com-

puted as defined in Equation 2.4 [HAR73]. The entropy for the normalised frequencies of occurrence was computed as defined in Equation 2.17 [HAR73].

A Random Forest (RF) [BRE01] was used to identify the object type using the measures of energy and entropy applied to cue responses. The effectiveness of classification depends on the features and the relationship between the classes. The RF classifier introduced in subsection 2.7.5 is used for identifying the type of each cue. RF classifier offers high classification accuracy [WAT08], an ability to identify which feature measurements should be used [WIL08], an ability to form complex decision surfaces [BOS07], an ability to accommodate missing values and avoid over fitting decision surfaces to data [BRE01].

Four classes of objects are considered; a pedestrian alone, a pedestrian pushing a pushchair, a pedestrian pushing a bicycle and a person riding a bicycle. The training data of the FT classifier consists of the entropy and the energy values of 800 sample images of different types of objects. The evaluation data consists of the entropy and the energy measures of 600 images of different cue classes. The degree of freedom for cue identification is identified by the number of object types plus the number of feature measurements, assuming that the parameters are independent. Four classes and two feature measurements give 8 degrees of freedom. Classification criteria are derived from the confusion matrix of the RF classifier, as presented in subsection 7.4.2.1.

An alternative SVM classifier based on LibSVM [CHA11] was also used to identify the object type using the measures described above and the same training and test datasets of the FT classifier. The parameter listing of the SVM is described in subsection 7.3.4.

Here what has been done is a simple approach designed to see if it might be possible to identify objects in this way.

4.5 Axis Generation

A least square linear regression is used to form an axis from selected cues. [DAV09]. The regression line which best fits the axes points was calculated such that the sum of squares of the y-axis variables is as small as possible from each axis point to the regression line. The sum of squares of the residuals allows the residuals to be treated as a continuous differentiable quantity. The least squares regression used to fit to a set of paired axes points to a line is given by:

$$\hat{y} = a + bx \tag{4.6}$$

Where:

b: is the slope, computed by: $b = r\sigma_y/\sigma_x$

a: is the intercept

r: is the correlation coefficient

σ_x: is the standard deviation of x

σ_y: is the standard deviation of y

$\bar{x}$: is the mean value of x

$\bar{y}$: is the mean value of y

$\hat{y}$: is the predicted value and y is the true observed value

The least squares line minimises the sum of the squares of the residuals as:

$$d_r = \sum_{i=1}^{h} (y_i - a - bx_i)^2 \tag{4.7}$$

Where: h is the number of points. The x-axis represents the estimates and the y-axis is the defined axes points.

The component axes are joined to the closest axis end point to form a composite axis. Here the axes detection is considered for people, pushchairs and bicycles. Fig. 4.2 shows a pose independent way to identify cues for a person. Fig. 4.2 shows a limited set of stylised poses for a person such that this diagram identifies a set of posterior, side and frontal poses. Fig. 4.2 (a) – (e) identifies a set of stylised posterior and frontal poses and Fig. 4.2 (f) – (j) identifies a set of stylised side poses. There are a differences between Fig. 4.2 (f) – (j) due changes in the positions of the arms and legs. Variations of the appearance of the pedestrians also arise when they carry accessories that present a minor change to the form of the axes. The major component axes for a standing person are the head, torso, arms and legs [WUB06]. Sub-regions were formed around cue points to represent the various components of a person as shown in Fig. 4.2. The w and h parameters stated on all figures in Fig. 4.2 were used to identify the regions and are the same regardless of the pose. The dimensions were evaluated and confirmed such that they were not pose dependent [WUB06], and the values used were recommended by Wu and Nevatia [WUB06]. These sub-regions are independent from each other such that if the head or any component is occluded this process will not fail. The head axis point is defined as the centre point for the axis points that are in the top region of the person, in a height range of 0 to 0.3h. The axis for the torso is identified from a single vertical axis in the region of 0.48 of the height to 0.73 of the height, as shown in Fig. 4.2 (c). The leg axes are identified in the lower half of the person as shown in Fig. 4.2 (d),

specifically, 0.5 of the height from the lower half of the person. The arms as axes are identified in the top half of the person and within 30% of each edge of the person, as shown in Fig. 4.2 (e).

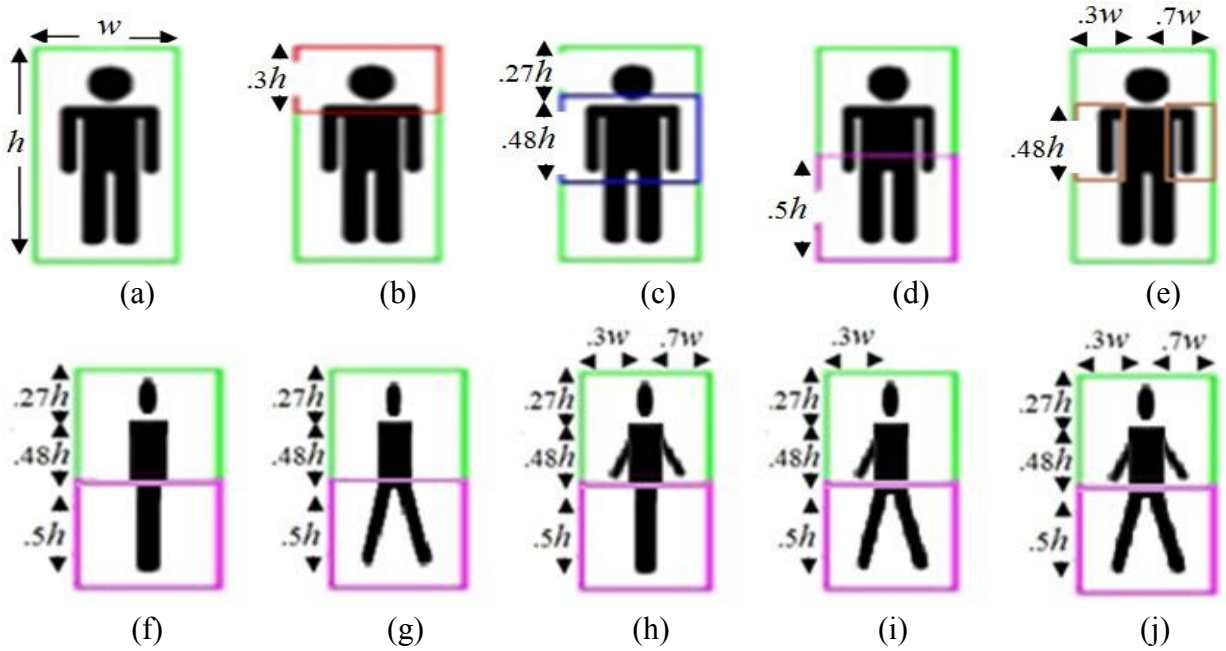


Fig. 4.2. Identification of sub-regions for a standing person in posterior, side and frontal poses: region of whole person,(b) region for posterior and frontal head, (c) region for posterior and frontal torso, (d) posterior and frontal legs, (e) posterior and frontal arms, (f) three regions of a side pose,(g) four regions of a side pose, (h) five regions of a side pose, (i) five regions of a side pose and (k) six regions of a side pose.

It is not necessary to determine the pose of the pedestrian because the axes are identified from the axis points for each sub-region. The number of major axes depends on the positions of the axes points detected. Axes can be identified with non-optimal values for *w* and *h* but the resulting axes may not reflect the symmetry of the shape concerned when the axes are axes of symmetry for parts of the object. The position of the component axes might be changed from the centre line due to that the shape they present is not symmetric. The analysis of the figures in Fig. 4.2 defines a strategy for identifying important sub-regions for a variety of pedestrian poses and a change of appearance. The results shown in subsections 7.4.3 and 7.4.4 present a wide variation of appear-

ances for generation cues and component axes. The evaluation issues for generating cues and axes for objects with a good degree of appearance and pose are shown in subsections 7.4.5-7.4.6.

The principal axes for a pushchair might differ in number and form depending on the structure and pose of the pushchair. Axes are sought as a first step towards locating the boundary of the pushchair. The axis points detected for a pushchair identify the regions that must be found to identify the component axes. A linear axis was derived by linear regression from the axis cue points [DAV09]. A consistent number of key points to satisfactorily represent a pushchair are needed regardless of the pose or the axes generated. The distance from the axis to the boundary of the pushchair varies, such that the axes might not represent the symmetry of a pushchair if the pushchair is not symmetric, it just can show the axes of symmetry for parts of the object. These axes were used to identify the points along the boundary of a pushchair as described in Section 4.6. The key points along the boundary of a pushchair were used to form a model and to interpret images containing pushchairs.

The axes for the bicycle represent the key components of a bicycle. The components of the bicycle, as shown in Fig. 4.3, are: the cross bar, the fork, the seat stay, the seat post, the down tube, the chain stay, the wheels, the saddle and the handlebar. The frame elements of the bicycle define the regions that must be found to identify each component axis. An axis will be generated for the points detected for each key component of a bicycle. The axes for the rims of the bicycle wheels are be fit to circles using RANdom SAmple Consensus (RANSAC) circle fitting [FIS81]. The axes for the bicycle wheels were first identified and the axes for the other frame elements were identified using the regions of the structure of the bicycle as shown in Fig. 4.3.

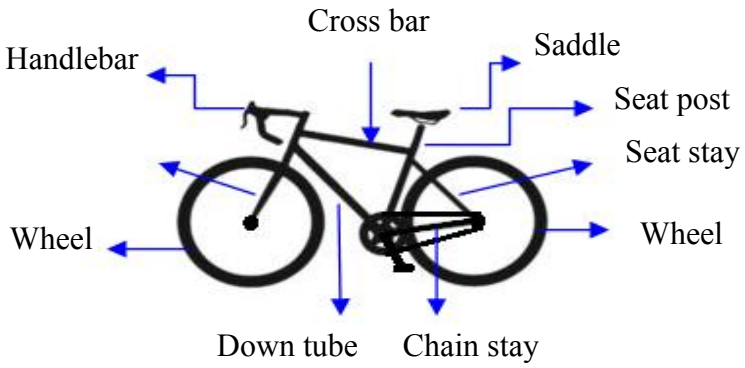


Fig. 4.3. Basic parts of bicycle.

RANSAC circle fitting [FIS81] is a robust fitting method and appropriate for a dataset containing outliers. This process starts with a small of initial dataset that is progressively enlarged to improve the fitting of a circle by minimizing the sum of the squared distances of the contour points to the circle using the error measure:

$$\varepsilon^2 = \sum_{i=1}^{m} \left(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r \right)^2 \tag{4.8}$$

Where: (x_i, y_i) represents the coordinates of the axes points, (x_c, y_c) represents the coordinates of the centre of the circle, *m* is the number of points and *r* is the radius.

The data is to be classified into inliers and outliers and the circle formed is subject to the condition that no points incorporated into the least squares circle fitting deviates from the fit circle by more than a distance *t*, such that *t* can be chosen so that probability for inliers is for example 0.95.

To fit a circle to a set of points, in the RANSAC circle fitting process three points are selected at random to form a circle. The support for the circle was measured by the number of points that lie within *t*. This process of selection was repeated until a large set of *consensus* points compared to

the total number of points is found and the circle with most support is the best fit. The points within distance *t* are the inliers that constitute the *consensus* set. The centre and the radius of the circumscribed circle to the triangle formed by the three points were allocated as shown in Fig. 4.4 (a). The distance from all the other points to the circle, the inliers, the outliers and the size of the *consensus* set were determined as shown in Fig. 4.4 (b). The largest *consensus* set and the best model was selected after a number of trials.

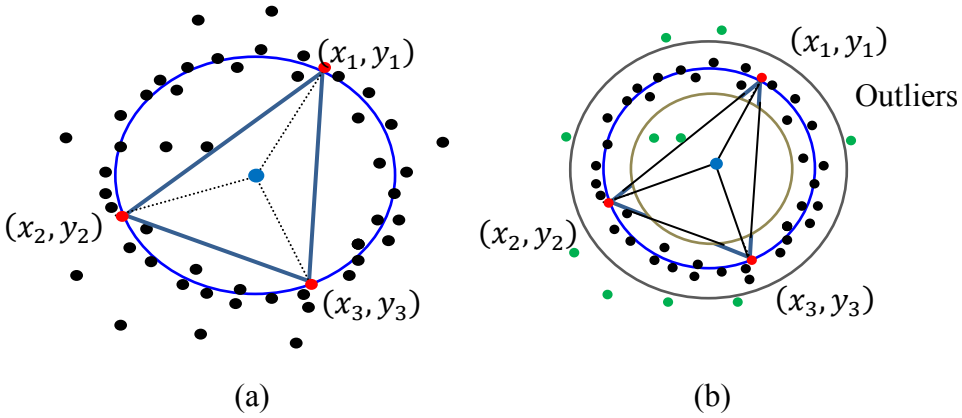


Fig. 4.4. RANSAC circle fitting: (a) circumcircle of the triangle, (b) the inliers and outliers.

The key points were identified on the component axes as described in Section 4.6.

4.6 Key Point Generation

The points on the component axes are used as starting points from which to search for "key points" that sample the boundary and on which the formulation of a model and its interpretation is based.

4.6.1 Search Paths and Edge Responses Detection

To identify the key points along the boundary of an object we search along a set of paths that are constructed perpendicular to selected points from the component axes. The number of axes that are identified depends on the object and is not pre-determined and might vary according to the pose of an object and the presence of occlusion. The total number of points selected on the detected axes is the same for any one class of object. The search path, shown in Fig. 4.5, is constructed perpendicular to the axis to increase the likelihood that the search path will crosses the boundary at angle close to 90 degrees. In Fig. 4.5 the red junctions are points on the component axes and the blue lines the search paths.

Fig. 4.5. A set of key point search paths (blue lines) from points on axes (red junctions).

The detection of edge responses along the boundary search paths formed on the axes was performed using the MLR computation for a pair of regions as described in Fig. 4.6 that are arranged to straddle the perpendicular paths. Therefore the regions considered for edge detection were extended beyond the likely extent of any object present. The regions A and B of the MLR edge detection mask shown in Fig. 4.6 were each set to a size of 7×7 pixels.

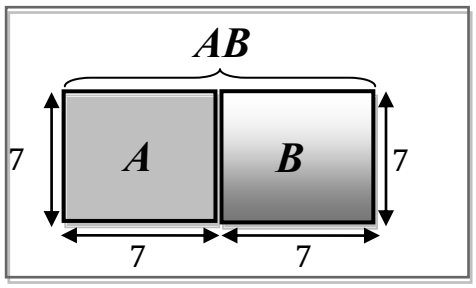
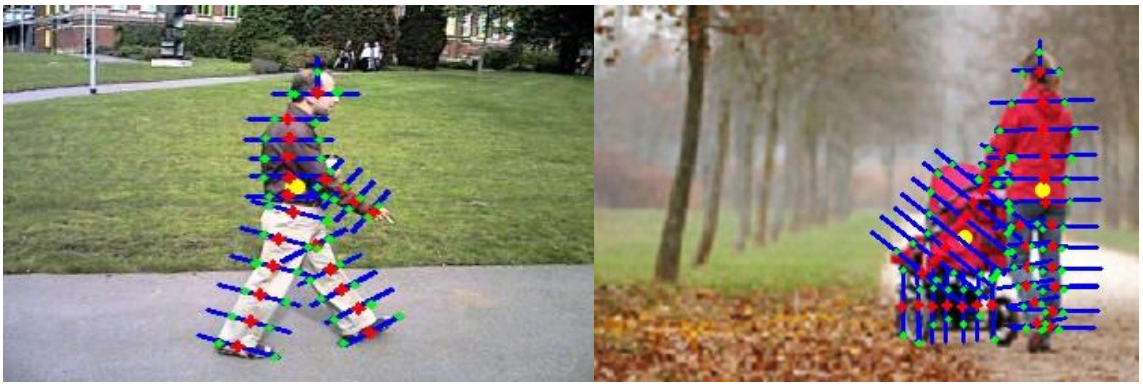


Fig. 4.6. The edge detection mask: MLR regions,
A, *B* and the whole region, *AB*.

4.6.2 Key Point Selection

The selection of key points through the perpendicular paths on the axes is similar to that described in subsection 3.5.2 where the cue is a single axis. A peak hysteresis detector as described in subsection 3.5.2.1 was used to obtain the position of the peak of edge "key points" in the MLR response. One key point on any perpendicular search path was selected to identify a boundary point. This selection process of key points over the complete set of perpendicular paths forms a representative set of key points. A sufficient number of key points on the boundary of each object were selected to represent the form of an object. The greater the number of key points the greater the potential accuracy of the model and interpretation. Initially key points were selected manually. As a model was established it was used to suggest key points and key points were selected from these prompts. The key points were selected along the search paths to identify the boundary of an object that can generate ESE curves with good generality. The density of search paths relates to the quality of the ESE model and computational burden. The key point selection process is illustrated in Fig. 4.7 for objects that vary in structure. In Fig. 4.7 the red points mark points on axes, the blue lines represent the perpendicular search paths, the yellow dots the central cue points for each identified object and the green dots the boundary key points.



(a) (b)

Fig. 4.7. Identified boundary points for: (a) a pedestrian, (b) a pedestrian with a pushchair.

The boundary of the pedestrian in Fig. 4.7 (a) is identified by 43 key points; one key point on each search path. Fig. 4.7 (b) shows a pedestrian pushing a pushchair with the boundary of the person identified by 43 key points. The boundary of the pushchair is identified by 28 key points, one on each search path. It is observed from Fig. 4.7 (a) and (b) that the number of search paths and the number of key points along the boundary of people are the same for the same type of object regardless of the pose of the pedestrian. The initial cue points and the boundary key points on each object are spatial coordinates that enable the sampling angle of the exponential function of an ESE to be estimated. The boundary key points that do belong to the object of interest are identified as not fitting the model and set aside. The applications of the reviewed edge methods in Section 2.5 are limited and not appropriate to find the edges along the search paths identified as described at the end of subsection 3.5.2.2.

4.7 Summary

This chapter has described an approach for generating axes for complex objects, identifying the type of object to which they relate and how to identify the boundary key points.

Chapter 5 GEOMETRY AND APPEARANCE MODELLING

5.1 Introduction

This chapter introduces the geometry and appearance models for the representation and interpretation of objects. This representation is based on a parametric Extended SuperEllipse (ESE) integrated with a statistical distribution representation is used to model the boundary shape of an object and referred to as an Extended Superellipse Geometry Model (ESGM). The texture appearance captured from images is modelled and combined with the ESGM to create the Extended Superellipse Appearance Model (ESAM).

The ESAM uses key points to represent a shape in the exponent space of the ESE function. In training to create a model similar representative objects are selected. The degree of similarity that is appropriate depends on the context and the degree of specificity required. The objective is to design a colour appearance model (that incorporates a geometric model), is flexible and reliable in representing a wide range of objects such as pedestrians, pushchairs, bicycles and vehicles.

5.2 Outline of the Model-Based Approach

In the development of ESAM the following issues were considered:

Cue detection: how to locate a single point or axis points for an object.

Edge detection: how to identify the key points along the boundary of an object.

Geometric modelling: how to create the exponent models for an ESE.

Appearance modelling: how to combine the ESGM with the colour intensity of the images to form the ESAM

5.3 ESE Modelling

The implicit form of a 2D ESE is defined as [ZH099]:

$$\left(\frac{x}{a}\right)^{\frac{2}{f_1(\theta)}} + \left(\frac{y}{b}\right)^{\frac{2}{f_2(\theta)}} = 1 \tag{5.1}$$

Where: a and b are scale parameters that define the size of the ESE in the $x - y$ plane, θ is the angular index parameter of the exponent functions in the $x - y$ plane, $f_1(\theta)$ and $f_2(\theta)$ are the relative shape exponent functions that vary with angle of orientation, θ, which control the shape of ESE surface and x and y are two dimensional points sampled on the ESE surface.

The ESE can be expressed in parametric form, in terms of x and y coordinates as:

$$\begin{bmatrix} x = a\ \text{sign}(\cos(\theta))|\cos(\theta)|^{f_1(\theta)} \\ y = b\ \text{sign}(\sin(\theta))|\sin(\theta)|^{f_2(\theta)} \\ \theta = \arctan\left(\frac{y}{x}\right) \end{bmatrix} \tag{5.2}$$

Where: θ is an angular parameter in the parametric representation of the ESE that is used for two purposes: as the argument of $\sin(\,)$ and $\cos(\,)$ and as the parameter of $f_1(\,)$ and $f_2(\,)$.

5.3.1 Representation of Exponent Functions of the ESE

Modelling in the exponent space is achieved by a re-parameterization of spatial coordinates to provide the exponent functions $f_1(\theta)$ and $f_2(\theta)$ as:

$$f_1(\theta) = \log\left(y/a * \text{sign}(\cos\theta) * \tan(\theta)\right)/\log|\cos\theta| \tag{5.3}$$

$$f_2(\theta) = \log\left(x * \tan(\theta)/b * \text{sign}(\sin\theta)\right)/\log|\sin\theta| \tag{5.4}$$

The 2D coordinates point x_i and y_i at index *i* can be represented by two respective exponent values f_{1i} and f_{2i} at angle θ_i as:

$$f_{1i}(\theta_i) = \log\left(y_i/a * \text{sign}(cos\theta) * \tan(\theta_i)\right)/\log|\cos\theta| \quad (5.5)$$

$$f_{2i}(\theta_i) = \log\left(x_i * \tan(\theta_i)/b * \text{sign}(\sin\theta)\right)/\log|\sin\theta| \quad (5.6)$$

$$\theta_i = \arctan\left(\frac{y_i}{x_i}\right) \quad (5.7)$$

Where: $\left(f_{1i}(\theta_i), f_{2i}(\theta_i)\right)$ is defined as a pair of exponent function values at angle θ_i.

Fig. 5.1 shows the pseudo-code for computing the spatial coordinate parameters x_i and y_i at index *i* using the ESE in parametric form, given the exponent points of the exponential functions $f_1(\theta)$ and $f_2(\theta)$ of the ESE curve.

$\|.\|$	denotes an absolute value.
sign	returns 1 if input is positive, -1 if it is negative and 0 if it is 0
K	Number of key points.
a and *b*	are scale parameters that define the size of the ESE in the *x* and *y* axes.
$f_{1i}(\theta_i)$ and $f_{2i}(\theta_i)$	are a pair of exponent function points at angle θ_i.
θ_i	the angular parameter of the exponent functions in the *x* – *y* plane.
x_i and y_i	are two 2D spatial coordinate points sampled at index *i*.

1. Read *K*
2. Set the values of *a* and *b*
3. The exponent key function points $f_{1i}(\theta_i)$ and $f_{2i}(\theta_i)$ and the angular index parameter θ_i are given in Equations 5.5, 5.6 and 5.7, respectively.
4. For *i= 1 to K*
   - 4.1 $x_i = a\ \text{sign}(\cos(\theta_i))|\cos(\theta_i)|^{f_{1i}(\theta_i)}$
   - 4.2 $y_i = b\ \text{sign}(\sin(\theta_i))|\sin(\theta_i)|^{f_{2i}(\theta_i)}$

   End For

Fig. 5.1. A pseudo-code description of the extended superellipse.

The exponent values for all key points of an example shape were used to construct the full curves of an ESE. To avoid computing log (0) zero was approximated as 10^{-12}.

The pairs of key point exponent vectors of an ESE for all example shapes of a set can be described as:

$$F_1(\theta) = [\mathbf{f_{11}}(\theta), \mathbf{f_{12}}(\theta), \cdots, \mathbf{f_{1i}}(\theta), \cdots, \mathbf{f_{1N}}(\theta)]^T \quad (5.8)$$

$$F_2(\theta) = [\mathbf{f_{21}}(\theta), \mathbf{f_{22}}(\theta), \cdots, \mathbf{f_{2i}}(\theta), \cdots, \mathbf{f_{2N}}(\theta)]^T \quad (5.9)$$

Where: $F_1(\theta)$ and $F_2(\theta)$ is the training set of ordered exponent vectors, $\mathbf{f_{1i}}(\theta)$ and $\mathbf{f_{2i}}(\theta)$ is pair of exponent vectors at index *i* and *N* is the number of exponent vectors.

The exponent vectors $\mathbf{f_{1i}}(\theta)$ and $\mathbf{f_{2i}}(\theta)$ can be combined to form $\mathbf{f_{ei}}(\theta)$, expressed as:

$$\mathbf{f_{ei}}(\theta) = \left(f_{ei_1}(\theta_{i_1}), f_{ei_2}(\theta_{i_2}), \cdots, f_{ei_j}\left(\theta_{i_j}\right), \cdots, f_{ei_K}(\theta_{i_K})\right)^T \quad (5.10)$$

Where: *e* denotes one of the pair of exponent vectors. Here *e = 1* denotes the first exponent vector and *e = 2* the second exponent vector, *K* is the number of points considered for each shape and $f_{ei_j}\left(\theta_{i_j}\right)$ is the j^{th} value of the exponent vectors $\mathbf{f_{1i}}\left(\theta_{i_j}\right)$ and $\mathbf{f_{2i}}\left(\theta_{i_j}\right)$ at angle, θ_{i_j}.

Each shape vector $\mathbf{f_{ei}}(\theta)$ can be represented by a list of exponent values of a pair of two exponent vectors, $\mathbf{f_{1i}}(\theta)$ and $\mathbf{f_{2i}}(\theta)$, as:

$$\mathbf{f_{ei}}(\theta) = \left[f_{1i1}(\theta_{i1}), \cdots, f_{1ij}(\theta_{ij}), \cdots, f_{1iK}(\theta_{iK}), f_{2i1}(\theta_{i1}), \cdots, f_{2ij}(\theta_{ij}), \cdots, f_{2iK}(\theta_{iK})\right]^T \quad (5.11)$$

Where: $f_{1ij}(\theta_{ij})$ and $f_{2ij}(\theta_{ij})$ represent the key point exponent values for the shape at angle, θ_{ij}.

The pair of exponent vectors $F_1(\theta)$ and $F_2(\theta)$ can be combined to form $F_e(\theta)$, expressed as:

$$F_e(\theta) = [\mathbf{f_{e1}}(\theta), \mathbf{f_{e2}}(\theta), \cdots, \mathbf{f_{ei}}(\theta), \cdots, \mathbf{f_{eN}}(\theta)]^T \quad (5.12)$$

The ESE is a parameterised representation of a curve, where fitting a dataset to an ESE model is a curve approximation process. Therefore, the curve of the best model interpretation is not a piece-wise linear curve passing through each key point but a smooth curve passing close to each point. Each full curve of an ESE constructed from a pair of exponent vectors can represent an example shape sampled with a set of key points and each shape key point can be constructed from the exponent values of an ESE curve.

5.4 ESGM Building

Using an ESE the key exponent points define a curve (or a surface), that is modelled in the exponent domain. In order to describe an object, both the exponent vectors representing the shape and the parameters representing the colour texture of the object are needed. The ESGM is built as described, in outline, in Fig. 5.2.

1. For the first *U* images generate candidate boundary key points.
   1.1 Select key points manually.
   1.2 Build statistical models, $x_1(\theta)$ and $x_2(\theta)$ of an ESE as described in Fig 5.2 and Section 5.4.2.
2. Repeat for each successive image:
   2.1 Find candidate boundary key points and transform to exponent values as described in Equations 5.5 and 5.6.
   2.2 Match instances of the exponent models, $x_1(\theta)$ and $x_2(\theta)$, to the exponent vectors of each successive image using the matching procedure described in Fig. 6.2.
   2.3 Select boundary key points for each previously unseen image using the model.
   2.4 Update the statistical models, $x_1(\theta)$ and $x_2(\theta)$.

   Until no more images

Fig. 5.2. Overview of initial model building algorithm.

Initially key points were manually selected from the edge responses generated along each search path as described in Sections 3.5 and 4.6 to generate an initial model. This initial model is used to guide the selection of subsequent key points and reduce manual interaction in model building.

Matching a model instance to an unseen image to create the key points for the new images is described in Chapter 6. A model is created by first registering the key point exponent vectors with the first vector. Then, each vector is re-registered with the average of the key point vector. Statistical models of exponent variation were generated as described in Fig. 5.3.

$F_1(\theta)$ and $F_2(\theta)$ are the pairs of ordered exponent vectors training set.
N is the number of pairs of the exponent vectors.

1. For each exponent vector in $F_1(\theta)$ and $F_2(\theta)$ from *2* to N
   - 1.1 Register to vector *1* as described in Section 5.4.1.
     - 1.1.1 Compute the average exponent vectors, $\bar{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\bar{\mathbf{f}}_{\mathbf{2}}(\theta)$.
   - 1.2 For each exponent vector in $F_1(\theta)$ and $F_2(\theta)$ from *1* to N
     - 1.2.1 Register each exponent vector in $F_1(\theta)$ and $F_2(\theta)$ to the respective average exponent vectors, $\bar{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\bar{\mathbf{f}}_{\mathbf{2}}(\theta)$.
2. Compute the covariance across the registered exponent vectors.
3. Compute the eigenvectors and eigenvalues of the vectors as described in Section 5.4.2.

Fig. 5.3. ESGM Building algorithm.

5.4.1 ESGM Registration Process

The ESGM registration process is performed in the exponent domain because the model is formed in that domain and it is necessary to remove systematic variation in that domain to optimise the statistical model. The exponent vectors in the training set represent images in a range of scales and at varied locations. Therefore, the correlation between both the translation and scale aspects and the exponent vectors is necessary to align the exponent vectors to remove any arbitrary variations before statistical analysis is performed.

Translation in the exponent domain corresponds to object rotation. Translation of magnitude is scale in the geometric domain. Changes in scale in the exponent domain are also a mixed influence of a similar nature. There is no great value to rotate the exponent values of the ESE curves in the exponent domain. Rotation is a curious change of orientation in the exponential

domain. Therefore, translation and scale aspects only are necessary and valid operations to register the points in the exponent domain.

The ESE curves were aligned to a common reference vector to allow the corresponding points to compare in different exponent vectors so that the exponent vectors correspond as closely as possible. There was a constraint placed on the registration method to ensure that the exponent vectors well-defined. The vectors were initially translated in such a way such that the exponent shapes are centred at the origin (centre of gravity).

A transformation matrix obtained in scale and translation was used to transform a new set of key points to the model set. The key point coordinates, transformed to the exponent domain of the ESE as defined in Equations 5.5 and 5.6 were aligned by minimising the sum of differences, as defined in Equation 5.13, between the points of the exponent vectors and the points of the mean vector. This places a new instance at an appropriate initial position with respect to the mean model. Therefore, each exponent vector has a scale similar to that of the mean vector.

At first glance, it seems more theoretically sound to register the key points in the spatial domain and then finding the ESE curves in the exponent domain, however, to register the points of the shapes in the spatial domain and then perform the ESE parameterization by transforming the aligned shapes to exponent vectors increases the computational burden during an image interpretation. Also, it is computationally more efficient to register the key points in the exponent domain because both the model and the statistical analysis are formed in that domain. Therefore, there is

no need to register the points in the spatial domain and then computing the ESE vectors and hence the registration process in the exponent space supports this as being expedient.

To illustrate how to align two exponent vectors of *g* and *j* with each other; assume that $\mathbf{f_{eg}}(\theta)$ is an exponent vector of *K* points which represents the g^{th} shape is to be aligned to $\mathbf{f_{ej}}(\theta)$ which represents the j^{th} shape. These exponent vectors can be defined as shown in Equation 5.10.

Initially the exponent vectors $\mathbf{f_{eg}}(\theta)$ and $\mathbf{f_{ej}}(\theta)$ were centred at the origin; then the scale and translation parameters are calculated to minimise the distance between the points of $\mathbf{f_{eg}}(\theta)$ and the equivalent points of the scaled and translated version of $\mathbf{f_{ej}}(\theta)$. This distance function was defined as given by Sabri et al. [SAB12]:

$$e_j = \left[\mathbf{f_{eg}}(\theta) - R_j\left[\mathbf{f_{ej}}(\theta)\right]\right]^T \left[\mathbf{f_{eg}}(\theta) - R_j\left[\mathbf{f_{ej}}(\theta)\right]\right] \tag{5.13}$$

Where: $\mathbf{f_{eg}}(\theta)$ is a pair of exponent vectors $\mathbf{f_{1g}}(\theta)$ and $\mathbf{f_{2g}}(\theta)$, $\mathbf{f_{ej}}(\theta)$ is a pair of exponent vectors $\mathbf{f_{1j}}(\theta)$ and $\mathbf{f_{2j}}(\theta)$ and R_j is a transformation matrix of scale and translation parameters as defined below [COO95] [SAB12]:

$$R_j = \begin{bmatrix} s_{1j} & s_{2j} \\ -s_{2j} & s_{1j} \end{bmatrix} + \begin{bmatrix} t_{1j} \\ t_{2j} \end{bmatrix} \tag{5. 4 }$$

Where: s_{1j} is a scale parameter for $\mathbf{f_{1j}}(\theta)$, s_{2j} is a scale parameter for $\mathbf{f_{2j}}(\theta)$, t_{1j} is a translation parameter for $\mathbf{f_{1j}}(\theta)$ and t_{2j} is a translation parameter for $\mathbf{f_{2j}}(\theta)$.

The aim is to minimise e_j such that $\mathbf{f_{eg}}(\theta)$ best maps to $\mathbf{f_{ej}}(\theta)$:

$$\mathbf{f_{eg}}(\theta) = R_j\left[\mathbf{f_{ej}}(\theta)\right] \tag{5.15}$$

Applying R_j to the exponent vector $\mathbf{f_{ej}}(\theta)$ yields [COO95] [SAB12]:

$$R_j \begin{bmatrix} \mathbf{f_{1j}} \\ \mathbf{f_{2j}} \end{bmatrix} = \begin{bmatrix} s_{1j}\mathbf{f_{1j}} - s_{2j}\mathbf{f_{2j}} + t_{1j} \\ s_{2j}\mathbf{f_{1j}} + s_{1j}\mathbf{f_{2j}} + t_{2j} \end{bmatrix} \tag{5.16}$$

Equations 5.13 and 5.16 are inspired from aligning the shapes in the spatial domain so that the distances of each shape to the mean is minimised [SAB12]. The solution of Equation 5.16 provides a set of linear equations that lead to the scale and translation parameters. This can be expressed as in the following matrix form [SAB12]:

$$\begin{bmatrix} a_2 & -b_2 & K & 0 \\ b_2 & a_2 & 0 & K \\ d & 0 & a_2 & b_2 \\ 0 & d & -b_2 & a_2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ t_1 \\ t_2 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ c_2 \end{bmatrix} \tag{5.17}$$

The values of scale and translation parameters and the variables defined in (5.17) are given in Tables 5.1 and 5.2, respectively.

Table 5.1. Scale and translation parameters.

Parameter	s_1	s_2	t_1	t_1
Value	c_1/d	c_2/d	a_1/K	b_1/K

Table 5.2. The values of the variables of Equation 5.17.

a_1	$\sum_{i=0}^{K-1} f_{1ji}(\theta_{ji})$	b_1	$\sum_{i=0}^{K-1} f_{2ji}(\theta_{ji})$
a_2	$\sum_{i=0}^{K-1} f_{1gi}(\theta_{gi})$	b_2	$\sum_{i=0}^{K-1} f_{2gi}(\theta_{gi})$
d	$\sum_{i=0}^{K-1} \left(\left(f_{1gi}(\theta_{gi})\right)^2 + \left(f_{2gi}(\theta_{gi})\right)^2 \right)$	c_2	$\sum_{i=0}^{K-1} \left(f_{2ji}(\theta_{ji}) f_{1gi}(\theta_{gi}) - f_{1ji}(\theta_{ji}) f_{2gi}(\theta_{gi}) \right)$
c_1	$\sum_{i=0}^{K-1} \left(f_{1ji}(\theta_{ji}) f_{1gi}(\theta_{gi}) + f_{2ji}(\theta_{ji}) f_{2gi}(\theta_{gi}) \right)$		

The exponent vectors were aligned with the same number of points with one-to-one point correspondences, which is sufficient for the model formulation. This registration procedure makes the geometric model independent of the size and position of the objects and is described in Fig. 5.4.

Initially set *iter=0,* $s_{Max} = 25$, the maximum number of iterations, N is the number of vectors, $\bar{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\bar{\mathbf{f}}_{\mathbf{2}}(\theta)$ are the mean vectors of the training set, $F_1(\theta)$ and $F_2(\theta)$.

1. The mean vectors are scaled to a unit and translated to centre of gravity.
2. Define $\hat{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\hat{\mathbf{f}}_{\mathbf{2}}(\theta)$ as constant means to $\bar{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\bar{\mathbf{f}}_{\mathbf{2}}(\theta)$, respectively.
3. Repeat
   - 3.1 Define $\grave{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\grave{\mathbf{f}}_{\mathbf{2}}(\theta)$ as last mean estimates to $\bar{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\bar{\mathbf{f}}_{\mathbf{2}}(\theta)$, respectively.
   - 3.2 For $i = 1, \cdots, N$
     - 3.2.1 Align each $\mathbf{f_{1i}}(\theta_i)$ and $\mathbf{f_{2i}}(\theta_i)$ to $\bar{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\bar{\mathbf{f}}_{\mathbf{2}}(\theta)$, respectively.
   - 3.3 Compute the new mean vectors, $\bar{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\bar{\mathbf{f}}_{\mathbf{2}}(\theta)$
   - 3.4 Align $\hat{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\hat{\mathbf{f}}_{\mathbf{2}}(\theta)$ to $\bar{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\bar{\mathbf{f}}_{\mathbf{2}}(\theta)$, respectively.
   - 3.5 Compute error term $e = \left[\bar{\mathbf{f}}_{\mathbf{e}}(\theta) - R_j\left[\hat{\mathbf{f}}_{\mathbf{e}}(\theta)\right]\right]^T \left[\bar{\mathbf{f}}_{\mathbf{e}}(\theta) - R_j\left[\hat{\mathbf{f}}_{\mathbf{e}}(\theta)\right]\right]$
   - 3.6 *iter*++

   Until $(iter < s_{Max} || \, e > 1e-4)$

Fig. 5.4. The alignment procedure of the exponent vectors of ESGM.

The exponent vectors were aligned to the mean exponent vector which is recalculated inside each iteration loop until convergence. The search for convergence is stopped when the maximum number of iterations was reached or the newly estimated mean vector has converged, as defined by a threshold on the difference between the points of the newly estimated mean vector at each iteration and the points of the mean of all the aligned exponent vectors.

A set of experiments was performed to test the appropriateness of registration in the exponent domain and to justify the use of scale and translation to register the points in the exponent domain. Fig. 5.5 (a) and (b) show the key point vectors in exponent space and Fig. 5.5 (e) the key point cloud in the coordinate space, before registration. These are shown after registration in the exponent domain in Fig. 5.5 (c), (d) and (f), respectively.

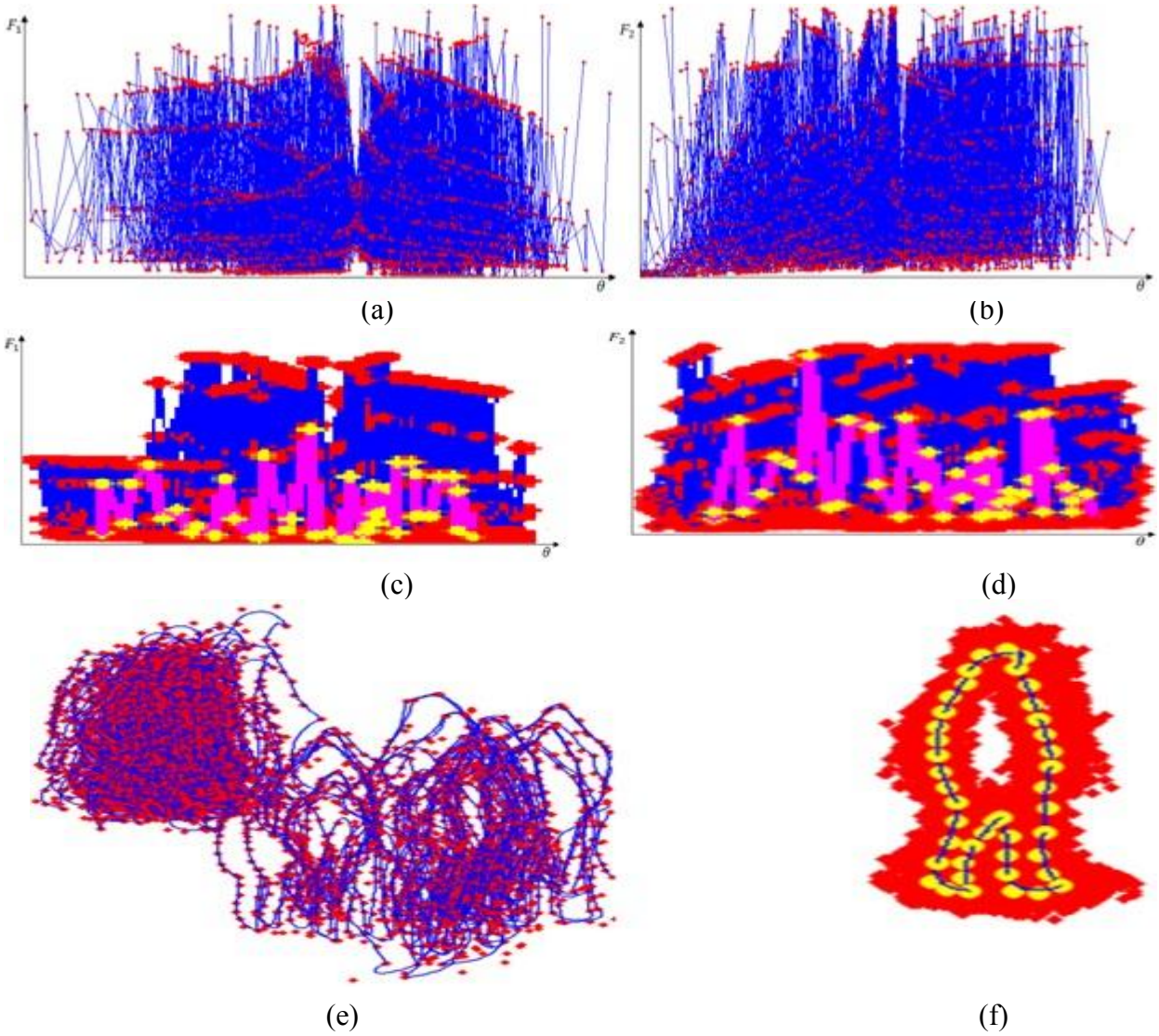


Fig. 5.5. Alignment of key point training data (a) unaligned exponent data $F_1(\theta)$ in red with connecting lines in blue, (b) unaligned exponent data $F_2(\theta)$ in red and connecting lines in blue, (c) aligned data $F_1(\theta)$ in red, connecting lines in blue and the mean curve in cyan, (d) aligned exponent data $F_2(\theta)$ in red, with connecting lines in blue and the mean curve in cyan, (e) unaligned key point data in red and connecting lines in blue and (f) corresponding aligned key point data in red with the mean model as a blue line connecting yellow points.

The red dots in Fig. 5.5 (a) to (d) identify the exponent points for the two exponent vectors in terms of θ, while the blue regions are the piece-wise linear exponent vectors that identify the segments of the curves. The red dots in Fig. 5.5 (e) to (f) identify the spatial coordinates of the key points, while the blue lines in Fig. 5.5 (e) identify the contours of the pedestrian shapes and

the blue line in Fig. 5.5 (f) identifies the mean model. In Fig. 5.5 (c) and (d) the mean exponent models are highlighted in cyan and the yellow dots identify the mean exponent values. Fig. 5.5 (a) and Fig. 5.5 (b) show plots for 180 exponent vectors which represent a set of 90 pedestrians of different sizes and locations. These pedestrians were taken from a training set of 90 images, each contains one pedestrian. It is observed from the plots in Fig. 5.5 (c) and (d) that the registration procedure does not alter the shape of the vectors but rather only find the best fit through scaling and translation. The satisfactory results in Fig. 5.5 (c), (d) and (e) confirm the appropriateness of registration in the exponent domain. This implies the importance of such transformations to significant aspects of each appearance of pedestrian pose and variation.

5.4.2 Modelling the ESGM

The registered set of exponent vectors $F_e(\theta)$ is modelled by generating a covariance matrix and performing eigenanalysis on that matrix. The mean exponent points and the associated covariance matrix are a model of the shape geometry. The variability of an object captured by this covariance matrix S_e is defined as:

$$S_e = \frac{1}{N}\sum_{i=1}^{N}\left(F_e - \bar{\mathbf{f}}_\mathbf{e}\right)\left(F_e - \bar{\mathbf{f}}_\mathbf{e}\right)^T \tag{5.18}$$

Where: $\bar{\mathbf{f}}_\mathbf{e}$ is the average vector of the ordered pairs of exponent vectors.

The eigenvectors selected to represent the (geometry) model are the first v_e with the largest eigenvalues, i.e. the sets of the first, v_1 and v_2 eigenvectors, which contribute in a major way to geometric model and its instances. The ESGM (in exponent space) is expressed as:

$$x_e = \bar{\mathbf{f}}_\mathbf{e} + P_{se}\mathbf{b}_\mathbf{se} \tag{5.19}$$

Where: $P_{se} = \left(\mathbf{p}_{\mathbf{se}_1}, \mathbf{p}_{\mathbf{se}_2} \cdots, \mathbf{p}_{\mathbf{se}_{\mathrm{ve}}}\right)$ represents the respective matrices whose columns are unit eigenvectors for the two geometry exponent models with the selected eigenvectors and $\mathbf{b}_\mathbf{se} = \left(b_{se_1}, \cdots, b_{se_{ve}}\right)^T$ are the vectors that control the variation of each mode for each model.

5.4.3 Modes of Variation of ESGM

Matching the variations of the ESGM to an image can be considered as selecting the best projection of ESGM, which is an exponent model, onto the exponent vectors. Rearranging Equation 5.19 to represent the geometric models of the training data:

$$\mathbf{b}_\mathbf{se} = P_{se}^{-1}\left(x_e - \bar{\mathbf{f}}_\mathbf{e}\right) \tag{5.20}$$

Equation 5.20 embodies the exponent models of the ESGM. The eigenvectors, P_{se}, and the associated eigenvalues are ordered such that $\lambda_{e_i} \geq \lambda_{e_i+1}$; where: λ_{e_i} is the i^{th} largest eigenvalues.

The limits of variation for $\mathbf{b}_\mathbf{se}$ were selected so that the degree of variation is within the range of the variation of the data used to form the ESGM, i.e. $\left|b_{e_i}\right| < 3\sqrt{\lambda_{e_i}}$, for $i = 1, \cdots, v_e$, that represents the indices of the elements of $\mathbf{b}_\mathbf{se}$ and $\boldsymbol{\lambda}_\mathbf{e}$. The v_e largest eigenvalues were chosen such that:

$$\sum_{i=1}^{v_e} \lambda_{e_i} \geq p \sum_{i=1}^{2K} \lambda_{e_i} \tag{5.21}$$

Where: p defines a fraction of eigenvectors selected to represent the first v_e of the total variation and $\sum_{i=1}^{2K} \lambda_{e_i}$ defines the total variance in the training set.

5.5 Augmentation with Texture Model

To model and identify objects in colour the colour texture was incorporated into the model and the interpretation process as described in Fig. 5.6.

1. For each key point warp the image values around that point.
   - 1.1 Normalise the image values to remove the global variations between images.
   - 1.2 Compute the mean and covariance of the warped image values.
   - 1.3 Compute the eigenvectors and eigenvalues of the warped image values.
2. Form the combined model by combining the ESGMs and appearance models.

Fig. 5.6. Adding a model of colour appearance to an ESGM.

The colour texture vector, **g**, is represented by:

$$\mathbf{g} = (g_1, g_2, \cdots, g_M)^T \tag{5.22}$$

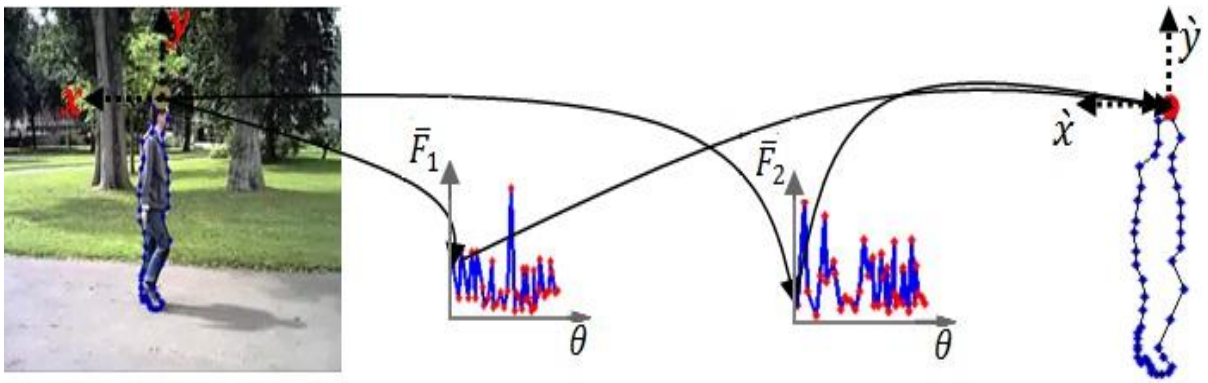
Where: *M* denotes the number of pixels in the texture vector of the object.

Building a statistical model of texture variation requires the pixel values of each training image to be warped to the geometrical mean of the geometry model using a triangulation based on the locations of the key points as described in subsections 5.5.1 and 5.5.2, below.

5.5.1 Warping the Images

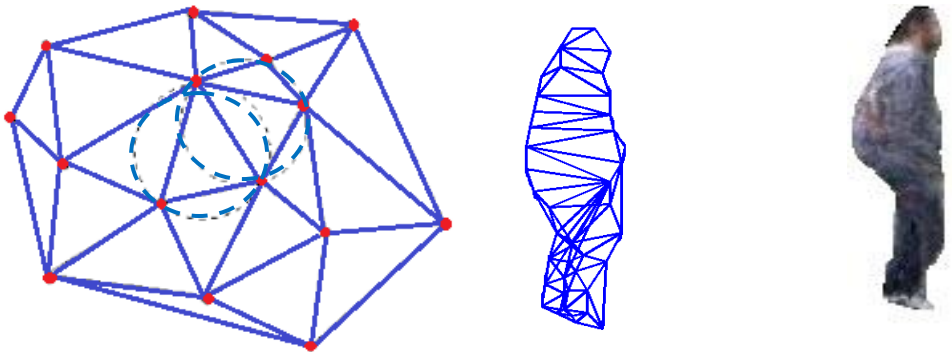
Image warping [COO04] was used to sample the image patch values at the key points, identified on the boundary of each object of interest, as shown in Fig. 5.7. These patches were extracted from the original images at the key points locations which were used to sample the shape of the object being modelled. The image pixel values were computed at positions within regions as determined by the position of points in the mean geometry model. This is to obtain a reference image, which is a geometric normalisation of the image texture values to the mean geometric model.

A triangulation technique, as described below, was used to divide each training image into a triangular tessleation based on the key points, whereas the tesellation covers a part of an image as shown in Fig. 5.8 (b). Normalised versions of each training image patch, which represents a small part of the target image, extracted at key points, were interpolated by relating the pixels of each image patch to the corresponding positions in the reference image patch. This is to eliminate arbitrary variations of texture values between the training image patches due to variations of geometry. These image texture patches cover the regions of objects of interest in the images as shown in Fig. 5.8 (c) where the background was eliminated from the image. Further details are presented in subsection 7.6.1. For example, consider an image $I(x, y)$ is to be warped to a new reference image $\acute{I}(x, y)$. That is, the control points of an object in $I(x, y)$, with K key points, denoted by $\{z_1, z_2, \cdots, z_K\}$ are to be interpolated to new positions $\{\acute{z}_1, \acute{z}_2, \cdots, \acute{z}_K\}$ in $\acute{I}(x, y)$ in a point-to-point correspondence strategy as illustrated in Fig. 5.7. Interpolation was calculated by transforming the point position from the destination image $\acute{I}(x, y)$, to sample and interpolate image values from the original image $I(x, y)$. This is a reverse mapping from $\grave{z}_i$ to z_i; this is formally expressed as a continuous valued mapping vector G. That is: $G(z_i) = \grave{z}_i, \quad \forall i = 1, \cdots, K$.



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

Fig. 5.7. Point to point correspondence, (a) the original image $I(x, y)$, (b) the mean exponent $\bar{\mathbf{f}}_1$, (c) the mean exponent $\bar{\mathbf{f}}_2$ and (d) the created shape $\acute{I}(x, y)$.



(a) (b) (c)

Fig. 5.8. a) A Delaunay triangulation example; b) mean shape Delaunay triangulation and c) an example of warped image.

Texture mapping was performed using a piece-wise affine warp in which the convex hull of geometric mean was partitioned into a set of triangles using the Delaunay triangulation algorithm [COO04] [MAR08]. An affine transformation was computed between the control points in the image and the vertices of the triangulation in the geometric mean.

5.5.2 Delaunay Triangulation

The triangulation technique was employed to warp the key points and the intermediate points by connecting three key points as a triangle. The triangulation of a set of points is a triangle network whose vertexes are the points and the triangles were chosen so that they do not intercept each other but form a tessellation. The pixels within each triangle were warped to correspond to define values for equivalent pixels in the geometric reference image. The triangles follow the Delaunay property mesh; in which for each triangle there is no point inside the circle passing through the three points of the triangle; the circum circle, see Fig. 5.8 (a). This process was implemented by partitioning the convex hull of the mean into a set of triangles. For a set of points with concave nature, this kind of triangulation produces triangles outside the shape control points, however, a

restrict Delaunay triangulation could be used to overcome this problem. Fig. 5.8 (b) shows the Delaunay triangulation result on the geometric mean control points. These control points will be the reference points since all the texture is processed on this normalised reference frame. Each triangle has its own affine warp and the overall collection is a piecewise affine warp. Each pixel in each image inside a particular triangle was mapped to a point inside the corresponding triangle in the geometrical reference image using barycentric coordinates [MAR08]. Every point inside the triangle can be expressed by relative distances from each of the vertices of the triangle. Fig. 5.9 shows an example of mapping a point z in a triangle defined as a function of corners (p_1, p_2, p_3) to a point $\grave{z}$ in a triangle with corners $(\grave{p}_1, \grave{p}_2, \grave{p}_3)$.

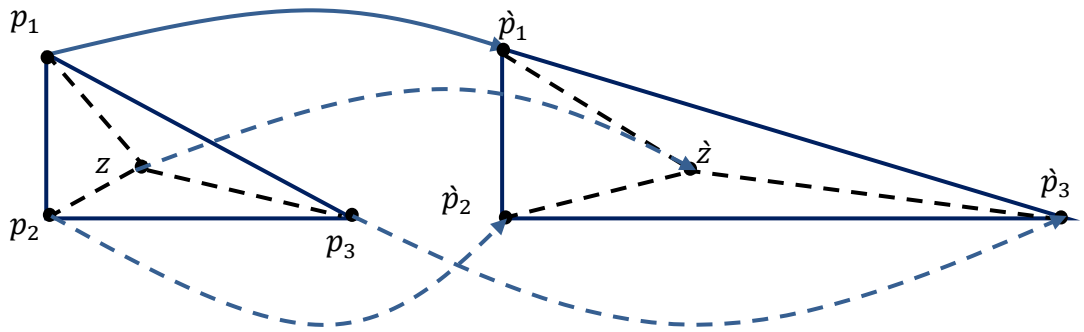


Fig. 5.9. Mapping a point z in a triangle to point $\grave{z}$ in another triangle.

In Fig. 5.9, the point z in a triangle of vertices (p_1, p_2, p_3) can b defined as:

$$z = \alpha p_1 + \beta p_2 + \gamma p_3 \tag{5.23}$$

Where: α, β and γ are real numbers, named as barycentric coordinates of z in relation to p_1, p_2 and p_3 [MAR08].

The barycentric coordinates [MAR08] were employed to determine if a point belongs to a particular triangle. The point z with the parameters α, β and γ is warped, using the affine transformation and relative positions to $\grave{z}$ as:

$$\grave{z} = \alpha \grave{p}_1 + \beta \grave{p}_2 + \gamma \grave{p}_3 \tag{5.24}$$

5.5.3 Modelling Texture Variations

A photometric colour normalisation was applied to the warped texture vectors as described by [COO98]. This colour normalisation process is described in the literature review in subsection 2.7.2.2. Analogous to the geometric model, PCA is conducted on the warped texture vectors to obtain a linear statistical model of texture variation, of the form [COO04]:

$$g = \bar{\mathbf{g}} + P_g \mathbf{b_g} \tag{5.25}$$

Where: $\bar{\mathbf{g}}$ is the mean texture vector, P_g is a set of orthogonal modes of texture variations, which contains the v_g highest texture eigenvectors and $\mathbf{b_g}$ is a set of texture deformation parameters.

The texture parameters for a given sample can be retrieved by:

$$\mathbf{b_g} = P_g^T (g - \bar{\mathbf{g}}) \tag{5.26}$$

The texture parameters are changed in $[-3\sigma, +3\sigma]$ interval. This presents the number of modes of variation composing the percentage variation expressed by the total texture in the training set.

5.6 Extended Superellipse Appearance Model

The ESAM combines both geometric and texture models. The model of texture also embodies aspects of geometry as one image is warped to align with another.

5.6.1 Building the ESAM

The ESAM exploits the relationship existing between the geometry and the texture of an object. The geometry vector, $\mathbf{b_{se}}$, and texture vector, $\mathbf{b_g}$, are concatenated so that benefit can be gained from the correlation between the geometry and texture models. For each image, a vector $\mathbf{c_{ae}}$ is generated as a combination of geometry and texture vectors [COO04]:

$$\mathbf{c_{ae}} = \begin{pmatrix} \Lambda_{se}\mathbf{b_{se}} \\ \mathbf{b_g} \end{pmatrix} = \begin{pmatrix} \Lambda_e P_{se}^T \left(x_e - \bar{\mathbf{f}}_\mathbf{e}\right) \\ P_g^T (g - \bar{\mathbf{g}}) \end{pmatrix} \tag{5.27}$$

Where: Λ_{se} is a matrix of weights that sorts the difference in units between the parameters of the geometry and the texture vectors, such that it was used to make the geometry and texture models appropriately correspond when a combined ESAM is created [MAR08].

5.6.2 Combined Extended Appearance Model

PCA is further applied to model the combined ESAM:

$$\mathbf{c_{ae}} = P_{ae}\mathbf{b_{ae}} \tag{5.28}$$

Where: $\mathbf{c_{ae}}$ are sets of appearance vectors that control the geometry and texture values of the training vectors, P_{ae} are sets of orthogonal modes of ESAM variation that holds the highest eigenvectors and $\mathbf{b_{ae}}$ is a common parameter vector of the appearance that control both the geometry and texture models of the ESAM.

The mean appearance vector, $\bar{\mathbf{c}}_\mathbf{ae}$ of the ESAM does not appear in Equation 5.28 as both $\mathbf{b_{se}}$, $\mathbf{b_g}$ and the ESAM have zero means. The linear structure of the geometry and texture models plus their linear combination means that the models can be expressed in terms of a single set of combined appearance parameters:

$$x_e = \bar{\mathbf{f}}_\mathbf{e} + P_{se}\Lambda_{se}^{-1}P_{cse}\mathbf{b_{ae}} \tag{5.29}$$

$$g = \bar{\mathbf{g}} + P_g P_{cg}\mathbf{b_{ae}} \tag{5.30}$$

Where: P_{cse} and P_{cg} are matrices describing the geometry and texture variation.

5.6.3 An ESAM Instance Example

A new image can be synthesised based on the statistical ESAM with the appearance vector $\mathbf{b_{ae}}$. The geometry parameters, $\mathbf{b_{se}}$, and the parameters of the texture model, $\mathbf{b_g}$, can be derived from the ESAM parameters. The texture of a new image is generated using Equation 5.30 and warped to fit the control points of the geometry model given by Equation 5.29. The appearance parameters can be retrieved for the image using:

$$\mathbf{b_{ae}} = P_{ae}^T\mathbf{c_{ae}} \tag{5.31}$$

The parameters of the ESAM are varied in the range of $\pm 3\sigma$. The eigenvectors are chosen to compose a fraction of the total model variance. Each eigenvector provides a mode of variation.

An ESAM can be used to represent an object as a single model and to model each object of a set of objects. The interpolation axis detection filter introduced in Chapter 4 is able to identify the axes for articulated or complex objects such as pedestrians alone or in groups and pedestrians associated with pushchairs and bicycles at a variety of angles. The components of complex objects can be identified by their axes using simple descriptive criteria as introduced in Section 4.4. Thus a hierarchy of object components can be created and used to direct further interpretation.

5.7 Modelling Objects of Variant Forms

The variability of some objects necessitates different models to be defined and selected for interpretation. Here, the ESAM was used to represent some objects of variable forms and poses. Confidence in an interpretation can be estimated by distance of the perturbed model to the data using log likelihood ratios and standard deviations (SDs) to allow variant models to be selected. The log likelihood ratios for the data points that sample the boundary of an object and the pixels values that represent the object and the SDs that represent the variation between the variant objects types were computed for each interpretation and each instance model. These criteria represent the variation between the variant models of variant objects based on experimental data that take into consideration the distribution of model parameters. The estimation of the log likelihood ratios and the data for which the SD of the variation between the models is computed is Gaussian distributed. The normal (or Gaussian) distribution is a distribution that represents random variation.

To illustrate the potentiality of ESAM to represents the variability of some objects, five vehicle types in a variety of structures have been considered. A model is generated for each vehicle type. To determine the appropriate model for an object interpretation; the data points that sample the boundary of an object and the pixels that represent the object of interest were employed, whereas an object is represented by the pixel values that describe the interior and those that describe the boundary. Here we consider five vehicle models *A*, *B*, *C*, *D* and *E*. The log likelihood ratios and the standard deviations are computed for each interpretation and each model of the five models:

$$\frac{L_A}{L_B},\frac{L_A}{L_C},\frac{L_A}{L_D},\frac{L_A}{L_E},\frac{L_B}{L_A},\frac{L_B}{L_C},\frac{L_B}{L_D},\frac{L_B}{L_E},\frac{L_C}{L_A},\frac{L_C}{L_B},\frac{L_C}{L_D},\frac{L_C}{L_E},\frac{L_D}{L_A},\frac{L_D}{L_B},\frac{L_D}{L_C},\frac{L_D}{L_E},\frac{L_E}{L_A},\frac{L_E}{L_B},\frac{L_E}{L_C},\frac{L_E}{L_D}$$

Where: L_A, L_B, L_C, L_D and L_E are the log likelihood value for the interpretation with respect to each model, *A*, *B*, *C*, *D* and *E*, respectively.

The SD of each interpretation of each variant model is represented by: $\mathrm{SD}(A)$, $\mathrm{SD}(B)$, $\mathrm{SD}(C)$, $\mathrm{SD}(D)$ and $\mathrm{SD}(E)$.

Given the five variant models, *A*, *B*, *C*, *D* and *E*, there are twenty log likelihood ratios and five SDs as described above. These data form a list of twenty five feature vectors. These vectors provide an indication of confidence in the interpretation of an image and are used to determine if the best model has been used and if not to determine what would be a better alternative vehicle model. The maximum Likelihood method is introduced below.

5.7.1 Log likelihood Function

The Maximum Likelihood (ML) estimation method is a standard tool for parameter estimation. The ML estimation method is consistent and efficient [EDW74]. A drawback of this method arises when applied to non-linear estimation cases, such that the associated likelihood equations required for the derivation of the estimator seldom have a closed form solution. This shortcoming produces a global optimisation issue whereas solving this issue using numerical methods is computationally complex. A set of observation data and a model that describes the distribution of the variables in the data are required in the ML estimation. The aim of the ML is to find the parameters of the model that best explain the data, which produces the largest probability or likelihood of explaining the data, whereas an assumption about the distribution of the data is required.

Let $y_1, \cdots, y_n$ be an independent and identically distributed sample with Probability Density Function (PDF) $f(\, y_i|\mu, \sigma)$, where $y_i \sim \mathcal{N}(\mu, \sigma^2)$. To estimate the mean and standard deviation for a single variable assuming the scores of the variables are normally distributed; the PDF for y_i is [KIN89]:

$$f(\, y_i|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2 \tag{5.32}$$

Where: y_i denotes the score of the variable for the i^{th} observation and the mean (μ) and the standard deviation (σ) are the parameters of the Gaussian distribution.

The likelihood of *n* independent and identically distributed observations is the product of their respective individual densities. The joint likelihood function $\mathcal{L}(\mu, \sigma|\mathbf{y}\,)$ for a vector $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ of observations is given by [KIN89]:

$$\begin{aligned} \mathcal{L}(\mu, \sigma|\mathbf{y}\,) = f(\, y_1,\, y_2, \cdots,\, y_n|\mu, \sigma) &= \prod_{i=1}^{n} f(\, y_i|\mu, \sigma) \\ &= \prod_{i=1}^{n} (2\pi\sigma^2)^{-1/2} \exp -\frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2 \\ &= (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2\right) \end{aligned} \tag{5.33}$$

Where: *n* denotes the number of observations and **y** is a vector of *n* data points.

The log likelihood function $\ell(\mu, \sigma)$ for Gaussian distributed data is [KIN89]:

$$\ell(\mu, \sigma) = \log \mathcal{L}(\mu, \sigma|\mathbf{y}\,)$$

$$= \log \prod_{i=1}^{n} (2\pi\sigma^2)^{-1/2} \exp -\frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2$$

$$= \sum_{i=1}^{n} \log \left[(2\pi)^{-1/2} (\sigma^2)^{-1/2} \exp -\frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2 \right]$$

$$= \sum_{i=1}^{n} \left[-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma^2) - \frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2 \right]$$

$$= -\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2 \qquad (5.34)$$

The ML Estimate (MLE) maximises the likelihood in terms of the mean and standard deviation [KIN89]. That is to differentiate $\ell(\mu, \sigma)$ with respect to μ and σ, then set the partials to 0, and solve for μ and σ. The MLE for the log likelihood which can maximise both μ and σ using the derivatives in terms of μ and σ are identified respectively by $\hat{\mu}_{MLE}$ and $\hat{\sigma}_{MLE}$. To solve for maximum μ, the first derivative of the log likelihood is set to 0 for any value of μ:

$$\hat{\mu}_{MLE} = \frac{\partial \ell(\mu, \sigma)}{\partial \mu}$$

$$= \frac{\partial}{\partial \mu}\left[-\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2 \right]$$

$$= \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n} y_i \qquad (5.35)$$

That is, the maximum likelihood estimate of $\hat{\mu}_{MLE}$ is $\bar{y}$.

To solve for maximum σ, set the first derivative of the log likelihood for any value of σ to 0:

$$\hat{\sigma}_{MLE} = \frac{\partial \ell(\mu, \sigma)}{\partial \sigma}$$

$$= \frac{\partial}{\partial \sigma}\left[-\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2\right]$$

$$= -\frac{n}{\sigma} + \sigma^{-3}\sum_{i=1}^{n}(y_i - \mu)^2$$

$$= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5.36}$$

ML estimates the model's parameters for the variant models that produce a distribution that makes the observed data the most probable. MLE's parameters maximise the log likelihood, which in turn optimises the fit between the data and the model. The mean and standard deviation characterise the Gaussian distribution and can be used to estimate the maximum likelihood. Thus, the parameters incorporated into the log likelihood of the model are suitable to be described using Gaussian distribution for each interpretation and each instance model.

A naïve Bayesian classifier [FUK90] was used with the log likelihood ratios for the data points that sample the boundary of an object and the pixel values that represent the object and the SD features that represent the variation between the variant objects types in training and testing to determine the confidence of the match to each model from a set of five variant models of an un-seen image. Here we used a set of variant types of vehicles. A very limited review of the naïve Bayesian classifier is introduced below.

5.7.2 Naïve Bayesian Classifier

A naive Bayesian classifier [FUK90] is a probabilistic classifier with strong assumptions of independence between parameters. The Bayesian classifier aims to identify a class from features. Classes are the categories into which data may be placed. In the Naive Bayesian classifier it is assumed that the input features are conditionally independent of each other. The naive Bayesian classifier is appropriate for the task of finding the confidence of match to each vehicle model because it is a simple and a computational efficient algorithm [CHE09], has high classification accuracy and possesses well defined criteria for classification purposes [WAN07a]. The naive Bayesian classifier used here is implemented in OpenCV based on [FUK90]. This classifier assumes that the attribute vectors for each class are normally distributed and independent. For more details about naive Bayesian see [FUK90]. Details of the configuration of the Bayesian classifier are introduced in the experimental method Section 7.5.

5.8 Summary

An ESGM that represents a combination of a statistical model with the ESE representation was described and used to model the boundary shape of an abject. A representation of ESAM that combines the ESGM and the texture appearance captured from images has been described and used to model the appearance and geometry of objects. The computational cost of the ESAM depends on the parameters of the coupling of geometry and texture models. How to identify the most appropriate models for each interpretation of an object of variable forms such as vehicles was described.

The next chapter will introduce model training and interpretation.

Chapter 6 MODEL TRAINING AND INTERPRETATION

6.1 Introduction

This chapter introduces two main issues:

- How to train and match the ESGM to unfamiliar images.
- How to train and match the ESAM to a previously unseen image.

6.2 Training and Fitting of the ESGM

6.2.1 ESGM–Based Training

The parameters generated in the training process of the ESGM are: the average exponent vectors $\bar{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\bar{\mathbf{f}}_{\mathbf{2}}(\theta)$, the eigenvectors, P_{s1} and P_{s2}, that are corresponding to the v_1 and v_2 of the respective largest eigenvalues, the vectors $\mathbf{b_{s1}}$ and $\mathbf{b_{s2}}$ that control the variation of each mode of the ESGM, the transformation matrix, R_t, of scale and translation parameters. The parameters generated from the ESGM-based training process are used for matching an instance of the ESGM to interpret an unseen image by generating the exponent vectors that model the boundary shape of an object in an unseen image. The ESGM search algorithm was used to locate points in an unseen image with constraints applied to the parameters of the ESGM, where the limits of variation for $\mathbf{b_{se}}$ are within the range $\left|b_{se_i}\right| \leq 3\sqrt{\lambda_{e_i}}$, where b_{se_i} is the i^{th} element of the vector $\mathbf{b_{se}}$ that controls the variation of a mode for the model and λ_{e_i} is the i^{th} largest eigenvalue of the model. Matching the ESGM to a previously unseen image is described in subsection 6.2.2, below.

6.2.2 ESGM-Based Interpretation

Model matching was designed to find the correspondence between the model and an image. For an unseen image to be interpreted twelve potential key points were identified along each search path by the edge detection method described in subsection 3.5.2.1. Twelve points were selected to offer a good likelihood that the correct boundary was identified and the number of points on each search path is a balance between speed and robustness; the number of points on each search path is not critical. The points identify a set of boundary contours of an object. Fig. 6.1 shows a selection of key points along a selected few radial search paths. The exponent vectors for an unseen image were computed using the selected twelve key points on the identified search paths.

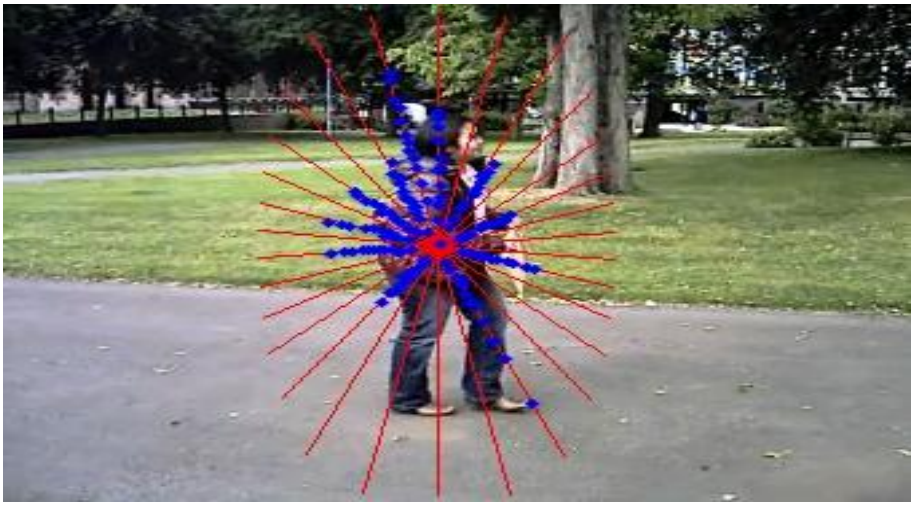


Fig. 6.1. Set of search paths, shown in red, radiating from the central cue point with a set of key points on few search paths, shown in blue, on each search line.

The exponent vectors of a previously unseen image were initially registered to the mean exponent vectors of the ESGM by scaling and translation such that:

$$\hat{\mathbf{f}}_{\mathbf{1}}(\theta) = s_1\hat{\mathbf{f}}_{\mathbf{1}}(\theta) - s_2\hat{\mathbf{f}}_{\mathbf{1}}(\theta) + t_1 \tag{6.1}$$

$$\hat{\mathbf{f}}_{\mathbf{2}}(\theta) = s_2\mathbf{f}_{\mathbf{2}}(\theta) + s_1\hat{\mathbf{f}}_{\mathbf{2}}(\theta) + t_2 \tag{6.2}$$

Where: $\hat{\mathbf{f}}_{\mathbf{1}}(\theta)$ and $\hat{\mathbf{f}}_{\mathbf{2}}(\theta)$ are a pair of exponent vectors of an unseen image, s_1, s_2, t_1 and t_2 are the scale and the translation parameters of the respective exponent vectors.

The ESE vectors are varied around the mean to find the best match between a variant of the model and the image characteristation. In interpretation the aim is to match the ESE vectors with those in the mean model. The scene exponent values were registered to the mean model to reduce the variations between the ESE curve of the exponent vectors of an unseen object and the full ESE curves of the exponent vectors. The model is varied during model matching and it was explained in subsection 5.4.1 that scaling and translation factors are appropriate actions to align the exponent vectors of the unseen image to the model. The matrix R_j, defined in Equation 5.14, was applied to the exponent vectors of a new image at registering the exponent values to the model.

To search for new positions of the key points of a previously unseen image, the vector $\mathbf{b_{se}}$ in Equation 5.20 was varied to identify an instance of the ESGM that best matched the estimated exponent vectors $\hat{\mathbf{f}}_{\mathbf{e}}$ of the unseen image. The model matching algorithm determines the curve representing the object. The match criterion was the sum of the absolute norm distance between the points of the exponent vectors, $\hat{\mathbf{f}}_{\mathbf{e}}$ of an unseen image and an instance of the model, x_e:

$$E_g = \sum_{j=1}^{K} \left\| \hat{f}_{e_j} - x_{e_j} \right\| \tag{6.3}$$

Where: K is the number of key points, j is the index of the key point $\hat{f}_{e_j}$denotes the points of the exponent vectors of the potential interpretations for the unseen image and x_{e_j} identifies the points of an instance of the geometric model as described in subsection 5.4.2.

In the iterative selection and matching of key points, the key point's indexes were considered in turn. All the first key points and third key points on successive search paths were considered for the exponent vectors of the previously unseen image when matched to an instance of the model. This can be described as rotating and translating to register the exponents in the image to the model instances. The iterative matching procedure of ESGM is summarised in Fig. 6.2.

c_1	Iteration counter.
Miter	Maximum number of iterations.
K	Number of key points.
d	A distance threshold.
x_e	An instance of the model.
$\bar{\mathbf{f}}_\mathbf{e}$	Average pair of the exponent vectors of the training set.
$\hat{\mathbf{f}}_\mathbf{e}$	A pair of exponent vectors for a previously unseen image.
$\acute{\mathbf{f}}_\mathbf{e}$	A new estimation vectors of the updated model.
E_g	The error between a pair of exponent vectors of an image and the model.
$\mathbf{t}$	A vector of scale and translation parameters (s_1, s_2, t_1, t_2).
R_j	The transformation matrix.

Set *Miter=25, dist=5,* $E_g = c_1 = 0$.

1. Initialise the ESGM parameters, $\mathbf{b_{s1}}$, $\mathbf{b_{s2}}$, to zero.
2. While (c_1 < *Miter AND* E_g < d) do
   - 2.1 For each exponent function of 12 key points on each search path
     - 2.1.1 Generate a model instance $x_e = \bar{\mathbf{f}}_\mathbf{e} + P_{se}\mathbf{b_{se}}$
     - 2.1.2 Search for $\mathbf{t}$ which best map x_e to $\hat{\mathbf{f}}_\mathbf{e}$ as described in subsection 5.4.1. The first and third points on successive search paths were considered for each modified $\hat{\mathbf{f}}_\mathbf{e}$.
     - 2.1.3 Invert the parameters of R_j and project $\hat{\mathbf{f}}_\mathbf{e}$ into the model, $\acute{\mathbf{f}}_\mathbf{e} = R_j^{-1}(\hat{\mathbf{f}}_\mathbf{e})$
     - 2.1.4 Find the vector $\mathbf{b_{se}}$ of the ESGM that fit to $\acute{\mathbf{f}}_\mathbf{e}$, $\mathbf{b_{se}} = P_{se}^T(\acute{\mathbf{f}}_\mathbf{e} - \bar{\mathbf{f}}_\mathbf{e})$
     - 2.1.5 Apply the limits of variation for $\mathbf{b_{se}}$ as described in subsection 6.2.1.
     - 2.1.6 Test the error term in Equation 6.4 to terminate an iteration

     End For
   - $c_1 = c_1 + 1$

   End While

Fig. 6.2. Identification of the key points in an unseen image using the ESGM.

The selection and matching of key points in Fig. 6.2 is iterated until (i) a maximum number of iterations (*Miter*) is exceeded or (ii) the distance between the model and the interpretation as

measured by E_g, falls below a defined value, d. This results in a pair of exponent vectors, $\acute{\mathbf{f}}_\mathbf{1}$ and $\acute{\mathbf{f}}_\mathbf{2}$, that determines the ESE curve representing the boundary of the object. The complexity of ESGM search is $O(n_d n_p)$; where n_d is the number of data points and n_p is the number of model points, however, since n_p is constant the search process can be considered linear.

6.3 Training and Fitting of ESAM

6.3.1 ESAM Training

The ESAM is trained for the interpretation process and then refined to generate a set of parameters that are required for matching an instance of the model to a previously unseen image. The ESAM training process captures the mean geometry and appearance properties and their variation for the selected images that can be used to constrain the search to identify similar objects in previously unseen images. In the ESAM training process, the vector, $\mathbf{b_{ae}}$, characterises the variability of the geometry and appearance, as described in Equations 5.29 and 5.30, respectively. The set of appearance parameters $\mathbf{b_{ae}}$ are perturbed to create a sequence of model instances. A Jacobian matrix, estimated from the training set, as discussed below, describes how the model parameters guide the interpretation process. The parameters of the ESAM result in the generation of a series of synthetic images, I_m. The difference between the texture of the image I_m and the texture that being interpreted I_i is defined as a vector of residual differences:

$$\mathbf{r}(\mathbf{z}) = I_i(\mathbf{z}) - I_m(\mathbf{z}) \tag{6.4}$$

Where: $\mathbf{z}$ denotes a vector of geometry, texture and appearance parameters of the model, $\mathbf{r}(\mathbf{z})$ is the vector of residuals, I_i is the texture of the current image and I_m denotes the texture in the warped regions of the synthetic image.

Equation 6.4 defines an error term that can be minimised by adjusting the model parameters, given by concatenating the geometry, texture and appearance parameters. The matching algorithm of ESAM searches to minimise the residuals $\mathbf{r}(\mathbf{z})$ between a model instance and the object of interest in a new image. Each image patch is warped from the current position to the reference mean position. The model $I_m(\mathbf{z})$ is derived from the appearance coefficients, $\mathbf{b_{ae}}$, and modified by the intensity parameters as described in subsection 5.5.3. During matching the pixels of the new image are sampled and projected into the texture model. The current texture model is given in Equation 5.30. The textures of image patches synthesised from the model can be represented as:

$$I_m = \bar{\mathbf{g}} + P_g P_{cg} \mathbf{b_{ae}} \tag{6.5}$$

Where: $\bar{\mathbf{g}}$ is the mean texture, P_g is a set of orthogonal modes of texture variations, P_{cg} is a matrix that describes the texture variation and $\mathbf{b_{ae}}$ controls both the geometry and texture models.

The residuals model was learnt during the training process, which generates a matrix of residuals and a matrix of model perturbations. These matrices were used to estimate the Jacobian matrix. The ESAM uses Jacobian matrix, as described below, with $\mathbf{r}(\mathbf{z})$ and the corresponding perturbations $\boldsymbol{\delta}\mathbf{z}$ in model matching. The aim is to find a matrix, R, satisfying the relation:

$$\boldsymbol{\delta}\mathbf{z} = -R\mathbf{r}(\mathbf{z}) \tag{6.6}$$

Where: $\boldsymbol{\delta}\mathbf{z}$ are the perturbations and $\mathbf{r}(\mathbf{z})$ are the corresponding texture residuals of the model.

The matrix R was used with model parameters and texture residuals in all searches to visualise the effects of perturbation and updating the parameters of the model. The correlation of texture

residuals and model parameters was assumed to be linear. Solving Equation 6.6 involves doing a set of *s* experiences to construct the matrices $\boldsymbol{\delta z}$ and $\mathbf{r}(\mathbf{z})$ as previously described [MAR08]. During training, the parameters of the vector $\boldsymbol{z}$ are displaced from its optimal value as suggested by Cootes and Taylor [COO04], and used in all the *s* experiences to compute the matrix *R*. The displacement of each parameter of the model is performed in a way to approximate the gradient of the Jacobian. The degree of displacement in percentage values for scale and translation are linked to the size of the reference means. Table 6.1 describes the model perturbation scheme.

Table 6.1. Perturbation scheme.

Parameter	Perturbation
$\mathbf{b_{a1}}, \mathbf{b_{a2}}$	$\pm.025\sigma_i, \pm0.5\sigma_i$
s_1, s_2	90%, 110% of the reference scale
t_1, t_2	±7%, ±12%

Where: $\mathbf{b_{a1}}$ and $\mathbf{b_{a2}}$ are vectors of appearance that control both the geometry and texture coefficients of the ESAM, s_1,s_2, t_1 and t_2 are the scale and translation parameters of the ESGM.

6.3.2 Jacobian

The squares of the residual vector, ***r***, was minimised to find a match as:

$$E(\mathbf{z}) = |\mathbf{r}(\mathbf{z})|^2 = \mathbf{r}(\mathbf{z})^T\mathbf{r}(\mathbf{z}) \tag{6.7}$$

Equation 6.4 is approximated using the first-order Taylor series expansion of **r** at **z** [COO01a]:

$$\mathbf{r}(\mathbf{z} + \boldsymbol{\delta z}) \approx \mathbf{r}(\mathbf{z}) + \Gamma\boldsymbol{\delta z} \tag{6.8}$$

Where: Γ is the Jacobian gradient matrix as previously defined [COO01a] [MAR08].

Substituting Equation 6.8 into Equation 6.7 and taking the derivative of $E(\mathbf{z} + \boldsymbol{\delta}\mathbf{z})$ with respect to $\mathbf{z}$ and setting it to zero we obtain [MAR08]:

$$R = (\Gamma^T \Gamma)^{-1} \Gamma^T \tag{6.9}$$

Where: R is the pseudo-inverse of the Jacobian matrix.

The Jacobian matrix was fixed and pre-computed over the training set [COO01a]. Its pseudo-inverse regression matrix R is estimated once during the training phase to improve numerical stability and speed and used in all subsequent searches with the model. The geometry and texture parameters along with the Jacobian, model perturbations and residuals matrices are used to fit an ESAM instance to an unseen image as described in subsection 6.3.3.

6.3.3 ESAM-Based Image Interpretation

The pixel values from the patches of an unseen image are registered to the model pixel values, with warping, to provide an accurate instantiation. The ESAM interpretation matches the image pixel values more closely by considering variations of the appearance model as expressed by a reconstruction of a model instance, varied to minimize the texture residual between the model instance and the unseen image. The search to align the deformable object uses the Gauss-Newton optimization method and a pre-computed Jacobian matrix [COO01a] [COO01b] for efficiency. The matrix, R, was estimated in order to correct the model parameters with the texture errors. The Jacobian matrix updates the parameters of the model, during the search progress, by exploiting the relationship between the residual differences and the parameter displacements. The model parameters are updated with a damped Gauss-Newton steepest descent method:

$$\mathbf{z_{k+1}} = \mathbf{z_k} - \zeta(\Gamma^T\Gamma)^{-1}\Gamma^T\boldsymbol{\delta}\mathbf{z} \tag{6.10}$$

Where: Γ is the Jacobian matrix, ζ is a damping factor, $\boldsymbol{\delta}\mathbf{z}$ is the residual texture vector, $\mathbf{z_k}$ is the current estimate appearance vector and $\mathbf{z_{k+1}}$ is the next appearance vector at index *k*.

In the absence of a better initialisation the process can start with $\mathbf{z_0}$ set to zero, where $\mathbf{z_0}$ defines the initial estimate parameters vector of ESAM. The model is initialised with the mean ESAM instance at the position defined by the cue detector. First the locations of the initial key points are identified and an ESAM instance is built with the current estimate of the model parameters, $\mathbf{z_k}$. The vector, $\mathbf{r(z)}$, and the initial error, $e_0 = \mathbf{r(z_0)}\,\mathbf{r(z_0)}^T$ were evaluated. The model parameters are updated using Equations 6.9 and 6.10. Fitting algorithm of ESAM is summarised in Fig. 6.3.

The procedure in Fig. 6.3 is iterated until a maximum number of iterations (s_{Max}) is reached or the improvement in the residual measure is small.

The complexity of the fitting process in Fig. 6.3 is $O\left(n_{pixels}n_{modes}n_{points}\right)$ at a given level. n_{pixels}, n_{modes} and n_{points} are the number of pixels, number of modes and number of points that represent each model. Each iteration involves sampling n_{pixels} points from the image, thus acquisition by sampling n_{points} and multiplying by a $n_{modes} \times n_{pixels}$ matrix.

Separate models for geometry and appearance are defined to reduce the search space for model matching and allowing matching to alternate between geometry and appearance. This also means that geometric matching can be considered first, independent of appearance. The geometry model

is first used to select key points and the ESAM to select the appearance interpretation based on pre-computed geometric constraints. This reduces the computational burden of interpretation. The computational cost of the algorithm in Fig. 6.3 depends on the parameters of both the geometry and texture models. This algorithm is an improvement over the algorithm in Fig. 6.2 since it considers both the positions of key points and the appearance of an object. It is important to search for key point locations as shown in Fig. 6.2 to evaluate the geometric model. This requires fewer computations than matching both the key point locations and the appearance of the objects.

$\mathbf{z}$	The parameter vector of the ESAM.
I_i	The texture of the current image.
I_m	The texture of the synthetic image.
$\mathbf{r}(\mathbf{z})$	The texture differences between I_i and I_m.
R	The pseudo-inverse of the Jacobian matrix.
e_0	$\mathbf{r}(\mathbf{z_0})\,\mathbf{r}(\mathbf{z_0})^{\mathrm{T}}$, is the initial error.
s_{Max}	The maximum number of iterations.
s	The current iteration (counter)
ζ	0.5

Set $s = 0$, $\mathrm{s_{Max}} = 25$, $R = (\Gamma^T\Gamma)^{-1}\Gamma^T$

1. Do
   - 1.1 Sample the texture of an image to get, I_i
   - 1.2 Build an ESAM instance and compute I_m
   - 1.3 Compute the residual texture vector, $\mathbf{r}(\mathbf{z}) = I_i - I_m$
   - 1.4 Evaluate the error, $e_0 = \mathbf{r}(\mathbf{z})\,\mathbf{r}(\mathbf{z})^T$.
   - 1.5 Predict model displacements, $\boldsymbol{\delta}\mathbf{z} = -R\mathbf{r}(\mathbf{z})$
   - 1.6 Set ζ=*1.0*
   - 1.7 Update model parameters, $\mathbf{z_{s+1}} = \mathbf{z_s} + \zeta\boldsymbol{\delta}\mathbf{z}$
   - 1.8 Calculate the new model texture, $(I_m)_s$, with the new model parameters
   - 1.9 Update points from the model and resample the texture of the image at the new points to get, $(I_i)_s$ at iteration *s*.
   - 1.10 Compute the new residual texture vector, $(\mathbf{r}(\mathbf{z}))_s = (I_i)_s - (I_m)_s$
   - 1.11 Evaluate the error at the new iteration, $e_s = (\mathbf{r}(\mathbf{z}))_s(\mathbf{r}(\mathbf{z}))_s^T$,
   - 1.12 if $(e_s < e_0)$
     - 1.12.1 Accept the model parameters, $\mathbf{z_{s+1}}$ and the new points
     - 1.12.2 Update the current error, $e_0 = e_s$
     - 1.12.3 else try other values for ζ such as $0.25, 0.125, 0.0625$.
   - 1.13 *s=s+1*
   - 1.14 Until $(s > s_{Max}\ OR\ e_0 < e_s)$

Fig. 6.3. Summary of algorithm for matching the ESAM to a new

6.3.4 Pyramid Search Approach

Video sequences have to deal with a wide range of input object sizes, which might result in a poor match of the initial model with an actual input shape. Therefore, a pyramid of up to 3 levels method was used for interpretation. In the construction of the pyramid the resolution in *x* and *y* is reduced by a factor of two in each axis from the previous level by averaging pixel values in non-overlapping 2x2 regions. The level *L0* image has been reduced by a factor of two from the original source image. Fig. 6.4 shows an image pyramid with three resolution levels (*L0*, *L1* and *L2*). Model fitting is performed from level *L2* to level *L0* with the model fit at each level used as the starting point for the model fitting at the next.

(a) (b) (c)

Fig. 6.4. Image at 3 levels of resolution: (a) *L0*, (b) *L1* and (c) *L2*.

6.4 Summary

The training and interpretation algorithms of ESGM and ESAM have been described. Using a fixed Jacobian matrix over the training data avoids the need to store all variations of the images generated by the model.

The next chapter will introduce the experimental results and discussions.

Chapter 7 EXPERIMENTAL RESULTS AND DISCUSSION

7.1 Introduction

This chapter describes the experiments conducted on the detection of cues, axes, geometric (ESGM) and appearance (ESAM) models. Section 7.2 introduces the dataset for training and testing. The results for non-interpolated and interpolated cue detectors are presented in Sections 7.3 and 7.4, respectively. Section 7.3 also presents the cue detection results for vehicles. The cue detectors are evaluated in terms of true positive and false positive rates. Axes detection results are also presented in Section 7.4. The results for the ESGM and the ESAM are presented in Sections 7.5 and 7.6 respectively. The results of interpretation for ESGM and ESAM are for people alone, people combined with other objects and vehicles with a brief summary presented in Section 7.7.

7.2 Training and Test Datasets

The dataset consists of 640 x 480 pixel images gathered from a variety of mainly outdoor contexts from the University of Birmingham and from sites on the Internet. The training and test datasets each contain images representing pedestrians, pushchairs, bicycles and vehicles; people are alone and in groups, carrying bags in their hands or over the shoulder, pushchairs are being pushed, bicycles are being ridden or pushed. The chief reason for collecting a new data set is the varied weather conditions and using a high degree of variability in appearance of pedestrians who are individually clothed and having a variety of pose and scale. Moreover, some collected images of pedestrians contain people that may be pushing a pushchair, pushing or riding a bicycle. The environments in which video scenes of pedestrian were captured vary from natural scenes with trees to man-made scenes with roads. As we will demonstrate, the new collected dataset is suit-

able for use with the established methodology of pedestrian detection. However, existing pedestrian datasets often contain a limited range of scale, occlusion and pose variation, making it difficult to assess real world performance. For example, the images of pedestrians collected by Viola et al. [VIO05] are varied in body pose and clothing and the resolution of the images is very low; they were captured in snow and rain conditions. A publicly available INRIA person dataset [DAL05] which has been contributed to spurring progress in the pedestrian detection has fairly high-resolution pedestrians. Further details on Viola et al. [VIO05] and INRIA person datasets [DAL05] and other publicly available datasets were provided in Table 2.3.

The datasets downloaded from the Internet are the Penn-Fudan person database [WAN07], the INRIA person set [DAL05], the ETHZ vehicle database [LEI04] and the TU-Darmstadt vehicle set [EVE06]. The training and test data images are independent and that objects that feature in the training set are not represented in the test set. The images of pedestrians from the University of Birmingham are drawn from more than 220 video sequences and there are more than 3000 images collected from the Internet. The pedestrians are in various poses where they vary in size, scale, level of occlusion and context. There are more than 1500 images representing pedestrians in a range pose and scale in the training sets and more than 1000 in the test sets that were gathered from the University of Birmingham. The images collected from sites on the Internet [DOL12] include a huge number of images with pedestrians at various scales. They also included images with no pedestrians. Finally, there are more than 2500 images in the vehicle image dataset, for a variety of models and body shapes such as: hatchback cars (model A), minivans (model B), compact cars (model C), city cars (model D) and vans (model E). The number of images for training and test datasets are shown in Table 7.1.

Table 7.1. Dataset for training and testing.

Images set	Number of images	
	Training sets	Test sets
Pedestrians (Birmingham University)	1500	1000
Pedestrians (Internet)	2000	1000
Pedestrians pushing pushchairs	500	300
Pedestrians pushing bicycles	500	300
persons riding pushchairs	500	300
Vehicles	1250	1000

Fig. 7.1 shows sample images of pedestrians captured from the University of Birmingham.

Fig. 7.1. Selected images captured from outdoor scenes at the University of Birmingham.

7.3 Non-Interpolated Cue Detector

7.3.1 Experimental Settings of Non-Interpolated Cue Detector

The cue detection mask was processed at a series of scales from 1 to 4. For pedestrian detection cue detector the mask width and height and separation were set to $w_B = 48, h_B = 72$ and 16 pixels, respectively for the first scale. These values were arrived at by evaluating performance for values of w_B varied from 48 to 6, of h_B from 72 to 9 and a gap of from 2 to 16 pixels. The high and low hysteresis thresholds for response selection were set to 40 and 25, respectively. The pedestrian detection requirement was for the detected cue point to be within a 5 pixel wide by 10 pixel high box at the centre of the pedestrian's body, as judged visually. This is a demanding cri-

terion. These experimental conditions were employed for the results reported for the pedestrian cue detector in the following subsections.

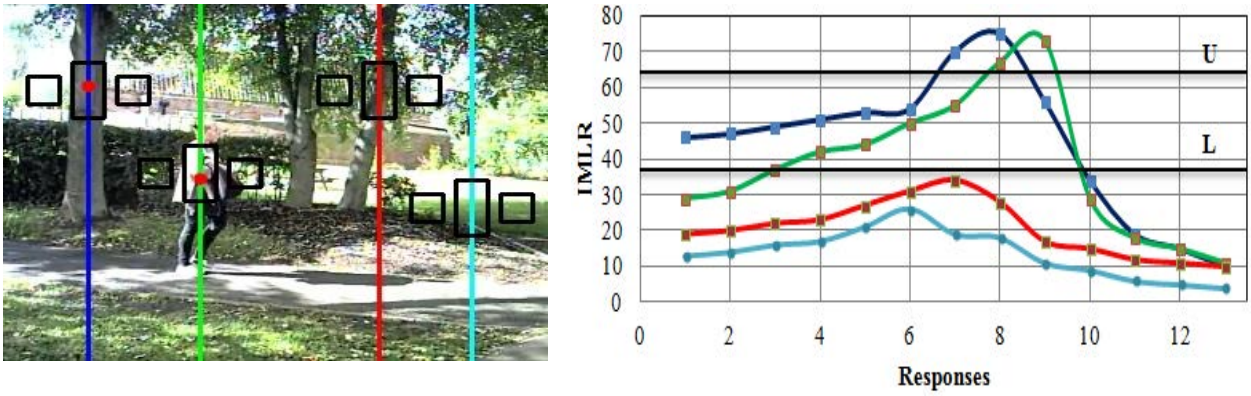
7.3.2 Pedestrian Cues for Various Combinations of Region Sizes

Results of pedestrian cue detection for variations in the size of the central region *B* and the two flanking regions *A* were based on the mask shown in Fig. 3.2 (a) and (b). This parameters were also considered at a series of scales. A set of experiments on a dataset containing a variety of pedestrian images representing a range of scenarios was performed. There is at least one pedestrian in each image and all pedestrians are standing upright. The mean height of the pedestrians captured from the University of Birmingham are between 90 and 210 pixels and the mean height of the pedestrians downloaded from the Internet fall between 180 and 390 pixels. There are 700 colour images representing 3098 pedestrians alone, in groups and carrying bags in good weather with pedestrians of various sizes with simple and complex backgrounds. The number of images and the number of pedestrians in each set of images used for pedestrian cue detection is listed in Table 7.2.

Table 7.2. Number of images and number of pedestrians in each image.

No. of images	150	10	84	68	62	55	44	40	33	27	13	11	8
No. of persons in each image	1	2	3	4	5	6	7	8	9	10	11	12	13

The mask of the pedestrian detector was designed to match to pedestrians and not other objects. However, the cue detector might respond to trees but this should be a rare event as evaluated below. Four sets of responses detected by the pedestrian detector using the MLR criteria, corresponding to the annotated columns in the image of Fig. 7.2 (a) are shown in Fig. 7.2 (b).



(a) (b)

Fig. 7.2. (a) MLR responses and annotated columns, (b) The MLR profile along the lines shown in (a). The lower (L) and upper (U) detection thresholds are shown.

It is observed from Fig. 7.2 (b) that there is a high response for a pedestrian and tree trunks in Fig. 7.2 (a). These regions are distinct from their surroundings and extend vertically. In addition to the pedestrian only the tree trunk laying beneath the blue line is detected. The responses for other instances such as small trees, road and grass are weaker than those for pedestrians. This illustration is representative of the results obtained when selecting a cue detection threshold. The threshold selected from a large number of evaluations was the same. Fig.7.2 illustrates the good localisation of the MLR cue detection method. The MLR response on each vertical line in (a) is plotted in (b) in the corresponding colour. The profile plot from left to right corresponds to a traversal of the corresponding line in (a) from bottom to top. Further evaluation is presented below.

The Precision (P), Recall (R) and F-score (F) metrics were considered to evaluate the performance of the cue detector. These measures were defined as shown in Equations 7.1, 7.2 and 7.3:

$$P = \frac{t_p}{(t_p + f_p)} \tag{7.1}$$

$$R = \frac{t_p}{\left(t_p + f_n\right)} \tag{7.2}$$

$$F = 2 \times P \times R/(P + R) \tag{7.3}$$

The precision and recall metrics are defined in terms of the following parameters:

t_p (true positive): is the number of correct detections that occur when a pedestrian has been detected.

f_p (false positive): is the number of wrong detections obtained that arise when the cue detector detects some part of an image that is not a pedestrian.

f_n (false negative): is the number of pedestrians present in a scene that are not detected.

The precision metric shows the proportion of true positives with respect to all positive responses. Recall relates to the proportion of true responses. Recall rate is a popular detection criterion that can be used to evaluate most practical pedestrian detection applications. *F*-score is a weighted geometric average of precision and recall, which provides an overall measure of performance.

A Receiver Operating Characteristic (ROC) curve was used to illustrate the performance of pedestrian detection systems. The ROC was originally defined as True Positive Rate (TPR) against the False Positive Rate (FPR) as y and x-axes respectively at various threshold values. The TPR is also known as sensitivity and the FPR can be calculated as (1 – specificity). The TPR (sensitivity) defines the proportion of positives that are correctly identified as the persons present in a scene among all persons available during the evaluation. The FPR (specificity) measures the pro-

portion of negatives that are correctly identified. A test result is negative when the person is not present during the evaluation. The TPR or sensitivity, specificity and FPR were defined as:

$$\text{Sensitivity} = t_p/(t_p + f_n) \tag{7.4}$$

$$\text{Specificity} = t_n/(t_n + f_p) \tag{7.5}$$

Where: t_n defines the objects that are not persons and the test is negative.

$$\text{FPR} = f_p/(f_p + t_n) \tag{7.6}$$

Where: FPR is the proportion of false positives (f_p) with respect to false positive count (f_p) and true negative count (t_n).

Dollar et al. [DOL12] defined a hit or a miss rate to illustrate the performance of seven pedestrian detectors. They plotted miss rate or False Negative Rate (FNR) versus False Positives Per Image (FPPI) on a log-log scale and used log-average miss rate at 1 FPPI as a common reference point to summarize the performance of the pedestrian detectors as threshold values were varied. The Miss rate (FNR) was defined as the proportion of negative tests among people present in a scene and hit rate refers to true detection rate. Dollar et al. [DOL12] claimed that log-log plot is preferred to precision - recall curves for tasks such as automotive and pedestrian detection since there is an upper limit on the FPPI rate.

7.3.3 Pedestrian Cue Detection

The parameter values and the results for a series of experiments to evaluate cue detection on the dataset shown in Table 7.2 are shown in Table 7.3. The experimental conditions are described in Section 7.3.1 and the results given in terms of the Precision (*P)*, Recall (*R)* and F-score (*F)* rates.

Table 7.3. Cue detection rates.

Mask dimensions			Mask1(gap)			Mask 2 (no gap)		
$A(w_A, h_A)$	$B(w_B, h_B)$	gap	*P* %	*R* %	*F* %	*P* %	*R* %	*F* %
(12,12)	(12,24)	4	78.7	80.2	79.5	72.4	76.3	74.4
(24,12)	(24,24)	4	79.2	81.2	80.2	72.6	77.9	75.3
(24,12)	(24,24)	8	82.1	84.2	83.2	78.4	80.6	79.5
(36,24)	(36,48)	8	85.3	86.8	86	82	83	82.5
(36,24)	(36,48)	12	85.8	88.2	87	82.6	84.4	83.5
(48,36)	(48,72)	16	92.8	94.8	93.8	89.8	93.7	91.8

It can be seen from Table 7.3 that there is a gradual improvement in the detection rate as the gap and region size increase together. Further, there is a small increase in the detection rate as the region size increases with a given gap size. The difference in recall rate between a mask with a gap and a mask without a gap for the different mask dimensions is less than 4 on average which is a small and not statistically significant. The results demonstrate that the size of the central, *B,* and flanking regions, *A,* over the range considered is not critical to detection performance. The application of the smaller mask will be more rapid. There is a factor of approximately 300 in mask area between the smallest and largest mask. This is a major difference in computational burden. It is also observed that the best cue detection results were obtained using the masks with the largest size of regions, as shown in the last row of the table. A relatively high recall rate of 94.8% for mask 1 which is slightly better than the recall rate of 93.7% as achieved for mask 2 for masks of the same area. This difference of 1.1 between the two masks is not significant. The dimensions of mask 1 appearing in the last row of Table 7.3 were used with the interpretation systems presented in Sections 7.5 and 7.6. The pedestrian detection rate of 94.8% for mask 1 and 93.7% for mask 2 were due to a failure to accurately detect a few people as shown by an example in Fig. 7.5 where

mask 1 The false detection of a tree using mask 2 is illustrated in Fig. 7.6 (b). The cases where cue detection fails, as illustrated in Fig. 7.5, are discussed.

Masks with the largest size of regions, as described in the last row of Table 7.3, are used throughout for the results reported in Figs. 7.3 to 7.6. The experimental conditions and the dataset are described in subsections 7.3.1 and 7.3.2, respectively. It is important that the cue detector locates pedestrians correctly to strengthen the effectiveness of the interpretation system, even if there is a high degree of occlusion. The results of cue detection for selected images are marked by red dots in Figs. 7.3, 7.5 and 7.6 and by a blue dot for the person on the right in Fig. 7.5 (d), to differentiate it from the red bag on the pedestrian's back. The black dots were used to obscure the facial features of the pedestrians as agreed in the consent form signed by the participant pedestrians.

Each image in Fig. 7.3 (a) – (c) shows a single pedestrian, in each case with lower level of lighting, more than occurs in normal variation of light level and with a moderately complex background. Fig. 7.3 (d), Fig. 7.3 (e) and Fig.7.3 (f) show groups of two pedestrians with various levels of crowding and complexity of foreground and backgrounds composition. Fig. 7.3 (d) shows two separate pedestrians, Fig. 7.3 (e) a group of two pedestrians that are relatively close and Fig. 7.3 (f) a group of two pedestrians where one pedestrian is partially occluded by the other. Each image in Fig. 7.3 (g) – (i) shows a group of three pedestrians with varying degrees of separation and background complexity. Fig. 7.3 (g) and (i) show a group of three pedestrians in similar poses, one walking in a different direction. Fig. 7.3 (j) shows a group of four pedestrians with relatively different poses and Fig. 7.3 (k) shows four pedestrians with similar poses. Fig. 7.3 (l)

shows five pedestrians, with two isolated pedestrians and a group of three pedestrians with one pedestrian partially occluding the group of three in the middle of the image.

Fig. 7.3. Cue detection for pedestrians with various degrees of crowding and in various poses: (a), (b) and (c) pedestrians alone; (d) two pedestrians at a distance; (e) two relatively close pedestrians; (f) partial occlusion between two pedestrians; (g), (h) and (i) show groups of three pedestrians; (j) and (k) a group of four well spaced pedestrians and (l) a group of five pedestrians in a crowded scene with partial occlusion.

Fig. 7.3 (a) – (l) shows correct detection in each case with no false positive or false negative error. In each case a pedestrian is detected once only. The clustered MLR profile responses and selection of cues for the images shown in Fig. 7.3 (c) and (e) and their clustered cue reference points are shown in red after Thresholding in Fig. 7.4.

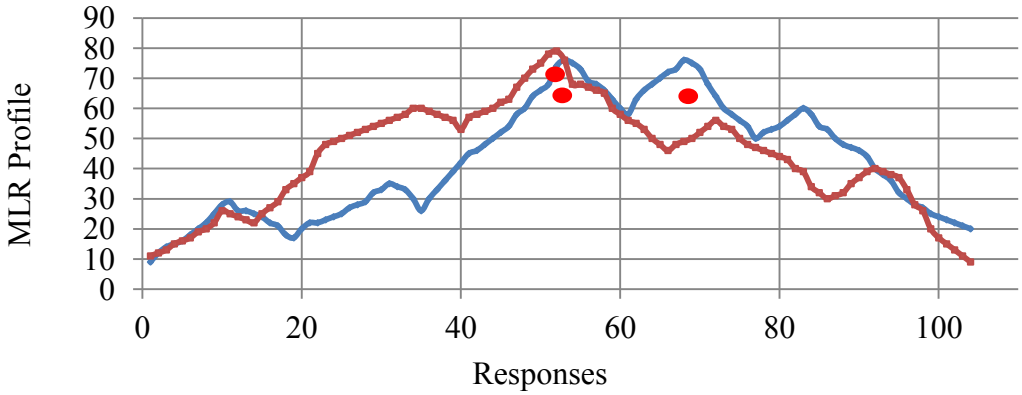


Fig. 7.4. The final MLR profile responses and selection of cues using pedestrian detector along the vertical paths of the images shown in Fig. 7.3 (c) and (e), with plots in red and blue, respectively. The cue points are shown by red dots.

The MRL profile responses and the cue points shown in Fig. 7.4 justify that the cue detection algorithm and the clustering process were appropriate to identify the pedestrian cues. Complex situations where the cue detection using mask 1 does fail are shown in Fig. 7.5.

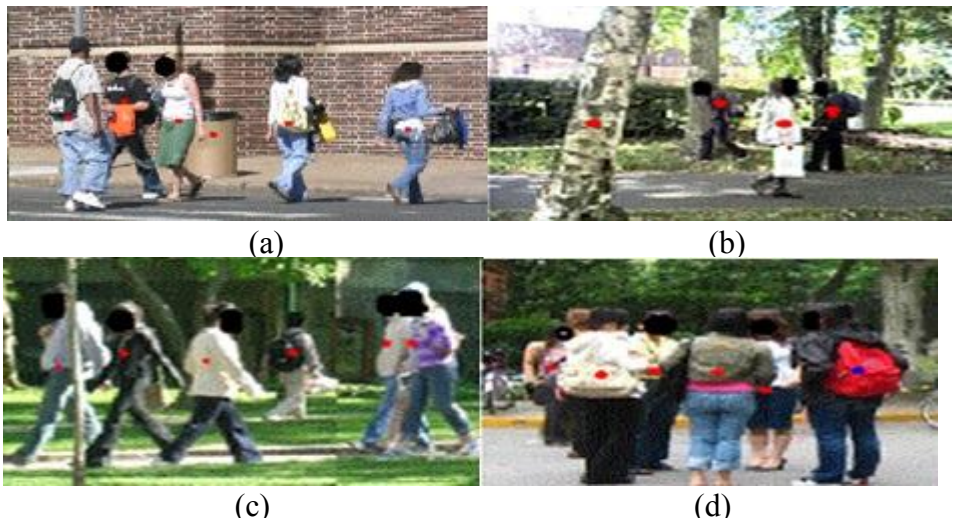


Fig. 7.5. Failure conditions of cue detection using mask 1 (a) waste bin detected,(b) tree detected, (c) and (d) almost completely occluded pedestrian not detected.

In Fig. 7.5 (a) a rubbish bin with dimensions similar to the body of a child is detected. A pedestrian detector might respond to a tree object which is a false positive that can usually be eliminated in the interpretation phase. In Fig. 7.5 (b), a tree is detected. A tree is occasionally detected because its dimensions satisfy the requirements of the pedestrian cue detector. The detection of the tree in Fig. 7.5 (b) is a significant false positive. The detection of the pedestrians that are substantially occluded in Fig. 7.5 (c) is good and that in Fig. 7.5 (d) is especially good. In Fig. 7.5 (c) an almost completely occluded pedestrian in the background is not detected and in Fig. 7.5 (d) a highly occluded person in the foreground in a group of three pedestrians with only a head visible is not detected. It is considered unreasonable to expect to detect either of these pedestrians and these cases are not considered serious failings. Further, Fig. 7.5 (d) shows the correct detection of one pedestrian who is largely occluded by two neighbouring pedestrians.

Fig. 7.6 shows cue detection results for two pedestrians using mask 1 in (a) and mask 2 in (b).

(a) (b)

Fig. 7.6. Cue detection results using: (a) mask 1 and (b) mask 2.

The benefit of using mask 1 is that there is no false positive result for the tree in the image. Fig 7.6 (b) shows an incorrect detection of a tree. The importance in this figure is to identify all pedestrians sought in the image without losing any pedestrian.

7.3.4 Cue Detection Rate

The dataset used to assess cue detection rate are described in subsection 7.3.2 and the experimental conditions are described in subsection 7.3.1. The experimental results of cue detection using the masks defined in the last row of Table 7.3 are summarised in Fig. 7.7, using the ROC curves, which show the true positive rate (sensitivity) on the y-axis plotted against the false positive rate (1 – specificity) on the x-axis at various threshold values. Each point on the ROC curve represents a pair of (sensitivity, 1-specificity) or (TPR, FPR) corresponding to a particular threshold value. Sensitivity is inversely related with specificity in the sense that sensitivity increases as specificity decreases across various threshold values. A test with 100% sensitivity correctly identifies all persons present in a scene. A test with 80% sensitivity detects 80% of persons present (t_p) but 20% of persons are undetected (f_n). On the other hand, a test with 100% specificity correctly identifies all the non-person objects. A test with 80% specificity accurately reports 80% of non-persons as a negative result (t_n) but 20% non-persons are falsely identified as a positive result (f_p). The error bars in Fig. 7.7 are shown for 1 standard error of the mean. The threshold values were selected experimentally as justified in Fig. 7.2. The low hysteresis threshold was varied on detection confidence from 5 to 40 and high hysteresis thresholds from 25 to 60, in steps of 5, in each case, to generate the ROC curves.

The two ROC curves are best partially separable at unit standard error of the mean which is nearly about 56% confidence level but not at higher levels of confidence; the difference between the two curves of the masks with a gap and without a gap is not significant. There is a small pref-

erence for the mask with a gap which is used in the interpretation system. The cue detector using a mask with a gap has more accuracy than the cue detector using a mask without a gap.

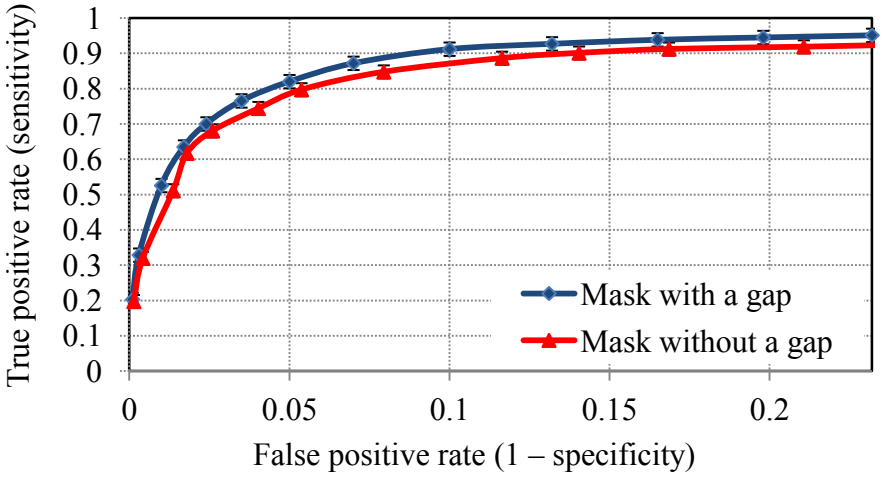


Fig. 7.7. ROC curve for pedestrian detection on the test sequences collected at the University of Birmingham. The vertical error bars represent one standard error of the mean of a set of measurements.

There is a trade-off between True Positive Rate (TPR) and False Positive Rate (FPR): to achieve a higher pedestrian detection rate will usually incur a higher false positive rate as a side effect. A threshold is selected to balance TPR (sensitivity) and FPR (1 – specificity). This balance was controlled through selecting appropriate threshold values where any increase in sensitivity will be accompanied by a decrease in specificity.

The cue detection results shown in Fig. 7.7 show that the cue detector using the mask with a gap presents a true positive rate of 0.95 with a false positive rate of 0.20 per image, while the cue detector using the mask without a gap presents a true positive rate of 0.92 with a false positive rate of 0.22 per image.

The ROC curves in Fig. 7.7 display all possible operating points, where it is possible to identify an optimal threshold value for correctly identifying pedestrians. For optimum operating conditions, it is important to maintain a high true positive rate. The point nearest to the upper left corner of the ROC curve is the optimal location which indicates to a high TPR value and a low FPR value. The optimal threshold point, on the ROC curve, shows an appropriate balance between the maximum TPR and the minimum FPR. Fig. 7.7 shows that the optimal operating point for cue detection using the mask with a gap occurs at a TPR of 0.82 with a FPR of 0.05. An appropriate balance for cue detection using the mask without a gap is likely to be found with a FPR of about 0.06 and a TPR count of about 0.8 or less.

It is important that the number of false positive detections is not large otherwise efficiency of target identification will be impaired. It is more important to maximize TPR (minimise false negatives) than to maximize specificity (minimise false positives). This condition is required because it is important to detect all pedestrians sought in an image without losing any pedestrian. If objects that are not pedestrians are detected then they can be eliminated during interpretation, therefore a slightly heightened false positive rate is not a serious problem, where there is a priority that true positive rate should be high. Overall, the cue detector approach of both masks results in high true positive rates.

The cue detector method was also demonstrated for the detection of humans at very high resolution (1920×1080 pixels) using full-length outdoor town centre sequence [BEN11]. In this sequence the majority of pedestrians are walking and looking in their direction of travel. This town centre sequence has 473412 images. This cue detector was applied every one hundredth image in

the sequence, a total of 4300 frames. Using the evaluation procedure described in [ESS07], the results of the cue detector using the town centre sequence with the masks defined in the last row of Table 7.3 are presented in Fig. 7.8, using the rate of recall on the y-axis plotted against the FPPI on the x-axis. FPPI evaluation metric is appropriate for evaluating the performance on full images [TRA08].

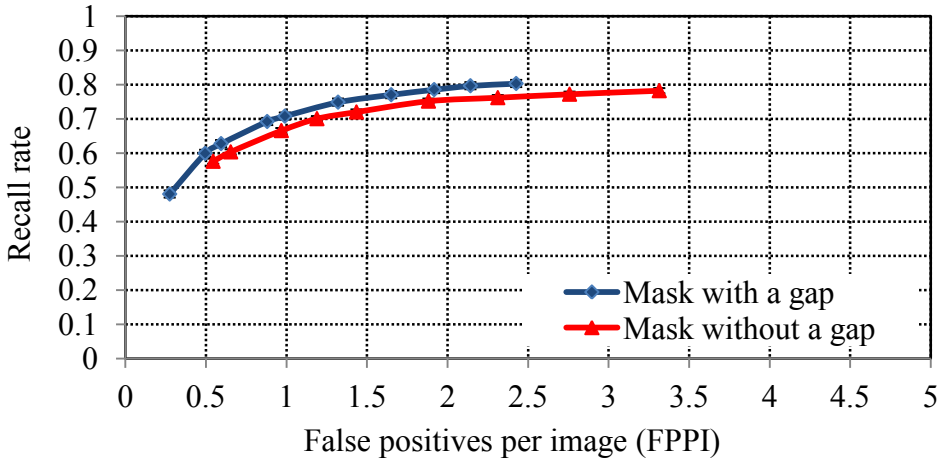


Fig. 7.8. False positives per image as the evaluation metric for pedestrian detection on a town centre sequence for the first 4300 frames. The vertical error bars represent one standard error of the mean of a set of measurements.

The detection results shown in Fig. 7.8 support that the cue detector achieved a high significant performance of pedestrian detection on a high resolution dataset, demonstrating its good generalization capabilities. Table 7.4 shows a comparison of the results of the pedestrian detector reported here with the results of other pedestrian detection approaches using data sets of other works of pedestrian approaches.

The configuration parameters of the SVM classifier implemented in OpenCV [CHA11] used to identify the pedestrian cues are as follows: The number of training images of pedestrians was

700. The number of test image size was 300. The dataset was divided into ten subsets; nine subsets were used for training the model and one subset was used for testing, so, the SVM was executed 10 times to allocate the best parameters over the subsets of training samples. The learned parameters considered the best if the test error of the testing model is minimal. The stopping criteria were the number of iterations, 150, or the tolerance error of 0.001. The recall rate of correct identification of pedestrian cues using this SVM classifier was 88%.

A review of the work reported by Dollar et al. [DOL12] is presented in the literature review at the end of subsection 2.3.1.1. There is an extensive evaluation and a comparison of a diverse set of 16 pedestrian detectors under various scenarios and for multiple datasets. These pedestrian detection approaches are each representative of various lines of research and favorable in terms of reported performance. Dollar et al. further described a set of evaluation measures that deal with extreme cases such as severe occlusion and poor illumination. Further, there is a statistical analysis of the significance of the results, where the detection rates were explored under varying levels of scale and occlusion and on clearly visible pedestrians. Dollar et al. [DOL12] adopted a log-log curve that plotted the miss rate against FFPI to illustrate performance of pedestrian detectors. The justification is to focus on the very low values of miss and false positive rates which would not be easy to visualize in a standard ROC curve. In the experiments reported in [DOL12], the performance was evaluated on publicly available datasets. The datasets consist of pedestrians who are between 90 and 210 pixels, 180 and 390 pixels and between 96 and100 pixels in height.

The performance of the pedestrian cue detector reported in this Thesis is compared against results of most recent promising pedestrian detection approaches in the literature which have best report-

ed performance. These reported ppedestrian detection approaches are fully reviewed in the literature review at the end of subsection 2.3.1.1. These previously pedestrian detectors were reported by Viola and Jones [VIO04], Dalal and Triggs [DAL05], Schwartz et al. [SCH09], Tang et al. [TAN12], Lim et al. [LIM13], Ouyang and Wang [OUY13], Tian et al. [TIA15] and Nguyen et al. [NGU15]. Another pedestrian detector presented here for the comparison is the work reported by Wang et al. [WAN07]. A promising pedestrian detection approach reviewed in subsection 2.3.1.1 is the work reported by Lim et al. [LIM13]. The results reported by Lim et al. are compared against recent set of results as presented in Table 2.2. Recall rate is a measurement commonly used for evaluating the performance of pedestrian detection. The ideas of Dollar et al. for presenting the results of pedestrian detectors was used to evaluate the pedestrian detectors by using the FPPI metric; where FPPI measure takes into account the number of windows presented to the classifier. The log-average miss rate at false positives per window was used to evaluate the performance of the pedestrian detectors which was computed by averaging the miss rate at nine FPPI rates that are evenly spaced in the log-space [DOL12].

Table 7.4 shows the recall and log-average miss rates for the cue detector presented in this Thesis compared to other pedestrian detectors, which are reviewed in subsection 2.3.1.1. The detection and log-average miss rates for the existing pedestrian approaches were reported by the authors, as reported in subsection 2.3.1.1. The pedestrian detection values reported for [WAN07] and [DAL05] have been estimated form images in those papers as they are not reported by the authors. This is a crude process and may be inaccurate.

The TUD person set was used as a benchmark for the pedestrian detection approach in [WAN07] and [TAN12], H3D [BOU09] and PASCAL VOC 2007-2009 [EVE10] was used as a benchmark for the pedestrian detection approach in [NGU15] and the INRIA person dataset was used as a benchmark for the other pedestrian detection approaches in Table 7.4. Subsection 2.3.1.1 and Table 2.3 in the literature review provides further details about the training and test datasets for each previously pedestrian detection method reported in Table 7.4. The work reported by Wang et al. [WAN07] used the TUD person dataset. The results of the pedestrian detection method presented in this Thesis in Table 7.4 used the datasets of reported works of Dalal and Triggs [DAL05] and Wang et al. [WAN07]. This reported pedestrian cue detector used a set of 320 images of pedestrians from the TUD person dataset [WAN07] and a set of 510 images of pedestrians from the INRIA person dataset [DAL05], as described in the last two rows of Table 7.4.

Table 7.4. A comparison between the proposed pedestrian cue detector and other pedestrian detection approaches.

Pedestrian detection method	Training data	Performance metrics	
		Recall rate (%)	Log-average miss rate
Lim et al. [LIM13]	INRIA dataset and PASCAL 2007	95	15%
Nguyen et al. [NGU15]	PASCAL VOC 2007-2009	95	67%
Tian et al. [TIA15]	INRIA person dataset	93	26%
Ouyang and Wang [OUY13]	INRIA person dataset	92	39%
Proposed pedestrian detector	TUD person dataset	92	43%
Schwartz et al. [SCH09]	INRIA person dataset	92	62%
Proposed pedestrian detector	INRIA dataset	91	44%
Tang et al. [TAN12]	TUD person dataset	90	57%
Dalal and Triggs [DAL05]	INRIA person dataset	89	68%
Wang et al. [WAN07]	TUD person dataset	87	65%
Viola and Jones [VIO04]	INRIA person dataset	80	95%

Table 7.4 compares the evaluation strategy and performance of the pedestrian cue detector presented in this Thesis to nine pedestrian detection methods described in the literature review. It is clear that the cue detector reported here performs considerably better than some state-of-the-art detectors such as Viola and Jones [VIO04], Dalal and Triggs [DAL05], Wang et al. [WAN07] and Schwartz et al. [SCH09] and achieved similar recall rates to the methods reported by Tang et al. [TAN12]. Further, the pedestrian cue detector reported a detection rate that is comparable to the detection rate reported by Ouyang and Wang [OUY13]. However, the pedestrian cue detector reported here did not perform as well as that reported by Lim et al. [LIM13], Nguyen et al. [NGU15] and Tian et al. [TIA15]. The difference is relatively small compared to Tian et al. [TIA15] for recall rate and significantly less than the recall rates reported by Lim et al. [LIM13] and Nguyen et al. [NGU15]. Further, the pedestrian cue detector reported here achieved miss rate that is significantly less than the miss rates reported by Viola and Jones [VIO04], Dalal and Triggs [DAL05], Wang et al. [WAN07] and Schwartz et al. [SCH09]. The difference in miss rate is relatively small compared to that reported by Ouyang and Wang [OUY13]. However, the pedestrian cue detector here reported a miss rate that is significantly larger than the miss rates reported by Lim et al. [LIM13] and Tian et al. [TIA15].

7.3.5 Vehicle Cue Detection

An adaptation of the pedestrian cue detector to detect vehicles was made by modifying the dimension of the operator mask and way of scanning images. The search strategy and the mask shape changes are explained in subsections 3.3.2 and 3.4, respectively. The dimensions of the mask, shown in Fig. 3.2 (b) for vehicle detection were set to as width, $w_B = 108$, width, $w_A = 108$, height, $h_B = 54$, and height, h_A=27. The low and high hysteresis thresholds were not

changed. This cue detector was applied to detect vehicles. The dataset employed to evaluate the vehicle detector includes 1000 images with a variety of vehicle body shapes such as: hatchback cars, minivans, compact cars, city cars and vans. The results of cue detection on images containing vehicles and a pedestrian are shown in Fig. 7.9 with their detected cues shown by small red dots. Their MLR profile responses are shown in Fig. 7.10.

(a) (b)

Fig. 7.9. Cue detection results for vehicle images on which vehicle detector is considered, the annotated rows and columns corresponding to the MLR profile responses for vehicle and pedestrian detectors as shown in Fig. 7.10.

Fig. 7.9 illustrates that a change in the geometry of the mask and the way that it was scanned across the image allow the cue detector to detect cue points within typical images of vehicles and to distinguish these potential targets from other objects in the background and foreground. To detect vehicles, the image was scanned row first. Fig. 7.9 shows also both horizontal and vertical scans using vehicle and pedestrian detectors respectively as shown in Fig. 7.10 by the MLR profiles for vehicle and pedestrian detectors.

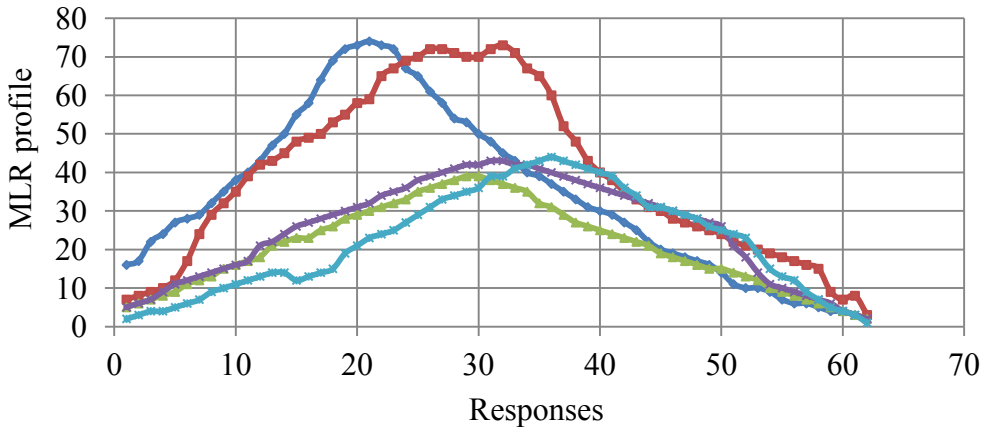


Fig. 7.10. MLR responses using the vehicle and pedestrian detectors for the images in Fig. 7.9 (a) and (b): the blue and red responses correspond to the annotated blue and red rows in Fig. 7.9 (a) and (b) using the vehicle detector, the green line shows the profile responses for the pedestrian in Fig. 7.9 (b) using the vehicle detector and the purple and light blue lines show the profile responses for the purple and light green lines for a bus and the left car in Fig. 7.9 (a) using the pedestrian detector.

The blue and red lines in Fig. 7.10 show the responses, detected by the vehicle detector, corresponding to the annotated blue and red rows in Fig. 7.9 (a) and (b). It can be seen from Fig. 7.10 that the response to the relevant detectors is larger than that for detectors designed for different objects.

The dataset used to evaluate the vehicle detector is described in the previous paragraph. The detection results for vehicle detector in terms of Precision (P), Recall (R) and F-score (F) rates are reported in Table 7.5.

Table 7.5. Vehicle cue detection rates.

Target	Mask dimension (pixel)				Detection (%)		
	w_A	h_A	w_B	h_B	P %	R %	F %
Vehicle	108	27	108	54	90.6	91.6	91.1

7.4 Component Axes Detection

The detection of multiple and composite axes that represent complex objects or a combination of objects is presented in this section. The composite axes were formed by linking axis component. The number of images representing different object types used to evaluate the detection of complex axes is shown in Table 7.6. The pedestrians can be alone and in groups with various degrees of occlusion with pedestrians of various sizes with simple and complex backgrounds.

Table 7.6. Number of images containing different object types.

Image	Pedestrians alone	Pedestrians pushing pushchairs	Pedestrians pushing bicycles	Persons riding bicycles
Number of images	700	200	200	200

The axis detection for isolated pedestrians and for pedestrians with a pushchair or a bicycle was investigated to generate the boundary points on these objects on which image interpretation is based while the key point generation method of pedestrian cue detector was employed to identify the key points around the pedestrians alone.

7.4.1 Experimental Settings for the Axes Detection

The regions *A* and *B* of the interpolated filter mask, shown in Fig. 3.2 (a), were defined to have a width, $w_B = 18$ and height, $h_B = 16$ pixels, for the first scale. Twelve local responses using the interpolated cue detector were computed at angular intervals of 30° about the horizontal and vertical at each of 4 levels of resolution by combining the responses for the detector when aligned with the y-axis and the x-axis. The high and low thresholds were set to 40 and 25, respectively.

7.4.2 Determination of an Object Type

In Section 4.4 a way to determine object type for complex objects or a combination of objects detected by the axis cue detector was described. Here, four object types are considered: a pedestrian alone, a pedestrian pushing a pushchair, a pedestrian pushing a bicycle and a person riding a bicycle. Clearly many object types are possible. Experimental results concerned to investigate the ability to identify object type are presented.

The data used to identify the nature of the cues is the entropy and the energy of the axes responses as introduced in Section 4.4. The type of a detected object was identified by forest tree classifier. Forest of trees classifier and why this classifier was chosen, as opposed to, for example, K-means clustering were explained in Section 4.4. The training and evaluation features used with the forest of trees classifier consists of the entropy and the energy of the axis points for 800 and 600 images, respectively. The number of each object type present in the images is described in Table 7.7. The data considered for this part were gathered from the University of Birmingham and from the Internet as described in Section 7.2 and subsection 7.3.2. Some of these images are from INRIA and TUD person datasets which used to evaluate previously reported pedestrian detection methods of Dalal and Triggs [DAL05] and Wang et al. [WAN07], respectively.

Table 7.7. Dataset for training and evaluating the forest tree classifier.

Type of objects	Pedestrians alone	Pedestrians pushing pushchairs	Pedestrians pushing bicycles	Persons riding bicycles
Training dataset	200	200	200	200
Evaluation dataset	150	150	150	150

7.4.2.1 Parameter Setting for the Forest of Trees Structure

The highest depth of the random forest of trees classifier is 15. The number of features considered to identify a split at each tree node is 2, which are equally weighted. The number of trees in the forest is 10. The combination of results with this number of trees in the random forest of trees classifier of 800 sample images for training and 600 sample images for testing has the potential to improve the outcome and therefore might improve performance. The number of degrees of freedom for this number of samples is approximately 100 sample images per degree of freedom for the training data set and 75 samples per degree of freedom for the test data set. The termination criterion is the maximum number of trees permitted. Table 7.8 implicitly contains the performance of the random forest classifier for detecting pedestrians alone and pedestrians associated with pushchairs or bicycles.

The confusion matrix obtained from the forest of trees classifier for the detection of different object types is shown in Table. 7.8. The class labels represent persons alone, persons pushing pushchairs, persons pushing bicycles and persons riding bicycles. The true class labels are listed along the x-axis and the forest of trees class predictions are shown along the y-axis. The correct classifications are shown along the first diagonal and all other entries show misclassifications. The bottom right cell shows the overall accuracy. The numbers in brown show the proportion of miss classifications with respect to hit and miss responses. The numbers in blue show the proportion of hit classifications with respect to hit and miss responses. The percentage numbers in black in the main diagonal are the proportion of hit responses with respect to all images and the percentage numbers in black in the other cells show the proportion of miss classifications with respect to all images. The hit rate seems relatively rather low, perhaps because the images of pedestrians alone

and associated with pushchairs or bicycles are highly varied. However, this hit rate is acceptable with approximately 100 samples per degree of freedom for the training data set and 75 samples per degree of freedom for the evaluation data set. The bold numbers in black in the main diagonal show the true classification and the bold numbers in black in the other cells show the number of miss responses.

Table 7.8. Confusion matrix for all classes and all attributes.

		Target class				
		Pedestrians	Pedestrians Pushing pushchairs	Pedestrians pushing bicycles	persons riding bicycles	
Output class	Pedestrians	**171** 21.4%	**14** 1.7%	**11** 1.3%	**10** 1.2%	83% 17.5%
	Pedestrians pushing pushchairs	**10** 1.2%	**168** 21%	**11** 1.3%	**14** 1.75%	82% 17%
	Pedestrians pushing bicycles	**11** 1.3%	**9** 1.1%	**166** 20.8%	**13** 1.6%	83.4% 16.5%
	Pedestrians riding bicycles	**8** 1%	**9** 1.1%	**12** 1.5%	**163** 20.4%	85% 14.5%
		85.5% 14.5%	84% 16%	83% 17%	81.5% 18.5%	83.4% 16.4%

A SVM based on LibSVM [CHA11] was also used to identify the type of an object using the same measures and the data of the FT classifier. The parameters used in the SVM classifier are illustrated in subsection 7.3.3. The average rates of correct classification using the FT and SVM classifiers for all cue types were respectively 83.4% and 85.6% for the augmented cue matrix having 800 samples in the training set and 600 samples in the test set. This is approximately 100

images per degree of freedom for the training data set and 75 images per degree of freedom for the set data set, which is a relatively good sample size, given that the degree of freedom for cue identification is identified by the number of object types plus parameters assuming that the parameters are independent [PAN08].

7.4.3 Interpolated Cue Detection Results

The interpolated cue detector was used to identify the cue points within pedestrians alone and a combination of pedestrians with pushchairs or bicycles, whereas the non-interpolated cue detector was used to identify the cue points within pedestrians alone. The data used to establish the performance of cue identification for the interpolated cue detection results are described in Section 7.4 and the experimental parameter values of the interpolated mask are described in subsection 7.4.1.

Examples of cue detection and identifying the identified cue object type using the interpolated cue detector for non-crowded scenes for people in different contexts are shown in Fig. 7.11 (a) – (c) and for people combined with other objects in Fig. 7.11 (d) – (f). The detected cue points are shown in red in each case in Fig. 7.11 (a) – (e) and in blue in Fig. 7.11 (f) to distinguish it from the colour of the pushchair.

(a) (b) (c)

(d) (e) (f)

Fig. 7.11. Correctly detected pedestrian cues: (a) two partially occluded pedestrians; (b) two pedestrians; (c) four pedestrians separated by a small distance; (d) pedestrian pushing a bicycle; (e) person riding a bicycle and (f) pedestrian pushing a pushchair.

In Fig. 7.11 (a) there is a significant variation in lighting with two pedestrians present but the pedestrians are in low contrast to the background. Fig. 7.11 (b) shows an image of two pedestrians with a number of trees present in the background. Fig. 7.11 (c) shows a group of four pedestrians and the area around the pedestrians has a relatively low illumination. The group of four pedestrians in Fig. 7.11 (c) can be considered as people alone because they are gaps around a large part of their outline. Fig. 7.11 (d) – (f) show a variety of scenes for pedestrians combined with a variety of other objects with good, uniform lighting in each case.

The cue detection results for the images in Fig. 7.11 are based on the interpolated cue detector and the cue detection results reported in Fig. 7.3 were based on the non-interpolated cue detector. The detection results of the interpolated cue and pedestrian cue detectors might differ in some cases because the interpolated cue detector used an extension strategy to pedestrian cue detector

as described in Chapter 4. A comparison of performance of the detection rates between the axis detector and pedestrian cue detector on the same set of pedestrian images is shown in Table 7.9.

Two situations where the interpolated cue detector failed are shown in Fig. 7.12.

(a) (b)

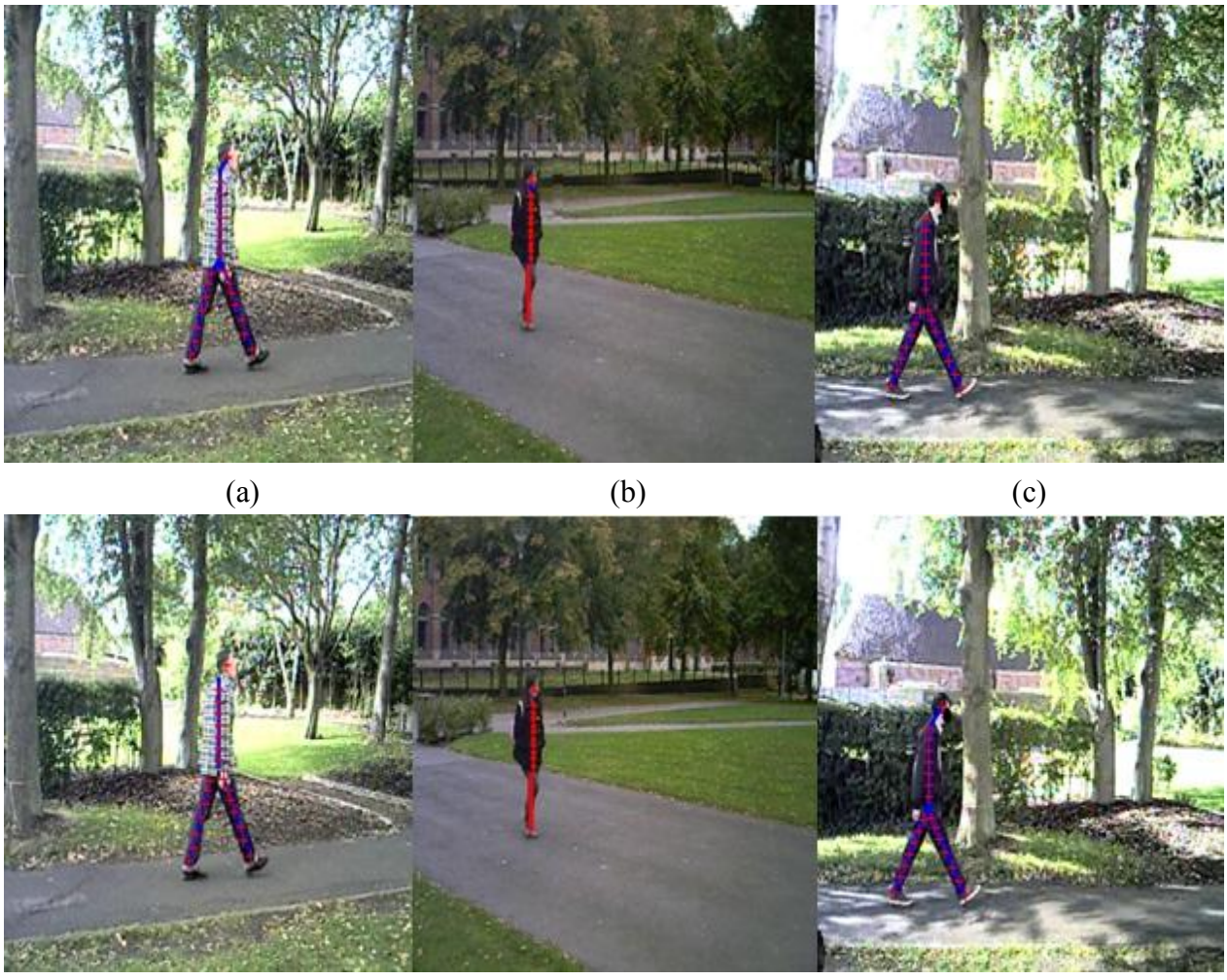
Fig. 7.12. Failure of cue detection: (a) two almost completely occluded pedestrians out of four are not detected and (b) a tree is incorrectly detected.

In Fig. 7.12 (a), the two pedestrians in the foreground each almost completely occlude a person in the background who is not detected. This is considered reasonable given the severe degree of occlusion. In Fig. 7.12 (b) a tree with dimensions similar to a pedestrian is detected. This is a rare event but a significant false positive.

7.4.4 Results of Axes Detection

The value of using component axes to identify key points on the object boundary for pedestrians alone and pedestrians with a pushchair or a bicycle was introduced in Section 4.6.

Fig. 7.13 (a), (b) and (c) show the major visible component axes for pedestrians, as a solid blue line, as defined by the symmetry of the objects and their component parts. The red crosses mark axis points. Fig. 7.13 (d), (e) and (f) show the corresponding composite axes to Fig. 7.13 (a), (b) and (c), respectively.



(a) (b) (c)

(d) (e) (f)

Fig. 7.13 Detected axes for pedestrians: (a), (b) and (c) for each component and (d), (e) and (f) composite axes for the pedestrians shown in (a), (b) and (c), respectively.

The component axes in the bottom row of Fig. 7.13 are joined end to end to the closest axes of the pedestrian axes to form a composite axis. The composite axis representation was designed to

provide a good representation of the visible component axes for pedestrians joined together. The difference between the images in Fig. 7.13 (a) - (c) and the respective images in Fig. 7.13 (d) - (c) in respect to the length position and structure of the axes is small because the sub-axes of the pedestrian are very close to each other. It is observed from Fig. 7.13 that the local axes show local symmetry of the pedestrian images. Fig. 7.13 (a), Fig. 7.13 (b) and Fig. 17.3 (c) show that the number of major axes detected depends on the pose of each pedestrian.

Fig. 7.14 (a) – (d) show the major visible component axes for pedestrians, as a solid blue line with red crosses mark axis points. In Fig. 7.14 (a) and (b) the interpolation process was used to find the output responses of the MLR filter at various orientations, whilst in Fig. 7.14 (c) and (d) the result of the actual MLR filter was synthesised by a linear combination of the output responses at various orientations. The interpolation process was introduced in Section 4.3.

It is observed that the difference between the locations of the component axes identified on the pedestrian shown in Fig. 7.14 (a) and (c) and that shown in Fig. 7.14 (b) and (d) is small as visually judged and not statistically significant. This is in turn means that the difference between the result of interpolation and the result of the actual MLR operator at various orientations is not significant. Fig. 7.14 (a) and (b) confirm the validity and the appropriateness of interepolation to identify the axis points on the major component axes of pedestrians.

(a) (b)

(c) (d)

Fig. 7.14. Detected axes for pedestrians: (a) and (b) using interpolation to approximate the MLR filter at various orientations; (c) and (d) synthesising the output of the MLR filter at various orientations.

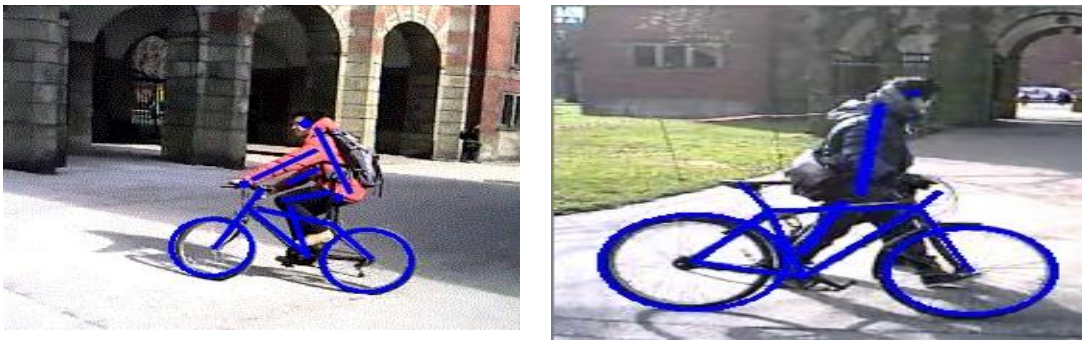
Fig. 7.15 (a) - (c) show the major component axes for pedestrians pushing pushchairs.

(a) (b) (c)

Fig. 7.15. Detected component axes for pedestrians pushing pushchairs.

The pushchairs vary in form and are not always symmetric, therefore, the detected axes of the pushchairs in Fig. 7.15 (a) – (c) are less than symmetric but they reflect the local axes of symmetry.

Fig. 7.16 (a) and (b) show the major axes for persons riding and pushing bicycles, respectively.



(a) (b)

Fig. 7.16. Detected axes: (a) a person riding a bicycle and (b) a pedestrian pushing a bicycle.

Variation in the number of axes as for different pushchair poses is shown in Fig.7.17 for pedestrians with pushchairs. This shows that the number of axes varies and in this case the process for locating key points, as described in Chapter 4, was used. In that chapter it was discussed that a variable number of axes does not matter as the axes serve only to help locate the boundary key points around the objects. For example, in Fig. 7.17 (a) the legs of the person are hidden beyond the pushchair and hence their axes are not detected. It is considered inappropriate to expect to detect the axes for occluded legs. The two arms of the pedestrian are completely occluded and hence their axes are not identified. It is not difficult to define a match when sometimes one arm and sometimes two arms are detected, as the main aim is to identify the key points along the boundary of the objects on which the matching algorithm is based. The strategy described in Sec-

tion 4.6 presented the importance of detecting the component axes of the objects to identify the key points along the boundary of these objects.

(a) (b)

Fig. 7.17. Variation of the number of component axes with pose.

The object model relates to the ESE and the key points identified along the axes that represent the object and were incorporated into this model. The method for key point detection and selection to form a model was introduced in subsections 4.6.2 and 6.3. A similar number of local axis points were selected along the axes detected for objects of the same type. The identified points of an object should be sufficient to describe the object even if some axes are or are not detected. For example, a pedestrian pushing a pushchair detected as a body with one or two arms, and axes for the pushchair is very different to a pedestrian detected as one body axis, no arms and axes for the pushchair has a different set of axes. Here the axis representation is different for different object types, but the same methodology was used to identify the axes for the different objects type. Also, the same methodology was used to identify the key points for the different objects types; with a consistent number of axis points selected on the axes of each object type regardless of the occlusion of some object axes. The axis representation and key point detection used the same methodology for pedestrians alone, pedestrians pushing pushchairs, pedestrians pushing bicycles and persons riding bicycles and they were cues for the same interpretation process.

The interpolated axes and cues for a combination of objects is fully evaluated in terms of the standard ROC curve and the metric measures of precision, recall and *F*- rates as shown below.

7.4.5 ROC Curves of Interpolated Cue Detector

This section evaluates the ability of the interpolated cue detector to identify cues for pedestrians alone and for pedestrians associated with pushchairs or bicycles. The dataset used is introduced in Table 7.6 and the nature of the pedestrian images is described in subsection 7.3.2. The mask and the experimental conditions are described in subsection 7.4.1. The ROC curves of Fig. 7.18 summarise the results for identifying pedestrians alone and pedestrians associated with push-chairs and bicycles using the interpolated cue detector. The results are presented as a plot of the TPR against the FPR with error bars for 1 standard error of the mean. The ROC curves were generated by varying the low and high hysteresis thresholds, from 5 to 40 and from 25 to 60 respectively and in steps of 5.

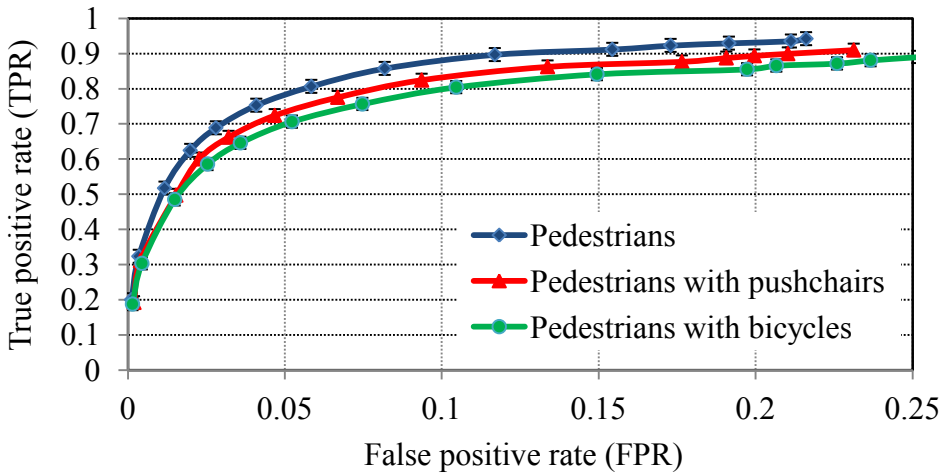


Fig. 7.18. ROC curves for pedestrian, pushchair and bicycle detection. The vertical error bars represent one standard error of the mean of a set of measurements.

The cue detection results shown in Fig. 7.18 show that the interpolated cue detector achieved a TPR of 0.94 with a FPR of 0.21 per image for detecting pedestrians alone and a TPR of 0.90 with a FPR of 0.23 per image for detecting pedestrians associated with pushchairs and a TPR of 0.89 with a FPR of 0.25 per image for detecting pedestrians associated with bicycles. Based on the cue detection results of Fig. 7.18, a true positive rate of 0.80 with a false positive rate of 0.06 for pedestrian detection are likely to represent an optimum operating point. The true positive rates of 0.78 and 0.76 and false positive rates of 0.07 and 0.08 are the optimum operating points for pedestrians associated with pushchairs and bicycles, respectively. These reported false positive rates for detecting pedestrians alone and pedestrians associated with pushchairs and bicycles at the optimal operating points are appropriate. At the optimum operating points a lower false positive rate was realised by the forest of trees classifier. If objects that are not the target objects are detected then they can be eliminated during the fitting process, therefore a slightly heightened false positive rate is not a serious problem. However, it is not desirable to have a large number of false positives since this takes time during the interpretation process.

7.4.6 Evaluation of Cue and Composite Axes Detector

The cue detection results and the performance for detecting composite axes for pedestrians alone and pedestrians combined with pushchairs and bicycles using the interpolated composite detector were evaluated in terms of precision (P), recall (R) and F- rates as shown in Table 7.9.

Table 7.9. *P*, *R* and *F* rates for detecting cues and axes for pedestrians alone and pedestrians associated with pushchairs and bicycles.

Criterion	Pedestrians alone		Pedestrians with pushchairs		Pedestrians with bicycles	
	Cues	Axes	Cues	Axes	Cues	Axes
P (%)	93.6	92	90.1	90	88.2	86.2
R (%)	94.2	94	91.8	90.6	90.0	88.1
F (%)	93.9	93	91.0	90.3	89.1	87.2

Table 7.9 shows that the interpolated cue detector reported recall rates for pedestrian and axes detection of 94.2% and 94%, respectively, and the recall rates for detecting cues and axes for pedestrians with a pushchair are 91.8% and 90.6%, respectively. Also, Table 7.9 shows that the recall rates for detecting cues and axes for pedestrians with a bicycle are 90.0% and 88.1%, respectively. Table 7.9 demonstrates that the detection of pedestrians alone and pedestrians associated with pushchairs and bicycles and their composite axes is good and reasonable. Table 7.3 reported an optimal recall rate of 94.8% for a set of 700 images representing 3098 persons, using the non-interpolated cue detector, compared to the recall rate of 94.2%, for pedestrian's detection, using the interpolated cue detector as shown in Table 7.9. These detection rates were reported for 1 standard error of the mean, making a difference in detection rate of 0.9 a small and not significant. This quantifies 68.27% of the values that lie within one standard error of the mean.

The axis detection results reported in Table 7.9 show that the axis detection method is appropriate for identifying the composite objects and the axes for people combined with other objects. A secondary goal of the axis detector was to identify the key points along the boundary of pedestrians with pushchairs and bicycles as described in Section 4.6.

Further, a secondary goal of the cue detector was to identify the key points along the boundary of the pedestrians. In subsection 3.5.2, where the generation of edge points on both radial and perpendicular search paths is described; the ratio of radial distance of the corresponding first and third key points from start of radial line to the key points on successive radial search paths was used to establish the need for a secondary search path. A set of values in the range of 0.1-1.0 and 1.7-3.5 pixels was applied to this ratio to identify the proper values for this ratio. Experimentally it was found that this ratio from each observation is nearly identical from the images of pedestrians considered and the choice of ratios of 0.5 and 2 were appropriate, although they were not critical. Further details on generating perpendicular search paths on an image of pedestrians with different values of ratios are described in subsections 3.5.1 and 3.5.2. In the peak detector strategy described in subsection 3.5.2.1 peak widths of 7 and 18 and peak heights of 23 and 70 pixels were evaluated by the peak detection algorithm. Typical values for w and h for the key point detection method introduced in subsections 3.5.2 and 4.6 were set to 7 and 23, respectively. The necessary or relevant peaks and valleys were detected by peak and valley hysteresis detectors as presented in subsection 3.5.2.2.

7.5 Fitting Results for the ESGM

Images with a complex background and foreground along with cases of people in isolation, in groups and in combination with other objects and vehicles are used in Fig 7.19 to 7.23 to illustrate the effectiveness of the ESGM. The ESGMs interpretations were created for pedestrians alone, pedestrians combined with other objects and five types of vehicles. The ESGM was produced as described in Section 5.4. The training and interpretation methods of the ESGM are described in Section 6.2. There are 18 modes in the models and the ESGM in all experiments ex-

plains 97.5% of the total variance. An ESGM for pedestrian interpretation was trained on a data-set of more than 200 images of pedestrians of various poses, sizes, with and without bags. The ESGM was evaluated on an independent set of 150 images. This ESGM was used in all the results reported in Figs. 7.19 – 7.21.

Fig. 7.19 shows the ESGM interpretation at the 6^{th} and 18^{th} iteration for a pedestrian in different poses. The key points are shown in green and blue at iterations 6 and 18 in Fig. 7.19 (a) and (b) respectively and the ESE curves are shown by a black line.

(a)

(b)

Fig. 7.19. Detection of points shown in (a) green at 6 iterations and (b) blue at 18 iterations.

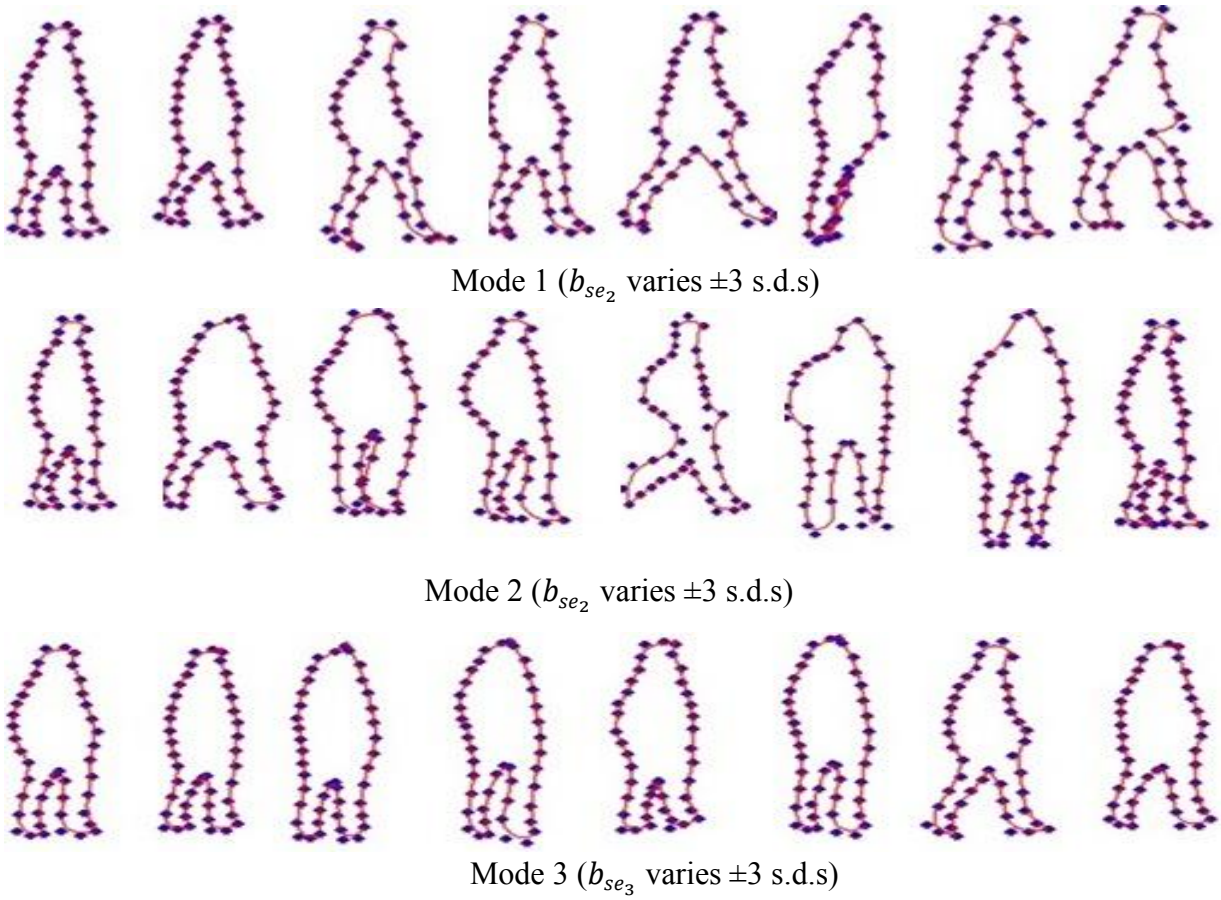
Fig. 7.19 (b) shows that a very small improvement is obtained after 18 iterations (shown in blue).

Fig. 7.20 illustrates the potential flexibility of ESGM in detecting points that locate the boundary of a pedestrian in three frames. The points are identified by blue colour after 18 iterations and the ESE curves are shown by a black line.

Fig. 7.20. Identifying the boundary of a pedestrian at various poses after18 iterations.

The interpretions in Figs. 7.19 and 7.20 demonstrate that good stability and consistent results are repeatedly obtained when the pose changes by a modest amount. Figs. 7.19 and 7.20 also support that the transformation aspects of the registration process are important so that the ESE curve of a new object in a new image corresponds as much as possible to the contour paths of the images in the training set.

Any dependence between the parameters of the ESGM would imply nonlinear relations between the original positions of points and would result in some combinations of parameters leading to inappropriate shapes. By varying the first three modes of variation of the ESGM a very large set of example shapes with large variability in pose are generated; a small set of example shapes bounded with an ESE curve red line, together with the points in blue identified on each example is shown in Fig. 7.21. These examples show the effect of varying the first three parameters of $\mathbf{b_{se}}$ of the ESGM interpretation, as identified from the training set, through ±3 standard deviations, where $\mathbf{b_{se}}$ is the vector that controls the variation of each mode of the ESGM. The first parameter corresponds to the largest eigenvalue which gives its variance across the training set.



Mode 1 (b_{se_2} varies ±3 s.d.s)

Mode 2 (b_{se_2} varies ±3 s.d.s)

Mode 3 (b_{se_3} varies ±3 s.d.s)

Fig. 7.21. A small set of generated shapes showing the effect of varying the first three modes of ESGM parameters through ±3 s.d.s from the mean.

To illustrate the potential of the ESGM to represent the variability of vehicles, an ESGM was created for five types of vehicle image. Variant models of ESGM were generated for each vehicle type. To determine the appropriate model for a vehicle interpretation; confidence in an interpretation of a vehicle was estimated using the log likelihood ratios and standard deviations (SDs) to allow variant models to be selected. These criteria represent the variation between the alternative vehicle models that take into consideration the distribution of model parameters. The estimation of the log likelihood ratios and the data for which the SD of the variation between the models is computed is Gaussian distributed. The likelihood ratios for the data points that sample the boundary of an object and the pixel values that represent the object and the SDs that represent the variation between the vehicle types were computed for each interpretation and each instance model as explained in subsections 5.7.1 and 5.7.2. A naïve Bayesian classifier used the log likelihood ratios and SD features in training and testing to determine the confidence of match to each vehicle type as described in subsection 5.7.3. The vehicle types are: hatchback cars, minivans, compact cars, city cars and vans. In training the naïve Bayesian classifier, the feature values of the log likelihood ratios and SDs for each vehicle model were collected in a list of twenty five feature vectors. In evaluation a Bayesian classifier, the log likelihood ratios and SDs were computed for the interpretation to identify the most appropriate model. There are twenty log likelihood ratios, five SDs and five classes of vehicle types.

The detection of cue points within the body of each vehicle and the key points required to form an ESGM for vehicle interpretation, the modelling and the interpretation algorithms are described in Sections 3.4, 3.5 and subsection 6.2.2, respectively. Given the five variant vehicle models, A, B, C, D and E, there are twenty log likelihood ratios and five SD features that represent the varia-

tion between the variant vehicle types. These data form a list of twenty five feature vectors that were used to train the naïve Bayesian classifier. In testing, the feature vectors of log likelihood ratios and SDs was computed for each interpretation and the classifier used these vectors to identify if the best vehicle model has been used and if not to identify what would be a better alternative vehicle model for a previously unseen vehicle. The number of images for training and test datasets of the naïve Bayesian classifier for each vehicle type is 200 and 150, respectively. This gives 100 degrees of freedom and 10 samples per degree of freedom for the training set and 7 to 8 samples per degree of freedom for the test data set [PAN08].

The classifier gave a rate of classification of 84.2% for 1000 training and 750 evaluation vehicle images. The identification of key points around the boundary of vehicles of different poses and structures using the ESGM interpretation of vehicle images in interpretation at iteration 15 is shown in blue in Fig. 7.22 and the ESE boundary curves are shown by a black line.

Fig. 7.22. Illustration of key points selected after 15 iterations and shown in blue, for vehicle images joined by a black line.

The interpretion in Fig. 7.22 demonstrates that the ESGM of vehicles can identify the boundary of vehicle types.

An ESGM with the same methodology as the ESGM for modelling and interpreting pedestrians alone was trained on a dataset of pedestrians alone and pedestrians associated with pushchairs and bicycles. This ESGM was evaluated on an independent set of images. The number of images for training and test datasets of the ESGM for modelling and interpreting people combined other objects, for each object type, is 200 and 150, respectively. There is 1 sample per degree of freedom for the training data set and less than 1 sample per degree of freedom for the test set [PAN08]. These images are used throughout the results reported in Figs. 7.23 and 7.25. The identification of key points around the boundary of a combination of objects using the ESGM interpretation for people combined with other objects at iteration 15 is shown in blue in Fig. 7.23 and the ESE curves are shown by a black line.

(a) (b) (c)

Fig. 7.23. Illustration of key points selected after 15 iterations and shown in blue: (a) a pedestrian pushing a pushchair, (b) a pedestrian pushing a bicycle and (c) a person riding a bicycle.

Fig. 7.23 (a) shows the boundary points from an ESGM for a person pushing a pushchair in (a), (b) a person pushing a bicycle and (c) a person riding a bicycle. The same model representation and interpretation strategy is used in these cases as for a pedestrian alone.

The interpretation illustrated in Figs. 7.19 to 7.23 shows the potential of the ESGM to identify the points that sample the boundary of objects of very different structures.

7.5.1 Evaluation of ESGM-Based Interpretation

The ESGM-based interpretation was evaluated in terms of the closeness of the interpretation to the best manually identified ground truth points along the boundary. The square root of the mean of the sum of the squared errors, i.e. the RMS value in the geometric domain was computed as:

$$E_z = \sqrt{\frac{1}{K}\sum_{j=1}^{K}\left(\hat{x}_j - x_j\right)^2 + \left(\hat{y}_j - y_j\right)^2} \tag{7.7}$$

Where: K represents the number of key points, $\left(\hat{x}_j, \hat{y}_j\right)$ represents the nearest single shape key point created by the interpretation that is re-sampled from a respective pair of exponent points and $\left(x_j, y_j\right)$ is a single boundary ground truth point.

Equation 7.7 is a measure of the precision of the match between the image and its interpretation. The accuracy of model interpretation was calculated by the square root of the mean of the sum of the squared differences between the ground truth points and model points.

7.5.2 Convergence Curve of ESGM-Based Interpretation

Six sets of pedestrian images for training and an independent set for testing were selected to evaluate the ESGM interpretation for pedestrian images interpretation. The training sets were: set A of 30, set B of 40, set C of 50, set D of 70, set E of 90 and set F of 120 images. The test data set for all training datasets consisted of 80 previously unseen images of pedestrians. The images

used in the interpretation were distinct from the training data set. The performance of ESGM interpretation for the six sets of images of pedestrians is shown for up to 25 iterations in Fig. 7.24. These convergence curves represent the square root of the mean of the sum of the squared errors per pixel between the points created by the ESGM interpretation for images of pedestrian's interpretation and the true manually defined key points. This RMS metric measure is defined in Equation 7.7. The vertical error bars in Figs. 7.24 to 7.28 represent unit standard error of the mean of the data presented in the curves. The standard error of the mean and its mathematical representation are described below. The data in Figs. 7.24 to 7.27 represents the square root of the mean of the sum of the squared differences (using Equation 7.7) between the points created by the ESGM interpretation and the best manually identified ground truth key points along the boundary for the same image as used for the interpretation and not in training.

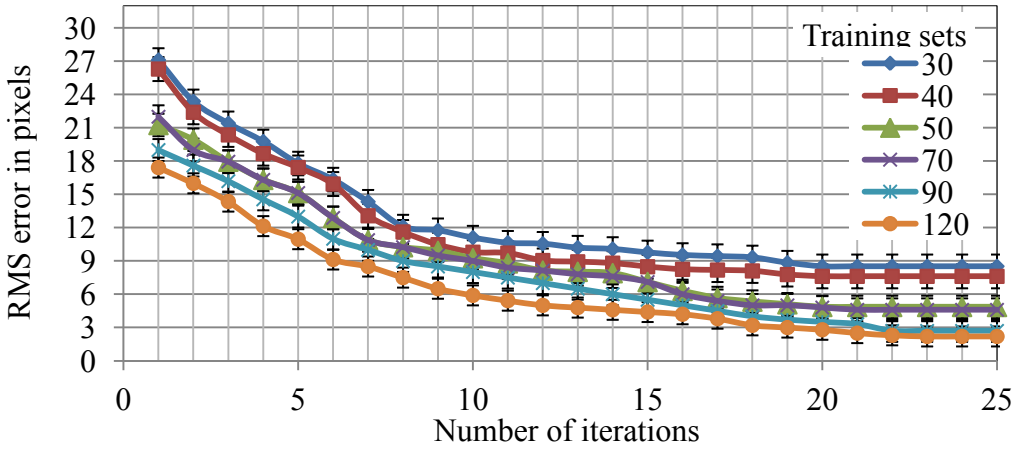


Fig. 7.24. Key point position estimation "error" for pedestrian data sets. The vertical error bars represent one standard error of the mean.

The curves in Fig. 7.24 show that a small "square root of the mean of the sum of the squared distances" was achieved after 8 iterations. There is a further small reduction in this value after 20 iterations. Fig. 7.24 also shows that the model created with 120 images is a slightly better repre-

sentation than the one created with 90 images and suggests that a large training set is basically to improve the interpretation process significantly [COO01a] [GAO10]. However, the model created with 120 images has much less than the standard error on the error measure. The reduction in the error with increase in size of training data set has at this stage become extremely small in proportion to the increase in size of training data set. Any further improvement in error is likely to need a very large increase in training data set size. It is possible that a large increase in the size of training data set will significantly further reduce the error measure. The error for some data sets continues to reduce a little with further iterations. The error measure may reduce further with more than 25 iterations with a data set of 120 images. There is a significant difference between the curves and indicates that the models created with 120 images and 90 images achieved a better interpretation than the other models. There is a consistent difference between the curves of 1 SD which suggests that the significance is greater than the 63% suggested by the error bars for each point on the curve. The error bars for adjacent curves of the models created with 40 and 50 images and 70 and 90 images do not overlap; this means that it would take an extreme deviation that occurs 66% in the data for both points for the curves to coincide at that point. This means that the joint likelihood of these curves are distinct is of the order of 85% meaning that they are very clearly separated. Therefore, there is a very high confidence that the difference is significant because this degree of separation exists at several points along the curves.

To calculate the standard error of the mean for a set of data measurements:

1) Calculate the mean ($\bar{\mathbf{x}}$) of the data, as defined by:

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{7.8}$$

Where: x_i represents the i^{th} value of the data and *n* represents the total number of data values.

2) Calculate the standard deviation (σ) of the data using:

$$\sigma = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \bar{\mathbf{x}})^2}{n-1}} \tag{7.9}$$

3) Calculate the standard error (SE) of the mean:

$$\text{SE} = \frac{\sigma}{\sqrt{n}} \tag{7.10}$$

The error bars of the standard error of the mean can be used to describe the uncertainty in data measurements. Standard Error of the mean is a statistical measure used to determine whether the vartiability and the difference between the means of two sets of measurements are statistically significant. When standard error bars between the possible ranges of two means do not overlap, then it can be concluded that the difference is statistically significant while the difference is probably not significant in a statistical sense when standard error bars overlap.

The performance of ESGM interpretation using images of pedestrians with puschairs or bicycles is shown for up to 25 iterations in Fig. 7.25. These convergence curves represent the square root of the mean of the sum of the squared errors per pixel between the best interpretation-based ESGM and the best manually identified ground truth boundary points. This RMS evaluation measure is defined in Equation 7.7. An independent set of 150 previously unseen pedestrains with objects of each type was used. The training and test datasets for the pedestrians associated with other objects for the results reported in Fig. 7.25 are described in Section 7.5. The vertical

error bars represent unit standard error of the mean of the data presented in the curves. The standard error is defined in Equation 7.10. The data of the curves represents the RMS errors (using Equation 7.7) between the points created by the ESGM interpretation and the best manually ground truth points.

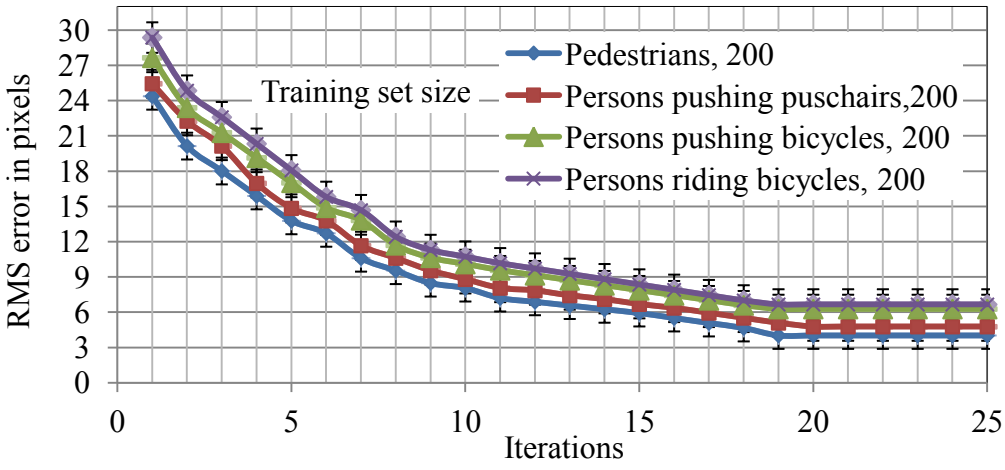


Fig. 7.25. Key point position estimation "error" for data sets of pedestrians and other objects. The vertical error bars represent one standard error of the mean of the data.

The curves in Fig. 7.25 show a measure of the accuracy of the match between the image and its interpretation. There is a small difference between curves of 1 SD which suggests that the significance is less than the 63% suggested by the error bars for each point on the curve. The degree of separation between the curves exists only at very few points along the curves. The error bars for the curves of the models created for pedestrians pushing bicycles and pedestrians pushing pushchairs and also for pedestrians alone and pedestrians pushing pushchairs overlap which means that the difference is not significant in a statistical sense. This means that it would not take an extreme deviation that occurs 66% in the data for both points for the curves.

The ability of the ESGM method to interpret unseen vehicle images was assessed for five variant vehicle body shapes. The training and independent test datasets consisted of 200 and 150 images of each vehicle body shape as described above. There is 1 sample per degree of freedom for the training data set and less than 1 sample per degree of freedom for the test data set [PAN08]. The convergence curves, of the performance of ESGM interpretation, for the boundary error for each vehicle type are presented for up to 25 iterations in Fig. 7.26. The boundary error was defined by the square root of the mean of the sum of the squared errors per pixel between the points created by the ESGM interpretation of vehicles and the equivalent manually ground truth points. The criterion measure of boundary error is defined in Equation 7.7. The error bars represent unit standard error of the mean of the data presented in the curves. The mean, standard deviation and standard error criteria are defined in Equations 7.8, 7.9 and 7.10, respectively. The data of the plots in Fig.7 26 represents the square root of the mean of the sum of the squared differences between the points created by the ESGM interpretation and the best ground truth key points along the boundary for the same image as used for the interpretation and not in training.

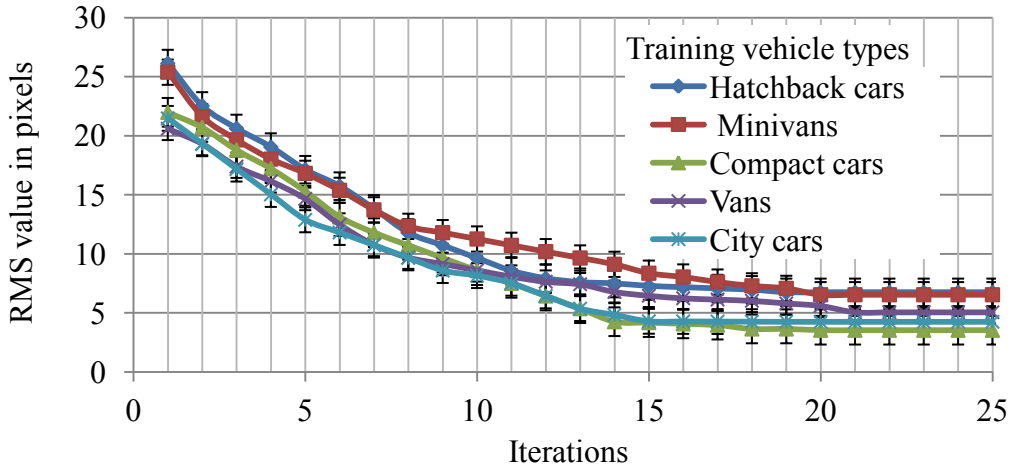


Fig. 7.26. The mean of the square root of the sum of the squared distances between the generated points of ESGM interpretation and the ground truth points for unseen vehicles. The vertical error bars represent one standard error of the mean of a set of measurements.

In Fig. 7.26, the smallest RMS distance between the ground truth and model boundary for each vehicle type is approximately 4 pixels. There is a significant difference between curves of 1 SD which suggests that the significance is greater than the 63% suggested by the error bars for each point on the curve. The error bars for adjacent curves of the models created for vans and city cars and the adjacent curves of the models created for vans and minivans do not overlap; this means that it would take an extreme deviation that occurs 66% in the data for both points for the curves to coincide at that point. Therefore, there is a high confidence that the difference is significant because this degree of separation exists at several points along the curves.

The error in Fig. 7.24 is very small, approximately 1.5 pixels and that in Figs. 7.25 and 7.26 is similar at approximately 4 pixels. It is the stability of model-based image interpretation with variant object types that is much more important than the steady improvement with increasing data set size and iterations. This is shown by smooth curves and consistent way in which model stability and interpretation accuracy improves with increased training data set size and number of iterations. The consistency with object types being within the standard error for Figs. 7.24 and 7.25. That the curves in Figs. 7.23 - 7.25 are similar show steady improvement that shows a similar pattern within the standard error. The convergence curves show that the ESGM provides a stable estimate of the boundary for a range of objects that differ in structure with large and small sizes of image datasets.

7.5.3 A Comparison Between PDM and ESGM

The dataset used to illustrate the behaviour of the ESGM against the PDM is described in Section 7.5. The range of pedestrian's heights is [85-190]. The boundary of each pedestrian is identified

by 43 key points. The images are of varying complexity of backgrounds and foregrounds along with a degree of variation in the form of the objects of interest, which are pedestrians. There are simple cases of people in isolation and in groups. Further, the images in the training set involve pedestrians with a large variation of pose. The models used in these experiments hold 97.5% of the total variance with 12 modes of variation. The mean width and height of the ESGM are respectively 52.3 and 137.6 pixels. The convergence curves of the boundary estimation using the ESGM and the PDM for the interpretation of images of pedestrians of an independent test set of 80 previously unseen images are presented for up to 25 iterations in Fig. 7.27. The boundary estimation was computed using the measure of Equation 7.7 which defines the RMS value in pixel between the points created by the model instances of images of pedestrians and the equivalent manually ground truth key points. The vertical error bars represent unit standard error of the mean of the measurements across the experiments conducted; this measure is defined in Equation 7.10. The data of the curves represents the RMS values (using Equation 7.7) between the points created by the ESGM and PDM interpretation and the best manually ground truth boundary points for the same image as used for the interpretation and not in training.

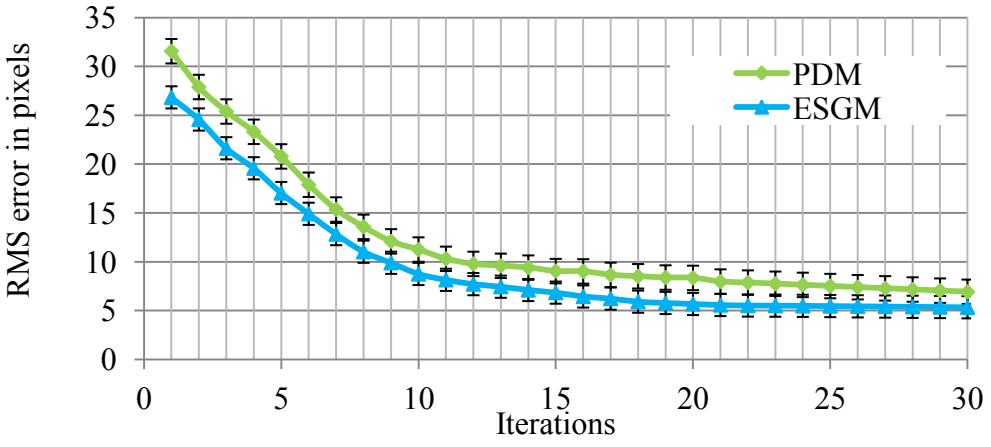


Fig. 7.27. Convergence curves of point position estimation using the mean of the square root of the sum of the squared distances between the ground truth and model points of pedestrian image interpretation for the ESGM and the PDM. The vertical error bars represent one standard error of the mean of a set of measurements.

The curves show a measure of the accuracy of the match between the image and its interpretation. The accuracy of ESGM and PDM interpretation was calculated by the mean of signed differences between the ground truth points and the model points.The two curves are just separable at around one standard error of the mean. This is a significant difference between the two curves and indicates that the EGSM can achieve a better interpretation with objects that vary in form. The significant difference between the two curves of 1 SD suggests that the significance is greater than the 63% suggested by the error bars for each point on the curve. The error bars of the curves for the models of ESGM and PDM do not overlap; this means that it would take an extreme deviation that occurs 66% in the data for both points for the curves to coincide at that point. This means that the joint likelihood of these two curves are distinct is of the order of 85% meaning that they are very clearly separated and because this degree of separation exists at several points along the curves there is a very high confidence that the difference is significant.

The eigenvalues in a model provide a measure of the amount of variance captured by each mode of variation. The sum of magnitudes of all eigenvalues gives a measure of the total variance present in the model. This property is better described as specificity where the compactness of a model is concerned with the number of parameters in the model. A large or a small variance can both be argued to be positive indicators. The contribution in percentage of the first set of modes to the overall variance of the training set is given by:

$$\eta_i = \lambda_i/\lambda_T = \lambda_i/\sum_{j=1}^{2K} \lambda_j * 100 \tag{7.11}$$

Where: λ_i is the eigenvalue at index *i* and λ_T is the total variance of the training set.

Fig. 7.28 shows a statistical comparison between the ESGM and the PDM using the first twelve modes of variation. The graph shows in percentage, for each model, the cumulative distribution of the total variance reported with respect to the number of modes of variation used.

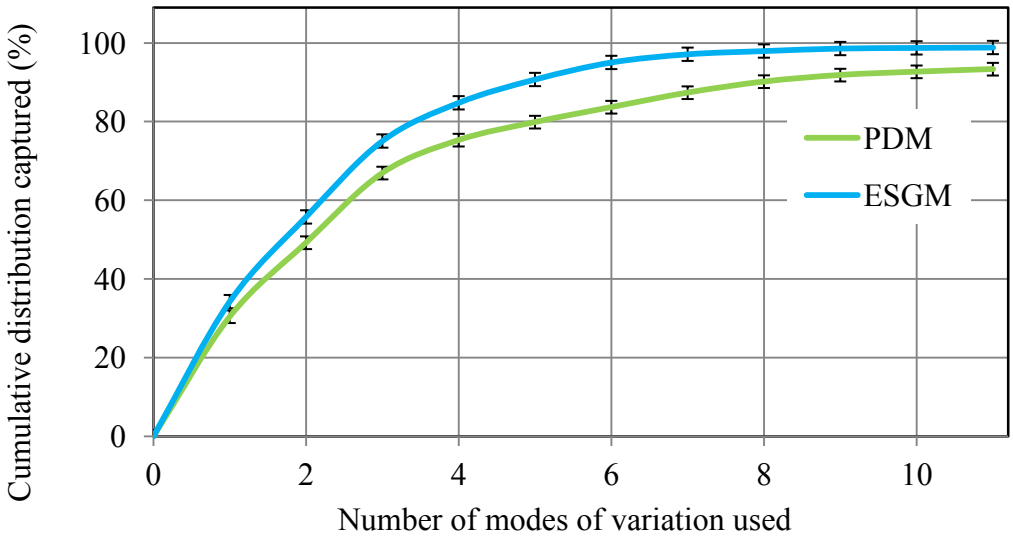


Fig. 7.28. The cumulative distribution of variance in the ESGM and PDM. The vertical error bars represent one standard error of the mean of the data.

The error bars in Fig. 7.28 are shown for one standard error of the mean of the measurements of the cumulative distribution across the experiments conducted; this measure is defined in Equation 7.11. It is observed that the behaviour of the ESGM and PDM is different with the same image dataset. There is a significant difference between the two curves of 1 SD which suggests that the significance is greater than the 63% suggested by the error bars for each point on the curve. The two sets of error bars do not overlap in Fig. 7.28. There is therefore a significant difference between the two curves which means that it would take a high degree of deviation that occurs 66% in the data for both points for the two curves to coincide at that point. The two curves are clearly separated at many points along the curves which indicate that there is a high confidence that the difference between the ESGM and PDM using the cumulative distribution of variance is signifi-

cant. Each eigenvalue of the ESGM is the variance of the corresponding mode of variation over the training set. The ESGM and PDM were generated with the same range of the respective vectors that control the variation of each mode for each model given within ±3 of the square root of the corresponding eigenvalue. It can be observed from Fig.7.28 that most of the variation in the ESGM is covered by the first few modes. There is still a fair amount of variation in the forth mode of the ESGM.

After 6 or 7 modes 95% of the variation has been explained by the ESGM. Each mode of variation plotted for ESGM contains significant information but that after 6 or 7 modes, the information is not significant. The PDM explains a slightly lower level of variance within the first 11 modes of variation. As can be seen, statistically, the PDM is little less compact than the ESGM, as the variance captured as the number of modes of variance incorporated is, in each case, a little lower. As a consequence, ESGM is more compact, has a smaller search space and specific at all modes because it needs fewer modes of variation to explain a high level of variation in the model.

The results are encouraging and suggest that the ESGM is a good model representation. An important property of the ESGM is that the parameterisation process provides a compact and complete model (with 6 or 7 modes). Cootes and Taylor [COO04] described a statistical model-based approach to the interpretation of images of faces using a shape model of 36 modes of variation, which explained 98% of the variance in the landmark positions in a training set of 300 faces. The PDM commonly used a total of 19 modes of variation for a data set of more than 200 images to hold 95% of the total variance of shape data [MAR08]. Here, the evaluation measures of the

ESGM using the metrics of statistical difference, cumulative distribution and the accuracy of model interpretation demonstrate a possible slight improvement of ESGM over the PDM.

7.6 Fitting Results of ESAM

Images with simple and complex cases of people alone, in groups and in combination with other objects and vehicles were used to illustrate the effectiveness of the ESAM, see examples and evaluation below. The ESAM allows the shape to be distorted slightly and in some situations significantly in positions where an object can be occluded and still be recognized. ESAMs were created for pedestrians alone, pedestrians combined with other object and five types of vehicles. The ESAM was produced as described in Section 5.6. The interpretation is performed by describing the ESAM instance by a set of parameters using an appearance vector. The training and interpretation methods of the ESAM are described in Section 6.3. The number of modes was 18 and the ESAM in all the experiments conducted explained 97.5% of the total variance.

7.6.1 Results of ESAM Image Interpretation

The ESAM for pedestrian image interpretation was trained on a dataset of more than 200 images of pedestrians of various poses, clothes and sizes, carrying bags with simple and complex foregrounds and backgrounds. The ESAM was evaluated on an independent set of 150 images of pedestrians. Some distortions are significant but most are small. The ESAM was trained by systematically displacing the model parameters of the training set as described in Table 6.1. The modelling was performed using an appearance vector as described in subsection 6.3.2. The ESAM interpretation algorithm in Fig. 6.3 by which the model is varied and matched was applied to fit an instance of ESAM in an unseen image.

The texture model of the ESAM modelled the colour variation of an image over the regions sampled by the reference mean and the texture component. Image warping was used to sample the image values between the key points, and the pixel values were computed at positions within regions as determined by the position of points in the mean model. This was to obtain a reference texture image. The triangulation technique described in subsection 5.5.1 was employed to warp the key points and the intermediate points. The ESAM interpretation minimises the texture difference between an ESAM instance and the part of the target image it represents. The interpretation was applied to the content extracted from the image by fitting an ESAM instance to a previously unseen image as described in subsection 6.3.3. The objects of interest were identified by key points which located along the boundary of the objects. Therefore, the background was eliminated from the model instances and the difference images, as shown in the ESAM interpretation examples in Figs. 7.29 to 7.31. The source images, model instances and the difference between each re-constructed image and the previously unseen images are at similar sizes in all interpretation examples as shown in Figs. 7.29 to 7.31. Some pedestrians in some test images might have poses and clothes similar to some pedestrians in the training images.

Fig. 7.29 shows interpretation results with the ESAM for pedestrians using 18 modes at the 6^{th}, 10^{th} and 18^{th} iteration for three snapshots of a pedestrian from three sequences, each with a modest variation of pose, for pedestrians walking in different directions and in different scenes.

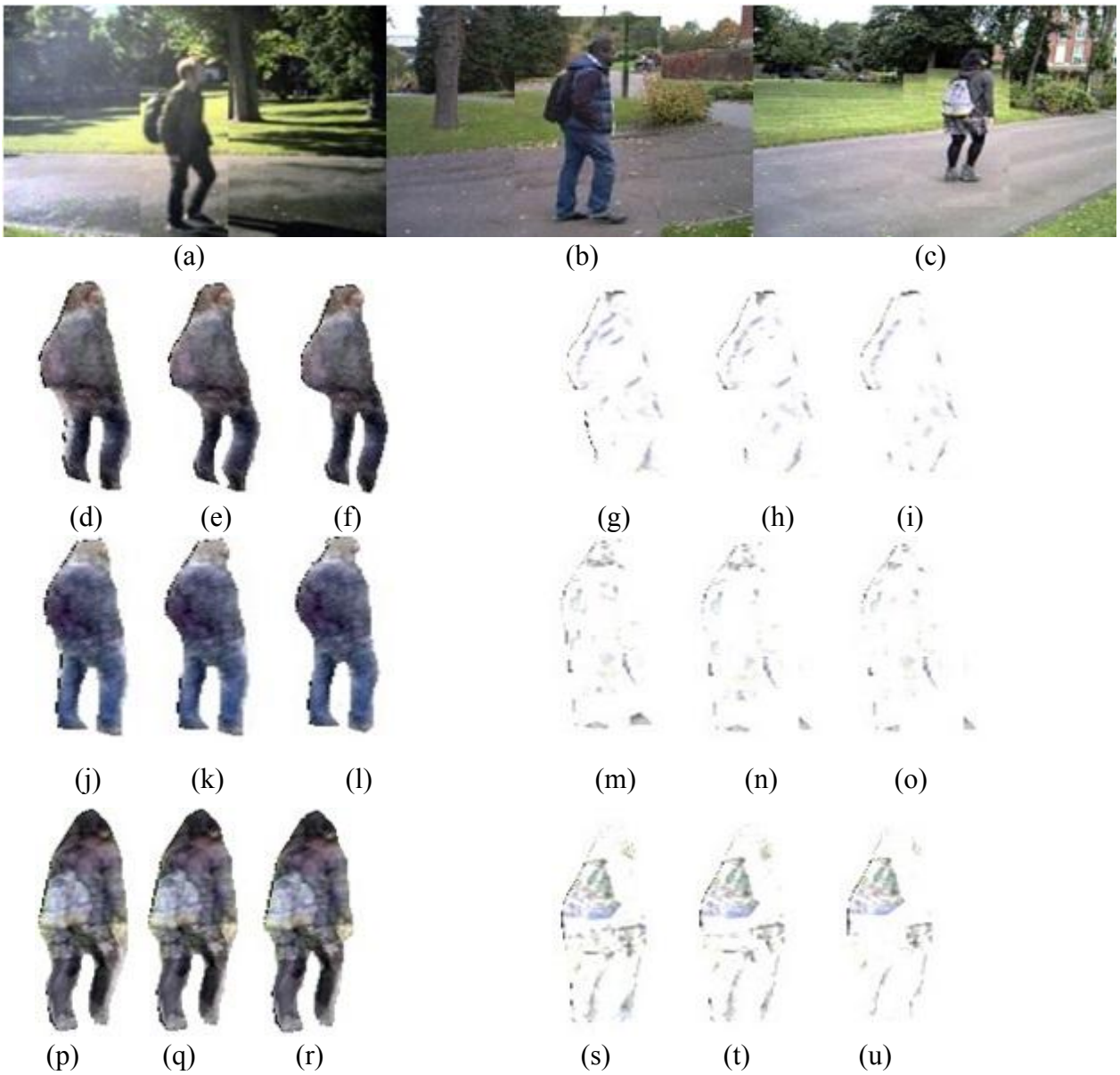


Fig. 7.29. Interpretation results: (a), (b) and (c) are the original source images, (d), (j) and (p) are the images reconstructed from the ESAM after 6 iterations, (e), (k) and (q) after 10 iterations and (f), (l) and (r) after 18 iterations; (g), (h) and (i) are the differences between the respective images in (d), (e) and (f) and the image in (a); (m), (n) and (o) represent the differences between the respective images in (j), (k) and (l) and the images in (b); (s), (t) and (u) represent the differences between the respective images in (p), (q), (r) and the image in (c).

The pedestrians in the images of Fig.7.29 (a) to (c) seems similar in size to those in the synthesised images of Fig 7.29. In Fig. 7.29, the previously unseen images are shown in the first row.

Rows 2 to 4 in columns 1 to 3 show the reconstructed images of pedestrians at iterations of 6, 10 and 18, respectively, rows 2, 3 and 4 for columns 4, 5 and 6 show the difference between each reconstructed image that best matches the previously unseen image. The results in this figure demonstrate a level of modelling and reconstruction that is good after 6 iterations and very good after 18 iterations. The error for the reconstructed images is visually judged to be small because the reconstruction process approximates the colour patches of the unseen images by a linear combination of the eigenvectors of the ESAM.

An instance of the ESAM for pedestrian images interpretation, after iteration 25, was fit to unseen images of pedestrians in different contexts and poses, as shown in Fig. 7.30. A very good match, as visually judged, was realised between the previously unseen source images of pedestrians and the model instances in Fig. 7.30.

The ESAM interpretation for pedestrian image interpretation captures the variation that arises as the pose, size and shape of a pedestrian changes, where $\mathbf{b}_{\mathbf{ae}}$ is a parameter vector of appearance that controls the mode of variation of the ESAM. Fig. 7.31 shows a small set of mode variations for the model to demonstrate the effects of varying the first three parameters of the ESAM, which represents the shape and the texture parameters, through ±3 standard deviations, with respect to the mean model. The first parameter varies the width and appearance of the object. The second and third parameters vary the shape of the body. Fig. 7.40 shows how the ESAM performs in comparison with the AAM.

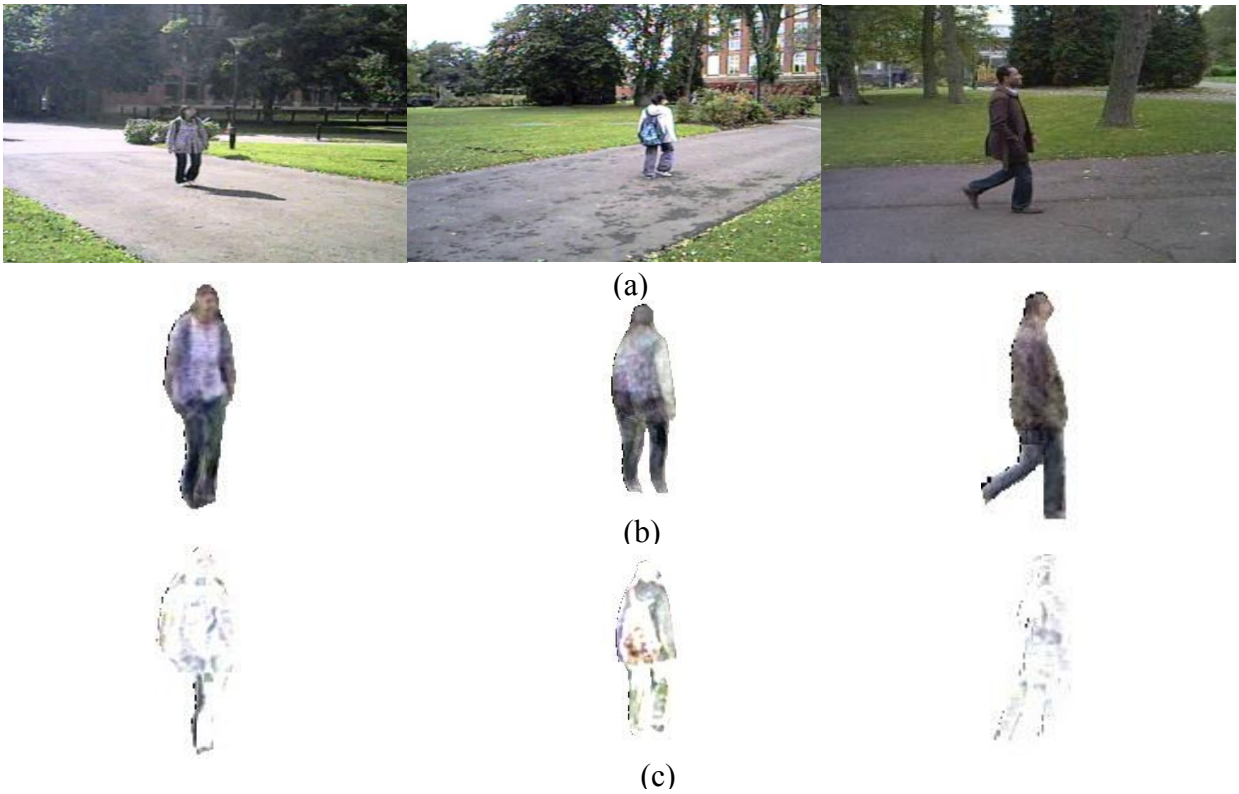
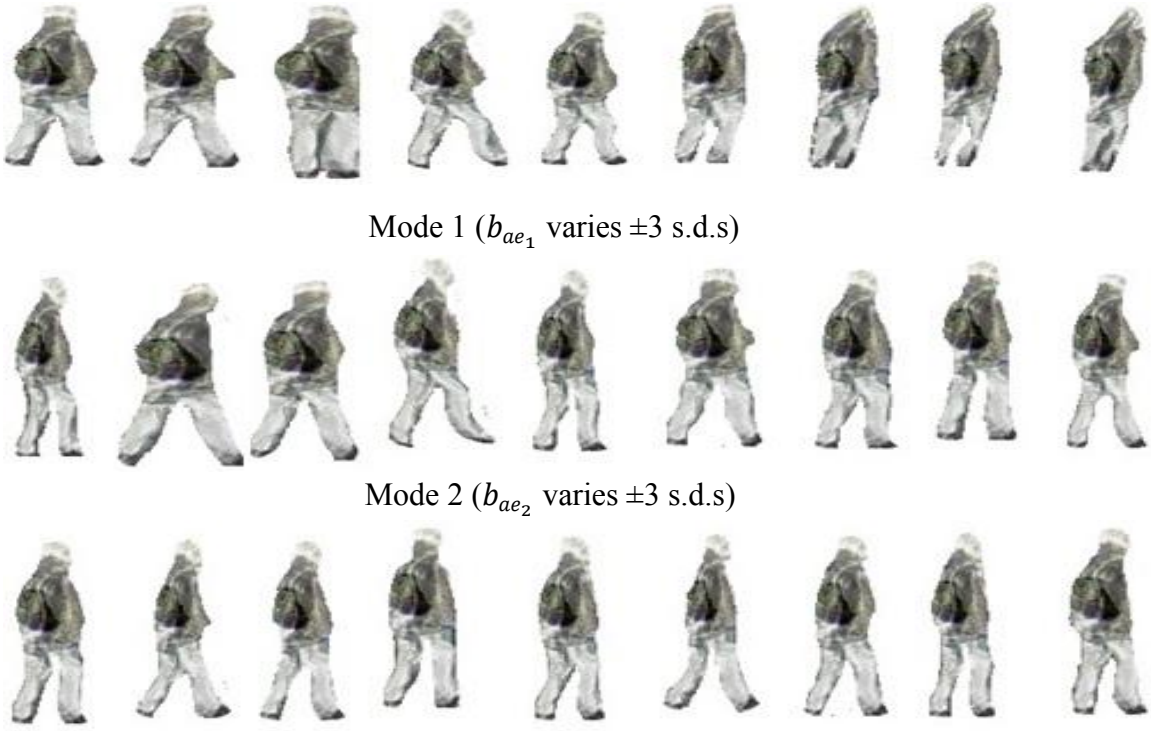


Fig. 7.30. Illustration of the interpretation after 25 iterations: (a) the source images, (b) the model instance and (c) the difference between the model instance and each source image.



Mode 1 (b_{ae_1} varies ±3 s.d.s)

Mode 2 (b_{ae_2} varies ±3 s.d.s)

Mode 3 (b_{ae_3} varies ±3 s.d.s)

Fig. 7.31. A small set of reconstructed images showing the effect of varying the first three modes of variation of ESAM, bae, by ±3 s.d.s from the mean.

An ESAM was created on a dataset of images for pedestrians alone and with pushchairs and bicycles. This ESAM was evaluated on an independent set of unseen images for pedestrians combined with pushchairs and bicycles. The number of images for training and test datasets for each object type of this ESAM was 200 and 150, respectively as introduced in Section 7.5. There is 1 sample per degree of freedom for the training data set and less than 1 sample per degree of freedom for the test data set [PAN08]. The results for the interpretation of pedestrians combined with other objects in images using ESAM are shown, after iteration 25, in Fig. 7.32. The previously unseen images, model instances and the corresponding difference images are shown in Fig. 7.32, parts (a), (b) and (c), respectively.

Fig. 7.32. Illustration of the interpretation for a combination of people with other objects: (a) the previously unseen source images (b) the model instances and (c) the difference between the model instances and each source image.

An ESAM was created to model and interpret images of five types of vehicle and evaluated on previously unseen sets of vehicle images. The number of images for the training and test sets of the ESAM for each vehicle type is 150 and 100, respectively. The number of degrees of freedom is 100 with 10 samples per degree of freedom for the training set and 7 to 8 samples per degree of freedom for the test data set [PAN08]. The vehicle images vary in structure and form as described in Section 7.5. The previously unseen vehicle images, ESAM instances for vehicle interpretation and the corresponding difference images at iteration 25 are shown in Fig. 7.33 (a), (b) and (c), respectively.

(a)

(b)

(c)

Fig. 7.33. Illustration of the interpretation for vehicle images: (a) the previously unseen source images (b) the model instances and (c) the difference between the model instances and the corresponding source images.

The results shown in Fig. 7.33 demonstrate that the ESAM interpretation of vehicle images interpretation is potentially reliable.

The reconstruction examples of objects of interest in Figs. 7.29 - 7.30 and. 7.32 - 7.33 look particularly good and there appear to be very small errors as visually judged by the difference between the previously unseen objects of interest and the model instances. This result is specifically good given a model with relatively large training data set and large number of images per degree of freedom. The interpretation results indicate that the ESAM has the potential flexibility to represent and interpret a wide variety of objects and accommodate a good range of configuration and pose variation.

Situations in which an ESAM-based interpretation for people combined with other objects is considered unsuccessful are shown for persons alone in Fig. 7.34 (a) and (b) and for a person riding a bicycle in Fig. 7.34 (c).

Fig. 7.34 shows situations in which the error rate for an ESAM interpretation of persons alone and a person riding a bicycle was visually judged to be large. These examples failed to converge to a satisfactory result. The ESAM interpretation in Fig. 7.34 might have failed because the relevant or a similar pose to the test images was not present in the training dataset. The training dataset is large enough but the image data set might not explore all the variations. The nature of errors in Fig. 7.34 (a) and (b) were related to a change in appearance and that the pose was not well detected such that the model instance in Fig. 7.34 (h) carry a bag on shoulder while the original

source image in Fig. 7.34 (b) does not carry a bag. The nature of errors in Fig. 7.34 (i) was related to a change in appearance.

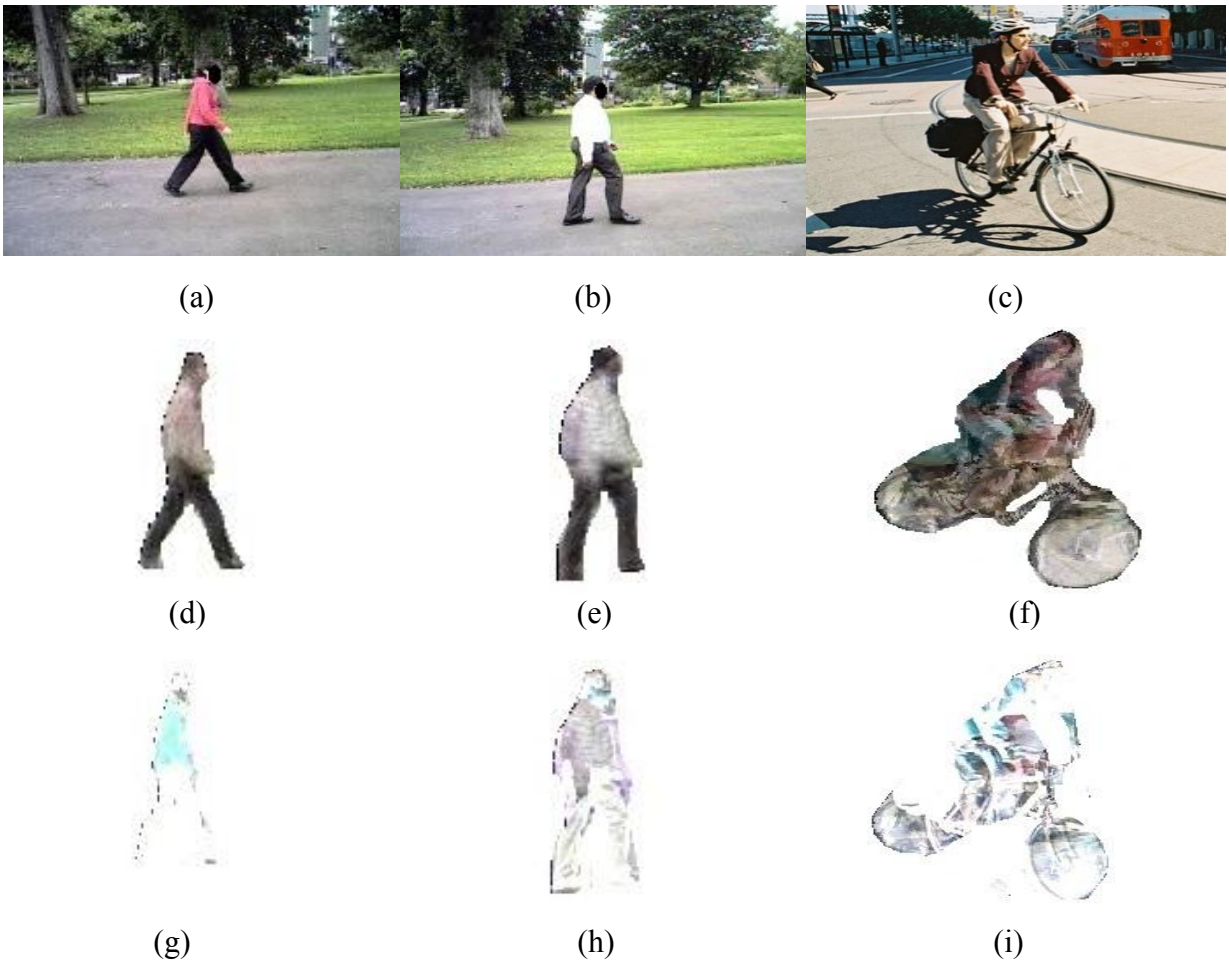


Fig. 7.34. Examples of failed interpretation where the error is related to a change in appearance such that the pose is not well detected: (a), (b) and (c) the source images; (d), (e) and (f) the model instances; (g), (h) and (i) the differences between the model instances and the corresponding source images in (a), (b) and (c).

7.6.2 Evaluation of ESAM-Based Interpretation

The performance of the ESAM was evaluated using the square root of the mean of the sum of the squared differences (RMS value) between the target image pixel values and the final model instance using:

$$E_e = \sqrt{\frac{1}{M}\sum_{j=1}^{M} \left(I_{i_{R_j}}(x_j, y_j) - I_{m_{R_j}}(x_j, y_j)\right)^2 + \left(I_{i_{G_j}}(x_j, y_j) - I_{m_{G_j}}(x_j, y_j)\right)^2 + \left(I_{i_{B_j}}(x_j, y_j) - I_{m_{B_j}}(x_j, y_j)\right)^2} \quad (7.12)$$

Where: M is the number of colour values across all three channels of either the image or the model., $I_{i_{R_j}}, I_{i_{G_j}}, I_{i_{B_j}}, I_{m_{R_j}}, I_{m_{G_j}}$ and $I_{m_{B_j}}$ are the respective colour pixel values for each channel of the source image, I_i, and the final model instance, I_m, respectively, at position (x_j, y_j).

The ESAM fitting algorithm in Fig. 6.3 decreases the texture error vector and adjusts the shape if that leads to an overall improvement in the texture match. The interpretation error using images of pedestrians is shown for up to 25 iterations in Fig. 7.35. The convergence curves in Fig. 7.35 represent the square root of the mean of the sum of the squared errors between the ESAM instances of pedestrians and 80 previously unseen pedestrians. The vertical error bars represent unit standard error of the mean of the measurements across the experiments conducted in Figs. 7.35 to 7.40. The data in Figs. 7.35 to 7.40 represents the square root of the mean of the sum of the squared differences (using Equation 7.12) between the pixel values of the re-constructed images created by the ESAM interpretation and the image pixel values of the previously unseen images for the same images as used for the interpretation and not in training. The mean, standard deviation and standard error criteria are defined in Equations 7.8, 7.9 and 7.10, respectively.

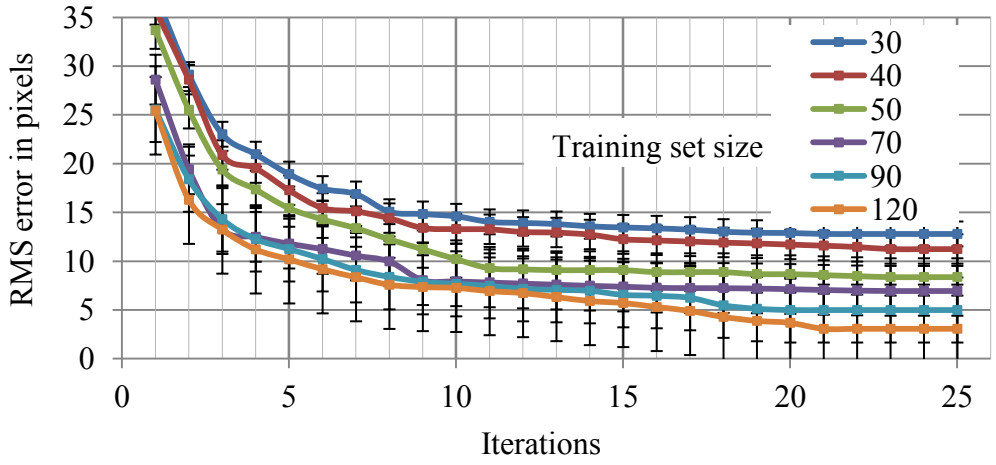


Fig. 7.35. The differences in pixels between the model instances and the corresponding unseen images of pedestrians as a function of training set size and iterations in ESAM interpretation. The vertical error bars represent one standard error of the mean.

The search converges to a smaller residual error for training set sizes of 90 and 120 training images. The ESAM interpretation again reaches a small error after 20 iterations and improves a little after 20 iterations. The difference between the model instances and the previously unseen images with training sets of 120 and 90 images is not significant after the first 15 iterations. Therefore, it seems unlikely any further increase in the size of the image training set beyond 120 would reduce the convergence error significantly. There is a significant difference between the curves and indicates that the models created with 120 images and 90 images achieved better interpretation than the other models. Further, there is a consistent difference between curves of 1 SD which suggests that the significance is greater than the 63% suggested by the error bars for each point on the curve. The error bars for adjacent curves of the models created with 40 and 50 images and adjacent curves of the models created with 70 and 90 images do not overlap; this means that it would take a very great deviation that occurs 66% in the data for both points for the curves to coincide at that point. This means that the joint likelihood of these curves are distinct is of the order of 85% meaning that they are very clearly separated. Therefore, there is a very high confidence

that the difference is significant because this degree of separation exists at several points along the curves. The RMS error measures are sensitive to outliers [CHA14] [GUP10]. This means that the impact errors that are outliers may be well represented in this measure. This is in turn means that there might still be some value in a large training and test datasets. An ESAM for images of pedestrian's interpretation was compared to the evaluation of AAM with a large data set as described below in Fig. 7.40.

The performance of ESAM interpretation for a training set of 120 images of pedestrians taken from the TUD person dataset [WAN07] and an evaluation dataset of 80 previously unseen pedestrians is shown in blue plot for up to 25 iterations in Fig. 7.36. The image dataset used to generate the results in red plot is an image dataset collected at the University of Birmingham. In the results shown in red line the ESAM was trained on 120 images of pedestrians and evaluated on 80 previously unseen images of pedestrians. These convergence curves represent the RMS error in pixels, as defined in Equation 7.12, between the ESAM instances of images of pedestrian's interpretation and previously unseen images of pedestrians. This metric measure is defined in Equation 7.12 by the square root of the mean of the sum of the squared errors. The vertical error bars represent unit standard error of the mean of the measurements across the experiments conducted; this measure is defined in Equation 7.10. The data of the curves represents the square root of the mean of the sum of the squared differences between the pixel values of the images created by the ESAM interpretation and the pixel values of the previously unseen images for the same images as used for the interpretation.

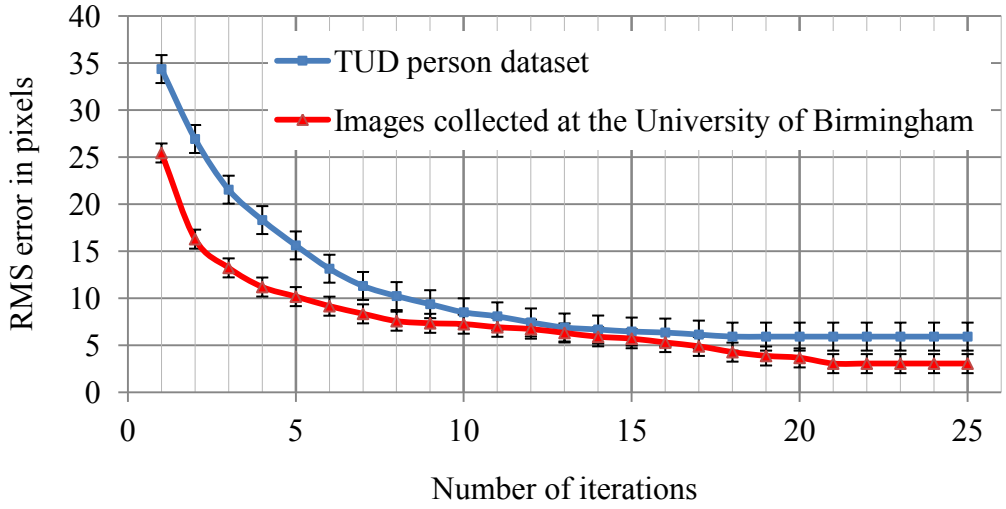


Fig. 7.36. The differences in pixels between the model instances and the corresponding previously unseen images of pedestrians; the blue plot shows the results generated for the TUD person dataset and the red plot shows the results generated for the images collected at the University of Birmingham. The vertical error bars represent one standard error of the mean of a set of measurements.

The images selected from the TUD person data set cover a range of configurations, views and pose variation. The images of pedestrians gathered at the University of Birmingham accommodate a range of contexts, poses and a greater degree of variation than the images of pedestrians selected from the TUD data set and further it provides a grearter variation in weather and lighting condition. The red plot in Fig. 7.36 converges to a slightly smaller residual error than the search shown in blue plot. This suggests that the TUD person dataset [WAN07] is more demanding and might indicate to that the training dataset not properly covers all the variation between people. There is a significant difference between the two curves of 1 SD which suggests that the significance is greater than the 63% suggested by the error bars for each point on the curve. The error bars for the curves of the models generated for the TUD person dataset and the dataset collected at the University of Birmingham do not overlap at large part; this means that it would take an extreme deviation that occurs in the data for both points for the curves to coincide at that point. The

curves are very clearly separated and because this degree of separation exists at several points along the curves there is a very high confidence that the difference is significant.

An evaluation of the ESAM for pedestrian image interpretation was performed with a further 6 sets of pedestrian training images, separate to those introduced in subsection 7.5.2. There are three sets of 90 training images and three sets of 120 training images in each set. The images in each set are different. An independent set of 80 unseen images of pedestrians was used to test the generated models of the 6 sets of pedestrians. The performance of ESAM interpretation for the 6 sets of pedestrians is shown for up to 35 iterations in Fig. 7.37. These convergence curves represent the difference in pixels between the ESAM instances of images of pedestrian's interpretation and previously unseen images of pedestrians. This difference metric criterion is defined in Equation 7.12 by the square root of the mean of the sum of the squared errors. The vertical error bars represent unit standard error of the mean of the measurements across the experiments conducted; this measure is defined in Equation 7.10. The data of the plots in Fig. 7.37 represents the square root of the mean of the sum of the squared differences between the pixel values of the images created by the ESAM interpretation and the pixel values of the previously unseen images for the same images as used for the interpretation and not in training.

It is observed from the curves in Fig. 7.37 that the optimum residual error rate for Set 3 of 90 images, as highlighted by the light green line (largely hidden behind the light orange line), is similar to that for Set 6 of 120 images (light orange line). This suggests that the difference in the number of training images is less important than the choice of images used. Each curve converges to a similar error rate after 25 iterations. The difference between the errors on each curve is less than

the standard deviation for each curve meaning that the differences are not significant. There is a very small difference between the curves of 1 SD which suggests that the significance is lesser than the 63% suggested by the error bars for each point on the curve. The error bars for the models created for the different six sets of pedestrian images overlap at all points; this means that there is no deviation in the data for both points for the curves.

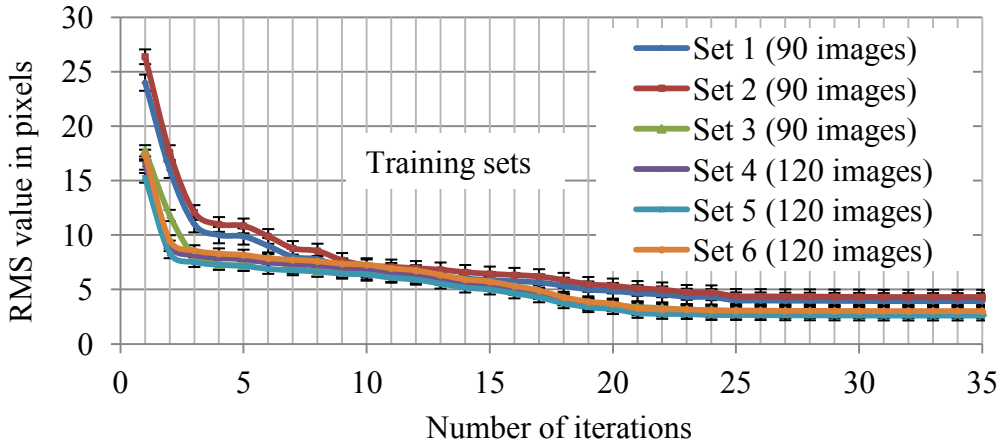


Fig. 7.37. The differences in pixels between the reconstructed model instances and the corresponding previously unseen images of pedestrians as a function of training dataset size and iterations in ESAM interpretation. The vertical error bars represent one standard error of the mean of a set of measurements.

The interpretation error using images of pedestrians with puschairs or bicycles is shown in Fig. 7.38. The RMS error, as defined in Equation 7.12, is shown for up to 25 iterations for each curve for 150 previously unseen pedestrains with objects of each type. Each curve converges to a similar error rate after 25 iterations. The difference between the errors on each curve is less than the standard deviation of the results generated for each curve. The vertical error bars represent unit standard error of the mean of the data across the experiments conducted; this measure is defined in Equation 7.10. The data represents the RMS value computed using Equation 7.12 between the pixel values of the images created by the ESAM interpretation and the image pixel values of the previously unseen images.

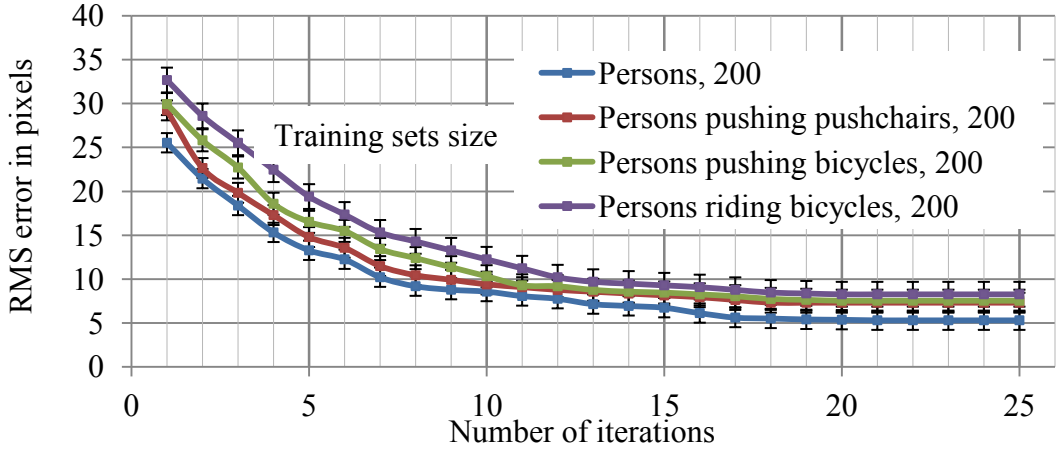


Fig. 7.38. The pixel-to-pixel differences between the reconstructed instances and the corresponding previously unseen images of people combined with other objects as a function of training dataset size and iterations in ESAM interpretation. The vertical error bars represent one standard error of the mean of a set of measurements.

Fig. 7.38 indicates that the models of objects that vary in form achieved a good degree of interpretation. There is a small difference between the curves of 1 SD which suggest that the significance is less than the 63% suggested by the error bars for each point on the curve. The error bars of the adjacent curves for the models created for pedestrians alone and pedestrians pushing pushchairs do not overlap at set of points which means that the difference is possibly significant. On the other hand, there is an overlap between the error bars of the adjacent curves for the models created for pedestrians pushing pushchairs, pedestrians pushing bicycles and persons riding bicycles, which means that the difference is possibly not significant. Therefore, there is no deviation that occurs in the data for both points for the curves to coincide at that point. The curves are not clearly separated and the overlapping exists at several points along the curves and so the difference is not significant.

The RMS error convergence curves for up to 25 iterations of ESAM interpretation of vehicle images using the datasets introduced in Section 7.5 are shown in Fig. 7.39. The RMS error between the ESAM interpretation of vehicle images and the previously unseen vehicle images is computed in pixels as defined in Equation 7.12. The error bars show unit standard error of the mean of the data across the experiments conducted; this measure is defined in Equation 7.10. The data in Fig. 7.39 represents the square root of the mean of the sum of the squared differences between the pixel values of the images created by the ESAM interpretation and the pixel values of the previously unseen images for the same images as used for the interpretation and not in training.

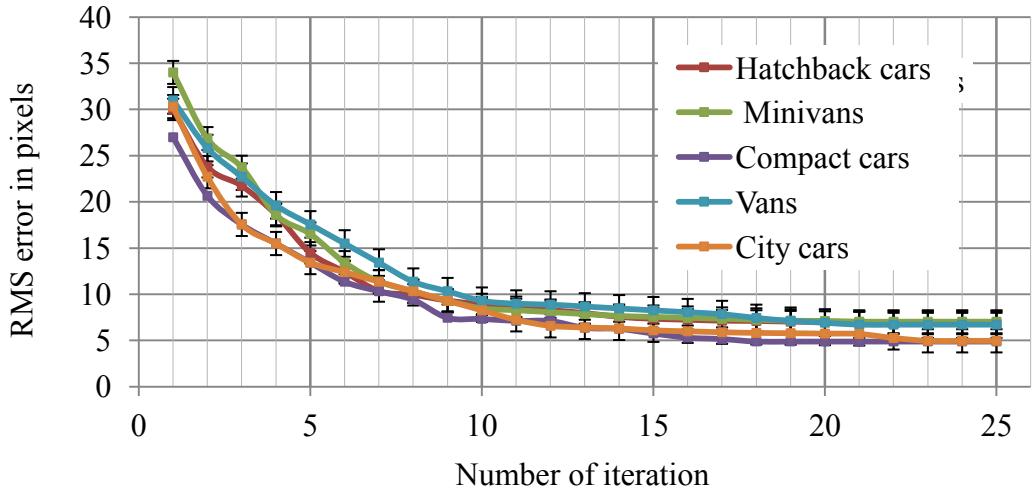


Fig. 7.39. The pixel to pixel differences values between the reconstructed ESAM instances of vehicles and the corresponding unseen vehicle images as a function of training dataset size and iterations in ESAM interpretation. The vertical error bars represent one standard error of the mean of the data.

There is a small difference as visually judged between the interpretation RMS errors of the various vehicle types considered after 25 iterations. In Fig. 7.39, the smallest RMS value between the image pixel values of the vehicles created by the ESAM interpretation of vehicles and the pixel values of the previously unseen images of vehicles is approximately 5 pixels. There is a small difference between the curves of 1 SD which suggests that the difference is not significant and

less than the 63% suggested by the error bars for each point on the curve. There is a significant overlap between the error bars of the curves created for the models of vehicles types which means that there is no significant deviation in the data for both points of the curves and because this degree of overlapping exists along the curves the difference is not significant

The curves in Figs. 7.35 to 7.39 show that the ESAM converges to a relatively small RMS error rates after 15 iterations, that the ESAM is able to represent and interpret images of objects with a wide variety of appearances.

To obtain a quantitative evaluation of the performance of the ESAM compared to AAM, we trained a model on 120 pedestrian images and tested it on a different set of 80 images. An AAM and an ESAM were created to identify pedestrians. The dataset used to train and evaluate these models is described in Section 7.5. The images are of varying complexity of foregrounds and backgrounds along with pedestrians exhibiting a degree of variation in pose. The models used in these experiments hold 97.5% of the total variance with 18 modes of variation. The ESAM and AAM interpretation errors using images of pedestrians is presented for up to 30 iterations in Fig. 7.40. These convergence curves represent the RMS error, defined in Equation 7.12, which computes the difference of pixel values between the model instances of a model trained on 120 pedestrian images and evaluated on a set of 80 previously pedestrians. The vertical error bars in Fig. 7.40 show one standard error of the mean of the measurements across the experiments conducted; this measure is defined in Equation 7.10. The data of the curves represents the square root of the mean of the sum of the squared errors between the pixel values of the re-constructed images created by the ESAM and AAM interpretation and the pixel values of the unseen images.

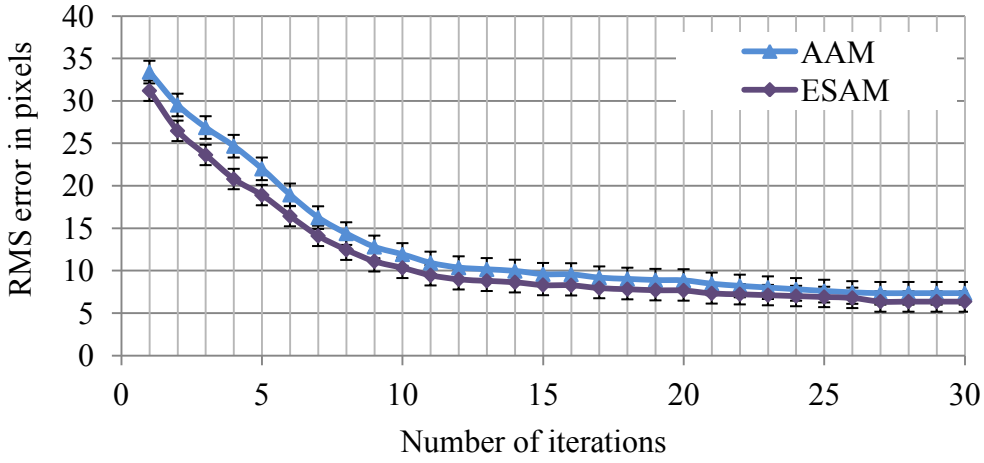


Fig. 7.40. Convergence curves of the mean of square root of the sum of the squared difference between the source image pixels and the final model instance of pedestrian image interpretation for the ESAM and the AAM. The vertical error bars represent one standard error of the mean of a set of measurements.

There is a small difference between the two curves of 1 SD which suggests that the significance is lesser than the 63% suggested by the error bars for each point on the curve. The error bars for the curves of ESAM and AAM overlap at several points and the degree of separation only exists at few points along the curves. Thus, there is no extreme deviation that occurs in the data for both points for the curves to coincide at that point. The curves show that both methods achieve a similar degree of the accuracy of the fit between the image and its interpretation. The difference between the two curves of ESAM and AAM interpretations is of low significance. However, for a small part of the curve this difference is significant and indicates that the ESAM can achieve a better interpretation with images of pedestrians. Also, the ESAM converges more rapidly than the AAM at the first five iterations.

The number of points sampling the boundary of objects has an effect on the speed with which a model can be built and an image interpreted. The model building process requires the identification of three parameters: 1) the number of model points, 2) the number of modes and 3) the num-

ber of pixels used to represent the texture. The search time for generating and matching a sequence of ESAM instances to an unseen image increases linearly with the number of key points. The number of key points needed to satisfactorily represent the boundary of an object depends on the complexity of the object. The number and position of the key points should be adequate to fully describe the object modelled and changes in pose. Too few points can lead to an inadequate representation due to under sampling the shape. Too many points will increase the computational cost. The number of modes should be chosen carefully such that a sufficient amount of object variation is captured, as described in subsection 5.6.4. The number of pixels representing each model relies on the boundary structure. The complexity of ESGM and ESAM search processes are presented in subsections 6.2.2 and 6.3.3, respectively.

The training algorithm of the ESAM from large datasets requires a large amount of training time. The texture represented by the pixel intensity contributes significantly to the discrimination of the fitting results. The computation time per iteration level of the ESAM interpretation algorithm in Fig. 6.3 is 26 milliseconds. This value was measured on several different images and the mean value taken. The tests were performed on an Intel core i3-2310M™ CPU running at 2.10GHz.

7.7 Summary

The non-interpolated and interpolated cue detectors have presented a high potential for identifying cue points within images of pedestrians. The ESGM interpretation method of the ESE curve has provided a high precision representation and been shown to be effective in identifying the points along the boundary of objects. The ESAM has achieved a similar degree of interpretations accuracy as the AAM.

Chapter 8 CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

In this Thesis, the development of a generic model-based approach with a single interpretation scheme has been presented as it was evolved from being based on a single axis to composite axes. The interpretation scheme has been applied to a range of objects that vary in structure and configuration. A set of adaptable strategies have been investigated in the proposed model-based interpretation. These strategies are:

1. A fixed orientation cue detector for generating cues as a basis for pedestrian detection and a variant to the pedestrian cue detector for generating cues for vehicle images,
2. An interpolated axis detector for generating composite axes as a basis for detecting pedestrians with pushchairs and bicycles,
3. A method for key point generation,
4. An extended superellipse appearance model representation.

The interpolated cue detector for generating composite axes and guiding the interpretation of complex objects has been shown to be appropriate for interpreting images of pedestrians not associated with pushchairs or bicycles. Specific conclusions related to each of the above strategies are detailed below.

8.1.1 Cue Detector

The Maximum Likelihood Ratio (MLR) criteria was shown to be reliable at detecting cues for pedestrians alone and in small groups with various degrees of occlusion, with simple and com-

plex backgrounds. The precision and recall rates for cue detection show a high degree of sensitivity and specificity and only a small number of false alarms are generated. A pedestrian detection rate of 0.95 and a false positive rate of 0.20 at an optimal operating point were reported on an evaluation test set of over 700 images.

A change of geometry in the mask for pedestrian cue detection and a simple change to the search strategy were shown to be effective in distinguishing vehicles in images. The number of missed images and the number of false detections were in single figures.

8.1.2 Axis Detector

A variant method of cue detection using an interpolated MLR detector to generate composite axes was presented. The effective detection of cues using this interpolated MLR operator was demonstrated on pedestrians stood alone or in a group, walking, pushing pushchairs or bicycles and people riding bicycles. The precision and recall rates for the detection of composite axes show a high degree of sensitivity and specificity. From a test set of over 700 images of people a true positive rate of 0.94 and a false positive rate of 0.21 at an optimal operating point were reported by this Omni directional axis detector. The false positive and false negative error rates are as low as those for the fixed orientation detector when used for pedestrian detection.

Identification of the axis was very sensitive to the position and width of the component an object. Variation of the form of a pedestrian was handled such that the position of the axis was changed from the centre line due to the position of the accessories.

8.1.3 Key Point Detector

The formulation of MLR criteria as an edge detector was effective in locating edge points on a selected set of search paths so that the key points were appropriately identified on the potential boundary of objects, such as pedestrians, that do not possess reliable landmarks. This simplifies the detection of model points over the demands for detecting landmark points as required in other model-based schemes of image interpretation.

8.1.4 Extended Superellipse Appearance Model

The Extended Superellipse Appearance Model (ESAM) was able to represent a large degree of variation in the objects in the training set using an ESE and proved to be an effective representation for the interpretation system. It is also demonstrated that it is not necessary to detect landmarks to model an object and that boundary sampling produces a reliable representation of an object that can be applied to a broad range of objects without changing the processing strategy.

The concepts developed have been shown to be effective in identifying pedestrians in video sequences without a need to track the movement of the pedestrian. The method developed is able to process video images of 640 x 480 pixels at better than video rate on a modest specification computer such as Intel core i3-2310M™ CPU running at 2.10GHz. The convergence curves for the interpretation systems of the geometric and appearance models show that high levels of reliability and specificity were achieved with low error rates in each case.

The ability to select between variant models for a pedestrian, a pedestrian pushing a pushchair and a pedestrian pushing or riding a bicycle demonstrates the potential to select models in a wider

context and to reason about interpretation strategies. In the wider context there will be a much larger set of alternative models and this will present a serious challenge to model selection.

The versatility of the modelling and interpretation methods presented is demonstrated by the way that it could be adapted to vehicle interpretation using a limited number of different vehicle body shapes to identify each vehicle reliably. With all five of the vehicles modelled the ability to reliably select the most appropriate model with a simple Bayesian classifier and log likelihood estimates on Gaussian distributed data was demonstrated. It might not be a simple matter to extend this strategy to a larger range of vehicle types although a reasonable range of body types were considered.

This approach has potential for widespread application in pedestrian monitoring for safety security surveillance, industrial inspection and could potentially be extended to create 3D avatars without the need for markers.

8.2 Future Work

Further works related to each of the above strategies are introduced below.

8.2.1 Cue Detection

The generalisation of the cue augmentation to a wider range of objects, where the selection criteria could be learnt and form a part of a model would significantly extend the application of model-based image interpretation. A particular challenge would be to make this aspect model-based.

8.2.2 Axes Detection

Further work is needed to learn the appropriate geometric design of the mask to avoid the need for prior knowledge. Further work is needed to automate how to search for the appropriate number of local axes points from the composite axes. This search process might form a part of a model that is likely better to reduce the search space and would be possible lead to a more robust and efficient model. In the future work there is a need for models to identify different objects cues that might be a way to avoid many of the ad hoc operations such as response clustering that are a feature of the cue detection process.

8.2.3 Key Point Detection

The ability to automate the selection of an appropriate number of key points for each model would reduce the size of models and make interpretation more efficient. This might involve increasing the number of search paths for some cases to determine if the additional search paths improve the representation of the shape.

8.2.4 Geometric and Appearance Models

Further study is needed to better distinguish between pedestrians and trees. The extension of the ESAM algorithm to 3D images would be valuable. For vehicle interpretation it might possible to extend the application to learn any vehicle body shape.

REFERENCES

[ABD79] Abdou, I.E. and Pratt, W. (979) "Quantitative design and evaluation of enhancement/thresholding edge detectors". In *Proc. of the IEEE*, 67(5):53-763.

[AIX03] Aixut, T., De Meneses, Y., Bourgeois, F. and Jacot, J. (2003) "Constraining deformable templates for shape recognition". *Quality Cont. by Artif. Vis.,* 17-26.

[ALH91] Al-Hinai, K. G., Khan, M. A. and Canas, A. A. (99) "Enhancement of sand dune texture from Landsat imagery using difference of Gaussian filter". *International Journal of Remote Sensing*, 12(5): 1063-1069.

[AME14] Amerehie, H., Dianat, R. and Keynia, F. (20 4) "A New Method to Improve the Difference of Gaussian Feature Detector". *IJSCE*, 4(4):1-7.

[AND08] Andriluka, M., Roth, S. and Schiele, B. (2008) "People-tracking-by-detection and people-detection-by-tracking". In *CVPR 2008,* 1-8.

[ARB11] Arbelaez, P., Maire, M., Fowlkes, C. and Malik, J. (20) "Contour detection and hierarchical image segmentation". *PAMI*, *33*(5), 898-916.

[BAL81] Ballard, D. H. (98) "Generalizing the Hough transform to detect arbitrary shapes". *Pattern Recogn*., 13(12):111–122.

[BAR81] Barr, A.H. (98) "Superquadrics and angle-preserving transformations". *IEEE Comp. Graph. and App.*, 1(1):11-23.

[BAT05] Batur, A.U. and Hayes, M.H. (2005) "Adaptive active appearance models". *Image Processing, IEEE Trans.*, 14(11):1707–1721.

[BAU95] Baumberg, A. (995) "*Learning deformable models for tracking human motion*". PhD Thesis, University of Leeds, 1-152.

[BEN11] Benfold, B. and Ian, R. (2011) "Unsupervised learning of a scene-specific coarse gaze estimator". In *ICCV 2011,* 2344-2351.

[BLU67] Blum, H. (967) "A transformation for extracting new descriptors of shape". In *Models for the perception of speech and vision forms,* MIT Press, 362-380.

[BOS07] Bosch, A., Zisserman, A., and Munoz, X. (2007) "Image classification using random forests and ferns". In *ICCV 11th,* 1-8.

[BOU09] Bourdev, L. and Malik, J. (2009) "Poselets: Body part detectors trained using 3D human pose annotations". In *ICCV*, 1365–1372.

[BOU10] Bourdev, L. Maji, S. Brox, T. and Malik, J. (20 0) "Detecting people using mutually consistent poselet activations". In *ECCV*, 168–181.

[BRE94] Bregler, C. and Omohundro, S. (994) "*Surface Learning with Applications to Lip Reading*". International Computer Science Institute.

[BRE01] Breiman, L. (200) "Random forests". *Machine Learning*, 45(1):5-32.

[BRE07] Breiman, L. and Cutler, A. (2007) "*Random forests-classification description*". https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

[CAN86a] Cannon, S.R., Jones, G.W., Campbell, R. and Morgan, N.W. (986) "A computer vision system for identification of individuals". *In Proc. IECON,* 1:347-351.

[CAN86] Canny, J. (986) "A computational approach to edge detection". *IEEE Trans. PAMI*, (6):679-698.

[CHA14] Chai, T. and Draxler, R. R. (20 4) "Root mean square error (RMSE) or mean absolute error (MAE)?– Arguments against avoiding RMSE in the literature". *Geoscientific Model Development*, *7*(3): 1247-1250.

[CHA11] Chang, C.C and Lin, C.J (2011) "LIBSVM: A library for support vector machines". *ACM Transactions on Intelligent Systems and Technology*, 2(3):1-39.

[CHE09] Chen, J., Huang, H., Tian, S., and Qu, Y. (2009) "Feature selection for text classification with Naïve Bayes". *Expert Systems with Applications*, 36(3):5432-5435.

[CIP03] Cipolla, R. Stenger, B. Thayananthan, A and Torr, P. (2003) "Hand tracking using a quadric surface model". In *Mathematics of Surfaces, LNCS 276:* 129–141.

[COH94] Cohen, F. and Wang, J. (994) "Part I: Modeling image curves using invariant 3-D object curve models-a path to 3-D recognition and shape estimation from image contours". *IEEE Trans. PAMI,* 16(1):1-12.

[COO92] Cootes, T.F., Taylor, C.J., Cooper, D.H. and Graham, J. (992) "Training models of shape from sets of examples". In *Proc. BMVC'92, pri n e r-Verlag*, 9-18.

[COO92a] Cootes, T. F. and Christopher J. T. (992) "Active shape models— 'smart snakes". In *BMVC92*, Springer London, 266-275.

[COO94] Cootes, T.F., Hill, A., Taylor, C.J. and Haslam, J. (994) "The use of active shape models for locating structures in medical images". *IVC*, 12(6):355-365.

[COO95] Cootes, T. F., Taylor, C. J., Cooper, D. H. and Graham, J. (995). "Active shape models-their training and application". *CVIU*, 61(1): 38-59.

[COO98] Cootes, T.F., Edwards, G.J. and Taylor, C.J. (998) "Active appearance models". In *Proc. 5th ECCV,* 2:484-498.

[COO01a] Cootes, T.F., Christopher, J. and Taylor, C.J. (2001) "Statistical models of appearance for medical image analysis and computer vision". *Med. Im., SPIE*, 236-248.

[COO01b] Cootes, T.F., Edwards, G.J. and Taylor, C.J. (200) "Active appearance models". *IEEE Trans. PAMI,* 23(6):681-685.

[COO04] Cootes, T.F. and Taylor, C.J. (2004) "*Statistical models of appearance for computer vision*". Tech. rep., University of Manchester, Wolfson Image Analysis Unit.

[DAL05] Dalal, N. and Triggs, B. (2005) "Histograms of oriented gradients for human detection*".* In *CV R' 05, IEEE Computer Society Conference*, 1:886-893.

[DAV06] Davidson, W. and Abramowitz, M. (2006) "Molecular expressions microscopy primer: Digital image processing-difference of gaussians edge enhancement algorithm". *Olympus America Inc., and Florida State University*.

[DAV09] David, A. (2009) " *tat i t ical Model : heory and rac tice"*. Cam. Univ. Press, 26.

[DEM05] De Meneses, Y.L., Roduit, P., Luisier, F. and Jacot, J. (2005) "Trajectory analysis for sport and video surveillance". *ELCVIA*, 5(3):148-156.

[DOL09] Dollar, P., Tu, Z., Perona, P. and Belongie, S. (2009) "Integral Channel Features". *Proc. British Machine Vision Conference*.

[DOL12] Dollar, P., Wojek, C., Schiele, B., and Perona, P. (20 2) "Pedestrian detection: An evaluation of the state of the art". *IEEE Trans. PAMI,* 34(4): 743-761.

[DUR08] Dura, E., Bell, J. and Lane, D. (2008) "Superellipse fitting for the recovery and classification of mine-like shapes in sides can sonar images". *Oceanic Eng., IEEE Journal.* 33(4):434-444.

[DUT99] Duta, N., Sonka, M. and Jain, A.K. (999) "Learning shape models from examples using automatic shape clustering and Procrustes analysis". In *Proc. IPMI,* 370-375.

[DUT09] Dutta, S. and Chaudhuri, B.B. (2009) "A color edge detection algorithm in RGB color space". In *ARTCom'09,* 337-340.

[EDW74] Edwards, A. W. (974) "The history of likelihood". *International Statistical Review/Revue Internationale de Statistique*, 9-15.

[ELE11] Eleyan, A. and Demirel, H. (20) "Co-occurrence matrix and its statistical features as a new approach for face recognition". *Turkish Journal of Electrical Engineering and Computer Sciences*, 19(1), 97-107.

[ENZ09] Enzweiler, M. and Gavrila, D. M. (2009) "Monocular pedestrian detection: Survey and experiments". *IEEE Trans. PAMI,* 31(12), 2179-2195.

[ESS07] Ess, A., Leibe, B. and Van Gool, L. (2007) "Depth and appearance for mobile scene analysis". In *Proc. ICCV'07 IEEE th Int. Conf.,* 1-8.

[EVA06] Evans, A.N. and Liu, X.U. (2006) "A morphological gradient approach to color edge detection". *IEEE Trans. Im. Proc.,* 15(6):1454-1463.

[EVE06] Everingham, M., Zisserman, A. and Williams, C.K. et al. (2006), "The 2005 Pascal visual object classes challenge". *In MI Challenges, Evaluating Predictive Uncertainty, Visual Obj. Classification and Recognising Tectual Entailment*, 117-176.

[EVE10] Everingham, M., Van Gool, L., Williams, C., Winn, J. and Zisserman, A. (2010) "The PASCAL Visual Object Classes (VOC) Challenge". *Int'l J. Comput. Vi .*, 88(2): 303-338.

[FAR10] Farenzena, M., Bazzani, L., Perina, A. and et al. (20 0) "Person re-identification by symmetry-driven accumulation of local features". In *CV R' 0,* 2360-2367.

[FEL08] Felzenszwalb, P., McAllester, D. and Ramanan, D. (2008) "A discriminatively trained, multiscale, deformable part model". In *CVPR 2008*, 1–8.

[FIS81] Fischler, M.A. and Bolles, R.C. (98) "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *CACM*, 24(6): 381-395.

[FRE91] Freeman, W.T. and Adelson, E.H. (99) "The design and use of steerable filters". *IEEE Trans. PAMI*, 13(9): 891-906.

[FUK90] Fukunaga, K. (990) "*Introduction to statistical pattern recognition*". 2nd Ed. Academic Press Prof., Inc., San Diego, CA.

[GKI14] Gkioxari, G., Hariharan, B., Girshick, R. and Malik, J. (20 4) "Using k-poselets for detecting people and localizing their keypoints". In *CVPR 2014*, 3582–3589.

[GAL06] Gallou, L.S., Breton, G., Garcia, C. and Séguier, R. (2006) "Distance maps: A robust illumination preprocessing for active appearance models". *VISAPP*, 6:35-40.

[GAL13] Gall, J. and Lempitsky, V. (20 3) "Class-specific Hough forests for object detection". *Decision Forests for Computer Vis. and Med. Image Anal.*, 143-157.

[GAO10] Gao, X., Su, Y., Li, X. and Tao, D. (20 0)."A review of active appearance models". *IEEE Trans. on Sys., Man, and Cyber., Part C (Apps and Rev.)*, *40*(2): 145-158.

[GAR65] Gardiner, M. (965) "The superellipse: a curve that lies between the ellipse and the rectangle". *Scientific American*, 213(3): 222-232.

[GAU93] Gauch, J.M. and Pizer, S.M. (993) "The intensity axis of symmetry and its application to image segmentation". *IEEE Trans. PAMI*, 15(8): 753-770.

[GER05] Gerónimo, D., Sappa, A., López, A. and Ponsa, D. (2007) "Adaptive image sampling and windows classification for on-board pedestrian detection". In *Proc. of the Int. Conf. on Com. Vis. Sys.,* 39, 1-10.

[GRA07] Gray, D., Brennan, S. and Tao, H. (2007) "Evaluating appearance models for recognition, reacquisition and tracking". In *IEEE Int. Workshop on PETS,* 1-7.

[GRO88**]** Gross, A. D. and Boult, T. E. (988) "Error of fit for recovering parametric solids". *In Proc. IEEE Int. Conf. Comput. Vis.*, 690–694.

[GUP10] Gupta, K., Kang, S. and Sandhu, P. S. (20 0) "Comparison of Resilient Back-propagation & Fuzzy Clustering Based Approach for Prediction of Level of Severity of Faults in Software Systems.

[GUY00] Gu, Y.H. and Tjahjadi, T. (2000) "Coarse-to-fine planar object identification using invariant curve features and B-spline modeling". *Patt. Recogn.*, 33(9): 1411-1422.

[HAR73] Haralick, R., Shanmugam, K. and Dinstein, I. (973) "Textural features for image classification". *IEEE Trans. on Systems, Man and Cybernetics*, 3(6): 610-621.

[HEA97] Heap, T. and Hogg, D. (997) "Improving specificity in pdms using a hierarchical approach". In *BMVC*, 80–89.

[HEA96] Heap, A. J. and Hogg, D. C. (996) "Extending the point distribution model using polar coordinates". *IVC*, 14(8): 589-599.

[HUW06] Hu, W., Min, H., Xue, Z. and et al. (2006) "Principal axis-based correspondence between multiple cameras for people tracking". *Trans. PAMI,* 28(4): 663-671.

[JOO05] Jorgensen, B. and Rajeswaran, J. (2005) "A Generalization of Hotelling's T^2". *Communications in Statistics-Theory and Methods*, 34(11): 2179-2195.

[KIN89] King, G. (989) "*Unifying political methodology: The likelihood theory of statistical inference*". University of Michigan Press.

[KIR90] Kirby, M. and Sirovich, L. (990) "Application of the Karhunen-Loeve procedure for the characterization of human faces". *IEEE Trans. PAMI*, 12(1):103-108.

[KOT97] Kotcheff, A. and Taylor, C. (997) "Automatic Construction of Eigenshape Models by Genetic Algorithms". In *Proc. Int. Conf. on Inf. Proc. in Med. Im.,* 1-14.

[KRI04] Krivic, J. and Solina, F. (2004) "Part-level object recognition using superquadrics". *CVIU*, 95(1):105-126.

[KRI12] Krizhevsky, A., Sutskever, I. and Hinton, G. (20 2) "Imagenet classification with deep convolutional neural networks". In *Advances in neural information processing systems*, 1097–1105.

[KYP08] Kyprianidis, J. and Dollner, J. (2008) "Image abstraction by structure adaptive filtering". In *TPCG*, 51–58.

[LEE03] Lee, S.M., Abbott, A.L., Clark, N.A. and Araman, P.A. (2003) "Spline curve matching with sparse knot sets: applications to deformable shape detection and recognition". In *Proc. 29th Annual Conf. IEEE IECON'03,* 2:1808-1813.

[LEI04] Leibe, B., Leonardis, A. and Schiele, B. (2004) "Combined object categorization and segmentation with an implicit shape model" In *Workshop on Statistical Learning in Computer Vis. ECCV*, 2(5): 1-16.

[LIM13] Lim, J., Zitnick, C. and Dollár, P. (20 3) "Sketch tokens: A learned mid-level representation for contour and object detection". In *Proc. of the IEEE Conf. on CVPR,* 3158-3165.

[MAR08] Martins, P.A. (2008) "*Active appearance models for facial expression recognition and monocular head pose estimation*". MSc Thesis, University of Coimbra.

[MAS93] Masuda, T., Yamamoto, K. and Yamada, H. (99 3) "Detection of partial symmetry using correlation with rotated-reflected images". *Patt. Recogn.*, 26(8): 1245-1253.

[MCI96] McInerney, T. and Terzopoulos, D. (996) "Deformable models in medical image analysis: a survey". *Medical image analysis*, 1(2): 91-108.

[MET91] Metaxas, and Terzopoulos, D. (99) "Dynamic 3D models with local and global deformations: deformable super quadrics". *IEEE Trans. PAMI*, 13(7):703-714.

[MOH01] Mohan, A., Papageorgiou, C. and Poggio, T. (200) "Example-based object detection in images by components". *IEEE Trans. PAMI,* 23(4): 349-361.

[MOS92] Moses, Y. and Ullman, S. (992) "Limitations of non model-based recognition schemes". In *ECCV'92,* 820-828.

[MUN06] Munder, S. and Gavrila, D.M. (2006) "An experimental study on pedestrian classification". *IEEE Trans. PAMI*, 28(11): 1863-1868.

[MUR95] Murase, H. and Nayar, S.K. (995) "Visual learning and recognition of 3-D objects from appearance". *IJCV*, 14(1): 5-24.

[NAN02] Nanda, H. and Davis, L. (2002) "Probabilistic template based pedestrian detection in infrared videos". In *Proc. IEEE y mp. IV'02*, 1: 15-20.

[NGU15] Nguyen, D., Tran, M. and Yeung, S. (20 5) "An MRF-Poselets Model for Detecting Highly Articulated Humans". In *Proc. of the Int. Conf. on Comput. Vis.,* 1-9.

[NOV87] Novak, C.L. and Shafer, S.A. (987) "Color edge detection". In *Proc. DARPA Image Understanding Workshop,* 1: 35-37.

[OUY13] Ouyang, W. and Wang, X. (20 3) "Joint deep learning for pedestrian detection". In *Proceedings of the IEEE Int. Conf. on Computer Vision,* 2056-2063.

[OVE08] Overett, G., Petersson, L., Brewer, N., Andersson, L. and Pettersson, N. (2008) "A new pedestrian dataset for supervised learning". In *Intell. Vehicle Sym.,* 373-378.

[PAN08] Pandey, S., and Bright, C. L. (2008). "What Are Degrees of Freedom?". Social Work Research, *32*(2):119-128.

[PAP00] Papageorgiou, C. and Poggio, T. (2000) "A trainable system for object detection". *IJCV*, 38(1), 15-33.

[PAR94] Park, J., Metaxas, D. and Young, A. (994) "Deformable models with parameter functions: application to heart-wall modeling". In *Proc. CVPR,* 437-442.

[PAR08] Park, M., Lee, S., Chen, P.C. and et al. (2008) "Performance evaluation of state-of-the-art discrete symmetry detection algorithms". In *Proc. CVPR*, 1-8.

[PEN86] Pentland, A. (986) "Perceptual organization and the representation of natural form". *Artifi. Intel.*, 28(3): 293-331.

[PEN87] Pentland, A. (987) "Recognition by parts". In *Proc. First Int. Conf. Comp. Vis.*, 612-620.

[PEN90] Pentland, A. (990) "Automatic extraction of deformable part models". *IJCV*, 4(2): 107-126.

[PEN94] Pentland, A., Moghaddam, B. and Starner, T. (994) "View-based and modular eigenspaces for face recognition". In *Proc. CVPR,* 84-91.

[PIL97] Pilu, M. and Fisher, R. B. (997) "Part segmentation from 2D edge images by the MDL criterion". *IVC*, 15(8): 563-573.

[PIL99] Pilu, M. and Fisher, R. B. (999) "Training PDMs on models: the case of deformable superellipses". *Pattern recognition letters*, 20(5): 463-474.

[PRA91] Pratt, W.K. (99)"*Digital Image Processing*". New York, NY: Wiley-Interscience, 491-556.

[PYC01] Pycock, D., Pammu, S., Goode, A. J. and Harman, S. A. (200) "Robust model-based signal analysis and identification". *Pattern Recognition*, 34(11): 2181-2199.

[RAJ92] Raja, N.S. and Jain, A.K. (992) "Recognizing geons from superquadrics fitted to range data". *IVC*, 10(3): 179-190.

[ROD65] Rodieck, R. W. (965) "Quantitative analysis of cat retinal ganglion cell response to visual stimuli". Vision Research, 5 (11): 583–601.

[ROS95] Rosin, P. and West, G. A. (995) “Curve segmentation and representation by superellipse”. *Inst. Electr. Eng. Proc. Vis. Image Signal Process.*, 142(5): 280–288.

[ROS93] Rosin, P. L. (993) “A note on least square fitting of superellipse”. *Pattern Recognition Lett.*, 14(1): 799–808.

[RUZ01] Ruzon, M.A. and Tomasi, C. (200) “Edge, junction, and corner detection using color distributions”. *IEEE Trans. PAMI,* 23(11): 1281-1295.

[SAB12] Sabri, M., Garakani, M.F. and Lotfinejad, M.M. (20 2) “Model-based interpretation using anatomic landmarks for defining shape variations of the lung and heart”. *Journal of Academic and Applied Studies*, 2(12):42-51.

[SAL99] Salous, M. (999) “*Context-based image transmission*”. PhD Thesis, University of Birmingham.

[SAT08] Satpathy, A., Eng, H. L. and Jiang, X. (2008) “Difference of Gaussian edge-texture based background modeling for dynamic traffic conditions”. In *Advances in Visual Computing,* 406-417, Springer Berlin Heidelberg.

[SAU11] Sauer, P., Cootes, T.F. and Taylor, C.J. (20) “Accurate regression procedures for active appearance models”. In *BMVC*, 1-11.

[SCH09] Schwartz, W.R., Kembhavi, A., Harwood, D. and Davis, L.S. (2009) “Human detection using partial least squares analysis”. In *Proc. 12th ICCV’09,* 24-31.

[SHA95] Shardanand, U. and Maes, P. (995) “Social information filtering: Algorithms for automating ’word of mouth’”. In *Proceeding of ACM CHI*, 210–217.

[SHE07] Shechtman, E. and Irani, M. (2007) “Matching local self-similarities across images and videos”. In *2007 IEEE Conference on CVPR, 1-8*.

[SIM02] Sim, T., Baker, S. and Bsat, M. (2002) "The CMU pose, illumination and expression database". In *Auto. Face and Gest. Recogn., Proc.5th IEEE Int. Conf.,* 46-51.

[SOL90] Solina, F. and Bajcsy, R. (990) "Recovery of parametric models from range images: The case for superquadrics with global deformations". *IEEE Trans. PAMI*, 12(2): 131-147.

[SOZ94] Sozou, P., Cootes, T., Taylor, C. and Di-Mauro, E. (994) "A non-linear generalisation of PDMs using polynomial regression". In *Proc. BMVC*, 94: 397–406.

[TAN12] Tang, D., Liu, Y. and Kim, T. (20 2) "Fast pedestrian detection by cascaded random forest with dominant orientation templates". *In BMVC,* 1-11.

[TAU91] Taubin, G. (99) "Estimation of planar curves, surfaces and nonplanar space curved defined by implicit equations with applications to range and edge image segmentation". *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(11): 1115–1138.

[TIA15] Tian, Y., Luo, P., Wang, X., and Tang, X. (20 5) "Pedestrian detection aided by deep learning semantic tasks". In *CVPR 2015*, 5079-5087.

[TIL00] Tillett, R., McFarlane, N. and Lines, J. (2000) "Estimating dimensions of free-swimming fish using 3D point distribution models". *CVIU*, 79(1): 123-141.

[TRA96] Trahanias, P.E. and Venetsanopoulos, A.N. (996) "Vector order statistics operators as color edge detectors". *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Trans.* 26(1): 135-143.

[TRA08] Tran, D. and Forsyth, D. (2008) "Configuration estimates improve pedestrian finding". In *NIPS,* 1529–1536.

[TUR91] Turk, M. and Pentland, A. (99) "Eigenfaces for recognition". *JoCN*, 3(1): 71-86.

[VIO04] Viola, P.A. and Jones, M. J. (2004) "Robust real-time Face Detection". *IJCV*, 57(2):137–154.

[VIO05] Viola, P., Jones, M.J. and Snow, D. (2005) "Detecting pedestrians using patterns of motion and appearance". *IJCV*, 63(2): 153-161.

[WAN90] Wang, H.P., Hewgill, D.E. and Vickers, G.W. (990) "An efficient algorithm for generating B-spline interpolation curves and surfaces from B-spline approximations". *Communications in applied numerical methods*, 6(5): 395-400.

[WAN07a] Wang, Q., Garrity, G., Tiedje, J., and Cole, J. (2007) "Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy". *Applied and environmental microbiology*, 73(16): 5261-5267.

[WAN07] Wang, L., Shi, J., Song, G. and Shen, I.F. (2007) "Object detection combining recognition and segmentation". In *Comp. Vis. –ACCV 2007*, 189-199.

[WIN11] Winnemöller, H. (20) "XDoG: advanced image stylization with eXtended Difference-of-Gaussians". In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering,* 147-156.

[WAN02] Wang, H., Wang, W., Yang, J. and Yu, P. S. (2002) "Clustering by pattern similarity in large data sets". In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, 394-405.

[WAN12] Wang, S., Li, W., Wang, Y., Jiang, Y., Jiang, S. and Zhao, R. (20 2) "An improved difference of gaussian filter in face recognition". *J. of Multimedia*, 7(6), 429-433.

[WAT08] Watts, J. D., and Lawrence, R. L. (2008) "Merging random forest classification with an object-oriented approach for analysis of agricultural lands". *The In. Archives of the Photogrammetry, Rem. Sen. and Spatial Inf. Sci.*, *37*: 579-582.

[WIL08] Williams, J. K., and Abernethy, J. (2008) "Using random forests and fuzzy logic for automated storm type identification". In *AMS 6th Conference on Artificial Intelligence Applications to Environmental Science*.

[WOJ09] Wojek, C., Walk, S. and Schiele, B. (2009) "Multi-cue onboard pedestrian detection". In *CVPR 2009,* 794-801.

[WON89] Wong, K. H., Law, H.H.M. and Tsang, P.W.M. (989) "A system for recognising human faces". *In Acoustics, Speech, and Signal Processing, ICASSP-89.,* 1638-1642.

[WUB05] Wu, B. and Nevatia, R. (2005) "Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors". In *10th IEEE ICCV'05,* 1: 90-97.

[WUB06] Wu, B. and Nevatia, R. (2006) "Tracking of multiple, partially occluded humans based on static body part detection". In *IEEE Conference CVPR,* 1: 951-958.

[XUM98] Xu, M. and Pycock, D. (998) "The Multiscale Medial Response of Grey-level Images". In *BMVC,* 1-11.

[YEN03] Yen, T. J. (2003) "*A qualitative profile-based approach to edge detection,* Doctoral dissertation, New York University.

[ZHA03] Zhang, Y. (2003) "Experimental comparison of superquadric fitting objective functions". *Pattern Recognition Letters*, 24(1): 2185–2193.

[ZHA03a] Zhang, X. and Rosin, P. L. (2003) "Superellipse fitting to partial data". *Pattern Recognition*, 36(3):743-752.

[ZHE06] Zheng, G., Ballester, M.Á., Styner, M. and Nolte, L.P. (2006) "Reconstruction of patient-specific 3D bone surface from 2D calibrated fluoroscopic images and point distribution model". In *Proc. MICCAI'06, Springer Berlin–Heidelberg*, 25-32.

[ZHO99] Zhou, L. and Kambhamettu, C. (999) "Extending superquadrics with exponent functions: modeling and reconstruction". In *Proc. CVPR,* 2:1-20.

[ZHO00] Zhou, L. and Kambhamettu, C. (2000) "Hierarchical structure and nonrigid motion recovery from 2D monocular views". In *Proc. CVPR*, 2: 752-759.

[ZHO02] Zhou, L. and Kambhamettu, C. (2002) "Representing and recognizing complete set of geons using extended superquadrics". In *Proc. the 16th ICPR*, 3: 713-718.

[ZHO97] Zhou, P. and Pycock, D. (997) "Robust statistical models for cell image interpretation". *IVC*, 15(4): 307-316.