Detection, Tracking and Classification of Vehicles in Urban Environments

Zezhi Chen

Submitted in partial fulfilment of the requirements of
Kingston University for the degree of
Doctor of Philosophy

June 2012

Kingston University London

Supervision:
Professor Tim Ellis

Digital Imaging Research Centre
School of Computing and Information Systems
Faculty of Science, Engineering and Computing
Kingston University
Penrhyn Road
Kingston Upon Thames
United Kingdom
KT1 2EE

http://dirc.kingston.ac.uk

Abstract

The work presented in this dissertation provides a framework for object detection, tracking and vehicle classification in urban environment. The final aim is to produce a system for traffic flow statistics analysis.

Based on level set methods and a multi-phase colour model, a general variational formulation which combines Minkowski-form distance L_2 and L_3 of each channel and their homogenous regions in the index is defined. The active segmentation method successfully finds whole object boundaries which include different known colours, even in very complex background situations, rather than splitting an object into several regions with different colours. For video data supplied by a nominally stationary camera, an adaptive Gaussian mixture model (GMM), with a multi-dimensional Gaussian kernel spatio-temporal smoothing transform, has been used for modeling the distribution of colour image data. The algorithm improves the segmentation performance in adverse imaging conditions. A self-adaptive Gaussian mixture model, with an online dynamical learning rate and global illumination changing factor, is proposed to address the problem of sudden change in illumination.

The effectiveness of a state-of-the-art classification algorithm to categorise road vehicles for an urban traffic monitoring system using a set of measurement-based feature (BMF) and a multi-shape descriptor is investigated. Manual vehicle segmentation was used to acquire a large database of labeled vehicles form a set of MBF in combination with pyramid histogram of orientation gradient (PHOG) and edge-based PHOG features. These are used to classify the objects into four main vehicle categories: car, van (van, minivan, minibus and limousine), bus (single and double decked) and motorcycle (motorcycle and bicycle). Then, an automatic system for vehicle detection, tracking and classification from roadside CCTV is presented. The system counts vehicles and separates them into the four categories mentioned above. The GMM and shadow removal method have been used to deal with sudden illumination changes and camera vibration. A Kalman filter tracks a vehicle to enable classification by majority voting over several consecutive frames, and a level set method has been used to refine the foreground blob.

Finally, a framework for confidence based active learning for vehicle classification in an urban traffic environment is presented. Only a small number of low confidence samples need to be identified and annotated according to their confidence. Compared to passive learning, the number of annotated samples needed for the training dataset can be reduced significantly, yielding a high accuracy classifier with low computational complexity and high efficiency.

ABSTRACT

Declaration

The candidate declare that all material contained in the dissertation is his own original work, and that any references to or use of other sources have been clearly acknowledged within the text.

Part of the research presented in this dissertation has appeared in the following publications:

1. Zezhi Chen, Tim Ellis and Sergio Velastin. Confidence based active learning for vehicle classification in urban traffic. The fourth Chilean Workshop on Pattern Recognition (CWPR'2012), 12-16th November 2012, Valparaiso, Chile. To appear.
2. Zezhi Chen, Tim Ellis and Sergio Velastin. Vehicle Detection, Tracking and Classification in Urban Traffic. *15th IEEE Annual Conference on Intelligent Transportation Systems,* Sept. 16-19, 2012, Anchorage, Alaska, USA, pp. 951-956.
3. Zezhi Chen, Tim Ellis. Self-adaptive Gaussian mixture model for urban traffic monitoring system. *The 11th IEEE International Workshop on Visual Surveillance (ICCV'2011 Workshop)*, November 13th, Barcelona, Spain, pp. 1769-1776.
4. Zezhi Chen, Tim Ellis. Multi-shape descriptor vehicle classification for urban traffic. *International Conference on Digital Image Computing: Techniques and Applications*, Dec 6, 2011, Noosa, Australia. pp. 456-461.
5. Zezhi Chen, Tim Ellis and Sergio Velastin. Vehicle type catergorization: A comparison of classification schemes. *14th IEEE Annual Conference on Intelligent Transportation Systems,* Oct. 5-7, 2011, the George Washington University, Washington, DC, USA. pp. 74-79.
6. Zezhi Chen, Nick Pears, Michael Freeman and Jim Austin. Background subtraction in video using recursive mixture models, spatio-temporal filtering and shadow removal. *Proceedings of 5th International symposium on Visual Computing (ISVC)*, Las Vegas, NV, USA, Nov. 30-Dec. 2, 2009. Lecture Notes in Computer Science, Vol. 5876, 2009, pp. 1141-1150.
7. Zezhi Chen, Nick Pears, Michael Freeman and Jim Austin. Road vehicle classification using support vector machines. *Proceedings of IEEE International Conference on Intelligent Computing and Intelligent Systems*, Nov. 20-22, 2009, Shanghai, China, Vol. 4, pp. 214-218.
8. Benjamin Gorry, Zezhi Chen, Kevin Hammond, Andy Wallace, and Greg Michaelson. Using mean-shift tracking algorithm for real-time tracking of moving images on an autonomous vehicle testbed platform. *International Journal of Computer Science and Engineering*, 1(3):165-170, 2007.
9. Benjamin Gorry, Zezhi Chen, Kevin Hammond, Andy Wallace and Greg Michaelson. Using mean-shift tracking algorithms for real-time tracking of moving images on an autonomous vehicle testbed platform. *The International Conference on Intelligent Robotics and Manufacturing Automation (IRMA07).* Venice, Italy,

November 23-25, 2007, Proceedings of the World Academy of Science, Engineering and Technology 25, World Academy of Science, Engineering and Technology, 2007, pp. 356--361.

10. Zezhi Chen, Andrew M Wallace. Active segmentation and adaptive tracking using level sets. *British Machine Vision Conference 2007*, University of Warwick, pp. 920-929, September 2007.

11. Zezhi Chen and Andrew M Wallace. Improved object tracking using an adaptive colour model. *The 6th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, EZhou Hubei, China, August 2007. pp. 280-294.

12. Zezhi Chen, Zsolt L Husz, Iain Wallace and Andrew M Wallace. Video object tracking based on a Chamfer distance transform. *IEEE International Conference on Image Processing*, San Antonio, Texas, USA, Sept. 2007. III 357-360.

13. Greg Michaelson, Andy Wallace, Kevin Hammond, Armelle Bonenfant, Zezhi Chen and Benjamin Gorry. Analysing and deploying resource-bound AV software in Hume. *Technology Centre (SEAS DTC), Annual Technical Conference, Edinburgh, July 2007.* pp. A2.

14. Armelle Bonenfant, Zezhi Chen, Kevin Hammond, Greg Michaelson, Andy Wallace, and Iain Wallace. Towards resource-certified software: a formal cost model for time and its application to an image-processing example. *ACM Symposium on Applied Computing (SAC '07), Seoul, Korea, March 11-15*, Seoul, Korea, 2007, pp. 1307-1314.

15. Greg Michaelson, Andy Wallace, Kevin Hammond, Iain Wallace, Armelle Bonenfant, and Zezhi Chen. Toward resource certified image processing software. *Systems Engineering for Autonomous Systems, Defence Technology Centre (SEAS DTC), Annual Technical Conference, July 2006, Edinburgh, Conference Proceedings*, UK MoD, 2006, pp. A15.

Acknowledgement

Firstly, I would like to thank my supervisor Professor Tim Ellis for his effort, help, motivation and invaluable guidance. It would not have been possible to complete this dissertation without his supervising. The experience I have gained from my research has been of great benefit in my academic career. Thanks to Professor Sergio A. Velastin and Dr James M. Orwell for helpful discussion of ideas and technicalities. I would also like to thank Professor Andrew M. Wallace and Professor Gregory J. Michaelson for being my supervisors during my work and study at Heriot-Watt University. Thanks to Professor Chris Taylor and Dr Sue Astley at the University of Manchester. They helped me to gain a deep understanding of how to apply signal processing theory to the area of image processing. Thanks to Dr Nick Pears and Professor Jim Austin for helpful discussions and comments when I worked at the University of York. I would also like to thank my colleagues and friends Fei Yin, Damien R. Simonnet, Raul A. Herrera Acuna, Jay Kiruthika, Eric Oppong, Roshan A. Welikala and Sameer Bhandari in our office in promoting a stimulating and welcoming academic and social environment. Thank you all very much! There are many more people, who are not named in person, to whom I am grateful.

I gratefully acknowledge the Royal Borough of Kingston for the providing of the video dataset, funding under the UK Technology Strategy Board project, CLASSAC, and support from Cybula Ltd. I also acknowledge the provision of data sequence (the indoor scene videos/images) from the Caviar project at the University of Edinburgh, Baris Sumengen for his basic level set function implementation Matlab toolbox (http://barissumengen.com/level_set_methods/index.html), Canu et al. for their "SVM and Kernel Methods" Matlab toolbox (http://asi.insa-rouen.fr/enseignants/~arakoto/toolbox/index.html), and Chung and Lin for their "LIBSVM" toolbox (http://www.csie.ntu.edu.tw/~cjlin/libsvm/). I am grateful for use of the i-LIDS dataset provided by the UK Home Office which was used for evaluation and comparison.

Last, but no means least, I would like to thank my wife and daughter for their constant support and encouragement while completing my doctoral research.

ACKNOWLEDGEMENT

To my wife and daughter

ACKNOWLEDGEMENT

Glossary of terms

2D	two-dimensional
3D	three-dimensional
ACC	accuracy
AdaBoost	adaptive boost
AS	active segmentation
AUC	area under the ROC curve
AutoVDCS	automatic vehicle detection and classification system
BAL	final classification boundary of active learning
BET	classification boundary from entire training data set
BG	background
CART	classification and regression tree
CCTV	closed-circuit television
COM	complete code
CSM	colour space model
CUDA	compute unified device architecture
CV	Chan-Vese active contour
CVV	Chan-Vese vector
DBW	dynamic bin-width
DT	distance transform
DTR	detection rate
ECOC	error correcting output coding
EKF	extended Kalman filter
EL	end line
ENO	essentially nonoscillatory
EPHOG	edge based pyramid HOG
F1	F-measure
FAR	false alarm rate
FG	foreground
FG1	foreground segmentation before shadow removal
FG2	foreground segmentation after shadows removal
FN	false negative
FP	false positive
FPR	false positive rate
GD	Gaussian distribution
GMM	Gaussian mixture model
GPU	graphics processing unit
GVF	gradient vector flow
HJ	Hamilton-Jacobi
HJS	hybrid joint-separable
HOG	histogram of orientation gradients
HSV	Hue-Saturation-Value
ICDM	illumination-invariant change detection model
ICM	iterated conditional modes

i-LIDS	the image library for intelligent detection systems
INS	initial negative samples
IP	internet protocol
IPHOG	intensity based pyramid HOG
IPS	Initial random positive samples
ITS	intelligent transportation systems
JACC	joint accuracy
JC	Jaccard coefficient
JDTR	joint detection rate
JFNR	joint FNR
JFPR	joint FPR
JPRE	the joint PRE
JREC	the joint REC
KNN	K-nearest neighbour
LAD	linear discriminant analysis
LCS	selected low confidence samples
LDA	linear discriminate analysis
LLF	local Lax-Friedrichs
MBC	model-based classification
MBF	measurement-based features
MB-LBP	multi-block local binary pattern
MCC	Matthews correlation coefficient
MCMC	Markov chain Monte Carlo
MD	Mahalanobis distance
MDGKT	multi-dimensional Gaussian kernel density transform
MHKF	multiple-hypothesis Kalman filter
ML	middle line
MofQ	median of quotient
MSE	mean square error
NCDT	normalised Chamfer distance transform
NGD	non-Gaussian distribution
NSP	negative samples
OVA	one-vs-all
OVO	one-vs-one
PCA	principal component analysis
PDE	partial differential equation
pdf	probability density function
PETS	performance evaluation of tracking and surveillance
PHOG	pyramid histogram of orientation gradients
PPV	positive predictive value
PRE	precision
PSP	positive samples
REC	recall
RF	random forests
RIFT	rotation invariant feature transform
RLSC	regularized least squares classification
ROC	receiver operating characteristic
SAGMM	self-adaptive Gaussian mixture model
SIDBW	smooth interpolated DBW histogram

SIFT	scale-invariant feature transform
SL	start line
SPE	specificity
SPV	support vectors from entire training data set
std	standard deviation
SVM	support vector machines
TN	true negative
TNR	true negative rate
TP	true positive
TPR	true positive rate
WENO	weighted ENO
wkNN	A weighted *k*-nearest neighbour
YCbCr	YCbCr colour space
ZHGMM	Zivkovic-Heijden Gaussian mixture model

List of figures

LIST OF FIGURES

List of tables

Contents

1. Introduction

1.1. Scope of the intelligent transportation systems

Over the past several decades, surveillance techniques have matured drastically. Analogue tapes and security personnel are being replaced with Internet Protocol (IP) technology, leveraging digital video cameras, remote access, and intelligent analytics. This evolution provided organizations with significant opportunities to improve surveillance systems and to reduce operating costs.

Intelligent Transportation Systems (ITS) comprise several combinations of communication, computer and control technology developed and applied in the domain of transport to improve system performance, transport safety, efficiency, productivity, and level of service, environmental impacts, energy consumption, and mobility. ITS represents the next step in the evolution of the entire transportation system. These technologies include the latest in computer, electronic, communication and safety systems. ITS can be applied to vast transportation infrastructure of highways, streets, bridges, tunnels, railways, ports and airport infrastructures, as well as to a growing number of vehicles, including cars, buses, trucks and trains (Sitavancova and Hajek, 2009).

Interests in ITS came from the problems caused by traffic congestion, and from a possible synergy of new information technology for simulation, real-time control, and communications networks. Traffic congestion has been increasing worldwide as a result of increased motorisation, urbanisation, population and economy growth, and changes in population density. Congestion reduces efficiency of transportation infrastructure and increases travel time, air pollution, and fuel consumption, which also led to increased costs.

Benefits of ITS (Bertini et al. 2005):

- Urban management systems can potentially reduce delays.
- Motorway management systems can reduce the occurrence of crashes.
- Freight management systems reduce costs.
- Transit management systems may reduce travel times, improved traffic flow.
- Incident management systems potentially reduce incident.

In urban environments, several monitoring objectives can be supported by the application of ITS, including the detection of traffic violations (e.g., illegal turns and driving against traffic flow) and the identification of road users (e.g., vehicles, motorbikes, and pedestrians). However, using general purpose surveillance cameras (i.e., monocular), this challenge is demanding. The quality of surveillance data is generally poor, this is usually caused by the following problems: the range of operational conditions (e.g., night time, inclement, and changeable weather), low camera resolution and low camera angle, high degree of occlusion, a limited amount of visual details of road user (less feature, size variation), high density of vehicles. In addition, the clutter on the streets increases the complexity of scenes. Therefore, the system requires robust techniques. Furthermore, the ITS generally requires real time processing, which further constrains the complexity of the proposed algorithms. Traffic analysis on highways appears to be less challenging than in the urban environment.

From an application perspective, the main technical challenge is the diversity of camera views and operating conditions in traffic surveillance. In addition, a large variety of

observation objectives, e.g., vehicle counting, classification, incident detection, or traffic rule enforcement, can be useful. This condition has generated a large diverse body of work, where it is difficult to perform direct comparison between the proposed algorithms (Buch et al, 2011). The main technical challenge in urban environments includes occlusions and dense traffic. There are several solutions for occlusion handling in highway scenes (Hsieh, et al. 2006; Su et al., 2007; Kanhere et al., 2008) for relatively sparse traffic, which cannot necessarily be transferred to urban environments. Shadows also cause motion silhouettes of vehicles to merge. Without a histogram of different vehicles moving in different lanes, as described in (Hsieh, et al. 2006) for a highway environment, the splitting of silhouettes is much harder. The current main bottleneck of surveillance is the limitation of human resources for observing millions of cameras. Automatic pre-processing allows efficient guidance for the operators to pick cameras to view and accumulate statistics, with the aim to improve traffic flow. Video cameras have been deployed for a long time for traffic and other monitoring purposes, because they provide a rich information source for human understanding. Video analytics may now provide added value to those cameras by automatically extracting relevant information and easing the bottleneck of operators viewing all cameras.

1.2. Research aims and objectives

The project will investigate the detection of vehicles moving in urban environments. It will investigate segmentation algorithms that are able to cope with the problem of both sparse and dense traffic streams, where it may be problematic to separate individual vehicles. The algorithms for vehicle segmentation should be robust to changing environmental conditions associated with weather and varying illumination. Robust operation of the algorithms will be validated on an appropriate set of CCTV datasets of urban traffic.

This will form part of the detection methodology monitoring urban road traffic to extract real-time estimates of journey times and network conditions using newly available abundant sources of data. This information will be used to guide more effective traffic management and control, and as a basis for on-line traffic status for route guidance to traveller-orientated systems. Video analysis and traffic modeling will be treated as complementary processes that will be used continuously to refine and update each other within an appropriate statistical framework.

This thesis focuses on video analysis from urban traffic management cameras. To address the bottleneck of surveillance systems, algorithms are to be developed for the detection, tracking and classification of vehicles from 12 CCTV cameras currently installed in Kingston town centre (the snapshot frames of 8 videos are given in Figure 6.2). 5 hours of video in different weather conditions (cloudy, raining), time (day, evening and night) are used. This provided different challenges to highway monitoring. Four general classes are identified for the vehicle classification:

- Car (car and taxi)
- Van (van, minivan, minibus, and limousine)
- Bus (single decked and double decked)
- Motorcycle (motorcycle and bicycle)

The work presented in this thesis contributes to the field of traffic visual surveillance through development and improvement of object detection, tracking and classification algorithms. The objectives are:

- An active segmentation algorithm based on level set methods and a multi-phase colour model;
- An adaptive object tracking algorithm using active segmentation and mean shift tracking algorithms;
- An adaptive Gaussian mixture model, with a multi-dimensional Gaussian kernel spatio-temporal smoothing transform for segmenting moving road vehicles;
- An algorithm with a dynamically adaptive learning rate and a model for global illumination change of the background;
- A novel approach to camera calibration utilizing calibrated images mapped by Google Earth to provide accurately-surveyed scene geometry;
- A comparison of methods for categorising vehicle types;
- A hybrid automatic vehicle detection, tracking and classification for traffic flow statistics analysis;
- A framework of confidence based active learning for vehicle classification in an urban traffic environment.

1.3. Outline of the dissertation

The remainder of this dissertation is organised as follows: Chapter 2 is a literature review of relevant previous researches on object detection, tracking and classification, particularly to traffic monitoring in urban and highway scenes together with generic visual surveillance technology. Active contour segmentation and tracking algorithms are investigated in Chapter 3. A generalized active contour model for multi-channel and multi-phase colour image segmentation and an adaptive object-tracking algorithm is proposed. Chapter 4 presents an adaptive Gaussian mixture model using a multi-dimensional spatio-temporal Gaussian kernel smoothing transform for background modeling in moving object segmentation applications. The model update process can robustly deal with slow light changes (from clear to cloudy or vice versa), blurred images, camera vibration in strong wind, and difficult environmental conditions, such as raining. In order to deal with sudden global illumination changes and camera automatic gain control, a dynamic adaptive learning rate and global illumination change factor are introduced into the GMM model. Extensive experiments of road vehicle type categorisation using manually created ground truth data are given in Chapter 5. In this Chapter measurement based features, pyramid HOG features and model based features are used. A new camera calibration method is also presented in the chapter. Chapter 6 describes the framework for an automatic system for vehicle detection, tracking and classification. Traffic flow statistics analysis and system evaluation is given in the chapter. Chapter 7 presents a framework for confidence based active learning for vehicle classification in an urban traffic environment. Chapter 8 concludes the thesis and gives a discussion of future problems still to be addressed. The datasets used in the thesis are described in the Appendix.

2. Literature Review

2.1. Introduction

The application of image processing and computer vision techniques to the analysis of video sequences of traffic flow offers considerable improvements over the existing methods of traffic data collection and road traffic monitoring. In recent years, there has been an increased scope for the automatic analysis of urban traffic activities. The aim of this chapter is to present a review of image/video processing techniques for traffic surveillance applications. It mainly considers three basic tasks: object detection, object tracking and object classification.

2.2. Object detection

One popular problem in the field of computer vision is object detection (or segmentation). Segmentation is an important technique used in image processing to identify the objects in the images and videos. Figure 2.1 shows some examples under a range of realistic conditions. It is very challenging to segment well defined vehicles. Figure 2.1(a) shows an image with very strong shadow. Figure 2.1(b) and (c) show images with strong highlight reflection. And Figure 2.1(d) shows an image with headlight reflection at night.

2.2.1. Spatial difference

2.2.1.1. Level set method

Since their introduction as a means of front propagation and their first application to edge-based segmentation in the early 90's, level set methods have become increasingly popular as a general framework for image segmentation (Cremers et al., 2007). The idea behind level sets or implicit active contours, or deformable models, for image segmentation is quite simple. The user specifies an initial guess for the contour, which is then moved by image driven forces to the boundaries of the desired objects. In such models, two types of forces are considered - the internal forces, defined within the curve, are designed to keep the model smooth during the deformation process, while the external forces, which are computed from the underlying image data, are defined to move the model toward an object boundary or other desired features within the image. The main problem of the level set representation lies in the fact that a level set function is restricted to the separation of two regions. As soon as more than two regions are considered, the level set idea loses parts of its attractiveness. Among level set methods, statistical region-based methods is more interesting, because the contour is not evolved by fitting to local gradient information (as with Snakes) but rather by fitting statistical models to intensity, colour, texture or motion within each of the separated regions. The respective cost functions tend to have fewer local minima for most realistic images. As a consequence, the segmentation schemes are far less sensitive to noise and to varying initialization.

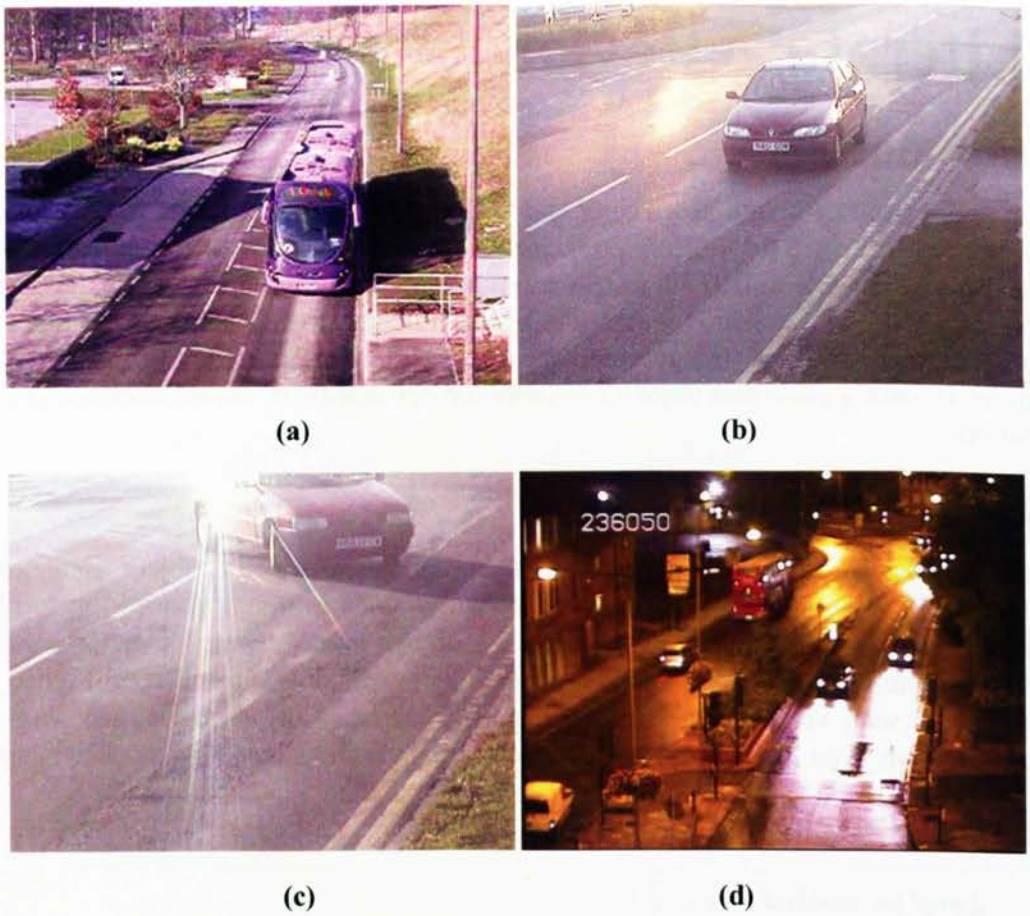


Figure 2.1. Example frames under realistic condition. (a) Sunny conditions with strong shadows. (b) Reflection on the floor. (c) Reflection on the car. (d) Headlight reflections at night.

Chan and Vese (2001) proposed a model for active contours to detect objects in a given image, based on the Mumford-Shah curve evolution functional (Mumford and Shah, 1989). The model can detect objects whose boundaries are not necessarily defined by a gradient. Other related works are Xu and Prince (1997, 1998) and Siddiqi et al. (1998) on active contours and segmentation, Zhao et al. (1998) on shape reconstruction from unorganized points, and Paragios and Deriche (1998, 1999) where a probability based geodesic active region model, combined with classical gradient based active contour techniques, has been proposed.

Chung and Vese (2009) introduced a multi-layer method, where the authors used more than one level of a level set to represent the discontinuity of the image, inspired by modeling island dynamics for epitaxial growth. Brox and Weickert (2004) addressed the difficulty of image segmentation methods based on the popular level set framework to handle an arbitrary number of regions. In this paper, the authors proposed a minimization strategy using the level set framework for minimizing the energy used in the paper of Zhu and Yuille (1996). A piecewise level set method was introduced by Lie et al. (2006), which used one level set for multiphase segmentation by representing each phase with a different constant value. The graph-cut algorithm was utilized for multiphase Mumford-Shah model proposed by Li and Tai (2007), and Bae and Tai (2008). Jung et al. (2007) introduced a relaxed model for multiphase segmentation using Γ-convergence analysis. More related work can be found at Pan et al. (2006). Esther et al. (2010) proposed an approach integrating geometric scene knowledge into a level-set for vehicle tracking. Prisacariu and Reid (2009, 2012) formulated

a probabilistic framework for simultaneous region-based 2D segmentation and 2D to 3D pose tracking, using a known 3D model. The foreground region was delineated by the zero-level-set of a signed distance embedding function, and an energy over this region and its immediate background surroundings based on pixel-wise posterior membership probabilities. The method could be extended for multi-camera and multi-object tracking. However, initialisation is the difficult part of this problem. Sandberg et al. (2010) focused on multiphase segmentation with a new regularization term that yields an unsupervised segmentation model. A functional that simultaneously chooses a reasonable number of phases while segmenting the image has been proposed. By using the scale measure of the phases in the regularization term, bigger objects are preferred to be identified while segmentation is driven by the intensity fitting term. An active contour with selective local or global segmentation algorithm proposed by Zhang and Zhang et al. (2010) over geodesic active contours and Chan-Vese model (Chan and Vese, 2001). Fang and Chan (2007) incorporated shape prior knowledge into geodesic active contours for detecting partially occluded objects.

2.2.1.2. Graph cut method

In spatially discrete approaches, the pixels of the image are usually considered as the nodes of a graph, and the aim of segmentation is to find cuts of this graph which have a minimal cost. One of the most common applications of graph cut segmentation is extracting an object of interest from its background. If there is any knowledge about the object shape (i.e. a shape prior), incorporating this knowledge helps to achieve a more robust segmentation. Every pixel of the image is represented by a node in the graph. The vertices between nodes and sources are set to a weight related to the data (data constraint). Sources represent the labels for a pixel (in this case, the foreground and the background). Vertices between nodes are used to introduce a smoothing constraint to avoid very small foreground or background regions. The graph cut completely separates the source and sink nodes and leaves the nodes connected to either source or sink node to indicate that this pixel corresponds to the respective label. The advantage of graph cuts is that the solution for this optimization problem can be found in polynomial time. Optimization algorithms for these problems include greedy approaches such as the Iterated Conditional Modes (ICM) (Besag, 1986) and continuation methods such as Simulated Annealing (Geman and Geman, 1984) or Graduated Non-convexity (Blake and Zisserman, 1987). Kolmogorov and Zabih (2004) indicated what energy function could be minimized via graph cut. A characterization of the energy functions that could be minimized by graph cuts has been given in the paper. Boykov and Funka (2006) presented a framework for extracting objects from images/volumes. Veksler (2008) proposed a star shape prior for graph cut image segmentation. The major assumption was that the centre of the star shape was known.

The approach to object extraction can be seen as a unifying framework for segmentation that combines many good features of the previous methods like snakes or active contours while providing efficient and robust global optimization applicable to N-D problems. Recent applications used graph cuts for scene understanding from moving vehicles in the paper Sturgess et al. (2009). But the graph cuts framework is very flexible with initialization.

2.2.1.3. Mean shift segmentation method

Mean shift is a nonparametric estimator of density which has been applied to image and video segmentation. Image segmentation refers to identifying homogenous regions in the image. The mean shift algorithm is a powerful clustering technique, which is based on an iterative scheme to detect modes in a probability density function. It has been utilized for

image segmentation by seeking the modes in a feature space composed of spatial and colour information (Kaftan et al., 2008). More specifically, mean shift algorithms estimate the local density gradient of similar pixels. These gradient estimates are used within an iterative procedure to find the peaks in the local density. All pixels that are drawn upwards to the same peak are then considered to be members of the same segment (Wang et al., 2004). The application of the mean shift algorithm to colour image segmentation has been proposed by Comaniciu and Meer (1997). Dementhon (2000, 2002) applied mean shift on a 3D lattice to get a spatio-temporal segmentation of the video. A hierarchical strategy was employed to cluster pixels of a 3D space-time video stack, which were mapped to 7D feature points. A general nonparametric technique was proposed (Comaniciu and Meer, 2002) for the analysis of a complex multimodal feature space and to delineate arbitrarily shaped clusters in it. Bailer et al. (2005) have applied mean shift algorithm to segment colour image sequences in Luv colour space. Tao et al. (2007) proposed a colour image segmentation method by incorporating the advantages of mean shift segmentation and normalized cut partitioning methods (Shi and Malik, 2000). This method required low computational complexity and was therefore feasible for real-time image processing.

2.2.2. Temporal difference

2.2.2.1. Simple method

A background model can be used to accumulate information about the scene background of a video sequence. The model is then compared to the current frame to identify, provided that the camera is stationary. This concept works very well for computer implementation but leads to problems with slow-moving traffic. Any vehicle should be considered foreground, but stationary objects are missed due to lack of motion. The simplest background reference image is formed by either taking an image without vehicles, or by a mathematical or exponential average of successive images. The detection is then achieved by subtracting the reference image from the current image (Bertozzi et al., 1997; Park et al., 2007). This difference image is thresholded and used as the foreground mask. However the background can change significantly over time with shadows cast by buildings and clouds, or simply due to changes in lighting conditions. With these changing environmental conditions, the background frame is required to be updated regularly. There are several background updating techniques. The most commonly used are averaging and selective updating. In the averaging model, the background is built gradually by taking the average of the previous background with the current frame. However, although this algorithm has less computational cost, it is likely to produce ghost objects behind moving objects due to the contamination of the background with the appearance of the moving objects (Hoose, 1992; Fathy and Siyal, 1995; Huang and Liao, 2004; Kanhere et al., 2005; Chan and Zhang, 2007; Kanhere and Birchfield, 2008; Ngugyen and Lee, 2008). Gupte at al. (2002) presented algorithms for vision based vehicle detection and classification in highway scenes. The background was updated by taking a weighted average of the current background and the current frame of the video sequence. Processing was done at three levels: raw images, region level and vehicle level. After background subtraction, vehicles were modeled as rectangular patches with dynamic behaviours. The detected vehicles were classified into two categories: car and non-car. In a 20 minute sequence of freeway traffic, 90% of the vehicles were correctly detected and tracked. Of those correctly tracked vehicles, 70% were correctly classified.

2.2.2.2. Single Gaussian model

To improve robustness compared to averaging, a temporal single Gaussian model can be used for every pixel in the background. Instead of using only the mean value for averaging, the variance of the background pixels is additionally calculated. This approach results in a mean image and a variance image for the background model. A new pixel is classified, depending on the position in the Gaussian distribution, which is the statistical equivalent of a dynamic threshold (Kumar et al, 2003; Morris and Trivedi, 2006). Su et al. (2007) proposed a non-Gaussian single thresholding background model for vehicle detection on a motorway.

2.2.2.3. Gaussian mixture model

The Gaussian Mixture Model (GMM) was introduced in the paper by Stauffer and Grimson (1999, 2000). Each pixel is temporally modeled as a mixture of two or more Gaussians and is updated online. The stability of the Gaussian distributions is evaluated to estimate if they are the result of a more stable background process or a short-term foreground process. Each pixel is classified to be background if the distribution that represents it is stable above a threshold. The model can deal with lighting changes and repetitive clutter. The computational complexity is higher than standard background subtraction methods. Veeraraghavan et al. (2002) and Martel-Brisson and Zaccarin (2007) extended the GMM to deal with shadows. Rad and Jamzad (2005) used a Kalman filter and background difference to track, classify and count vehicles on highways. The detected vehicles were classified into three categories: motorcycle and cycle, car, bus and minibus. Messelodi et al. (2005) utilized background subtraction and a feature based tracking methodology. For each detected vehicle, the system was able to describe its path, to estimate its speed and to classify it into seven categories (bicycle, motorcycle, car, van, urban bus, extra-urban bus and lorry). Other GMM methods could be found in the paper of Power and Schooners (2002), KaewTraKulPong and Bowden (2001), Wang and Ma et al. (2009), Zhang and Wu et al. (2008), Johansson et al. (2009), Bloisi and Iocchi (2009), and Buch et al. (2010).

2.2.3. Spatio-temporal difference

Pless (2004, 2005, 2006) and Georg and Pless (2009) utilized a classical relationship between image motion and spatio-temporal image derivatives, where road features could be extracted as image regions that have significant image variation and a motion consistent with its neighbours. The video pre-processing to generate image derivative distributions over arbitrarily long sequences was implemented in real time on standard laptops, and the flow field computation and interpretation involved a small number of 3 by 3 matrix operations at each pixel location. They presented an algorithm for extracting parametric descriptions of roads from motion cues inherent in static video of traffic. Statistics of spatio-temporal derivatives were accumulated from a static video. New energy terms for fitting B-spline snakes to roads by aligning their direction and speed to be maximally consistent with the spatio-temporal derivatives. Due to the constraint on the derivative of the snake, a stable open-ended snake without specifying the location of endpoints was able to be produced to allow it to cover the entire road.

2.2.4. Feature-based object detection

In order to deal with the problem of increasing congestion on freeways, Coifman et al. (1998) applied corner features to detect and track vehicles under shadow and lighting transition conditions. Instead of tracking the entire vehicle, vehicle features were tracked to

make the system robust to partial occlusion. A prototype system implementation was provided. Hasegawa and Kanade (2005) described a vision system that could recognize moving targets such as a mule, sedan, van, truck, pedestrian and others using an 11-dimensional feature vector of an image. The recognition rate was 91.1% under the condition that both the recognition results of type and colour agreed with the operator's judgment. Morris and Trivedi (2006a, 2006b, 2008) presented two different types of visual activity analysis module based on vehicle tracking. The first system was the visual vehicle classifier and traffic flow analyzer module for robust real-time vehicle classification, traffic statistic accumulation, and highway modeling for flow analysis. The second activity analysis module introduced was the path behaviour block, which builds a probabilistic scene motion model in an unsupervised manner for activity analysis. The object detection module determined foreground pixels by using an adaptive background subtraction scheme. The measurements were intended to characterize an object by providing a unique signature of any potential scene object. The measurement vector used here was composed of 17 simple blob features {area, breadth, compactness, elongation, perimeter, convex hull perimeter, length, long and short axis of fitted ellipse, roughness, centroid, 5 image moments}. A weighted *k*-nearest neighbour (wkNN) classifier was used to classify vehicles into eight different types, namely, Sedan, Pickup, SUV, Van, Merged, Bike, Truck and Semi. The wkNN database used for classification was constructed from 10 minutes of hand-labeled training video. The experimental results showed that the system accurately classified vehicles, built a highway model by collecting traffic flow statistics, and categorized live traffic flow in real time.

In the paper of Buch et al. (2009a, 2009b, 2009c), a method was proposed that overcomes limitations in the use of 2D HOG. Full 3D models were used for the object categories to be detected and the feature patches were defined over these models. A calibrated camera allowed an affine transform of the observation into a normalised representation from which "3DHOG" features were defined. A variable set of interest points was used in the detection and classification processes, depending on which points in the 3D model were visible. Experiments on real CCTV data of urban scenes demonstrate the proposed method. Feris et al. (2011) presented an attribute-based vehicle search in crowded surveillance videos. The data was captured from a set of city surveillance cameras using a semi-automatic approach: motion blobs in the input videos were initially detected via background modeling and the blobs in a user-defined region of interest having an acceptable size, aspect ratio, and direction of motion were automatically added to the training set. In order to deal with different vehicle types (bus, track, SUV and car), the motionlet detectors in a shape-free appearance space were learnt, where all training samples were resized to the same aspect ratio.

2.3. Object tracking

Once objects have been detected, the next logical step is to track or classify these detected objects. Tracking has a number of benefits. Firstly, detection is normally quite computationally expensive, so by using tracking, the detection step does not need to be computed for each frame. Secondly, tracking adds temporal consistency to sequence analysis because otherwise, objects may appear and disappear in consecutive frames due to detection failure. Also, tracking can incorporate validity checking to remove false positives from the detection step. Thirdly, if tracking multiple objects, detection of occlusion is made easier, as the occlusion is expected when two or more tracked objects move past each other (Kumar et al., 2008). Tracking techniques can be divided into two main approaches: 2D models with or without explicit shape models and 3D models. For example, Messelodi et al. (2005) presented

a real time vision system to compute traffic parameters. The system uses a combination of segmentation and motion information to localize and track moving objects on the road plane, utilizing a robust background updating, and a feature-based tracking model. Ferecatu and Sahbi (2009) presented an alternative framework for multi-view object matching and tracking based on canonical correlation analysis, but it required an overlapping region. Leotta and Mundy (2011) applied a generic 3D vehicle model that deforms to match a wide variety of passenger vehicles. The model is aligned to images by predicting and matching image intensity edges. Novel algorithms were presented for fitting models to multiple still images and simultaneous detecting and tracking while estimating shape in video.

2.3.1. Kalman filter tracking

The Kalman filter was originally introduced in (Kalman, 1960) and has been successfully used in many applications. The optimal state of a linear time-invariant motion model is estimated, assuming a Gaussian process and measurement noise. The prediction stage of the Kalman filter is used to extrapolate the position of the objects in a new frame based on a constant velocity constraint. The prediction can be associated with new measurements or can be used to trigger detectors. A correction step uses the detection as a measurement to update the filter state. Kalman filters propagate a single object state between frames. It should be clear that the Kalman filter can be an extremely powerful estimation tool. One advantage of this filter is that it combines multiple sources of information in a principled and optimal manner. Specifically, the Kalman filter can incorporate measurements from numerous sources to improve upon the state estimation, regardless of their accuracy or format. Another advantage of the Kalman filter is its recursiveness.

Welch and Bishop (2001) provided a good introduction to the Kalman filter. A two-dimensional token-based tracking system using a Kalman filter was designed to track individual objects under occlusion conditions in the paper (Jung et al. 2001). Messelodi et al. (2005) applied a combination of segmentation and motion information to localize and track moving objects on the road plane, utilizing background subtraction and a feature-based tracking methodology. The initial estimate of the background image is performed as in the configuration step, using the median image of a short frame sequence. Background updating is performed by Kalman filtering the pixels in the same location through time. Tracking is done by combining a simple image processing technique with a 3D extended Kalman fillter and a measurement equation that projects from the 3D model to the image space. No ground plane assumption is made (Dellaert and Thorpe, 1997). In the paper (Du and Yuan, 2009) a Kalman filter was used to predict the possible location of the vehicle in the next frame, and then Gabor wavelet features were used to match points in the predicted region, for accurate location of vehicles. This concept was also used in the papers of Marslin et al. (1991), Morris and Trivedi (2006), Rad and Jamzad (2005).

However, there are some common problems associated with using the Kalman filter. One problem is the requirement for prior knowledge about the process and the measurement procedure. This problem is confirmed by experiments. Specifically, the values of the process and measurement covariance matrices are needed, yet are difficult to obtain in most cases. Iterative methods are proposed to estimate these matrices, however they fail to provide useable results in the cases tested (Funk, 2003). Either more prior knowledge is required or better techniques for determining these matrices need to be developed. Most tests are performed on synthetic images with known motion patterns (constant velocity and constant acceleration) and specified noise levels. Brock & Schmidt (1970) indicated that the main

problem with Kalman filtering is that statistical models are required for the system and the measurement instruments. Unfortunately, they are typically not available, or difficult to obtain.

The extended Kalman filter (EKF) is the nonlinear version of the Kalman filter which linearises about an estimate of the current mean and covariance (Ribeiro, 2004). Yang and Welch (2005) presented a model-based object tracking system using an improved EKF with graphics rendering as the measurement function. The approach can be used to track a rigid object from multiple views in real time. Hatanaka et al. (2011) utilized EKF approach to estimate the position and deformations of a curve as it evolves in the plane for visual tracking.

2.3.2. Particle filter tracking

The particle filter that is known and used in computer vision has historic roots in the fields of mathematics and physics. Particle filters commonly used in computer vision are sometimes referred to as sequential Monte Carlo methods, an extension to the original Monte Carlo technique. Petrovskaya and Thrun (2008) utilized the particle filter method for Bayesian estimation for vehicle tracking in urban environments, and they indicated it is more suitable for multimodal distributions than the EKF. Aghagolzadeh and Seyedarabi (2010) presented vehicle tracking in multi-sensor networks by fusing data in a particle filter framework by fusing several cues including colour, edge, texture and motion constrained by structures in the environment. Fusion of features in a particle filter framework helps to achieve an accurate tracking algorithm in a single view. Particle filters have also been used in conjunction with colour histograms (Nummiaro et al., 2002). More related work can be found at Jaward et al. (2006), Cho et al. (2006) and Rathi et al. (2007). A comparison of the MultiHypothesis Kalman Filter and Particle Filter-based tracking was presented by Bazzani et al. (2009). In this work, two kinds of multi-target tracking approaches have been presented: Multiple-Hypothesis Kalman filter (MHKF) and Particle filter based Hybrid Joint-Separable (HJS) (Lanz, 2006). A comparison of these methods has been undertaken analyzing a recent challenging dataset (PETS 2009), and the results showed the robustness of both approaches. In particular, HJS performs better than MHKF when occlusions occur keeping the identity of the target after occlusion, while MHKF tends to generate a new target ID.

Advantages of Particle Filters

- Under general conditions, the particle filter estimate becomes asymptotically optimal as the number of particles goes to infinity.
- Non-linear, non-Gaussian state update and observation equations can be used.
- Multi-modal distributions are not a problem.
- Particle filter solutions to inference problems are often easy to formulate.

Disadvantages of Particle Filters

- Naive formulations of problems usually result in significant computation times.
- It is hard to tell if you have enough particles.
- The best importance distribution and/or resampling methods may be very problem specific.

2.3.3. Mean shift tracking

The most prominent kernel histogram tracker is the original mean shift tracker (Comaniciu et al., 2000) along with its numerous variants and extensions (Comaniciu et al., 2003; Leung and Gong, 2006; Parameswaran et al, 2006; Cannons and Wildes, 2007; Cannons, 2008; Fan et al., 2007). At a high level, the basic components of a mean shift

tracker that need to be defined are as follows: the histogram representations for the template and candidate regions; the matching metric between histograms; and the method of locating the best candidate match in the current frame. A number of trackers were proposed that extended the basic mean shift tracker. The majority of these extensions were more incremental than radically changing the mean shift framework. Some works, such as Comaniciu et al. (2003) extended their method by introducing predictive Kalman filters to smooth the target tracks.

Some works have attempted to include additional spatial information (Cheng and Yang, 2004; Birchfield and Rangarajan, 2005; Leung and Gong, 2006; Fan et al., 2007, Vilapana and Marques, 2008). Other researchers have considered increasing the speed of the algorithm (Leung and Gong, 2006). The matter of feature spaces and automatically selecting the most discriminative features for tracking has been investigated (Cannons and Wildes, 2007; Avidan, 2007; Yin et al., 2008; Parag et al., 2008). Song and Nevatia (2007) proposed a Markov Chain Monte Carlo (MCMC) based method to segment multiple merged vehicles into individual vehicles with their respective orientation, and then mean shift has been successfully applied to track an image region. The mean shift method presented by Quast and Kaup (2009) used an asymmetric kernel which is retrieved from an object mask instead of using a symmetric kernel like in traditional mean shift tracking. Yi et al. (2008) and Yilmaz (2011) applied a kernel which has the shape of the target object, and with probabilistic estimation of the orientation change and scale adaptation to object tracking. An adaptive bandwidth mean shift algorithm for 2D object tracking was proposed by Chen and Zhou et al. (2008). Khan et al. (2009) combined a subset of scale-invariant point features from SIFT and mean shift together to track an object in a video. Li and Xiao (2009) proposed a parallel mean shift tracking algorithm on Graphics Processing Unit (GPU) using the Compute Unified Device Architecture (CUDA). Recently, Khan et al. (2011) proposed a tracking scheme that jointly employs particle filters and multi-mode anisotropic mean shift. The tracker estimates the dynamic shape and appearance of objects, and also performs online learning of the reference object.

The main advantage of the mean shift is that it has low complexity, is robust and invariant to object deformation. However, it has several disadvantages. First, the spatial information of the object is not strongly encoded in the representation, thus the scale and orientation information will be lost during tracking. Moreover, there are other constraints for mean shift tracking: it assumes that the object will not move more than its own size between the two consecutive frames, thus the search window size is limited to the size of the object.

2.3.4. Active contour tracking

The first examples of contour-based trackers started appearing in the late 1980s and early 1990s (Kass et al., 1988; Terzopoulos and Szeliski, 1992; Leymarie and Levine, 1993). Following the introduction of the standard snakes, several extensions were made. For instance, in (Terzopoulos and Szeliski, 1992), the authors continued the discussion of snakes by describing how: (1) snakes can be derived in a probabilistic framework; (2) Kalman filters can be used in conjunction with snakes. These additional contributions are closely intertwined, as the probabilistic derivation of snakes can be conveniently linked with the Bayesian perspective of Kalman filters. Introducing Kalman filters into the snake tracking paradigm can provide superior initial guesses for the solution in the current frame. As a result, larger inter-frame target displacements can be estimated and local minima pose less of a problem. An example of snake extensions was provided by the work of Peterfreund

(Peterfreund, 1997, 1999). In this work, the author includes velocity information in the snake energy. The general rationale is that the velocity of the snake should mimic the velocity of the contour in the image.

Level sets can provide a representation of a curve or interface and can be used to track their movement. In essence, level sets provide an alternative to representing a curve as a parameterized function. Avoiding contour parameterization by using level sets offers numerous advantages, including the ability to easily track curves with corners and cusps as well as allowing topological changes within the contour. Although level set is designed for single frame segmentation rather than tracking, the work of Caselles et al. (1995) is one of the first and most influential examples of applying level sets to the image processing domain. The basic level set contour trackers was presented by Paragios and Deriche (2000). The geodesic active contours were applied to the problem of detection and tracking. The tracking component of the system in (Paragios and Deriche, 2000) was simply an image gradient-based geodesic energy integral. This tracking energy was equivalent to that seen in (Caselles et al., 1995). Other interesting methods that attempt to incorporate region information into a contour tracker were presented in the paper of Jehan-Besson et al. (2001), Chen, Rui and Huang (2001), Mansouri (2002), Yilmaz et al. (2004), Shi and Karl (2005), Li et al. (2006), Zhou et al. (2007), Rathi et al. (2007), Fussenegger et al. (2006), and Thida et al. (2006).

Active contour energy function used to parameterize the lanes of travel based only on the accumulation of spatio-temporal image derivatives, and a tracking algorithm that exploits longer temporal constraints is made possible by the compact data representation (Jacobs et al., 2009; Georg and Pless, 2009 and Dixon et al., 2009). Chockalingam et al. (2009) presented an approach to visual tracking based on dividing a target into multiple regions, or fragments. The target is represented by a Gaussian mixture model in a joint feature-spatial space, with each ellipsoid corresponding to a different fragment.

Active contours have multiple advantages over classical feature attraction techniques.

- Active contours are autonomous and self-adapting in their search for a minimal energy state.
- They can be easily manipulated using external image forces.
- They can be made sensitive to image scale by incorporating Gaussian smoothing in the image energy function.
- They can be used to track dynamic objects in the temporal as well as spatial dimensions.

The key drawbacks of the traditional active contour are

- They can often get stuck in local minima states; this may be overcome by using simulated annealing techniques, but at the expense of longer computation times.
- They often overlook small features in the process of minimizing the energy over the entire path of their contours.
- Their accuracy is governed by the convergence criteria used in the energy minimization technique; higher accuracies require tighter convergence criteria and hence, longer computation times. (Wikipedia, 2012)

2.4. Classification

The purpose of object recognition is to identify a correspondence between a three-dimensional (3D) object and some part of a two-dimensional (2D) image taken from an arbitrary viewpoint in a cluttered real-world scene. The challenges involved in object

recognition and classification are mainly feature extraction, the efficient representation and then the comparison of two objects based on their representations (Islam et al., 2011). Object representation is based on visual cues such as edge elements, boundaries, corners, junctions, brightness or colour features obtained from object images (Belongie et al., 2002). Due to many challenges such as variations in viewpoint, illumination, occlusion etc this task is still under research.

2.4.1. Region-based classification

Region-based features are usually extracted from the image in the region of the object. In video sequences, this area is mainly the foreground silhouette extracted by the foreground segmentation algorithm. Image moments are often used to generate a feature vector for the silhouette. Without any feature generation, the convex hull of the silhouette (binary mask) can be used for comparison. A length based vehicle classification method was proposed by Avery et al. (2004) and Zhang et al. (2007). The length of the silhouette was exploited to distinguish long vehicles from short vehicles, and hence the need for complicated camera calibration can be eliminated. An approach for region matching is used in (Buch et al., 2008, 2010; Song and Nevatia, 2007). Morris and Trivedi (2006a, 2006b, 2008) use 17 different region features, including seven moments for eight different types and collect traffic flow statistics by leveraging tracking information. A comparison between image-based features, and image measurement features, is given. Both feature types are used with principal component analysis (PCA) and linear discriminant analysis (LDA) as the dimensionality reduction techniques. Image measurement features with LDA was used for the final algorithm, because it gave the best performance.

2.4.2. Feature-based classification

The concept of grids/cells of Histograms of Oriented Gradients (HOG) was introduced in (Dalal and Triggs, 2005). To calculate the feature vector, the gradient input image window is divided into a grid of cells. For every cell, a HOG in pixels is calculated. The histogram represents an 8 dimensional local feature vector. The vectors of all cells are concatenated to give one global feature vector for the image window. This concept was extended to vehicle detection in (Buch et al., 2009) by introducing 3D histograms of oriented gradients (3-DHOG), which uses 3D model surfaces rather than 2D grids of cells. This approach allows the algorithm to resolve scale and use a single model for variable viewpoints of road users. A boosting HOG was used for vehicle classification by Zhang and Li et al. (2008) and Cao et al. (2011). Rybski et al. (2010) focused on vision-based algorithms for determining vehicle orientation of vehicles in images. They train a set of HOG classifiers to recognize different orientations of vehicles detected in images. Zhang and Li et al. (2008) proposed a method for object classification by boosting different local feature descriptors in motion blobs. SIFT descriptor (Lazebnik et al., 2005), HOG descriptor, Spin Image descriptor and RIFT descriptor (Lowe, 2004) have been compared in the paper. Zhang and Yu et al. (2005) combined local texture features (PCA-SIFT), global features (shape context), and spatial features within a single multi-layer AdaBoost model for object class recognition. SIFT features and other local features for generic object recognition are combined in (Opelt et al., 2006), and (Zhang and Chen et al., 2006) uses a derivation of SIFT, i.e., the PCA–SIFT, for generic object recognition. The local features are used in combination with global edge features in an adaptive boost (AdaBoost) classifier. Modified SIFT descriptors are used in (Ma and Grimson, 2005) to generate a rich representation of vehicle images. Re-identified SIFT interest points is used between frames for tracking vehicles in urban scenes (Gao et al.,

2009). Zhang, Li and Yuan et al. (2007) described an appearance-based method to achieve real-time and robust object classification in diverse camera viewing angles. A new descriptor, i.e., the Multi-block Local Binary Pattern (MB-LBP), was proposed to capture the large-scale structures in object appearances. Based on MB-LBP features, an adaBoost algorithm was introduced to select a subset of discriminative features as well as construct the strong two-class classifier. The vehicles were classified into car, van, truck, person, bike and group of people by extracting distinct visual features.

2.4.3. Classifiers

Many classification algorithms have been presented, such as Boosting, K-nearest neighbour algorithm, K-means cluster, LogitBoost, Naïve Bayes classifier, Radial basis function network, and Fuzzy clustering for different tasks and objectives. This section mainly reviews SVM and random forests methods which are used in the thesis. They are popular method for vehicle classification.

2.4.3.1. Support vector machines

The support vector algorithm is a nonlinear generalization of the generalized portrait algorithm developed in Russia in the sixties (Vapnik and Lerner, 1963; Vapnik and Chervonenkis, 1964). However, a similar approach using linear instead of quadratic programming was taken at the same time in the US, mainly by Mangasarian (1965, 1968, 1969). The support vector machine (SVM) was largely developed at AT&T Bell Laboratories by Vapnik and co-workers. Ma and Grimson (2006) proposed an approach to vehicle classification under a mid-field surveillance framework. They discriminated features based on edge points and modified SIFT descriptors. Eigenvehicle and PCA-SVM were proposed and implemented to classify vehicle into trucks, passenger cars, van and pick-ups in paper (Zhang and Chen et al. 2006). Other SVM methods for vehicle classification can be found in the paper of Chen and Zhang (2007), Creusen and Wijnhoven (2009), Dalal and Triggs (2005), Serre at al. (2007) and Thi et at. (2008). The details of SVM will describe in Section 5.4.1.1.

The advantages of the SVM technique can be summarised as follows (Laura and Rouslan, 2008):

- By introducing the kernel, SVMs gain flexibility in the choice of the form of the threshold separating solvent from insolvent companies, which needs not be linear and even needs not have the same functional form for all data.
- SVMs deliver a unique solution, since the optimality problem is convex.
- Produce very accurate classifiers.
- Less overfitting, robust to noise.

The disadvantages are:

- Computationally expensive.
- Lack of transparency of results.

2.4.3.2. Random forests classifier

Random forests (RF) is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees. The algorithm for inducing a random forest was developed by Breiman (2001). The method combines Breiman's "bagging" idea and the random selection of features, introduced independently by

Ho (1995, 1998) and Amit and Geman(1997) in order to construct a collection of decision trees with controlled variation. Three classes of vehicles (sedans, van and tracks) were recognized by random forests, artificial neural network and k-nearest neighbours algorithm in the paper (Dalka and Czyzewski, 2010). For traffic accident data mining analysis, Krishnaveni and Hemalatha (2011) indicated that random forests outperformed the Naive Bayes Bayesian classifier (Rish, 2001), AdaBoostM1 Meta classifier (Freund and Schapire, 1996), PART Rule classifier (Lewicki and Hill, 2007) and J48 Decision Tree classifier (Quinlan, 1993). Leshem and Ritov (2007) combined the random forests algorithm with the Adaboost algorithm as a weak learner to predict traffic flow for traffic flow management. Harb et al. (2009) employed random forests to rank the importance of the drivers/vehicles/environments characteristics on crash avoidance manoeuvres. The details of RF will be described in Section 5.4.1.2.

The advantages of RF are (Caruana et al., 2008):

- It is one of the most accurate learning algorithms available.
- It runs efficiently on large databases
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

The disadvantage is (Segal, 2003):

- Random forests have been observed to overfit for some datasets with noisy classification/regression tasks.

2.5. Summary

One of the main objectives of visual surveillance is to analyse and interpret individual behaviours and intersections between objects of event detection. Even though significant progress has been made in computer vision and other areas, there are still major technical challenges to be overcome before the dream of reliable automated surveillance is realized. This chapter has presented a comprehensive review of computer vision and image processing techniques for traffic analysis systems, with a specific focus on urban environments. There is an increasing scope in intelligent transport systems to adopt video analysis for traffic measurement. The research expands from the highway environment to the more challenging urban domain. This condition opens many more application possibilities with traffic management and enforcement. Traditional methods use background estimation and perform top-down classification, which can raise issues under challenging urban conditions. Methods from the object recognition domain have shown promising results, overcoming some of the issues of traditional methods, but are limited in different ways. An obvious requirement for a surveillance system is real time performance. Moreover requirement for robustness and accuracy tend make the algorithm design complex and of high computational cost.

3. Active Segmentation and Adaptive Tracking Using Level Sets

3.1. Introduction

"The goal of image segmentation is to partition the image plane into meaningful areas, where meaningful typically refers to a separation of areas corresponding to different objects in the observed scene from the area corresponding to the background" (Cremers et al., 2007). Object segmentation is associated with boundary detection and integration, when a boundary is roughly defined as a curve or surface separating homogeneous regions. A mathematical definition of homogeneity is the fundamental component of any segmentation algorithm.

Image segmentation has been studied since the early days of computer vision and image processing, for example, the papers published by Comaniciu and Meer (2002), Jehan-Besson et al. (2003a), Yang and Foran (2005), Cremers et al. (2004 and 2007), Brox and Weickert (2004) and Chan et al. (2000) and their references. While earlier approaches were often based on heuristic processing steps, optimization methods have become established as being more principled and transparent. Segmentations of a given image were obtained by minimising appropriate cost functions.

Based on level sets and mean shift algorithm, an active segmentation (AS) and adaptive object tracking algorithm is presented in this chapter. Based on level set methods and a multi-phase colour model, a general variational formulation which combines Minkowski-form distance L_2 and L_3 of each channel and their homogenous regions in the index is defined. The AS method successfully finds whole object boundaries which include different known colours, even in very complex background situations, rather than split an object into several regions with different colours. An improved mean shift algorithm is developed to track a moving object. The new object tracking algorithm adaptively changes the colour space model (CSM) throughout the processing of a video by a similarity distance is developed. Once the contours of the object to be tracked are obtained by the AS method, rather than using the standard Epanechnikov kernel, a kernel weighted by the normalized Chamfer distance transform has been used to improve the accuracy of target representation and localization, minimising the distance between the two distributions of foreground and background using the Bhattacharya coefficient. Experiments conducted on various synthetic and real colour images illustrate the segmentation and tracking capability and versatility of the algorithm in comparison with results using the Chan-Vese vector method (Chan et al., 2000) which is a new approach for active contour to detect objects without using edges.

The outline of the remainder of the chapter is as follows. The next section gives literature review of level sets method. Section 3.3 introduces the AS model as an energy minimization function. The regularization of the Heaviside and delta function by a complementary error function (erfc), formulate the model in terms of a level set function and compute the associated Euler-Lagrange equation are defined in section 3.4. The implementation is given in section 3.5. In section 3.6, the adaptive object-tracking algorithm is described. Section 3.7 shows the validation of the method using synthetic and wide range real images. Finally, a summary can be found in Section 3.8

3.2. Level sets related research

Since their introduction as a means of front propagation (Osher et al., 1988) and their first application to edge-based segmentation in the early 90's, level set methods have become increasingly popular as a general framework for image segmentation. "Snakes" or active contours were proposed seminally by Kass et al. (1987) to represent flexible contours, and have received a great deal of attention from the image processing community. Xu et al. (1997 and 1998) developed Gradient Vector Flow (GVF), for active contours. GVF was computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from that image. This solved the problem associated with initialization and poor convergence to concave boundaries. The GVF snake often outperforms other gradient-based models, because of its insensitivity to initial positions and its larger capture region (Yang and Foran, 2005). Jehan-Besson et al. (2003a and 2003b) proposed the use of region-based active contours using geometrical and statistical features for image segmentation. These features were globally attached to the boundary or to the region, and they explicitly took into account their evolution in the derivation. The geodesic active contours proposed by Caselles et al. (1997), Goldenberg et al. (2001) and Xie et al. (2003) were based on active contours evolving in time according to intrinsic geometric measures of the image. This approach was based on the relation between active contours and the computation of geodesics or minimal distance curves. The minimal distance curve lies in a Riemannian space whose metric is defined by the image content. The model was given by a geometric flow - Partial Differential Equation (PDE), based on mean curvature motion, and not by an energy function. This model allowed automatic changes in topology when implemented using the level set numerical algorithm introduced by Osher et al (2003).

The aim of object extraction is to find closed contours that separate the image into foreground (objects) and background regions. All the classical snakes and active contour models rely on the edge-function, depending on the image gradient to stop the curve evolution. These models can detect only objects with edges defined by gradient, e.g. based on the region competition method proposed by Zhu et al. (1996) and the continuously adaptive mean shift tracking method proposed by Bradski (1998). In practice, the discrete gradients are bounded and then the stopping function is never zero on the edges, and the curve may pass through the boundary. If the image is very noisy, then the isotropic smoothing Gaussian has to be strong, which will smooth the edges too. In contrast, the Chan-Vese active contour model (CV model) without edges does not use the stopping edge-function to find the boundary (Chan et al., 2001; Vese et al., 2002). Instead, the stopping term is based on Mumford-Shah segmentation techniques. What happens when an object has several homogenous regions with different colours? This occurs in a multi-phase colour image, or in textured images. A number of generalizations have been developed to deal with this problem. Chan and Vese extended their method to a vector-valued image (Chan et al., 2000), which is designated as CVV method (Chan-Vese Vector method). Similarly, Chan and Sandberg proposed a mathematical framework for active contour object detection in multi-channel or textured images using logical operations (Chan and Vese, 2000, 2002; Sandberg and Chan, 2002; Chung and Vese, 2009). By using a multiphase level set representation, it is possible to solve the minimal partition problem for an arbitrary number of partitions.

3.3. Description of the active contour model

The basic idea in active contour models (or snakes) is to evolve a curve, subject to constraints from a given image, in order to detect objects in the image. "*Let* Ω *be a bounded open subset of* R^2, *with* $\partial\Omega$ *the boundary. Let* $\boldsymbol{I}$ *be a given image such that* $\boldsymbol{I} : \overline{\Omega} \rightarrow \boldsymbol{R}$. *Let* $C(s) : [0, 1] \rightarrow \boldsymbol{R}^2$ *be a piecewise parameterized* C^1 *curve*" (Chan et al., 2000). The assumptions are: 1) $\boldsymbol{I}$ is composed by a maximum of M regions Ω_i; 2) the interface between the regions Ω is regular. The segmentation problem in computer vision, as formulated by Mumford and Shah (Mumford and Shah, 1989), can be defined as follows: "*Given an observed image* $\boldsymbol{I}_0$, *find a decomposition* Ω_i *of* Ω, *such that the new segmented image* $\boldsymbol{I}$ *varies smoothly within each* Ω_i, *and discontinuously across the boundaries of* Ω_i ".

The method proposed in the chapter includes the minimization of an energy based function to perform segmentation. Describing image segmentation by a variational model increases the flexibility of the representation, allowing the future employment of additional features, such as shape knowledge, texture, motion vectors, etc. As implemented here, a-prior knowledge of the colours of the object is assumed to be isolated. Currently, this can be user-driven by "clicking" on the desired colours in a graphical display or by pre-segmentation. Given an N-channel image $\boldsymbol{I}(I_1,...I_N)$, and a set of different colours/intensities c=(c_1, c_2, …, c_M). Then, $c_i,(i=1,...M)$ are vectors of length N. The components of foreground and background colours of the k^{th} channel are $c_{fg}^k = (c_{k1}^f,...,c_{kR}^f)$ and $c_{bg}^k = (c_{k1}^b,...,c_{kS}^b)$, $R+S=M$. Figure 3.1 gives an illustration.

An energy formulation with the following form is chosen:

$$E(C) = \mu \cdot length(C) + \lambda_{fg} \iint_{\Omega_{fg}} \left(F_{fg}(I(x,y), c_{fg})\right) dxdy + \lambda_{bg} \iint_{\Omega_{bg}} \left(F_{bg}(I(x,y), c_{bg})\right) dxdy \tag{3.1}$$

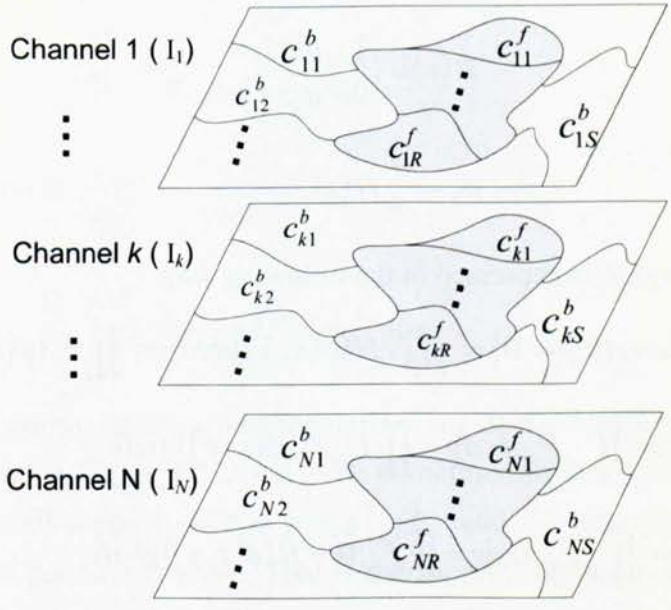


Figure 3.1. Image with *N* channels and a set of *M* different colours.

where C is the boundary curve of Ω_{fg} (shaded in Figure 3.1). $\Omega_{fg} = c_{k1}^{f} \cup \cdots \cup c_{kR}^{f}$ is the foreground (object) which is inside C, and the complement of $\Omega_{bg} = c_{k1}^{b} \cup \cdots \cup c_{kS}^{b}$ is the background which is outside C. Then, according to the bin-by-bin dissimilarity measurement – Minkowski-form distance (Rubner et al., 2000), the mean of L_2 (the standard deviation) and L_3 (the third root of the skewness) is used in each channel to get the expressions:

$$F_{fg}(I(x,y),c_{fg}) = \sum_{r=2}^{3}\left(\prod_{q=1}^{R}\left(\frac{1}{N}\sum_{p=1}^{N}\left(w_q^f \left|I_p(x,y) - c_{pq}^f\right|^r\right)\right)^{1/r}\right)^{1/R} \tag{3.2}$$

$$F_{bg}(I(x,y),c_{bg}) = \sum_{r=2}^{3}\left(\prod_{q=1}^{S}\left(\frac{1}{N}\sum_{p=1}^{N}\left(w_q^b \left|I_p(x,y) - c_{pq}^b\right|^r\right)\right)^{1/r}\right)^{1/S} \tag{3.3}$$

where $c_i = averag\{I_p(x,y)\}$ inside the i^{th} region. μ, λ_{fg}, λ_{bg}, w_q^f and w_q^b are nonnegative weights for the regularizing term and the fitting term, respectively. The model is robust to symmetric and asymmetric noise (e.g. Gaussian white noise and Gamma noise).

The optimal partition is obtained by minimizing the energy $E(C)$. *The key idea is to evolve the boundary C to the boundary of the object from some initialization in the direction of the negative energy gradient under the constraints from the image* (Cremers et al., 2007).

3.4. Level set formulation of the model

For the level set formulation of the variational active contour model, the unknown variable C is replaced by the unknown variable ϕ, and follow (Zhao et al., 1996), using the Heaviside function H, and the one-dimensional Dirac measure δ_0 defined respectively by

$$H(z) = \begin{cases} 1, & if\ z \geq 0 \\ 0, & if\ z < 0 \end{cases} \tag{3.4}$$

$$\delta_0 = \frac{d}{dz} H(z) \tag{3.5}$$

The terms in the energy E is expressed in the following way:

$$length(C) = length\{\phi = 0\} = \iint_{\Omega} \left|\nabla H(\phi(x,y))\right| dxdy = \iint_{\Omega} \delta_0(\phi(x,y)) \left|\nabla \phi(x,y)\right| dxdy$$

$$\iint_{\Omega_{fg}} F_{fg} dxdy = \iint_{\phi>0} F_{fg} dxdy = \iint_{\Omega} F_{fg} H(\phi(x,y)) dxdy$$

$$\iint_{\Omega_{bg}} F_{bg} dxdy = \iint_{\phi<0} F_{bg} dxdy = \iint_{\Omega} F_{bg} (1 - H(\phi(x,y))) dxdy$$

Thus, the energy functional can be written as

$$E(C)=\iint_{\Omega}\left(\mu\cdot\delta_0(\phi(x,y))|\nabla\phi(x,y)|+\lambda_{fg}\cdot F_{fg}H(\phi(x,y))+\lambda_{bg}\cdot F_{bg}(1-H(\phi(x,y)))\right)dxdy \quad (3.6)$$

In order to compute the associated Euler-Lagrange equation for the unknown function ϕ, numerical simulations use a slightly regularized version of H and δ_0, denoted here by H_ε and δ_ε, as $\varepsilon\to 0$. The first possible regularization of H, proposed in (Zhao et al., 1996), used the following $C^2(\overline{\Omega})$ (respectively C^1) function

$$H_{1,\varepsilon}(z)=\begin{cases}1, & z>\varepsilon\\ 0, & z<-\varepsilon\\ \frac{1}{2}\left(1+\frac{z}{\varepsilon}+\frac{1}{\pi}\sin\left(\frac{\pi z}{\varepsilon}\right)\right), & |z|\le\varepsilon\end{cases} \quad (3.7)$$

$$\delta_{1,\varepsilon}(z)=\begin{cases}0 & |z|>\varepsilon\\ \frac{1}{2\varepsilon}\left(1+\cos\left(\frac{\pi z}{\varepsilon}\right)\right), & |z|\le\varepsilon\end{cases} \quad (3.8)$$

In (Chan et al., 2000, 2001), the authors used the $C^\infty(\overline{\Omega})$ regularization of H is

$$H_{2,\varepsilon}(z)=\frac{1}{2}\left(1+\frac{2}{\pi}\arctan\left(\frac{\pi z}{\varepsilon}\right)\right) \quad (3.9)$$

$$\delta_{2,\varepsilon}(z)=\frac{\varepsilon}{\varepsilon^2+(\pi z)^2} \quad (3.10)$$

The regularization of Heaviside is approximated by the complementary error function (erfc) in this chapter.

$$H_{3,\varepsilon}(z)=\frac{1}{2}erfc\left(-\frac{\sqrt{\pi}z}{\varepsilon}\right) \quad (3.11)$$

The delta function is

$$\delta_{3,\varepsilon}=H'_{3,\varepsilon}=\frac{e^{-\left(\frac{\sqrt{\pi}z}{\varepsilon}\right)^2}}{\varepsilon} \quad (3.12)$$

These different approximations and regularizations of the functions H and δ_0 (taking $\delta_\varepsilon=H'_\varepsilon$) are presented in Figure 3.2. As $\varepsilon\to 0$, all approximations converge to H and δ_0. Whereas $\delta_{1,\varepsilon}$ has a small support interval $[-\varepsilon,\varepsilon]$. $H_{3,\varepsilon}$ and $\delta_{3,\varepsilon}$ approach to the H and δ_0, and $\delta_{3,\varepsilon}$ is different of zero everywhere. That is the support interval is $(-\infty,+\infty)$. Since the energy function may be non-convex (allowing many local minima), the solution may depend on the initial values, while with $H_{3,\varepsilon}$ and $\delta_{3,\varepsilon}$, the algorithm has the tendency to compute a

global minimum, independent of the initial curve. *Moreover, this allows automatic detection of interior contours* (Chan et al., 2001).

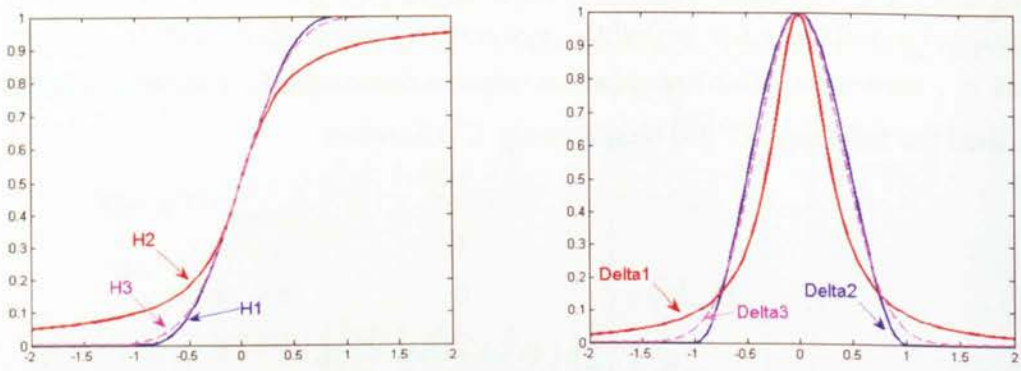


Figure 3.2. Three different regularizations of the Heaviside function H (left) and delta function δ_0 (right).

Minimizing *E(C)* with respect to ϕ yields the following Euler-Lagrange equation for ϕ, parameterizing the descent direction by time, t>0. The equation in $\phi(t,x,y)$ (with $\phi(0,x,y)=\phi_0(x,y)$ defining the initial contour) is:

$$\frac{\partial \phi}{\partial t}=\delta_{3,\varepsilon}(\phi)\left[\mu\nabla\bullet\left(\frac{\nabla\phi}{|\nabla\phi|}\right)-\lambda_{fg}F_{fg}+\lambda_{bg}F_{bg}\right] \tag{3.13}$$

in Ω, and with the boundary condition $\frac{\delta_{3,\varepsilon}(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}}=0$ on $\partial\Omega$, where $\vec{n}$ denotes the normal at the boundary of Ω. Actually, $\frac{\nabla\phi}{|\nabla\phi|}$ is the unit (outward) normal, and the divergence of the normal $\nabla\bullet\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ is the mean curvature of the ϕ. $\delta_{3,\varepsilon}$ was used in the formula (3.13) to improve the accuracy of the level sets algorithm.

3.5. Numerical implementation

To solve the evolution problem, the level set method proposed by Osher et al. (1988 and 2003) is used, and an implicit function ϕ using a signed distance function is defined in this chapter.

A distance function $d(\vec{x})$ is defined as $d(\vec{x})=\min\left(|\vec{x}-\vec{x}_I|\right)$ for all $\vec{x}_I\in\partial\Omega$, implying that $d(\vec{x})$ = 0 on the boundary where $\vec{x}\in\partial\Omega$. A signed distance function is an implicit function ϕ with $\phi(\vec{x})=d(\vec{x})$ for all $\vec{x}$. Thus, $\phi(\vec{x})=d(\vec{x})=0$ for all $\vec{x}\in\partial\Omega$, $\phi(\vec{x})=-d(\vec{x})$ for all $\vec{x}\in\Omega^-$, and $\phi(\vec{x})=d(\vec{x})$ for $\vec{x}\in\Omega^+$.

In the level set method (Osher and Sethian, 1988), $C \in \Omega$ is represented by the zero level set of a Lipschitz function $\phi : \Omega \to R$, such that

$$\begin{cases} C = \partial\omega = \{(x,y) \in \Omega : \phi(x,y) = 0\} \\ inside(C) = \omega = \{(x,y) \in \Omega : \phi(x,y) > 0\} \\ outside(C) = \Omega \setminus \overline{\omega} = \{(x,y) \in \Omega : \phi(x,y) < 0\} \end{cases}$$

Recall that $\omega \subset \Omega$ is open, and $C = \partial\omega$. Figure 3.3 illustrates the above assumptions and notations on the level set function ϕ, defining the evolving curve C. Osher et al. (1988 and 2003) give more details.

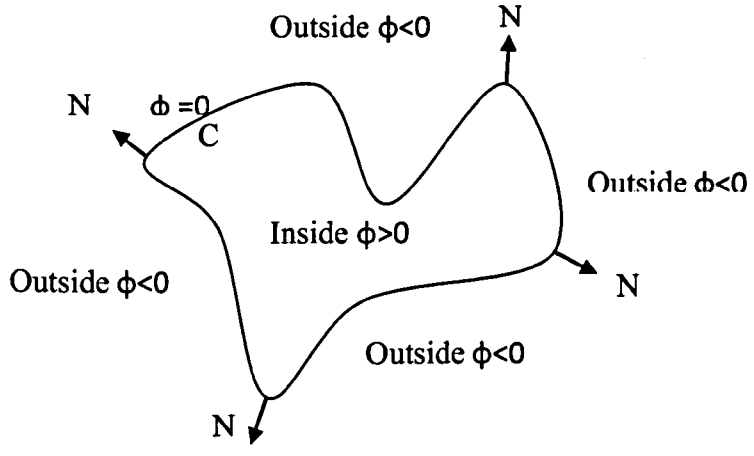


Figure 3.3. Curve C={(x,y): . ϕ (x,y)=0} propagating in normal direction.

For the level set formulation of the variation active contour model, the unknown variable C was replaced by the unknown variable ϕ. This function is positive on the interior, negative on the exterior, and zero on the boundary. Meanwhile, the extra condition of $|\nabla\phi| = 1$ should be satisfied. Note that ϕ does not have to be a signed distance function, for example a Euclidean distance transform or Chamfer distance transform could be chosen as a level set function ϕ. However, a signed distance function will increase the stability and quality of the evolution (especially if a vector field-based force and a force in the normal direction are combined). This is because the signed distance is the path of steepest descent for the function. In order to improve numerical efficiency, a discrete form of the Hamilton-Jacobi (HJ) equation with high order ENO (Essentially Nonoscillatory) and WENO (Weighted ENO) accuracy and a Local Lax-Friedrichs (LLF) scheme (Osher and Shu, 1991) is used. The upwind derivative by using a second order ENO scheme is calculated as well.

When working with level set and Dirac delta functions, ϕ will no longer be a distance function (i.e. $|\nabla\phi| \neq 1$). ϕ can become irregular after some period of time. A standard procedure is to reinitialize the signed distance function to its zero-level curve. This prevents the level set function from becoming too flat; it can be seen as a rescaling and regularization function.

The reinitialization procedure is made by the following evolution equation:

$$\begin{cases} \psi_t = sign(\phi(t))(1-|\nabla\psi|) \\ \psi(0,\bullet) = \phi(t,\bullet) \end{cases} \tag{3.14}$$

where $\phi(t,\bullet)$ is the solution ϕ at time t. *Then the new* $\phi(t,\bullet)$ *will be* ψ, *such that* ψ *is obtained at the steady state of (3.14). The solution of (3.14) will have the same zero-level set as* $\phi(t,\bullet)$ *and away from this set,* $|\nabla\psi|$ *will converge to 1* (Chan et al.,2001). Paper (Sussman et al., 1994) provides more details.

3.6. Adaptive object tracking

Real-time feature or object tracking is a critical task in many computer vision applications. Typically, tracking algorithms have two essential components (Comaniciu et al., 2003).

- Target representation and localization. This is usually a bottom-up process, which has to adapt to changes in the appearance of the target.
- Filtering and data association. This is mostly a top-down process, incorporating the dynamics of the tracked object, learning of scene priors, and evaluation of different hypotheses.

The way the two components are combined and weighted is application dependent and plays a decisive role in the robustness and efficiency of the tracking. In real time video applications, joint spatial and temporal analysis can be used to extract regions in the dynamic scene. This task is exceptionally difficult and has been extensively studied for several decades. Tracking has been based on simple Gaussian or adaptive Gaussian mixture models (McKenna et al., 1998), Kalman-filters (Chen and Rui et al., 2001) or dynamic programming (Geiger et al., 1995) to track fixed and deformable contours. Exemplars include face tracking in a crowded scene (DeCarlo et al., 2000; Hsu and Abdel-Mottaleb et al., 2002), aerial video surveillance (Kumar et al., 2001), and the detection and tracking of multiple people and vehicles in cluttered scenes using multiple cameras (Collins et al., 1999; Cohen and Medioni, 1999). In many real time applications only a small percentage of the system resources can be allocated for tracking, the rest being required for the pre-processing stages or for high-level tasks such as recognition, trajectory interpretation, and reasoning. Therefore, it is desirable to keep the computational complexity of a tracker as low as possible. An object tracking algorithm using mean shift and active contours was presented by Chang et al. (2005). Search location of the objects was simply computed by the zeros and first moments of the probability density function (pdf).

Detecting vehicles in an urban environment requires a robust and real-time algorithm that is capable of dealing with the impacts of camera vibration, shadow, sun and headlight glare, and illumination sudden change. Occlusions must also be properly dealt with to ensure accuracy. Occlusions occur when one vehicle appears next to another and blocks the line of sight either partially or completely. Occlusions are often regarded as results of poor camera placement. When a camera is placed too low or off-center, the line of sight is likely to be blocked by vehicles themselves, resulting in vehicle occlusions. Many video-based vehicle detection approaches intend to minimize occlusion problems by placing a camera into a position that enables a nearly birds-eye view of the region under detection. While this observation angle is ideal for minimizing occlusions, it is not always possible to achieve such a camera setting, especially when a pre-mounted road side CCTV camera is used for video

input. Occlusion reasoning becomes indispensible for vehicle detection under congested scenarios in urban traffic, when vehicle spacing becomes minimal and vehicle occlusions increase drastically. Once vehicles occlude with each other, it becomes very difficult to segment them (Wang et al., 2008). Many approaches turn to using the vehicle's appearance, thus relying on various models to determine an occluded vehicle's position. Segmentation of several composing vehicles from a single foreground moving object, can be done only through some prior knowledge of the objects' appearance and/or behaviour.

Based on AS and an improved mean shift tracking algorithm, an adaptive object tracking algorithm is presented in this section. This procedure adaptively changes the colour space model (CSM) throughout the processing of a video. The idea behind this adaptation is to allow continuous tracking when the object moves against environments of changing colour, and may interact visually with other moving objects. When the boundary of a tracked object was obtained, rather than use the standard, Epanechnikov kernel, a normalized Chamfer distance transform kernel that changes shape according to a level set definition was used, to correspond to changes in the perceived 2D contour as the object rotates or deforms to improve the accuracy of target representation and localization, minimising the distance between the two distributions of foreground and background using the Bhattacharya coefficient.

It can effective to deal with partially occlusion problem. The definition of similarity using Bhattacharyya coefficient which is used scale-invariant histogram metric. So the adaptive object tracking algorithm can track on a scale and shape changeable object.

Multi-object tracking, especially through occlusion, is a difficult research topic in urban traffic ITS. The main difficulty lies in data association of the measurements to the appropriate tracks. This is further compounded by the presence multiple splits and merges of tracks in multi-object tracking systems. Thus, it needs an efficient representation to integrate the spatial and temporal information. Such a representation should to establish patterns and relationships among detected moving regions reliably. In this thesis, the vehicle is detected only when it moves into the pre-defined detection region lane by lane. Only one vehicle is detected in each lane per frame. So it is not consider the case of multi-object tracking.

3.6.1. Defining similarity by histograms

So far research has focused on finding new distance functions that correspond better to the perceptual similarity of colour histograms. In tracking an object through a colour image sequence, the object can be represented by use of a discrete distribution of samples from a region in colour space, initially localised by a kernel whose centre defines the current position. Hence, the maximum in the distribution of a function, ρ, that measures the similarity between the weighted colour distributions as a function of position (shift) in the *candidate* image with respect to a previous *model* image, can be found. The Bhattacharyya coefficient is an approximate measurement of the amount of overlap of two sets of parameters for the respective densities $p(x)$ and $q(x)$. It was defined by Kailath (1967)

$$\rho = \int \sqrt{p(x)q(x)}dx \tag{3.15}$$

Since discretely sampled data from colour images is used, the discrete densities stored as m-bin histograms are used in both the *model* and *candidate* image. The discrete density of the model is defined as

$$\mathbf{q} = \{q_u\}, u = 1,2,\cdots,m \qquad \sum_{u=1}^{m} q_u = 1 \tag{3.16}$$

Similarly, the estimated histogram of a candidate at a given location **y** in a subsequent frame is

$$\mathbf{p}(\mathbf{y}) = \{p_u(\mathbf{y})\}, u = 1,2,\cdots,m \qquad \sum_{u=1}^{m} p_u = 1 \tag{3.17}$$

According to the definition of Equation (3.15), the sample estimate of the Bhattacharyya coefficient is given by

$$\rho(\mathbf{y}) = \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}] = \sum_{u=1}^{m} \sqrt{p_u(\mathbf{y}) q_u} \tag{3.18}$$

The Bhattacharyya distance between two distributions can be defined as

$$d(\mathbf{y}) = \sqrt{1 - \rho(y)} \tag{3.19}$$

Clearly the distance $d(y)$ lies between zero and unity, and obeys the triangle inequality. In a discrete space, $\{\mathbf{x}_i\}$, i=1,2,...n are the pixel locations of the model, centred at a spatial location **0**, which is defined as the position of the window in the preceding frame. A function b: $\mathbf{R}^2 \rightarrow \{1,2,\cdots,m\}$ associates to the pixel at location $\mathbf{x}_i$ the index b($\mathbf{x}_i$) of the histogram bin corresponding to the value of that pixel. Hence a normalized histogram of the region of interest can be formed (using **q** as an example)

$$\mathbf{q}_u = \frac{1}{n} \sum_{i=1}^{n} \delta[b(\mathbf{x}_i) - u], u = 1,2,\cdots m \tag{3.20}$$

where δ is the Kronecker delta function.

3.6.2. Mean shift tracking algorithm review

3.6.2.1. Sample mean shift

Given a set $\{x_i\}_{i=1,2,\cdots,n}$ of n points in the d-dimensional space R^d, the multivariate kernel density estimate with kernel $K(x)$ and window radius (band-width) h, computed in the point x is given by (Comaniciu et al., 1997, 2002 and 2003)

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{3.21}$$

The minimization of the average global error between the estimate and the true density yields the multivariate Epanechnikov kernel (E-kernel)

$$K_E(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)\left(1 - \|x\|^2\right) & \text{if } \|x\| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.22}$$

where c_d is the volume of the unit d-dimensional sphere.

The profile of the E-kernel is

$$k_E(x)=\begin{cases}\frac{1}{2}c_d^{-1}(d+2)(1-x) & if \quad x<1\\ 0 & otherwise\end{cases} \tag{3.23}$$

Employing the profile notation, the density estimate (3.21) can be written as

$$f(x)=\frac{1}{nh^d}\sum_{i=1}^{n}k\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \tag{3.24}$$

denote

$$g(x)=-k'(x) \tag{3.25}$$

Assuming that derivative of k exists for all $x\in[0,\infty)$, except for a finite set of points. A kernel G can be defined as

$$G(x)=Cg\left(\|x\|^2\right) \tag{3.26}$$

where C is a normalization constant.

3.6.2.2. Target model

Let $\{x_i^*\}_{i=1,2,\cdots,n}$ be the normalized pixel locations in the region defined as the target model. The region is centred at 0. An isotropic kernel, with a convex and monotonic decreasing kernel profile $k(x)$, assigns smaller weights to pixels farther from the centre. Using these weights increases the robustness of the density estimation since the peripheral pixels are the least reliable, being often affected by occlusions (clutter) or interference from the background.

The function b: $R^2 \rightarrow \{1, 2, ..., m\}$ associates to the pixel at location x_i^* the index $b(x_i^*)$ of its bin in the quantized feature space. The probability of the feature $u=1, 2, ..., m$ in the target model is then computed as

$$\hat{q}_u = C\sum_{i=1}^{n}k\left(\|x_i^*\|^2\right)\delta\left[b(x_i^*)-u\right] \tag{3.27}$$

where δ is the Kronecker delta function. The normalization constant C is derived by imposing the condition $\sum_{u=1}^{m}\hat{q}_u=1$, from where

$$C=\frac{1}{\sum_{i=1}^{n}k\left(\|x_i^*\|^2\right)} \tag{3.28}$$

Since the summation of delta functions for $u=1, 2, ..., m$ is equal to one.

3.6.2.3. Target candidates model

Let $\{x_i^c\}_{i=1,2,\cdots,n_h}$ be the normalized pixel locations of the target candidate, centred at y in the current frame. The normalization is inherited from the frame containing the target model.

Using the same kernel profile $k(x)$, but with bandwidth h, the probability of the feature $u=1, 2, ...,m$ in the target candidate is given by

$$\hat{p}_u = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i^c}{h}\right\|^2\right) \delta\left[b(x_i^c) - u\right] \tag{3.29}$$

where

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i^c}{h}\right\|^2\right)}$$

is the normalization constant. Note that C_h does not depend on y, since the pixel locations x_i^c are organized in a regular lattice and y is one of the lattice nodes. Therefore, C_h can be precalculated for a given kernel and different values of h. The bandwidth h defines the scale of the target candidate, i.e., the number of pixels considered in the localization process.

3.6.2.4. Distance minimization

Minimizing the distance (3.19) is equivalent to maximizing the Bhattacharyya coefficient $\rho(y)$. The search for the new target location in the current frame starts at the location y_0 of the target in the previous frame. Thus, the probabilities $\{\hat{p}_u(y_0)\}_{u=1,2,\cdots,m}$ of the target candidate at location y_0 in the current frame have to be computed first. Using Taylor expansion around the values $\hat{p}_u(y_0)$, the linear approximation of the Bhattacharyya coefficient (3.18) is obtained after some manipulations as

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(y_0)\hat{q}_u} + \frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(y)\sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \tag{3.30}$$

The approximation is satisfactory when the target candidate $\{\hat{p}_u(y)\}_{u=1,2,\cdots,m}$ does not change drastically from the initial $\{\hat{p}_u(y_0)\}_{u=1,2,\cdots,m}$, which is most often a valid assumption between consecutive frames. The condition $\hat{p}_u(y_0) > 0$ (or some small threshold) for all $u=1, 2, ..., m$, can always be enforced by not using the feature values in violation. Recalling (3.19) results in

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\mathbf{y}_0)\hat{q}_u} + \frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y - x_i^c}{h}\right\|^2\right) \tag{3.31}$$

where

$$w_i = \sum_{u=1}^{m}\delta\left[b(\mathbf{x}_i^c) - u\right]\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{y}_0)}} \tag{3.32}$$

Thus, to minimize the distance (3.19), the second term in (3.31) has to be maximized, the first term being independent of y. Observe that the second term represents the density estimate computed with kernel profile $k(x)$ at y in the current frame, with the data being weighted by w_i (3.32). The mode of this density in the local neighborhood is the sought maximum which

can be found employing the mean shift procedure. In this procedure the kernel is recursively moved from the current location y_0 to the new location y_1 according to the relation

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i^c w_i g\left(\left\|\frac{y_0 - x_i^c}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y_0 - x_i^c}{h}\right\|^2\right)} \tag{3.33}$$

where $g(x) = -k'(x)$, *it was defined in formula (3.25).*

3.6.3. Selection of the best colour space model

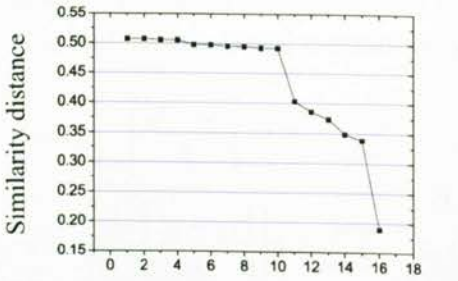
Segmentation or tracking success (or failure) depends primarily on how distinguishable an object is from its surroundings. If the object is very distinctive, it is easy to track. Otherwise it is hard to track. Normally, the features that best distinguish between foreground and background are the best features for tracking. The choice of feature space will need to be continuously re-evaluated over time to adapt to changing appearances of the tracked object and its background. How to define the criterion by which a tracker can discriminate an object from its surrounding background?

Numerous colour space models (*CSMs*) are used for segmentation and tracking of objects. Most *CSMs* represent colours in a three coordinates colour space such as RGB, Luv, HSV. Conversion formulae are used to transform from one space to another. There are many researchers who use *CSMs* for the selection, segmentation or tracking of objects by colour (Stern and Efros, 2002; Collins et al., 2005). The main objective is to achieve the best segmentation of an object of a certain colour from the background.

In order to select the best *CSM*, the different *CSMs* are sorted using the Bhattacharyya coefficient (Bhattacharyya, 1943) which is an approximate measurement of the amount of overlap between the two distributions of foreground and background. A "centre-surround" approach is used to sample pixels from object and background. A rectangular set of pixels covering the object is chosen to represent the object pixels, while a larger surrounding ring of pixels of the rectangle is chosen to represent the background. For an internal rectangle of size $h \times w$ pixels, the outer margin of width $(\sqrt{2}-1)\sqrt{hw}/2$ pixels forms the background sample. The foreground and background have the same number of pixels if $h=w$.

The distance criterion (3.19) is used to measure the similarity between the two histograms of the internal region and external region. The best colour space is selected by finding the *CSM* with the maximum distance value (i.e. the sum of the distance *d(y)* from each component of *CSM*) for each image.

Each potential colour feature set typically has dozens of tunable parameters and therefore the full number of potential features that could be used for tracking is enormous. 16 components from *CSMs* are constructed from different colour spaces (RGB, Lab, HSV, YIQ/NTSC/YCbCr, CMYK). All the values of pixels are normalized between 0 to 255, yielding feature histograms with 16 or 256 bins.



(a) (b)

(c)

Figure 3.4. (a) A sample image with concentric boxes delineating the object and background. (b). The similarity distance of each component from CSMs. (c) Rank-ordered 16 components from CSMs.

Figure 3.4(a) shows a sample image with concentric boxes delineating the object and background. The similarity distances between foreground and background of each colour component are shown in Figure 3.4(b) and the set of all 16 candidate images after rank-ordering the feature according to the criterion formula (3.19) are shown in Figure 3.4(c). The image with the most discriminative feature (best for tracking) is at the upper left. The image with the least discriminative feature (worst for tracking) is at the lower right.

3.6.4. Using a kernel based on the normalized Chamfer distance transform

In previous work, it has been shown that the equivalence of the mean shift procedure to gradient ascent on the similarity function holds for kernels that are radially symmetric, non-negative, non-increasing and piecewise continuous over the profile (Cheng, 1995). A radially symmetric kernel can be described by a 1D profile rather than a 2D (or higher order) image. The most popular choice for K is the optimal E-kernel that has a uniform derivative of $G=1$ which is also computationally simple. However, when tracking an object through a video sequence and applying the mean shift algorithm to move the position of the target window,

the bounds of the domain R^2 are altered on each successive application of the algorithm. There is no reason to suppose that the target has radial symmetry, and even if an elliptical kernel is used, i.e. there is a variable bandwidth, the background area that is being sampled for the colour distribution will change. If the background is uniform this will not affect the colour *pdf*, and hence the gradient ascent will be exact. However, if it is not uniform, but varies markedly and in a worst case has similar properties to the target, as it will be illustrated in the next section, then multiple modes will be formed in the *pdf* and the mean shift is no longer exact.

Therefore, a *distance transform* (DT) is used, matched to the shape of the tracked object, as a kernel function (see Figure 3.6(b)). This kernel can change shape through the sequence. For the DT each foreground pixel is given a value that is a measure of the distance to the nearest edge pixel. The edge and background pixels are set to zero. The normalised Chamfer distance transform (NCDT) is used rather than the true Euclidean distance, as it is an efficient approximation, as shown in Figure 3.5. The NCDT kernel represents the colour distribution of the tracked target better, yet retains the more reliable centre weighting of the radially symmetric kernels.

This transform is applied to the target area, separated from the background by AS methods described in Section 3.2-3.5. Figure 3.6(a) shows the AS algorithm result and the NCDT kernel of the segmented foreground sample image is shown in Figure 3.6(b). The aim is to show that this weight increases the accuracy and robustness of representation of the *pdf* as the target moves, since it excludes peripheral pixels that occur within a radially symmetric window applied to successive frames. In order to investigate the performance of the NCDT transform applied to define the regions of interest and add weights the colour densities in the video images, it is worth to investigate the anticipated gain in excluding background pixels from the density estimates, and increase the weights on those more reliable pixels towards the centre of the tracked object. This will outweigh the possibility of forming false modes because of the shape of the NCDT, bearing in mind that the radially symmetric kernels may produce false modes due to badly defined densities. Figure 3.6(c) shows the E-kernel which was used in the paper (Comaniciu et al., 1997, 2002 and 2003).

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	0	0
0	1	1	1	1	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0

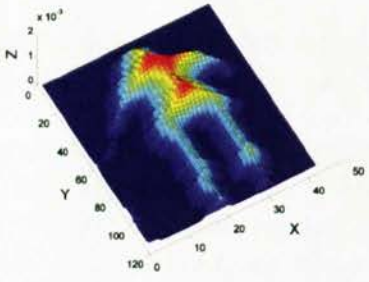
(a) Binary image

0	0	0	0	0	0	0	0
0	0.0189	0.0189	0.0189	0.0189	0.0189	0.0189	0
0	0.0189	0.0377	0.0377	0.0252	0.0189	0	0
0	0.0189	0.0377	0.0252	0.0189	0	0	0
0	0.0189	0.0377	0.0189	0	0	0	0
0	0.0189	0.0377	0.0252	0.0189	0.0189	0	0
0	0.0189	0.0377	0.0440	0.0377	0.0252	0.0189	0
0	0.0189	0.0377	0.0377	0.0377	0.0252	0.0189	0
0	0.0189	0.0189	0.0189	0.0189	0.0189	0	0
0	0	0	0	0	0	0	0

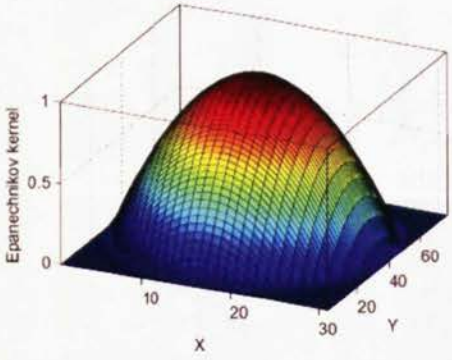
(b) NCDT

Figure. 3.5. Chamfer distances transform.

(a)



(b)



(c)

Figure 3.6. (a) AS segmentation result, (b)3D NCDT kernel with pseudo colour, (c) Epanechnikov kernel with pseudo colour.

3.6.5. NCDT kernel density estimation and gradient ascent

Kernel density estimation is widely applied to estimate the continuous *pdf* of a random variable from discrete data measurements (Epanechnikov, 1996; Wand and Jones, 1995; Scott, 1979; Park and Marron, 1990; Park and Turlach, 1992; Cao et al., 1994; Jones et al., 1996; Botev et al., 2010). In general, a kernel of appropriate bandwidth will produce a smooth, continuous distribution that nevertheless retains the constituent modes. In general, such kernels should be piecewise continuous, bounded, symmetric around zero, and monotonically decreasing from the centre. In the context, the kernel is applied to a set of samples that arise from pixels distributed in image space. That is the kernel is defined by the image rather than the colour space. Let $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ be an independent random sample drawn from $f(\mathbf{x})$, the colour densities function. If K is the normalized kernel function, then the kernel density estimate is given by

$$\mathbf{q}_u = \sum_{i=1}^{n} K(\mathbf{x}_i)\delta[b(\mathbf{x}_i) - u] \tag{3.34}$$

Estimating the colour density in this way, the mean shift algorithm can be used to iteratively shift the location $\mathbf{y}$ in the target frame, to find a mode in the distribution of the

Bhattacharya coefficient (Eq.3.18). Using Taylor expansion around the values, $p_u(\mathbf{y}_0)$, the Bhattacharyya coefficient is approximated by (Comaniciu and Meer, 2002):

$$\rho[\mathbf{p}(\mathbf{y}),\mathbf{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{p_u(\mathbf{y}_0)q_u} + \frac{1}{2}\sum_{i=1}^{n} w_i K(\mathbf{x}_i) \tag{3.35}$$

where

$$w_i = \sum_{u=1}^{m}\delta[b(\mathbf{x}_i)-u]\sqrt{\frac{q_u}{p_u(\mathbf{y}_0)}}$$

To maximize Equation (3.18), the second term in Equation (3.35) is maximized as the first term is independent of **y**. In the mean shift algorithm, the kernel is recursively moved from the current location $\mathbf{y}_0$ to a new location $\mathbf{y}_1$ according to the relation:

$$\mathbf{y}_1 = \sum_{i=1}^{n}\mathbf{x}_i w_i G(y_0 - x_i) \Big/ \sum_{i=1}^{n} w_i G(y_0 - x_i) \tag{3.36}$$

where G is the gradient function computed on K. This is equivalent to a steepest ascent over the gradient of the kernel-filtered similarity function based on the colour histograms.

For the mean-shift algorithm, the kernel G in Eq. (3.36) is based on the magnitude of the derivative of the NCDT kernel, K, that is

$$G(.) = \nabla K_{NCDT} = \sqrt{\left(\frac{\partial K_{NCDT}}{\partial x}\right)^2 + \left(\frac{\partial K_{NCDT}}{\partial y}\right)^2} \tag{3.37}$$

A local mean shift tracking is used in each component of the best *CSM*. The y with the minimum distance $d(y)$ is the optimal location of the tracked object.

3.6.6. Outline of the adaptive tracking algorithm

It is easy to apply AS and mean shift methods to track an object in a video. The main idea is to sequentially segment the frames of a video sequence by using the final partition from one frame as the initial partition of the next. Once the desired object is identified, the segmentation should occur in the region of interest surrounding the object to maintain global minima. This method will fail if the estimated position of the object in a frame is far from its true position. This can happen when the frame-to-frame motion of the object is too large relative to the size of the object. In this case, a motion vector is needed. The problem is solved by using the mean shift tracking algorithm (Comaniciu et al., 2003) to get the motion vector, translating the contours of the object to the new position, and then substituting length changes of the contour for the length in formula (3.1), minimising the energy function (3.38) instead of (3.1) to increase the convergence speed.

$$E(C^{t+1}) = \mu \cdot \left|length(C^{t+1}) - length(C^{t})\right| + \lambda_{fg}\iint_{\Omega_{fg}}\left(F_{fg}(I(x,y),c_{fg}^{t+1})\right)dxdy + \lambda_{bg}\iint_{\Omega_{bg}}\left(F_{bg}(I(x,y),c_{bg}^{t+1})\right)dxdy \tag{3.38}$$

where t and $t+1$ are the index numbers of two successive images.

The outline of the adaptive tracking algorithm is as follows:

Define internal and external rectangles covering the object centroid at y0 in the first image. Sort CSMs by similarity distance criterion (Eq.(3.19)).

Choose preferred CSM.
Get active contour (ϕ) of the tracked object by AS methods.
Repeat
Input the image i (initial value t = 1).
Obtain the set of foreground and background pixels by ϕ .
Sort and choose preferred CSM.
Get the set of constant colours by clustering using mean-shift or k-means clustering segmentation algorithms.
Compute NCDT kernel using Chamfer distance transform.
Form model histogram, q, in the preferred colour space.
Fetch the next frame t+1.
Compute candidate histogram $p(y_0)$ in the preferred CSMs using NCDT-kernel
Find the optimum location y_1 of candidate using mean shift tracking algorithm.
Get the motion vector.
Translate the contours.
Update ϕ by AS method.
t=t+1,
Until (end of input sequence)

3.7. Experimental results

The following experimental results were obtained using the active segmentation (AS) and adaptive object tracking algorithms proposed in the chapter. There are two purposes: Firstly, the AS method successfully finds whole object boundaries which include different known colours, even in very complex background situations, rather than split an object into several regions with different colours, for both synthetic data and real data. The AS method outperforms the CVV method. Secondly, the object tracking algorithm which combines new level set method and improved mean shift tracking algorithm improved the accuracy of target representation and localization.

3.7.1. AS method

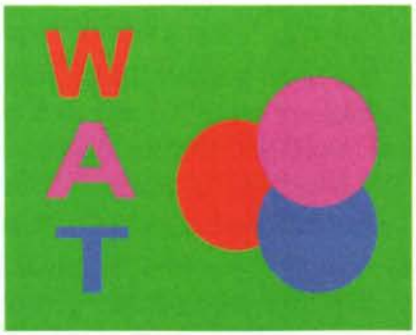
3.7.1.1. Synthetic image

In all numerical experiments, the parameters: $\lambda_{fg}=\lambda_{bg}=1$, $w_q^f = w_q^b = 1$ are chosen. The approximations $H_{3,\varepsilon}$ and $\delta_{3,\varepsilon}$ of the Heaviside and Dirac delta functions ($\varepsilon = \Delta x = \Delta y$) are used, in order to automatically detect interior contours, and to insure the computation of a global minimizer. Only the length parameter μ, which has a scaling role, is not the same in all experiments. If all or as many as possible objects have to be detected and of any size, then μ should be smaller. Otherwise, μ has to be larger.

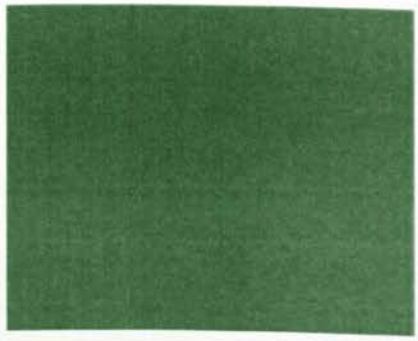
Figure 3.7(a) is the original input image. There are three objects with different colours (red, magenta, blue). The corresponding gray scale image is uniform because all the objects have the same intensity as the background (Figure 3.7(b)). The transformation is: Intensity=0.2989R+0.5870G+0.1140B, included in the Matlab toolbox (Vlad, 2001). Therefore, all segmentation algorithms associated to the gray/intensity image or binary image fail. But the AS algorithm successfully segments the objects as expected. In the experiment,

the target is the object with three colours. The initial value ϕ_0 is a circle that covers the whole image. The radius of the circle is 5 pixels. The grid size is 20 pixels. The centre of the circle is set to the centre of the grid. The inner and outer values are set to 0.5 and -0.5, respectively. Therefore, the location of the detected objects is not important. But for this experiment, the regular circular pattern over the image should cover the hole of the letter 'A'. Otherwise the letter 'A' will not be detected completely. $\mu = 0.0005*255^2$. Figure 3.7(c)-(e) shows the procedure for detecting the objects in a synthetic image; (c) shows the evolution of the contours; (d) shows the evolution of the associated piecewise-constant approximation of image $\boldsymbol{I}$; (e) is the evolution of associated 3D surface of ϕ. This experiment illustrates that the AS method has a large capture range (not sensitive to the initial value) and is able to move snakes to convex, concave or isolated boundaries. The detection true positive rate TPR = 91.80%.

The accuracy of the AS method can be compared with that of the CVV method by calculating the energy of every evolution. Though the energy formulation of AS is different to that of CVV, and the initial value is different, the energy can be compared after normalisation, because they should converge to the same global minima, that is, inf($E(C)$). For a perfect image and contours, $\inf(F_{fg}) = \inf(F_{bg}) = 0$, so $\inf(E(C)) = \mu \cdot length(C)$. The comparison is shown in Figure 3.8. AS/CVV (three colours) means that the object contours can be detected using three colour channels by the AS/CVV method. AS/CVV (one or two colours) has similar meaning. The experimental results show that the energies of the AS method only need a small number of iterations to reach the minimum value, for example, for the object with three colours, the initial energy of AS is bigger than that of CVV. After 25 iterations, AS obtained the minima, but CVV needed 150 iterations to obtain its final minimum for the experiment. In the other experiments, which try to find the objects with one or two colours, and other colours belong to the background, the results are equivalent.



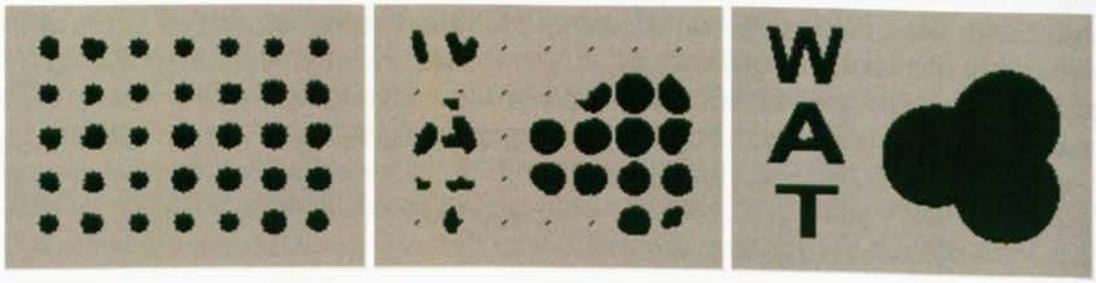
(a) Original input image



(b) Corresponding gray image

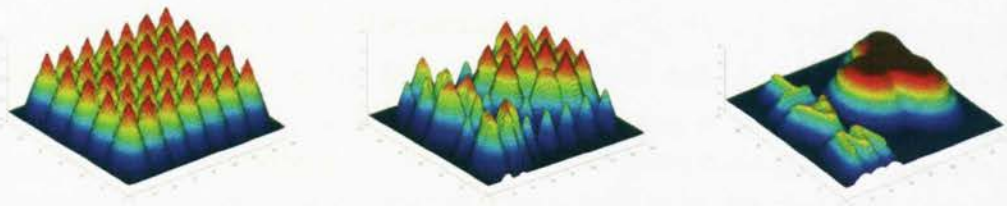
Iteration = 1 Iteration = 10 Iteration = 30

(c) Evolution of the contours



Iteration = 1 Iteration = 10 Iteration = 30

(d) The associated piecewise-constant approximation



Iteration = 1 Iteration = 10 Iteration = 30

(e) The associated 3D surface of ϕ

Figure 3.7. Detection of objects from a synthetic image.

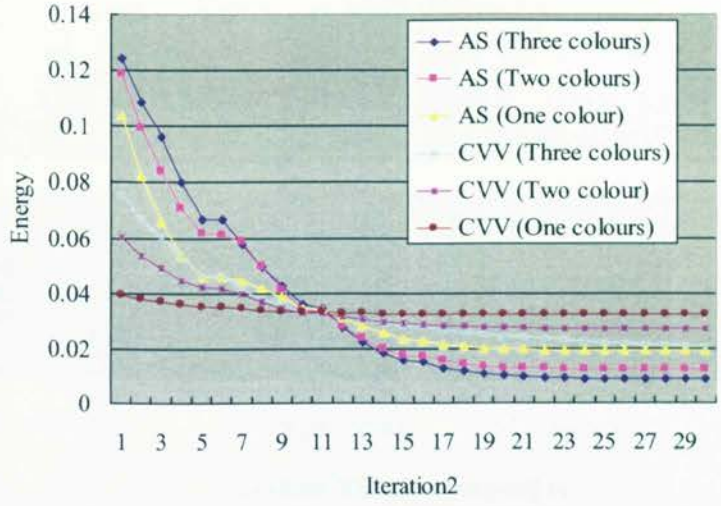
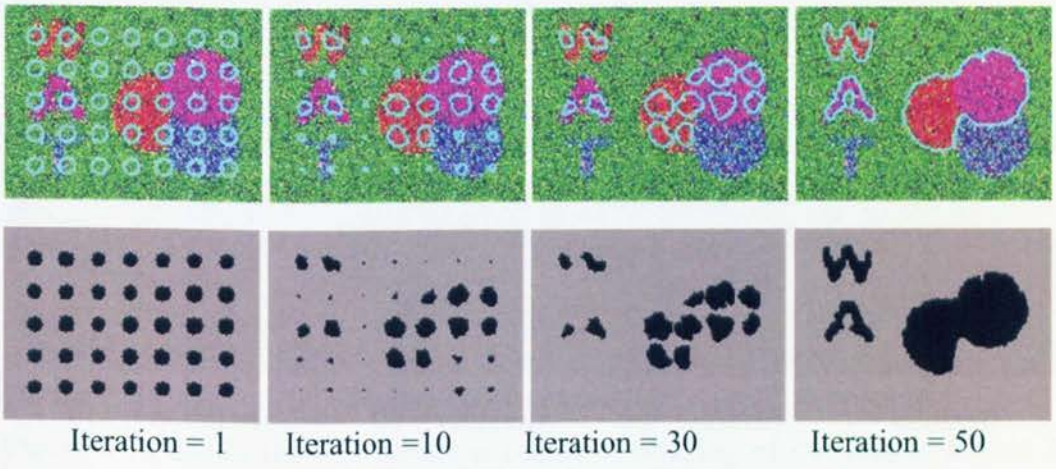
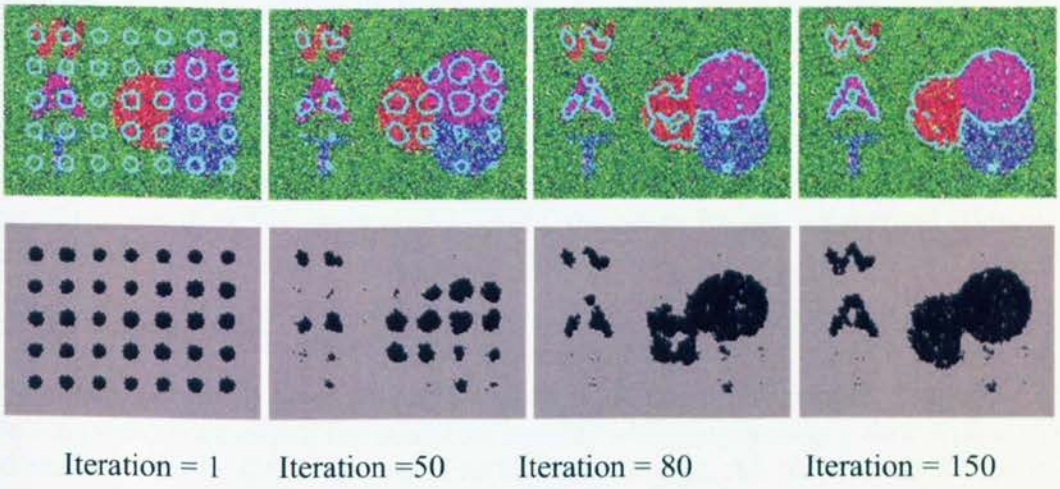


Figure 3.8. Comparisons of the energy evolution.

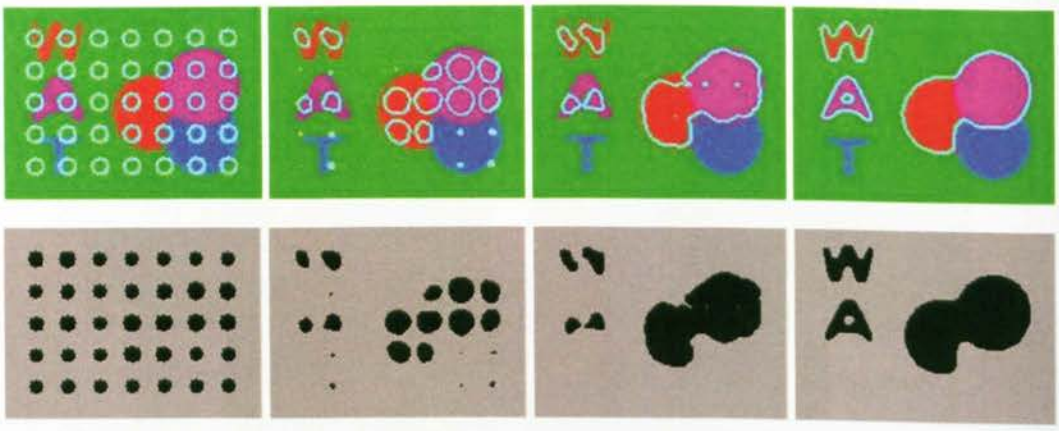


(a) Results of the AS method.



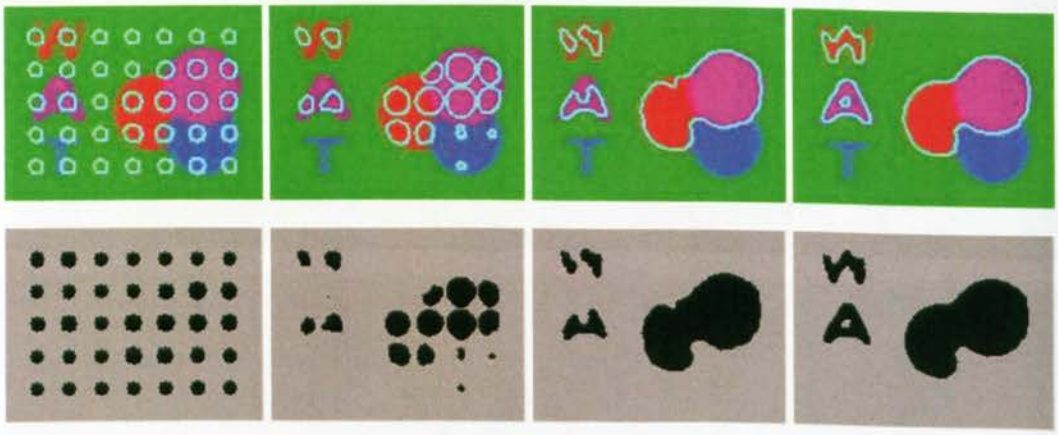
(b). Results of the CVV method.

Figure 3.9. Detection of objects from an image with Gaussian white noise (mean 0, variance 0.1, when the image values is in the range of [0, 1.0]) and salt and pepper noise (noise density = 0.1).



Iteration = 1 Iteration =10 Iteration = 30 Iteration = 50

(a) Results of the AS method



Iteration = 1 Iteration =50 Iteration = 80 Iteration = 150

(b). The results of CVV method.

Figure 3.10. Detection of objects from a blurred image.

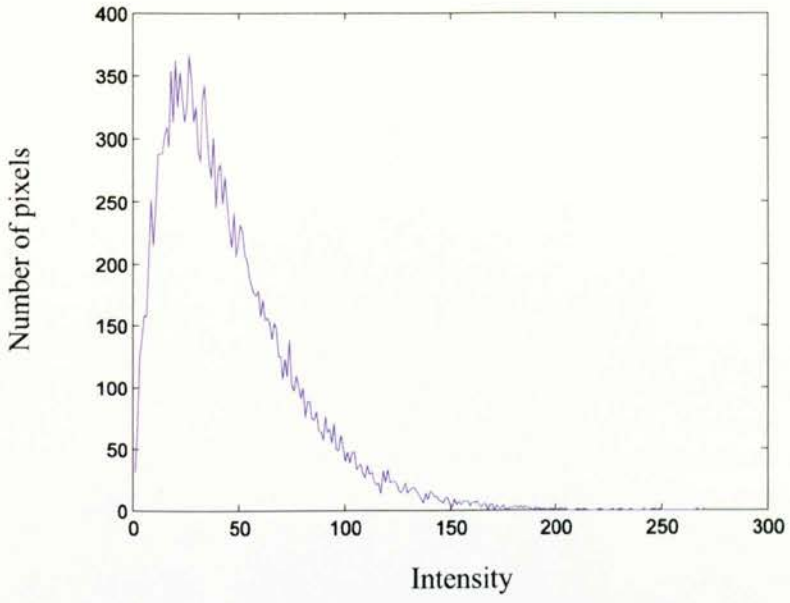


Figure 3.11. The distribution of noise.

An image with Gaussian white noise (mean 0, variance 0.1, when the image values is in the range of [0, 1.0]), and salt and pepper noise (noise density = 0.1) is used to test robustness of the algorithm. It is a symmetric noise. Intuitively, the image of Figure 3.9(a) is so noisy that it is hard even for a human to find the correct contours. The aim is to detect the contours of objects with two different colours, red and magenta. The blue one will merge with the background. Figure 3.9(a) shows the results obtained by the model. Figure 3.9(b) shows the results of the CVV method, by which the curve cannot be detected completely.

Figure 3.10 shows that the model is robust to blurring of the image. Figure 3.10(a) shows the results of AS, and (b) shows the results of CVV. The differences are evident, for example, the foreground should find the letter "W". The result of AS is clearer and more complete than that of CVV.

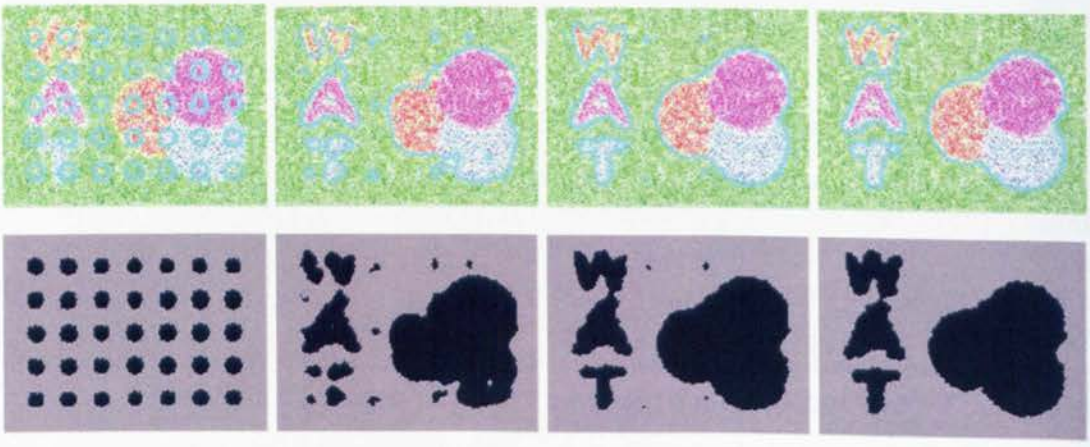
In order to test the effect of the L_3 Minkowski-form distance in the energy function, an asymmetric noise (i.e. a Gamma distribution) is added to the synthetic image.

Figure 3.11 shows the noise distribution. Figure 3.12(a) shows the results of the AS method, (b) shows the CVV method. As before, the result for the AS method is much better than that of CVV.

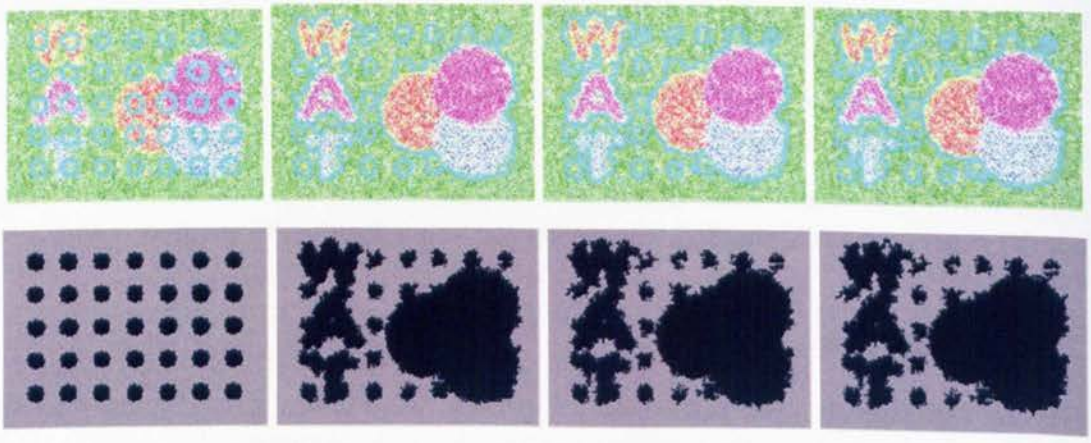
3.7.1.2. Real image data

Several experiments were performed using wide range real images. Figure 3.13 shows that the AS method can obtain the complete contour when segmenting a shoe with a coloured, striped texture, but the CVV can not. For the input image, AS only needs 87 iterations to converge to the optimal solution, but the CVV takes 202 iterations. Some examples with different types of contours or shapes are shown in Figure 3.14, Figure 3.15 and Figure 3.16. They illustrate the advantages of the model. In Figure 3.14, an object with different colours is detected. For these pictures, piecewise constant segmentation algorithms would not be able to

isolate the entire objects. For example, if the image is segmented by using mean shift segmentation algorithm, or by k-means clustering, at least three different regions for the detected objects will be segmented. But an entire object can be segmented by the AS method. The rectangle is the initial value in each sequence. The last one is the final (optimal) result. Figure 3.15 shows that the AS method successfully segments the objects from a complex background. The three images in the Figure 3.15(a) show that the yellow tree within the garden can be segmented successfully, even though it has no distinct contours (or edges). The three images in the Figure 3.15(b) show that the UK flag can be isolated as a single object, although it is again composed of three colours (red, white and blue). Figure 3.15(c) and (d) show that the AS algorithm can split objects with specific colour from background with multicolour. Figure 3.16 shows that AS method successfully segments the object in an infrared image. Figure 3.17 shows the vehicle segmentation result. (a) Shows the segmentation result obtained using background subtraction algorithm (details will describe in the Chapter 4). Two vehicles merge together. (b) Gives the segmentation result obtained from level set method. Two vehicles separate successfully. (c) – (f) illustrate the level set evolution procedure.

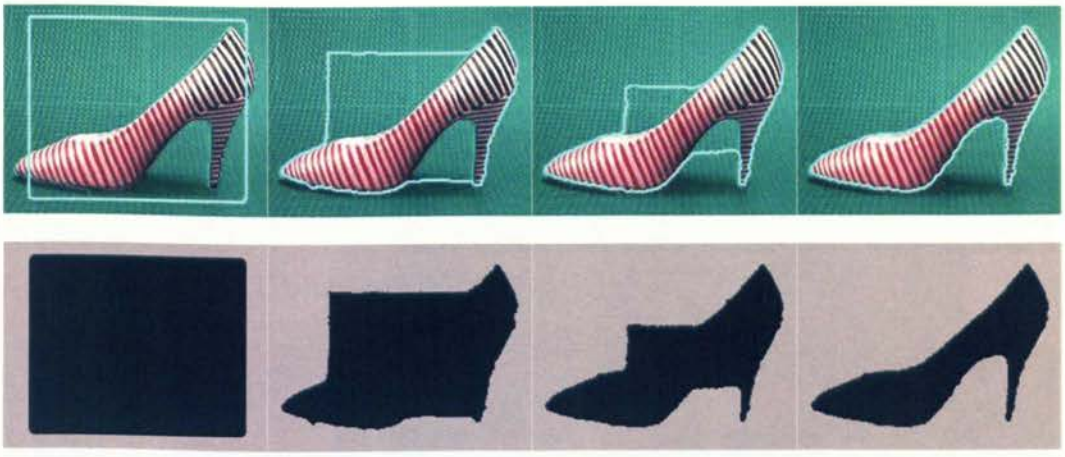


(a) Results of iterations of 1, 25, 40 and 60 of the AS method.

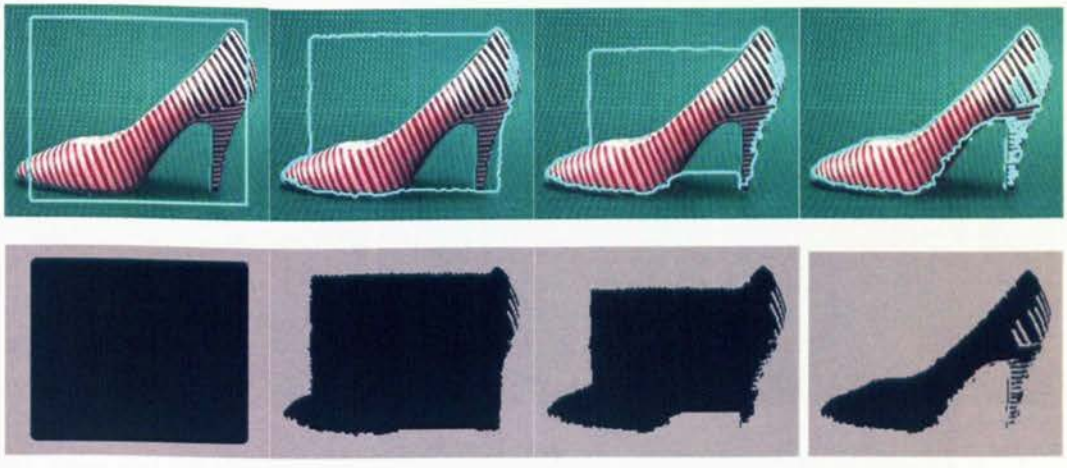


(b) Results of iterations of 1, 25, 40 and 60 of the CVV method.

Figure 3.12. Detection of objects from an image with a Gamma distribution noise.



(a) The results of AS (the iterations of 1, 30, 60 and 87)



(b) The results of CVV (the iterations of 1, 30, 60 and 202)

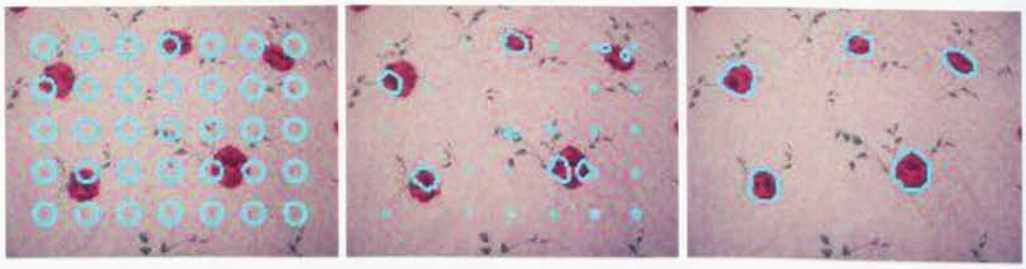
Figure 3.13. The comparison results of AS and CVV using real image.

(a) (b)

Figure 3.14. Detecting objects with different colours using AS method.

(a)

(b)

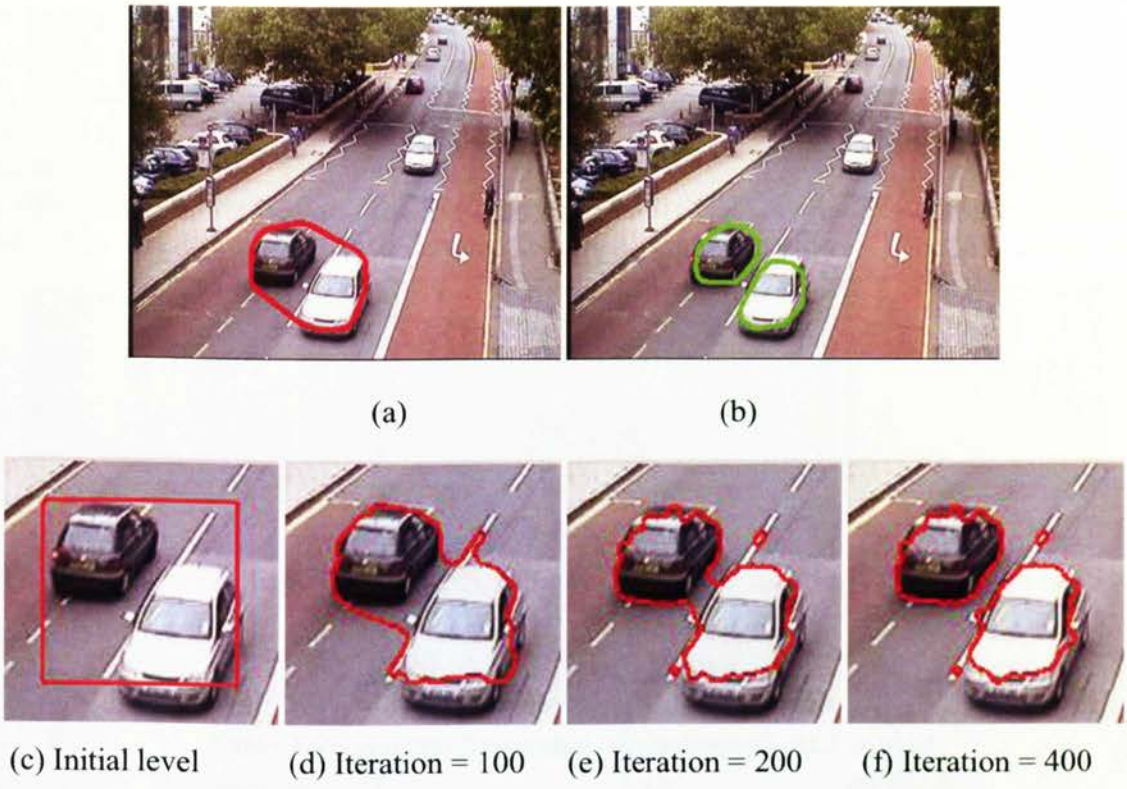


(c)

(d)

Figure 3.15. Detecting objects against a complex background using AS method.

Figure 3.16. Detecting objects in infrared image using AS method.



(a) (b)

(c) Initial level (d) Iteration = 100 (e) Iteration = 200 (f) Iteration = 400

Figure 3.17. Vehicle segmentation result. (a) result from background subtraction; (b) result from level set; (c) – (f) level set evolution procedure.

3.7.2. Tracking object with dynamic shape using NCDT kernel

This section presents an evaluation of the modified mean shift object tracking using the NCDT-kernel in comparison with the radially symmetric E-kernel. Moving objects, a static object with a moving camera and a combination of the two were tracked. Examples with variation of scale and the addition of Gaussian noise are given. All the tests were carried out on a Pentium 4 CPU 3.40 GHz with 1GB RAM. The code was implemented in Matlab, so that it would be reasonable to assume a considerable increase in processing speed if re-implemented in another language. Even so, real-time operation is possible.

In the first experiment, tracking of a moving male pedestrian in a video sequence of a shopping centre (CAVIAR dataset) that includes 75 frames of 320×240 pixels is presented, comparing the normal E-kernel with the NCDT kernel. The target location was initialized by a rectangular region (shown) of size 77×31 pixels. Figure 3.18 shows the first frame and the foreground image of the tracked object. In this case a simple regional homogeneity criterion has been applied as the target had relatively uniform intensity. Figure 3.19 shows the minimum value of a distance function, $d = 2(1 - \rho(\mathbf{y}))$, computed for each frame from the Bhattacharyya coefficient (Eq. 3.18). By definition, the distance of the first frame is 0, meaning a perfect match. The peak in the E-kernel data is 0.643 which corresponds to the wrong candidate frame 53. After this frame the distance reduces but the algorithm was tracking another object which has nearly the same colour as the target. Figure 3.20(a) and (b) show some examples, frames 1, 15, 30 and 60, from the whole sequence. In frame 30 some of

the original pedestrian is still contained within the window, but after the 52nd frame, the pedestrian is lost completely in Figure 3.20(b), as the tracker finally latches on to another crossing pedestrian. This experiment demonstrates that the inclusion of the background of the tracked pedestrian (in this case another pedestrian) includes pixels that are similar in colour space, so that the algorithm fails to identify the correct distribution in succeeding frames and hence follows the wrong target. Figure 3.21 shows the manifestation of the problem in the ρ-space, using Eq. 3.18 – effectively the E-kernel filtered density estimate has additional, confusing modes caused by the inclusion of the crossing pedestrian in the background.

Figure 3.18. Rectangular window and segmentation result

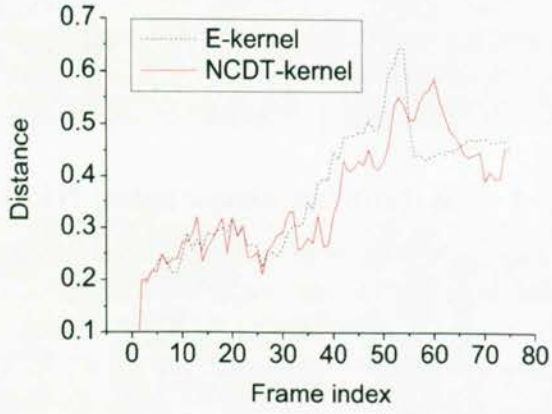
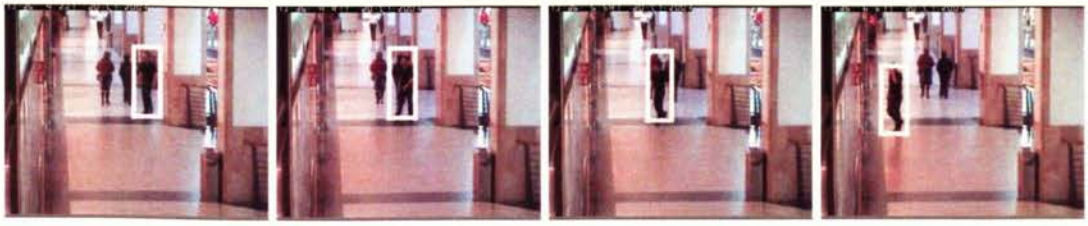


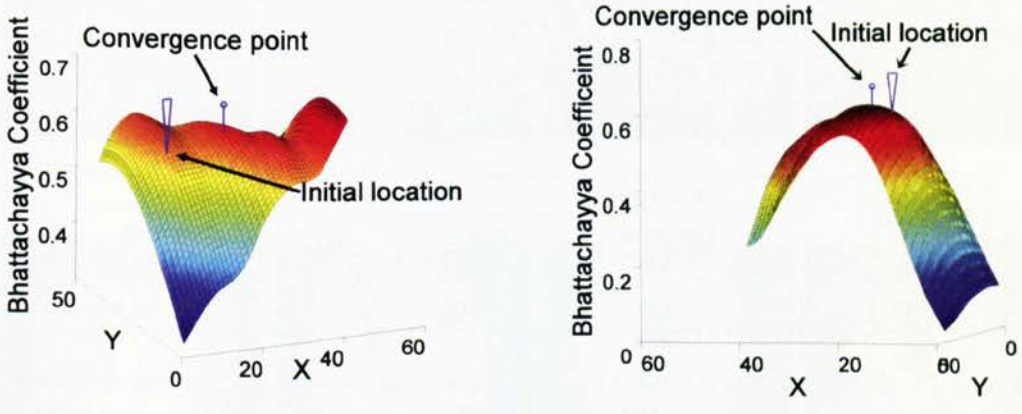
Figure 3.19. The Bhattacharya distance values, for the male pedestrian



(a) NCDT-kernel

(b) E-kernel

Figure 3.20. Tracking the crossing pedestrian.



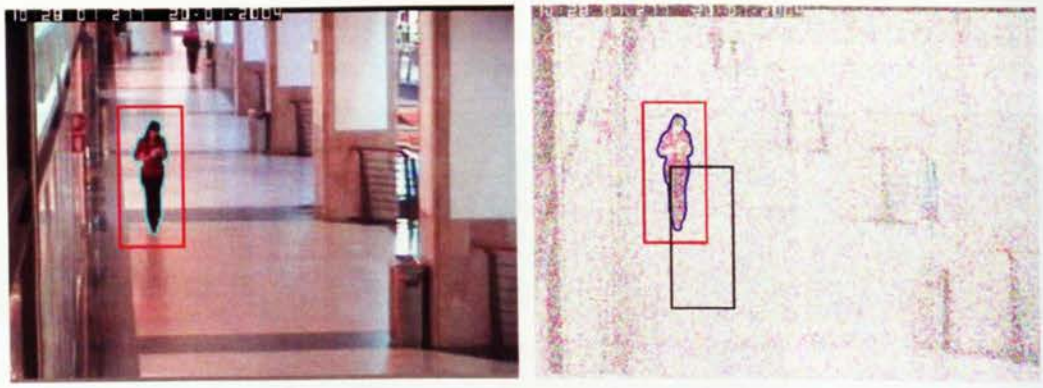
(a) (b)

Figure 3.21. The similarity surfaces (values of the Bhattacharyya coefficient) for frame 52. The initial points, (▽), and convergence points, (lines), are shown. (a) The result from the E-kernel. (b) The result from the NCDT-kernel.

In terms of complexity, computed from 200 executions of the program, the average frames per second of the NCDT-kernel and the E-kernel are 36.95 and 37.06 respectively. The maximum numbers of iterations within a single frame are 16 and 21, respectively. The average times per frame are roughly comparable because although the speed of convergence is quicker with the NCDT-kernel, additional processing is required to segment the target window, in order to get more robust and accurate tracking.

To further test the robustness of the NCDT-kernel algorithm and convergence properties, a combined random Gaussian and uniformly distributed noise of mean zero with 0.5 and 0.05 variance were added to the frame respectively. It was shown in Figure 3.22(a), the intensities ranging from 0 to 1. Figure 3.22(b) shows the result from 1000 trials superimposed on the noisy image. For this level of noise, the algorithm is successful on all occasions, beyond this level the success rate diminishes but this data is not presented. The black rectangle is the initial position, the red one is the optimal solution. The initial position is far from the target ($\Delta x = 20, \Delta y = 50$ pixels). From Table 3.1, which shows quantitative

results, the NCDT kernel algorithm needs on average only 4.4 iterations to converge to the optimal result, but the E-kernel needs 21 iterations on average. Again, the greater complexity of computing the NCDT kernel is balanced by the greatly reduced number of iterations, so the processing speed per frame is comparable.



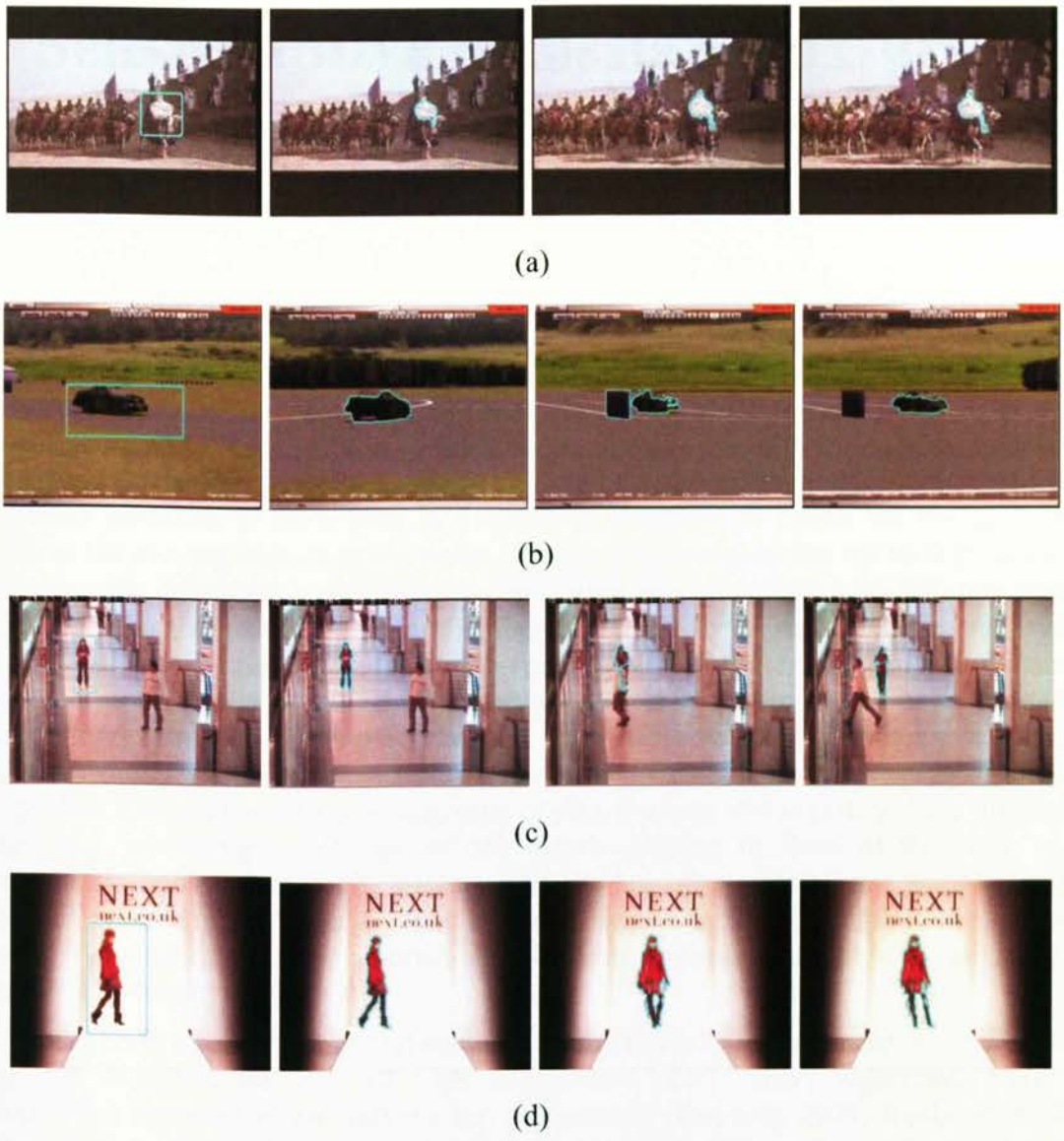
(a) (b)

Figure 3.22. (a) Original image and segmentation result. (b) Noised image, the initial rectangle (black rectangle) and the optimal solution (red rectangle).

Table 3.1. Comparison results of NDCT and E-kernel method

Method	Average iterations	CPU time (sec./frame)		
		max	min	mean
E-kernel	21	0.3205	0.2604	0.2801
NCDT kernel	4.4	0.2504	0.0300	0.1815

Figure 3.23 illustrates that the tracking algorithm can cope with dynamic deformation of object shape, location and scale in different sequences, even when the camera pans so that both the foreground and background move in the camera coordinate system (Figure 3.23(a) and (b)). All of these illustrations are from much longer sequences, typically more than one hundred frames. Figure 3.23(c) shows that the tracking algorithm is very robust to clutter and crossing objects. Two people cross in the pictures in the Figure 3.23(c), yet the algorithm adapts the contour to track the non-occluded portion of the woman, then re-grows the contour as she re-emerges from behind the man (the women partially occluded). The results of the fashion-show video (Figure 3.23(d)) illustrated that the algorithm is able to capture much more information about the person being tracked, such as gait and posture. In each of the sequences, the rectangle in the first image defines the initial region, in which the object to be tracked is contained. The second image in each figure shows the optimal contour (final segmentation). The third and fourth images in each figure show the tracking results.



(a)

(b)

(c)

(d)

Figure 3.23. Tracking video objects and dynamics of deformation.

3.8. Summary

A generalized active contour model for multi-channel and multi-phase colour image segmentation and an adaptive object-tracking algorithm have been proposed. By using level set methods, mean shift tracking algorithm, a Chamfer distance transform (NCDT kernel) and sorted *CSM*s, boundaries of detected and tracked objects are not necessarily defined by gradient or by very smooth boundaries, for which the classical active contour models are not applicable. The position of the initial curve can be anywhere in the image, and it does not necessarily surround the object to be detected. However, if the initial estimate is far from the true contour, it will take a long time to converge to the optimal solution. Several experiments have demonstrated the ability of the model to detect and track an object in a video. Comparing the new method with the CVV method, the comparison results also show that the method is more accurate and robust in the presence of symmetric and asymmetric noise.

4. Self-Adaptive Gaussian Mixture Model for Urban Traffic Monitoring System

4.1. Introduction

Identifying moving objects in a video sequence is a fundamental and critical task in video surveillance, traffic monitoring and analysis, human detection and tracking, as well as other visual tracking tasks, such as gesture recognition in the human-machine interface. A static camera observing a scene is a common case of a surveillance and monitoring system. Background modeling is often used in different applications to model the background and then detect the moving objects in the scene. The general theory is that the background model is built from the data and objects are detected if they appear significantly different from the background. The foreground pixels are further processed for object detection and tracking. The principal challenges are how to correctly and efficiently model and update the background model and how to deal with shadows. A robust system should be independent of the scene, and robust to lighting effects and changeable weather conditions. It should be capable of dealing with movement through cluttered areas, objects overlapping in the visual field, gradual illumination changes (e.g. time of day, evening and night), sudden illumination changes (e.g. switching a light on or off, clouds moving in front of the sun), camera automatic gain control (e.g. white balance and iris are often applied to optimally map the amount of reflected light to the digitizer dynamic range), moving background (e. g. camera vibration, swaying trees, snow or rain), slow-moving objects, cast shadows and geometric deformation of foreground objects.

At the heart of any background subtraction algorithm is the construction of a statistical model that describes the state of each background pixel. Many algorithms have been developed and the most recent surveys can be found in (Piccardi, 2004; Radke et al., 2005; Bouwmans et al., 2008; Elhabian et al., 2008). The simplest form of the background model is a time-averaged background image of the scene without any intruding objects. However, this method suffers from many problems requiring a large memory and a training period where foreground objects are absent, to ensure that static foreground objects are not considered as a part of the background. This will limit their utility in real time applications. In an urban traffic environment, the 'pure' background isn't available and can always be changed under critical situations like objects being introduced or removed from the scene, slow-moving or stationary objects. To account for these problems of robustness and adaptation, many background modeling methods have been developed. A single Gaussian model was used in real time tracking of the human body (Wren et al., 1997). However, individual pixel values can often have a complex distribution and more elaborate models are needed. A Gaussian mixture model (GMM) was proposed for real-time tracking by Stauffer and Grimson (1999, 2000). The algorithm relies on assumptions that the background is visible more frequently than the foreground and that the model has a relatively narrow variance. This approach has been found to cope reliably with slow lighting changes, repetitive motions from clutter, and long-term scene changes. Many adaptive GMM models have been proposed to improve the original background subtraction method. Power and Schoonees (2002) used a hysteresis

threshold to extend the GMM model. They introduced a faster and more logical application of the fundamental approximation than that used in (Stauffer and Grimson, 2000). The standard GMM update equations were extended to improve the speed and adaptation rate of the model (KaewTraKulPong et al., 2001; Lee, 2005; Cheung et al., 2004; Haque et al., 2008). Greggio et al. (2010) proposed self-adaptive Gaussian mixture models for real-time background subtraction recently. All these GMMs use a fixed number of components. Zivkovic and Heijden (2004, 2006) presented an improved GMM model using a recursive computation to constantly update the parameters of a GMM, which adaptively chose the appropriate number of Gaussians to model each pixel on-line, from a Bayesian perspective. The Zivkovic-Heijden Gaussian mixture model will be denoted as ZHGMM later in the text.

However, GMMs have drawbacks. Firstly, they are computationally intensive and the parameters require careful tuning. Second, they are sensitive to sudden changes in global illumination. If a scene remains stationary for a long period of time, the variance of the background components may become very small. A sudden change in global illumination can then turn the entire frame into foreground. For a low learning rate, it produces a very wide and inaccurate model that will have low detection sensitivity. On the other hand, for a high learning rate, the model updates too quickly, and slow moving objects will be absorbed into the background model, resulting in a high false negative rate.

Rapid illumination changes constitute one of the main difficulties when applying background subtraction in a real life setting. This work is motivated by the need for robust vehicle detection and classification algorithm that can be used in a traffic monitoring system to deal with sudden illumination changes. Martel-Brisson et al. (2007) proposed a novel statistical model based on a GMM. This model can deal with scenes with complex and time-varying illumination, including regions that are highly colour saturated, whilst suppressing false detection in regions where shadows cannot be detected. Vosters et al. (2010) combined the eigenbackground algorithm with a statistical illumination model and proposed a background subtraction method to cope with sudden illumination changes. Self-organisation through artificial neural networks can be used to handle scenes containing moving backgrounds, gradual illumination variations and camouflage. It can model shadows cast by moving objects (Maddalena et al., 2008; Singh et al., 2010). Reddy et al. (2010) proposed an adaptive patch-based background modeling for improved foreground object segmentation and tracking. Liao et al. (2010) developed a scale invariant local ternary pattern operator and pattern kernel density estimation technique to tackle illumination variations and dynamic backgrounds. Pilet et al. (2008) presented a fast background algorithm relying on a statistical model, not of the pixel intensities, but of the illumination effects. Zhao et al. (2009) extended the background GMM to spatial relations, the joint colours of each pixel-pair are modeled by a GMM to suppress the effects of illumination changes. Izadi et al. (2008) and Javed et al. (2002) employed colour and gradient information to build up the background GMM to reduce the foreground detection false alarm rate at pixel level.

Unfortunately, none of the existing background models can achieve robust performance to sudden changes in global illumination. Moreover, the parameters of a GMM may vary for different scenarios. Existing work either openly admits to setting blending and thresholding parameters by hand, or more commonly, does not mention how they are set.

Non-parametric density estimations also lead to flexible models (Elgammal et al., 2000; Han et al., 2008; Zhong et al., 2009). These approaches modeled the pixel as a random variable in feature space with an associated probability density function. Unlike GMMs, non-parametric models do not require the selection of a number of Gaussians to be fitted, but the

model adaptation is trivial. The major drawback is its computation cost.

Another main challenge in the application of background subtraction is identifying shadows that objects cast which also move along with them in the scene. Shadows cause serious problems while segmenting and extracting moving objects due to the misclassification of shadow points as foreground. Prati et al. (2003) presented a comprehensive survey of moving shadow detection approaches. Cucchiara et al. (2003) proposed the detection of moving objects, ghosts and shadows in HSV colour space and gave a comparison of different background subtraction methods.

Most background modeling techniques use a single leaning rate of adaptation which is inadequate for complex scenes as the background model cannot deal with sudden illumination changes. This chapter proposes a self-adaptive Gaussian mixture model to address these problems. An online dynamical learning rate is proposed for global illumination of background model adaptation to deal with fast changing scene illumination. The algorithm combines a pixel-level and frame-level process. Systematic statistical analysis shows how to select some significant parameters to obtain an optimal background model for a given scene. There are four main contributions: 1) An online dynamical learning-rate adaptation; 2) Global illumination of background model changing adaptation to deal with background illumination fast changes; 3) Spatio-temporal smoothing transform to deal with noises and camera vibration; 4) Use of statistical measurements to select some significant parameters to obtain optimal results.

Results of experiments using manually-annotated urban traffic video with sudden illumination changes illustrates that the algorithm achieves consistently better performance in terms of ROC curve, foreground detection accuracy, Matthews correction coefficient and Jaccard coefficient compared with other approaches based on the widely-used Gaussian mixture model.

The remainder of the chapter is organised as follows. In the next section the ZHGMM approach is reviewed. A self-adaptive Gaussian mixture model is presented in Section 4.3. In Section 4.4, spatio-temporal smoothing transform is presented. A comprehensive analysis of various shadow removal methods is given in Section 4.5. The algorithms validation is given in Section 4.6. Finally, the chapter summary is given in Section 4.7.

4.2. ZHGMM review

This section provides a brief outline of the recursive mixture model estimation procedure described by Zivkovic and Heijden (2004, 2006). First, a reasonable time adaption period of T frames (eg T=100 frames) is chosen to generate the background model so that, the training set at the time t is $X_T = \{x^{(t)}, x^{(t-1)}, \cdots, x^{(t-T)}\}$ for each pixel. For each new sample, the training data set X_T and re-estimation of the density is updated. In general, these samples contain values that belong to both the background (BG) and foreground (FG) object(s). Therefore, the estimated density is denoted as $\hat{p}(x^{(t)}|X_T, BG+FG)$. A GMM with M components (normally between 3-7, it is set as 4) is used in the experiment.

$$\hat{p}(x^{(t)}|X_T, BG+FG) = \sum_{m=1}^{M} w_m N(x^{(t)}; \mu_m, \Sigma_m) \tag{4.1}$$

where μ_m is the estimate of the mean of mth Gaussian and Σ_m is the estimate of the variances that describe the mth GMM component. For computational reasons (easily invertible), an assumption is usually made that the dimensions of X_T are independent so that Σ_m is diagonal. A further assumption is that the components (eg red, green and blue pixel values) have the same variances (Stauffer et al., 1999) so that the covariance matrix is assumed to be of the form $\Sigma_m = \sigma_m I$, where I is a 3×3 identity matrix. Note that a single σ_m may be a reasonable approximation in a linear colour space, but it may be an excessive simplification in non-linear colour spaces. Thus, in this work, the covariance of a Gaussian component is diagonal, with three separate estimates of variance. The estimated mixing weights (what portion of the data is accounted for by this Gaussian) of the mth Gaussian in the GMM at time t, denoted by w_m, are non-negative and normalized (sum to unity).

Given a new data sample $x^{(t)}$ at time t, the recursive update equations are (Titterington, 1984)

$$w_m \leftarrow w_m + \alpha\left(o_m^{(t)} - w_m\right) + \alpha c_T \tag{4.2}$$

$$\mu_m \leftarrow \mu_m + o_m^{(t)}\left(\alpha / w_m\right)\delta_m \tag{4.3}$$

$$\sigma_m^2 \leftarrow \sigma_m^2 + o_m^{(t)}\left(\alpha / w_m\right)\left(\delta_m^T \delta_m - \sigma_m^2\right) \tag{4.4}$$

where $x^{(t)} = [x_1, x_2, x_3]^T$, $\mu_m = [\mu_1, \mu_2, \mu_3]^T$, $\delta_m = [\delta_1, \delta_2, \delta_3]^T$, $\sigma_m^2 = [\sigma_1^2, \sigma_2^2, \sigma_3^2]^T$ for a 3 channel colour image, $\delta_m = x^{(t)} - \mu_m$. Instead of the time interval T, mentioned above, here the constant α defines an exponentially decaying envelope that is used to limit the influence of the old data, and note that $\alpha = 1/T$. c_T is the negative Dirichlet prior evidence weight (Zivkovic et al., 2004), which means that the existing class will be accepted only if there is enough evidence from the data for its existence. It will suppress the components that are not supported by the data and the components with negative weights will be discarded. This also ensures that the mixture weights are non-negative.

For a new sample the ownership o_m^T is set to 1 for the "close" component with largest w_m and the others are set to zero. A sample is defined "close" to a component if the Mahalanobis distance (MD) from the component is, for example, less than three. The squared Mahalanobis distance from the m*th* component is calculated as

$$D_m^2\left(x^{(t)}\right) = \delta_m^T \Sigma_m^{-1} \delta_m \tag{4.5}$$

If there are no "close" components, a new component is generated with $w_{m+1} = \alpha$, $\mu_{m+1} = x^{(t)}$, $\sigma_{m+1} = \sigma_0$, where σ_0 is some appropriate initial variance. If the maximum number of components M is reached, the component with smallest w_m will be discarded. After each weight update, using equation (4.2) to renormalize the weights so that they again sum to unity.

Usually, the foreground objects will be represented by some additional mixture components with small weights w_m. Therefore, the background model can be approximated

by the first B largest components $\hat{p}(x^{(t)}|X_T,BG)\sim\sum_{m=1}^{B}w_m N(x^{(t)};\mu_m,\Sigma_m)$. If the components are sorted to have descending weights w_m, then $B=\arg\min_b(\sum_{m=1}^{b}w_m>(1-c_f))$, where c_f is a measure of the maximum portion of the data that should be accounted for by the foreground objects without influencing the background model.

One of the significant advantages of this method is that, when something is allowed to become a part of the background, it doesn't immediately destroy the existing model of the background, which can cause 'ghost foregrounds' to appear on the uncovered background when the object moves again. The original background colour remains in the GMM until the object remains static for long enough for its weight to become larger than c_f. From equation (4.2), to be known that the object should be static for approximately $\log(1-c_f)/\log(1-\alpha)$ frames. For the experiment, c_f=0.05, α=0.001 to be set, giving a 51 frame learning period. If the object moves again before this number of frames, the distribution describing the previous background still exists with the same μ and α, albeit with a reduced weight.

4.3. Self-adaptive Gaussian mixture model

Illumination-invariant change detection model (ICDM) is a process of identifying illumination variation over time. Changing image illumination causes problems for many computer vision applications operating in unconstrained environments, and especially affects background subtraction methods. The most trivial approach for ICDM is the subtraction of intensities of two sequential video frames. The main disadvantage of such a simple method is its sensitivity to noise. Beiderman et al. (2010) proposed an illumination insensitive reconstruction and pattern recognition using spectral manipulation and K-factor spatial transforming. Xie et al. (2004) developed a change detection algorithm that deals with sudden illumination changes using order consistency. Various intensity estimation methods are compared in the paper (Withagen et al., 2010). Usability is evaluated with background classification. They give an accurate non-iterative estimate of the apparent gain factor by experimentally comparing six algorithms (weighted least square, standard least square, quotient of the average, average of the pixel-wise quotient, median of the pixel-wise quotient, median of quotient). According to simulation results, many algorithms seem to perform very well, the Median of Quotient (MofQ) performing best, both with and without outlier removal. So the MofQ will be employed in the self-adaptive Gaussian mixture model background subtraction algorithm.

For all pixels s in set S, the MofQ global illumination changing factor g between reference image i_r and current image i_c is defined as

$$g=median_{s\in S}\left(\frac{i_{r,s}}{i_{c,s}}\right) \tag{4.6}$$

A self-adaptive Gaussian mixture model (SAGMM) incorporating a modified adaptive schedule into the recursive ZHGMM learning procedure is developed. The global illumination change MofQ factor g between the learnt background and the current input image, and a counter c for each Gaussian component in the mixture model are introduced. The factor g tracks the global illumination changes and the counter c keeps tracking of how

many data points have contributed to the parameter estimation of that Gaussian. Each time the parameters are updated, a learning rate β is calculated based on the basic learning rate α and current accumulative counter of c. Given a new data sample $x^{(t)}$ at time t the recursive update equations are

$$\beta_m = \alpha(h + c_m)/c_m \tag{4.7}$$

$$\mu_m = \mu_m + o_m^{(t)}(\beta_m / w_m)\delta_m \tag{4.8}$$

$$\sigma_m^2 = \sigma_m^2 + o_m^{(t)}(\beta_m / w_m)(\delta_m^T \delta_m - \sigma_m^2) \tag{4.9}$$

$$c_m = c_m + 1 \tag{4.10}$$

$$\hat{\delta}_m = g \cdot x^{(t)} - \mu_m \tag{4.11}$$

where h is a constant (it is set as 1 in the experiments), c_m is the counter. It is increased when parameters of the Gaussian are updated. When the Gaussian is re-assigned, it is reset to 1 since the old Gaussian has perished and a new one is initiated with a single value (Lee, 2005).

The significant advantages for this approach are: i) from the dynamic learning rate update equation (4.7), it can be seen that if the background changes quickly, the value of c_m will become smaller and the new learning rate β_m will increase. So the background model will update quickly. The model will quickly achieve a good estimate of mean and variance. Maintaining a dynamic learning rate for each Gaussian component will improve convergence and approximation of a smaller data cluster. Otherwise, if the background is stable, as more data samples are included in its parameter estimation, β_m will approach the basic learning rate α, while still maintaining the same temporal adaptability, because the weights update equation (4.2) still uses the basic learning rate. ii) From formula (4.11) shows that the MD calculation will compensate for the global illumination change. This makes the MD insensitive to sudden illumination change.

4.4. Spatio-temporal pre-processing smoothing transform

An image is typically represented as a two-dimensional matrix of p-dimensional vectors, where p=1 in the gray-level case, p=3 for colour images, and p>3 for multispectral images. The space of the matrix is known as the *spatial* domain, while the gray, colour or multispectral is known as the *spectral* domain (Chen et al., 2007a, 2007b, 2007c, 2009a; Bonenfant et al. 2007; Comaniciu et al., 2002; Michaelson et al., 2007). For algorithms that use image sequences, there is also the *temporal* domain.

In order to provide spatio-temporal smoothing for each spectral component, a multivariate kernel is defined as the product of two radially symmetric kernels and the Euclidean metric allows a single bandwidth parameter for each domain.

$$K_{h_t,h_s}(x) = \frac{L}{h_s h_t} k\left(\left\|\frac{x^s}{h_s}\right\|^2\right) k\left(\left\|\frac{x^t}{h_t}\right\|^2\right) \tag{4.12}$$

where x^s is the spatial part and x^t is the temporal part of the feature vector. $k(x)$ is a common kernel profile used in both spatial and temporal domains, h_s and h_t are the kernel bandwidths, and L is the corresponding normalization constant. In order to improve stability and robustness of the ZHGMM, this Multi-Dimensional Gaussian Kernel density Transform (MDGKT) is used as a pre-process, which only requires a pair of bandwidth parameters (h_s, h_t) to control the size of the kernel, thus determining the resolution and time interval of the ZHGMM. $k(x)$ is set as a Gaussian kernel in the experiment.

4.5. Shadow removal

The background subtraction algorithm is susceptible to both global and local illumination changes such as shadows and highlight reflections (specularities). These often cause subsequent processes, such as tracking and recognition, to fail, and thus it is desirable to discriminate between targets and their shadows. Prati et al. (2003) present a comprehensive survey of moving shadow detection approaches. It is important to recognize the type of features utilized for shadow detection. Basically, these features are extracted from three domains: spectral, spatial and temporal. Approaches can exploit different spectral features, i.e. using gray level or colour information (KaewTraKulPong et al., 2001; Horprasert et al., 1999; Mikic et al., 2000). Some approaches improve performance by using spatial information working at a region level or at a frame level instead of pixel level (Elgammal et al., 2000). Finlayson et al. (2002) proposed a method to remove shadows from a still image using illumination invariance.

Several different shadow removal methods are formulated in this section, working in different colour spaces. For the sake of clarity, two different foreground segmentations are distinguished as: segmentation **FG1**, is the foreground segmentation which includes shadows, while **FG2** is the foreground segmentation after shadows removal.

4.5.1. Working with RGB and normalized RGB colour space

(i) RGB colour space Horprasert et al. (1999) describe the deviation between the expected RGB values of a pixel and the measured RGB values as a *distortion*, such as could be caused by the shadow cast by a foreground object onto a true background pixel. They decompose this distortion measurement in RGB space into two components, brightness distortion and chromaticity distortion.

The observed colour vector is projected onto the expected colour vector, and the *i*th pixel's brightness distortion ξ_i is a scalar value (less than unity for a shadow) describing the fraction of remaining 'brightness'. This may be obtained by minimizing

$$\phi(\xi_i) = (I_i - \xi_i E_i)^2 \tag{4.13}$$

where $I_i = [I_{Ri}, I_{Gi}, I_{Bi}]$ denotes the *i*th pixel value in RGB space, $E_i = [\mu_{Ri}, \mu_{Gi}, \mu_{Bi}]$ represents the *i*th pixel's expected (mean) RGB value. The solution to equation (4.13) is an alpha value equal to the inner product of I_i and E_i, divided by the square of the Euclidean norm of E_i.

Considering balancing colour bands by rescaling the colour values by the pixel standard deviation σ_i, the brightness and chromaticity distortion become

$$\xi_i = \frac{g\sum_{k\in[R,G,B]} I_{ki}\mu_{ki}/\sigma_{ki}^2}{\sum_{k\in[R,G,B]}\left[\mu_{ki}/\sigma_{ki}\right]^2} \qquad CD_i = \sqrt{\sum_{k\in[R,G,B]}\left(gI_{ki}-\xi_i\mu_{ki}\right)^2/\sigma_{ki}^2} \tag{4.14}$$

Then a pixel in the foreground segmentation (**FG1**) may be classified as either a shadow or highlight on the true background as follows:

$$\begin{cases} Shadow\ CD_i < \beta_1\ and\ \xi_i < 1 \\ Highlight CD_i < \beta_1\ and\ \xi_i > \beta_2 \end{cases} \tag{4.15}$$

β_1 is a selected threshold value, used to determine the similarities of the chromaticity between the SAGMM background and the current observed image. If there is a case where a pixel from a moving object in the current image contains a very low RGB value, then this dark pixel will always be misclassified as a shadow, because the value of the dark pixel is close to the origin in RGB space and all chromaticity lines in RGB space meet at the origin. Thus a dark colour point is always considered to be close or similar to any chromaticity line. A threshold β_2 is introduced for the normalized brightness distortion to avoid this problem. This is defined as: $\beta_2 = 1/(1-\varepsilon)$, where ε is a lower band for the normalized brightness distortion. An automatic threshold selection method was provided by Horprasert et al. (1999).

(ii) Normalized RGB Colour information is useful for suppressing shadows and highlights from the detection, by separating colour information from brightness information. Given three colour variables, R_i, G_i and B_i, the chromaticity coordinates are $r_i = R_i/(R_i+G_i+B_i)$, $g_i = G_i/(R_i+G_i+B_i)$ and $b_i = B_i/(R_i+G_i+B_i)$, where $r_i + g_i + b_i = 1$ (Elgammal et al., 2000). $s_i = R_i + G_i + B_i$ is a brightness measure. Let a pixel value of the SAGMM background be $< r_i, g_i, s_i >$. Assume that this pixel is covered by a shadow in frame t and let $< r_{ii}, g_{ii}, s_{ii} >$ be the observed value for this pixel at this frame. Then, for a pixel in the foreground segmentation (**FG1**):

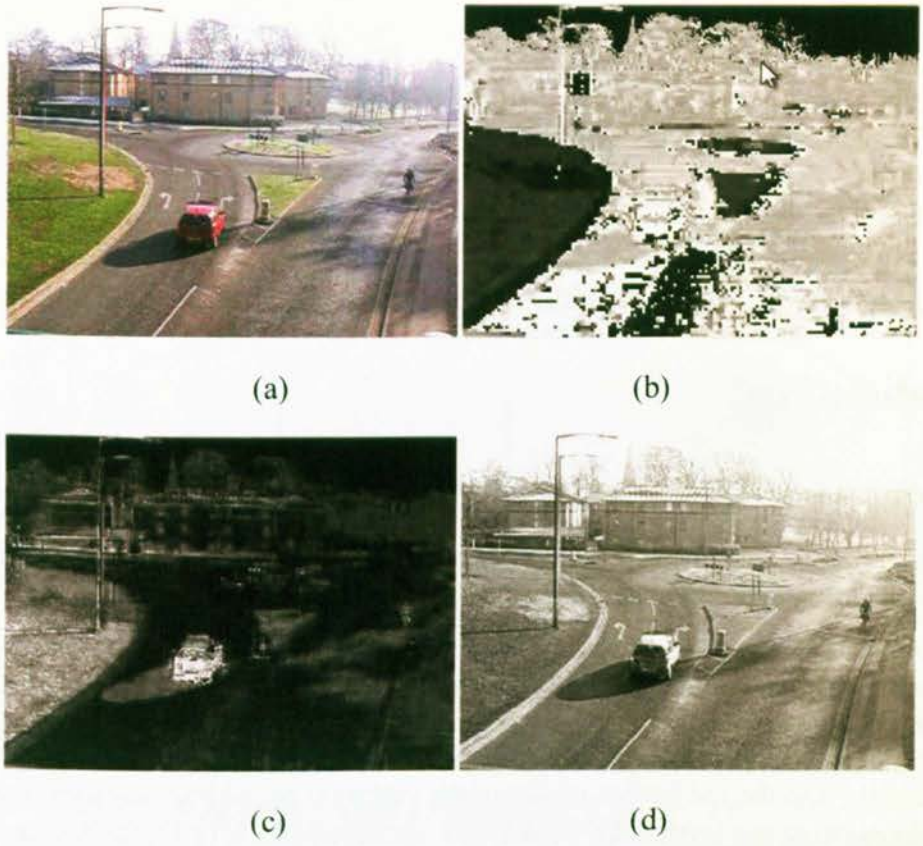
$$\begin{cases} Shadow & \beta_1 < s_{ii}/s_i \le \beta_2 \\ Highlight & \beta_3 < s_{ii}/s_i \end{cases} \tag{4.16}$$

where β_1, β_2 and β_3 are selected threshold values used to determine the similarities of the normalized brightness between the SAGMM background and the current observed image. It is expected that, in the shadow area, the observed value s_{ii} will be darker than the normal value s_i, up to a certain limit. On the other hand, in the highlight area, $s_{ii} > s_i$. So that $\beta_1 > 0, \beta_2 \le 1$ and $\beta_3 > 1$. These thresholds may be adapted for different environments (e.g. indoor image, outdoor image or brightness of the source light).

4.5.2. Working with HSV colour space

The HSV (Hue-Saturation-Value) colour space explicitly separates chromaticity and luminosity and has proven easier than RGB space to set a mathematical formulation for shadow detection (Prati et al., 2003; Cucchiara et al., 2003 and 2001). HSV space is more closely related to the human visual system than RGB (Herodotou et al., 1998) and it is more

sensitive to brightness changes due to shadows. According to the following consideration to check each pixel in **FG1**, that initially has been segmented as foreground, if it is a shadow on the background or not. If a shadow is cast on a background, the hue and saturation components change, but within a certain limit. The difference in saturation is an absolute difference, while the difference in hue is an angular difference.



(a) (b)

(c) (d)

Figure 4.1. (a) is the RGB image and (b)-(d) are its corresponding H, S and V images.

$$\begin{cases} Shadow\ \beta_1 < V_{Ii}/V_{Bi} < \beta_2\ and |H_{Ii} - H_{Bi}| < \tau_H\ and |S_{Ii} - S_{Bi}| < \tau_S \\ Highlight\quad V_{Ii}/V_{Bi} > \beta_3\quad and |H_{Ii} - H_{Bi}| < \tau_H\ and |S_{Ii} - S_{Bi}| < \tau_S \end{cases} \tag{4.17}$$

with $0 < \beta_1, \beta_2, \tau_H, \tau_S < 1$ and $\beta_3 > 1$. Intuitively, this means that a shadow darkens a covered point, and a highlight brightens a covered point, but only within a certain range. Prati et al. (2003) stated that the shadow often has a lower saturation. But the experimental results show that sometimes the shadow has a higher saturation than that of the background. In the experiment, the mean values of saturation in the shadow and background area are 0.0027 and 0.0003 respectively. However, a shadow or highlight cast on a background does not change its hue and saturation as significantly as intensity. Figure 4.1 shows a sample RGB image and its corresponding HSV image. Obviously, the shadow has higher saturation than that of the background in the S image.

4.5.3. Working with YCbCr and Lab colour spaces

The luminance and chrominance (YCbCr) colour space is used in video systems. There are no chromaticity coordinates in YCbCr space, Y is the luminance and Cb and Cr are chrominance colour values. If a shadow is cast on a background, the shadow darkens the point. The luminance distortion is $\alpha_i = Y_{Ii}/Y_{Bi} < 1$, and chrominance components difference is $CH_i = \left(\left|C_{bi}^I - C_{bi}^B\right| + \left|C_{ri}^I - C_{ri}^B\right|\right)/2 < \beta_1$, where Y_{Ii}, C_{bi}^I, C_{ri}^I and Y_{Bi}, C_{bi}^B, C_{ri}^B are Y, Cb, Cr components in the current image and SAGMM background respectively. A pixel in the foreground segmentation (**FG1**) is classified as follows:

$$\begin{cases} Shadow \;\; \alpha_i < 1 \;\; and CH_i < \beta_1 \\ Highlight \alpha_i > \beta_2 \; and CH_i < \beta_1 \end{cases} \tag{4.18}$$

where $\beta_1 < 1$ and $\beta_2 > 1$. There is a similar criterion for shadow removal in Lab colour space.

4.6. Validation

4.6.1. Background learning using MDGKT

The purpose of the following experiments is to show the improvement in terms of stability and accuracy of GMM using the MDGKT.

A sample RGB image is shown in Figure 4.2(a). The red asterisk in the centre of the blue rectangle shows a sample pixel stream over a video (596 frames) in an area where no intruding object appears over time. The variation of red and blue values of a pixel stream is shown in Figure 4.2(e) and (f). The black curves show the variation of the original red and blue components, and the red curves illustrate the variation of red and blue components in the MDGKT image. Here the bandwidth parameters are set as $(h_s, h_t) = (5,5)$. A Gaussian kernel with standard deviation (*std*) of 0.5 was chosen as the kernel profile. The *std* of the red and blue colour components of original image is 1.834 and 1.110, but after the MDGKT processing the *std* of the output of red and blue colour components is only 1.193 and 0.832. Obviously, MDGKT reduces the *std* figures, illustrating the smoothing effect of the spatio-temporal filtering in the time domain. Figure 4.2(c) and (d) show the scatter plots of the original and MDGKT image (red, blue) values of the same pixel. Figure 4.2(d) shows that the distribution of MDGKT image is more localised within two Gaussian components of the mixture model, illustrating the effect of the spatio-temporal filtering in the spectral domain. The mixture of these two Gaussians for the blue colour component of the original pixel and the estimated GMM distribution using MDGKT are shown in Figure 4.2(b).

The MDGKT algorithm described in Section 4.4 allows identifying the foreground pixels and background pixels more accurate and robust. This procedure is effective in determining the boundary of moving objects, thus moving regions can be characterized not only by their position, but also size, aspect ratio, moments and other shape and colour information. These characteristics can be used for later processing and classification, for example, using a support vector machine (Joachims et al., 2006). A dynamic scene is used to analyse the performance of the algorithm. The results are shown in Figure 4.3. (a) and (d) are original images. One is an outside scene, another is an inside scene. (b) and (e) are the results

of the ZHGMM algorithm. (c) and (f) are the results of the ZHGMM using MDGKT algorithm. Note that the results shown are without the application of any post-processing.

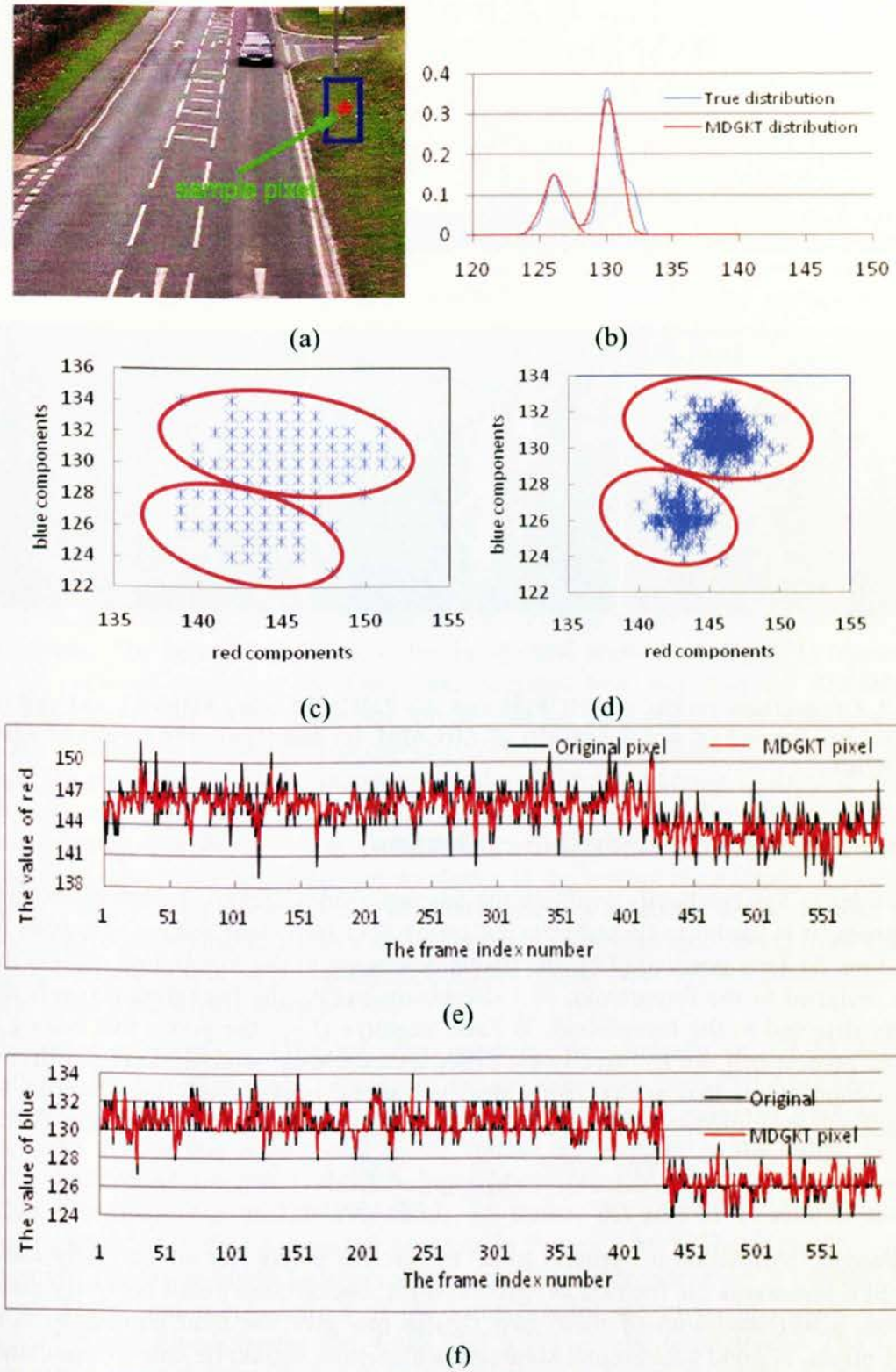


Figure 4.2. A sample image and the variation of a pixel value over time. (a) A sample image. (b) GMM distribution of the blue sample pixel. (c) and (d) scatter plots of the red and blue colour

components of the sample pixel in original image and MDGKT image, respectively. (e) and (f) the variation of red and blue colour components over time

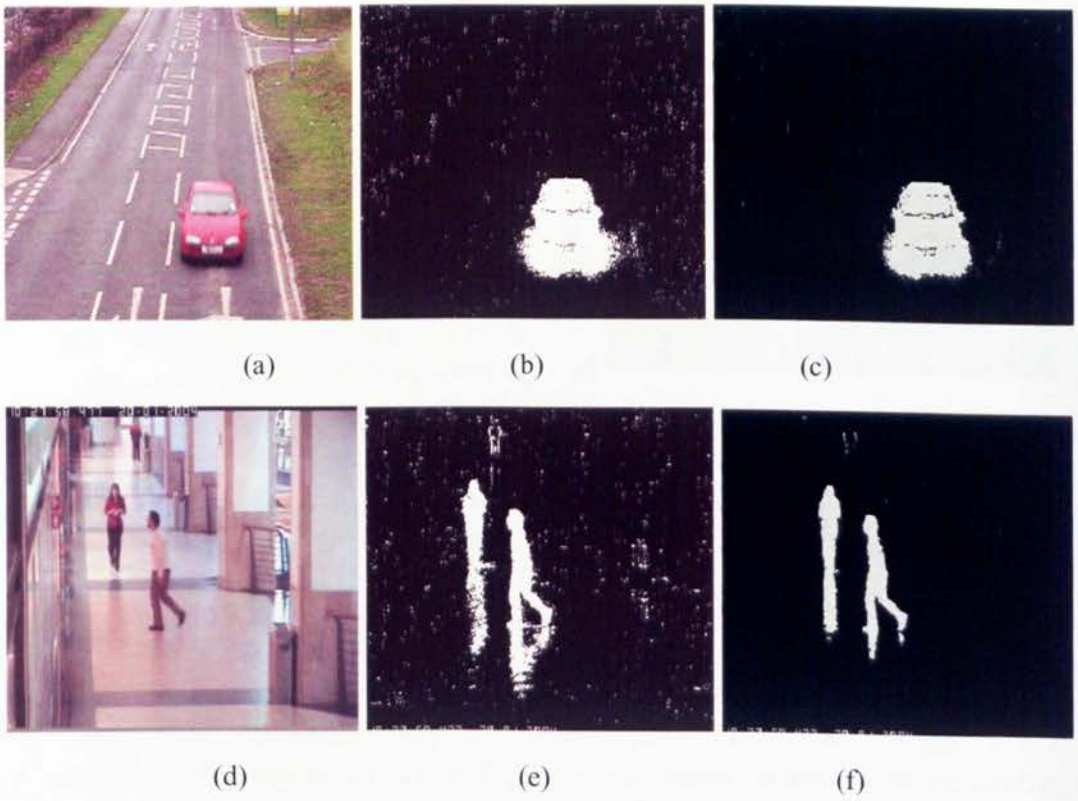


Figure 4.3. Comparison results of ZHGMM and the ZHGMM using MDGKT. (a) and (d) are original images, (b) and (e) are the results of ZHGMM, (c) and (f) are the results of ZHGMM using MDGKT.

4.6.2. Shadow removal algorithm evaluation

In order to systematically evaluate the various shadow removal methods in different colour spaces, it is useful to identify the following four important quality measures for the segmentation. 1) True positive (TP): the pixels belonging to the foreground objects that are correctly assigned to the foreground. 2) False positive (FP): the background pixels that are incorrectly detected as the foreground. 3) False negative (FN): the pixels that belong to the foreground objects that are incorrectly classified as the background. 4) True negative (TN): the background pixels that are correctly detected as the background. Two metrics for the segmentation evaluation: true positive rate (TPR) and specificity (SPE), defined as follows:

$$TPR = \frac{TP}{TP + FN} \qquad SPE = \frac{TN}{FP + TN} \tag{4.19}$$

TPR represents the fraction of 'ground truth' foreground pixels that are correctly classified, whereas SPE represents the fraction of 'ground truth' background pixels correctly classified. If required, a weighted sum of these two figures can give the total number of correctly classified pixels. A good background subtraction algorithm should be able to simultaneously generate both good TPR and SPE values.

This section compares the performance of the described algorithms in section 4.5 on several videos of both indoor and outdoor scenes, using an image size of 320×240. A quantitative comparison of two GMMs (ZHGMM and ZHGMM using MDGKT) with different shadow removal methods is presented. A set of videos to test the algorithms is chosen, and in order to compute the evaluation metrics regarding the ground truth for each frame. This is obtained by manually segmenting the images as foreground, background and shadow regions. Red and blue pixels illustrate the foreground and shadow regions. 41 ground truth frames are prepared in a 'walking people' sequence, and 26 in a 'moving car' sequence. The frames which have been used for GMM learning are not annotated. Sample frames of each sequence and their ground truth mark-up are given in Figure 4.4. All shadow removal methods in five colour spaces using the two GMMs have been fully implemented. Quantitative results for true positive rate (TPR) and specificity (SPE) metrics are reported in Table 4.1. The larger number is better. TPR=1.0 and SPE=1.0 for perfect background subtraction results.

Figure 4.5 shows an example of object segmentation results from the 'walking pepole' video using the ZHGMM with MDGKT algorithm in different colour spaces. Table 4.1 and Figure 4.5 illustrate that the results in RGB space provide the best segmentation. The object segmentation results in RGB colour space are shown in Figure 4.6, which intuitively illustrates good performance.

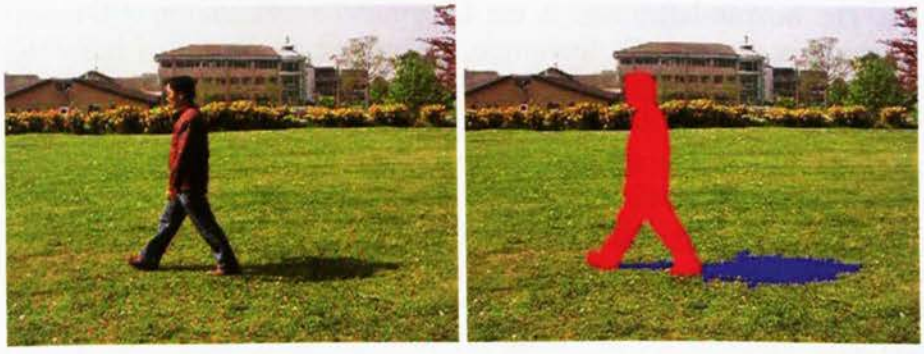
Figure 4.7 shows sample frames (9, 17, 24 and 30) of the 'walking people' video and the 'moving car' video (3, 8 15 and 20) with strong shadow. In this figure, each two-by-two block of images refers to the same frame in the original video. The top-left image is the original frame. The bottom-left image is the foreground segmentation (**FG1**) results. In this image, all coloured pixels are the foreground segmentation output of the ZHGMM using MDGKT algorithm, while the black pixels represent the modeled background. The coloured pixels are categorized as foreground object (coloured yellow), shadow (coloured green) or highlight (coloured red) by the shadow removal algorithm operating in RGB colour space. The shadow and highlight pixels are then removed and this is then followed by a post-processing binary morphology stage of dilatation and erosion to remove sparse noise. This gives the final foreground segmentation, as shown in the bottom right image of each two-by-two block. Finally, the top-right image in each block is a synthetic image, created by using the final foreground segmentation as a mask to extract the foreground object from the original frame, superimposing this on the background model (mean value of each pixel). Clearly these synthetic images are largely shadow-free.

In order to illustrate MDGKT can improve the accuracy of GMM background subtraction in RGB colour space in terms of camera vibration, both ZHGMM and ZHGMM with MDGKT is compared using a video acquired by a pole mounted road side CCTV camera under very strong wind weather. 781 frames are included in the video. 113 frames including foreground objects (vehicles) have been manually annotated to evaluate the algorithm. The image size is 320x240 pixels, 25 frames per second. A sample image with annotated ground truth (red silhouette) is given in Figure 4.8(a). The white 'X' shows the sample point. The ground truth location of this sample pixel stream over 781 frames has been annotated. The x and y coordinates of the sample point are given in Figure 4.8(b). The variance of x and y coordinates are 4.918 and 5.115 pixels, respectively. The maximum variance range along x direction is [-8, 9] pixels, along y direction is [-11, 6] pixels. Figure 4.8(c) and (d) illustrate the background subtraction results from ZHGMM using MDGKT approach. Figure 4.8(e) and (f) show the background subtraction result from ZHGMM. (c)

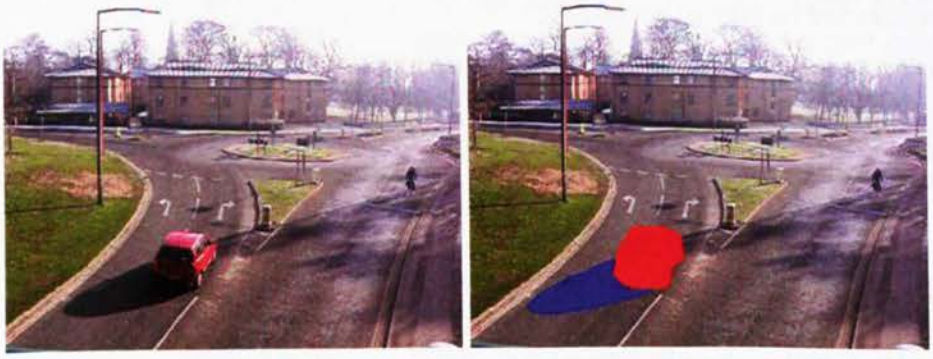
and (e) are the raw background subtraction result. Yellow pixels are foreground, green pixels are shadow and black pixels are background. Shadow removal followed by a post-processing binary morphological opening (dilation and erosion) to remove noises and small area objects (the area of the objects less than 200 pixels) to create final foreground objects masks which are given in Figure 4.8(d) and (f). The TPR and PRE of the ZHGMM with MDGKT are 0.7840 and 0.9513, respectively. The TPR and PRE of the ZHGMM algorithm are 0.5823 and 0.9760, respectively. This experimental results illustrate that MDGKT can improve the accuracy of GMM in terms of vibration effectively.

Table 4.1. Experimental quantitative results

	ZHGMM		ZHGMM using MDGKT	
	TPR	SPE	TPR	SPE
RGB	**0.855**	**0.984**	**0.955**	**0.985**
Lab	0.717	0.983	0.850	0.985
YCbCr	0.618	0.981	0.675	0.981
Normalized RGB	0.608	0.963	0.636	0.971
HSV	0.504	0.967	0.633	0.971



(a)



(b)

(a) Walking people (b) Moving car

Figure 4.4. A sample ground truth image (the red pixels illustrate the foreground and the blue pixels illustrate shadow region).

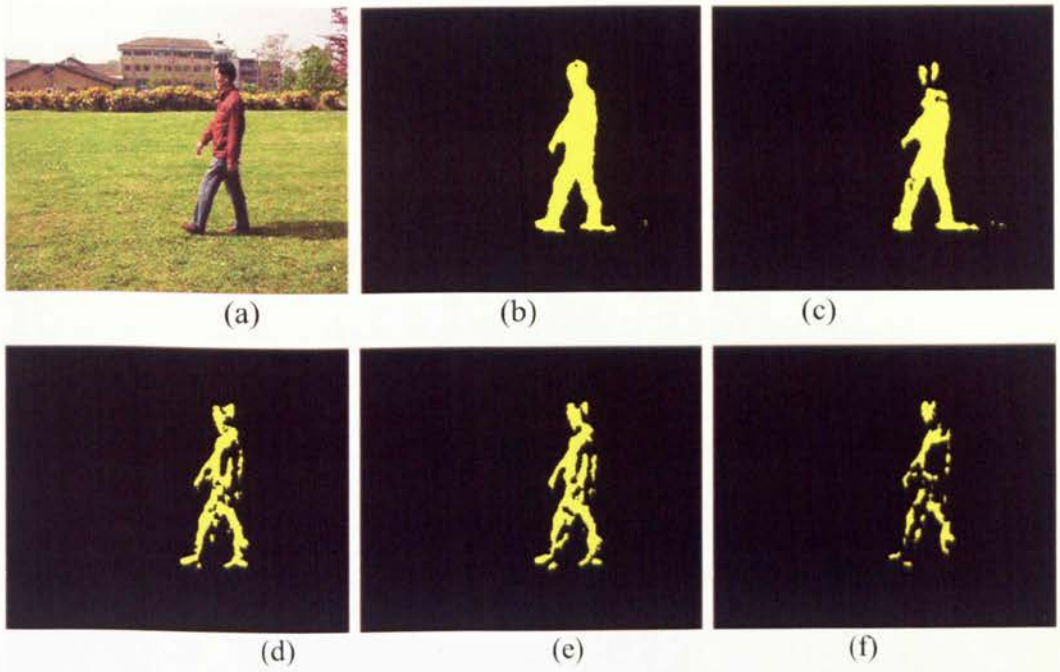


Figure 4.5. Visual results of background subtraction from walking people video using ZHGMM with MDGKT in different colour spaces. (a) original image. (b)-(f) are the results from RGB, Lab, YCbCr, Normalized RGB and HSV colour space, respectively.

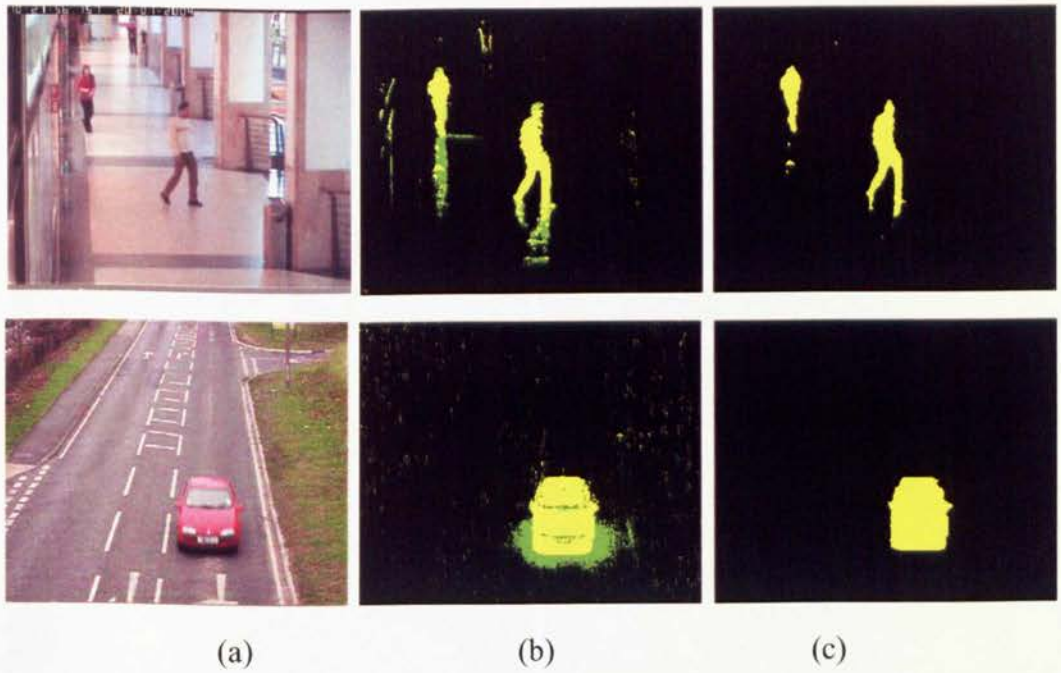


Figure 4.6. Object segmentation results for a set of videos with light shadow in RGB colour space. (a) original input image, (b) background subtraction results (background pixels coloured black, foreground pixels coloured yellow and shadow pixels coloured green), (c) results of shadow removal and morphological process of erosion and dilation for the results from (b)

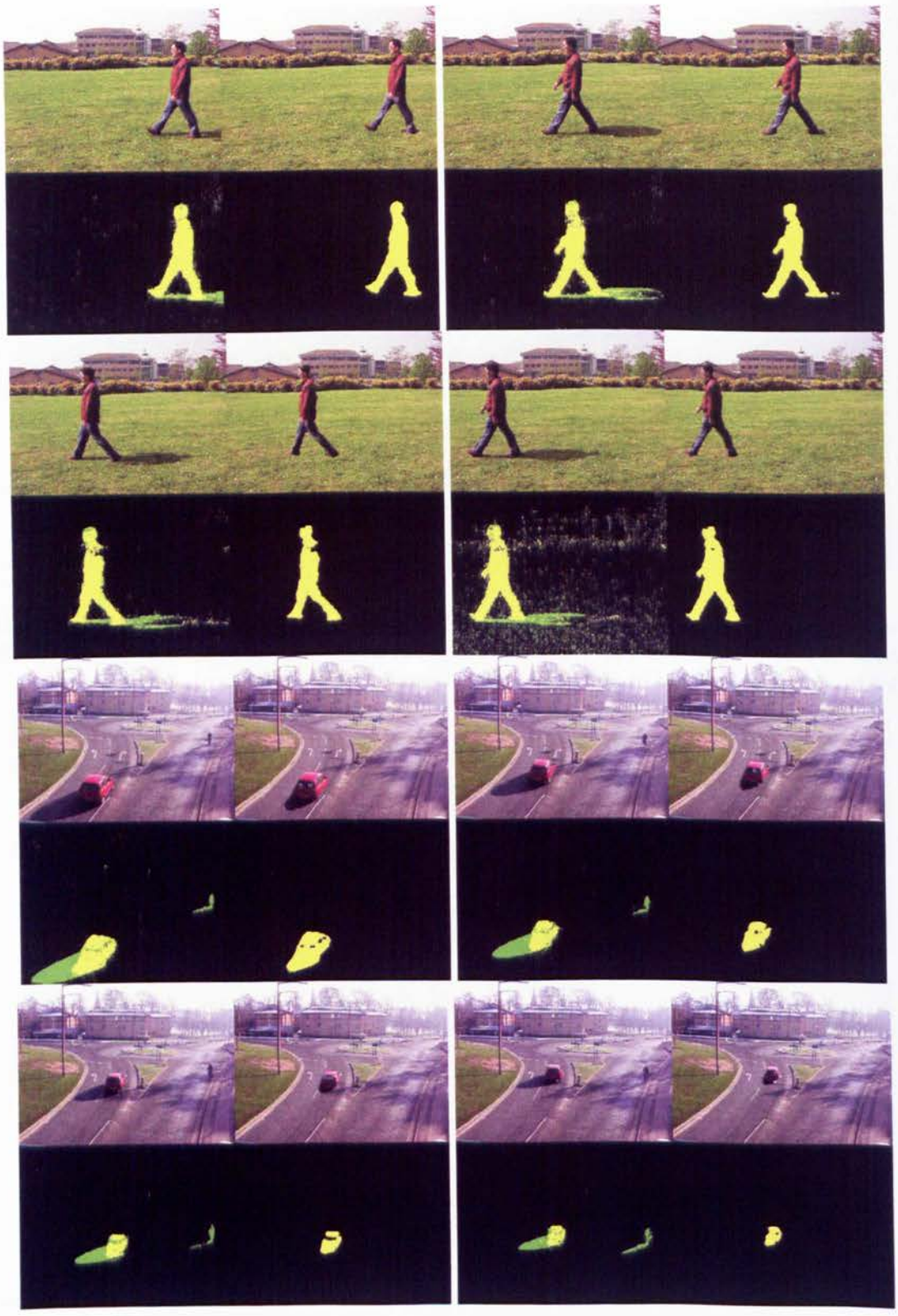


Figure 4.7. Foreground segmentation results in RGB colour space. The top-left image is the original frame. The bottom-left image is the foreground segmentation results. The black pixels represent the modeled background. The foreground object (coloured yellow), shadow (coloured green) or highlight (coloured red). The bottom right image is the final foreground segmentation after shadow and highlight reflection removal. The top-right image is a synthetic shadow free image

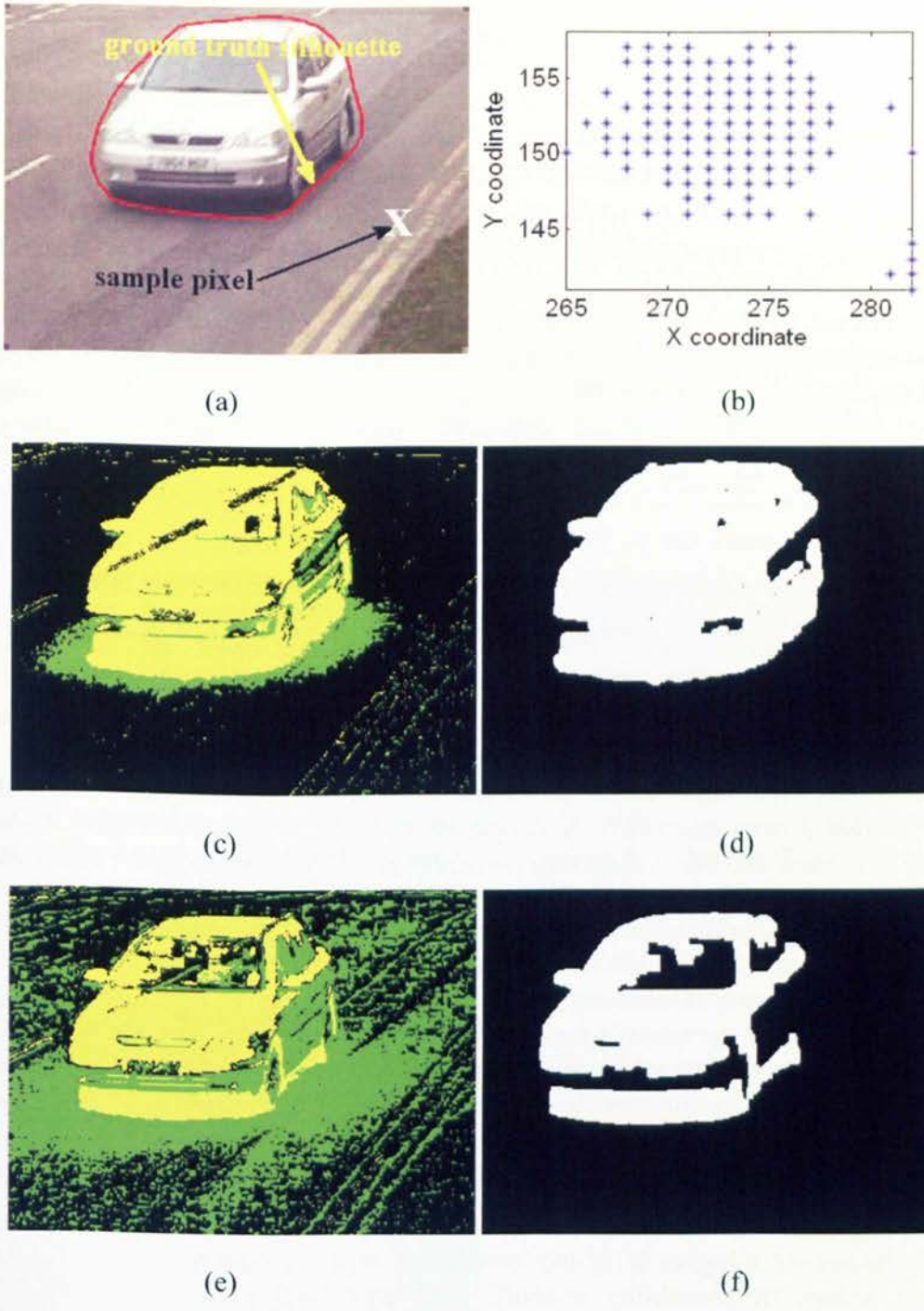


Figure 4.8 Experimental results for a video acquired by a vibrated road side CCTV camera. (a) A sample image with annotated ground truth (red silhouette) and the sample pixel used to annotate ground truth location (white 'x'). (b) Scatter plot x and y coordinates of sample pixel stream over the video. (c) and (d) background subtraction result from ZHGMM using MDGKT. (e) and (f) background subtraction result from ZHGMM. The foreground object (coloured yellow), shadow (coloured green) and background (coloured black).

4.6.3. Comparison of ZHGMM and SAGMM

The objective of the experiments in this section is to illustrate the advantages of SAGMM comparing with ZHGMM. An online dynamical learning rate and global illumination of background model adaptation to deal with fast changing scene illumination.

To get a sense of how well the background model, estimated by GMM, matches the original input image, the similarity between the predicted background and the original input frame is compared. The mean of the Euclidean distance is used as the similarity measure. A test area is selected from a rectangular area that is placed in a part of the scene where the background is always visible, i.e. no intruding object appears (the black rectangular area in Figure 4.9(a)). The similarity is defined as:

$$S = \frac{\sum_{j \in A} \left(\sum_{p \in [R,G,B]} \left(I_{j,p}^{cr} - I_{j,p}^{bg} \right)^2 / 3 \right)^{1/2}}{A_{\Delta}} \tag{4.19}$$

where, S is the average colour components difference over the rectangular area A; A_{Δ} is the area of A in pixels. $I_{j,p}^{cr}$ and $I_{j,p}^{bg}$ represent the jth pixel value in R, G or B component in the area A in the current input reference image and current modeled background, respectively.

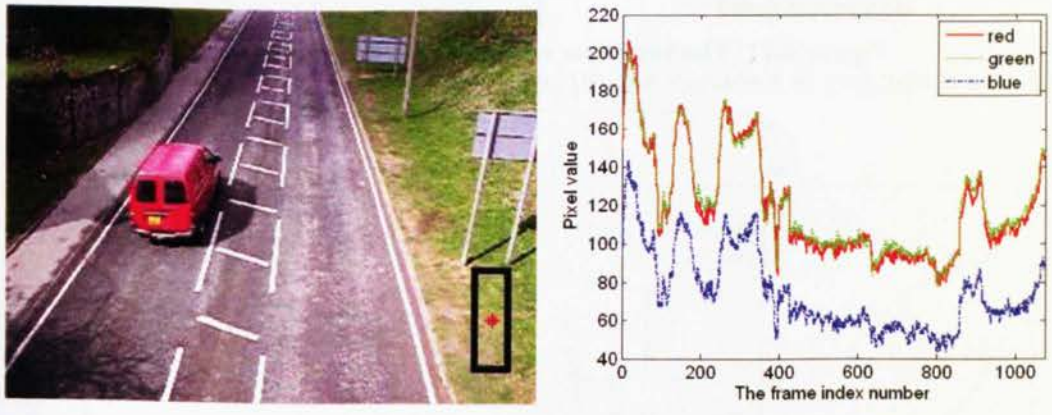
Extensive experiments have been performed to compare the results of SAGMM with ZHGMM. For the first experiment, the input data is a traffic video captured under intermittent cloudy conditions, where the clouds move in front of the sun. A total of 1079 frames are analysed from this video. A sample RGB frame is shown in Figure 4.9(a). The variation of red, green and blue values of a particular pixel (red asterisk in the centre of black rectangle) in an area where no intruding object appears over time is shown in Figure 4.9(b). It can be seen that the range of intensity variation is very large. The standard variation of R, G and B is 28.3, 27.4 and 21.2, respectively, over a period of 42.9 seconds. The ground-truth data was created by Viper (Doermann et al., 2000). The receiver operating characteristic (ROC) is used to analyse the model's performance. For a GMM, any input distribution can be converted to ROC curves. One may combine multiple ROC plots for different values of some of the fixed parameters. In this case, the two parameters of most significant interest are the threshold MD and the learning rate α. For a typical fixed α=0.001, if $c_f = 0.1$, the moving object should be static for at most $\log(1-c_f)/\log(1-\alpha) \approx 105$ frames before it will merge into the background. As the Mahalanobis distance increases, DTR and FAR decrease accordingly. However, a higher DTR and lower FAR are more important. MCC can be used to explicitly select the probability tradeoff. The large number is better. A MCC of 1 represents a perfect prediction. The variation of MCC corresponding to a different threshold for MD is given in Figure 4.10(a). It shows that the SAGMM is much better than ZHGMM for all thresholds MD $\in [1, 40]$, (the increment of MD is 2). The maximum MCC of SAGMM is 0.42, but the maximum MCC for ZHGMM is 0.38. Each point of MCC here is the mean of each threshold across the entire video (of which only 484 frames include the moving vehicle). Obviously, the best cutting point threshold is MD=7. Under the optimal threshold of MD, the ROC curve is shown in Figure 4.10(b). The lower rate of FAR is more interesting, for instance, FAR<0.021, illustrated by the green dashed vertical line in Figure 4.10(b), the DTR of SAGMM is much higher than that of ZHGMM. At the point of FAR = 0.021, the DTR of SAGMM is 0.7301, but the DTR of ZHGMM is only 0.5683. The comparison of ACC and

JC values in Figure 4.11 also shows the advantages of the algorithm. In order to get stable and reasonable performance metrics, the results during the learning stage weren't included in the performance metric calculations above.

The box plot figure of the similarity distance between modeled background (SAGMM and ZHGMM) and original input image is given in Figure 4.12. The red solid line is the median of the results. The edges of the box are the 25th and 75th percentiles. The mean is 12.25 pixels and the *std* is 5.22 pixels for the SAGMM background. The mean is 18.52 pixels and the *std* is 8.13 pixels for the ZHGMM background. It shows that the proposed method has a good background prediction.

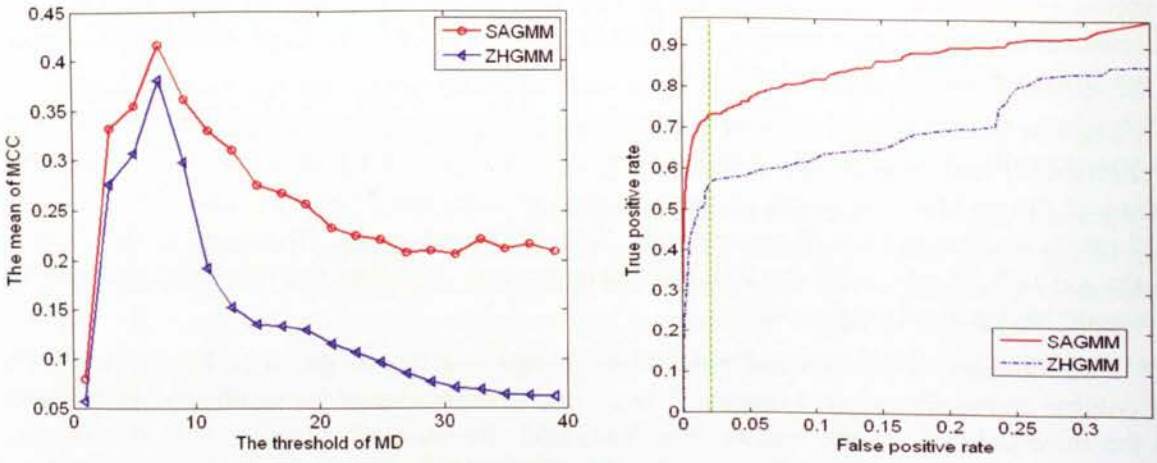
Two samples of background subtraction image results are given in Figure 4.13. The first column shows the original images. The second column shows the results from ZHGMM, and the third column are the results from SAGMM. Because of sudden global illumination change, it can be seen from the first row that the ZHGMM performs poorly, but SAGMM can detects most of the foreground.

The second experiment was performed using part of the video from i-LIDS (image library of intelligent detection systems) dataset which is provided by the Home Office of the United Kingdom. A sample image and the variation of a pixel (red asterisk) values are shown in Figure 4.14. This shows that the variation of intensity is smaller than in the previous sample video, but it is still large. There are 2000 frames included in the video. 494 randomly selected frames were annotated using Viper to create ground-truth data. Parameters α =0.001, MD=5 were used for this experiment. The ROC and JC curve are shown in Figure 4.15. A sample of image results is given in Figure 4.16. It illustrates that the SAGMM can get better background subtraction results than that of ZHGMM, although some heavy shadows still included in the results from SAGMM. Figure 4.17 shows that the algorithm proposed in this chapter is able to detect highlight reflections. The coloured pixels are categorized as foreground object (coloured yellow), shadow (coloured green) or highlight (coloured red) using the shadow removal algorithm operating in RGB colour space.



(a) (b)

Figure 4.9. (a) A sample image. (b) The variation of red, green, blue of a pixel (red asterisk in the centre of black rectangle) value over time. The black rectangle is used to measure the similarity between the modeled background and the original input image.



(a) (b)

Figure 4.10. (a) The variation of MCC corresponds to different threshold of MD. (b) ROC curve.

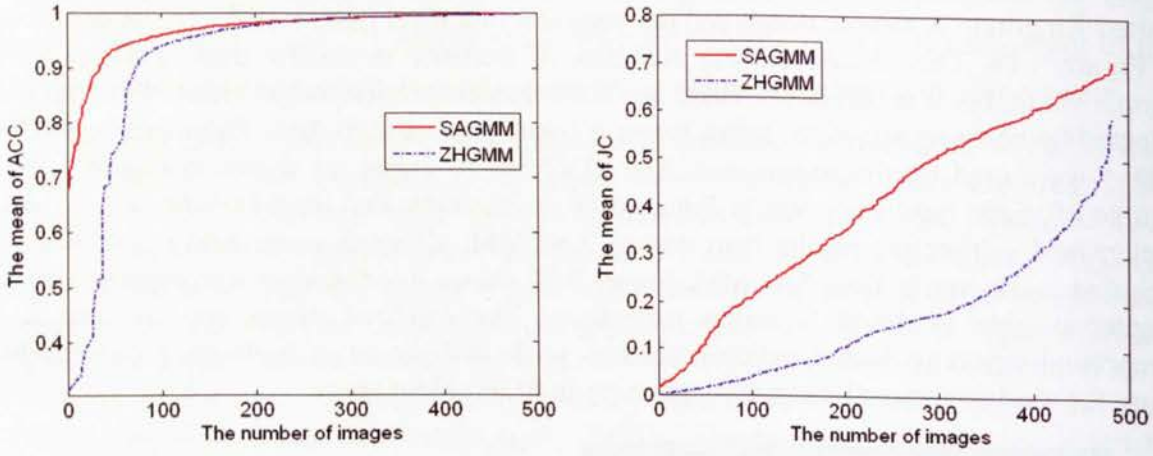


Figure 4.11. The variation of ACC (left) and JC (right).

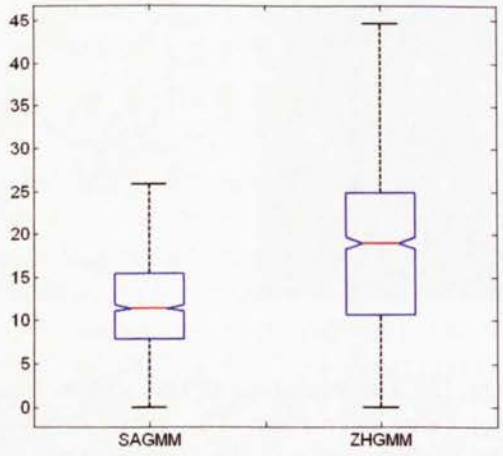


Figure 4.12. The comparison of similarity measurements between modeled background and original input image.

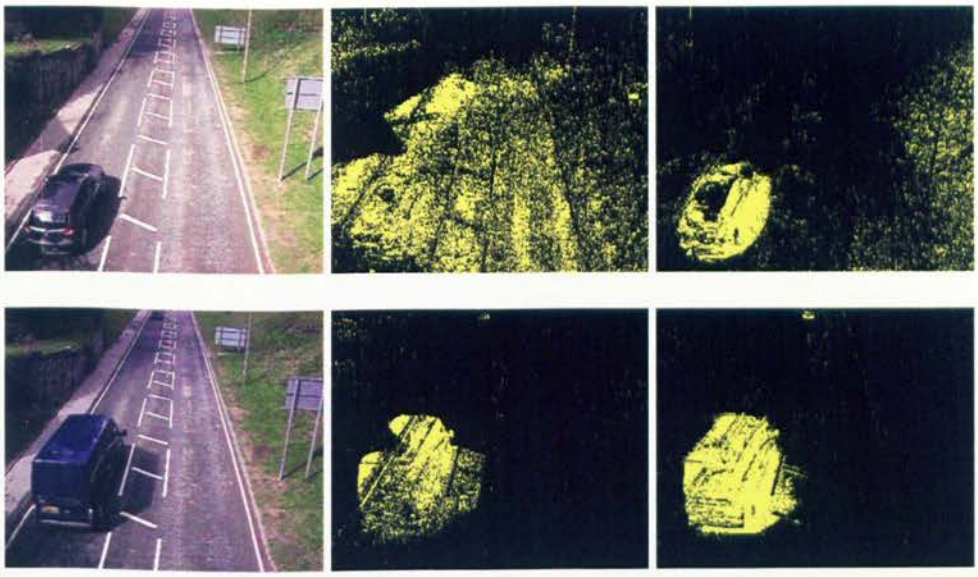
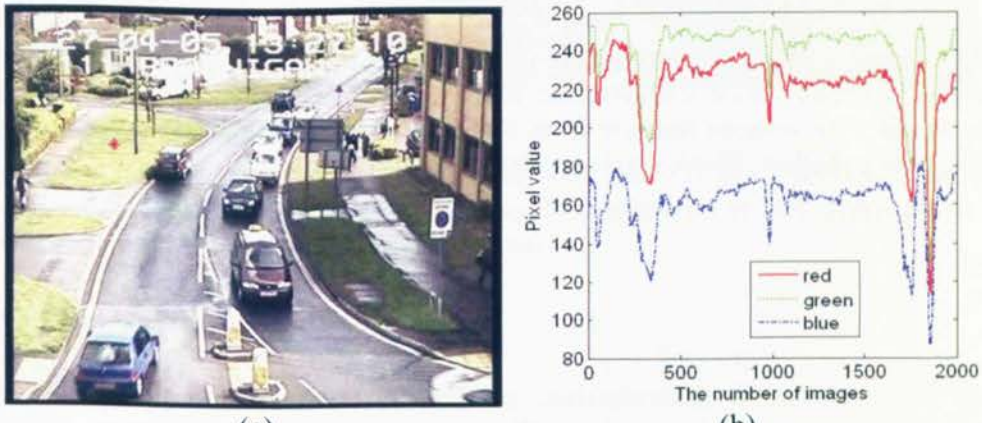


Figure 4.13. Example frames (#151 and #189). The first column is original images. The second column is the results from ZHGMM. The third column is the results from SAGMM.



(a) (b)

Figure 4.14. (a) A sample image of i-LIDS data, (b) the variation of red, green and blue of a pixel (red asterisk in the image) value over time.

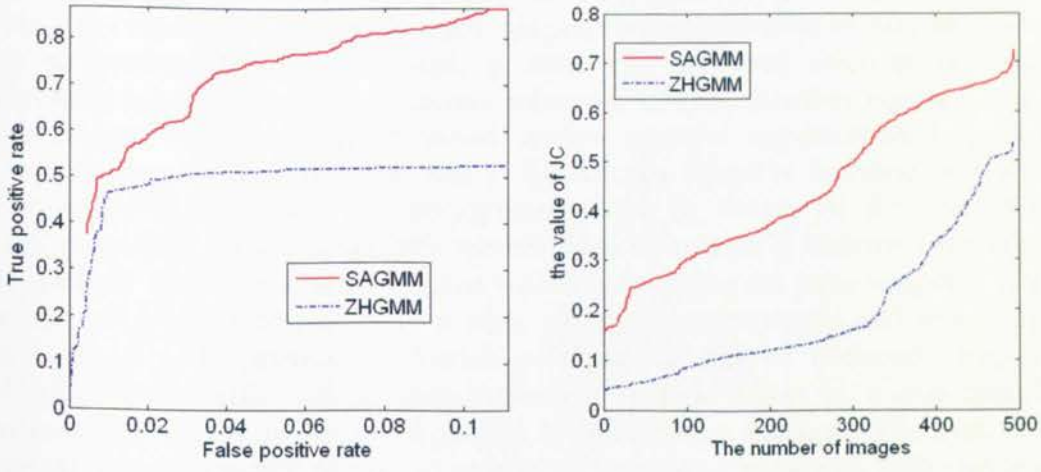
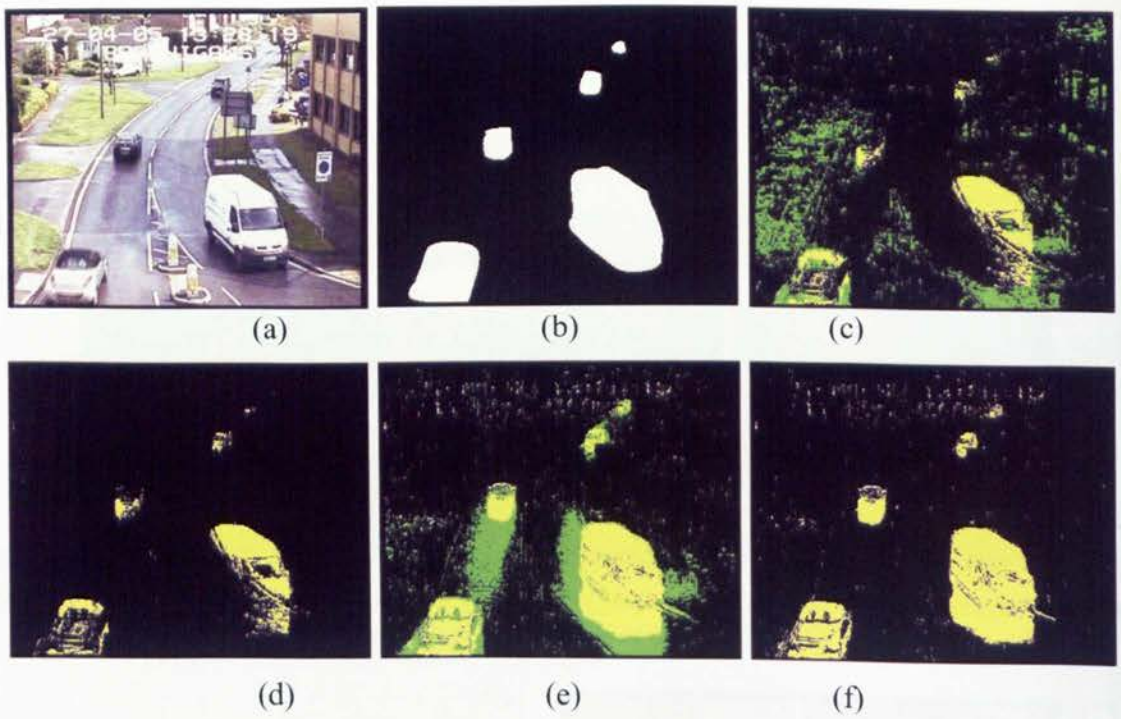


Figure 4.15. ROC curve (left) and JC variation (right) of i-LIDS data set.



(a) (b) (c)

(d) (e) (f)

Figure 4.16. Example frame. (a) Original image. (b) Annotated ground truth. (c) and (d) background subtraction result with/without shadow from ZHGMM. (e) and (f) background subtraction result with/without shadow from SAGMM. The yellow pixels are foreground and the green pixels are shadow. Black pixels are detected background.

(a) (b)

(c) (d)

Figure 4.17. (a) and (c) are the input images. (b) and (d) are corresponding results of background subtraction. Yellow pixels are belong to foreground object, green pixels belong to shadow and red pixels belong to highlight reflections.

4.7. Summary

Online learning of adaptive GMMs on nonstationary distributions is an important technique for moving object segmentation. This chapter has presented two background subtraction methods. One is an adaptive Gaussian mixture model using a multi-dimensional spatio-temporal Gaussian kernel smoothing transform for background modeling in moving object segmentation. The model update process can robustly deal with slow light changes (from clear to cloud or vice versa), blurred images, camera vibration in very strong wind, and difficult environmental conditions, such as rain. The proposed solution has significantly enhanced segmentation results over several recursive GMMs. Another one is a self-adaptive Gaussian mixture model with an improved shadow removal algorithm to deal with sudden illumination changes. The algorithm has a dynamically adaptive learning rate and models global illumination changes of the background frame by frame. At the cost of only one additional parameter per Gaussian, this modification dramatically improves the convergence and the accuracy of background subtraction whilst maintaining the same temporal adaptability. Comprehensive analysis of results in a wide range of environments and colour spaces for shadow removal and statistical evaluation performance metrics obtained using annotated ground truth from traffic videos shows that the method leads to a significantly better performance than others, in situations subject to illumination changes. The system has been successfully used to segment objects in indoor and outdoor scenes with both strong and light shadows, and highlight reflections.

5. Road Vehicle Type Categorization

5.1. Introduction

Traffic monitoring is an important tool in the development of ITS involving the detection and categorisation of road vehicles. Such monitoring can support the assessment of a range of needs: traffic volume and speed estimation, flow and congestion control, incident detection, usage type, queue lengths, illegal manoeuvres. Road ITS employs a range of disparate sensors for estimating traffic parameters, including inductive loops, buried cables, surface cables, microwaves and lasers. In contrast, vision-based video monitoring systems offer the capability to report on a rich set of behaviours such as lane changing and illegal stopping and turning, as well as the simple counting and categorisation of vehicle types. In addition, they are low cost, less disruptive, low maintenance, non-contact, high reliability devices. However, to operate robustly and reliably in unconstrained and complex environments demands sophisticated image analysis tools capable of adapting to a wide range of conditions associated with varying weather and illumination conditions, traffic density etc.

Visual object recognition aims to classify observed objects into semantically meaningful categories. This chapter focuses on vehicle classification in a mid-field video surveillance framework using a single static camera. Several scenarios motivate the work. ITS is increasingly used for the management of road transport systems, where the transport infrastructure is stressed by steadily increasing traffic flow levels, leading to poor network performance. The resulting high levels of congestion and delay cause increasing frustration for drivers and also carry a high economic and environmental cost.

Vehicle classification is important for determining the proportion of vehicle classes in ITS. This type of information is traditionally collected by human operators in a manual survey of road usage that may be performed periodically. This approach provides a limited snapshot of traffic distributions and will have a diminishing accuracy over time. An automated system offers a more accurate, lower cost solution that can provide continuous, real-time output. Compared with object recognition from still images, analysis of video sequences simplifies the recognition task, because moving objects can be easily separated from a static background using background modeling and subtraction, so problems of clutter can be minimized. Despite the large amount of literature on vehicle detection and tracking, there has been comparatively little work on vehicle classification. This may be because vehicle classification is an inherently hard problem, but also because detection and tracking are usually necessary precursors to vehicle classification.

In general, vehicle recognition must cope with a number of complexities that complicate the task: vehicles are generally textureless and of limited size and quality in an image with a wide field of view. They exhibit a wide variety of shape and size (even within a single category), are subject to partial visibility associated with occlusion and changing pose and are imaged under varying lighting and weather conditions, confused by shadows and camera noise. The requirement to distinguish sub-classes such as minivan vs. car vs. taxi complicates the task, though in urban ITS it is more common to use broader categories of road user such as car, van, bus and motorcycle.

By adopting a manual approach to segmentation, the aim is to discover the potential of using simple low level features to achieve high levels of classification performance in the absence of image segmentation noise. This chapter assesses the performance of different

classification methodologies to categorize vehicles for an automatic traffic survey process using data taken from a pole-mounted roadside CCTV camera in an urban environment. Initially, vehicles are manually detected within a constrained region of the image and at their closest distance to a (virtual) fiducial marker. The aim is to find the best classification method and the best features to classify vehicles. A set of size and shape features are extracted from the binary silhouettes of manually segmented vehicle outlines. The results of a supervised classification experiment compares the performance of two popular classifiers, support vector machines (SVM) and random forests (RF), to assess their ability to categorize the vehicle data into four broad groups: car, van, bus and motorcycle/bicycle. The silhouettes are also matched to 3D wireframe models of proto-typical vehicle shapes projected into the image space and compared with the performance of the feature-based methods.

The reminder of the chapter is organised as follows. The next section presents the related work. Model based classification method and camera calibration approach are given in Section 5.3. SVM and random forests methods are compared using measurement-based feature and intensity-based feature to find the best classification method is presented in Section 5.4. Quantitative evaluation of the classification methods is given in Section 5.5. Finally, Section 5.6 summarises the chapter.

5.2. Related research

Image analysis for vehicle classification can be generally categorised into three principal approaches: model-based, feature based and measurement-based. ***i) Model-based classification.*** Early research was undertaken by Tan et al. (1998) and Sullivan et al. (1997) who investigated tracking and classification of moving vehicles using 3D wireframe models. They utilized camera calibration and the ground-plane constraint to match image edge features to a set of wireframe models for both tracking and classification. More recently Nieto et al. (2011) have used 3D models to detect and classify vehicles by integration of temporal information and model priors within a Markov Chain Monte Carlo method. Messelodi et al. (2005) also used 3D models and low level image features, whilst Buch et al. (2009, 2010, 2011) applied a 3D model and 3DHOG to detect and classify vehicles. Gupte et al. (2002) modeled vehicles as rectangular patches with certain dynamic behaviour to detect and classify vehicles. They classified vehicles into only two categories, car and non-car, from side-on snapshots of vehicles on a highway. They report a 70% classification accuracy using only measures of width and height of the binary silhouette. One advantage of these methods is that they are not learning-based and hence do not require a training stage, but do rely on a calibrated camera. ***ii) Feature-based classification.*** Hasegawa and Kanade (2005) described a vision system to recognize moving targets such as vehicle type and pedestrians on a public street using an 11-dimensional vector of image features. They used features of the object bounding box, including, the width, height and area; first, second and third image moments and the centroid coordinate. They reported an overall classification accuracy of 91% for 6 object categories using linear discriminant analysis, but acknowledge poorer performance under illumination variation. Ma and Grimson (2005) used edge-based features and modified scale invariant feature transform (SIFT) descriptors. They were able to distinguish objects at a more detailed level, discriminating between vans, minivans, sedans and taxis. An appearance learning-based method is presented by Zhang and Avery et al. (2007) that can distinguish between moving objects such as cars, vans, trucks, people and bikes using multi-block local binary patterns. Ji et al. (2007) take vehicle side-on views and use Gabor features and a minimum distance classifier to classify vehicles into five categories, with recognition

rates of up to 95.17%. Morris and Trivedi (2006a; 2006b; 2008) presented a tracking system with the ability to classify vehicles into three classes. They construct a 10-feature measurement vector and then apply either principal component analysis (PCA) or linear discriminantanalysis (LDA) to manage the size of the data, followed by a weighted K-nearest neighbour (KNN) classifier, fuzzy C-means clustering and a weighted k-nearest neighbour (wkNN) classifier. Results indicated an accuracy of up to 88%, though low confidence objects were rejected. Eigenvehicle and PCA-SVM were proposed to classify vehicles into four class types: trucks, passenger car, van and truck in (Zhang et al., 2006; Thi et al., 2008).
iii) Measurement-based classification. Zhang and Li et al. (2007) and Avery (2004) proposed a length-based vehicle classification ITS using un-calibrated video cameras. Lipton et al. (1998) classify moving targets into three categories: human, vehicle and background clutter simply using dispersedness. The vehicle was classified into only two categories: cars and noncars. Chen et al. (2009b) classify road vehicles type (car, van and HGV) and colour using support vector machines. The feature vector to describe the vehicle silhouette encodes size, aspect ratio, width, solidity and 3D colour histograms.

5.3. Model-based vehicle classification

The main idea of model-based classification (MBC) method is to employ 3D wireframe models. After camera calibration, the ground plane constraint and the 3D wireframe model projections are used to match detected vehicle silhouette. However, for a road CCTV camera, normally only one view image is provided. One view image is not enough for camera calibration, so calibrating the camera is a problem. To address this problem, a novel approach to camera calibration utilizing images mapped by Google Earth is presented in this section.

5.3.1. Model fitting

Model-based vehicle classification is performed using simple 3D wireframe models to represent proto-typical vehicle types if the road side CCTV camera is calibrated. The extracted vehicle silhouettes are matched against 2D projections of the 3D models. A measure of the quality of fit (Q) between the object blob B_{obj} and model mask M_i is defined by the normalised overlap area:

$$R_i = \frac{B_{obj} \cap M_i}{B_{obj} \cup M_i}; \qquad Q = \max\ (R_i), i = 1..4 \tag{5.1}$$

This is the proportion of pixels in the area of the intersection to the pixels in the area of union of both masks and will be in the range is 0 to 1. Why using of $B_{obj} \cup M_i$ instead of using M_i in the denominator in the formula (5.1)? That is because M_i not related to the detected object blob. For example, if there is a bus, all the 2D projection of small vehicles such as van, car and motorcycle will be inside the detected silhouette. So that $B_{obj} \cap M_i = M_i$, $R_i = 1$. All the Q values of projections from small vehicles are equal to 1. There are no differences.

In order to cope with noise and make the classification method robust, a grid search is made along the x and y axes (around the centroid of the blob) to determine the location of the optimum match. The search range is related to the size of the blob ([-25, 25] in both x and y direction in the experiments). The blob is classified as the category with the largest degree of fit. Figure 5.1 shows the 3D models with true size (Buch et al., 2010). Figure 5.2 shows

examples of the 2D projections of 4 category models matched to an equivalent vehicle silhouette. Figure 5.3 shows the procedure of model projection matching. (a) Shows the original image with the silhouette of detected vehicle. (b) Shows the initial position of the projection from the best wireframe vehicle model. (c) Shows the final position of the projection of the best wireframe model, Q=0.8656, it is corresponding to car wireframe model. (d) The 3D surface of the variation of the matching value measurement Q for the best matching wireframe model.

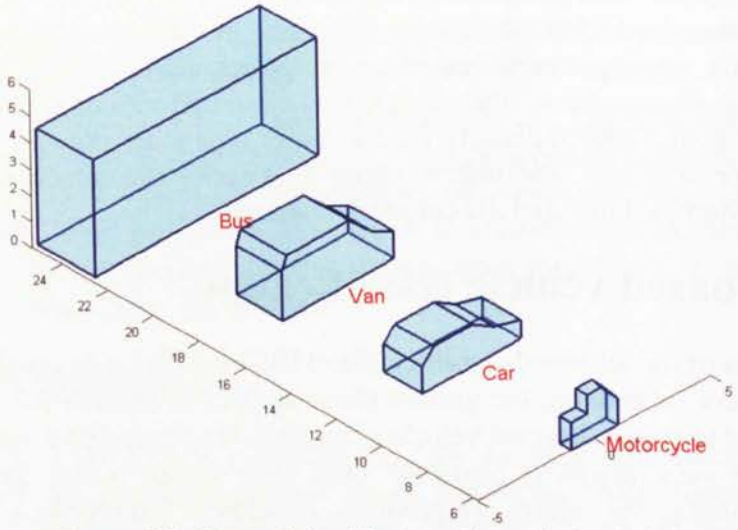


Figure 5.1. 3D models with true size scale in metres.

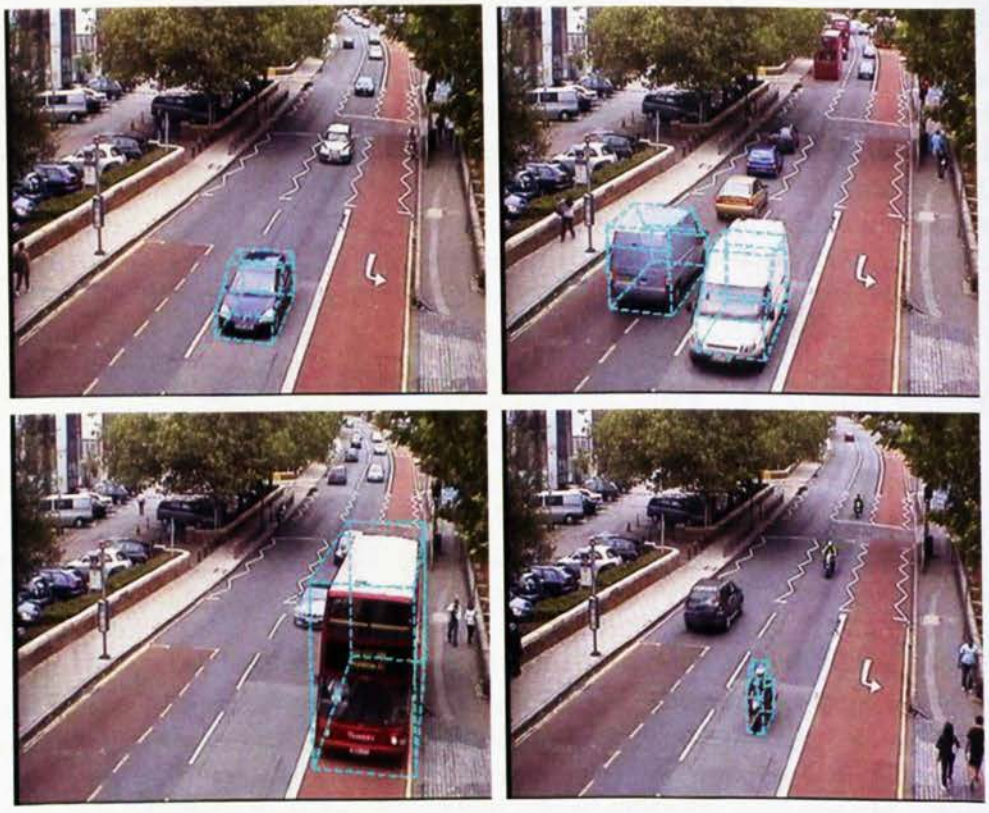


Figure 5.2. 3D models (Car, Van, Bus, Motorcycle) projected onto the ground plane.

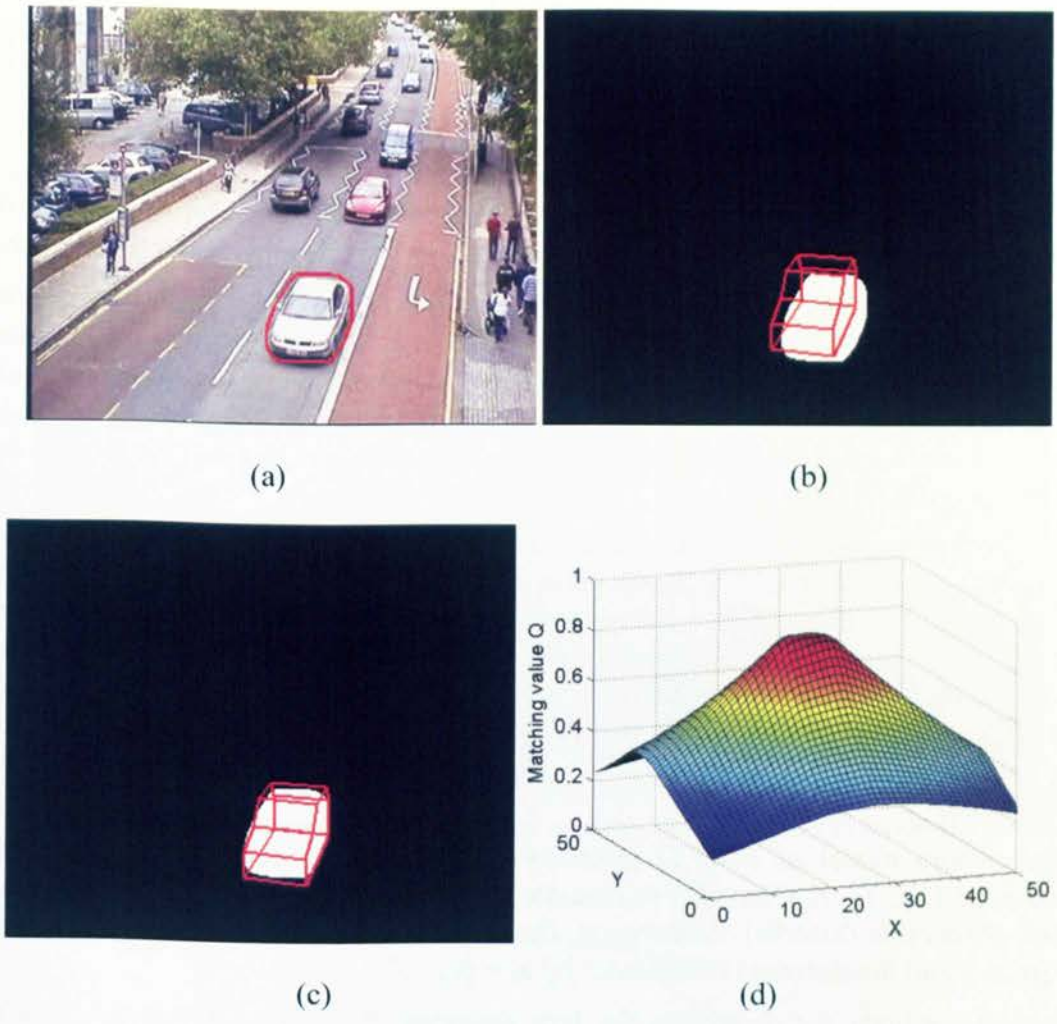


Figure 5.3. The procedure of model projection matching. (a) Original image with the silhouette of detected vehicle. (b) The initial position of the projection from the best wireframe vehicle model. (c) The final position of the projection of the best wireframe model. (d) The 3D surface of the variation of the matching value measurement of Q for the best matching wireframe model.

5.3.2. Camera calibration

Camera calibration was used to recover the projective transformation between 2D image points (u, v) and the corresponding 3D world points (X_w, Y_w, Z_w) on the ground plane. Four coordinate systems need to be determined in order to compute the transformation: world, camera, camera sensor plane and image plane coordinate systems respectively. Referring to the pinhole camera model, a camera matrix is used to denote a projective mapping from world coordinates to pixels coordinates.

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A}\left[\mathbf{R}|\mathbf{T}\right] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{5.2}$$

where the intrinsic parameters are

$$\mathbf{A}=\begin{bmatrix} f\alpha_x & \gamma & u_0 \\ 0 & f\alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

(X_c, Y_c, Z_c) represents the coordinates of the same visible point (X_w, Y_w, Z_w) in a camera-centred coordinate system, α_x and α_y are scale vectors (pixels/mm) related to the spatial quantization, f the focus length, u_0 and v_0 are the coordinates of the principle point and γ represents the skew coefficient between the x and y axis, and typically takes a value near 0. The 3×3 rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$ are the extrinsic parameters which denote the coordinate system transformations from 3D world coordinates to 3D camera coordinates. Equivalently, the extrinsic parameters define the position and orientation of the camera in world coordinates.

Many implementations of camera calibration methods are freely available on the internet. For the roadside CCTV camera, there is only one view. Without the opportunity to acquire camera calibration data using an appropriate target, the Google Earth Map is used to provide a plan view of the scene onto which the road side camera is facing. The centre of the world coordinates is set at the top-left corner of the plan view image. The scale at ground level is approximately 20 pixels per meter. Heikkilä's (2000) method has been used for camera calibration and the lens distortion has been corrected using the normal radial distortion model.

The camera model of Eq.(5.1) produces the ideal image coordinates $[u, v]^T$ of the projected point (X_w, Y_w, Z_w). In order to separate these errorless but unobservable coordinates from their observable distorted counterparts, the corrected coordinates of Eq.(5.1) is denoted by $\mathbf{a}_c = [u_c, v_c]^T$ and the distorted coordinates by $\mathbf{a}_d = [u_d, v_d]^T$.

Several methods for correcting the lens distortion have been developed. The most commonly used approach is to decompose the distortion into radial and decentring components (Slama, 1980). Given the distorted image coordinates a_d, the corrected coordinates a_c are approximated by

$$a_c = a_d + \Re(a_d, \delta) \tag{5.4}$$

where

$$\Re(a_d,\delta)=\begin{bmatrix} \bar{u}_d\left(k_1 r_d^2 + k_2 r_d^4 + \cdots\right) + \left(2p_1\bar{u}_d\bar{v}_d + p_2\left(r_d^2 + 2\bar{u}_d^2\right)\right)\left(1 + p_3 r_d^2 + \ldots\right) \\ \bar{v}_d\left(k_1 r_d^2 + k_2 r_d^4 + \cdots\right) + \left(p_1\left(r_d^2 + 2\bar{v}_d^2\right) + 2p_2\bar{u}_d\bar{v}_d\right)\left(1 + p_3 r_d^2 + \ldots\right) \end{bmatrix}$$

$\bar{u}_d = u_d - u_0$, $\bar{v}_d = v_d - v_0$, $r_d = \sqrt{\bar{u}_d^2 + \bar{v}_d^2}$, and $\delta = [k_1, k_2, \cdots, p_1, p_2, \cdots]^T$. The parameters k_1, k_2, ... are the coefficients for the radial distortion that an actual image point may be displaced radially in the image plane, and the parameters p_1, p_2, ... are the coefficients for the decentring or tangential distortion which may occur when the centers of curvature of the lens surfaces of the camera optics are not strictly collinear.

Figure 5.4 shows the calibration reference image (left) and the corresponding plan view image (right) from Google Earth. The cyan circles indicate the corresponding location and index number. The blue asterisks indicate re-projected coordinates from the 3D world coordinate after camera calibration, demonstrating the accuracy of the calibration result

(average re-projection error of 0.97 pixels). 3D reconstruction of corresponding points are given in Figure 5.5. Figure 5.6 shows the retrieved parameters of the camera model, the pose of the camera and road plan and its orientation in the camera coordinate frame. The car wireframe model (blue outline) is projected into the image to illustrate the correct scale and pose onto the road. Figure 5.7 shows a 3D wireframe model projected onto the image plane. It shows the pose and scale of the model are correct.

Figure 5.4. Calibration reference image (left) and plan view image (right). Cyan circles and index number indicate the corresponding points and the blue asterisks indicate the re-projected points.

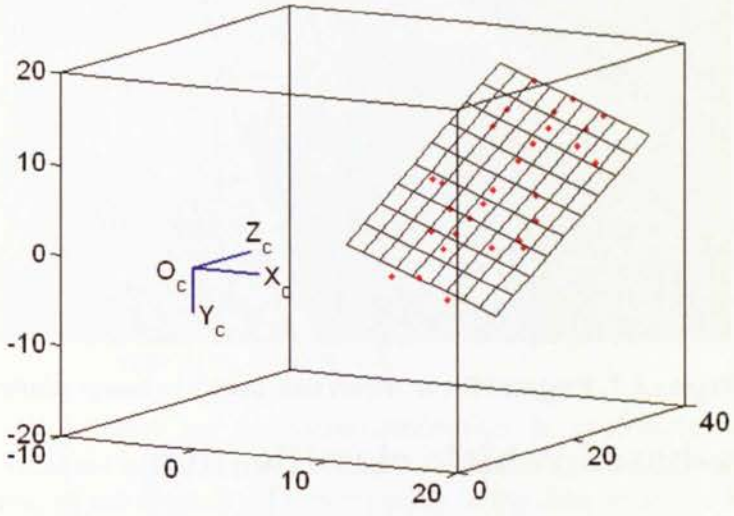


Figure 5.5. 3D reconstruction of corresponding points.

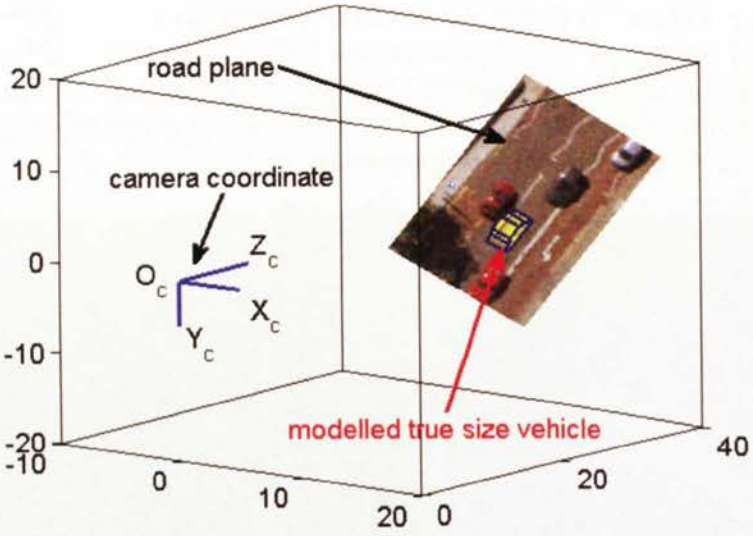


Figure 5.6. Retrieved camera model.

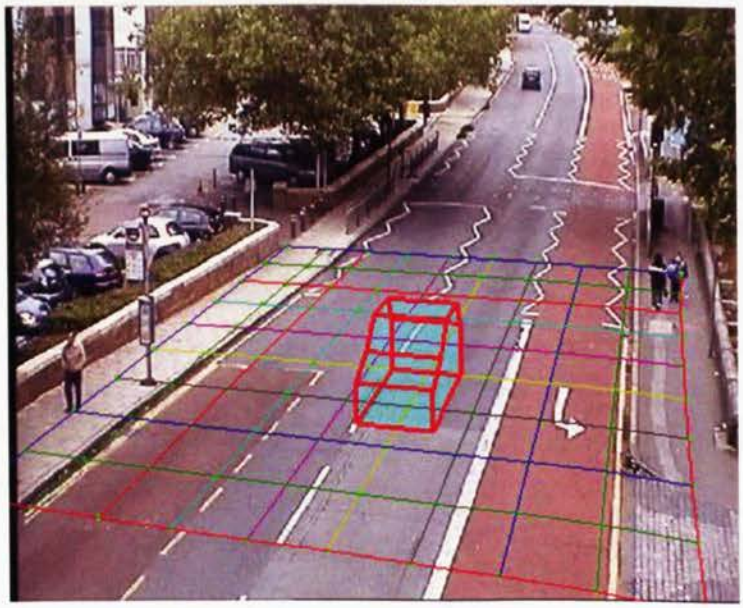


Figure.5.7. Project 3D car wireframe model to image plane.

5.4. Feature-based vehicle classification

5.4.1. Classifiers review

5.4.1.1. Support Vector Machines

The support vector algorithm is a nonlinear generalization of the generalized portrait algorithm developed in Russia in the sixties (Vapnik et al., 1963, 1964). However, a similar approach using linear instead of quadratic programming was taken at the same time in the US, mainly by Mangasarian (1965, 1968 and 1969). As such, it is firmly grounded in the framework of statistical learning theory, which has been developed over the last three decades by Vapnik (1982, 1995) himself and Cortes et al. (1995). In its present form, the

support vector machine (SVM) was largely developed at AT&T Bell Laboratories by Vapnik and co-workers. SVM have been recently proposed as a very effective method for general purpose classification and pattern recognition. Intuitively, given a set of points which belong to either of two classes, a SVM finds the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. According to the papers (Vapnik, 1995 and 1999), this hyperplane minimizes the risk of misclassifying examples of the test set.

Prior related work includes that of Baek et al. (2007), who presented a vehicle colour classification based on the SVM. The implementation results showed a 94.92% of success rate for 500 vehicles with 5 colours. Ambardekar et al. (2008) used optical flow and knowledge of camera parameters to detect the pose of a vehicle in the 3D world. This information is used in a model-based vehicle detection and classification technique employed by their traffic surveillance application. Ma et al. (2005) proposed an approach to vehicle classification under a mid-field surveillance framework. They discriminate features based on edge points and modified SIFT descriptors.

5.4.1.1.1. Two class SVM

This section gives a brief review of the SVM and refers the reader to (Cortes et al., 1995; Vapnik, 1999) for further details. Assume a set S which has N training samples of points $x_i \in \mathbf{R}^m$ with i=1,2,..., N. Each point x_i belong to either of two classes and thus is given a label $y_i \in \{-1,1\}$. The goal is to establish the equation of a hyperplane that divides S leaving all the points of the same class on the same side while maximizing the distance between the two classes and the hyperplane (Ma et al., 2005).

A hyperplane in the feature space can be described as the equation

$$\langle w, x \rangle + b = 0 \tag{5.5}$$

where $w \in \mathbf{R}^m$ and b is a scalar. When the training samples are *linearly separable*, SVM yields the optimal hyperplane that separates two classes with no training errors, and maximises the minimum distance from the training samples to the hyperplane. There is some redundancy in Eq. (5.5), and without loss of generality it is appropriate to consider a canonical hyperplane (Vapnik, 1995), where the parameters w, b are constrained by

$$\min_i \left| \langle w, x_i \rangle + b = 1 \right| \tag{5.6}$$

This incisive constraint on the parameterisation is preferable to alternatives in simplifying the formulation of the problem. It states that the norm of the weight vector should be equal to the inverse of the distance of nearest point in the data set to the hyperplane.

A separating hyperplane in the canonical form must satisfy the following constraints:

$$y_i \left[\langle w, x_i \rangle + b \right] \geq 1, \; i = 1,2,\cdots,N \tag{5.7}$$

The optimal hyperplane is given by maximizing the margin ρ, subject to the constraints of Eq. (5.7). The margin is given by:

$$\rho(w,b) = \frac{1}{\|w\|} \left(\min_{x_i, y_i = 1} \left| \langle w, x_i \rangle + b \right| + \min_{x_i, y_i = -1} \left| \langle w, x_i \rangle + b \right| \right) = \frac{2}{\|w\|} \tag{5.8}$$

Since $\|w\|^2$ is convex, minimizing it under linear constraints (5.7) can be achieved with Lagrange multipliers. If the N non negative Lagrange multipliers associates with constraints (5.7) denoted by $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_N)$, the solution to the optimisation problem of Eq.(5.8) under the constraint Eq.(5.7) is given by the saddle point of the Lagrange function (lagrangian)

$$\phi(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i \left(y_i \left[\langle w, x_i \rangle + b \right] - 1 \right) \tag{5.9}$$

Eq. (5.9) can be transformed to its dual problem, which is easier to solve. The dual problem is given,

$$\max_{\alpha} w(\alpha) = \max_{\alpha} \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \right) \tag{5.10}$$

Subject to,

$$\alpha_i \geq 0, \; i = 1,2,\cdots,N,$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

This can be achieved by the use of a standard quadratic programming method (Vanderbei, 1999). Once the vector $\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_N^*)$ solution of the maximization problem (5.10) has been found, the optimal separating hyperplane is given by,

$$w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i \tag{5.11}$$

$$b^* = -\frac{1}{2} \langle w^*, x_r + x_s \rangle \tag{5.12}$$

where x_r and x_s are any support vector from each class satisfying, $\alpha_r, \alpha_s > 0$ and $y_r = -1$, $y_s = 1$.

The hard classifier is then

$$f(x) = sign\left(\langle w^*, x \rangle + b^* \right) \tag{5.13}$$

So far the discussion has been restricted to the case where the training data is linearly separable. However, in general this will not be the case. When the data is not linearly separable, there are two approaches to generalising the problem, which are dependent upon prior knowledge of the problem and an estimate of the noise on the data. In the case where it is expected (or possible even known) that a hyperplane can correctly separate the data, a method of introducing an additional cost function associated with misclassification is appropriate. More generally, a slack variable $\xi = (\xi_1, \xi_2, \cdots, \xi_N)$, with $\xi_i \geq 0$ was introduced in (Cortes et al., 1995), such that

$$y_i \left[\langle w, x_i \rangle + b \right] \geq 1 - \xi_i, \; i = 1,2,\cdots,N \tag{5.14}$$

to allow the possibility of examples that violate (5.7). The ξ_i are a measure of the misclassification errors. The generalised optimal separating hyperplane is determined by the vector w that minimise the function

$$\phi(w,\xi)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i \tag{5.15}$$

The purpose of the first term is minimized to control the learning capacity as in the separable case; the second term controls the number of misclassified points. The parameter C is chosen by the user, a larger C corresponding to assigning a higher penalty to errors.

The only difference is that α_i have upper bound C here. SVM training requires that C, the penalty term for misclassifications, are fixed. So C must be chosen to reflect the knowledge of noise in the data.

For applications where linear SVM does not produce satisfactory performance, nonlinear SVM is suggested. The basic idea is to map x by nonlinearly mapping $\phi(x)$ to a much higher dimensional feature space, and by working with linear classification in that space in which the optimal hyperplane is found. The nonlinear mapping can be implicitly defined by introducing the so called kernel function $K(x_i,x_j)$ which computes the inner product of vectors $\phi(x_i)$ and $\phi(x_j)$. If $K(x_i,x_j)=\phi(x_i)\cdot\phi(x_j)$, then only K is needed in the training algorithm and the mapping ϕ is never explicitly used.

From Mercer's theory (Osuna et al., 1997; Girosi, 1997), for a given symmetric positive kernel $K(x,y)$, there exists a mapping ϕ such that $K(x,y)=\phi(x)\phi(y)$, as long as it satisfies Mercer's condition.

Under feature mapping, the solution of a SVM has the form:

$$f(x)=sign\left(\langle\hat{w},\phi(x)\rangle+\hat{b}\right) \tag{5.16}$$

where

$$\hat{w}=\sum_{i=1}^{N}\alpha_i^* y_i\phi(x_i)\quad \hat{b}=-\frac{1}{2}\langle\hat{w},\phi(x_r)+\phi(x_s)\rangle \tag{5.17}$$

A convenient formula to determine the sign is

$$\begin{aligned} f(x)&=sign\left(\left\langle\sum_{i=1}^{N}\alpha_i^* y_i\phi(x_i),\phi(x)\right\rangle+\hat{b}\right)\\ &=sign\left(\sum_{i=1}^{N}\alpha_i^* y_i K(x_i,x)+\hat{b}\right)\end{aligned} \tag{5.18}$$

where $\hat{b}=-\frac{1}{2}\sum_{i=1}^{N}\alpha_i^* y_i[K(x_i,x_r)+K(x_i,x_s)]$.

Though new kernels are being proposed by researches, the following four basic kernels are normally used in SVM books (Chang and Lin, 2001):

- Liner: $K(x_i,x_j)=x_i^T x_j$
- Polynomial: $K(x_i,x_j)=(\gamma x_i^T x_j+r)^d,\gamma>0$

- Gaussian radial basis function (RBF): $K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0$
- Sigmoid: $K(x_i, x_j) = \tanh\left(\gamma x_i^T x_j + r\right)$

here γ, r and d are kernel parameters.

5.4.1.1.2. Multi- class SVM

The solution of binary classification problems using SVMs is well developed. Multi-class problems such as object recognition and image classification (Chapelle et al., 1999) have typically been solved by combining independently produced binary classifiers. In the one-vs-all (OVA, or one-vs-rest) method, one constructs k classifiers, one for each class. The mth classifier constructs a hyperplane between class m and the k-1 other classes. If say the classes of interest in an image include car, HGV and van, classification would be effected by classifying car against non-car (i.e. HGV and van) or HGV against non-HGV (i.e. car and van). This method has been used widely in the support vector literature to solve multi-class pattern recognition problems (Blanz et al., 1996; Scholkopf et al., 1995; Melgani et al., 2004). Alternatively, the one-vs-one (OVO, or all-vs-all) approach involves constructing a machine for each pair of classes resulting in $k(k-1)/2$ machines. For each distinct pairs m_1 and m_2, the learning algorithm runs on a binary problem in which examples labeled y= m_1 are considered positive, and those labeled y= m_2 are negative. All other examples are simply ignored. When applied to a test point, each classification gives one vote to the winning class and the point is labeled with the class having most votes. This approach can be further modified to give weighting to the voting process. Rifkin and Klautau (2004) gave an extensive theoretical-based analysis, comparing multiple classifications, such as OVA, OVO, RLSC (regularized least squares classification), COM (complete code) and ECOC (error correcting output coding). They indicated that the OVA scheme is extremely powerful, producing results that are often at least as accurate as other approaches. Anthony et al. (2007) gave a similar conclusion that the resulting classification accuracy of OVA is not significantly different from the OVO approach. However, some researchers have a different opinion. They reported that the OVO scheme has a simple conceptual justification, and can be implemented to train faster and demonstrated better performance than the OVA scheme although OVO trains $O(k^2)$ classifiers rather than $O(k)$ for the OVA scheme (the reason is the individual classifiers are much smaller, and given the time required to train is generally superlinear), and the OVO scheme offers better performance than the OVA scheme (Allwein et al., 2000; Furnranz, 2002; Hsu and Lin et al., 2002).

5.4.1.2. Random forests

Random forests (RF) is an ensemble classifier that consists of many decision trees. The assumption is that the user knows about the construction of single classification trees. RF grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest) (Breiman, 2001). Random forests does not overfit. You can run as many trees as you want.

Each tree is grown as follows:

1. If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.
2. If there are M input variables, a number $m<<M$ is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
3. Each tree is grown to the largest extent possible. There is no pruning.

Each tree has three major tasks:

(i) How to partition the data at each step?
(ii) When to stop partitioning?
(iii) How to predict the value of y for each x in a partition?

When the training set for the current tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample. This oob (out-of-bag) data is used to get a running unbiased estimate of the classification error as trees are added to the forest. It is also used to get estimates of variable importance. In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally, during the run, as follows: Each tree is constructed using a different bootstrap sample from the original data. About one-third of the cases are left out of the bootstrap sample and not used in the construction of the kth tree. Put each case left out in the construction of the kth tree down the kth tree to get a classification. In this way, a test set classification is obtained for each case in about one-third of the trees. At the end of the run, take j to be the class that got most of the votes every time case n was oob. The proportion of times that j is not equal to the true class of n averaged over all cases is the oob error estimate. This has proven to be unbiased in many tests.

If the number of variables is very large, forests can be run once with all the variables, then run again using only the most important variables from the first run.

The basic idea for prediction tree is very simple. To predict a response or class Y from inputs $X_1, X_2, \ldots, X_p$ by growing a binary tree. At each internal node in the tree, a test to one of the inputs, say X_i is applied. Depending on the outcome of the test to go to either the left or the right sub-branch of the tree, eventually come to a leaf node, where a prediction is made. This prediction aggregates or averages all the training data points which reach that leaf.

In the experiments of vehicle classification, for a set of N training samples of M-dimensional feature vectors associated with one of L classes, a RF classifier constructs a set of tree predictors. Each tree is built using a standard classification and regression tree (CART) algorithm with a bootstrap sample of the training data - a set of N samples chosen randomly, with replacement. The optimal split at each tree node is assessed using a random subset of $m < M$ dimensions. The trees are constructed until a leaf is reached that contains samples from only one class and hence do not need to be pruned. RF is particularly suited to learning non-linear relationships in high-dimensional class training data. They benefit from rapid training, resistance to overtraining, no critical parameters to select, and can provide a near state-of-the-art performance.

When an unseen feature vector is presented for independent classification by each tree in the forest, each tree casts a unit class vote, and the most popular class is assigned to the input vector. Alternatively, the proportion of votes assigned to each class can be used to provide a probabilistic labeling of the input vector. More details about random forests please reference the RF toolbox provided by Breiman and Cutler (Breiman, 2001; Breiman and

Cutler).

5.4.2. Feature extraction

5.4.2.1. Measurement based feature

A set of measurement-based features (MBF) is used to classify vehicles, because the MBF is cheap to compute and store to build a vehicle feature database. 13 different measurements make up the classification vector. The components comprise measures of size and shape from the binary silhouette and encompassing bounding box (width, height, and area), circularity (dispersedness, equivdiameter), ellipticity (length of major and minor axis, eccentricity), and shape-filling measure (filled area, convex area, extent, solidity).

The separability of the four classes can be observed by rank ordering the feature vector. Each feature is independently used to classify the dataset using SVM. The features are ranked as follows in decreasing order of the resulting true positive classification rate (TPR):

1- perimeter
2- length of minor axis of best-fitting ellipse
3- width of the bounding box
4- area
5- height of the bounding box
6- length of major axis of best-fitting ellipse
7- filled area (the pixels in the region with holes filled in)
8- dispersedness = l^2/A, A is the area, and l is the perimeter of the object.
9- convex area
10- equivdiameter = $\sqrt{4A/\pi}$
11- extent (proportion of pixels in the bounding box of object)
12- eccentricity
13- solidity (proportion of pixels in the convex hull of object)

Figure 5.8 plots the distribution of the three highest-ranked features over the dataset. The plot illustrates that the vehicle categories are not simple. For instance, the two green clusters are associated with two different types of buses – single and double decker buses. The van category also exhibits a range of feature measures that are not clearly separable. The motorcycle/bicycle category is unambiguously distinguishable. Some samples of vehicles are shown in Figure A.5 in the Appendix.

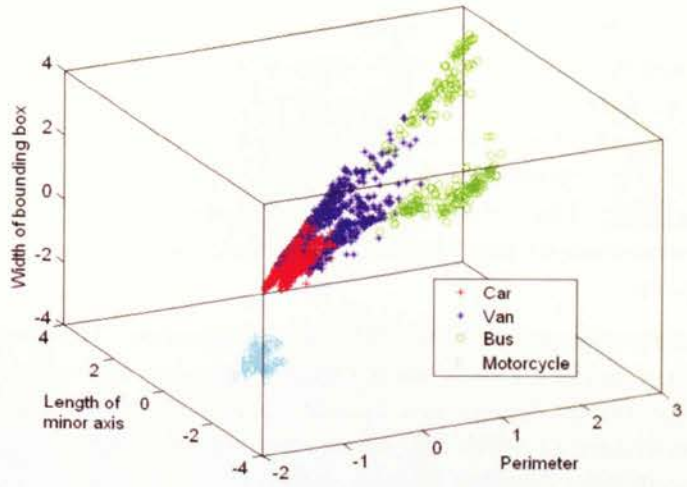


Figure 5.8. The first three features of the labeled vehicle silhouettes.

5.4.2.2. Image intensity-based features

To improve the accuracy of classification, a potentially effective feature based on a pyramid histogram of gradient orientations (PHOG) is investigated. PHOG was first proposed by Bosch et al. (2007b) and has been successfully applied to object recognition, human expression recognition and image classification (Bosch et al., 2007a; Bai et al., 2009; Han et al., 2010). As a spatial shape descriptor, it can represent the statistical information of global shape and local shape (in a sub-region), which is effective for object recognition. The local shape is captured by the distribution over edge orientations within a region, and the spatial layout by tiling the image into regions at multiple resolutions. The descriptor consists of a histogram of orientation gradients over each image sub-region at each resolution level of the detection bounding box. The gradient of an image is given by the formula:

$$\nabla f = \left(\frac{\partial f}{\partial x}\right)\hat{x} + \left(\frac{\partial f}{\partial y}\right)\hat{y} \tag{5.19}$$

where $(\partial f/\partial x)$ and $(\partial f/\partial y)$ are the gradient in the x and y direction, respectively.

For the orientation range from 0° to 180°, the gradient orientation can be calculated by the formula:

$$\theta = atan\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) \tag{5.20}$$

$atan(\cdot)$ is in the range $[-\pi/2, \pi/2]$.

For the orientation range from 0° to 360°, the gradient orientation can be calculated by the formula:

$$\theta = atan2\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) \tag{5.21}$$

$atan2(\cdot)$ is in the range $[-\pi, \pi]$.

For vehicle classification, set 3 levels and 9 orientation bins, evenly spaced over 0° - 360°, extract the PHOG features and normalise them at each level. The number of levels and orientations are the optimal numbers for the experiments. The dimension of the resulting vector is constructed by concatenating these into a $9+4\times9+4\times4\times9=189$ element vector. For local shape representation there are two kinds of PHOG available. One is the edge-based PHOG (EPHOG), which is represented by a histogram of edge orientations within an image region and its sub-region. The other is the intensity-based PHOG (IPHOG), which is represented by the distribution of local intensity gradients, without precise knowledge of the corresponding edge point.

Figure 5.9 illustrates the procedure of IPHOG construction. The descriptor consists of a histogram of orientation gradients over each image subregion at each resolution level. The final IPHOG descriptor for the image is a concatenation of all the IPHOG vectors. Figure 5.10 illustrates the procedure of EPHOG construction for the same image. The edge of the object detected by Canny edge detector (Canny, 1986).

Figure 5.11 shows an input image and the shape spatial pyramid representation of IPHOG from level 0 to level 2 of each category of bus, van, car and motorcycle. It also shows that the IPHOGs of different types of vehicle are different. Figure 5.12 and Figure 5.13 show the IPHOG of two cars and vans from level 0 to level 2. A Euclidean distance $D_2(A,B)$ (L_2-normal) between two histograms was used to measure the similarity of them (Cha and Srihari, 2002).

$$D_2(A,B)=\sqrt{\sum_{i=1}^{b}\left(H_i(A)-H_i(B)\right)^2} \tag{5.22}$$

where $H_i(A)$ and $H_i(B)$ are two d-dimensional histograms. The distances between histograms are given in Table 5.1. The table shows that the distance between the histogram of cars is 0.0358, the distance between the histogram of vans is 0.1917. But the average distance between the histogram of cars and vans is 0.3201. The table illustrates that the same type of vehicles has similar IPHOG, the distance between their histograms is smaller, thought they have different colour and shape. The distance between histograms of different type is bigger. Figure 5.14 plots the first three most important PCA (principal component analysis) components of a combined MBF+IPHOG feature vector for the manually-labeled data, where one class (motorcycle) is unambiguously separable.

Table 5.1. Euclidean distance between the histogram of cars and vans.

	Car1	Car2	Van1	Van2
Car1	0	0.0358	0.3316	0.3116
Car2	0.0358	0	0.3268	0.3104
Van1	0.3316	0.3268	0	0.1917
Van2	0.3116	0.3104	0.1917	0

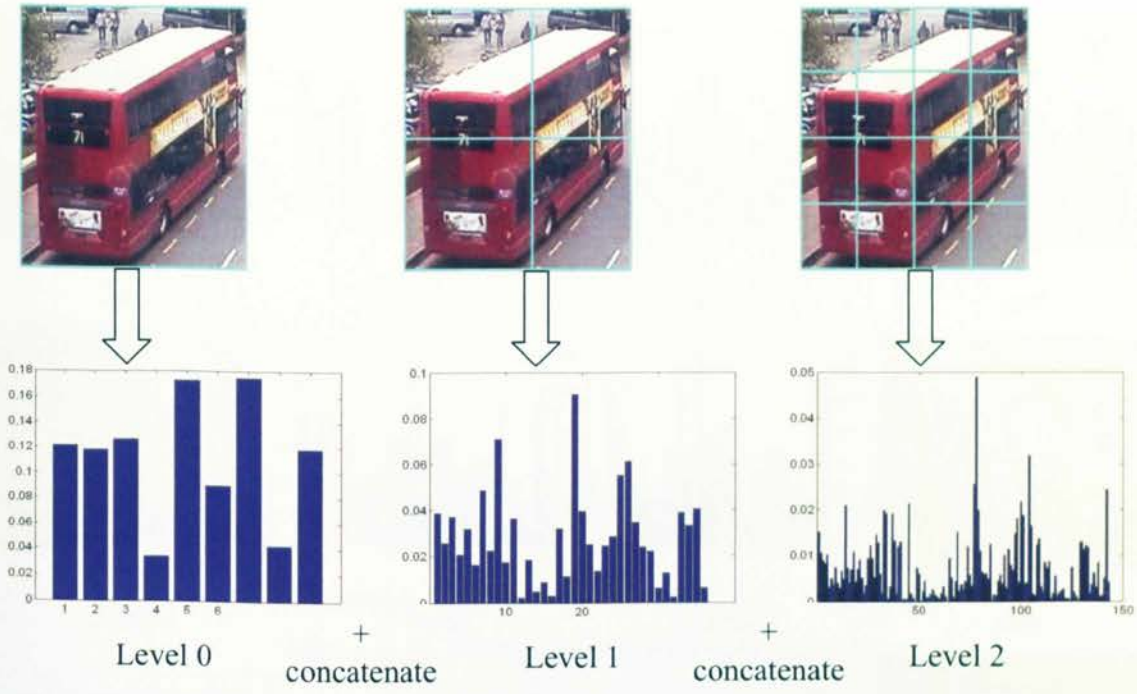


Figure 5.9. The procedure of IPHOG construction.

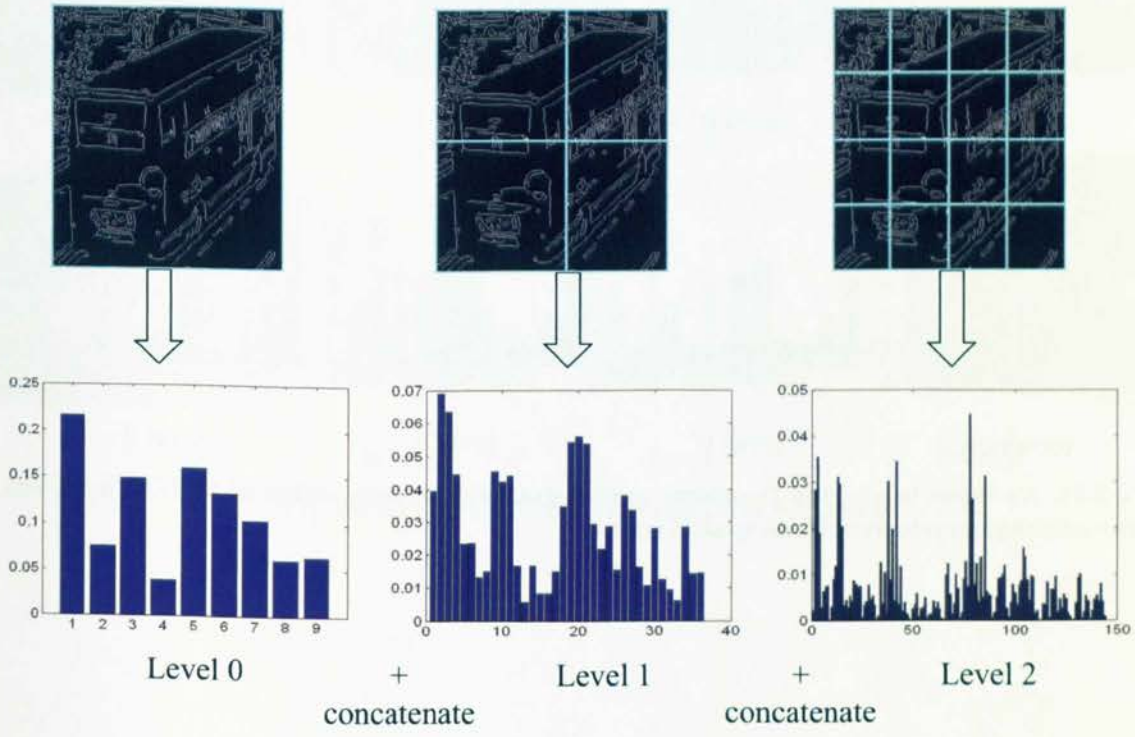


Figure 5.10. The procedure of EPHOG construction.

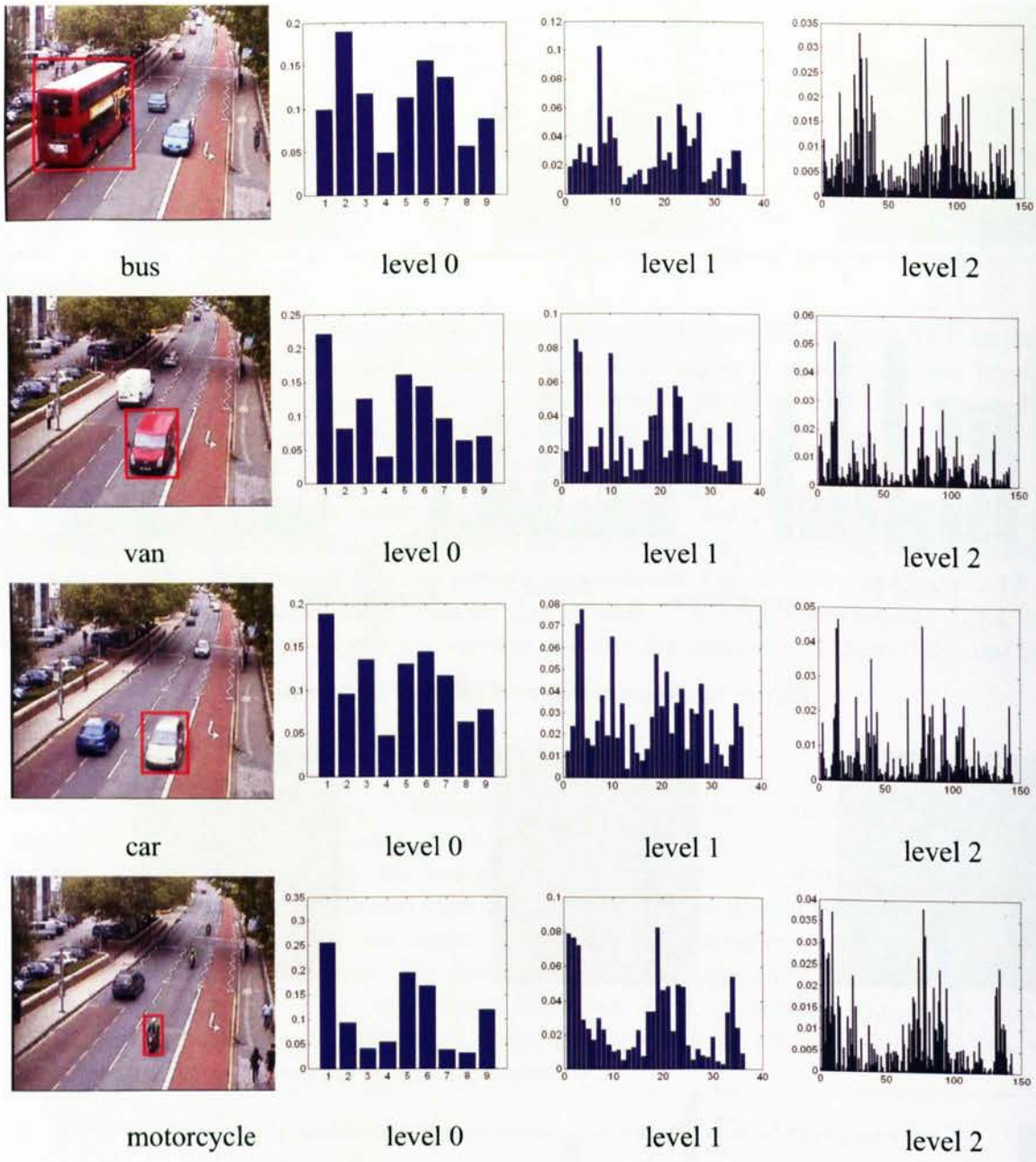


Figure 5.11. An input image and the shape spatial pyramid representation of IPHOG for a bus, van, car and motorcycle over three spatial scales.

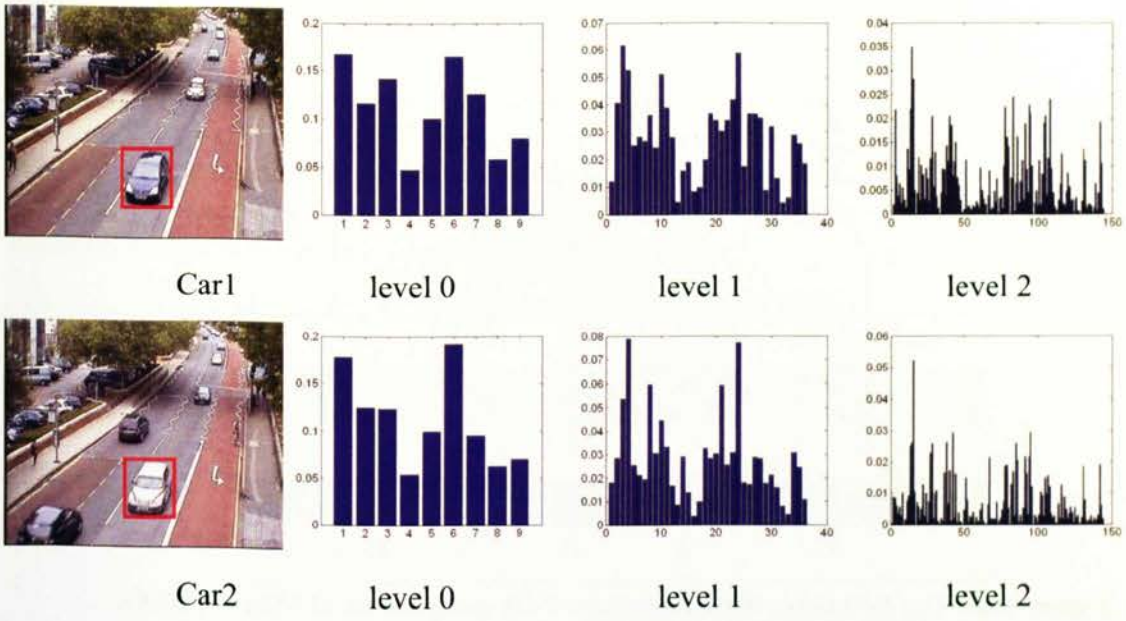


Figure 5.12. IPHOG of cars with different colour.

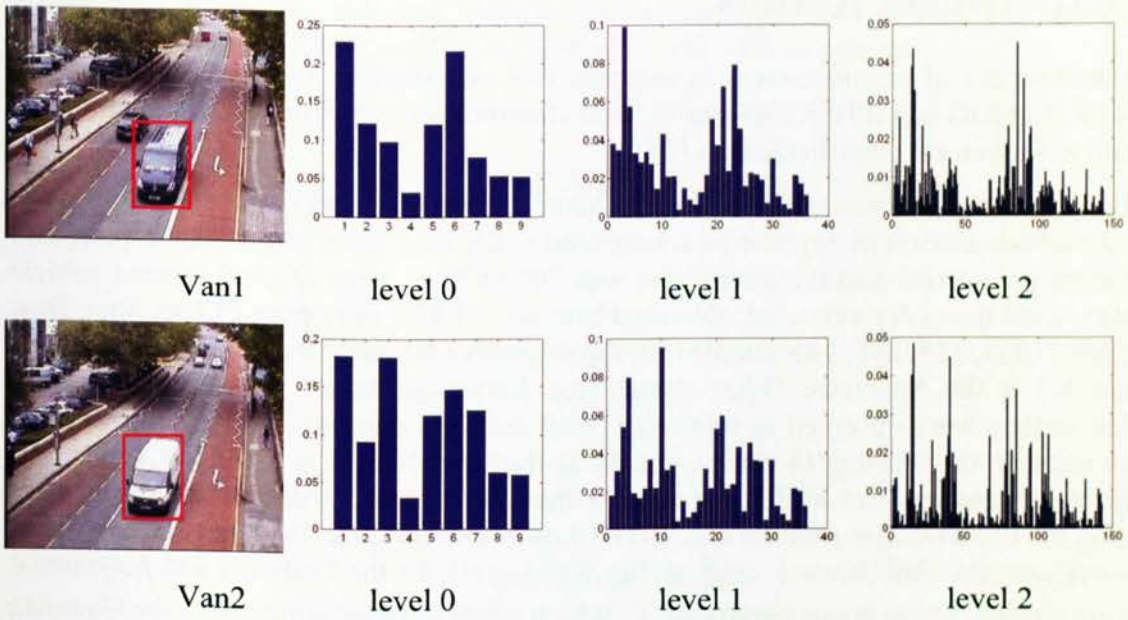


Figure 5.13. IPHOG of vans with different colour.

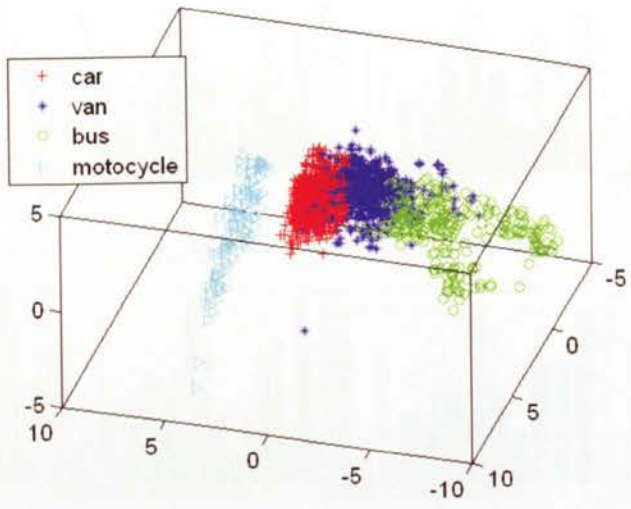


Figure 5.14. The first three most important PCA components of MBF+IPHOG.

5.5. Experiment Results

The objective of experiments is to compare two classification methods-SVM and RF, using MBF, EPHOG and IPHOG, to find the best classification algorithm and the best feature combination for vehicle classification in ITS.

The experimental data is taken from 5 hours of video recorded from a single pole-mounted roadside camera in daytime on a busy road in the local town centre. The capture rate is 25 frames per second and the image size was 704×576. A total of 2055 labeled vehicle silhouettes were manually extracted, allocated into one of four categories as Car, Van, Bus, Motorcycle (1033, 589, 290, 143 samples for the respective classes). Some samples are given in Figure A.5 in the Appendix. Other classes (e.g. lorries and heavy goods vehicles) were discarded as they were observed in relatively small numbers compared to the other classes. For data balance, the labeling of cars is limited as they proliferated in the video. Using half the dataset for training and half for testing, a modified 10-fold cross validation strategy (averaging the classification performance over 10 runs with different random selections) was used to evaluate the classification method. For SVM, $\gamma=1$ for the Gaussian and polynomial kernels are chosen. There is one parameter, C, which needs to be determined for the Gaussian kernel, and two parameters, C and d, for the polynomial kernel. In order to obtain a good value for C (so that the classifier can accurately predict unknown data), a "grid-search" on C and d (for example $C=2^{-5}, 2^{-3}, \ldots, 2^{15}$, $d=1,2,3$) was used. The observation MBF vectors were normalised to a standard score ($z=(x-\mu)/\sigma$, μ and σ are the mean and standard deviation of the raw vector x; the mean and standard deviation variation of z is from 0 to 1). For the random forests (RF) method, following published guidelines (Breiman, 2001) to construct forests containing 60 unpruned trees (the optimal number of trees) and setting $m=\sqrt{M}$, where M is the dimension of the feature vector. In order to evaluate the multi-class classification, the true positive rate (TPR) for each class defined as:

$$TPR = \frac{number\ \ of\ \ true\ \ postive}{total\ \ number\ \ of\ \ samples} \tag{5.23}$$

5.5.1. Compare SVM, RF and MBC

Experimental results indicate that model-based classification gives the poorest result (Table 5.2). The TPR of MBC for (car, van, bus, motorcycle/bicycle) are (0.9613, 0.7929, 0.7759, 1.0) respectively, giving an un-weighted average TPR of 0.88. One drawback of MBC is the need for camera calibration, which can be difficult to obtain, though high calibration accuracy was not found to be a critical factor.

Table 5.2. Confusion matrix for MBC

	Car	Van	Bus	Motorcycle
Car	993	40	0	0
Van	106	467	16	0
Bus	0	65	225	0
Motorcycle	0	0	0	143

The variation of TPR with increasing dimension of features is shown in Figure 5.15. Figure 5.15 illustrates that the first 6 features are the best choice for vehicle type classification for both SVM and RF, beyond which the TPR decreases.

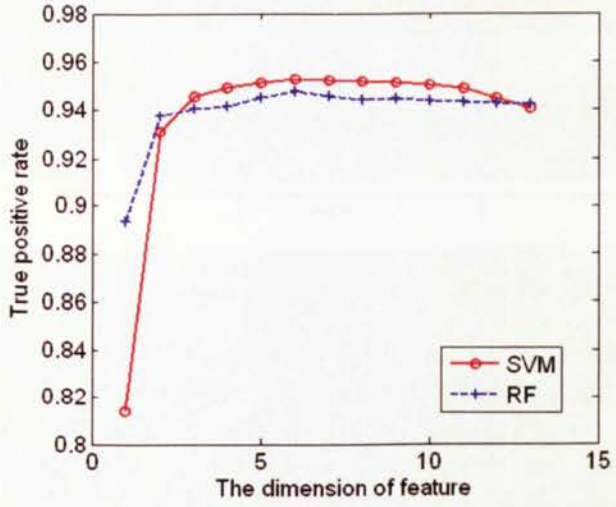


Figure 5.15. Variation of TPR with increase of feature dimension.

The TPR cross validation results of the best performing SVM (Gaussian kernel) and RF is plotted in Figure 5.16. The mean and standard deviation of TPR for SVM and RF are 0.9533 ± 0.0059 and 0.9479 ± 0.0043, respectively. Again, Figure 5.16 clearly shows that SVM outperforms RF.

The support vectors of the best performing SVM classifier are used to classify the entire data set. The resulting TPR for car, van, bus, and motorcycle/bicycle are 0.9661, 0.8913, 0.9931 and 1.0, respectively. The average TPR is 0.9626. The associated confusion matrix is given in Table 5.3. The mean and standard deviation of the ratio of the number of support vectors over the sample size for the 10-fold cross-validation is 0.3165 ± 0.002. The computational cost of the SVM classifier is also significantly lower than that of RF.

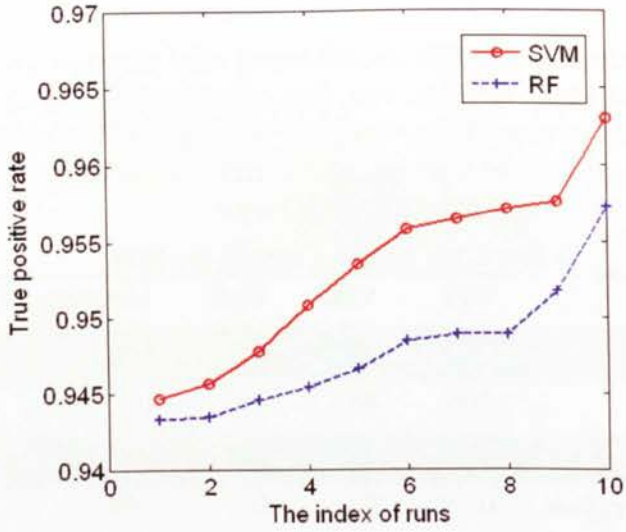


Figure 5.16. Results from cross validation for the best performance of SVM and RF.

Table 5.3. Confusion matrix for SVM

	Car	Van	Bus	Motorcycle
Car	998	35	0	0
Van	63	525	1	0
Bus	0	2	288	0
Motorcycle	0	0	0	143

5.5.2. Using MBF and PHOG

Table 5.4 gives the comparison of TPR results between SVM and RF using four different features: MBF, EPHOG, IPHOG and MBF+IPHOG. The TPR is the median of the 10-fold cross validation results. Table 5.4 shows that SVM outperforms RF, using MBF, EPHOG or IPHOG feature sets individually to classify vehicle types. In this case, IPHOG outperforms EPHOG and EPHOG outperforms MBF. Experiments combining pairs of feature sets produced the best result. The box plot in Figure 5.17 shows the stability and accuracy of TPR for 10 runs of SVM and RF using MBF+IPHOG. Figure 5.17 and Table 5.4 clearly illustrate that SVM, using a combination of MBF+IPHOG, is the best choice for vehicle types classification.

Using the support vectors of the best performing SVM to classify the entire data set resulted in an overall TPR of 0.9978. The associated confusion matrix is given in Table 5.5. False positive samples are given in Figure 5.18.

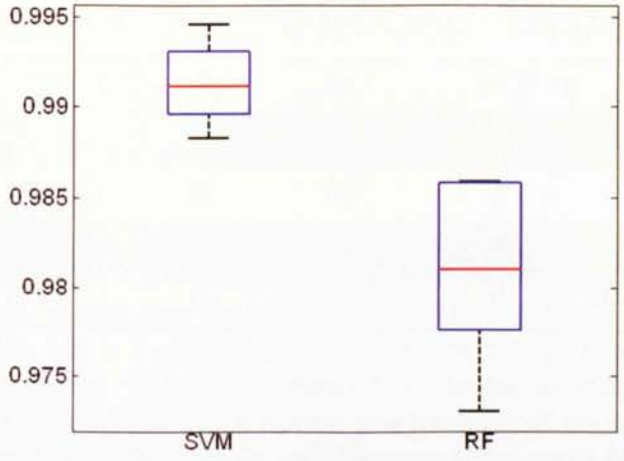


Figure 5.17. Box plot of TPR of the best performance of SVM and RF.

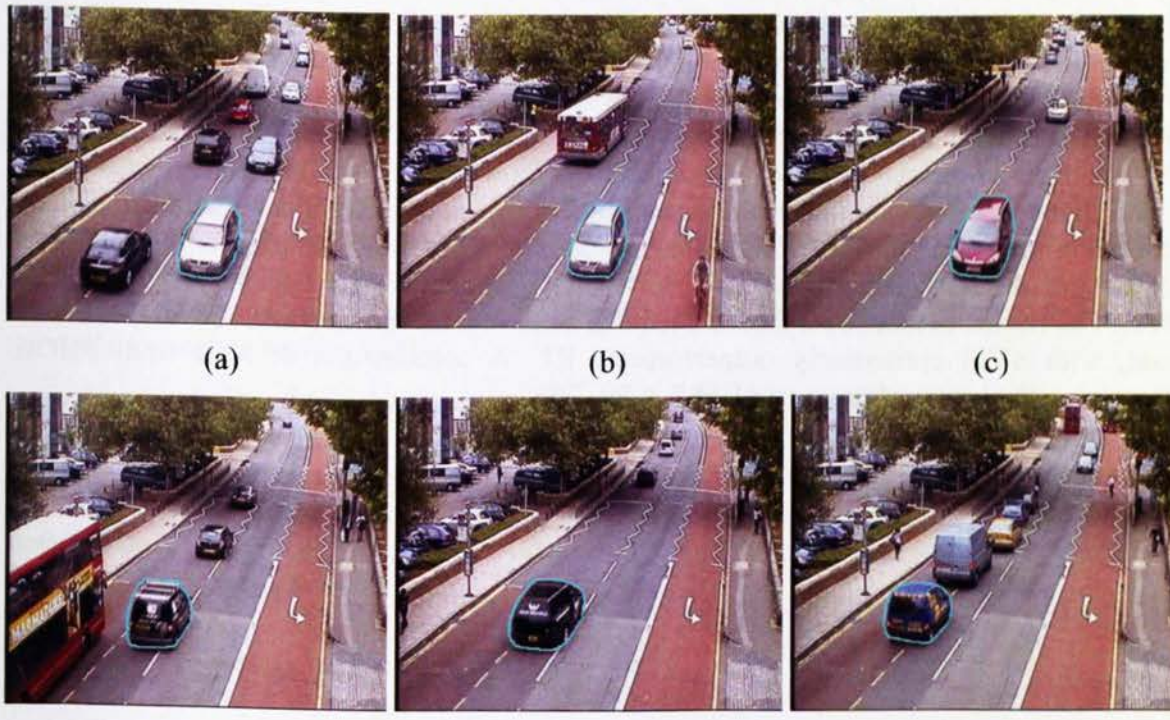


Figure 5.18. False positive samples (miss classify cars (a), (b) and (c) to vans, and miss classify vans (d), (e) and (f) to cars).

Table 5.4. The median TPR of SVM and RF using four different features.

	MBF	EPHOG	IPHOG	MBF+IPHOG
SVM	0.954	0.979	0.985	0.991
RF	0.951	0.969	0.973	0.981

Table 5.5. Confusion matrix for SVM and MBF+IPHOG over the entire data set

	Car	*Van*	*Bus*	*Motorcycle*
Car	1029	4	0	0
Van	3	586	0	0
Bus	0	0	290	0
Motorcycle	0	0	0	143

5.6. Summary

This chapter has presented a comparison of methods for categorising vehicle types into a broad range of classes using different approaches based on intensity features (EPHOG and IPHOG) and features derived from the manually-extracted silhouette (MBF). 10-fold cross validation has been used to evaluate the performance of the classification methods. Applying two popular classifiers to the silhouette feature set demonstrated that SVM consistently out-performed RF, with a final average true positive classification accuracy of 96.26%. The highest number of misclassifications occurs between the car and van categories, where both size and shape features exhibit significant similarity. The experimental results illustrate that a combination of MBF and IPHOG using SVM were the best choice for vehicle type classification, although using either MBF or IPHOG separately performs very well in terms of TPR. The results demonstrate that all methods achieve a recognition rate above 95% on the dataset, with SVM consistently outperforming RF. A combination of MBF and IPHOG features gave the best performance of TPR = 99.78%.

6. Automatic System for Vehicle Detection, Tracking and Classification

6.1. Introduction

Applying image processing technologies to vehicle detection and classification has been a hot focus of research in ITS over the last decade. Urban traffic flow analysis is important for traffic management, but is a challenging problem under high vehicle densities which can result in frequent occlusion. Several problems have to be solved, ranging from low and middle level vision tasks, such as the detection and tracking of multiple moving objects in a scene, to high level analyses, like vehicle classification (Messelodi et al., 2005). It is of great importance for dealing with the growing problem of urban congestion. Vehicle classification is particularly useful for re-identification (Kogut et al., 2001) in multi-sensor networks and anomalous event detection as well as the more standard applications of traffic flow analysis and unobtrusive path tracing (Bhonsle et al., 2000; Chang et al., 2004; Hu et al., 2004). The system presented in this chapter is an Automatic Vehicle Detection and Classification System (AutoVDCS). The AutoVDCS is able to detect vehicles as they move through the detection area of the camera's field of view, to track them and to classify individual objects into one of four main categories: motorcycle (motorcycle and bicycle), car, van (van, minivan, minibus and limousine), bus (bus and 2 decked bus), and counting them according to the category, collecting traffic data for statistical analysis. This chapter demonstrates the effectiveness of combining tracking with classification for significantly improved classification results on low resolution traffic video. The technique is general enough to be applied to a wide variety of surveillance scenes besides traffic.

The reminder of the chapter is organised as follows. The system overview is given in the next section. Section 6.3 describes vehicle detection. Vehicle detection and labeling is given in Section 6.4. Evaluation metrics are given is Section 6.5. Section 6.6 describes how to train the SVM classifier using synthetic data. Training the SVM system using automatic detected data is given in Section 6.7. AutoVDCS evaluation is given in Section 6.8. Finally, Section 6.9 summarises the chapter.

6.2. System overview

The system has four modules: background learning, foreground extraction, vehicle detection, vehicle classification and counting using the methods described in chapter 3-5. Figure 6.1 illustrates the flow chart of the AutoVDCS. In the vehicle detection module, the user specifies a vehicle census zone and the three lines as a virtual loop detector: StartLine (SL), MiddleLine (ML) and EndLine (EL) as shown in Figure 6.3(a). If the camera is fully calibrated, the system can be trained using synthetic data. The details are described in Section 6.6. However if the camera is uncalibrated, the system must be trained using manually annotated data, as described in chapter 5, or by manually adding vehicle class annotations to the automatically detected foreground blobs, which will be presented in Section 6.7. Figure 6.2 shows some snapshots of the GUI interface for the system. The details of background learning module and foreground extraction module have been described in chapter 4. The

next sections will give details of vehicle detection module and vehicle classification and counting module.

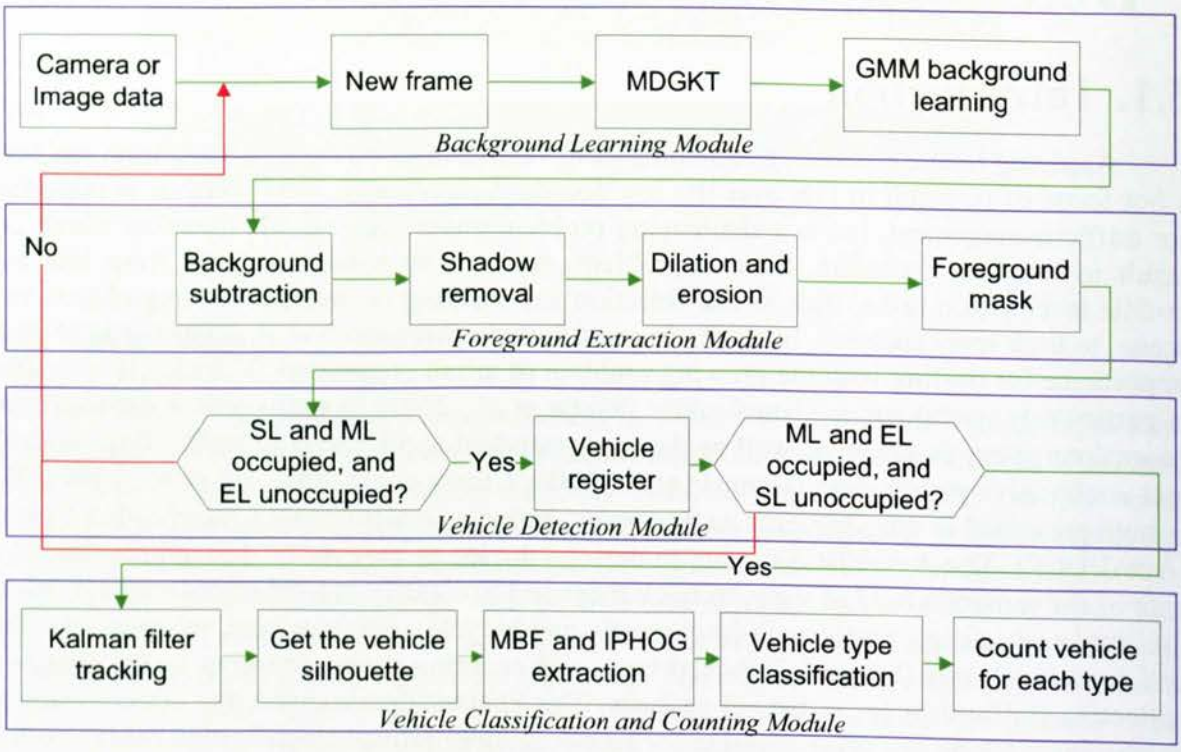


Figure 6.1. Flow chart of the AutoVDCS.

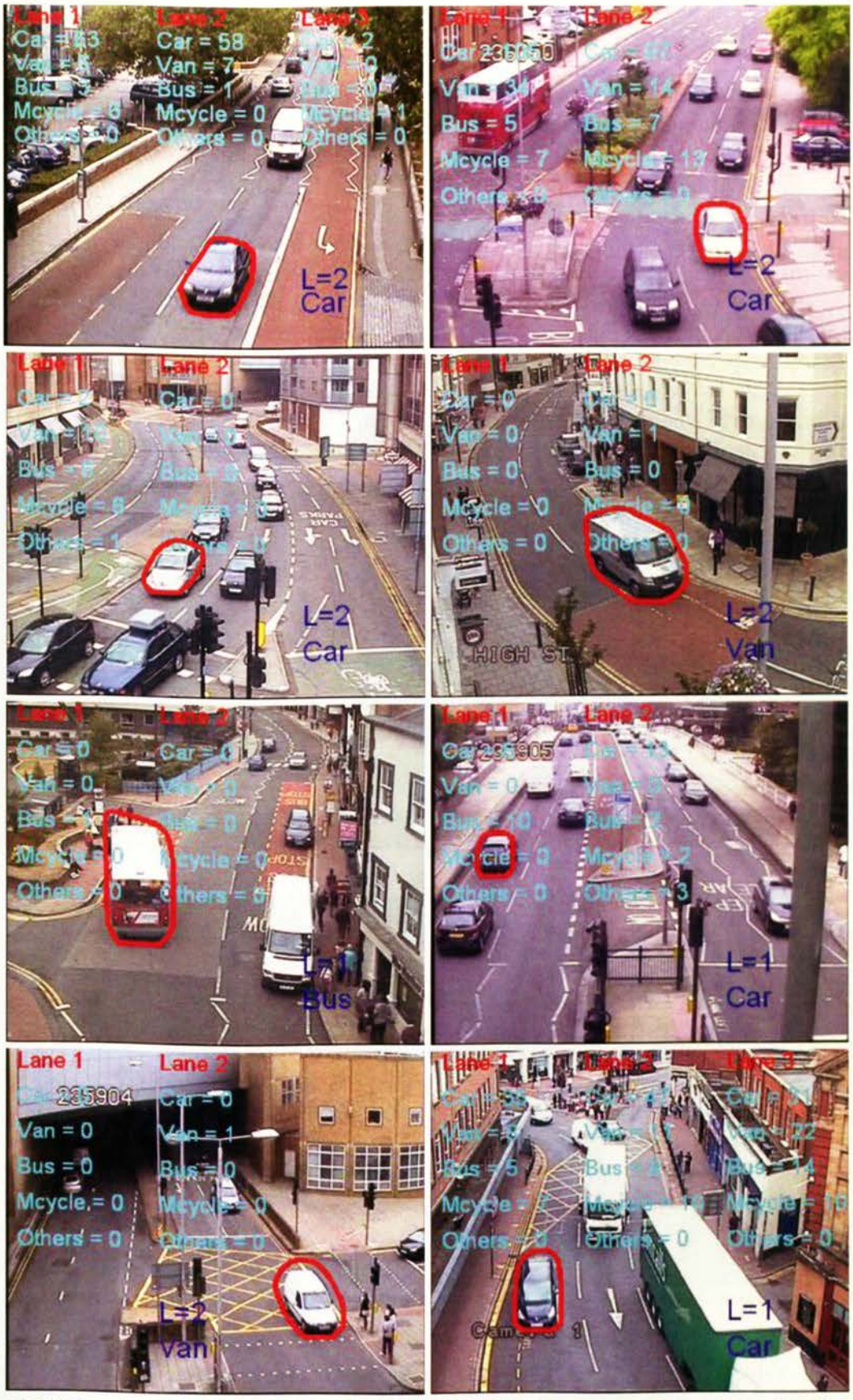


Figure 6.2. The snapshot of GUI interface of AutoVDCS showing summary of vehicle type counts per lane.

6.3. Vehicle detection

There are several key considerations when implementing a vehicle detection algorithm, and they vary depending on the specific task. For traffic flow statistics, it is essential to count each vehicle only once. To ensure that vehicles will only be counted as they appear in the detection zone, a virtual loop detector is applied. The virtual loop is comprised of three detect lines, StartLine (SL), MiddleLine (ML) and EndLine (EL). These line detectors are sensitive to miss-detection as a consequence of the ragged edge of a vehicle boundary. To minimize this effect the detectors have a finite width to ensure a stable detection of the vehicle when it intersects the line (a width of 5 pixels was used in the experiments described later). The separation between detector lines depends on the average traffic speed, and was set to 30 pixels in the experiments. The detector is configured to operate in both directions, to accommodate the two directions of traffic flow, and should be placed at a location where vehicles are clearly visible with minimal occlusion, i.e. usually closest to the camera. A detector is allocated to each lane to handle the measurements for each traffic stream.

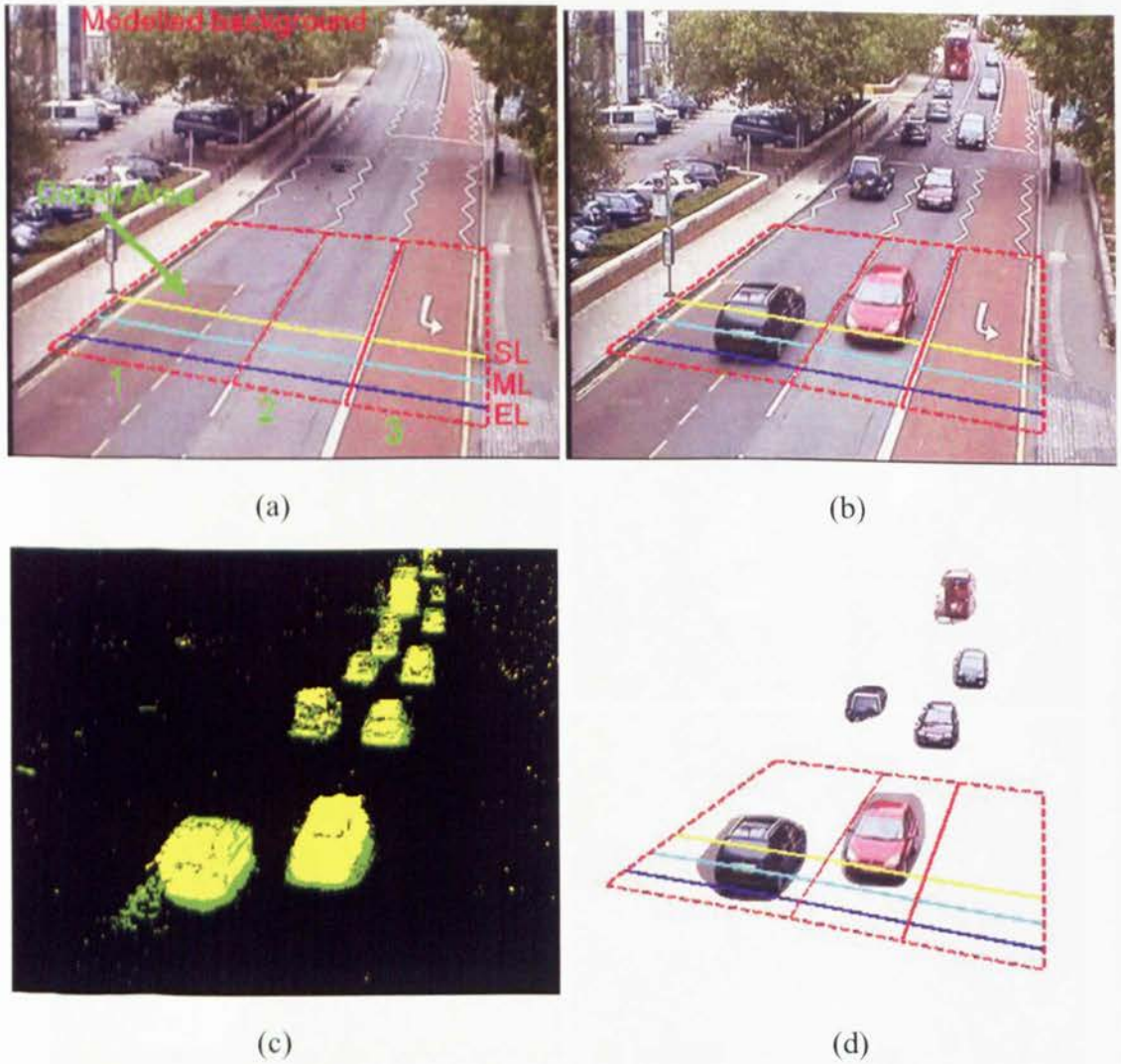
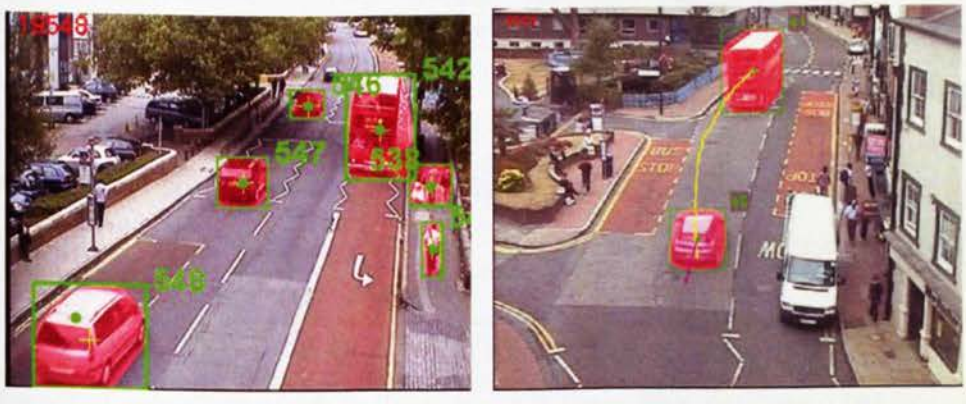


Figure 6.3. Vehicle detection. (a) GMM of background and predefined detection lines, (b) current input image with detection lines, (c) background subtraction results, (d) detected foreground blob with original colour pixels.

Figure 6.3 illustrates the object detection procedure. Shadow and reflection highlights pixels are removed, followed by a post-processing binary morphological opening to remove noise and small area objects. To ensure that vehicles are only counted once the detector considers a vehicle to be "present" only when both SL and ML are occupied and EL is unoccupied (for traffic moving towards the camera, i.e. lane 2 and 3). A vehicle is said to be "leaving" when ML and EL are occupied and SL unoccupied. A vehicle is counted only when it changes from the "present" state to the "leaving" state. This is reasonable in congested situations and even stopped traffic. In this way, the detector will not over-count in either case. If the proportion of pixels on the detect line is above a threshold (30% of the lane width), the line is considered occupied, otherwise it is unoccupied. This threshold is chosen as a tradeoff between detecting small vehicles (such as bicycles and motorbikes) but being insensitive to small blobs associated with noise. It is only necessary to swap SL and EL to account for vehicles in the traffic stream moving away from the camera (e.g. lane 1 in Figure 6.3(a)).

6.4. Vehicle tracking and labeling

The output of the vehicle detection step is a binary object mask which is used to perform region tracking. This provides multiple instances of the same vehicle, each of which are independently classified. Tracking employs the centroid of each detected blob, using a constant velocity Kalman filter model (Welch and Bishop, 2001). The state of the filter is the centroid location and velocity, $s=[c_x,c_y,v_x,v_y]^T$, and the measurement is an estimate of this entire state, $y=\hat{s}=[\hat{c}_x,\hat{c}_y,\hat{v}_x,\hat{v}_y]^T$. The data association problem between multiple blobs is solved by comparison of the predicted centroid location with the centroids of the detections in the current frame. The blob with it's centroid closest to the predicted location is chosen as the best match for the track. The track class label is computed at each frame but the final label is assigned by a majority voting scheme which considers the entire track to make a decision on class type, rather than employing a single frame that could be corrupted by different noise sources. Figure 6.4 illustrates the results of vehicle labeling with a track identifier.



(a) (b)

Figure 6.4. Kalman filter tracking results. (a) automatic label the detected objects and track them, (b) shows the tracking trace, yellow line is the trace of the bus and pink line is the trace of the car .

6.5. Evaluation metrics

Considering a two-class prediction problem (binary classification), in which the outcomes are labeled either as positive (p) or negative (n). There are four possible outcomes from a binary classifier. If the outcome from a prediction is p and the actual value is also p, then it is called a *true positive* (TP); however if the actual value is n then it is said to be a *false positive* (FP). Conversely, a *true negative* (TN) has occurred when both the prediction outcome and the actual value are n, and *false negative* (FN) is when the prediction outcome is n while the actual value is p.

Recall (REC, sensitivity, true positive rate – TPR, detection rate - DTR)

$$REC = \frac{TP}{TP + FN} \tag{6.1}$$

Precision (PRE, positive predictive value – PPV)

$$PRE = \frac{TP}{TP + FP} \tag{6.2}$$

False alarm rate (FAR, false positive rate – FPR)

$$FAR = \frac{FP}{FP + TN} \tag{6.3}$$

Specificity (SPE, true negative rate – TNR)

$$SPE = \frac{TN}{FP + TN} = 1 - FAR \tag{6.4}$$

Accuracy (ACC)

$$ACC = \frac{TP + TN}{TP + FN + TN + FP} \tag{6.5}$$

Jaccard coefficient (JC)

$$JC = \frac{TP}{TP + FP + FN} \tag{6.6}$$

Matthews correlation coefficient (MCC)

$$MCC = \frac{TP \times TN - FP \times FN}{(TP + FP)(TP + FN)(TN + FP)(TN + FN)} \tag{6.7}$$

F-measure (F1) considers both precision and recall of the test to compute the score

$$F_1 = \frac{2 \cdot REC \cdot PRE}{REC + PRE} \tag{6.8}$$

In order to evaluate the performance of automatic vehicle detection and classification system, an extended confusion matrix for N classes (with absolute counts in rows/columns C_1, C_2, ..., C_N) is used as follows:

Actual classes

Predicted classes	C_1	C_2	$\cdots$	C_N	FP
C_1	$c_{1,1}$	$c_{1,2}$	$\cdots$	$c_{1,N}$	$c_{1,N+1}$
C_2	$c_{2,1}$	$c_{2,2}$	$\cdots$	$c_{2,N}$	$c_{2,N+1}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
C_N	$c_{N,1}$	$c_{N,2}$	$\cdots$	$c_{N,N}$	$c_{N,N+1}$
FN	$c_{N+1,1}$	$c_{N+1,2}$	$\cdots$	$c_{N+1,N}$	$c_{N+1,N+1}$

If no overlapping ground truth is found, the classified label is entered in row FP (false positive, i.e. over counted). All misdetected vehicles in the ground truth are entered into column FN (false negative).

The number of true positives (TP) for any class C_i are the corresponding diagonal elements $c_{i,i}$. The recall REC_{ci} for each class and the precision PRE_{ci} for each class are defined as:

$$REC_{c_{i,j}} = \frac{c_{i,j}}{\sum_{j=1}^{N+1} c_{j,i}} \qquad PRE_{c_{i,j}} = \frac{c_{i,j}}{\sum_{j=1}^{N+1} c_{i,j}} \tag{6.9}$$

The joint REC, PRE scores for all classes are:

$$JREC = \frac{\sum_{i=1}^{N} c_{i,i}}{\sum_{i=1}^{N}\sum_{j=1}^{N+1} c_{j,i}} \qquad JPRE = \frac{\sum_{i}^{N} c_{i,i}}{\sum_{i=1}^{N}\sum_{j=1}^{N+1} c_{i,j}} \tag{6.10}$$

The joint classification accuracy ACC and detection rate DTR are defined as:

$$JACC = \frac{\sum_{i=1}^{N} c_{i,i}}{\sum_{i=1}^{N}\sum_{j=1}^{N} c_{i,i}} \qquad JDTR = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N+1} c_{i,j}}{\sum_{i=1}^{N+1}\sum_{j=1}^{N} c_{i,j}} \tag{6.11}$$

The joint false positive rate FPR and false negative rate FNR are defined as:

$$JFPR = \frac{\sum_{i=1}^{N} c_{i,N+1}}{\sum_{i=1}^{N+1}\sum_{j=1}^{N} c_{i,j}} \qquad JFNR = \frac{\sum_{j=1}^{N} c_{N+1,j}}{\sum_{i=1}^{N+1}\sum_{j=1}^{N} c_{i,j}} \tag{6.12}$$

Receiver operating characteristic (ROC):

A receiver operating characteristic or ROC curve is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting the fraction of true positives out of the positives (TPR) vs. the fraction of false positives out of the negatives (FPR), at various threshold settings. The Area Under Curve (AUC), when using normalized units, is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. An area of 1.0 represents a perfect test; an area of 0.5 represents a worthless test.

6.6. Training SVM by synthetic data

Manual data collection and annotation is time consuming and costly. Once the camera is calibrated, a vehicle wireframe model (Figure 5.1) can be projected onto the road surface to create a view-independent synthetic Measurement Based Feature (MBF, a silhouette outline of the model) to train the SVM which described in the Section 5.4.1.1. To model the variety and range of real vehicle silhouettes and noise effects in the background subtraction results, Gaussian noise (standard normal distribution, mean=0, $std = 5$) is added to the control points of the wireframe model, projected position and projected lane direction. Figure 6.6 shows sample of noise data. This creates variation in the size, shape, position and orientation of the projected vehicle wireframe and hence the resulting vehicle silhouette. A closed convex polygon around their extremal boundary is constructed to create synthetic MBF. A total of 3600 synthetic samples (car: 1482, van: 927, bus: 878, motorcycle: 313) were created with type distribution similar to the ground truth data presented in chapter 5. The separability of the four vehicle classes can be visualized by plotting the first three most significant PCA components of a normalized (mean and standard deviation scaled in range 0-1) feature vector, where one class (motorcycle) is unambiguously separable (see Figure 6.5). The car, van and bus categories exhibit a range of feature measures that are not clearly separable. For synthetic data, the best classifier parameters are chosen to be $\gamma=2$ for the Gaussian kernel, and $C=50$. A 10 fold cross-validation strategy was employed to evaluate the performance of the classification method. The mean and *std* of the REC, PRE and F1 of each class are given in Table 6.1. The support vectors of the best performing SVM classifier are used to predict the entire data set. The associated confusion matrix is given in Table 6.2. The column sum is the ground truth number of each vehicle class. Table 6.1 and 6.2 illustrate that motorcycles are perfectly discriminated, van and bus is easier to discriminate, but car and van are hard to distinguish. The overall four-class classification rate is 92.61%.

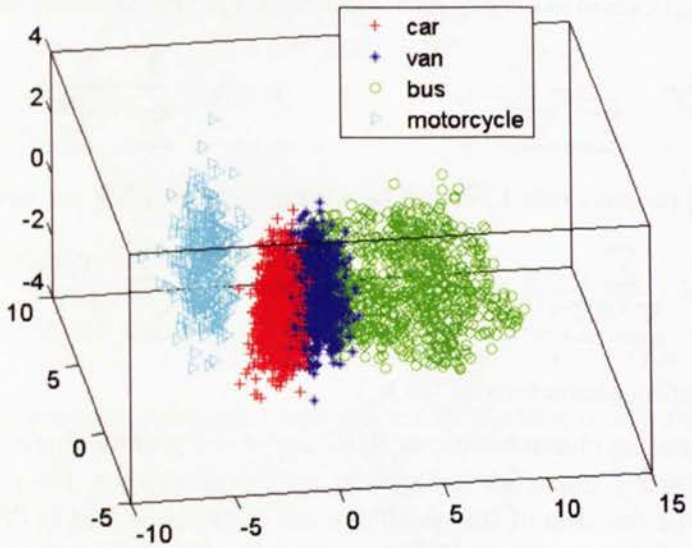
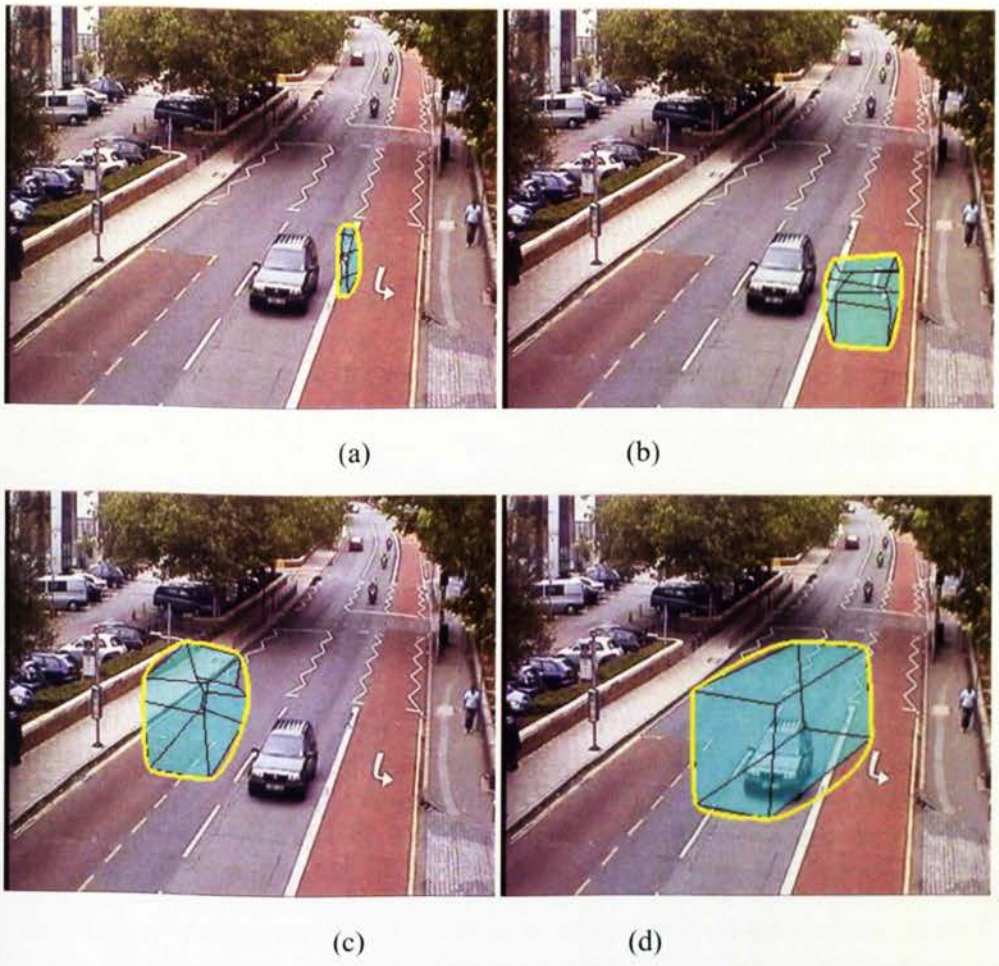


Figure 6.5. The first three most important PCA components of synthetic data.



(a) (b)

(c) (d)

Figure 6.6. Synthetic vehicle data set for training SVM classifier, yellow curve shows the convexhull silhouette. (a) motorcycle, (b) car, (c) van, (d) bus.

Table 6.1. The mean and *std* of the REC, PRE and F1 of each class

	Car	Van	Bus	Motorcycle
REC	0.949 ± 0.020	0.808 ± 0.033	0.951 ± 0.024	1.0 ± 0.0
PRE	0.898 ± 0.016	0.863 ± 0.033	0.976 ± 0.015	1.0 ± 0.0
F1	0.923 ± 0.015	0.833 ± 0.027	0.963 ± 0.014	1.0 ± 0.0

Table 6.2. Confusion matrix for noises synthetic data

	Car	Van	Bus	Motorcycle
Car	1414	68	0	0
Van	148	761	18	0
Bus	0	32	846	0
Motorcycle	0	0	0	313

6.7. Training SVM by automatic detected data

In order to investigate if using synthetic data to train the system is sufficient to predict a vehicle's category, the SVM is trained with manually labeled and automatically detected foreground blobs, and compare the predicted accuracy with the results form classifier which obtained by training synthetic data. The silhouettes of the foreground blob are not well defined. Occasionally due to occlusion, only a small part of a vehicle is detected. Because of the complexity of the detected foreground, blobs were annotated only where more than 1/3 of the pixels are detected. 8 video clips (totally 117941 frames, approximately 78.6 minutes) which were acquired under different weather conditions (overcast sky, light and heavy rain) were used. The capture rate is 25 frames per second and the image size was 352×288. Vehicle classification uses the majority vote for 5 sequential video frames. Each vehicle contributes 5 separate samples, giving a sample of 5760 car, 530 van, 180 bus and 100 motorcycle observations. A 202-dimensional feature vector is constructed, comprising the MBF and a pyramid-based HOG (Intensity-based Pyramid Histogram of Oriented Gradient, IPHOG) to train the SVM. For automatically detected silhouettes, the best parameters are γ=2 for the polynomial kernel, and C=5000. The first three most important PCA components are plotted in Figure 6.7.

A 10-cross validation strategy was employed, repeating the process 10 times and averaging the results in order to evaluate the performance of the classification methods. The mean and *std* of the REC, PRE and F1 of each class are given in Table 6.3. The support vectors of the best performing SVM classifier are used to predict the entire data set. The associated best confusion matrix is given in Table 6.4. It shows that all samples are perfectly classified.

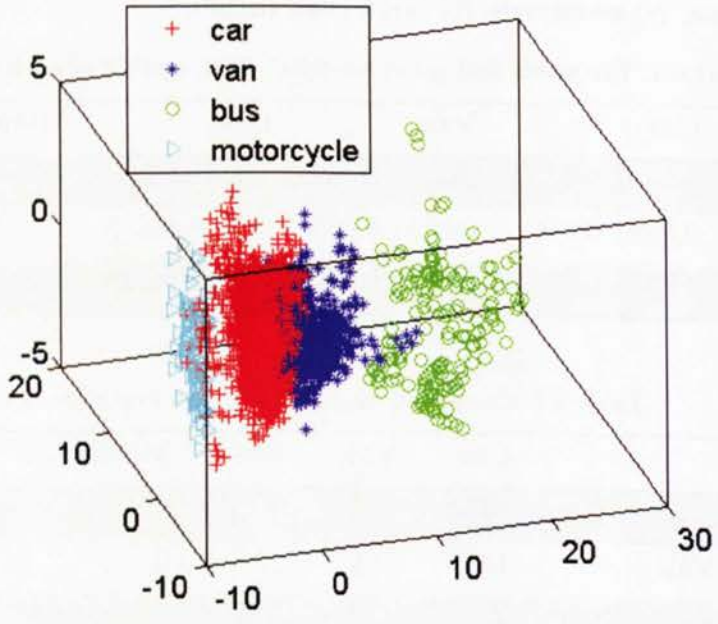


Figure 6.7. The first three most important PCA components of detected data.

Table 6.3. The mean and *std* of the *REC*, *PRE* and *F1* of each class from automatic detected dataset

	Car	Van	Bus	Motorcycle
REC	0.998±0.002	0.983±0.017	0.983±0.027	0.970±0.068
PRE	0.998±0.002	0.976±0.021	0.995±0.017	0.989±0.035
F1	0.998±0.001	0.979±0.014	0.989±0.015	0.979±0.051

Table 6.4. Confusion matrix for entire automatic detected data

	Car	*Van*	*Bus*	*Motorcycle*
Car	5760	0	0	0
Van	0	530	0	0
Bus	0	0	180	0
Motorcycle	0	0	0	100

6.8. AutoVDCS evaluation

To evaluate AutoVDCS in real scenarios, the video clips described in Section 6.7 is used. As previously emphasized, each vehicle must only be counted once. A total of 1402 cars, 216 vans, 82 buses and 27 motorcycles were identified in the video. In order to analyze performance, 7 different combinations of test and training data are compared:

Method 1: train SVM using synthetic MBF, classify foreground blobs obtained by background subtraction;

Method 2: train SVM using synthetic MBF, classify foreground blobs obtained by background subtraction and majority vote over 5 frames;

Method 3: train SVM using manually extracted MBF, classify foreground blobs obtained by background subtraction and majority vote over 5 frames;

Method 4: train SVM using detected MBF obtained by background subtraction, classify foreground blobs and majority vote over 5 frames;

Method 5: train SVM using detected MBF+IPHOG from background subtraction, classify foreground blobs and majority vote over 5 frames;

Method 6: train SVM using detected MBF from background subtraction, classify foreground blobs obtained by background subtraction and levelset detection, and majority vote over 5 frames;

Method 7: train SVM using detected MBF+IPHOG from background subtraction, classify foreground blobs obtained by background subtraction and levelset detection, and majority vote over 5 frames.

Methods 6 and 7 employ the improved level set method proposed in chapter 3. The main advantage of this method is that it is an active segmentation method. The level set energy formulation includes information on the mixture of multiple channels and multiple

regions. Object boundaries that include different known colours are segmented against complex backgrounds and it is not necessary for the object to be homogeneous.

At the training stage, 10-cross validation strategy is used to choose the best parameters of SVM classifier. The support vectors from the best performance are selected to classify the entire dataset. The comparison of vehicle detection accuracy is given in Table 6.5 shows that the level set algorithm improves detection performance.

The classification results in terms of JREC, JPRE and JF1 are given in Table 6.6. It shows that method 5 results in the best combination, indicating that whilst using level set detection improves the vehicle detection rate, it results in a lower classification rate. This is because when multiple vehicles are too close and in the detection area, they will be merged into a single blob by background subtraction, but level set detection segments them into multiple regions (Figure 6.8). However, if a vehicle's colour is similar to the background, it's silhouette may be significantly reduced (Figure 6.9), and as a result SVM classifies is incorrectly. The extended confusion matrix of method 5 is given in Table 6.7. The PRE of the extended confusion matrix of method 5 for each class is given in Table 6.8. REC of extended confusion matrix of method 5 is given in Table 6.9. The classification accuracy is 94.69%.

Much better results can be obtained if only using the video taken in good weather (video clips 1-4). 55432 frames (about 37 minutes) which include a total number of 769 vehicles (625 cars, 87 vans, 41 buses and 16 motorcycles). Only the detection results obtained by Method 5 are given. The extended confusion matrix is given in Table 6.10. The detect rate is 95.71%, false negative rate is 5.33%, and false positive rate is 1.04%. The classification accuracy is 96.43%, $JREC = 0.9129$ and $JPRE = 0.9538$, JF1 = 0.9329.

In the second experiment, only method 5 is used for a gray video sequence sampled in the evening. A total of 751515 frames (about 50 minutes) are included in the video. 858 vehicles were detected from the total of 921 vehicles (738 cars, 85 vans, 22 buses and 76 motorcycles) pass the road. The capture rate is 25 frames per second and the image size was 360×288. The confusion matrix is given in Table 6.11. The detect rate is 93.16%, false negative rate is 6.84%, and false positive rate is 1.74%. The classification accuracy is 95.22%, $JREC = 0.8871$ and $JPRE = 0.9348$, JF1 = 0.9103. The snapshot of the GUI interface is shown in Figure 6.10.

Table 6.5. Vehicle detection accuracy.

	DTR	FNR	FPR
Background subtraction	0.944	0.072	0.017
Background subtraction + level set	0.964	0.050	0.014

Table 6.6. The comparison of classification accuracy.

	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6	Method 7
JREC	0.814	0.819	0.805	0.867	0.878	0.840	0.882
JPRE	0.862	0.868	0.853	0.919	0.930	0.871	0.915
JF1	0.837	0.843	0.829	0.892	0.904	0.855	0.898

(a) (b)

Figure 6.8. Level set improve the vehicle detection results, (a) shows the result only use background subtraction, two buses and a car merge together, (b) shows the result after level set process, car is detected correctly.

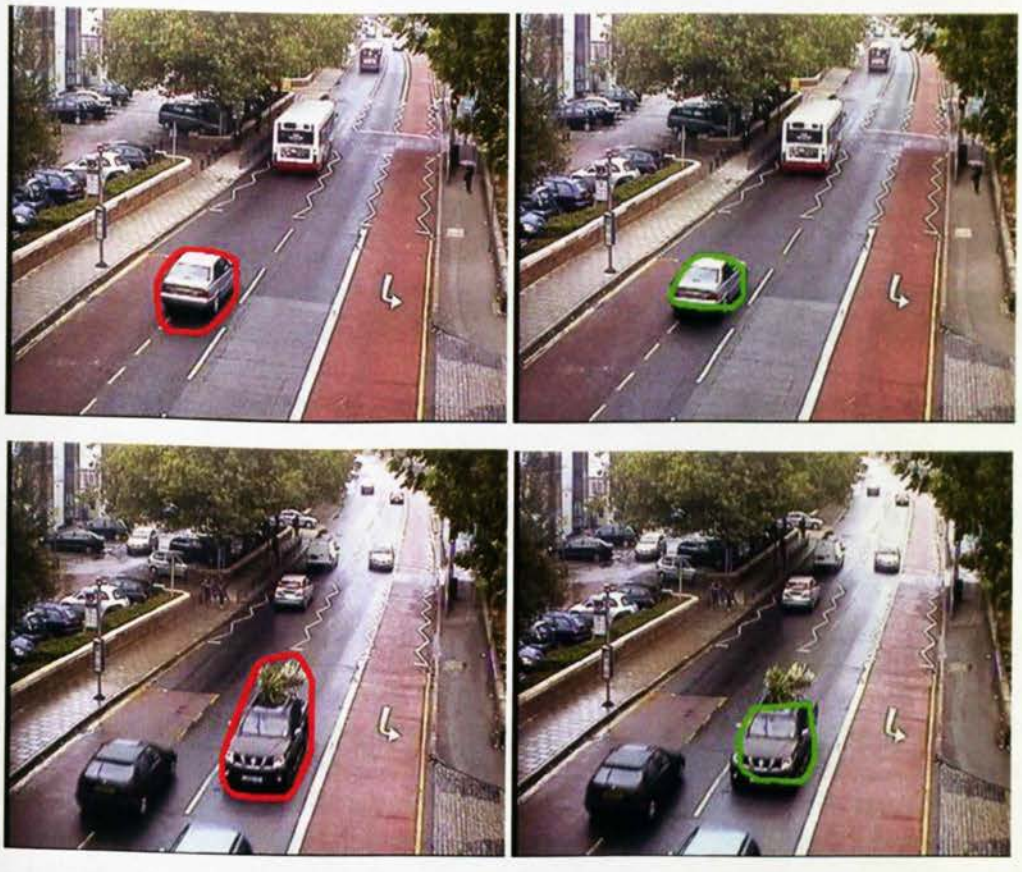


Figure 6.9. Compare the results of background subtraction and level set. Left column are the original background subtraction results. Right column is the results from level set.

Table 6.7. The extended confusion matrix of method 5.

	Car	Van	Bus	Motorcycle	FP
Car	1279	21	1	0	12
Van	6	158	3	1	6
Bus	32	20	59	0	6
Motorcycle	1	0	0	21	5
FN	84	17	19	5	0

Table 6.8. PRE of extended confusion matrix of method 5.

	Car	Van	Bus	Motorcycle	FP
Car	0.9741	0.0160	0.0008	0	0.0091
Van	0.0345	0.9080	0.0172	0.0057	0.0345
Bus	0.2735	0.1709	0.5043	0	0.0513
Motorcycle	0.0370	0	0	0.7778	0.1852
FN	0.0640	0.0004	0.0005	0.0024	0

Table 6.9. REC of the confusion matrix of method 5.

	Car	Van	Bus	Motorcycle	FP
Car	0.9123	0.0972	0.0122	0	0.0086
Van	0.0043	0.7315	0.0366	0.0370	0
Bus	0.0228	0.0926	0.7195	0	0.0001
Motorcycle	0.0007	0	0	0.7778	0.0003
FN	0.0599	0.0787	0.2317	0.1852	0

Table 6.10. The extended confusion matrix of method 5 for the good weather data.

	Car	Van	Bus	Motorcycle	FP
Car	595	9	1	0	2
Van	0	66	3	1	2
Bus	5	7	28	0	2
Motorcycle	0	0	0	13	2
FN	25	5	9	2	0

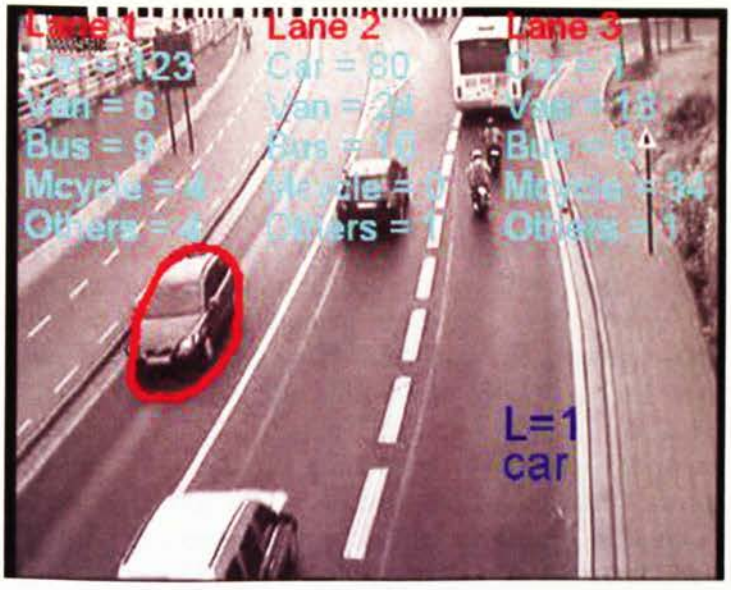


Figure 6.10. The snapshot interface of the second experiment.

Table 6.11. The extended confusion matrix of the second experiment.

	Car	Van	Bus	Motorcycle	FP
Car	703	22	1	0	7
Van	9	51	4	0	8
Bus	2	1	7	0	0
Motorcycle	1	0	1	56	1
FN	23	11	9	20	0

6.9. Summary

Acquisition of reliable vehicle counts and classification data is necessary to establish an enriched information platform and improve the quality of ITS. The approach proposed in this chapter is a hybrid algorithm. In the background learning module, foreground extraction module and vehicle detection module, the improved background subtraction method (chapter 4) is integrated to alleviate the negative impacts from camera vibration, shadow and reflection highlights, sudden illumination changes and gradual changes. A level set method (chapter 3) has been used to refine the foreground blob. In the vehicle classification module, a Kalman filter and SVM (chapter 5) are integrated to improve accuracy. Extensive experiments have been undertaken, comparing 7 combinations of detection and classification methods. Results show that the best combination is to train the SVM using MBF+IPHOG features extracted by background subtraction, classifying the foreground blobs using a majority vote over 5 consecutive frames. The results demonstrate a vehicle detection rate of 96.39% and classification accuracy of 94.69% under varying illumination and weather conditions (from cloud to rain).

7. Confidence Based Active Learning

7.1. Introduction

Data collection and annotation is a crucial part of pattern classification system development because it determines the success of later stages. Moreover, data collection and annotation is surprisingly time consuming and costly. The widely used approach for data collection and annotation is called passive learning, where samples are randomly and independently selected from the underlying distributions; human assessors then manually annotate these samples. Considering the time and cost associated with this process, it is always the case that there are not enough training samples to assure a certain level of performance after training. In many applications, such as Web searching, information retrieval, and speech recognition, it is relatively easy and inexpensive to collect a large amount of data, while it is very expensive and time consuming to annotate the data. In these situations, active learning would be a suitable approach to minimize the effort of annotation (Lewis and Catlett, 1994). In active learning, the learning process repeatedly queries unlabeled samples to select the most informative samples to annotate and update its learned rules. Therefore, the unnecessary and redundant annotation is avoided, greatly reducing the annotation cost and time. Active learning is also helpful in reducing computational complexity. Active learning has been shown to be more powerful than learning from random examples in the papers (Khammari et al., 2005; Li and Sethi, 2006; Roth and Bischof, 2008; Sivaraman and Manubhai, 2010). The chapter 6 described that the best performance results from training the SVM using MBF+IPHOG features detected using background subtraction, classifying foreground blobs and tracking over consecutive 5 frames using a Kalman filter. One question need to answer is, how many annotated data is enough. This chapter uses confidence-based active learning for training a SVM classifier. The approach takes advantage of the classifiers probability preserving and ordering properties (Zadrozny and Elkan, 2002; Drish, 2001). It calibrates the output scores of current classifiers to the class-conditional error. Thus, it can estimate the uncertainty level of each sample according to the output score of a classifier and select only those samples for annotation whose output scores are in the uncertain range.

Only support vectors play a role in SVM learning and the removal of non-support vectors does not change the training results (Burges, 1998). Support vectors should be located in the boundary between two classes. That means if the classifier is trained only using the high uncertainty data, it won't affect the training results.

The reminder of the chapter is organised as follows. Next section describes how to convert SVM scores into probabilities. Section 7.3 presents confidence based active learning using a SVM classifier. Extensive experiments using Gaussian and non-Gaussian distribution data set are given in Section 7.4. Section 7.5 summarises the chapter.

7.2. Converting SVM scores into probabilities

The output of a classifier should be a calibrated posterior probability to enable post-processing. Standard SVMs do not provide such probabilities. The SVM output score is not a probability but a distance from the separating hyperplane. The sign of the score indicates if the example is classified as positive or negative. The magnitude of the score can be taken as a

measure of confidence in the prediction, since examples far from the separating hyperplane are presumably more likely to be classified correctly. Thus, it needs a way to transform SVM output scores to probabilities according to confidence-based active learning. Platt (1999) trained a SVM, and then trained the parameters of an additional sigmoid function (7.1) to map the SVM outputs into probabilities.

$$P(y=1|f)=\frac{1}{1+\exp(Af+B)} \tag{7.1}$$

where f is the SVM score. This sigmoid model is equivalent to assuming that the output of the SVM is proportional to the log odds of a positive sample. It has two parameters, A and B, trained discriminatively. For the probability estimation problem in one dimensional space, the histogram is an effective and efficient method. Drish (2001) proposed a binning method. For the fixed bin-width allocation method, sorting the training examples according to their scores firstly, and then dividing them into b equal sized bins, each having an upper and lower bound. Given a test example x, it is placed in a bin according to its score. The corresponding probability P(j=1|x) is the fraction of positive training examples that fall within the bin.

A difficulty of the binning method is that the number of bins have to be chosen by cross-validation. A very large bin width will produce a smooth histogram with too little detail; on the other hand, a very small bin width will result in a jagged histogram and a small number of samples in each bin will make a too large contribution. Ideally, the width of bins is chosen so that the estimated probability reflects the true underlying probability distributions without giving too much credence to the dataset at hand.

Zadrozny et al. (2002) improved Platt's method by an intermediary approach between sigmoid fitting and binning. However, there is no theoretical proof that a posterior probability has a sigmoidal shape and most likely it does not. Instead of equal bin width, Li and Sethi (2006) presented a method using an equal number of samples in each bin, called dynamic bin-width (DBW) allocation. This will give a smooth histogram where conditional probabilities are small and it will also give more detail where conditional probabilities are large. In other words, it adapts the underlying probability distribution.

In order to further improve the accuracy of converting SVM scores to probabilities using DBW, a smooth interpolated DBW histogram (SIDBW) is proposed in this chapter. The experimental results demonstrate the high accuracy of the new algorithm.

7.3. Confidence based active learning using SVM

The active-base learning algorithm used here is based on the algorithm proposed by Li and Sethi (2006). A SIDBW allocation strategy is used to convert SVM scores to probabilities. The pseudo code of the algorithm as follows:

// initialisation
Randomly select a small set of samples from the unlabeled sample pool *U*, assign a class label to each of them, and add them into training set *L*;
Train an SVM using *L*;
// find the query function thresholds in output score space using SIDBW method
While stopping criterion is not satisfied
    // Find the negative band
    Find $minY = min$(negative scores of support vectors from positive class);
    Sort scores $S_{neg} \in [minY, 0]$ of all training samples in ascending order;

Define the number of samples in each bin as N_{neg};
Put the S_{neg} into M_{neg} bins, M_{neg} = floor(length(S_{neg})/N_{neg});
***NegativeBound* = 0;**
For (*i*=1; *i*<M_{neg}; *i*++)
Compute error as percentage of positive samples in *i*th bin;
If the error is bigger than a threshold C_{neg}
***NegativeBound* = the bin starting point;**
break;
End if
End for
// Find the positive band
Find *maxY* = *max*(positive scores of support vectors from negative class);
Sort scores $S_{pos} \in [0, maxY]$ of all training samples in ascending order;
Define the number of samples in each bin as N_{pos};
Put the S_{pos} into M_{pos} bins, M_{pos} = floor(length(S_{pos})/N_{pos});
***PositiveBound*=0;**
For (*i*=1; *i*<M_{pos}; *i*++)
Compute error as percentage of positive samples in *i*th bin;
If the error is less than a threshold C_{pos}
***PositiveBound* = the bin ending point;**
break;
End if
End for
Classifying the samples in unlabeled sample pool U using SVM;
Finding samples U_Q with score S_u, $NegativeBound \le S_u \le PositiveBound$;
Sort score $|S_u|$ in ascending order;
Select first k samples from U_Q and assign a class label to each of them;
Add them into training set *L*;
Retrain the SVM using updated training set *L*;
End while

There are several ways to stop the closed loop. First, the loop would be automatically terminated if there are no uncertain samples in the unlabeled sample pool. This is the ideal situation since all the most informative samples are labeled in the pool. The second termination method is that human annotators stop annotation. This always happens. The best time to stop the training is when the targeted performance is achieved.

7.4. Experiments

In order illustrate that the SIDBW improves the accuracy of the conversion method, two algorithms to estimate probabilities from SVM scores have been compared in this section using both Gaussian distribution dataset and non-Gaussian distribution dataset. The two algorithms are: dynamic bin width (DBW) histogram algorithm and smooth interpolated dynamic bin width (SIDBW) histogram algorithm. Two experiments have been used to compare the accuracy of the algorithms. One uses synthetic examples with a Gaussian distribution. Another one uses synthetic examples with a Gamma (non-Gaussian) distribution. Mean square error (MSE) is used to calculate the accuracy of the conversion methods. Square error (SE) is defined as

$$SE = \sum_j (t(j|x) - p(j|x))^2 \tag{7.2}$$

where $p(j|x)$ is the probability estimated by the method for example x and class j, and $t(j|x)$ is the true probability of class j for x. For data sets where true labels are known and the probabilities are unknown, $t(j|x)$ is defined to be 1 if the label of x is j and 0 otherwise.

7.4.1. Gaussian distribution dataset

7.4.1.1. Passive learning from GD

In the first experiment, 4000 2D points are created from each of two Gaussian distributions (GD), $x \in N(-0.3, 0.3)$ and $y \in [-0.5, 0.5]$. The training data set is shown in Figure 7.3. A 10-cross validation strategy has been used to get the best parameters for the linear SVM classifier. Using half training and half testing dataset to get SVM scores and convert them to probabilities. 111 support vectors (5.55% training data) are obtained. The *ACC*, *REC*, *PRE*, and *F1* for training and testing data are given in Table 7.1.

Table 7.1. The ACC, REC, PRE, and F1 for training and testing entire synthetic Gaussian distribution data set.

	ACC	REC	PRE	F1
training	0.978	0.980	0.976	0.978
testing	0.973	0.973	0.973	0.973

The histogram for $p(f|y=\pm 1)$ for SVM on training synthetic data is given in Figure 7.1. The solid red line is $p(f|y=+1)$, while the dashed blue line is $p(f|y=-1)$. Notice that these histograms are approximate Gaussian distributions.

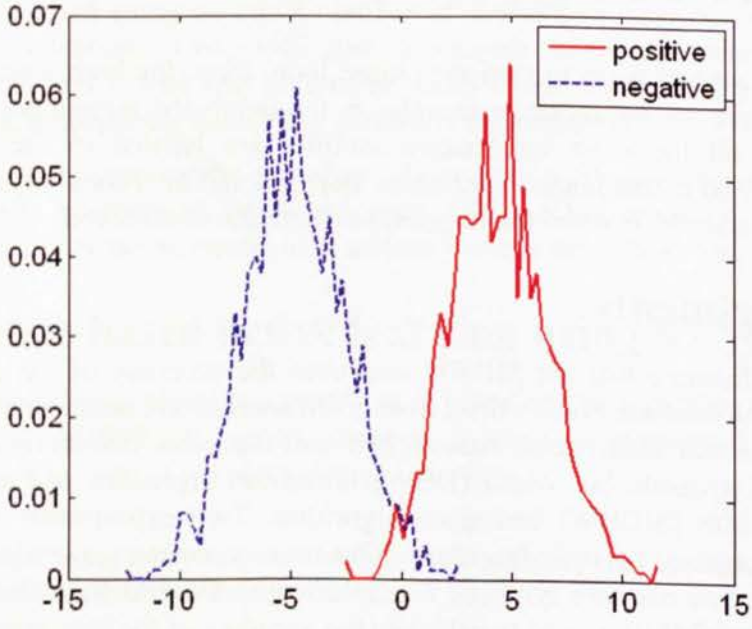


Figure 7.1. The histogram for $p(f|y=\pm 1)$ for SVM on training synthetic Gaussian distributed data set.

Figure 7.2 shows the histograms of probability estimation from DBW and SIDBW. The MSE of probability estimation using DBW and SIDBW from SVM scores of training dataset

are 0.0198 and 0.0036, respectively. Obviously, SIDBW improves the accuracy of probability estimation.

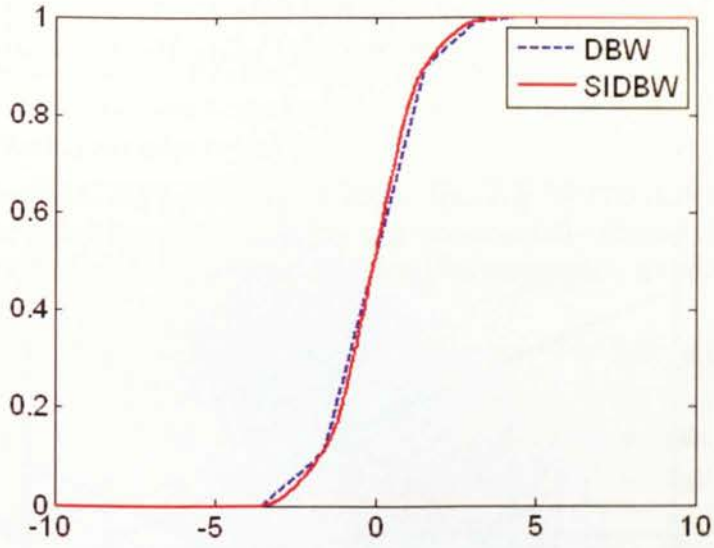


Figure 7.2. Histograms of probabilities from DBW and SIDBW.

7.4.1.2. Active learning from GD with known calibrated probability estimation

Active learning from GD with known calibrated probability estimation is presented in this Section. Using the classifier's output calibrated probabilities as confidence, the query function is:

$$Q(x) = \begin{cases} 1 & T_1 < f(x) < T_2 \\ 0 & otherwise \end{cases} \tag{7.3}$$

where $f(x)$ is the output probability from the classifier. The threshold T_1=0.2 and T_2=0.95 for the experiment. 109 low confidence samples (5.45% of training data) are obtained. 83.78% support vectors are included in the set of low confidence samples. Only the low confidence samples are used to train classifier, and the classifier to be used to classify the entire training and testing data. Table 7.2 gives the results of *ACC*, *REC*, *PRE*, and *F1*.

Table 7.2. The ACC, REC, PRE, and F1 for training and testing entire synthetic Gaussian distribution data set only use low confidence samples to train the classifier.

	ACC	REC	PRE	F1
training	0.979	0.980	0.977	0.979
testing	0.973	0.972	0.973	0.973

Comparing Table 7.1 and Table 7.2, it shows that only 5.45% low confidence samples are used to achieve the same training results as those from all samples. It is obvious that active learning does do a good job to figure out influential samples and reduce the cost of annotation and training. On the other hand, for a huge training data set, to train a classifier is a time consuming procedure. The low confidence samples are only a small proportion of data

compared to the original training data set. It is very efficient to train the classifier by using low confidence samples. This is another advantage of active learning.

Figure 7.3 shows the passive learning results of 2000 samples from two Gaussian distributed datasets, all the dataset has been used to train the classifier. Figure 7.4 shows the active learning results of the same data set, but only using 109 low confidence samples to train the classifier.

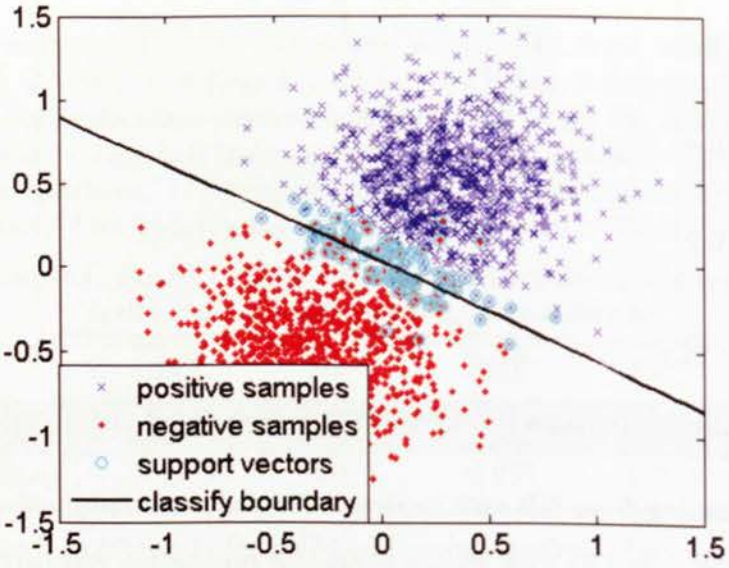


Figure 7.3. Passive learning SVM results of two Gaussian distributed classes, 2000 samples have been used to train the classifier.

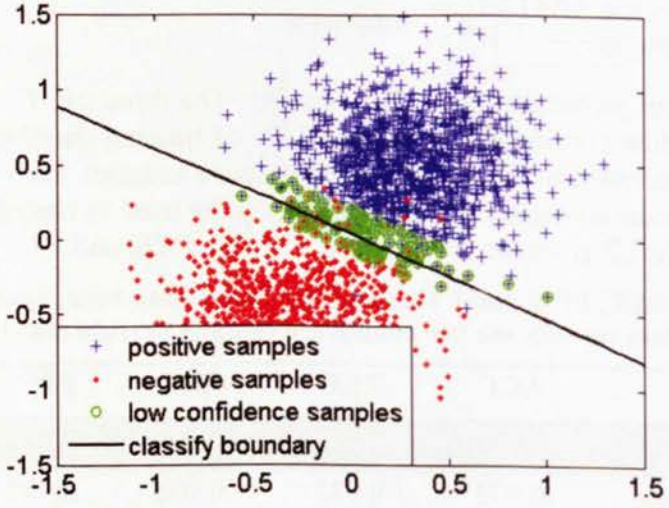


Figure 7.4. Active learning SVM results of two Gaussian distribution classes, 109 low confidence samples have been used to train the classifier.

7.4.1.3. Active learning from GD with unknown probability distribution

In this Section, active learning from GD with unknown probability distribution is presented. 50 samples are randomly chosen as an initial training set to train the linear SVM

classifier. The ACC of the entire training data set is 96.40%. Obviously, this is not as good as the ACC from all the 2000 training samples. After converting the SVM scores to probabilities, 43 low confidence samples are selected from remaining 1950 unlabeled samples. In the next round, 93 samples (the 50 initial samples plus 43 new selected the most informative samples) are used to form the training set. The ACC of the training data set increases to 98.25%. But the passive learning accuracy of training dataset is 97.80%. That means only 93 samples are used to achieve the same (or even better) training results as those from all 2000 samples. The proportion of the training samples is just 4.65%.

Figure 7.5 illustrates the processing of active learning for GD data set. From the plot it can be seen that the active learning algorithm can incrementally choose the most informative samples and correct the training error from the previous incomplete training.

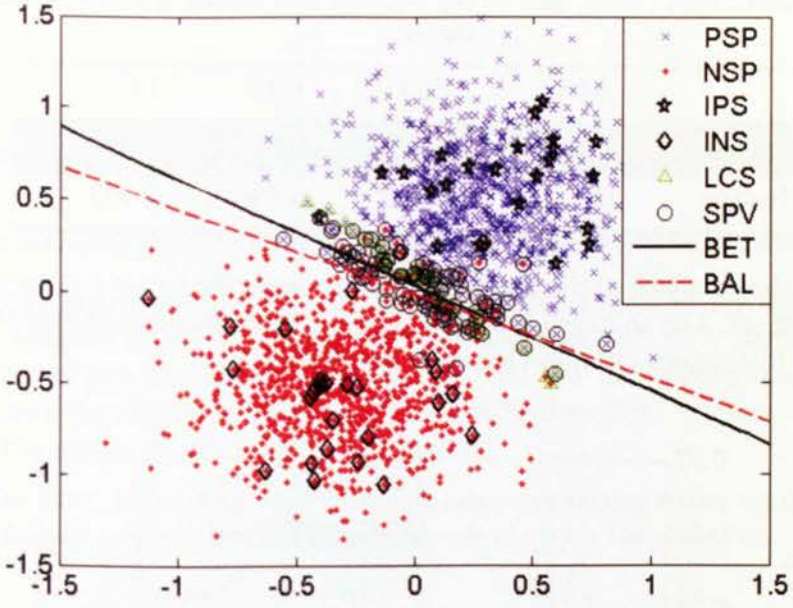


Figure 7.5. Active learning and passive learning processing for GD. PSP: positive samples; NSP: negative samples; LCS: selected low confidence samples; IPS: Initial random positive samples; INS: initial negative samples; SPV: support vectors from entire training dataset; BET: classification boundary from entire training dataset; BAL: final classification boundary of active learning.

7.4.2. Non-Gaussian distribution dataset

7.4.2.1. Passive learning from NGD

In the second experiment, a Gamma distribution has been used to create non-Gaussian distributed (NGD) samples. The gamma distribution, like the lognormal distribution, is an alternative to consider for ecological variables that seem to be highly skewed. If the random variable Y is gamma-distributed with parameters α and β, then the likelihood of Y is

$$p(Y) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} Y^{\alpha-1} e^{-\beta Y} \tag{7.4}$$

where the Gamma function $\Gamma(x)$ is defined as

$$\Gamma(x) \equiv \int_0^\infty t^{x-1} e^{-t} dt \tag{7.5}$$

As in the lognormal distribution, Y and the parameters α and β must be positive. The parameter α is called the shape parameter.

4000 2D points are created from two Gamma distributions, with shape parameter $\alpha = 4$. 10-cross validation strategy has been used to get the best parameters for a linear SVM classifier using half training and half testing dataset to get SVM scores and convert them to probabilities. 90 support vectors (4.5% training data) are obtained. The *ACC*, *REC*, *PRE*, and *F1* for training and testing data are given in Table 7.3.

Table 7.3. The ACC, REC, PRE, and F1 for training and testing entire synthetic NGD dataset.

	ACC	REC	PRE	F1
training	0.983	0.982	0.983	0.983
testing	0.973	0.960	0.986	0.973

Figure 7.6 shows a plot of the class-conditional densities $p(f|y=\pm 1)$ for the linear SVM trained on the data set. The solid red line is $p(f|y=+1)$, while the dashed blue line is $p(f|y=-1)$. Notice that these histograms are non-Gaussian distributed.

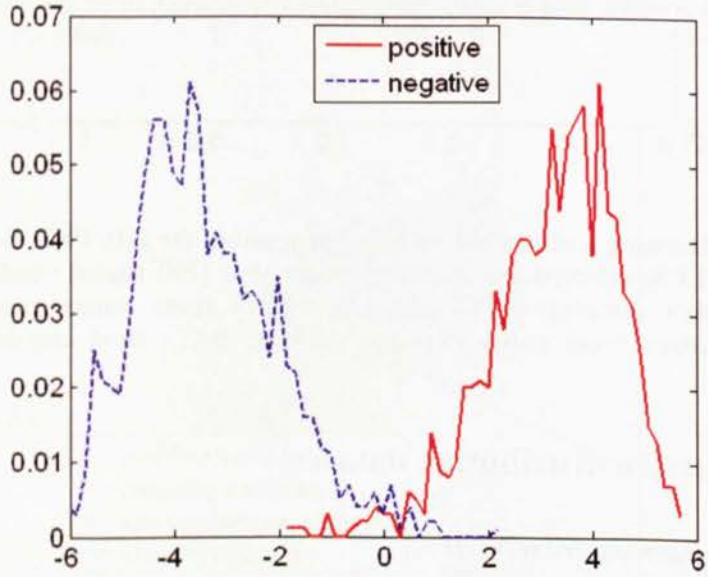


Figure 7.6. The histogram for $p(f|y=\pm 1)$ for SVM on training synthetic NGD dataset.

Figure 7.7 shows the histogram of probability estimation from DBW and SIDBW. The MSE of probability estimation using DBW and SIDBW from SVM scores of training dataset are 0.0164 and 0.0032, respectively. Obviously, SIDBW improve the accuracy of the probability estimation.

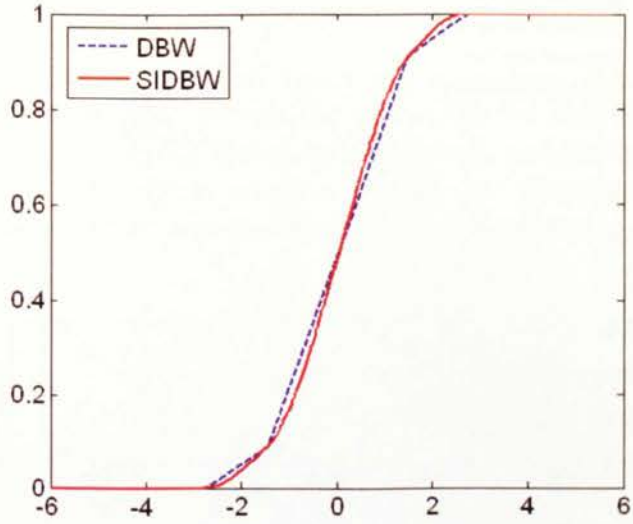


Figure 7.7. Histograms of probabilities of NGD from DBW and SIDBW.

7.4.2.2. Active learning from NGD with known calibrated probability estimation

For the query function (7.3), the threshold T_1=0.2 and T_2=0.85 for the experiment. 96 low confidence samples (4.8% of training data set) are obtained. 85% support vectors were included in the set of low confidence samples. Only the low confidence samples are used to train classifier, and the classifier to be used to classify the entire training and testing data. Table 7.4 gives the results of *ACC*, *REC*, *PRE*, and *F1*.

Table 7.4. The ACC, REC, PRE, and F1 for training and testing entire synthetic NGD dataset only use low confidence samples to train the classifier.

	ACC	REC	PRE	F1
training	0.984	0.983	0.984	0.984
testing	0.973	0.960	0.986	0.973

Comparing Table 7.3 and Table 7.4, it shows that only 4.8% low confidence samples are used to achieve better results for a non-Gaussian distribution dataset.

Figure 7.8 shows the passive learning results of 2000 samples from two non-Gaussian distribution classes dataset, all the data set has been used to train the classifier. Figure 7.9 shows the active learning results of the same dataset, but using only 96 low confidence samples.

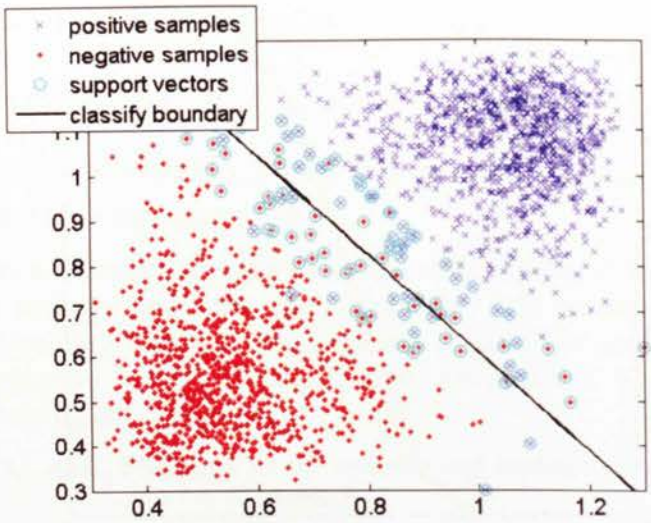


Figure 7.8. Passive learning SVM results of two NGDs, 2000 samples have been used to train the classifier.

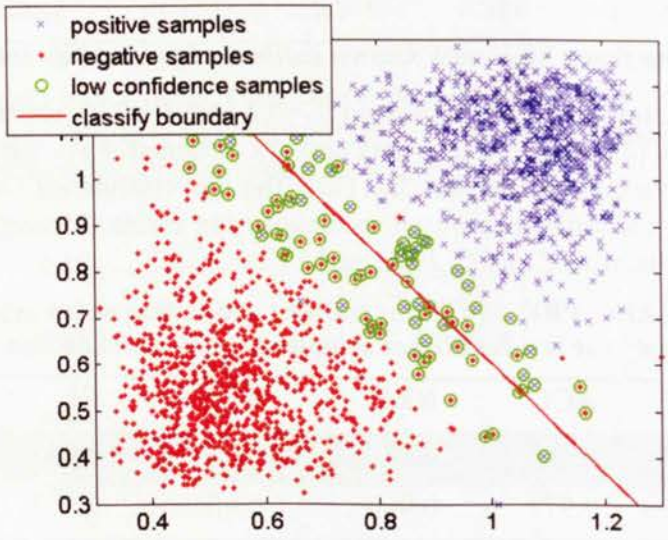


Figure 7.9. Active learning SVM results of two NGD, 96 low confidence samples have been used to train the classifier.

7.4.2.3. Active learning from NGD with unknown probability distribution

20 samples are randomly chosen as an initial training set to train the linear SVM classifier. The ACC of the entire NGD training data set is 97.85%. Obviously, this is not as good as the ACC from all the 2000 training samples which is 98.25%. After converting the SVM scores to probabilities, 81 low confidence (the most informative) samples are selected from remaining 1980 unlabeled samples. In the second round, 40 samples (the 20 initial samples and 20 new random selected low confidence samples) are used to train the linear SVM classifier. Then 15 low confidence samples are selected from the remaining 1960 unlabeled samples. In the third round a total of 55 samples (40 previous samples plus 15 new samples) are used to train the classifier. The final ACC of the training data set increase to

98.55%. It is better than that of the results from all 2000 samples. Bear in mind, the proportion of the training samples is just only 2.75%.

Figure 7.10 illustrates the processing of active learning for NGD data set if the probability distribution is unknown. From the plot it can be seen that the active learning algorithm can incrementally choose the most informative samples and correct the training error from the previous incomplete training. Finally the classification boundary would converge to the position which creates high classification accuracy.

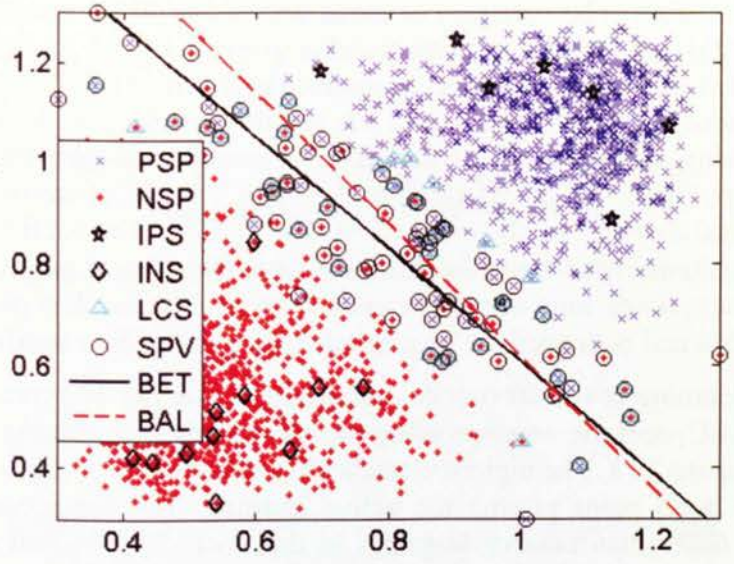


Figure 7.10. Active learning and passive learning processing for NGD. PSP: positive samples; NSP: negative samples; LCS: selected low confidence samples; IPS: Initial random positive samples; INS: initial negative samples; SPV: support vectors from entire training dataset; BET: classification boundary from entire training dataset; BAL: final classification boundary of active learning.

7.4.3. Real dataset

Foreground blobs of vehicles were obtained using AutoVDS described in the chapter 6. In order to test the active learning algorithm, the detected vehicles have been manually labeled into four categories: car, van (van, minivan, minibus and limousine), bus (single and double decked) and motorcycle (motorcycle and bicycle). The database comprises 2130 vehicles (car: 1500, van: 530, bus: 180 and motorcycle: 100). High dimensional (13+189=202 dimensions) vector of MBF+IPHOG was chosen as the vehicle observation vector. Figure 7.11 plots the first three most important PCA components of the feature vector for the manually-labeled data. The figure illustrate that car vs. van is very challenging to separate, whilst three binary class classifiers (motorcycle vs. car, car vs. van and van vs. bus) are sufficient to solve the 4-class classification problem.

For multi-class passive learning, the entire data set has been used to train the SVM system, one-vs-one method has been used, taking 20.43 hours on a 2.39GHz Pentium laptop. The number of support vectors is 441. The ratio of the number of support vectors over the sample size is 22.05%. This ratio reflects the classification complexity of the training dataset. The ACC is 97.51%.

For multi-class active learning, firstly three binary classifiers are trained separately to calibrate their SVM scores to probabilities. For each classifier, 50 samples are randomly selected to form the initial training dataset. In each round of query, a maximum 25 additional most informative samples are selected from the unlabeled sample pool according to the probability (if more than 25 most informative samples are obtained from the query function (7.3), randomly select 25 samples from them), and added into the training set. Using these most informative samples selection training is repeated until reasonable classification accuracy is achieved, or for a maximum of 10 rounds. Then a multiple classifier is trained using the selected samples sequentially. In order to test the stability of AL for real data, the program was run 10 times. The variation of mean and *std* of ACC, and the corresponding average number of training samples of each round is given in Figure 7.12. The figure shows that when the number of training samples increases, the mean of ACC increases and *std* of ACC decreases gradually, which means that the stability of AL increases accordingly. In round 10, 440 training samples (290 of the most informative samples plus 150 randomly selected initial samples) are selected to train the classifier. The classification model is used to classify the whole real dataset, and the mean of ACC is 97.57%. The confidence-based active learning procedure is terminated since the ACC is higher than that of passive learning which is 97.51%. The mean training time is just 31 minutes, 40 times faster than passive learning. In addition, only 20.66% real data needs to be annotated for training the classifier.

If the active learning is run 20 rounds, the program was run 10 times, the variation of mean and *std* of ACC, and the corresponding average number of training samples of each round is given in Figure 7.13. The highest classification accuracy of 97.95% is obtained at the 14th round. It is a good point to stop the active learning. The mean training time is 72 minutes, 17 times faster than passive learning. In this case, 24.65% real data needs to be annotated for training the classifier.

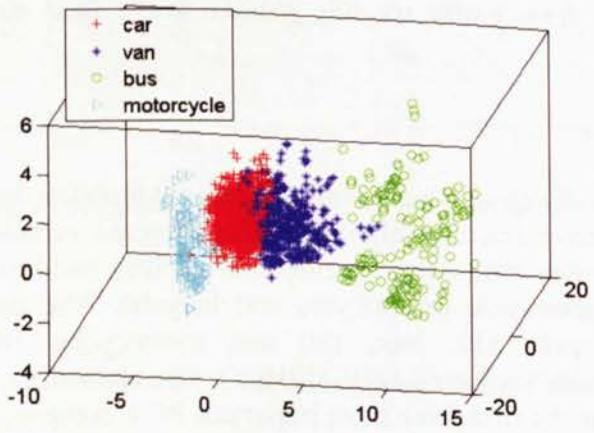


Figure 7.11. The three most important PCA components of MBF+IPHOG features.

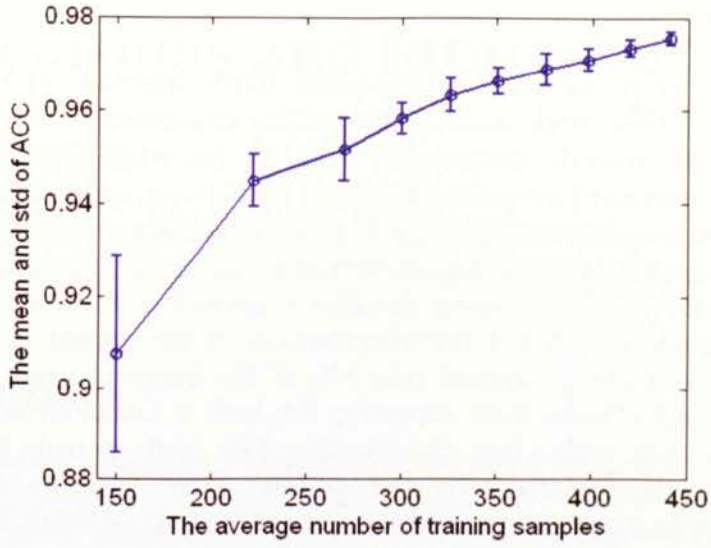


Figure 7.12. Variation of classification accuracy (10 rounds).

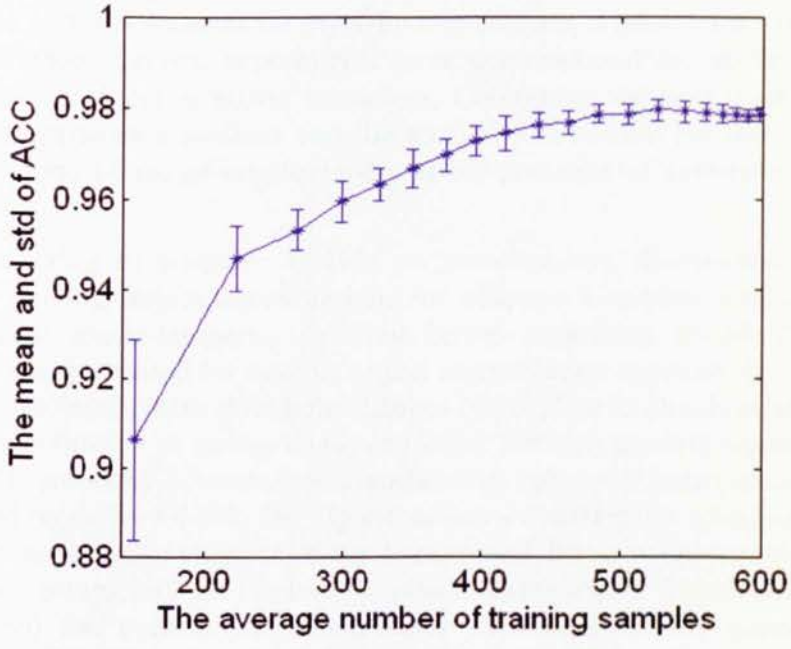


Figure 7.13. Variation of classification accuracy (20 rounds).

7.5. Summary

This chapter has proposed a confidence-based active learning approach for vehicle classification in urban traffic. High accuracy probability estimation is obtained from linear SVM scores using the smooth interpolated dynamic bin-width histogram. The most informative samples are selected for greater accuracy. In this way, only low confidence (most informative) samples are used to train the classifier. This can dramatically reduce the number of annotated samples required for training, as well as reducing the overall training time and classification complexity. Finally, a strong classifier is generated. Extensive experiments on synthetic and real data demonstrated the effectiveness of the approach. Compared with passive learning, active learning required only 5% of the training samples to achieve the comparable or improved classification accuracy, for both a Gaussian and non-Gaussian distributed dataset. For a multi-class classification task with a real, high-dimensional observation dataset, only 20.66% annotated samples were used to achieve superior classification results, with a computational improvement of some 40 times faster than that of using the entire dataset to train the classifier.

8. Conclusions and future work

8.1. Introduction

This is the final chapter of the thesis. Conclusions are given in the next Section, followed by suggestions for future work. The main achievements are listed in the final paragraph.

8.2. Conclusions

This dissertation addressed a framework for object detection, tracking and vehicle classification in an urban environment. A generalized active contour model for multi-channel and multi-phase colour image segmentation and an adaptive object-tracking algorithm have been proposed. By using level set methods, the mean shift tracking algorithm, a Chamfer distance transform (NCDT kernel) and sorted CSMs, objects can be detected and tracked. Their boundaries are not necessarily defined by a gradient or by very smooth boundaries, and hence classical active contour models are not applicable. The position of the initial curve can be anywhere in the image, and it does not necessarily surround the object to be detected. However, if the initial estimate is far from the true contour, it takes a long time to converge to the optimal solution. Several experiments have demonstrated the ability of the model to detect and track an object in movie sequences. Comparing the new method with the CVV method, the comparisons also show that the method proposed in the thesis is more accurate and robust in terms of image segmentation in the presence of symmetric and asymmetric noise.

Online learning of adaptive GMMs on nonstationary distributions is an important technique for moving object segmentation. An adaptive Gaussian mixture model using a multi-dimensional spatio-temporal Gaussian kernel smoothing transform for background modeling has been presented for moving object segmentation applications. The model update process can deal robustly with slow light changes (from clear to cloudy or vice versa), blurred images, camera vibration in strong wind, and other difficult environmental conditions, such as raining. The proposed solution has significantly enhanced segmentation results over a commonly used recursive GMM. The algorithm has a dynamically adaptive learning rate and models global illumination changes of the background frame by frame. At the cost of only one additional parameter per Gaussian, this modification dramatically improves the convergence and the accuracy of background subtraction whilst maintaining the same temporal adaptability. This is achieved by incorporating a modified adaptive schedule (a counter keeping track of the number of pixels that have contributed to a Gaussian component in temporal space) into a recursive filter. A comprehensive analysis of results in a wide range of environments and colour spaces are given. The system has been successfully used to segment and track objects in indoor and outdoor scene with both strong and light shadows, and highlight reflections and the system has been verified by rigorous evaluation.

A novel approach to camera calibration utilizes calibrated images mapped by Google Earth to provide an accurately-surveyed scene geometry that is manually corresponded with visible ground plane landmarks in the CCTV images. The effectiveness of state-of-the-art classification algorithms to categorise road vehicles for an urban traffic monitoring system

using simple measurement-based feature and a multi-shape descriptor is investigated. The analysis is applied to monocular video acquired from a static pole-mounted road side CCTV camera on a busy street. Manual vehicle segmentation was used to acquire a large (>2000 sample) database of labeled vehicles from which a set of measurement-based features (MBF) in combination with a pyramid of HOG (histogram of orientation gradients, both edge and intensity based) features are obtained. These are used to classify the objects into four main vehicle categories: car, van (van, minivan, minibus and limousine), bus (single and double decked) and motorcycle (motorcycle and bicycle). Results are presented for a number of experiments that were conducted to compare support vector machines (SVM) and random forests (RF) classifiers. 10-fold cross validation has been used to evaluate the performance of the classification methods. The results demonstrate that all methods achieve a recognition rate above 95% on the dataset, with SVM consistently outperforming RF. A combination of MBF and IPHOG features gave the best performance of 99.78%.

Acquisition of reliable vehicle counts and classification data is necessary to establish an enriched information platform and improve the quality of ITS. An automatic vehicle detection, tracking and classification system has been presented in this dissertation. There are four modules. In the background subtraction module, foreground extraction module and vehicle detection module, the improved background subtraction method is integrated to alleviate the negative impacts from camera vibration, shadow and reflection highlights, sudden illumination changes and gradual changes. A level set method has been used to refine the foreground blob. In the vehicle classification and counting module, a Kalman filter and SVM are integrated to improve accuracy. Extensive experiments have been undertaken, comparing 7 combinations of detection and classification methods. Results show that the best combination is to train the SVM using MBF+IPHOG features extracted by background subtraction, classifying the foreground blobs using a majority vote over 5 consecutive frames. The results demonstrate a vehicle detection rate of 96.39% and classification accuracy of 94.69% under varying illumination and weather conditions (from cloud to rain).

Though it is hard to compare with other methods, because there is no commonly used database for evaluation of ITS, the system proposed in the thesis is competitive with other algorithms. For example, Hasegawa and Kanade (2005) reported an overall classification accuracy of 91% for 6 object categories using linear discriminant analysis using 11-dimensional vector of image features, but poorer performance under illumination variation. Ji et al. (2007) take vehicle side-on views and use Gabor features and a minimum distance classifier to classify vehicles into five categories, with recognition rates of up to 95.17%. But the experiment database is constructed from manually segmented vehicles. Morris and Trivedi (2006a) classified vehicles into eight classes using simple morphological measurements of the detected blob, followed by linear discriminant analysis (LDA), fuzzy C-means clustering and a weighted k-nearest neighbour (wkNN) classifier. Results indicated an accuracy of up to 88%, though low confidence objects were rejected.

Finally, a confidence-based active learning approach has been proposed for vehicle classification in urban traffic. High accuracy probability estimation is obtained from linear SVM scores using the smooth interpolated dynamic bin-width histogram. The most informative samples are selected for greater accuracy. In this way, only low confidence (most informative) samples are used to train the classifier. This can dramatically reduce the number of annotated samples required for training, as well as reducing the overall training time and classification complexity. Finally, a strong classifier is generated. Extensive experiments on synthetic and real data demonstrated the effectiveness of the approach. Compared with

passive learning, active learning required only 5% of the training samples to achieve the comparable or improved classification accuracy, for both a Gaussian and non-Gaussian distributed dataset. For a multi-class classification task with a real, high-dimensional observation dataset, only 20.66% annotated samples were used to achieve superior classification results, with a computational improvement of some 40 times faster than that of using the entire dataset to train the classifier.

8.3. Future work

There is no perfect system. Background modeling and subtraction in itself is application oriented. Some of these problems cannot be solved simultaneously because of the differing needs associated with the semantic interpretation of the moving foreground and background. Only the basic pixel-level processes and frame-level global illumination changes have been discussed, excluding any additional pre- or post processing operations, for example, checking for spatial and temporal correlation, ghost removal, occlusion problems, etc. Furthermore, the performance should be evaluated over a wider range of conditions, including sequences captured during the evening and at night, and under heavy rain conditions. In order to identify vehicles that stop for long periods, e.g. at traffic lights, low level and high level processes should be combined. On the other hand, as traffic density increases and vehicles follow more closely, identifying individual vehicles becomes more difficult.

The appearance of a vehicle in the thesis depends on its pose and is affected by nearby objects. Pose independent classification and recognition is still an unsolved problem. For tracking cross a network of non-overlapping camera area, the key issue is the re-identification appearance model. Further work will be to extend the approach to operate in a view independent way and apply the best classifier design to assess its performance when applied to automatically segmented binary silhouettes of vehicles detected from video sequences.

In order to facilitate tracking vehicles as they move from one camera view to the next, the connectivity or topology of the camera network should be established. This can be learnt adaptively by correlating disappearance and re-appearance events between camera views (Ellis et al., 2003, 2005). Whilst a significant benefit of this method is that is correspondence-free, it can have difficulties with regular events typified by traffic streams. Having learnt the connectivity between cameras, when vehicles in the traffic stream transit between two non-overlapping camera views the order sequence may change according to various situations: vehicles may overtake; vehicles may not reappear, either because they turn off the observed network, or temporarily stop or park; similarly, previously unseen vehicles may appear in the traffic stream, having entered the stream from an unseen junction or region. The surveillance system can track vehicle through occlusion by learning probabilistic model of re-appearance using temporal and appearance-based feature.

Given that the vehicle pose with respect to a given CCTV road side camera view is strongly constrained; the vehicles are limited to move in particular directions (except abnormal events), further restricting the freedom associated with a change to each vehicle's pose; and the vehicles normally maintain an ordered sequence, and recognition of individual vehicles can be supported by joint consideration of neighbouring one. These three different kinds of information can be combined using Bayesian network inferences to establish a tracking system. In this case, the Bayesian prior will represent information accumulated from previous observations of routing and transit times, whilst the matching of appearance will provide specific observations to be used in updating.

8.4. Achievements

The work presented in this thesis has following achievements:

1. An active segmentation algorithm using multi-phase colour model and level set method.
2. An adaptive object tracking algorithm using active segmentation and mean shift tracking algorithms
3. An adaptive Gaussian mixture model with a multi-dimensional Gaussian kernel spatio-temporal smoothing transform.
4. An algorithm with a dynamically adaptive learning rate and a model for global illumination sudden change.
5. A novel approach to camera calibration utilizes calibrated images mapped by Google Earth.
6. A comparison of methods for vehicle types classification using simple measurement feature and PHOG feature.
7. A hybrid automatic vehicle detection, tracking and classification system for traffic flow analysis.
8. A frame work of confidence based active learning for vehicle classification in an urban traffic environment.

Bibliography

1. Aghagolzadeh, A., Seyedarabi, M.H., 2010. Vehicle tracking in multi-sensor networks by fusing data in particle filter framework. *IEEE Asia Pacific Conference on Circuits and Systems*, pp. 96-99.
2. Allwein, E.L., Schapire, R.E. and Singer, Y., 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, vol.1, pp. 113-141.
3. Ambardekar, A., Nicolescu, M., Bebis, G., 2008. Efficient vehicle tracking and classification for an automated traffic surveillance system. *Signal and Image Processing.*
4. Amit, Y., Geman, D., 1997. Shape quantization and recognition with randomized trees. *Neural Computation,* **9** (7): 1545–1588
5. Anthony, G., Gregg, H. and Tshilidzi, M., 2007. Image classification using SVMs: one-against-one vs one-against-all. *Proceedings of the 28th Asian Conference on Remote Sensing.*
6. Avery, R.P., Wang, Y., Rutherford, G.S., 2004. Length-based vehicle classification using images from uncalibrated video cameras. In: *Proceedings of the 7th International IEEE Conference on Intelligent Transportation System*, 737-742.
7. Avidan, S., 2007. Ensemble tracking. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 29(2)*, 261-271.
8. Bae, E. and Tai, X.-C., 2008. Graph cuts for the multiphase mumford-shah model using piecewise constant level set methods. *UCLA CAM Report 08-36.*
9. Baek, N., Park, S.-M., Kim, K.-J., Park, S.-B., 2007. Vehicle color classification based on the support vector machine method. *ICIC 2007, CCIS 2*, pp. 1133-1139.
10. Bai, Y., Guo, L., Jin, L., Huang, Q., 2009. A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition. *In: Proceedings of IEEE International Conference on Image Processing*, 3305-3308.
11. Bailer, W., Schallauer, P., Haraldsson, H.B., Rehatschek, H., 2005. Optimized mean shift algorithm for solor segmentation in image sequences. *SPIE Proceedings*, Vol 5685. Pp. 522-529.
12. Bazzani, L., Bloisi, D., Murino, V., 2009. A comparison of multi-hpothesis Kalman filter and particle filter for multi-target tracking. In: *Performance Evaluation of Tracking and Surveillance Workshop at CVPR 2009*, pp. 47-54.
13. Beiderman, Y., Rivlin, E., Teicher, M. and Zalevsky, Z., 2010. Illumination insensitive reconstruction and pattern recognition using spectral manipulation and K-factor spatial transforming. *Recent Patents on Signal Processing*, 2, 22-27.
14. Belongie, S., Malik, J. and Puzicha, J., 2002. Shape matching and object recognition using shape context. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509-522.

15. Bertini, R.L., Monsere, C.M., Yin, T., 2005. Benefit of intelligent transportation systems technologies in urban areas: A literature review. Porland State University, Technique Report.

16. Bertozzi, M., Broggi, A., Castelluccio, S., 1997. A real-time oriented system for vehicle detection. *Journal of System Architecture*, 43, 317–325.

17. Besag, J., 1986. On the statistical analysis of dirty pictures. *J. Roy. Statist. Soc., Ser. B.*, 48(3):259–302.

18. Bhattacharyya, A., 1943. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematics Society*, 35, pp99-110.

19. Bhonsle, S., Trivedi, M. and Gupta, A., 2000. Database-centered architecture for traffic incident detection, management, and analysis. In *Proc. IEEE Conference on Intelligent Transport. System*, Dearborn, Michigan, pp. 149–154.

20. Birchfield, S. and Rangarajan, S., 2005. Spatiograms versus histograms for region-based tracking. *CVPR, 2*, 1158-1163.

21. Blake, A. and Zisserman, A., 1987. *Visual Reconstruction.* MIT Press.

22. Blanz, V., Scholkopf, B., Bulthoff, H., Burges, C., Vapnik, V. and Vetter, T., 1996. Comparison of view-based object recognition algorithms using realistic 3D models. In *Artificial Neural Networks* – ICANN'96, pp. 251-256, Berlin, Springer Lecture notes in Computer Science, Vol. 1112.

23. Bloisi, D. and Iocchi, L., 2009. Argos-A video surveillance system for boat traffic monitoring in Venice. In *Proc. IJPRAI*, pp. 1477–1502.

24. Bonenfant, A., Chen, Z., Hammond, K., Michaelson, G., Wallace, A.M. and Wallace, I., 2007. Towards resource-certified software: a formal cost model for time and its application to an image-processing example. *ACM Symposium on Applied Computing (SAC '07), Seoul, Korea,* Seoul, Korea, 11-15.

25. Bosch, A., Zisserman, A., Munoz, X., 2007a. Image classification using random forests and ferns. In *Proceedings of the IEEE 11th International Conference on Computer Vision*, 1-8.

26. Bosch, A., Zisserman, A., Munoz, X., 2007b. Representing shape with a spatial pyramid kernel. In: *Proceedings of the ACM International Conference on Image and Video Retrieval.*

27. Botev, Z.I.; Grotowski, J.F.; Kroese, D.P., 2010. Kernel density estimation via diffusion. *Annals of Statistics* **38** (5): 2916–2957.

28. Bouwmans, T., Baf, F.E., Vachon, B., 2008. Background modeling using mixture of Gaussian for foreground detection: A survey. *Recent Patents on Computer Science*, 1(3):219-237.

29. Boykov, Y., Funka Lea, G., 2006. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 69(2), 109–131.

30. Bradski, G.R., 1998. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal 2th quarter*, pp. 12-21.

31. Breiman, L., 2001. Random Forests. *Machine Learning* **45** (1): 5–32.

32. Breiman, L. and Cutler, A., http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#overview.

33. Brock, L.D., Schmidt, G.T., 1970. General Questions on Kalman Filtering in Navigation Systems, Chapter 10 of "*Theory and Applications of Kalman Filtering*" C.T. Leondes, Editor, NATO AGARD.

34. Brox, T., Weickert, J., 2004. Level set based image segmentation with multiple regions. Pattern Recognition, Lecture Note in Computer Science, Vol. 3175/2004, 415-423.

35. Buch, N., Cracknell, M., Orwell, J., Velastin, S.A., 2009b. Vehicle localisation and classification in urban CCTV streams. *16th World Congress and Exhibition on Intelligent Transport Systems and Services*, Stockholm, Sweden.

36. Buch, N., Orwell, J. and Velastin, S. A., 2008. Detection and classification of vehicles for urban traffic scenes. In *Proc. Int. Conf. VIE*, pp. 182–187.

37. Buch, N., Orwell, J. and Velastin, S. A., 2010. Urban road user detection and classification using 3-D wireframe models. *IET Comput. Vis.*, vol. 4, no. 2, pp. 105–116.

38. Buch, N., Orwell, J., Velastin, S.A., 2009a. 3D extended histogram of oriented gradients (3DHOG) for classification of road users in urban scenes. *British Machine Vision Conference.*

39. Buch, N., Velastin, A.V., and Orwell, J., 2011. A review of computer vision techniques for the analysis of urban traffic. *IEEE Transaction on Intelligent Transportation Systems*, 12(3), pp. 920-939.

40. Buch, N., Yin, F., Orwell, J., Makris, D., Velastin, S.A., 2009c. Urban vehicle tracking using a combined 3D model Detector and classifier. *13th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, KES2009.

41. Burges, C., 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery,* vol. 2, no. 2, pp. 121-167.

42. Cannons, K. and Wildes, R., 2007. Spatiotemporal oriented energy features for visual tracking. In *ACCV,* pp. 532-543.

43. Cannons, K., 2008. A review of visual tracking. *Technical Report CSE-2008-07*, York University.

44. Canny, J., 1986. A Computational Approach To Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986

45. Canu, S., Grandvalet, Y., Guigue, V. and Rakotomamonjy, A., 2005. SVM and Kernel Methods Matlab Toolbox, Perception Systèmes et Information, INSA de Rouen, Rouen, France. Source code available at http://asi.insa-rouen.fr/enseignants/~arakoto/toolbox/index.html

46. Cao, X., Wu, C., Yan, P., Li, X., 2011. Linear AVM using boosting HOG feature for vehicle detection in low-altitude airborne videos. *18th IEEE International Conference on Image Processing*, 2469-2472.

47. Cao, R.; Cuevas, A.; Manteiga, W. G., 1994. A comparative study of several smoothing methods in density estimation. *Computational Statistics and Data Analysis* **17**: 153–176.

48. Caruana, R.; Karampatziakis, N., Yessenalina, A., 2008. An empirical evaluation of supervised learning in high dimensions. *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 96-103.

49. Caselles, V., Kimmel, R. and Sapiro, G., 1997. Geodesic active contours. *International Journal of Computer Vision*, Vol. 22 (1), pp. 61–79.

50. Caselles, V., Kimmel, R. and Sapiro, G., 1995. Geodesic active contours. In *Fifth Interna- tional Conference on Digital Object Identifier*, pp. 694-699.

51. Cha, S.H. and Srihari, S.N., 2002. On measuring the distance between histograms. *Pattern Recognition*, 35, 1355-1370.

52. Chan, T., Sandberg, B.Y. and Vese, L., 2000. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, No 11, p. 130-141.

53. Chan, T.F. and Vese, L.A., 2001. Active contours without edges. *IEEE Trans. Image Processing*, 10(2):266–277.

54. Chang, C.C.; Lin, C.J., 2011. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

55. Chang, J.S., Kim, E.Y., Jung, K., Kim, H.j., 2005. Object tracking using mean shift and active contours. *Proceedings of the 18th international conference on Innovations in Applied Artificial Intelligence*, Lecture Notes In Computer Science, Vol. 3533, 26-35.

56. Chang, C.-C. and Lin, C.-J., 2001. LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

57. Chang, R., Gandhi, T. and Trivedi, M.M., 2004. Vision modules for a multisensory bridge monitoring approach. In *Proc. IEEE Conf. on Intelligent. Transport Sysem.*, pp. 971–976.

58. Chapelle, O., Haffner P. and Vapnik, V., 1999. Support vector machines for histogram-based image classification. *IEEE Transaction on Neural newworks*, Vol. 10, No. 5, pp. 1055-1064.

59. Chen, X. and Zhang, C.C., 2007. Vehicle classification from traffic surveillance videos at a finer granularity. In *Advances in Multimedia Modeling*. Berlin, Germany: Springer-Verlag, pp. 772–781.

60. Chen, X., Zhou, Y., Huang, X., Li, C., 2008. Adaptive bandwith mean shift object tracking. *IEEE International Conference on Robotics*, Automation and Mechatronics, 1011-1017.

61. Chen, Y., Rui, Y. and Huang T., 2001. JPDAF-based HMM for real-time contour tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawaii, volume I, pp. 543-550.

62. Chen, Z. and Wallace A.M., 2007a. Improved object tracking using an adaptive colour

model. *The 6th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, EZhou Hubei, China, pp. 280-294.

63. Chen, Z., and Wallace, A.M., 2007b. Active segmentation and adaptive tracking using level sets. *British Machine Vision Conference*, University of Warwick, pp. 920-929.

64. Chen, Z., Husz, Z.L., Wallace I. and Wallace, A.W., 2007c. Video object tracking based on a Chamfer distance transform. *IEEE International Conference on Image Processing*, San Antonio, Texas, USA, III 357-360.

65. Chen, Z., Pears, N., Freeman, M. and Austin, J., 2009a. Background subtraction in video using recursive mixture models, spatio-temporal filtering and shadow removal. *Proceedings of 5th International symposium on Visual Computing (ISVC)*, Las Vegas, NV, USA, Nov. 30-Dec. 2. Lecture Notes in Computer Science, Vol. 5876, pp. 1141-1150.

66. Chen, Z., Pears, N.E., Freeman, M. and Austin, J., 2009b. Road vehicle classification using support vector machines. In *Proceedings of IEEE International Conference on Intelligent Computing and Intelligent System*, Shanghai, China, 214-218, 2009.

67. Cheng, J. and Yang, J., 2004. Real-time infrared object tracking based on mean shift. Progress in Pattern Recognition, Image Analysis and Applications, LNCS, vol. 3287, 530-542.

68. Cheng, Y.Z., 1995. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), pp790-799.

69. Cheung, S., and Kamath, C., 2004. Robust techniques for background subtraction in urban traffic video. *Visual Communication and Image Processing*, Proc. SPIE 5308, No. 1. 881-892.

70. Cho. J.U., Jin, S.H., Pham, X.D. et al., 2006. A real-time object tracking system using a particle filter. *International Conference on Intelligent Robotics and Systems*, pp. 2822-2827.

71. Chockalingam, P., Pradeep, N., Birchfield, S., 2009. Adaptive fragments-based tracking of non-rigid objects using level sets. *IEEE 12th International Conference on Computer Vision*, 1530-1537.

72. Chung, G. and Vese, L.A., 2009. Image segmentation using a multilayer level-set approach. *Computing and Visualization in Science*, 12(6), 267-285.

73. Cohen, I. and Medioni, G., 1999. Detecting and tracking moving objects in video surveillance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages II: 319--325.

74. Coifman, B., Beymer, D., McLauchlan, P., Malik, J., 1998. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C 6*, 271-188.

75. Collins, R.T., Lipton, A.J. and Kanade, T., 1999. A system for video surveillance and monitoring. In *Proc. American Nuclear Society on the International Topical Meeting on Robotics and Remote Systems*, Pittsburgh, PA, pages 1-15.

76. Collins, R.T., Liu Y. and Leordeanu, M., 2005. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, 1631-1643.

77. Comaniciu, D. and Meer, P., 1997. Robust analysis of feature spaces: color image segmentation. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp. 750-755.

78. Comaniciu, D. and Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 24(5), pp. 603–619.

79. Comaniciu, D., Ramesh, V. and Meer, P., 2003. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, 564-577.

80. Comaniciu, D., Ramesh, V. and Meer, P., 2000. Real-time tracking of non-rigid objects using mean shift. *IEEE Conf. on Computer Vision and Pattern Recognition*, pp.142-149

81. Cortes, C. and Vapnik, V., 1995. Support vector network. *Machine Learning*, vol. 20, pp. 1-25.

82. Cremers, D., Osher, S.J. and Soatto, S., 2004. Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk. In *C. E. Rasmussen, editor, Pattern Recognition*, volume 3175 of LNCS, pages 36–44.

83. Cremers, D., Rousson, M. and Deriche, R., 2007. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, Vol. 72 (2), 195-215.

84. Creusen, I.M., Wijnhoven, R. J. G. De, P. H. N. and Eindhoven, V. B. V., 2009. Applying feature selection techniques for visual dictionary creation in object classification. In *Proc. Int. Conf. IPCV Pattern Recog.*, pp. 722-727.

85. Cucchiara, R., Grana, C., Neri, G., Piccardi, M., Prati, A., 2001. The sakbot system for moving object detection and tracking. In: *Video-based Surveillance Systems - Computer Vision and Distributed Processing*, chapter 12. Kluwer Academic, pp. 145-157.

86. Cucchiara, R., Piccardi, M., Prati, A., 2003. Detecting moving objects, ghosts and shadows in video streams. *IEEE Transaction on Pattern Analysis and Machine Intelligence*. 25(10), pp. 1337-1342.

87. Dalal, N. and Triggs, B., 2005. Histograms of oriented gradients for human detection. In *Proc. IEEE Comput. Soc. Conf. CVPR*, vol. 1, pp. 886-893.

88. Dalka, P., Czyzewski, A., 2010. Vehicle classification based on soft computing algorithms. *Rough Sets and Current Trends in Computing*, LNCS 6086, 70-79.

89. DeCarlo, D. and Metaxas, D., 2000. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, Vol. 38, No. 2, 99-127.

90. Dellaert, F., Thorpe, C., 1997. Robust car tracking using Kalman filtering and Bayesian templates. In *Proceedings of Conference on Intelligent Transportation Systems*.

91. DeMenthon, D., 2002. Spatio-temporal segmentation of video by hierarchical mean shift analysis. In *Proc. Statistical Methods in Video Processing Workshop*, Copenhagen, Denmark, 2002. Also CAR-TR-978 Center for Automat. Res., U. of Md, College Park.

92. DeMenthon, D., Megret, R., 2000. Spatio-temporal segmentation of video by hierarchical mean shift analysis. *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 142-151.

93. Dixon, M., Jacobs, N., Pless, R., 2009. An efficient system for vehicle tracking in multi-camera networks. In *Proc. IEEE/ACM Int. Conf. Distributed Smart Cameras*, Como, Italy, pp. 1–8.

94. Doermann, D. and Mihalcik, D., 2000. Tools and techniques for video performances evaluation. *ICPR*, pages 167-170 (http://viper-toolkit.sourceforge.net/).

95. Drish, J., 2001. Obtaining calibrated probability estimates from support vector machines. *Technique Report, Department of Computer Science and Engineering*, University of California, San Diego, CA, 2001.

96. Du, Y., Yuan, F., 2009. Real-time vehicle tracking by Kalman filtering and Gabor decomposition. *1st International Conference on Information Science and Engineering (ICISE'09)*, 1386-1390.

97. Elgammal, A.M., Harwood, A., Davis, L.S., 2000. Non-parametric model for background subtraction. Lecture Notes in Computer Science, vol. 1843, *Proceedings of the 6th European Conference on Computer Vision-Part II*, 751-767.

98. Elhabian, S.Y., El-Sayed, K.M. and Ahmed, A.H., 2008. Moving object detection in spatial domain using background removal techniques-state-of-art. *Recent patents on computer science*, 1(1):32-54.

99. Ellis, T.J., Black, J., Xu, M., Makris, D., 2005. A distributed multicamera surveillance system. in 'Ambient Intelligence – A Novel Paradigm', Springer.

100. Ellis, T.J., Makris, D., Black, J.K., Xu, M., 2003. Learning a multi-camera topology. Joint IEEE Int. Workshop on Visual Surveillance and performance Evaluation of Tracking and Surveillance, Nice, October, pp. 165-171.

101. Epanechnikov, V.A., 1969. Non-parametric estimation of a multivariate probability density. *Theory of Probability and its Applications* **14**: 153–158.

102. Esther, H., Dennis, M., Bastian, L., 2010. Geometrically constrained level set tracking for automotive application. *Proceedings of the 32nd DAGM Conference on Pattern Recognition*, 472-482.

103. Fan, Z., Yang, M. and Wu, Y., 2007. Multiple collaborative kernel tracking. *PAMI, 29(7):* 1268-1273.

104. Fang, W., Chan, K.L., 2007. Incorporating shape prior into geodesic active contours for detecting partially occluded object. *Pattern Recognition*, 40(8): 2163-2172.

105. Fathy, M., Siyal, M.Y., 1995. An image detection technique based on morphological edge detection and background differencing for realtime traffic analysis. *Pattern Recognition Letters*, 16, 1321–1330.

106. Ferecatu, M. and Sahbi, H., 2009. multi-view object matching and tracking using canonical correlation analysis. 16th IEEE International Conference on Image Processing (ICIP), pp 2109-2112.

107. Feris, R., Siddiquie, B., Zhai, Y., Petterson, J., Brown, L., Pankanti, S., 2011. Attribute-based vehicle sear in crowed surveillance videos. *In proceedings of 1st ACM ICMR.*

108. Finlayson, G., Hordley, S., Drew, M., 2002. Removing shadows from images. In: *European Conference on Computer Vision*, Lecture Notes in Computer Science Vol. 2353, pp. 4:823-836.

109. Freund, Y. and Schapire, R., 1996. Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, 148–156.

110. Funk, N., 2003. A study of the Kalman filter applied to visual tracking. *University of Alberta, Project for CMPUT 652*, Dec. 7, 2003.

111. Furnkranz, J., 2002. Round robin classification. *Journal of Machine Learning Research*, vol. 2 pp. 721-747.

112. Fussenegger, M., Deriche, R., Pinz, A., 2006. Multiregion level set tracking with transformation invariant shape priors. *ACCV'2006*, LNCS 3851, 674-683.

113. Gao, T. Liu, Z.G., Gao, W.C. and Zhang, J., 2009. Moving vehicle tracking based on SIFT active particle choosing. In *Proc. Adv. Neuro-Inf. Process.*, vol. 5507, Lect. Notes Comput. Sci., pp. 695–702.

114. Geiger, D., Gupta, A., Costa L.A., Vlontzos J., 1995. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 3, 294-302.

115. Geman, S. and Geman, D., 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. On Patt. Anal. and Mach. Intell.*, 6(6):721–741.

116. Georg, M., Pless, R., 2009. Fitting parametric road models to spatio-temporal derivatives. In: *proceedings of the 12th IEEE International Conference on Computer vision Workshops*, 421-427.

117. Girosi, F., 1997. An equivalence between sparse approximation and support vector machines. *Technical Report: AIM-1606. Massachusetts Institute of Technology*, Cambridge, USA.

118. Goldenberg, R., Kimmel, R., Rivlin, E. and Rudzsky, M., 2001. Fast geodesic active contours. *IEEE Transaction on Image Processing*, Vol. 10, N0. 10, Oct. pp. 1467-1475.

119. Greggio, N., Bernardino, A., Laschi, C., Dario, P., Santos-Victor, J., 2010. Self-adaptive Gaussian mixture models for real-time video segmentation and background subtraction. *10th International Conference on Intelligent Systems Design and Applications (ISDA)*, pp.983-989.

120. Gupte, S., Masoud, O., Martin, R.F.K., Papanikolopoulos, N.P., 2002. Detection and classification of vehicles. *IEEE Transaction on intelligent Transportation Systems*, 3(1): 37-47.

121. Han, B., Comaniciu, D., Davis, L., 2008. Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 30(7), 1186-1197.

122. Han, X., Chen, Y., Ruan, X., 2010. Image recognition by learned linear subspace of combined bag-of-features and low-level features. In: *Proceedings of IEEE International Conference on Image Processing*, 1049-1052.

123. Haque, M., Murshed, M. and Paul, M., 2008. Improved Gaussian mixtures for robust object detection by adaptive multi-background generation. In: *Proceedings of 19th International Conference on Pattern Recognition*, 1-4.

124. Harb, R., Yan, X., Radwan, E., Su, X., 2009. Exploring precrash maneuver classification threes and random forests. *Accident Analysis & Prevention*. 41(1): 98-107.

125. Hasegawa, O., Kanade, T., 2005. Type classification, color estimation, and specific target detection of moving targets on public streets. *Machine Vision Applications*, 16: 116-121.

126. Heikkilä, J., 2000. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10): 1066-1077.

127. Herodotou, N., Plataniotis, K.N., Venetsanopoulos, A.N., 1998. A color segmentation scheme for object-based video coding. In: *Proc. IEEE Symp. Advances in Digital Filtering and Signal Processing*, pp. 25-29.

128. Ho, T.K., 1995. Random decision forest. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, pp.278–282.

129. Ho, T.K., 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20** (8): 832–844.

130. Hoose, N., 1992. IMPACT: an image analysis tool for motorway analysis and surveillance. *Traffic Engineering Control Journal*, 140–147.

131. Horprasert, T., Harwood, D., Davis, L.S., 1999. A statistical approach for real-time robust background subtraction and shadow detection. *Proceedings of IEEE ICCV'99 Frame rate workshop*, pp. 1-19.

132. Hsieh, J.W., Yu, S.H., Chen, Y.S. and Hu, W.F., 2006. Automatic traffic surveillance system for vehicle tracking and classification. *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 175–187.

133. Hsu, C. and Lin, C., 2002. A comparison of methods for multi-class support vector machines. *IEEE Transaction on Neural Networks*, 13(2): 415-425, 2002.

134. Hsu, R.L., Abdel-Mottaleb, M. and Jain, A.K., 2002. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.24, No.5, pp 696-706.

135. Hu, W., Xiao, X., Xie, D., Tan, T. and Maybank, S., 2004, Traffic accident prediction using 3-d model-based vehicle tracking. *IEEE Transaction on Vehicle Technology*, vol. 53, no. 3, pp. 677–694.

136. Huang, C.L. and Liao, W.C., 2004. A vision-based vehicle identification system. In *Proc. 17th International Conference on Pattern Recognition*, vol. 4, pp. 364–367.

137. Islam, M.K., Jahan, F., Min, J.H., Baek, J.H., 2011. Object classification based on visual and extended features for video surveillance application. *8th Asian Control Conference (ASCC)*. 1398-1401.

138. Izadi, M., Saeedi, P., 2008. Robust region-based background subtraction and shadow removing using color and gradient information. *19th International Conference on Pattern Recognition (ICPR)*, 1-5.

139. Jacobs, N., Dixon, M., Satkin, S., Pless, R., 2009. Efficient tracking of many objects in structured environments. *12th IEEE International Conference on Computer Vision Workshops*, 1161-1168.

140. Javed, O., Shafique, K., Shah, M., 2002. A hierarchical approach to robust background subtraction using color and gradient information. *Workshop on Motion and Video Computing*, 22-27.

141. Jaward, M., Mihaylova, L., Canagarajah N. and Bull, D., 2006. Multiple objects tracking using particle filters. IEEE AC 2006.

142. Jehan-Besson, S., Barlaud, M. and Aubert, G., 2001. Video object segmentation using Eulerian region-based active contours. In: *International Conference on Digital Object Identifier,* pp. 353-360.

143. Jehan-Besson, S., Barlaud, M., Aubert, G., 2003a. DREAM2S: Deformable regions driven by an Eulerian accurate minimization method for image and video segmentation. *International Journal of Computer Vision*, 53(1): 45-70.

144. Jehan-Besson, S., Barlaud, M., Aubert, G., Faugeras, O., 2003b. Shape gradients for histogram segmentation using active contours. *Proceedings of the Ninth IEEE International Conference on Computer Vision*, Volume 1, pp. 408-415.

145. Ji, P., Jin, L., Li, X., 2007. Vision-based vehicle type classification using partial Gabor filter bank. In: *Proc IEEE Int Conf on Automation and Logistics*, China, pp.1037-1040.

146. Joachims, T., 2006. Training linear SVMs in linear time. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, 217-226, Philadelphia, Pennsylvania, USA.

147. Johansson, B., Wiklund, J., Forssén, P. and Granlund, G., 2009. Combining shadow detection and simulation for estimation of vehicle size and position. *Pattern Recognition Letters*, vol. 30, no. 8, pp. 751–759.

148. Jones, M.C.; Marron, J.S.; Sheather, S. J., 1996. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association* **91** (433): 401–407.

149. Jung, Y. M., Kang, S. H. and Shen, J., 2007. Multiphase image segmentation via Modica-Mortola phase transition. *SIAM applied Mathematics*, 67:1213–1232.

150. Jung, Y., Lee, K. and Ho,,Y., 2001. Content-based event retrieval using semantic scene interpretation for automated traffic surveillance. *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 3, pp. 151–163.

151. KaewTraKulPong, P. and Bowden, R., 2001. An improved adaptive background mixture model for real-time tracking with shadow detection. *Proc. of 2nd European workshop on Advanced Video Based Surveillance Systems*, chapter 11, 135-144.

152. Kaftan, J.N., Bell, A.A., Aach, T., 2008. Mean shift segmentation evaluation of optimization techniques. *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, 22-25.

153. Kailath T., 1967. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, Vol. 15, No.1, 52-60.

154. Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. *Trans. ASME-J. Basic Eng.*, ser. D, vol. 82, pp. 35–45.

155. Kanhere, N.K., Pundlik, S.J. and Birchfield, S.T., 2005. Vehicle segmentation and tracking from a low-angle off-axis camera. In: *Proc. IEEE Computer Vision and Pattern Recognition*, vol. 2, pp. 1152–1157.

156. Kanhere, N.K. and . Birchfield, S.T., 2008. Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features. *IEEE Transaction on Intelligent Transport Systems*, vol. 9, no. 1, pp. 148–160.

157. Kass, M., Witkin, A. and Terzopoulos, D., 1988. Snakes: active contour models. *International Journal of Computer Vision*, Vol. 1 (4), pp 321-331.

158. Khammari, A., Nashashibi, F., Abramson, Y. and Laurgeau, C., 2005. Vehicle detection combining gradient analysis and Adaboost classification. In: *Proc. IEEE Conf. Intell. Transp. Syst.*, pp. 66–71.

159. Khan, Z.H., Gu, I.Y., Backhouse, A.G., 2011. Robust vision object tracking using multi-mode anisotropic mean shift and particle filters. *IEEE transaction on Circuits and Systems for Video Technology*, 21(1): 74-87.

160. Khan, Z.H.,Gu, I.Y.H, Wang, T., Backhouse, A., 2009. Joint anisotropic mean shift and consensus point feature correspondences for object tracking in video. *IEEE International Conference on Multimedia and Expo*, 1270-1273.

161. Kogut, G.T., Trivedi, M.M., 2001. Maintaining the identity of multiple vehicles as they travel through a video network. In: *Proc. IEEE Conference on Intelligent Transport System, Oakland, California*, pp. 756-761.

162. Kolmogorov, V., Zabih, R., 2004. What energy function can be minimized via graph cuts? *IEEE Transaction on PAMI*, 26(2), 147–159.

163. Krishnaveni, S., Hemalatha, M., 2011. A perspective analysis of traffic accident using data mining techniques. *International Journal of Computer Applications*, 23(7): 40-48.

164. Kumar R., Sawhney, H., et al., 2001. Aerial video surveillance and exploitation. *Proceedings of the IEEE*, Vol. 89, No.10, 1518-1539.

165. Kumar, P., Mittal, A., Kumar, P., 2008. Study of robust an intelligent surveillance in visible and multimodal framework. *Informatica*, 32, 63-77.

166. Kumar, P., Ranganath, S. and Huang, W.M., 2003. Bayesian-network-based computer vision algorithm for traffic monitoring using video. In: *Proceedings of IEEE Intelligent Transportation Systems*, vol. 1, pp. 897–902.

167. Lanz, O., 2006. Approximate bayesian multibody tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(9):1436–1449.

168. Laura, A. and Rouslan, A. M., 2008, Support vector machines (SVM) as a technique for solvency analysis. *tttp://www.econstor.eu/bitstream/10419/27334/1/576821438.PDF.*

169. Lazebnik, S., Schmid, C. and Ponce, J. 2005. A sparse texture representation using local affine regions. *IEEE Transactions on PAMI*, 27(8): 1265-1278.

170. Lee, D-S., 2005. Effective Gaussian mixture learning for video background subtraction. IEEE Transaction on Pattern Analysis and Machine Intelligence, 27(5): 827-832.

171. Leotta, M.J., Mundy J.L., 2011. Vehicle surveillance with a generic, adaptive, 3D vehicle model. *IEEE Transation on Pattern Analysis and Machine Learning*, 33(7): 1457-1469.

172. Leshem G. and Ritov, Y., 2007. Traffic flow prediction using adaboost algorithm with random forests as a weak learner. In: *Proceedings of the International Conference on Computer, Information, and Systems Science, and Engineering.*

173. Leung, A. and Gong, S. 2006. Mean-shift tracking with random sampling. *BMVC, 2,* 729-738.

174. Lewicki, P., Hill, T., 2007. Statistics: methods and applications. Tulsa: StatSoft.

175. Lewis, D.D. and Catlett, J., 1994. Heterogeneous uncertainty sampling for supervised learning. *Proc. 11th Int'l Conf. Machine Learning*, pp. 148-156.

176. Leymarie, F. and Levine, M.D., 1993. Tracking deformable objects in the plane using an active contour model. *PAMI, 15(6)*, 617-634.

177. Li, H. and Tai, X.-C., 2007. Piecewise constant level set methods for multiphase motion. *International J. Numer. Anal. Modelling*, 4(2):291–305.

178. Li, K., Miller, E.D., Weiss, L.E., Campbell, P.G. and Kanade, T., 2006. Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy. In: *Workshop on Mathematical Methods in Biomedical Image Analysis,* pp. 65-72.

179. Li, M. and Sethi, I.K., 2006. Confidence-based active learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1251–1261.

180. Li, P., Xiao L., 2009. Mean shift parallel tracking on GPU. *Pattern Recognition and Image Analysis*, LNCS, vol 5524, 120-127.

181. Liao, S., Zhao, G., Kellokumpu, V., Pietikainen, M., Li., S.Z., 2010. Modeling pixel process with scale invariant local patterns for background subtraction in complex scene. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1301-1306.

182. Lie, J., Lysaker, M. and Tai, X.-C., 2006. A variant of the level set method and applications to image segmentation. *AMS Mathematics of Computation*, 75:1155-1174.

183. Lipton, A.J., Fujiyoshi, H. and Patil, R.S., 1998. Moving target classification and tracking from real-time video. *4th IEEE Workshop on Applications of Computer Vision*, Princeton, 8–14.

184. Lowe, D.G., 2004. Distinctive image features from scaleinvariant keypoints. *International Journal of Computer Vision*, 60, 2, pp. 91-110.

185. Ma, X. and Grimson, W.E.L., 2005. Edge-based rich representation for vehicle classification. In *Proc10th IEEE International Conference on Computer Vision*, vol. 2, pp. 1185–1192.

186. Maddalena, L. and Petrosino, A., 2008. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image processing* 17(7):1168-1177.

187. Mangasarian, O.L., 1965. Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13:pp. 444-452.

188. Mangasarian, O.L., 1968. Muti-surface method of pattern separation. *IEEE Transactions on Information Theory IT-14*, pp. 801-807.

189. Mangasarian, O.L., 1969. Nonlinear Programming. *McGraw-Hill*, New York.

190. Mansouri, A.R., 2002. Region tracking via level set PDEs without motion correspondence. *PAMI, 24(7)*, 947-961.

191. Marslin, R., Sullivan, G.D., Baker, K.D., 1991. Kalman filters in constrained model based tracking. *Proceedings British Machine Vision Conference*, Glasgow, Scotland (1991), pp. 371–374.

192. Martel-Brisson, N. and Zaccarin, A., 2007. Learning and removing cast shadows through a multidistribution approach. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29(7):1133-1146.

193. McKenna, S.J. Raja, Y. and Gong S., 1998. Object tracking using adaptive colour mixture models. In: *R. Chin and T.-C. Pong, editors, Third Asian Conference on Computer Vision*, Hong Kong, China, number 1, pages 615--622

194. Melgani, F., Bruzzone, L., 2004. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transaction on Geoscience and Remote Sensing*, 42, pp. 1778-1790.

195. Messelodi, S., Modena, C.M., Zanin, M., 2005. A computer vision system for the detection and classification of vehicles at urban road intersections. *Journal of Pattern Analysis & Applications*, 8(1), 17-31.

196. Michaelson, G., Wallace, A.M., Hammond, K., Bonenfant, A., Chen, Z. and Gorry, B., 2007. Analysing and deploying resource-bound AV software in Hume. *Technology Centre (SEAS DTC), Annual Technical Conference, Edinburgh,* pp. A2.

197. Mikic, I., Cosman, P.C., Kogut, G.T., 2000. Trivedi, M.M., Moving shadow and object detection in traffic scenes. In: *Proceedings of 15th International Conference on Pattern Recognition*. vol. 1, pp. 321-324.

198. Morris, B. and Trivedi, M., 2008. Learning, modelling and classification of vehicle track patterns from live video. *IEEE Trans on Intelligent Transport Systems*, 9(3): 425-437.

199. Morris, B. and Trivedi, M., 2006a. Improved vehicle classification in long traffic video by cooperating tracker and classifier modules. In: *AVSS '06: Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*, page 9, Washington, DC, USA.

200. Morris, B. and Trivedi, M., 2006b. Robust classification and tracking of vehicles in traffic video streams. In: *Proceedings of IEEE Intelligent Transportation Systems Conference*, 1078-1083.

201. Mumford, D. and Shash, J., 1989. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure Applied Mathematics*, Vol. 42, 577-685.

202. Ndiour, J. and Vela, P.A., 2010. A local extended Kalman filter for visual tracking. *Proc. of the 49th IEEE Conference on Decision and Control*, pp. 2498–2504.

203. Nguyen, P.V. and Le, H.B., 2008. A multimodal particle-filter-based motorcycle tracking system. In: *PRICAI 2008: Trends in Artificial Intelligence*, Lect. Notes Comput. Sci. Berlin, Germany: Springer- Verlag, pp. 819–828.

204. Nieto, M., Unzueta, L., Cortes, A., Barandiaran, J., Otaegui, O. and Sanchez, P., 2011. Real-time 3D modeling of vehicles in low-cost monocamera systems. In: *Proc. Int. Conf. on Computer Vision Theory and Applications VISAPP2011*, pp. 459-464.

205. Nummiaro, K., Koller-Meier, E. and Gool, L.V., 2002. An adaptive color-based particle filter. *Image and Vision Computing, 21*, 99-110.

206. Opelt, A., Pinz, A., Fussenegger, M. and Auer, P., 2006. Generic object recognition with boosting. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 416–431.

207. Osher S. and Fedkiw R., 2003. Level Set Methods and Dynamic Implicit Surfaces. Springer.

208. Osher, S. and Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulation. *J. Comput. Phys.*, vol. 79, 12-49.

209. Osher, S. and Shu, C-W., 1991. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM journal on numerical analysis*, Vol. 28, No. 4, pp.907-922.

210. Osuna, E., Freund R. and Girosi, F., 1997. Support vector machine: training and applications. *Technical Report: AIM-1602*, Massachusetts Institute of Technology, Cambridge, USA.

211. Pan, Y., Birdwell, J.D. and Djouadi, S.M., 2006. Bottom-up hierarchical image segmentation using region competition and the mumford-shah functional. In: *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 117–121, Washington, DC, USA, 2006. IEEE Computer Society.

212. Parag, T., Porikli, F. and Elgammal, A., 2008. Boosting adaptive linear weak classifiers for online learning and tracking. In *CVPR*.

213. Paragios, N. and Deriche, R., 1998. Geodesic active regions for texture segmentation *INRIA RR-3440.*

214. Paragios, N. and Deriche, R., 1999. Geodesic active regions for motion estimation and tracking. *INRIA RR-3631.*

215. Paragios, N. and Deriche, R., 2000. Geodesic active contours and level sets for the detection and tracking of moving objects. *PAMI, 22(3)*, 265-280.

216. Parameswaran, V., Ramesh, V. and Zoghlami, I., 2006. Tunable kernels for tracking. In *CVPR*, pp. 2179-2186.

217. Park, B.U.; Marron, J.S., 1990. Comparison of data-driven bandwidth selectors. *Journal of the American Statistical Society* **85** (409): 66–72.

218. Park, B.U.; Turlach, B.A., 1992. Practical performance of several data driven bandwidth selectors (with discussion). *Computational Statistics* **7**: 251–270.

219. Park, K., Lee, D. and Park, Y., 2007. Video-based detection of street-parking violation. In: *Proc. Int. Conf. Image Process. CVPR'*2007.

220. Peterfreund, N., 1997. The velocity snake. In *CVPR,* pp. 1-8.

221. Peterfreund, N., 1999. Robust tracking of position and velocity with kalman snakes. *PAMI, 21(6)*, 564-569.

222. Petrovskaya, A., Thrun, S., 2008. Model based vehicle tracking for autonomous driving in urban environments. In: *Proceedings of Robotics: Science and Systems IV.*

223. PETS, 2009. Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, *CVPR Workshop*, 2009.

224. Piccardi, M., 2004. Background subtraction techniques: A review. In: *IEEE International Conference on System*, Man, and Cybernetics, 3099-3104.

225. Pilet, J., Strecha, C., Fua, P., 2008. Making background subtraction robust to sudden illumination changes. *European Conference on Computer Vision.*

226. Platt, J.C., 1999. Probability outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, MIT Press, 61-74.

227. Pless, R., 2005. Spatio-temporal background models for outdoor surveillance. EURASIP Journal on Applied Signal Processing, 2281-2291. *GeoInformatica*, 10(1), 37-53.

228. Pless, R., 2006. Detecting roads in stabilized video with the spatio-tempral structure tensor. *Geoinformatica*, 10(1): 37-53.

229. Pless, R., Jurgens, D., 2004. Road extraction from motion cues in aerial video. In: *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems (GIS'04)*, 31-38.

230. Power, P.W., Schoonees, J.A., 2002. Understanding background mixture models for foreground segmentation. *Proceedings of Image and Vision Computing*, New Zealand, November.

231. Prati, A., Mikic, I., Trivedi, M.M., Cucchiara, R., 2003. Detecting moving shadows: algorithms and evaluation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(7), pp. 918-923.

232. Prisacariu, V. and Reid, I., 2009. PWP3D: Real-time segmentation and tracking of 3D objects. In: *BMVC2009.*

233. Prisacariu, V. and Reid, I., 2012. PWP3D: Real-time segmentation and tracking of 3D objects. *International Journal of Computer Vision*, pp.1-20.

234. Quast, K., Kaup, A., 2009. Scale and shape adaptive mean shift object tracking in video sequences. *17th European Signal Processing Conference*, 1513-1517.

235. Quinlan, J.R. C4.5: Programs for machine learning. Morgan Kaufmann Publishers, 1993

236. Rad, R., Jamzad, M., 2005. Real time classification and tracking of multiple vehicles in highways. *Pattern Recognition Letters*, 26, 1597-1607.

237. Radke, R.J., Andra, S., Al-Kofahi, O. and Roysam, B., 2005. Image change detection algorithms: A systematic survey. *IEEE transaction on Image Processing*. 14(3): 294-307.

238. Rathi, Y., Vaswani, N., Tannenbaum, A., Yezzi, A., 2007. Tracking deforming objects using particle filtering for geometric active contours. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29(8): 1470-1475.

239. Reddy, V., Sanderson, C., Sanin, A., Lovell, B.C., 2010. Adaptive patch-based background modelling for improved foreground object segmentation and tracking. In: *Proceedings of 7th International Conference on Advanced Video and Signal Based Surveillance*, pp. 172-179.

240. Ribeiro, M., 2004. Kalman and extended Kalman filters: concept, derivation and properties. *Technique Report*, Institute for Systems and Robotics - Instituto Superior Tecnico, Lisbon.

241. Rifkin, R. and Klautau, A., 2004. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5, pp. 101-141.

242. Rish, I., 2001. An empirical study of the naive Bayes classifier. *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.*

243. Rosin, P., Ioannidis, E., 2003. Evaluation of global image thresholding for change detection. *Pattern Recognition Letters*, 24(14): 2345-2356.

244. Roth, P.M. and Bischof, H., 2008. Active sampling via tracking. In: *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 1–8.

245. Rubner, Y., Tomasi, C. and Guibas, L.J., 2000. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2), 99-121.

246. Rybski, P.E., Huber, D., Morris, D.D., Hoffman, R., 2010. Visual classification of coarse vehicle orientation using histogram of orientated gradients features. *IEEE Intelligent Vehicle Symposium (IV)*, 921-928.

247. Sandberg, B. and Chan T.F., 2002. Logic operations for active contours on multi-channel images. *UCLA CAM Report 02-12.*

248. Sandberg, B., Kang, S.H., Chan, T.F., 2010. Unsupervised multiphase segmentation: a phase balancing model. *IEEE Trans. Image Processing*, 19(1), pp. 433–443.

249. Scholkopf, B., Burges, C. and Vapnik, V., 1995. Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy, editors, In *the Proceedings of the First International Conference on Knowledge Discovering & data Mining*, AAAI Press, Menlo Park, CA, pp. 252-257.

250. Scott, D., 1979. On optimal and data-based histograms. *Biometrika,* **66**: 605–610.

251. Segal, M. R., 2003. Machine learning benchmarks and random forest regression. *Center for Bioinformatics & Molecular Biostatistics*, 1-14.

252. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M. and Poggio, T., 2007. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 411-426.

253. Shi, J. and Malik, J., 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905.

254. Shi, Y., Karl, W.C., 2005. Real time tracking using level set. *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, 34-41.

255. Siddiqi, K., Lauziére, Y. B., Tannenbaum, A. and Zucker, S.W., 1998. Area and length minimizing flows for shape segmentation. *IEEE Trans. Image Processing*, vol. 7, pp. 433–443.

256. Singh, Y., Gupta, P. and Yadav, V.S., 2010. Implementation of a self-organising approach to background subtraction for visual surveillance approach. *International Journal of Computer Science and Network Security*, 10(3):136-143.

257. Sitavancova, Z., Hajek, M., 2009. Intelligent transport systems thematic research summary.

258. Sivaraman, S. and Manubhai, M., 2010. A general active-learning framework for on-road vehicle recognition and tracking. *IEEE Transaction on Intelligent Transportation Systems*, 11(2), pp. 267-276.

259. Slama, C.C., 1980. Manual of Photogrammetry. 4th ed., *American Society of Photogrammetry*, Falls Church, Virginia.

260. Song X. and Nevatia, R., 2007. Detection and tracking of moving vehicles in crowded scenes. In: *Proceedings of IEEE workshop on Motion and Video Computing (WMVC'07)*, p. 4.

261. Stauffer, C., Grimson, W., 1999. Adaptive background mixture models for real-time tracking. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 246-252.

262. Stauffer, C., Grimson, W., 2000. Learning patterns of activity using real-time tracking. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8): 747-757.

263. Stern, H. and Efros, B., 2002. Adaptive color space switching for face tracking in multi-colored lighting environments. *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 249-254.

264. Sturgess, P., Alahari, K., Ladicky, L. and Torr, P.H.S., 2009. Combining appearance and structure from motion features for road scene understanding. In: *Proc. Brit. Mach. Vis. Conf.*, pp. 1–11.

265. Su, X., Khoshgoftaar, T.M., Zhu, X. and Folleco, A., 2007. Rule-based multiple object tracking for traffic surveillance using collaborative background extraction. In: *Advances in Visual Computing*. Berlin, Germany: Springer-Verlag, pp. 469–478.

266. Sullivan, G.D., Baker, K.D., Worral, A.D., Attwood, C.I., Remagnino, P.M., 1997. Model-based vehicle detection and classification using orthographic approximations. *Image and Vision Computing* 15: 649-654.

267. Sumengen, B.: A Matlab toolbox implementing level set methods. Source code available at http://barissumengen.com/level_set_methods/index.html

268. Sussman, M., Smereka, P. and Osher, S., 1994. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, Vol. 114 (1), pp. 146-159.

269. Tan, T.N., Sullivan, G.D., Baker, K.D., 1998. Model-based localization and recognition of road vehicles. *Int J of Comp Vision*, 27(1): 5-25.

270. Tao, W., Jin, H., Zhang, Y., 2007. Color image segmentation based on mean shift and normalized cuts. *IEEE Transaction on Systems, Man, and Cybernetic*, part B: Cybernetics, 37(5), 1382-1389.

271. Terzopoulos, D. and Szeliski, R., 1992. Tracking with Kalman snakes. In A. Blake and A. Yuille (Eds.), *Active Vision,* MIT Press, pp. 553-556.

272. Thi, T., Robert, K., Lu, S., Zhang, J., 2008. Vehicle classification at nighttime using eigenspaces and support vector machine. In: *Proceedings of Congress on Image and Signal Processing*, 422-426.

273. Thida, M., Chan, K.L. Eng, H. L., 2006. An improved real-time contour tracking algorithm using fast level set method. *Advances in Image and Video Technology*, LNCS 4319, 702-711.

274. Titterington, D., 1984. Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society*, Series B (Statistical Methodological) 2 (46), pp. 257-267

275. Vapnik, V. and Chervonenkis, A., 1964. A note on one class of perceptrons. *Automation and Remote Control*, 25.

276. Vapnik, V. and Lerner, A., 1963. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24, pp. 774-780.

277. Vapnik, V., 1982. Estimation of Dependences Based on Empirical Data, Springer, Berlin.

278. Vapnik, V., 1995. The Nature of Statistical Learning Theory, Springer, New York.

279. Vapnik, V., 1999. An overview of statistical learning theory. *IEEE Transaction on neural networks*, vol. 10, No. 5, pp. 988-999.

280. Veeraraghavan, H., Masoud, O. and Papanikolopoulos, N., 2002. Vision-based monitoring of intersections. In: *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, pp. 7–12.

281. Veksler, O., 2008. Star shape prior for graph cut image segmentation. *ECCV*, Lecture Note in Computer Science, Vol. 5304/2008, 454-467.

282. Vese, L.A. and Chan, T.F., 2002. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3), 271-293.

283. Vilapana, V., Marques, F., 2008. Region-based mean shift tracking: Application to face tracking. *15th IEEE International Conference on Image Processing*, 2712-2715.

284. Vlad, I., 2001. Short Note: Using the NTSC color space to double the quantity of information in an image. *Stanford Exploration Project, Report 110*, September 18.

285. Vosters, L.P.J., Shan, C. and Gritti, T., 2010. Background subtraction under illumination changes. In: *Proceedings of 7th International Conference on Advanced Video and Signal Based Surveillance*, pp. 384-391.

286. Wand, M.P; Jones, M.C., 1995. Kernel Smoothing. London: Chapman & Hall/CRC.

287. Wang, J., Ma,Y., Li, C., Wang, H. and Liu, J., 2009. An efficient multiobject tracking method using multiple particle filters. In *Proceedings of WRI World Congress on Computer Science and Information Engineering,* vol. 6, pp. 568–572.

288. Wang, J., Thiesson, B., Xu, Y., Cohen, M., 2004. Image and video segmentation by anisotropic kernel mean shift. *ECCV*, LNCS 3022, pp. 238-249.

289. Wang, Y., Malinovskiy, Y. and Wu, Y., 2008. Occlusion robust and environment insensitive algorithm for vehicle detection and tracking using surveillance video cameras. *Technical Report, TransNow Budget No. 61-6020.*

290. Welch, G. and Bishop, G., 2001. An introduction to the Kalman filter SIGGRAPH 2001 course 8. In: *Computer Graphics, Annual Conference on Computer Graphics & Interactive Techniques*, ACM Press, Addison-Wesley, Los Angeles, CA, USA (August 12–17), SIGGRAPH 2001 course pack edition.

291. Wikipedia, 2012. *http://en.wikipedia.org/wiki/Active_contour_model.*

292. Withagen, P.J., Schutte, K. and Groen, F.C.A., 2010. Global intensity correction in dynamic scenes. *International Journal of Computer Vision*, 86: 33-47.

293. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P. 1997. Pfinder: Real-time tracking of the human body. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(7): 780-785.

294. Xie X. and Mirmehdi M., 2003. Geodesic colour active contour resistent to weak edges and noise. In: *Proceedings of the 14th British Machine Vision Conference*, BMVA Press, pages 399--408.

295. Xie, B., Ramesh, V. and Boult, T., 2004. Sudden illumination change detection using order consistency. *Image and Vision Computing*, 22: 117-125.

296. Xu, C. and Prince, J.L., 1997. Gradient vector flow: a new external force for snakes. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR' 97), pp. 66-71.

297. Xu, C. and Prince, J.L., 1998. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*. Vol. 7, No. 3, 359-369.

298. Yang, H., and Welch, G., 2005. Model-based 3D object tracking using an extended-extended Kalman filter and graphics rendered measurements. *Proceedings of Computer Vision for Interactive and Intelligent Environment*, 2005, pp. 85-96.

299. Yang, L. and Foran, D.J., 2005. Unsupervised segmentation based on robust estimation and color active contour models. *IEEE Transaction on Information Technology in Biomedicine*, Vol. 9, No. 3, pp. 475-486.

300. Yi, K.M., Ahn, H.S., Choi, J.Y., 2008. Orientation and scale invariant mean shift using object mask-based kernel. *19th International Conference on Pattern Recognition*, 1-4.

301. Yilmaz, A., 2011. Kernel-based object tracking using asymmetric kernels with adaptive scale and oritation selection. *Machine Vision and Applications*, 22(2): 255-268.

302. Yilmaz, A., Lin, X. and Shah, M., 2004. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *PAMI, 26(11)*, 1531-1536.

303. Yin, Z., Porikli, F. and Collins, R. T., 2008. Likelihood map fusion for visual object tracking. In *WACV* pp. 1-7.

304. Zadrozny B. and Elkan, C., 2002. Transforming classifier scores into accurate multiclass probability estimates. *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

305. Zhang Z., Li, M, Huang, K., Tan T., 2008. Boosting local feature descriptors for automatic objects classification in traffic scene surveillance. *19th International Conference on Pattern Recognition*, 1-4.

306. Zhang, C., Chen, X., Chen, W.B, 2006. A PCA-based vehicle classification framework. In: *proceedings of the 22nd International Conference on Data Engineering Workshops.*

307. Zhang, G., Avery, R.P., Wang, Y., 2007. A video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras. *Transportation Research Record: Journal of the Transportation Research Board*, 1993: 138-147.

308. Zhang, K., Zhang, L., Song H., Zhou, W., 2010. Active contours with selective local or global segmentation: A new formulation and level set method. *Image and Vision Computing*, 28(4): 668-676.

309. Zhang, L., Li, S.Z., Yuan, X., Xiang, S., 2007. Real-time object classification in video surveillance based on appearance learning. In: *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, 1-8.

310. Zhang, W., Wu, Q.M.J., Yang, X. and Fang, X., 2008. Multilevel framework to detect and handle vehicle occlusion. *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 161–174.

311. Zhang, W., Yu, B., Zelinsky, G. J. and Samaras, D., 2005. Object class recognition using multiple-layer boosting with heterogeneous features. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog*, vol. 2, pp. 323–330.

312. Zhao, H.K., Chan, T., Merriman, B. and Osher, S., 1996. A variational level set approach to multiphase motion. *J. Comput. Phys.*, vol. 127, 179-195, 1996.

313. Zhao, H.-K., Osher, S., Merriman, B. and Kang, M., 1998. Implicit, nonparametric shape reconstruction from unorganized points using a variational level set method. *UCLA CAM Rep. 98-7.*

314. Zhao, S.L., Lee, H.J., 2009. A spatial-extended background model for moving blob extraction in indoor environments. *Journal of Information Science and Engineering*, 25, 1819-1837.

315. Zhong, B., Liu, S. and Yao, H., 2009. Local spatial co-occurrence for background subtraction via adaptive binned kernel estimation. *9th Asian Conference on Computer Vision*, vol. III, 152-161.

316. Zhou, Z., Hu, W., Chen, Y. and Hu, W., 2007. Markov random field modeled level sets method for object tracking with moving cameras. In *ACCV*, pp. 832-842.

317. Zhu, S.C. and Yuille, A., 1996. Region competition: unifying snakes, region growing, and Bayes/MDL for muti-band image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, No.9, p.884-900.

318. Zivkovic, Z. and Heijden, F. van der., 2004. Recursive unsupervised learning of finite mixture models. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(5): 651-656.

319. Zivkovic, Z. and Heijden, F. van der., 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7): 773-780.

BIBLIOGRAPHY

Appendix A: Data sets

A.1. Caviar datasets

For the CAVIAR project a number of video clips were recorded acting out the different scenarios of interest. These include people walking alone, meeting with others, window shopping, entering and exitting shops, fighting and passing out and last, but not least, leaving a package in a public place. The data can be down load at: http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/. The resolution is half-resolution PAL standard (384 x 288 pixels, 25 frames per second) and compressed using MPEG2. The file sizes are mostly between 6 and 12 MB, a few up to 21 MB. Figure A.1 gives two image samples extracted from two videos which be used in evaluation of level set segmentation algorithm.

Figure A.1. Image samples from Caviar data sets.

A.2. iLIDS datasets

The image Library for Intelligent Detection Systems (i-LIDS) data sets is the UK government's benchmark for video analysis systems development in with partnership with the Centre for the Protection of National Infrastructure. It is licensed by the UK Home Office for image research institutions and manufacturers. Each data set comprises 24 hours of video sequences under a range of realistic operational conditions. They are used by the UK government to benchmark video analysis products. They are useful for evaluating and comparing algorithms by the computer vision community. There are currently five scenarios within i-LIDS

1. sterile zone monitoring
2. parked vehicle detection
3. abandoned baggage detection
4. doorway surveillance
5. multiple camera tracking scenario

Each of these scenarios is made up of three datasets. Only parked vehicle detection videos are used to test the algorithms proposed in the thesis. Image size is 720 x 576 pixels, 25 frames per second. Some sample images are given in Figure A.2.

Figure A.2. Image samples from i-LIDS data sets.

A.3. Kingston datasets

A.3.1. Datasets description

12 CCTV cameras distributed at busy main junctions in Kingston town centre. 5 hour video is recoded under different weather conditions (cloudy, raining) and different times (day, evening and night). Image size is 704 x 576 pixels, 25 frames per second. The videos compressed using Xvid MPG4 Codec. Some sample images are given in Figure A.3.

A.3.2. Ground truth object database

An object feature database was constructed by manually segmenting vehicles from the Kingston data sets. Each vehicle is individually detected within a constrained detection zone in the image and at it's closest distance to a (virtual) fiducial marker (indicated by the red line in Figure A.4. A closed minimally-enclosing convex polygonal boundary was manually drawn around the outline of the vehicle, and tagged with one of 4 vehicle category labels: car, van, bus and motorcycle/bicycle. Figure A.5 illustrates examples of the result of the manual object segmentation procedure. The database comprises 2055 vehicle outlines (car: 1033, van: 589, bus: 290, motorcycle: 143) and category assignments (there were insufficient observations of lorry/truck vehicles over the observation period to include in the study).

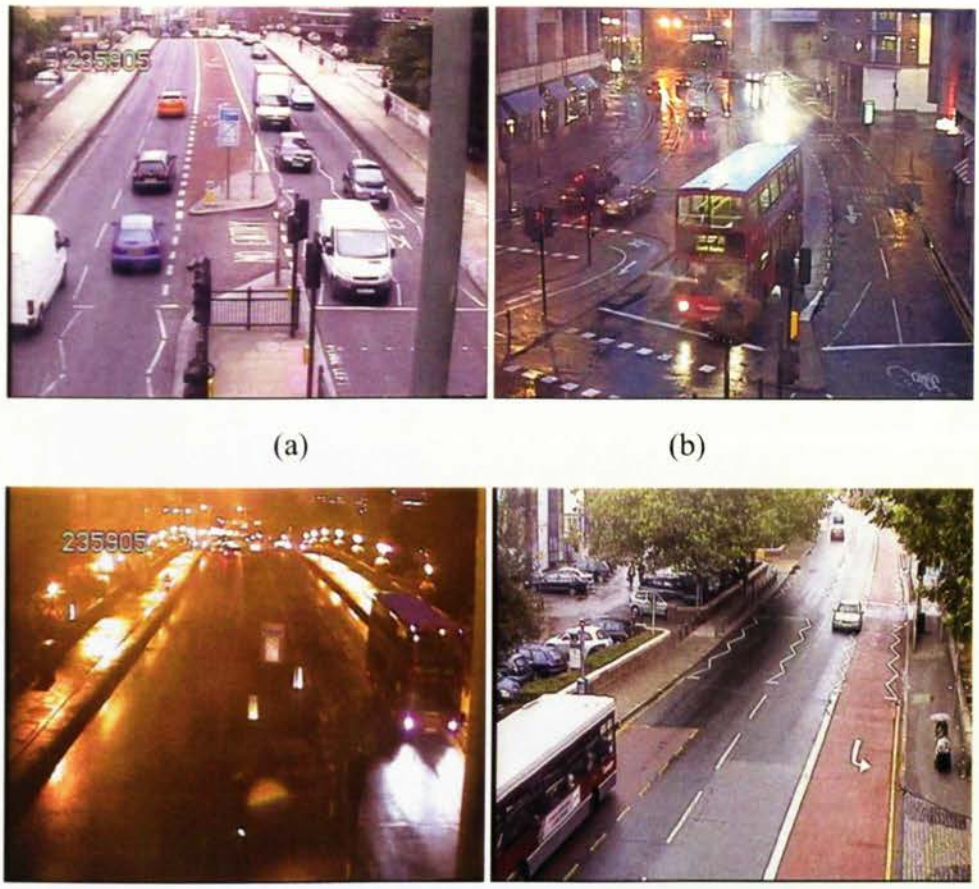


Figure A.3. Image samples from Kingston data sets. (a) In day time. (b) In the evening. (c) At night. (d) On a raining day.

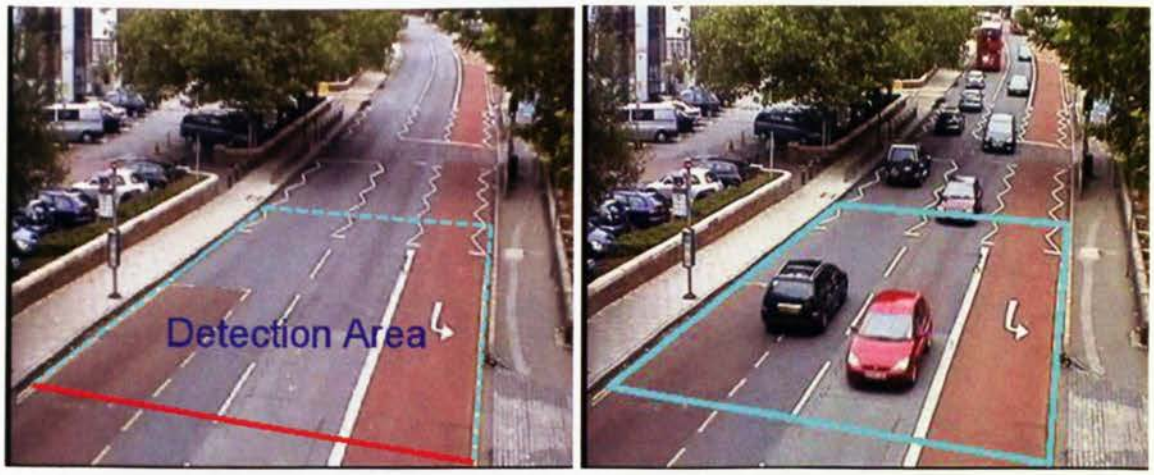
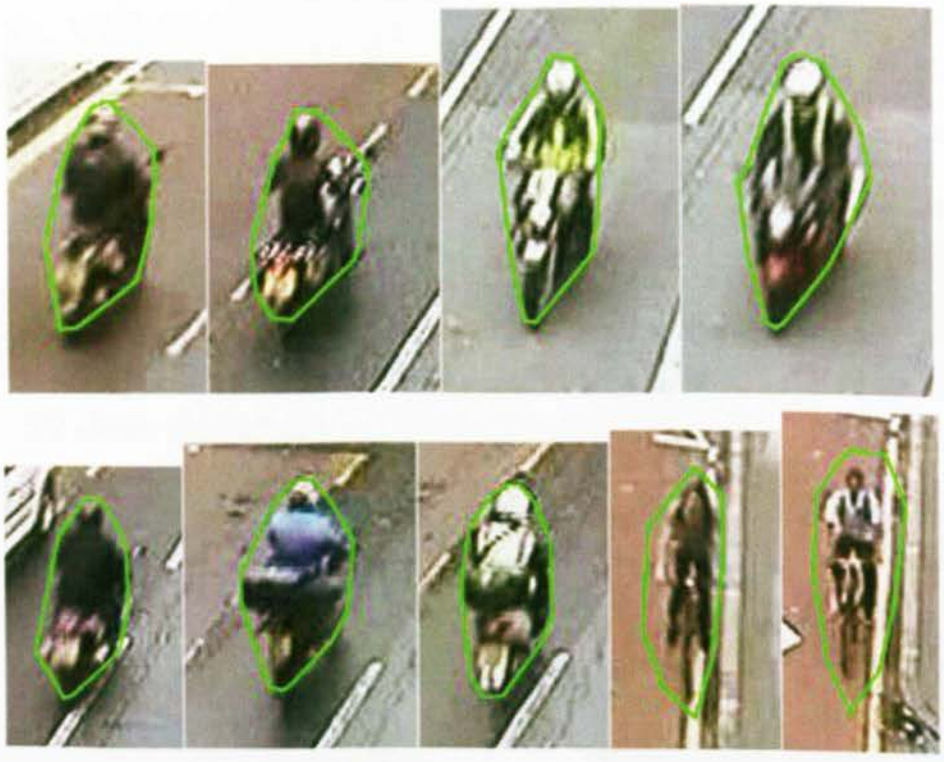


Figure A.4. Vehicle detection zone and fiducial marker (red line)



(a)

(b)

(c)

(d)

Figure A.5. Examples of manual segmentation for each vehicle category (green line): (a) motor cycle (b) car (c) van (d) bus.

A.4. York datasets

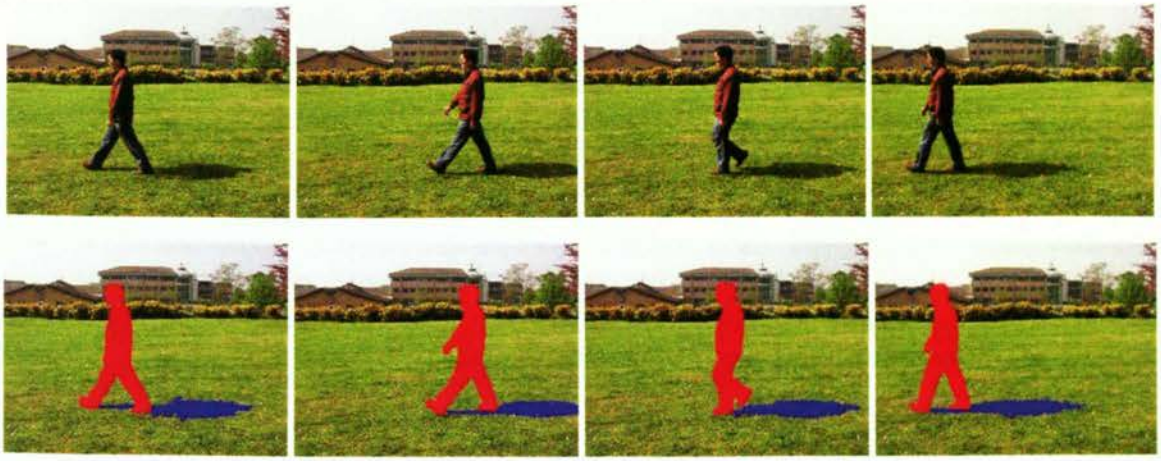
A.4.1. Datasets description

The York data sets are videos taken at the campus of the University of York using a hand held camera. The image is not very stable. In order to test the accuracy and robustness of the background subtraction algorithm, the video was taken under different weather conditions (sunny, raining, snowing and strong wind). Some sample images with strong shadow are given in Figure A.6.

Figure A.6. Some sample images with strong shadow.

A.4.2. Ground truth database for shadow removal

Figure A.7 shows two videos (walking people and moving car) which manually annotated ground truth foreground and shadow by using Paint Shop Pro 7 software for evaluating background subtraction and shadow removal algorithms.



(a)

(b)

Figure A.7. Example images of ground truth for shadow removal algorithm evaluation. (a) Walking person. (b) Moving car.

A.4.3. Vibration Datasets

The video was took by a pole mounted road side CCTV camera under very strong wind weather.781 frames are included in the video. 113 frames including foreground objects (vehicles) have been manually annotated. The image size is 320x240 pixels, 25 frames per second. Sample frames with annotated ground truth (red silhouette) are given in Figure A.8. The white 'x' shows the sample point. The ground truth location of this sample pixel stream over 781 frames has been annotated. The variance of x and y coordinates are 4.918 and 5.115 pixels, respectively. The maximum variance range along x direction is [-8, 9] pixels, along y direction is [-11, 6] pixels.

(a) (b)

Figure A.8. Samples images with vibration. (a) vehicle with annotated ground truth (green silhouette). (b) the sample pixel is used to annotate its location.

A.5. Gray datasets

Totally 751515 frames (about 50 minutes) include in the video. The capture rate is 25 frames per second and the image size is 360× 288. The video was taken in the evening. Some sample images are given in Figure A.9.

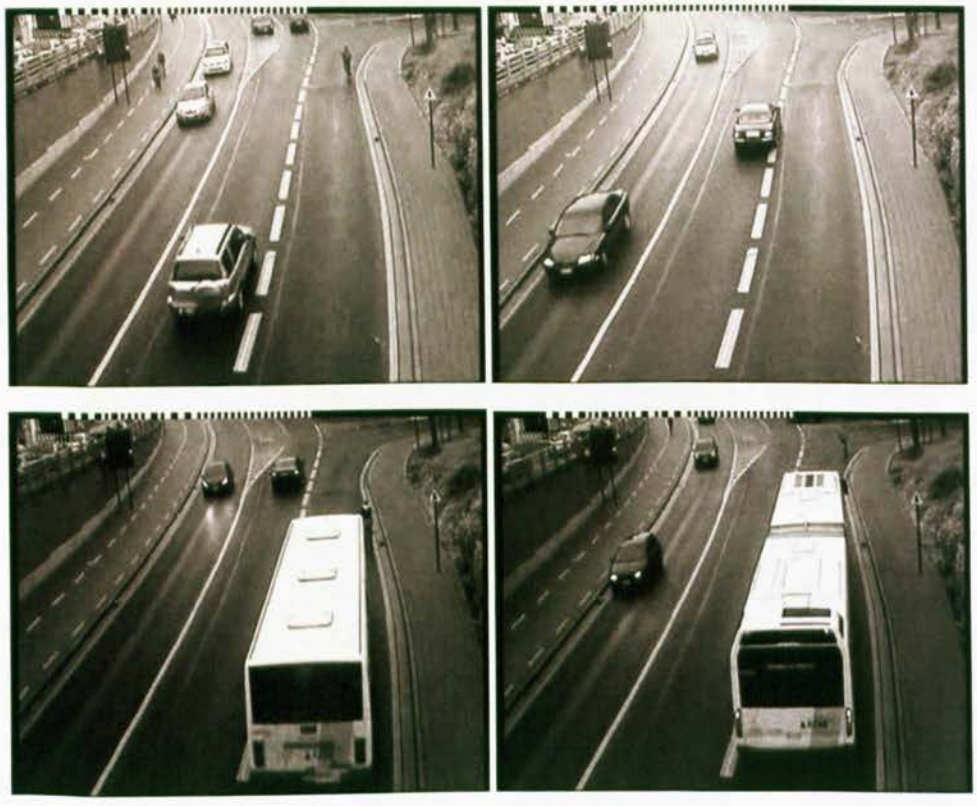


Figure A.9. Some sample images of gray video.

Appendix B: Toolbox

The following toolboxes have been used to implement the algorithms proposed in the dissertation.

- A Matlab toolbox implementing level set method (Sumengen Baris). This set of Matlab files implements level set methods following Osher and Fedkiw's book (Osher and Fedwik, 2003). Source code available at http://barissumengen.com/level_set_methods/index.html#download.
- SVM and kernel methods Matlab toolbox (Canu et al., 2005). Source code available at ttp://asi.insa-rouen.fr/enseignants/~arakoto/toolbox/index.html.
- LIBSVM: a library for support vector machines (Chang and Lin, 2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.