Motion Segmentation Using Differential Geometry of Curves and Edges.

by
Vishal Jain
B. Tech., Indian Institute of Technology, 2002

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Division of Engineering at Brown University

Providence, Rhode Island
May 2009

ⓒ Copyright 2008 by Vishal Jain

This dissertation by Vishal Jain is accepted in its present form by
the Department of Engineering as satisfying the dissertation requirement
for the degree of Doctor of Philosophy.

Date ________________ ____

Benjamin B. Kimia, Advisor
Division of Engineering

Recommended to the Graduate Council

Date ________________ ____

Gabriel Taubin, Reader
Division of Engineering

Date ________________ ____

David B. Cooper, Reader
Division of Engineering

Approved by the Graduate Council

Date ________________ ____

Sheila Bonde
Dean of the Graduate School

Curriculum Vitae

Vishal Jain was born on 24^{th} August, 1980 in Patiala, Punjab. He did his schooling at Jaspal Kaur Public School from 1987 to 1998. He graduated from the school as a valedictorian. He secured an All India Rank of 185 out of approximately 100,000 in Joint Entrance Exam. He joined Electrical Engineering Department at Indian Institute of Technology, Delhi in 1998. He received prestigious Student Undergraduate Research Award at IIT for a research project done in the Physics Department at IIT-Delhi in summer 2000. He graduated from IIT in 2002. Since 2002 he has been at the Laboratory of Engineering Man, Machine and Systems (LEMS) at Brown University. He has been recipient of Brown University Fellowship for 2002-2003 and since 2003 has been Research Assistant working on primarily video related computer vision problems. He has published several articles in peer reviewed journals, conferences and workshops. His paper at an international computer vision workshop received the Best Paper Award. His research interests in computer vision include motion segmentation, motion deblurring and recognition.

Dedication

This thesis is dedicated to Anshika Jain whose constant support and guidance has led me compile this dissertation.

Acknowledgements

Foremost I would like to thank my mother Santosh Jain and my father Naresh Jain for their constant love, support and patience throughout my life. I am grateful to them for always holding my hands through all the stages of my life. Their continual encouragement and guidance to strive for knowledge has led me to grow intellectually. It was because of their sacrifices ranging in the sector of finance to time, I have been able to come this far.

I would like to thank my sister Anshika Jain whom I dedicate this thesis to. She has been my role model, a parent and a friend all through my life. Her unselfish advice, help, and commitment of prioritizing my well being over anything else has always eased my way for my studies and otherwise. She has been instrumental and indispensable in my accomplishments throughout my life.

My advisor Prof. Benjamin B. Kimia has been a great mentor at Brown University. I thank him for inviting me to Brown for graduate studies. His constant support and guidance not only for research but also in areas of writing and presentation has affected my attitude and aptitude. He enabled numerous opportunities for me to teach, review papers and attend conferences.

I would like to thank Prof. Joseph L. Mundy who has been a second mentor to me. I learnt a lot of things from him and he has been my role model. He has guided and shared his valuable experience with me throughout my stay at Brown University. His wisdom and work ethics have always inspired me. I would also like to express my gratitude towards Prof. David Cooper and Prof. Gabriel Taubin for their support and discussions and feedback during the famous “DARPA meetings”.

My girlfriend Amanda Adams has been tremendous support during my last phase of studies at Brown. She used to provide me late night tea and snacks and gave me ride homes. Her cheerful and unselfish support was enough to keep off the stress. I would like to thank my lab mates and friends Amir Tamrakar, Ming-Ching Chang, Nhon Trinh, Ricardo Fabbri, Ozge Ozcanli, Fatih Calikali, Raghavan Dhandapani, Mathew Leotta, Daniel Crispell, Ibrahim Eden, Isabel Restrepo, Pradeep Krishna, John Raiti, Gamze Tunali and Eduardo Almeida for their help and providing a fun atmosphere in the lab.

I would like to thank the staff at Brown who were supportive and kind during my stay at Brown. I would like to thank Virginia Novak (Ginny), Gail Lee, Cheryl Carvalho and Rich Minogue.

Contents

List of Tables **xi**

List of Figures **xii**

1 Introduction **1**

1.1 Notation 11

1.2 Theory On Curves 13

1.2.1 Different Types of Curves 13

1.2.2 Image curves: one-parameter family of curves 14

1.2.3 3D curves under Motion 19

1.2.4 Relation of Image Curves to 3D Scene 19

1.3 Detailed Overview of Approach 20

1.3.1 Stage I(a): Segmentation based on curves from two views 20

1.3.2 Stage I(b): Segmentation based on edges in a registered/static scene 22

1.3.3 Stage II: Multiple-frame consistency 24

1.3.4 Extension to 3D motion 27

1.4 Contribution 28

1.5 Organization of the thesis 29

2 Related Work **30**

2.1 Static Camera 30

2.2 Non-static camera: Dense flow based 31

2.2.1 Optical Flow 31

2.2.2 Motion Segmentation based 2D Motion 33

2.3 Non-static camera: Factorization based 35

3 Segmentation using Curves **37**

3.1 Related Work 38

3.2 Alignment between two curves 39
3.3 Curves Transitions 41
3.4 Curve Tracking via Transition-based Elastic Matching 41
  3.4.1 Extracting Contours 43
  3.4.2 From a similarity metric to curve correspondence: 43
  3.4.3 Transition-sensitive elastic Matching: 43
  3.4.4 Explicit handling of transitions: 46
3.5 Use of Epipolar constraint to reduce ambiguity: 46
3.6 Use of color and intensity to reduce ambiguity 48
3.7 Transformation-Induced Figure-Ground Segregation 51
  3.7.1 Comparisons: 54
3.8 System parameters: 55

4 Edge-based Segmentation of Moving Objects in Videos from Stationary Camera or Registered Images 61
4.1 Introduction 61
4.2 Effect of illumination change 64
4.3 Mixture of Gaussians Background Model 66
4.4 Different variations of Observations 68
4.5 Experiments & Results 75
4.6 Conclusion 76

5 Multi-Frame Enrichment of Motion Segmentations 80
5.1 Introduction 80
5.2 Existing Approaches 84
5.3 Alignment of Edge-Maps 85
  5.3.1 Similarity Between Two Edges 86
  5.3.2 Choice of Transformation 88
  5.3.3 Thin Plate Spline 89
  5.3.4 Representing Correspondence by Softassign 92
  5.3.5 Efficient Alternative for the TPS Model 94
5.4 Experimental Results 97
5.5 Conclusion 101

6 3D Motion Estimation 105
6.1 Introduction 105

6.2 3D motion based approaches 110
6.3 Differential 3D structure and motion of a fixed curve from observed differential trace. 113
6.4 Numerical Estimation of Observable from the trace. 119
6.5 Stability of V as a Function of Observable. 123
6.6 Analysis and Experiments 128

A Evolution of curves: trace and parameterization 132

Bibliography 139

List of Tables

1.1 This table compares three types of representations, features, pixels, and curves in terms of properties such as density, correspondence estimation, robustness to illumination changes and the delineation of the object boundaries. 4
1.2 This table summarizes the relationship between 2D and 3D motion and shape parameters and clearly demarcates between shape and motion as well as unknowns and observable. 8
1.3 Notations 14
1.4 Visual description of one-parameter family of curves. The one-parameter family of curves have two notions *(i)* trace and *(ii)* parameterization. The parameters corresponding to each of the above two notions are segregated. 15

3.1 Overall curve performance for four pair of video frames. 52
3.2 Performance of segregating curves in four pair of video frames. 54
3.3 System parameters are listed along with their sensitivities and effects. 56

4.1 System parameters 71

5.1 System Parameters 97

6.1 This table compares three types of representations, features, pixels, and curves in terms of properties such as density, correspondence estimation, robustness to illumination changes and the delineation of the object boundaries. 106
6.2 This table summarizes the relationship between 2D and 3D motion and shape parameters and clearly demarcates between shape and motion as well as unknowns and observable. 112
6.3 Visual description of segregation of 3D Structure and Motion parameters into 2D observables and 2D and 3D unobservables. The first row shows the unknowns of 3D structure and motion can be split into unknowns of one-parameter family of curves and derivatives of depth ρ. 114

List of Figures

1.1 This figure shows the segmentation of the figure on the left using implementation based on [75, 65]. Note how the segmentation is far from the desired segmentation as the whole van. 1

1.2 The goal of motion segmentation is the delineation of objects relative to a background. The image on the right shows the segmentation of the scene on the left in different colors. 2

1.3 This figure shows the motion segmentation using features. The left image shows tracking of features and the right shows segmentation of features into different motions shown in different colors namely green, red, and blue. 3

1.4 This figure shows the motion segmentation using dense-flow computed on pixels . The left image shows dense flow estimated on each pixel and the right shows segmentation of regions based on their flow into different motions shown in different colors namely green, pink, and orange. 3

1.5 This figure shows the motion segmentation using curves . The left image shows curves and the right shows segmentation of curves into different motions shown in different colors namely green, blue , and red. The black color curves do not belong to any group. 4

1.6 The feature-based segmentation of a moving vehicle from a video of sequence of frames one of which is shown in (a) gives a sparse representation in (c) in contrast to curve-based segmentation using the methods of this thesis (b). On the other hand the pixel-based approach gives a dense segmentation than both (b) and (c) but misses a lot of regions due to ambiguous flow in regions of uniform intensity. 5

1.7 This figure shows the 2D curves obtained from a moving 3D curve or moving camera. (a) $\mathbf{\Gamma}(s,t)$ is a 3D stationary curve shown in green and non-stationary curve shown in red and its projection $\boldsymbol{\gamma}(s,t)$, (b) projections of a 3D curve onto different frames which is shown in (c). 6

1.8 This figure shows an example of local one-parameter family of curves obtained by projection of a local 3D curve. (a) A video sequence of moving truck whose edge maps are shown in the same frame in different colors in (b). A zoomed in image of local window is shown in (c) which is sample of one-parameter family of curves. . 7
1.9 Flow of the Approach for using curves and edges to do motion segmentation and enrich the segmentations with a second stage Multiple frame consistency approach. 9
1.10 This figure classifies curves into different categories into *(i)* reflectance curve (green) due to reflectance discontinuity, *(ii)* occluding contour (blue) due to depth discontinuity, *(iii)* highlight contours (red) and *(iv)* ridges due to surface discontinuity. . . . 15
1.11 (a) Projection of a 3D curve onto different frames under the camera motion $c(t)$. (b) combined into a single frame.The curves represent our geometric model and the cyan curve is the reference frame and other curves are superimposed from the neighboring frames. (c) The edges represent ”trace” (d,e) arbitrary parameterization of family of curves. 16
1.12 Top row shows the matched curves using adjacent frame. Note that the same colors denote the correct match and for a curve which didn’t find any match is colored black. The bottom row shows the motion segmentation from the matches of the top row. Observe how the segmentation is missing few curves due to spurious matching. 22
1.13 Top row shows an aerial video sequence and its edge map is shown in the middle row. The foreground detections are shown in the last row. 23
1.14 ROC curve obtained for foreground detection using Intensity (Black), Gradient (Pink), Pixel Edges (Yellow) and Subpixel Edges (Red) for sequence Figure 4.6 . . 24
25
1.16 ROC curve for comparing figure edge-maps after using multi-frame consistency (shown in Pink) with raw figure-edge maps (shown in blue). 27

2.1 The segmentation of object viewed from a static camera using [86]. (a) shows a typical background frame and (b) shows image (a) with a moving object and the segmentation of the moving object is shown in (c) (in black color). 30
2.2 (a) frame from ”Wallflower” sequence, (b) segmentation results from [4], (c)segmentation results from [6] and (d, e) segmentation result from [85]. 34

3.1 The contour fragments of a moving vehicle are segregated from the background using only the two adjacent frames shown. 38

3.2 Typical changes in curve fragments extracted from two frames of a video sequence using the topologically-driven edge operator [76] are illustrated for two frames of the UHAUL sequence. Typically, about half of curve fragments change smoothly as illustrated in (a). However, the remaining half can be expected to undergo abrupt changes as classified into six *transitions*: (b) a curve fragment can be split into two, or two can be joined into one. (c) The formation or disappearance of a T-junction. (d) The complete disappearance or appearance of a curve. (e) Compound fragmentation when two curve fragments join and split differently, a combination of two transitions of type “b”. (f) Compound T-junction, a combination of transitions “c” and “b”. (g) Compound fragmentation of closed curves, a combination of two transitions of type b. 42

3.3 From [78] The *alignment curve* (left) represents a correspondence between two curves (right). The notion of an alignment curve allows for predicting correspondences mapping an entire interval to a point; This aspect of the correspondence is crucial here as it does occur in the context of transitions. The optimal alignment curve α is efficiently found by dynamic programming [78]. 44

3.4 (a,d) The elastic curve matching alignments are incorrect in the presence of a transition but modifying the energy function to allow for such cases corrects the alignment (b,e) and furthermore recovers the fragmented “tail”. The alignment when excluding the tail is then used to define a geometric transform (similarity) between the two curves, which in turn is used to find and recover the broken curve fragment (c,f). . 45

3.5 The stage two similarity metric fails to identify the corresponding pair when multiple similar structures exist (a) or when curves do not depict significant structure (b). 46

3.6 Epipolar lines through the sample points of the first curve should pass closely to the corresponding sample points on the second point, and vice versa. The distances between these corresponding points and the lines through he original sample points indicate deviation from the epipolar constraint and is used as an addition clue towards finding the correct curve correspondence. 47

3.7 The curve $C(s)$ is shown in blue while $C^{+}(s)$ and $C^{-}(s)$ are show in red and green, respectively. The remaining contours encode other information and should be ignored in this context. (a) non-occluding curve and (b) occluding curve 48

3.8 The attributes at point p are computed at C_p^{+} and C_p^{-} neighborhoods as indicated. 49

3.9 (a) The set of curves in an image and, (b) the distance transform indicating the largest possible distance from each curve. 50

3.10 (a) The HSV color space representation, (b) Binning for building a 3D histogram . 51

3.11 Matched curves in a pair of video frames (top and bottom, on the left) and corresponding zoomed areas on the right. Corresponding curve fragments are shown in the same color. 52

3.12 Green curves in frame 2 are modeled similarity transformations of the red curves in frame 1 while blue curves are the actual curves in frame 2. 53

3.13 Results of Figure-ground segregation based on two adjacent frames for a Van (first frame shown on the left) and an SUV(first frame shown on the right). The top row shows the original image, the second row shows the contours extracted and last row shows the segmented object. 57

3.14 Two-frame segregation of moving vehicles in two subsequent video sequences. Observe how our segregation produces a rich description of the figure which can than be used for recognition. 58

3.15 (a) original video sequence, (b) segmentation of curves using our approach, (c) segmentation using KLT tracker [81], (d) region-based segmentation using optical flow and (e) edge based tracking using geodesic active contours, which requires a periodic manual initialization. 59

3.16 (a) original video sequence with an occlusion, (b) edge maps of the above video sequence, and (c) segregation of curves using our approach. 60

4.1 The effect of sudden illumination change on intensity-based background modeling . (a,b) a pair of typical background images,(c) a frame when the illumination has changed and (d) the corresponding foreground detection. Observe that the foreground, the pedestrian, has not been identified and a large part of the background is labeled as the foreground. 62

4.2 This figure shows an image formed by two surfaces S_1 and S_2 and light source L . 64

4.3 This figure compares the accuracy of background modeling using pixel and subpixel edge maps. (a) One of the background images, (b) latest observed image, (c) Pixel-edge foreground detection, (d) Sub-pixel edge foreground detection, (e) zoomed image of (c), (f) zoomed image of (d). Note how the roof of the car disappears in the case of pixel-edge modeling as the railing behind the car has the same orientation as of the roof edges. Another example with a jeep against the railing with output based on (g) pixel edges and (h) sub-pixel edges. 69

4.4 Covariance $\Sigma_{x,y}$ (green ellipse) for an edge distribution and four red dots shown represents the sites for the edge under consideration. Any edge which lies in the distribution is considered to be background. 70

4.5 The effect of change in the gain of the camera is depicted for different background modeling schemes. Top Row: a background input image, and its (b) gradient-map, (c) edge-map and (d) sub-pixel edge-map. Middle row: (a) a new input image with a change in the gain of camera, and (b) intensity gradient, (c) edge map and (d) sub-pixel edge-map. Bottom row: Foreground detection results based on (a) intensity, (b) intensity gradient, (c) pixel edge map and (d) sub-pixel edge-map. Observe that the extent of the spurious responses reduces from left to right. 72

4.6 The effect of sudden illumination change on different background modeling schemes is illustrated. First and Second row: a pair of typical (a) background images, and their (b) gradient maps, (c) edges and (d) sub-pixel edges. Third row: (a) a frame when the illumination has changed and its (b) intensity gradient, (c) edge map and (d) sub-pixel edgemap. Fourth row: Foreground detection using (a) intensity, (b) intensity gradient, (c) edge map and (d) sub-pixel edgemap. 73

4.7 A comparison on standard video sequence (a) "Akiyo" and the (b) "occluded pedestrian" video sequence. The first column shows a typical frame and its sub-pixel edgemap used for the background model, the second column shows a new frame for which a figure needs to be segregated and the last column shows the foreground detection. It is clear in all cases the sub-pixel edge is more selective. 74

4.8 Plot of detection rate versus number of frames to build the model on sequence in Figure 4.3 for intensity and sub-pixel edges based method. 75

4.9 (a) An example of the ground truth image. (b) ROC curve obtained for foreground detection using Intensity (Black), Gradient (Pink), Pixel Edges (Yellow) and Sub-pixel Edges (Red) for sequence Figure 4.6 77

4.10 This figure shows the results of our approach and intensity based approach on multiple frames of a video sequence where illumination changes at frame 37. 78

4.11 Two additional examples highlight the effectiveness of background modeling of edges. Figure and background edges are colored in green and red, respectively. First row: The mouse video is obtained from psychology department at our university where they want to do a behavioral study of the mouse by tracking its head. Second row: a moving person (note the reflections of the person from the window and table are captured). 79

5.1 This synthetic video illustrates some of the problems of motion segemtnation in a controlled setting. (a) A single frame of a video of a truck moving to the right. (b) A single frame of a video of the same truck moving to the right, but now embedded in a rich scene. (c) The segmentation of the truck in (a) using background modeling of edges [47]. (d) The same approach applied to (b) shows gaps, missing edges, due to partial occlusion, spurious edges, highlight edges due to inter-reflections, *etc.* While the segmentation in (c) can be used for recognition, tracking, *etc.* that in (d) represents a greater challenge. Realistic videos show an even greater degree of complexity of interaction and therefore greater deviation from expected models and a greater degree of degradation. 81

5.2 This figure shows the motion segmentations from different approaches. (a) Three frames of an outdoor video sequence and the motion segmentation of this video sequence using (b) KLT features based segmentation, (c) region based segmentation and (d) its curve-based object segmentation [48] is shown. Note how all these segmentation have noise and missing gaps. 82

5.3 (a) A central frame with 3 frames following it; we do not show the 3 preceding frames for clarity of presentation. (b) The edge map of the central frame is red followed by three other edge maps. (c) The composite edge map show superimposition of all the registered edges on a central frame and **(d)** show consistent edges retained. Observe how most of the gaps are completed and the spurious edges are removed. . 85

5.4 This figure illustrates the advantage of using point to curve distance. (a) Point to point distance would be problematic as the sampling of the curve is different as shown in red and blue edges. (b) an estimate of point-curve distance and (c) point to curve distance allows the samples from other frames (red) to align with the underlying curve (shown in green). 87

5.5 This figure shows the plot of $K(r) = -r^2 \log r$ from [11]. 93

5.6 (a) The correspondence between two edge maps is shown as green lines connecting corresponding edge points (red and blue) for two frames of an image sequence. (b) zoomed in (a). Observe that the majority of outliers are correctly deleted. 93

5.7 Multiple groupings or curvelet bundles from [92] shown in pink and green for an edge (blue circle). Each edge can have multiple groupings. 94

5.8 A synthetic simulation of a sequence of 5 adjacent frames which are samples from an ideal edge map (a), but where some edges are removed (20%) and where noise is added in the form of random edges (300%). Each of these five edge maps in shown in a different color (b-f). The composite edge map where all five edge maps are aligned and superimposed is shown in (g). The collection of all curve bundles is shown in (h). The edges not participating in the local curve models are labeled as spurious and removed, leaving behind only the structural edges of these only those with significant temporal presence are kept as shown in (i). 95

5.9 The two types of edges in the composite map: (a) structural edge: an edge form a frame is supported by edges from other frames is that it can be considered as a sample of curve. (b) spurious edge: an edge is spurious when it is neither structure or frequent. 96

5.10 The composite edge map top left and curve bundles top right. this is magnified to show how the enriched edge map edges are selected. 98

5.11 Results of our approach on a video sequence. (a) four frames from a video sequence and, (b) corresponding subpixel edge-map of (a) using [91], (c) Foreground edge-map determined by [**?**], (d) the composite edge map is formed by superimposing edge-maps of 5 frames for each frame, (e) the enriched edge map is the set of consistent edges which are retained. Observe the significant difference between the original edge maps in (c) and the enriched edge maps in (e). 99

5.12 Results of our approach on a video sequence. (a) four frames from a video sequence and, (b) corresponding subpixel edge-map of (a) using [91], (c) Foreground edge-map determined by [**?**], (d) the composite edge map is formed by superimposing edge-maps of 5 frames for each frame, (e) the enriched edge map is the set of consistent edges which are retained. Observe the significant difference between the original edge maps in (c) and the enriched edge maps in (e). Observe that when the vehicle undergoes an occlusion an individual frame is severely affected, in contrast to the enriched edge map which is stable and retains the occluded structure. 100

5.13 This figure shows our synthetic video rendered using a 3D rendering software [73]. (a) shows a vehicle moving against a simple background with ambient light source. (b) The vehicle moving in a scene with a more complex background, with a more complex light source model. The scene contains multiple objects, fences, posts, *etc.* to make it more realistic and model some of the factors like blending of objects into background, inter-reflections, occlusion, *etc.* which are responsible for degradation of the figure edge-maps. Figure 5.1 shows in greater detail the degradation of an ideal edge map under these conditions. 101

5.14 This figure shows the sequence used for the experiment out on synthetic video to evaluate our approach quantitatively. (a) image and (b) its edge map of the object in simple setting, (c) cropped image and (d) its edge map of the object in more realistic conditions, (c) Foreground edge-detection using [**?**], (d) Enriched edge maps corresponding to edge-map (c). 102

5.15 ROC curve for comparing figure edge-maps after using multi-frame consistency (shown in Pink) with raw figure-edge maps (shown in blue). 103

5.16 This figure shows the zoom in on the results from Figures 5.11, 5.12 and 5.14. The left column shows the single frame foreground detections and the right column shows the multi-frame composite map. 104

6.1 The feature-based segmentation of a moving vehicle from a video of sequence of frames one of which is shown in (a) gives a sparse representation in (c) in contrast to curve-based segmentation using the methods of this thesis (b). On the other hand the pixel-based approach gives a dense segmentation than both (b) and (c) but misses a lot of regions due to ambiguous flow in regions of uniform intensity. 107

6.2 The goal of motion segemtnation is the delineation of moving objects relative to a background from the sequences of frames in a video. A frame of a video is shown in (a) and the ideal results are depicted in (b). (c) dense optical flow estimation and (d) segmentation into component based on fitting a paramteric 2D motion model, shown in pink, green and orange. (e) KLT features and (f) factorization based segmentation shown in red, green and blue.(g) curves superimposed on the image and (h) segmentation based on affine motion model where distinct groups are identified in green, blue, and red with black curves are not grouped. Observe how curve-based motion segmentation provides a rich representation of objects contrast to dense-flow based and feature-based methods. 108

6.3 (a) A frame from the movie “Groundhog Day” with optical flow superimposed on it. (b) the frame can then be segmented based on 2D parametric motion models and the resulting segmentation. This clearly shows the van is divided into two pieces. Similarly the tracked KLT features of the same frame are shown in (c) and a factorization-based segmentation of these features is shown in (d), where each color represents a different group. Clearly, the van’s features are grouped in the different segments, one of which shares features from background. Finally, the curves on the van (e) are tracked and segmented based on the similarity of 2D motion using an affine model. The resulting segmentation in (f) clearly shows that the van is segmented into two regions. 109

6.4 (b) Curve-based segmentation of moving objects of original sequence in (a) and compare it with sparser set of KLT features segmented. 110

6.5 This figure shows the 2D curves obtained from a moving 3D curve or moving camera. (a) $\mathbf{\Gamma}(s,t)$ is a 3D stationary curve shown in green and non-stationary curve shown in red and its projection $\boldsymbol{\gamma}(s,t)$, (b) projections of a 3D curve onto different frames which is shown in (c). 111

6.6 The third-order edge [88] detector (shown in green) provides a better estimate of edge-orientation as compared to traditional operators (shown in red). 119

6.7 Top Row: Shows the image sequence, Middle Row: edge-maps of the above image sequence and Bottom Row: zoomed in edge-maps of neighborhood of 7 × 7. . . . 120

6.8 This figures shows the pre-processing step for computing the spatial and temporal derivatives. (a) shows edge-map and (b) shows curvelet-map of of (a). Different colors show different local groupings. 121

6.9 Top row: shows curvelet bundles computed on edge-maps of Figure 6.7(b) and bottom row shows the local grouping of edges in different frames. 121

6.10 (a) shows edge-maps from five consecutive frames (in different colors) in a single frame and (b) shows a local window of edge-maps from different frames. 122

6.11 The local edge grouping in the central frame, $t = 0$, is shown in magenta. Note the forward normal velocity (a) is computed from edges to local grouping, where as backward normal velocity (b) is computed from a local grouping to an edge. (c) shows backward β-map computed for $t < 0$ frames and forward β-map computed for $t > 0$ frames. (d) shows computation of forward β and (e) backward β. 123

6.12 This figure shows the relationship between focal length (f), angle of view (θ) and the dimension of the image (l). 127

6.13 Left column shows the plots of δV for all (θ, ϕ) for *(a)* near the optical center, $l = 0.05$, *(c)* in the middle of the image $l = l^{max}/2$ and *(e)* at the periphery of the image, $l = l^{max}$. Right column shows a binary map of the plots of the left column where blue represents error less than 1 and maroon represents error equal to or greater than 1. 130

6.14 This figure shows the plots of x, y and z components of δV for all (θ, ϕ) for *(a)* near the optical center, $l = 0.05$, *(c)* in the middle of the image $l = l^{max}/2$ and *(e)* at the periphery of the image, $l = l^{max}$. Note that the errors in $\delta\hat{\boldsymbol{V}}_x$ and $\delta\hat{\boldsymbol{V}}_y$ is not large. 131

Chapter 1

Introduction

"Scene understanding" requires identifying the component objects of a scene in relationships to their surroundings, *e.g.*, car, lamp-post, human, *etc.* in relation to road, sidewalk, *etc.* These components have to be meaningful and complete for the task in hand. This segmentation of a scene into objects is required for various applications such as traffic monitoring, surveillance activities like tracking and identifying a target, as well as inputs to more generic computer-vision algorithms such as object recognition, object tracking, and object detection.

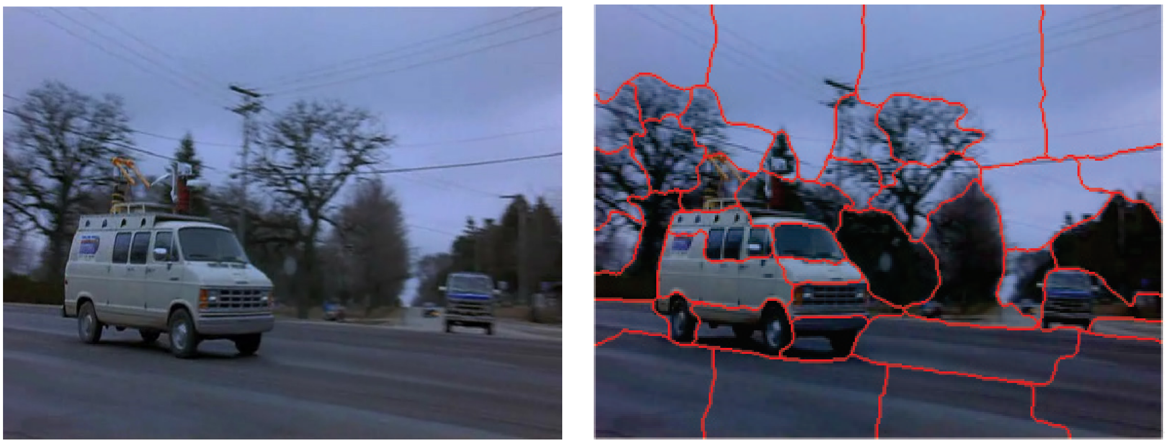


Figure 1.1: This figure shows the segmentation of the figure on the left using implementation based on [75, 65]. Note how the segmentation is far from the desired segmentation as the whole van.

The segmentation of a scene into objects from a single image in a generic, not application specific setting, has not been successful because objects in general do not satisfy any coherent cues for segmentation, *e.g.* "intensity", "color" and "texture". Rather, portions of the object may have coherent intensity, texture, color, *etc.*, but that cannot be expected to hold for the entire object. Thus, image-based segmentation algorithms, *e.g.*, [75], only can segment images into regions with low variation of intensity and texture, and this is not likely to result in a segmentation of the full object in a realistic setting, Figure 1.1. The alternative approach to this bottom-up approach is to use top-down model matching where several models of every object category are searched in the whole image. These methods are in general computationally infeasible as there exists tens of thousands of object categories, each requiring multiple prototypes to represent in this category variations. The limited success of bottom-up and top-down segmentation of single images motivates the use of a **sequence of images** in the segmentation, in our case the segmentation of several adjacent frames of a video sequence.

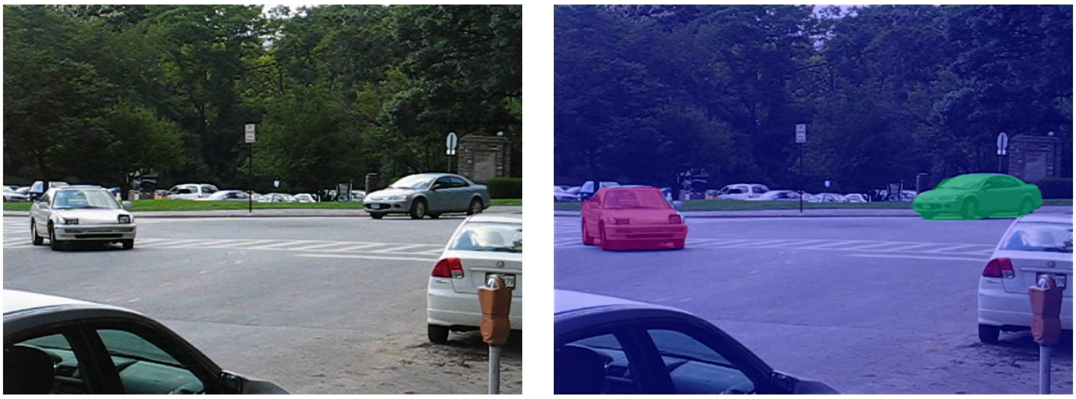


Figure 1.2: The goal of motion segmentation is the delineation of objects relative to a background. The image on the right shows the segmentation of the scene on the left in different colors.

A video clip with moving objects, moving camera, or both, provides information about different views of the same object. Since the object in the world is the same, the new views of the same object offer redundant information about the object. Of course new views can only be related to previous views if the camera projection matrices can be related, but these unknowns are few compared to the extent of new information provided by the additional views. Observe that while it is unreasonable to expect an object in its entirety to share single frame cues such as intensity, color, and texture, it is often the case that the full object has the same motion. The use of **common motion cue** for segmentation cue defines the problem of “**motion segmentation**” as the delineation of image into regions with coherent motion attributes, Figure 1.2. Specifically, the problem is one of segmenting independently moving objects which is specifically useful in applications for compression, initialization for tracking, input for recognition of the target and others.

The existing approaches for motion segmentation can be broadly classified into two classes, namely *(i)* **feature-based approaches** and *(ii)* **dense image flow-based approaches**. Both sets of approaches estimate the correspondence of features or pixels respectively, followed by stage of grouping these according to a motion model, discussed below. Each of the classes of approaches

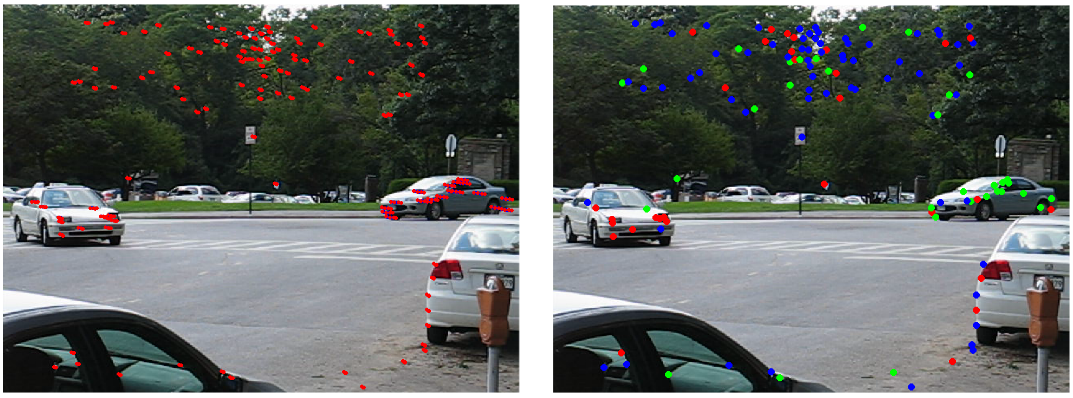


Figure 1.3: This figure shows the motion segmentation using features. The left image shows tracking of features and the right shows segmentation of features into different motions shown in different colors namely green, red, and blue.

has fundamental drawbacks. First, features such as KLT/SIFT are typically sparse and insufficient to estimate motion models (and hence the segmentation) unless the object is rich in texture. This is especially an issue for man-made artifacts, *e.g.*, the office environment, and for low resolution images, *e.g.*, aerial images, Figure 1.3.

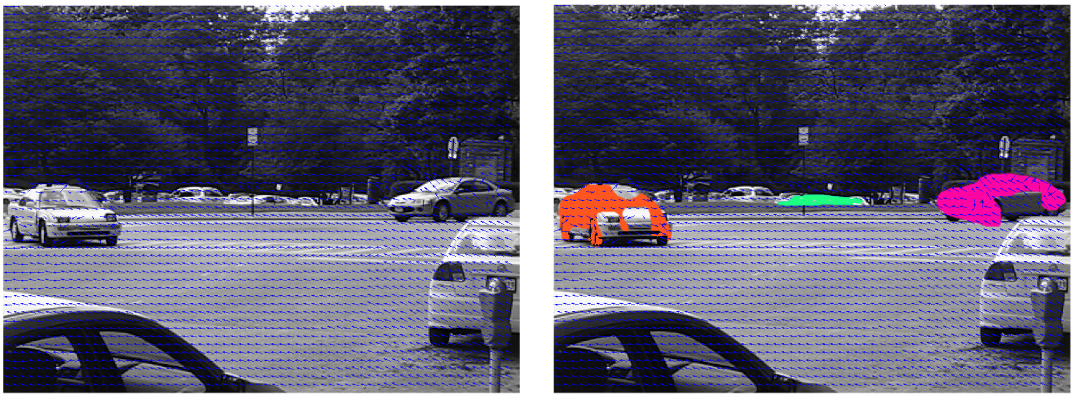


Figure 1.4: This figure shows the motion segmentation using dense-flow computed on pixels . The left image shows dense flow estimated on each pixel and the right shows segmentation of regions based on their flow into different motions shown in different colors namely green, pink, and orange.

Second, the pixel-wise dense computation of flow is ambiguous/erroneous mainly because techniques using brightness constancy have a very low signal to noise ration at low-gradient regions images which comprises a significant portion of the image, *i.e.*, the pixels away from edges. The pixels near or on the edge have higher signal to noise ratio and provide a better estimation of flow. The main difficulty is these approaches is that it is not clear when flow estimation is reliable, Figure 1.4. In this thesis it is argued that these fundamental limitations can be addressed by using curves/edges are denser than sparse features and also make clear places where flow can be reliably estimated, Figure 1.5 as elaborated below.

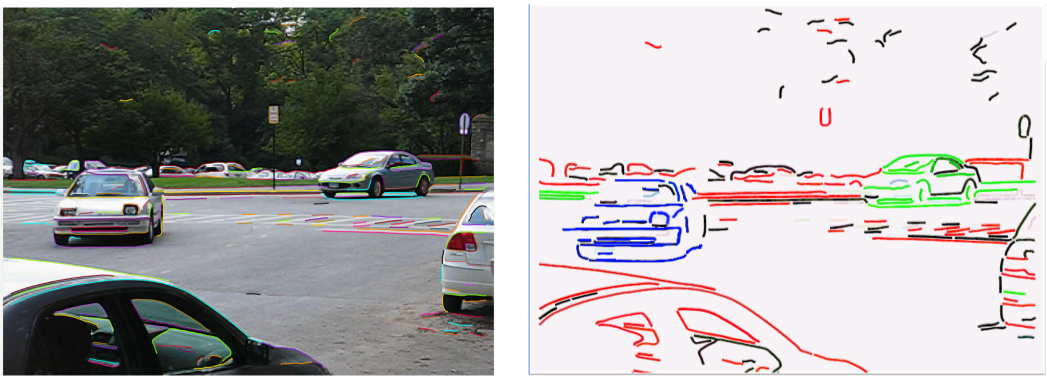


Figure 1.5: This figure shows the motion segmentation using curves . The left image shows curves and the right shows segmentation of curves into different motions shown in different colors namely green, blue , and red. The black color curves do not belong to any group.

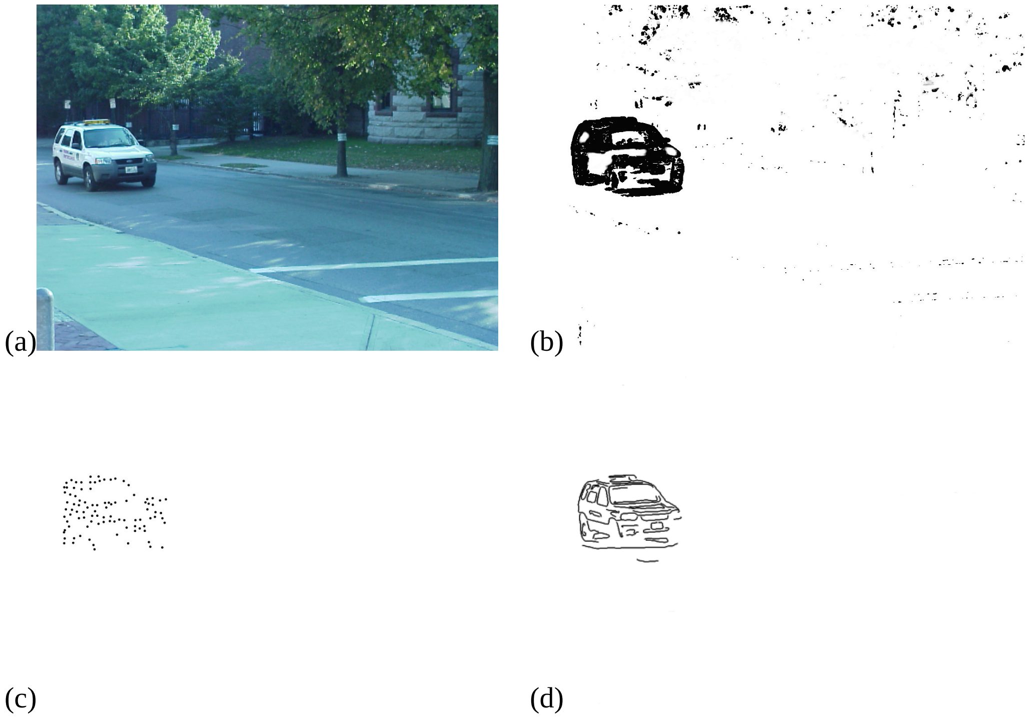
Edges are typically detected as local maxima of the gradient magnitude of the image along the gradient direction of the image. Curve fragments are obtained by linking these edges in a local neighborhood. We now discuss the merits of using curves/edges in relation to using sparse features and dense 2D pixels in estimating motion. Given that a pair of curves bounds a region and the curve-set inherently has access to the regions as shown in "visual fragments" representation of [90]. We argue a curve-based motion segmentation is most effective in further correlating results over disparate views of the same object.

	Features	Pixels	Curves
Correspondence	unambiguous	ambiguous	ambiguous along the curve
Computational Complexity	Low	High	Medium
Illumination changes	moderately invariant	variant	moderately invariant
Segmentation results	Sparse 2D cloud of points	connected set of pixels (region)	Collection of curve fragments.
Nature of Objects	Objects rich in texture	should not be homogeneous	have boundaries and reflectance edges

Table 1.1: This table compares three types of representations, features, pixels, and curves in terms of properties such as density, correspondence estimation, robustness to illumination changes and the delineation of the object boundaries.

First, in comparing the quality of correspondence of **features**, **pixels** and **curves/edges**, observe that the features are spatially localized and sparse so that a correspondence is often unambiguous.

In contrast, a dense correspondence between the two sets of 2D region pixels faces two dimensions of ambiguity. The correspondence of curves/edges, on the other hand is ambiguous in the direction along the curve but is unambiguous in direction transversal to it. Second, in comparing the computational complexity of working with features, pixels, and curves/edges, observe that features are sparse and significantly fewer than the significantly high number of pixels in an image, while curves have an in-between complexity. Third, in comparing the segmentation result of methods using these representations, observe as in Figure 1.6(c) that feature-based methods result in a sparse 2D cloud of points, dense flow methods result in regions, Figure 1.6(d), while curve-based methods present a collection of curves. Finally, the curves/edges show a greater degree of invariance to illumination changes [47] as compared to features. On the other hand dense flow based estimation is typically affected by illumination changes. These comparisons are summarized in Table 1.1. A central tenant of this thesis is to propose that curves provide a middle-ground representation in between features and pixels and act as complimentary representation in regard to features and pixels as most of the features (mainly corners) lie on the curves and a pair of curves do contain the regions of pixels.


(a) (b)

(c) (d)

Figure 1.6: The feature-based segmentation of a moving vehicle from a video of sequence of frames one of which is shown in (a) gives a sparse representation in (c) in contrast to curve-based segmentation using the methods of this thesis (b). On the other hand the pixel-based approach gives a dense segmentation than both (b) and (c) but misses a lot of regions due to ambiguous flow in regions of uniform intensity.

The work proposed in this thesis uses curves for motion segmentation for a moving camera or a moving object. The thesis examines the relationship between a 3D curve $\mathbf{\Gamma}(s,t)$, and its projection $\boldsymbol{\gamma}(s,t)$ under a video sequence, *i.e.*,

$$\mathbf{\Gamma}(s,t) = \rho(s,t)\boldsymbol{\gamma}(s,t)$$

where s is the parameterization along the curve and t is the time index and ρ is the depth, Figure 1.7(a). The one-parameter family of curves $\boldsymbol{\gamma}(s,t)$ can be examined in a single central frame, Figure 1.7(b). For example, consider the moving truck Figure 1.8 (a) whose edge maps are superimposed to give rise to this one-parameter family of curves, where each color denotes a separate time sample, Figure 1.8(b) and a zoomed area in Figure 1.8(c).

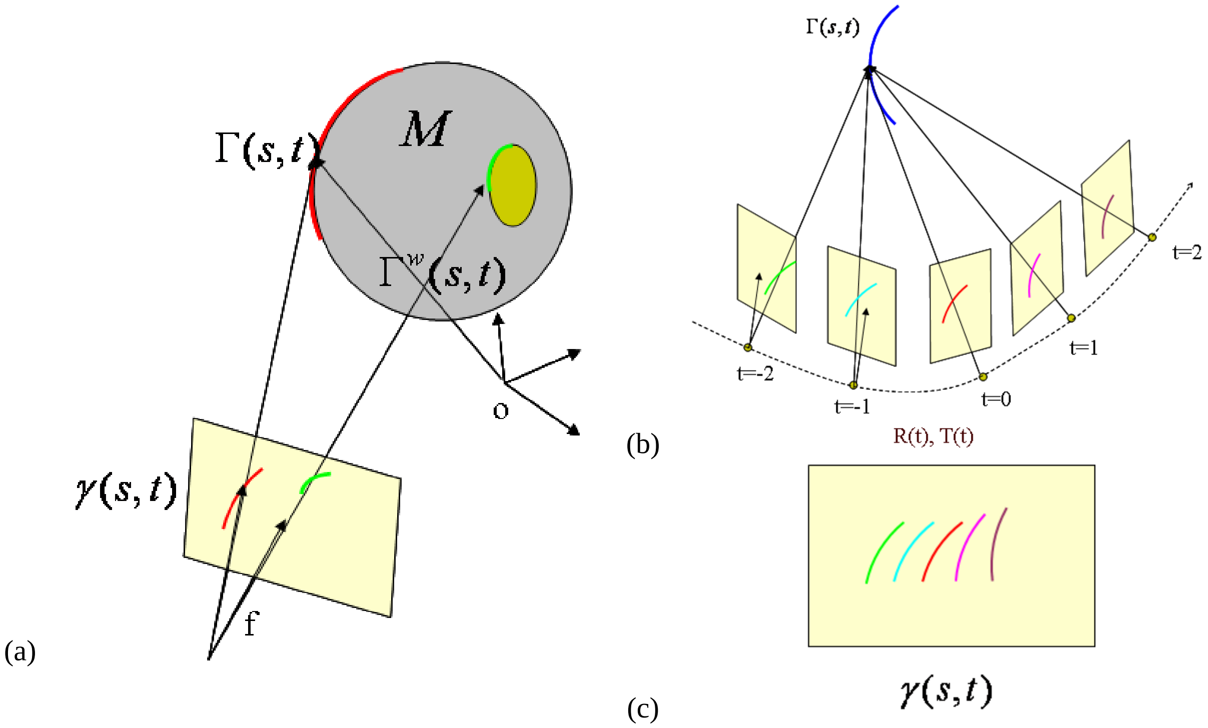


Figure 1.7: This figure shows the 2D curves obtained from a moving 3D curve or moving camera. (a) $\mathbf{\Gamma}(s,t)$ is a 3D stationary curve shown in green and non-stationary curve shown in red and its projection $\boldsymbol{\gamma}(s,t)$, (b) projections of a 3D curve onto different frames which is shown in (c).

In this approach, we focus on a camera moving with respect to an object with rotation $R(t)$ and translation $T(t)$ such that in a few local frames. this can be approximated using $\Omega(t) = \frac{dR(t)}{dt}$ and $V(t) = \frac{dT(t)}{dt}$. The shape of the 3D curve $\mathbf{\Gamma}(s,0)$ can also be locally described using a point $\mathbf{\Gamma}_0$, tangent $\vec{T}$, normal $\vec{N}$, speed of parameterization G, and curvature K. Clearly, the desirable unknown are the 3D shape of the curve $\{\mathbf{\Gamma}_0, \vec{T}, \vec{N}, G, K\}$ and the 3D motion of the curve $\{\Omega, \boldsymbol{V}\}$.

Since $\mathbf{\Gamma}(s,t) = \rho(s,t)\boldsymbol{\gamma}(s,t)$ we can also describe these unknowns in terms of the local form of the depth at ρ_0, $\{\rho_s, \rho_t\}$ and $\{\rho_{ss}, \rho_{st}, \rho_{tt}\}$ and the local form of $\boldsymbol{\gamma}(s,t)$. The one-parameter family of curves $\boldsymbol{\gamma}(s,t)$ can be first described as curve evolution in an intrinsic framework, *i.e.*,

$$\boldsymbol{\gamma}(s,t) = \alpha(s,t)\boldsymbol{t}(s,t) + \beta(s,t)\boldsymbol{n}(s,t)$$

where α and β are tangential and normal velocities, respectively. A local second-order description of this family thus requires a point $\{\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \alpha, \beta\}$ as well as $\{\kappa, \alpha_s, \beta_s, \beta_t, \alpha_t\}$.

We can now discuss which of the unknowns is observable. The depth is not directly observable. Similarly, $\boldsymbol{\gamma}(s,t)$ is only observed as a trace so that the parameterization is not directly observable from the shape of the curves (unless intensity correlation is used). We will show that only β, β_t, β_s are observable while $\alpha, \alpha_t, \alpha_s$ are not. The shape unknowns $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n},$ and κ are also observable. these are summarized in Table 1.2.

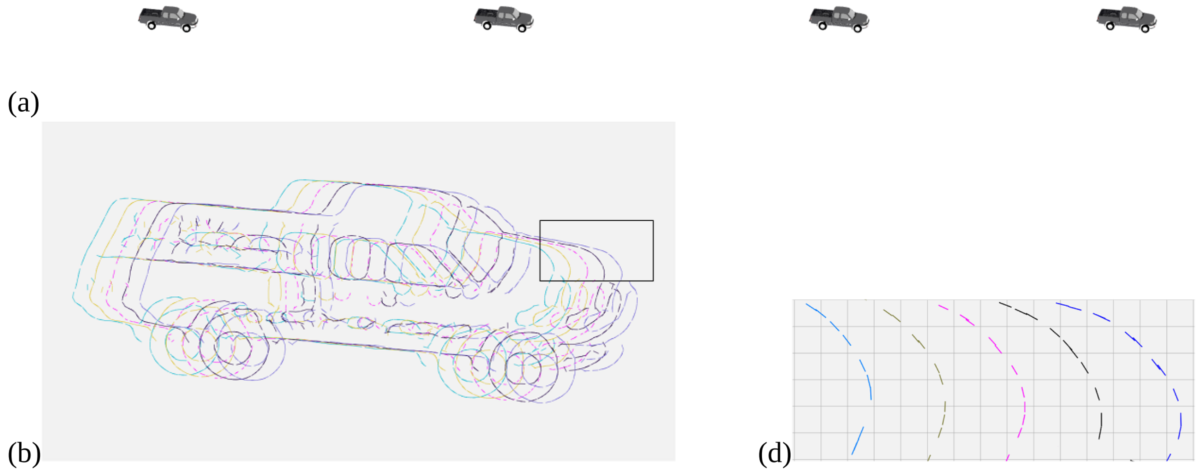


Figure 1.8: This figure shows an example of local one-parameter family of curves obtained by projection of a local 3D curve. (a) A video sequence of moving truck whose edge maps are shown in the same frame in different colors in (b). A zoomed in image of local window is shown in (c) which is sample of one-parameter family of curves.

Two approaches are now possible. First, common motion cue in 3D for compact objects implies common 2D motion (although the converse is not true). Thus estimating the missing tangential velocity $\alpha(s,t)$ can be used to group points for motion segmentation. Second, since the above approach fails for elongated objects moving towards the camera, where common 3D motion translating into a diverse range of 2D motion, we follow the approach of estimating 3D motion. We will show that for translating objects ($\Omega = 0$), the velocity of the object can be written as

$$\boldsymbol{V} = -\rho\left[\frac{\alpha(\alpha\kappa + \beta_s) + \beta_t}{2\beta}\gamma + \gamma_t\right]. \tag{1.1}$$

Unfortunately, the computation is sensitive for estimating V_z. We also show that the ratio of $\frac{\boldsymbol{V}_x}{\boldsymbol{V}_y}$ remains a powerful cue for motion segmentation. The general overview of our approach is described next.

Projection Model	3D curve $\mathbf{\Gamma}(s,0)$ 3D motion $R(t), T(t)$	γ(s,t) $\frac{d\gamma(s,t)}{dt} = \alpha \boldsymbol{t} + \beta \boldsymbol{n}$ One-parameter family of curves	Depth $\rho(s,t)$
Differential Geometry	3D curve $\mathbf{\Gamma}(s,0)$ $\mathbf{\Gamma}_0, \vec{T}, \vec{N}, G, K$ 3D motion $R(t), T(t)$ $\Omega = \frac{dR(t)}{dt}, \boldsymbol{V} = \frac{dT(t)}{dt}$	point γ_0 tangent $\boldsymbol{t}, \boldsymbol{n}$ flow α, β $\kappa, \alpha_s \beta_s, \alpha_t, \beta_t$	ρ_0 ρ_s, ρ_t $\rho_{ss}, \rho_{st}, \rho_{tt}$
Observable	None	Theorem: 2D shape $\gamma_0, \boldsymbol{t}, \boldsymbol{n}, \kappa$ Normal flow β, β_s, β_t	None
Unknowns	shape $\mathbf{\Gamma}_0, \vec{T}, \vec{N}, G, K$ motion $\Omega, \boldsymbol{V}$	tangential flow $\alpha, \alpha_s, \alpha_t$	depth ρ_0 gradient ρ_s, ρ_t Hessian $\rho_{ss}, \rho_{st}, \rho_{tt}$

Table 1.2: This table summarizes the relationship between 2D and 3D motion and shape parameters and clearly demarcates between shape and motion as well as unknowns and observable.

General Overview: This work uses the following assumptions: *(i)* **rigidity**, which implies objects to be rigid so as to ensure the coherent motion for all parts of the object and *(ii)* objects are **not too close** to the camera so as to ensure a valid approximation of 3D motion by 2D parameteric models. The approach proposed in this thesis comprises of two stages as shown in Figure 1.9. The first stage is to obtain segmentation from a single/two views and different approaches are discussed for case of stationary camera and moving camera. The second stage uses redundancy of segmentation from multiple frames to enhance the true positive and suppress the false positives. For the case of moving camera of stage one, the correspondence of the curves extracted from two views using a notion of similarity. Since the true corresponding pair of curves originate from a moving 3D curve, the similarity metric is computed by minimizing the motion of the 3D curve along the depth as well as the extent of the curve along the depth. This gives the alignment between two curves The approach proposed in this thesis comprises of two stages as shown in Figure 1.9. The first stage is to obtain segmentation from a single/two views and different approaches are discussed for case of stationary camera and moving camera. The second stage uses redundancy of segmentation from multiple

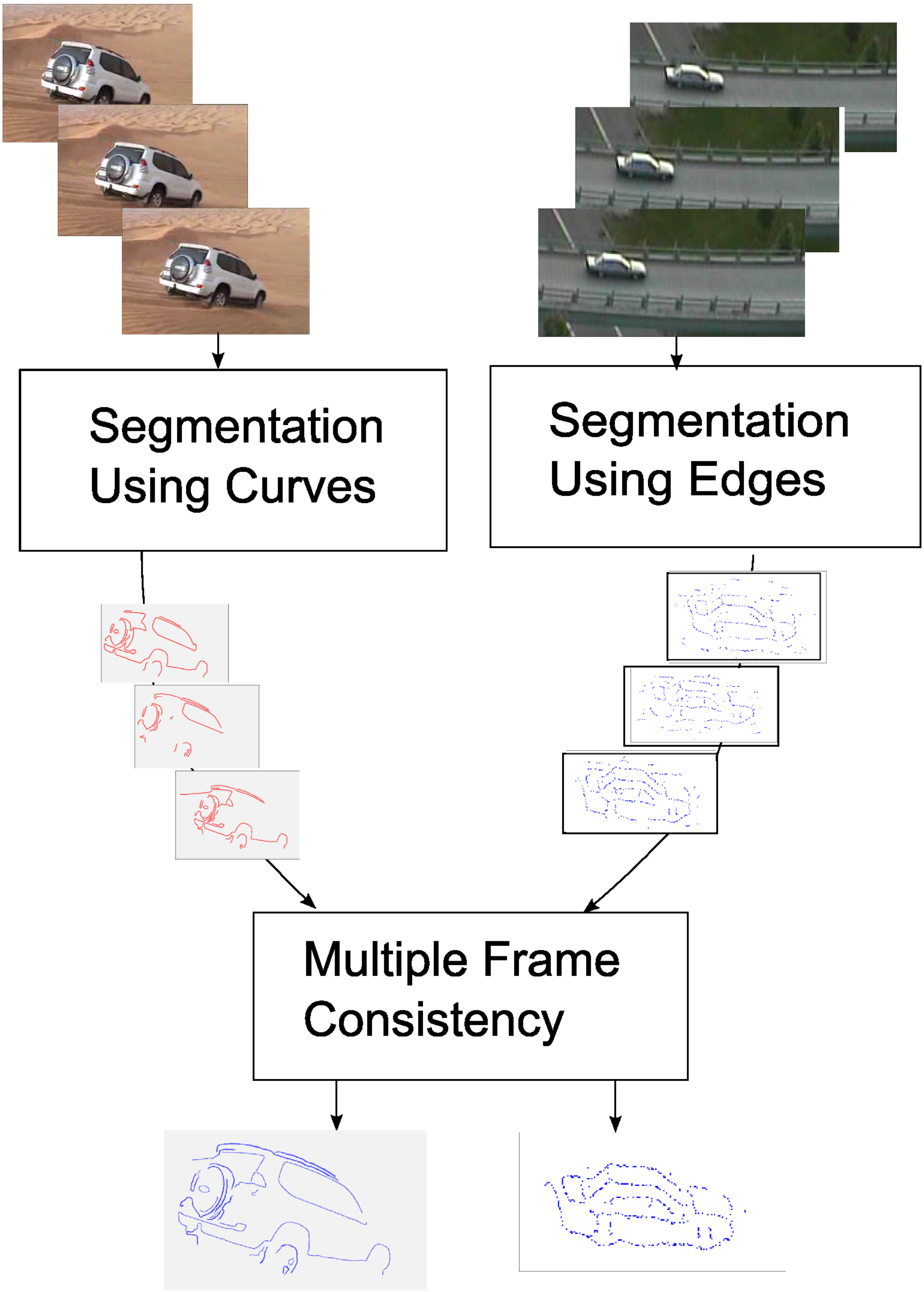


Figure 1.9: Flow of the Approach for using curves and edges to do motion segmentation and enrich the segmentations with a second stage Multiple frame consistency approach.

frames to enhance the true positive and suppress the false positives. For the case of moving camera of stage one, the correspondence of the curves extracted from two views using a notion of similarity. Since the true corresponding pair of curves originate from a moving 3D curve, the similarity metric is computed by minimizing the motion of the 3D curve along the depth as well as the extent of the curve along the depth. This gives the alignment between two curves as well as a similarity metric. This alignment for the correct pair of corresponding curve is used to fit a 2D parameteric model such as affine or similarity and the curves with similar parameteric models are grouped to obtain segmentation. Such segmentations are obtained for the whole video sequences using adjacent frames. For the case of stationary camera, the position and orientation of sub-pixel edges of the background are modeled and any deviation from this model is detected as foreground. Since the edges moving objects are different from the background edges, the moving edges are detected as foreground. This modeling of edges compared to traditional intensity based approaches is more robust to sudden changes in illumination and are more selective in distinguishing between foreground and background. The segmentation obtained in these two cases still have spurious curves/edges and some portions missing.

The second stage of this approach overcomes these impoverished segmentations by using multiple frame consistency. First , edges of the segmentations from different frames typically 5 or 7 are aligned on a central frame. This alignment is obtained by estimating the parameters of Thin Plate Spline (TPS) model. Due to the object being locally planar and far from the camera, the image flow is modeled locally by a linear flow and the minimization of the second order flow gives TPS as the solution. After the edge maps have been aligned, the edges with the notion of geometric consistency in fewer frames are pruned which leads to throwing away of noise and filling in the gaps. This has also been useful to overcome significant occlusions. One limitation of the above proposed work is that it is unable to segment objects with significant motion along depth.

In order to overcome the above limitation, this work also shows a theoretical study and analysis to estimate 3D translation. The analysis shows infeasibility to obtain reasonable estimate of 3D translation as well as shows a bias in the estimation of 3D translation direction towards optical axis. This shows an unsuccessful attempt to estimate 3D motion. This thesis shows the use of geometry of curves as the complimentary representation for motion segmentation and also lays the foundation for a unifying theory of curves in 3D and 2D for motion segmentation.

as well as a similarity metric. This alignment for the correct pair of corresponding curve is used to fit a 2D parameteric model such as affine or similarity and the curves with similar parameteric models are grouped to obtain segmentation. Such segmentations are obtained for the whole video sequences using adjacent frames. For the case of stationary camera, the position and orientation of sub-pixel edges of the background are modeled and any deviation from this model is detected as foreground. Since the

edges moving objects are different from the background edges, the moving edges are detected as foreground. This modeling of edges compared to traditional intensity based approaches is more robust to sudden changes in illumination and are more selective in distinguishing between foreground and background. The segmentation obtained in these two cases still have spurious curves/edges and some portions missing.

The second stage of this approach overcomes these impoverished segmentations by using multiple frame consistency. First , edges of the segmentations from different frames typically 5 or 7 are aligned on a central frame. This alignment is obtained by estimating the parameters of Thin Plate Spline (TPS) model. Due to the object being locally planar and far from the camera, the image flow is modeled locally by a linear flow and the minimization of the second order flow gives TPS as the solution. After the edge maps have been aligned, the edges with the notion of geometric consistency in fewer frames are pruned which leads to throwing away of noise and filling in the gaps. This has also been useful to overcome significant occlusions. One limitation of the above proposed work is that it is unable to segment objects with significant motion along depth.

In order to overcome the above limitation, this work also shows a theoretical study and analysis to estimate 3D translation. The analysis shows infeasibility to obtain reasonable estimate of 3D translation as well as shows a bias in the estimation of 3D translation direction towards optical axis. This shows an unsuccessful attempt to estimate 3D motion. This thesis shows the use of geometry of curves as the complimentary representation for motion segmentation and also lays the foundation for a unifying theory of curves in 3D and 2D for motion segmentation.

The rest of the chapter is organized by first laying out the notations used throughout this work in Section 1.1 and the notations are summarized in Table 1.3. This is followed by discussion of theory of curves in Section 1.2. This section discusses the relationship of curves in 3D to the curves in 2D. Then a detailed overview of an approach is discussed in Section 1.3. This chapter is concluded by discussing the contributions of this work.

1.1 Notation

Let $\Gamma(s) = [X(s), Y(s), Z(s)]$ be a curve in a scene on surface $\mathcal{M}$, where s is the parameterization of the curve. The perspective projection of this 3D curve on an image is denoted by $\boldsymbol{\gamma}(s) = [\xi(s), \eta(s), 1]$ where $(\xi(s), \eta(s))$ are the image coordinate whose focal-length is normalized to 1 and s is arc-length parameterization is given by

$$\mathbf{\Gamma}(s) = \rho(s)\boldsymbol{\gamma}(s)$$

where $\rho(s)$ is the depth of the object. This could be further extended for moving camera or moving curve given by

$$\mathbf{\Gamma}(s,t) = \rho(s,t)\boldsymbol{\gamma}(s,t)$$

where t is the time. $\boldsymbol{\gamma}(s,t)$ is a *one-parameter family.*

Definition 1.1. *One-parameter family of curves, $\boldsymbol{\gamma}(s,t)$, represents an evolving curve where s is the length parameter and t is the time.*

$\boldsymbol{\gamma}(s,t)$ a one-parameter family of a curve is defined by $\frac{\partial\boldsymbol{\gamma}(s,t)}{\partial s} = g(s,t)\boldsymbol{t}(s,t)$ where $g(s,t) = \|\frac{\partial\boldsymbol{\gamma}(s,t)}{\partial s}\|$ is the parameterization speed along the constant curve and $\boldsymbol{t} = \frac{\partial\boldsymbol{\gamma}(s,t)}{\partial s}/\|\frac{\partial\boldsymbol{\gamma}(s,t)}{\partial s}\|$ is the tangent and $\boldsymbol{n}$ is the vector perpendicular to $\boldsymbol{t}$ in the image plane evolving by the following model

$$\frac{\partial\boldsymbol{\gamma}(s,t)}{\partial t} = \alpha(s,t)\boldsymbol{t}(s,t) + \beta(s,t)\boldsymbol{n}(s,t), \tag{1.2}$$

where α and β are the component of image velocities along tangent and normal respectively, which determines the parameterization of the family of curves.

Let us assume a differential model of a camera, centered at $(0,0,0)$ at $t=0$ given by rotation, $R(t)$ and translation $\mathcal{T}(t)$ where t is the time. Let us denote $\Omega_\times(t) = \frac{dR(t)}{dt}R^\top(t)$ and $V(t) = \frac{d\mathcal{T}(t)}{dt}$ where

$$\Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix},\ \Omega_\times = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix} \text{ and } \boldsymbol{V} = \begin{bmatrix} \boldsymbol{V}_x \\ \boldsymbol{V}_y \\ \boldsymbol{V}_z \end{bmatrix}. \tag{1.3}$$

$R(t)$ and $\mathcal{T}(t)$ can be written as

$$\begin{cases} R(t) = I + \Omega_\times(0)t + \frac{1}{2}([\Omega_\times]_t(0) + \Omega^2_\times(0))t^2 + O(t^3) & \text{where} \quad R_t(0) = \Omega_\times(0) \\ \mathcal{T}(t) = \boldsymbol{V}(0)t + \frac{1}{2}\boldsymbol{V}_t(0)t^2 + O(t^3) & \text{where} \quad \mathcal{T}_t(0) = \boldsymbol{V}(0). \end{cases}$$

$\Gamma(s)$ in the coordinate frame of camera at time t is given by

$$\Gamma(s,t) = R(t)\Gamma(s,0) + \mathcal{T}(t) \tag{1.4}$$

Similarly a 3D curve moving by $\mathbf{\Gamma}^w(s,t)$ with moving camera is given by

$$\Gamma(s,t) = R(t)(\Gamma(s,0) + \mathbf{\Gamma}^w(s,t)) + \mathcal{T}(t) \tag{1.5}$$

Note that the $R(t)$ and $\mathcal{T}(t)$ are motion of the camera and $\mathbf{\Gamma}^w(s,t)$ is the motion of the curve. Under the assumption of linear translation the above equation simplifies to

$$\Gamma(s,t) = \Gamma(s,0) + \mathbf{\Gamma}^w(s,t) + \boldsymbol{V}t \tag{1.6}$$

The motion of the 3D curve can be decomposed into two: one due to translation of the surface $\mathcal{M}$ on which the curve resides and second due to its own deformation. The translation of the surface $\mathcal{M}$ can be combined with $\boldsymbol{V}$ of the camera as the goal is to distinguish between different motions. From now on $\boldsymbol{V}$ will be regarded as translational velocity of the curve relative to the camera. Rest of the notation is tabulated in Table 1.3.

1.2 Theory On Curves

1.2.1 Different Types of Curves

In order to understand the challenges associated with using curves for motion segmentation, the process of formation of different types of curves is discussed. The formation of curves is classified into four categories which is sufficient for the scope of this work. The four categories are:

1. Surface reflectance discontinuity: This curve arise due to discontinuity of reflectance coefficient on the same surface which leads to discontinuity in the intensity of the image, shown in green color in Figure 1.10. This curve is a function of the surface reflectance.
2. Surface normal discontinuity: This curve arises due to sharp discontinuity in the normal of the surface which means sharp bending of the surface so that the angle of light direction with the surface normal varies a lot and hence, leads to discontinuity in the intensity of the image, shown in black color in Figure 1.10. This curve is a function of the surface.
3. Depth discontinuity: This curve arises when a viewing ray is tangent to a surface $\mathcal{M}$ and this means that depth is discontinuous along the viewing ray, shown in blue color in Figure 1.10. This curve in 3D is different depending on the viewpoint. It slides on the surface as the viewpoint changes. This curve is a function of the viewpoint.
4. Highlights: This curve arises due to the surface being a mirror-like and projection of surrounding objects on-to the surface, shown in red color in Figure 1.10. This curve is a function of the surrounding objects if the surface is mirror-like.

Note that last category depends on the surrounding objects and would vary as the object is moving and hence has nothing to do with the object itself. It means that the same object in a different surrounding will produce different highlights. This work treat these curves as outliers. Since the first and second categories of curves are property of the surface and remain fixed to the surface are classified as “stationary contours”. The third-category is a function of viewpoint so it changes as the camera or the object moves. Unlike highlights, these curve are structural and needs to be retained.

Notation	Meaning
$\mathbf{\Gamma}$	a 3D point
s	arc-length parameter of a curve
t	time index
$\mathbf{\Gamma}(s,t)$	a deforming or moving 3D curve $\mathbf{\Gamma}(s)$ as function of time
$\boldsymbol{\gamma}_0$, $\boldsymbol{\gamma}(s,t)$	image point, evolving 2D curve $\boldsymbol{\gamma}(s)$ as function of time
ρ, $\rho(s,t)$	depth of a 3D point, moving 3D curve $\mathbf{\Gamma}(s)$ as function of time
$\boldsymbol{t}$, $\boldsymbol{t}(s,t)$	tangent of the image curve at a single point, at any point
$\boldsymbol{n}$, $\boldsymbol{n}(s,t)$	normal of the image curve at a single point, at any point
κ, $\kappa(s,t)$	curvature of the image curve at a single point, at any point
$U_1(s,t)$	$\boldsymbol{\gamma}(s,t) \times \boldsymbol{t}(s,t)$
$U_2(s,t)$	$\boldsymbol{\gamma}(s,t) \times \boldsymbol{n}(s,t)$
$\mathcal{M}$	local piece of surface in 3D
β	normal displacement in the image
α	tangential displacement in the image
β_s, β_t	spatial and temporal derivatives of normal image velocity
α_s, α_t	spatial and temporal derivatives of tangential image velocity
g, G	speed of parameterization of image curve and 3D curve respectively
ρ_s, ρ_t	first-order spatial and temporal derivatives of depth ρ
$\rho_{ss}, \rho_{st}, \rho_{tt}$	second-order spatial and temporal derivatives of depth ρ
$\vec{T}, \vec{N}$	tangent and normal of the 3D curve
K	curvature of the 3D curve

Table 1.3: Notations

They are classified as ”non-stationary contours”. But these curves pose a correspondence problem as the curve in 3D itself changes. This problem is handled by assuming epipolar correspondence between the cameras similar to work proposed in [20]. The classification has been simplified to two, *(i)* stationary contours and *(ii)* non-stationary contours.

1.2.2 Image curves: one-parameter family of curves

Consider a local piece of curve fragment $\mathbf{\Gamma}(s,t)$ that is observed in several frames of a video sequence, Figure 1.11(a). For simplicity, we can consider this curve evolving in a common frame, Figure 1.11(b). From a geometric perspective this is a one parameter family of curves, with *time* indexing the family of curves. The observations however, are not directly geometric curves, but rather a set of unorganized edges in each of the frame, which are sampled from an underlying curve as shown in Figure 1.11(c). These observations are indexed by time and are spatially ordered to form curves. The key ingredient underlying the spatial organization and temporal tracking of these edge segments is in that of spatial and temporal continuity in the above family of curves. The issue here is whether temporal continuity can be captured in the form of groups of individual edge elements.

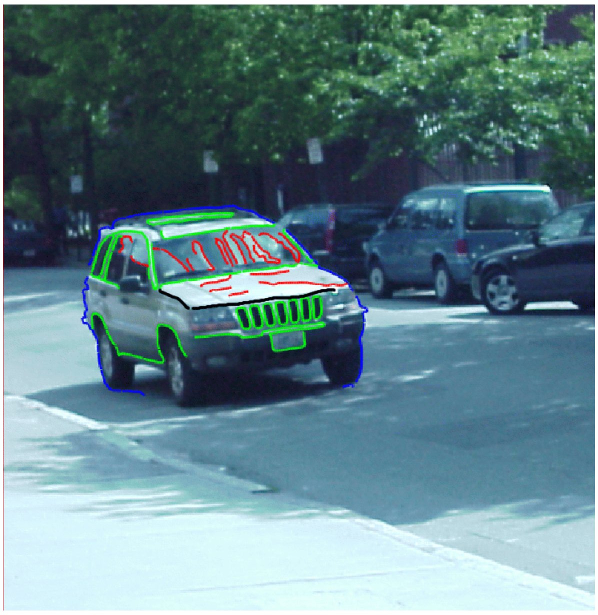


Figure 1.10: This figure classifies curves into different categories into *(i)* reflectance curve (green) due to reflectance discontinuity, *(ii)* occluding contour (blue) due to depth discontinuity, *(iii)* highlight contours (red) and *(iv)* ridges due to surface discontinuity.

One-parameter family of curves $\boldsymbol{\gamma}(s,t)$	=	Trace of One-parameter family of curves	+	Parameterization giving the correspondence
Evolution Model $\begin{cases} \boldsymbol{\gamma}_t & = \alpha \boldsymbol{t} + \beta \boldsymbol{n} \\ \boldsymbol{\gamma}(s,0) & = \boldsymbol{\gamma}_0(s) \end{cases}$	=	Trace is given by $\boldsymbol{\gamma}_t = \beta \boldsymbol{n}$	+	Parameterization is given by $\alpha \boldsymbol{t}$
Second order model given by $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \beta, \alpha, \kappa, \beta_s, \alpha_s, \beta_t, \alpha_t$	=	Trace defined from second order model is given by $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \beta, \kappa, \beta_s, \beta_t$	+	Parameterization for second order model is given by $\alpha, \alpha_s, \alpha_t$

Table 1.4: Visual description of one-parameter family of curves. The one-parameter family of curves have two notions *(i)* trace and *(ii)* parameterization. The parameters corresponding to each of the above two notions are segregated.

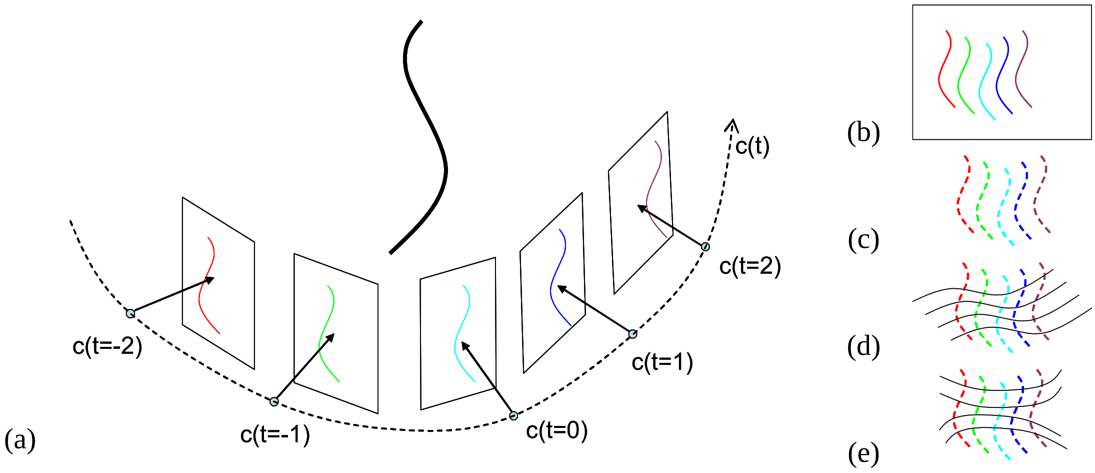


Figure 1.11: (a) Projection of a 3D curve onto different frames under the camera motion $c(t)$. (b) combined into a single frame.The curves represent our geometric model and the cyan curve is the reference frame and other curves are superimposed from the neighboring frames. (c) The edges represent ”trace” (d,e) arbitrary parameterization of family of curves.

The challenge is relating group of edges to some idealized model of continuously moving curves. There are a couple of notions that need to be distinguished for curves in different frames,

1. parameterized model $\gamma(s,t)$, which defines the parameterization of this one-parameter family of curves.

2. ”trace” of a one-parameter family of curves, Figure 1.11(c): this is what we get from our video sequence, namely, a set of points for each time.

The process of obtaining the parameterized model $\gamma(s,t)$ is complicated because a parameterized model has an *inherent ambiguity* in parameterizing the family of curves. Specifically, one ambiguity arises because it is not clear how a given point on one curve (in one time frame) is assigned to a point on the next curve (time frame incremented), Figure 1.11(e, f). Formally, let $\gamma(s,t)$ denote the family of curves. Then, the vector $\frac{\partial\gamma}{\partial t}$ points to the direction of its corresponding point. Figure 1.11(e,f) show different parameterizations. Since two different parameterizations $\gamma(s,t)$ and $\gamma(\tilde{s},\tilde{t})$ can give identical family of curves, there is an inherent freedom which can cause ambiguity in the various parameters when comparing two family of curves or when fitting samples to a one parameter family of curves. This is akin to the case of sampling a curve by $\gamma(s)$ or $\gamma(\tilde{s})$, where ambiguity can be avoided by selecting arc-length parameterization. Each curve can be parameterized by arc length, but given that for an edge only normal velocity is typically measurable (the aperture problem) flow is associated only along the normal direction. This can be formally shown by the following proposition.

Proposition 1.2. *For a one-parameter family of curves $\boldsymbol{\gamma}(s,t)$ defined by the evolutionary model by*

$$\begin{cases} \frac{\partial \boldsymbol{\gamma}}{\partial s}(s,t) &= g(s,t)\boldsymbol{t}(s,t) \\ \frac{\partial \boldsymbol{\gamma}}{\partial t}(s,t) &= \alpha(s,t)\boldsymbol{t}(s,t) + \beta(s,t)\boldsymbol{n}(s,t). \end{cases}$$

where $g(s,t)$ is defined as speed of parameterization, then the trace of $\boldsymbol{\gamma}(s,t)$ is given by

$$\frac{\partial \boldsymbol{\gamma}}{\partial t}(s,t) = \beta(s,t)\boldsymbol{n}(s,t). \tag{1.7}$$

Proof. For proof, see Appendix A.1 ■

The above proposition shows that $\beta, \boldsymbol{n}$ is observed from the trace where as α is not as tabulated in Table 1.4. Next, this distinction of trace and parameterized model is extended to second-order. The one-parameter family of curves, $\boldsymbol{\gamma}(s,t)$, is expressed up-to second-order derivatives in terms of its spatial and temporal derivatives using Taylor expansion in a local-neighborhood.

$$\boldsymbol{\gamma}(s,t) = \boldsymbol{\gamma}(s_0,t_0)+\frac{\partial \boldsymbol{\gamma}}{\partial s}(s-s_0)+\frac{\partial \boldsymbol{\gamma}}{\partial t}(t-t_0)+\frac{1}{2}(\frac{\partial^2 \boldsymbol{\gamma}}{\partial s^2}(s-s_0)^2+\frac{\partial^2 \boldsymbol{\gamma}}{\partial s \partial t}(s-s_0)(t-t_0)+\frac{\partial^2 \boldsymbol{\gamma}}{\partial t^2}(t-t_0)^2) \tag{1.8}$$

Note that these derivatives needs to be computed in order to determine the one-parameter family of curves. The following proposition enlists the unknowns required to estimate the one-parameter family of curves.

Proposition 1.3. *For a one-parameter family of curves $\boldsymbol{\gamma}(s,t)$ defined by the evolutionary model by*

$$\begin{cases} \frac{\partial \boldsymbol{\gamma}}{\partial s}(s,t) &= g(s,t)\boldsymbol{t}(s,t) \\ \frac{\partial \boldsymbol{\gamma}}{\partial t}(s,t) &= \alpha(s,t)\boldsymbol{t}(s,t) + \beta(s,t)\boldsymbol{n}(s,t). \end{cases}$$

where $g(s,t)$ is defined as speed of parameterization,then the second order derivatives $\frac{\partial^2 \boldsymbol{\gamma}}{\partial s^2}$, $\frac{\partial^2 \boldsymbol{\gamma}}{\partial s \partial t}$ and $\frac{\partial^2 \boldsymbol{\gamma}}{\partial t^2}$ are given by

$$\begin{cases} \frac{\partial^2 \boldsymbol{\gamma}}{\partial s^2} &= g_s\boldsymbol{t} + g^2\kappa\boldsymbol{n} \\ \frac{\partial^2 \boldsymbol{\gamma}}{\partial s \partial t} &= (\alpha_s - \beta g\kappa)\boldsymbol{t} + (\alpha g\kappa + \beta_s)\boldsymbol{n} \\ \frac{\partial^2 \boldsymbol{\gamma}}{\partial t^2} &= (\alpha_t - \beta(\alpha\kappa + \frac{\beta_s}{g}))\boldsymbol{t} + (\alpha(\alpha\kappa + \frac{\beta_s}{g}) + \beta_t)\boldsymbol{n}. \end{cases} \tag{1.9}$$

Furthermore, assuming arc-length parameterization at $t = 0$, i.e., *$g(s,0) = 1$ and $g_s(s,0) = 0$, the second order-derivatives are given as*

$$\begin{cases} \frac{\partial^2 \boldsymbol{\gamma}}{\partial s^2} &= \kappa\boldsymbol{n} \\ \frac{\partial^2 \boldsymbol{\gamma}}{\partial s \partial t} &= (\alpha_s - \beta\kappa)\boldsymbol{t} + (\alpha\kappa + \beta_s)\boldsymbol{n} \\ \frac{\partial^2 \boldsymbol{\gamma}}{\partial t^2} &= (\alpha_t - \beta(\alpha\kappa + \beta_s))\boldsymbol{t} + (\alpha(\alpha\kappa + \beta_s) + \beta_t)\boldsymbol{n}. \end{cases} \tag{1.10}$$

Proof. For proof, see Appendix A.2 ■

The above proposition defines $\boldsymbol{\gamma}(s,t)$ in terms of its second-order spatial and temporal derivatives. The unknowns that determine the family up-to second order are $\boldsymbol{\gamma}(s_0,t_0), \boldsymbol{t}, \boldsymbol{n}, \kappa, \alpha, \beta, \alpha_s, \beta_s, \alpha_t, \beta_t$. Note that only some of the unknowns describe the trace of $\boldsymbol{\gamma}(s,t)$ but rest define the parameterization of the family of curves. The following proposition shows which of the above parameters are obtained from the trace and which define the parameterization.

Proposition 1.4. *Given a 1-parameter family of curves under an arbitrary regular parameterization* $\gamma(s,t)$ *where* $\|\frac{\partial\gamma}{\partial s}(s,0)\| = 1$, *there are two first-order intrinsic measure (invariant to parameterization),*

$$\begin{cases} I & = \frac{\partial\gamma}{\partial s} = \boldsymbol{t} \\ II & = \frac{\partial\gamma}{\partial t} \cdot \boldsymbol{n} \end{cases} \tag{1.11}$$

and three second-order intrinsic measures

$$\begin{cases} III & = \boldsymbol{\gamma}_{ss} \cdot \boldsymbol{n}, \\ IV & = \boldsymbol{\gamma}_{st} \cdot \boldsymbol{n} - \kappa \boldsymbol{\gamma}_t \cdot \boldsymbol{t}, \\ V & = (\boldsymbol{\gamma}_{st} \cdot \boldsymbol{n})^2 - \kappa \boldsymbol{\gamma}_{tt} \cdot \boldsymbol{n}, \end{cases} \tag{1.12}$$

where $\boldsymbol{t}$ *and* $\boldsymbol{n}$ *represent the unit tangent and normal, respectively. In other words, the remaining degrees of freedom from the first-order derivative* $\frac{\partial\gamma}{\partial t} \cdot \boldsymbol{t}$, *and from the second-order derivatives,* $\frac{\partial^2\gamma}{\partial s \partial t} \cdot \boldsymbol{t}$ *and* $\frac{\partial^2\gamma}{\partial t^2} \cdot \boldsymbol{t}$ *are dependent on the choice of parameterization.*

The correspondence of these images curves across frames need to be computed. The correspondence across image curves is formulated as estimating the parameterization $\boldsymbol{\gamma}(s,t)$ of the trace observed in different frames. The trace of $\boldsymbol{\gamma}(s,t)$ is given by unorganized edges in different frames, Figure 1.11 (c) which is observed. The parameterization giving the correspondence is unknown. Proposition 1.2 shows that numerous parameterizations, α, can give the same trace as shown in Figure 1.11 (d,e) and the one which is desired $\alpha = w$ gives the true *correspondence* related to the same 3D point in the case of stationary contour or epipolar correspondence in case of non-stationary contours.

Proposition 1.3 shows that $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \kappa, \alpha, \beta, \alpha_s, \beta_s, \alpha_t, \beta_t$ are required to estimate $\boldsymbol{\gamma}(s,t)$ up-to second order. But all of them cannot be obtained from observation and Proposition 1.4 further shows that $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \kappa, \beta, \beta_s, \beta_t$ can be computed from the trace and rest of the $\alpha, \alpha_s, \alpha_t$ cannot be observed and would determine the parameterizations, as summarized in Table 1.4.

1.2.3 3D curves under Motion

One of the assumptions of this work is that the curve in moving with linear translation for an infinitesimal small amount of time. This assumption of constant translation refers to high frame rate of video as compared to the motion so that the trajectory of the curve can be modeled locally by linear motion. The stationary curve can be modeled as

$$\mathbf{\Gamma}(s,t) = \mathbf{\Gamma}(s,0) + \boldsymbol{V}t.$$

where $\boldsymbol{V}$ is the constant velocity of the object relative to the camera and the non-stationary curve is modeled by

$$\mathbf{\Gamma}(s,t) = \mathbf{\Gamma}(s,0) + \boldsymbol{V}t + \mathbf{\Gamma}^w(s,t). \tag{1.13}$$

where $\mathbf{\Gamma}^w(s,t)$ represents the motion of the curve due to the sliding of the curve on the surface due to change in viewpoint and $\boldsymbol{V}$ is the 3D motion of the object on which $\mathbf{\Gamma}$ resides. The general equation for moving 3D curve relative to the camera is given by

$$\mathbf{\Gamma}(s,t) = \mathbf{\Gamma}(s,0) + \boldsymbol{V}t + \mathbf{\Gamma}^w(s,t).$$

where $\mathbf{\Gamma}^w(s,t) = 0$ for stationary curve.

1.2.4 Relation of Image Curves to 3D Scene

Since the correspondence of the image curves is unknown as $\alpha, \alpha_s, \alpha_t$ cannot be observed, additional constraints are required. The curves in 3D moving relative to the camera is given by:

$$\mathbf{\Gamma}(s,t) = \mathbf{\Gamma}(s,0) + \boldsymbol{V}t + \mathbf{\Gamma}^w(s,t).$$

Proposition 6.1 shows that the time derivatives of one-parameter family of curves can be expressed in terms of $\boldsymbol{V}$

$$\begin{cases} \boldsymbol{\gamma}_t &= \frac{1}{\rho}(\boldsymbol{V} - \boldsymbol{V}_z\boldsymbol{\gamma}) \\ \boldsymbol{\gamma}_{st} &= (\boldsymbol{V}_z\boldsymbol{\gamma} - \boldsymbol{V})\frac{\rho_s}{\rho^2} - \frac{\boldsymbol{V}_z}{\rho}\boldsymbol{\gamma}_s \\ \boldsymbol{\gamma}_{tt} &= \frac{(-2\boldsymbol{V}_z)}{\rho}\boldsymbol{\gamma}_t + \frac{e_3^\top(\mathbf{\Gamma}_t^w)}{\rho}\boldsymbol{\gamma}_t \end{cases} \tag{1.14}$$

where $\boldsymbol{V}_z = e_3^\top \boldsymbol{V}$. Note that effect of motion of occluding contour plays a role only in $\boldsymbol{\gamma}_{tt}$. It is important to note that, occluding contours do not need to be specially treated for first-order motion but only second order derivative $\boldsymbol{\gamma}_{tt}$. Further [30] shows that $\mathbf{\Gamma}_t^w$ is given by

$$\mathbf{\Gamma}_t^w = \frac{\boldsymbol{V} \cdot U_1}{\rho K^t \|U_1\| \|\boldsymbol{\gamma}\|^2}\boldsymbol{\gamma} \tag{1.15}$$

where K^t is the transversal curvature of $\mathbf{\Gamma}(s,t)$ on surface $\mathcal{M}$. The second term of $\boldsymbol{\gamma}_{tt}$ becomes

$$\frac{e_3^\top(\mathbf{\Gamma}_t^w)}{\rho}\boldsymbol{\gamma}_t = \frac{\boldsymbol{V}\cdot U_1}{\rho^2 K^t \|U_1\| \|\boldsymbol{\gamma}\|^2}(e_3^\top\boldsymbol{\gamma})\boldsymbol{\gamma}_t. \tag{1.16}$$

Note the degenerate case when the viewing ray is along a low curvature surface, then the curve can move significantly. But otherwise, for far way objects

$$|\frac{e_3^\top(\mathbf{\Gamma}_t^w)}{\rho}| << |\frac{(-2\boldsymbol{V}_z)}{\rho}|.$$

Therefore,

$$\boldsymbol{\gamma}_{tt} = \frac{(-2\boldsymbol{V}_z)}{\rho}\boldsymbol{\gamma}_t$$

for fixed curves as well as far way objects. Under the assumption of far-away objects the occluding curves behave like fixed curves up-to second-order. With this theoretical background, the work in this paper is discussed further.

1.3 Detailed Overview of Approach

The work in this thesis presents a novel paradigm to use curves/edges for segmentation of objects in a monocular sequence obtained by a moving camera. This paradigm consists of two stages:

Stage I This stage uses curves/edges to segment different objects based on motion from two or less views. This stage handles two cases differently:

- (a) Segmentation using curves from two-views.
- (b) Segmentation using edges using model for background edges.

Stage II Use multiple frame consistency on segmentations from Stage I to improve the segmentation from Stage I.

1.3.1 Stage I(a): Segmentation based on curves from two views

This work is based on finding correspondence of the extracted contour fragments on every image, in contrast to traditional approaches which rely on feature points, regions, and unorganized edge elements. Consider curve fragments extracted in two views which are of the order of several hundred. Each curve in one image can potentially match to any other curve in the image. This could lead to combinatorial explosion. So therefore the potential candidates for a match in the next frame are pruned by *(i)* considering curves with in certain neighborhood of the curve under consideration and *(ii)* if the color along the one of the two sides of the curves are not very different. There are two tasks: one to find the correct curve-to-curve match and to find the correspondence of the samples on the curves. If the two curves, $\boldsymbol{\gamma}(s,t_0)$ and $\boldsymbol{\gamma}(s,t_1)$ from different frames t_0 and t_1 respectively match, $\boldsymbol{\gamma}(s,t_1)$ can be written as

$$\begin{aligned}\boldsymbol{\gamma}(s,t_1) &= \boldsymbol{\gamma}(0,t_0)+\boldsymbol{\gamma}_s(s)+\boldsymbol{\gamma}_t(t_1-t_0)+\tfrac{1}{2}(\boldsymbol{\gamma}_{ss}s^2+2\boldsymbol{\gamma}_{st}s(t_1-t_0)+\boldsymbol{\gamma}_{tt}(t_1-t_0)^2)\\ &= \boldsymbol{\gamma}(s,t_0)+\boldsymbol{\gamma}_t(t_1-t_0)+\tfrac{1}{2}(2\boldsymbol{\gamma}_{st}s(t_1-t_0)+\boldsymbol{\gamma}_{tt}(t_1-t_0)2)\end{aligned} \tag{1.17}$$

Note that the $\boldsymbol{\gamma}_t$ and $\boldsymbol{\gamma}_{tt}$ are global 2D transformation for the whole curve. Since the alignment is independent of these two terms, the alignment or correspondence between the two curves is given by minimizing the $\boldsymbol{\gamma}_{st}$ as

$$\min_{s_1}\int_s |\boldsymbol{\gamma}(s_1,t_1)-\boldsymbol{\gamma}(s,t_0)| = \min_{s_1}\int_s |\boldsymbol{\gamma}_{st}(s-s_0)(t_1-t_0)| \tag{1.18}$$

where l is the length of the curve. This minimization given below is implemented using dynamic programming as in [77].

$$\int_s |\boldsymbol{\gamma}_{st}|ds = \int_s |g_t\boldsymbol{t}+g\theta_t\boldsymbol{n}|ds \tag{1.19}$$

This notion of similarity between pairs of curve fragments appearing in two adjacent frames is developed and used to find the curve correspondence. This notion of similarity minimizes

1. the extent of a curve along the depth, and
2. the displacement of the curve along the depth.

The curve fragments undergo transitions from one frame to another such as breaking of a curve into two or forming a T-junction. The algorithm in [77] is modified in order to handle these transitions. But these transitions could explode combinatorially if the edge-linking is unstable which can be due to poor resolution of the video or the linker itself. This limits the matching of the curves to adjacent frames rather than extending it to multiple frames. Top row of Figure 1.12 shows the results of curve matching from frame to another. The retrieved curve correspondence is then used to estimate the transformation either affine or similarity. This transformation is good for small motion of far away objects and the model is then used group curves in each frame into clusters based on the pairwise similarity of how they transform from one frame to the next. Results on video sequences of moving vehicles show that using curve fragments for tracking produces a richer segregation of figure from ground than current region or feature-based methods. This yields a performance rate of 85% correct correspondence on a manually labeled set of frame pairs. The main advantage of this approach is rich and well delineated object segmentation. The limitations of this approach are *(i)* would decompose the segmentation when the object is moving along the depth, *(ii)* the segmentations based on two frames can be spurious and *(iii)* requires good quality videos.

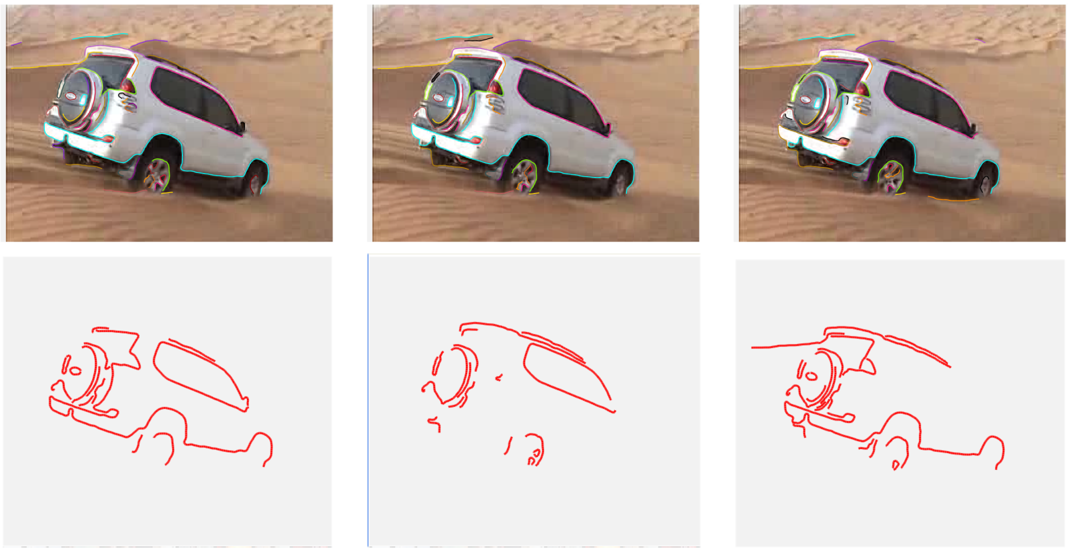


Figure 1.12: Top row shows the matched curves using adjacent frame. Note that the same colors denote the correct match and for a curve which didn't find any match is colored black. The bottom row shows the motion segmentation from the matches of the top row. Observe how the segmentation is missing few curves due to spurious matching.

The first limitation is discussed in Chapter 5. The second limitation is handled in stage II where multiple frames are used to enrich the pair-wise segmentation and a partial solution (only for static/registered low quality video) to the third limitation is discussed next.

1.3.2 Stage I(b): Segmentation based on edges in a registered/static scene

The curve extraction is unstable for aerial videos where the object of interest is small in size. This instability refers to numerous transition taking place from one frame to another. The curve linkers tend to link largely different edges in each frame and resulting in very different curves. The approach discussed in previous section would fail due to numerous transitions. This motivates use of edges for such aerial videos. This work describes an approach of detecting foreground edges by modeling the registered/static background edges. The aerial videos are easier to register as compared to the rest as the background can be modeled largely by a plane [70] or in case of non-planar scene, [24] could be used to register.

Methods for the analysis of moving objects in video sequences obtained from stationary cameras, *e.g.*, for surveillance and monitoring, typically model the stationary background and detect moving objects as those pixels which do not fit this model. Background modeling using multiple distributions is used to handle images with slowly moving objects, slight lighting variations, and repetitive object movements [86, 51, 74, 57, 63]. The most popular schemes use the Mixture of

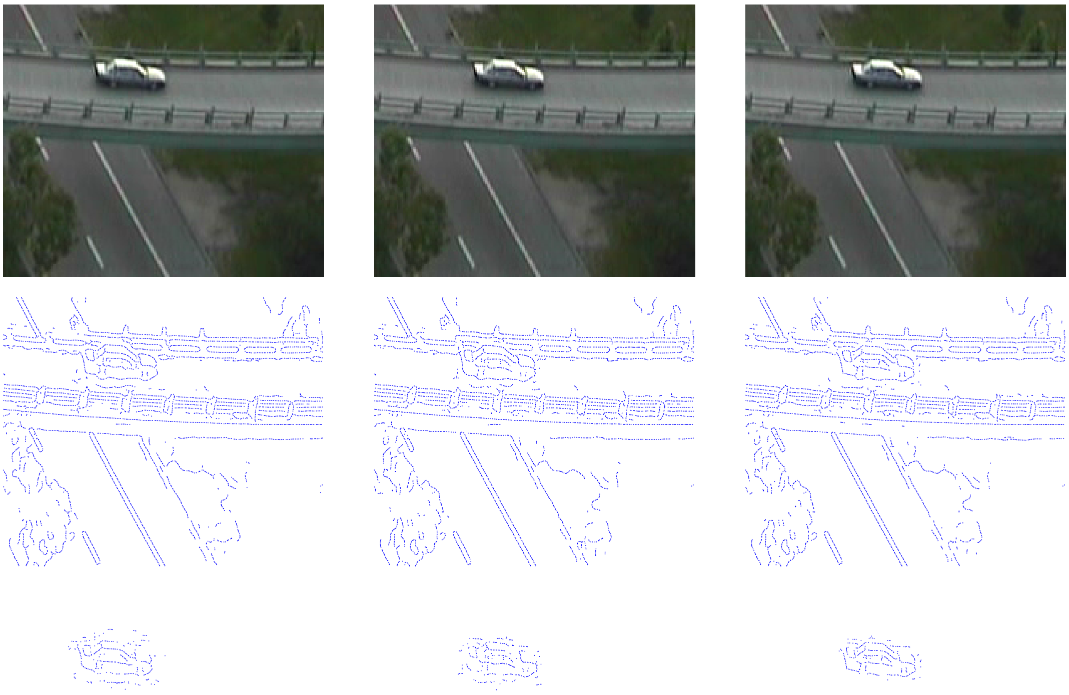


Figure 1.13: Top row shows an aerial video sequence and its edge map is shown in the middle row. The foreground detections are shown in the last row.

Gaussian (MoG) model for each pixel. The intensity at each pixel is modeled using a fixed number of Gaussians which are updated on every observation. Any pixel which is unlikely to come from the MoG is classified as foreground. A key limitation of intensity and intensity gradient background models is that background models do not take spatial interactions into account. Alternatively, edge maps tag those background pixels which maximize local gradient in a neighborhood of pixels. Modeling of sub-pixel edges is robust to sudden illumination changes and as well as it is more selective and hence uses fewer frames as compared to intensity based approaches. It is shown in Section 4.2 that edges are least variant to illumination changes.

The sub-pixel edges are modeled by a mixture of 3D Gaussian $\chi(x, y, \theta)$. Here (x, y) are the sub-pixel positions of the edges and θ is the orientation of these edges. But θ is modeled only for $[0, 2\pi)$. Extra care has been taken to ensure the circular nature of the range *i.e.*, distance between 0 and 2π is zero. Another important thing to note is that the since edges are sub-pixel, the indexing for MoG's is lost. Therefore, the MoG's for a sub-pixel edge are stored on the four neighboring pixels of the edge.The qualitative results shows the superiority of using edges over intensity and gradient which is also evident in the ROC curve shown in Figure 1.14. Figure 1.13 shows the foreground detection of edges from a video scene. The limitations of this approach are *(i)* detections of spurious edges and missing gaps and *(ii)* requires static background. The first limitation is handled by the Stage II which is discussed next but the second will remain.

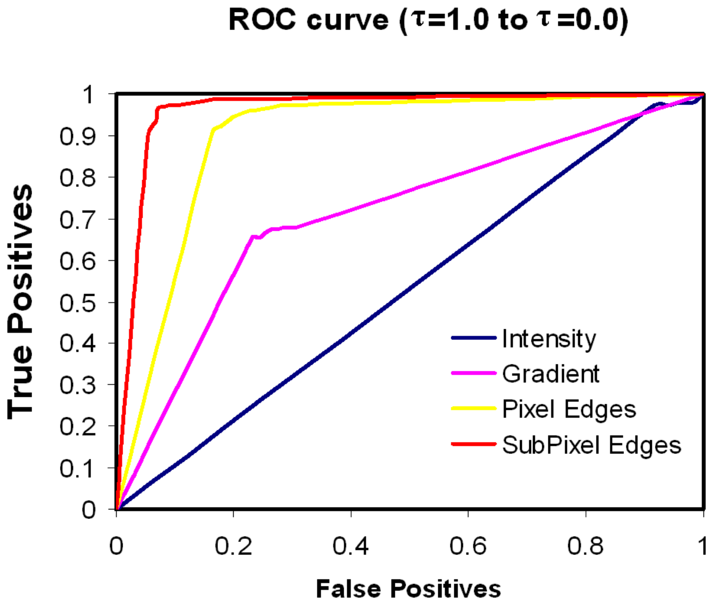
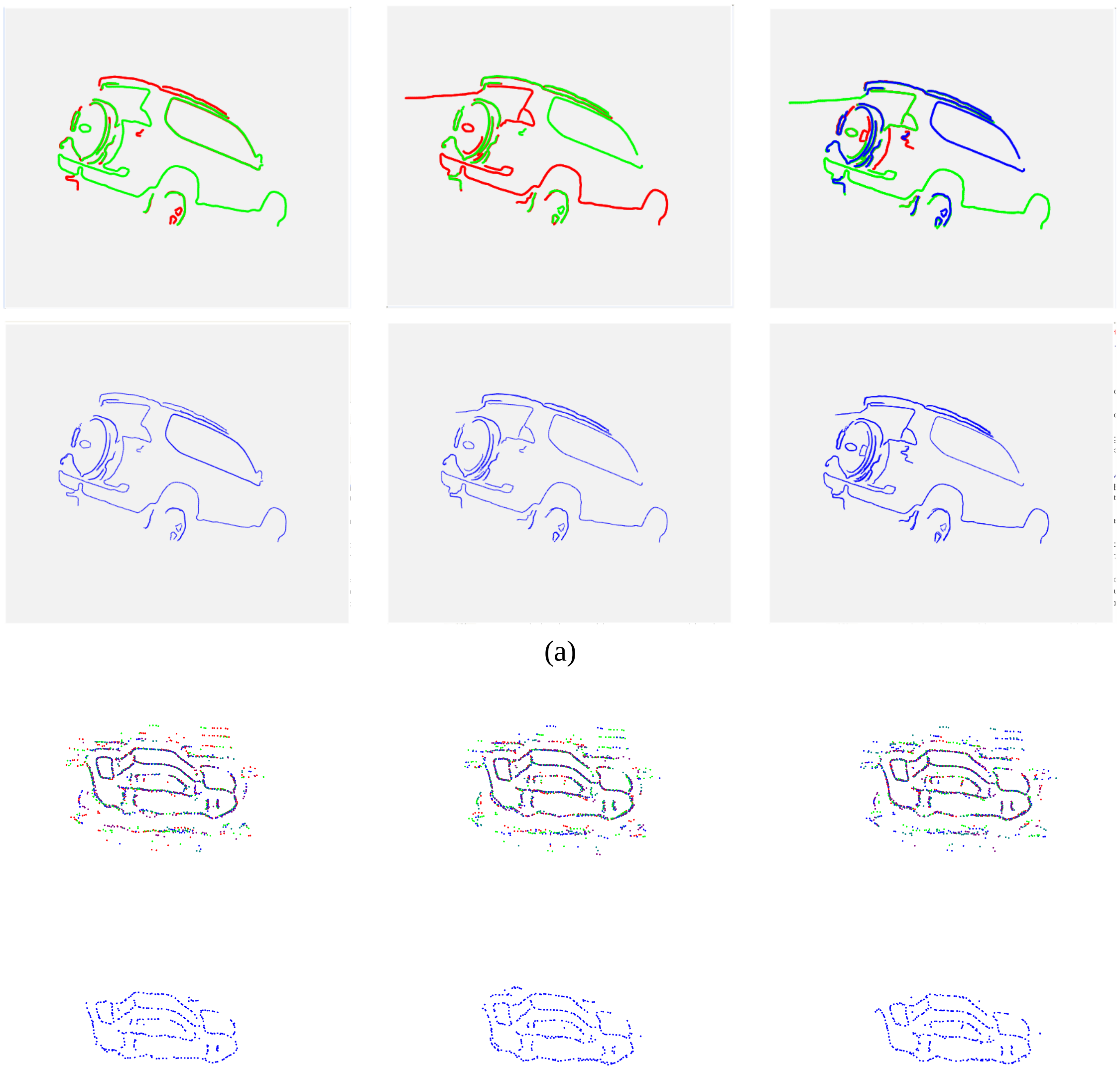


Figure 1.14: ROC curve obtained for foreground detection using Intensity (Black), Gradient (Pink), Pixel Edges (Yellow) and Subpixel Edges (Red) for sequence Figure 4.6

1.3.3 Stage II: Multiple-frame consistency

The detection of curves/edges belonging to the foreground can be very challenging due to the transitions of curves, blending of foreground with background, inter-reflections from surrounding objects, partial occlusions, *etc.*, leading to missing edges, spurious edges from background, highlights, and missing edges due to partial occlusion, *etc*. This renders the segmentation of curves and edges, Figure 1.12 and Figure 1.13, unreliable and unusable. This work proposes an approach to this problem by integrating information across multiple adjacent frames (typically 5 or 7 frames). The input to the approach in this work was obtained from the segmentation sequence using curve-based segmentation [48] and edge-based segmentation [47]. Note the detections from individual frames are generally successful in obtaining the object to a good extent. However, there are also spurious edges and missing gaps due to the factors mentioned previously. These problems are even more significant for low-resolution imagery. There are couple of assumptions required for this work: *(i)* small



(a)

(b)

Figure 1.15: (a) Top rows of (a) and (b) shows the alignment of neighboring edge-maps from segmentation shown in Figure 1.12 and Figure 1.13 respectively onto each frame. Bottom rows of (a) and (b) shows the pruned edge-maps using geometric consistency. Note the difference in segmentation obtained after multiple-frame consistency.

inter-frame motion and the object is not too close to the camera and *(ii)* the surface of the object is considered to piecewise planar. The second assumption is a relaxed version of the assumption of each curve being planar to the object being piecewise planar. The above two assumptions are necessary to assume an affine flow which is a linear model in a local neighborhood in the image given by

$$\chi(\xi, \eta) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}. \tag{1.20}$$

where (ξ, η) are the image coordinates and $\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \begin{bmatrix} e \\ f \end{bmatrix}$ are the affine parameters. Some of the object would consists of multiple planes in case of boxy looking objects and some of the objects with smooth surfaces would be modeled by a number of local planar surfaces. The image flow for the object then consists of multiple local linear models. In order to obtain a common transformation consisting of local linear models the functional given by ,

$$\iint_{(\xi,\eta)} \left(\frac{\partial^2 \chi}{\partial \xi^2}\right)^2 + 2\left(\frac{\partial^2 \chi}{\partial \xi \partial \eta}\right)^2 + \left(\frac{\partial^2 \chi}{\partial \eta^2}\right)^2 \partial\xi\partial\eta. \tag{1.21}$$

is minimized. The solution to this minimization is given by Thin Plate Spline[11]. This spline model has two parts, one is affine and second component is number of radial basis function with kernel of the form $r^2 \log r$, where r is the distance between two points. This spline model gives us a transformation from one frame to another .

First step is to align the edge-maps onto the central frame $I(t)$ to form a “spatio-temporal” compound edge-map. . A moving window of size n (n is typically 5 or 7 frames) is considered at each frame. This means at frame t, the edge-maps from $\{I(t-n), I(t-n+1), ..., I(t-1), I(t+1), ..., I(t+n)\}$ are considered. The alignment of the edge-maps is done by estimating the thin plate spline model between pair of fames. This enables us to “transport” temporal information into the central frame. This superimposed map behaves as a voting space for all the edges. The spurious structure or noise would inconsistent and less voted and missing gaps in the frame would be filled in by other frames.

Second step is to use geometric consistency from [92] of these spatio-temporal edges to distinguish structural from spurious edges. This geometric consistency forms local grouping of edges, called *curvelet bundle,* with constant curvature curve model typically over a neighborhood of 7×7 pixels. These curvelet bundles would consist of edges from different frames and the number of the frames contributing to these local groupings. The higher number of frames contributing to a curvelet bundle, the more likely it would be classified as structure. Note that spurious edges would be ruled out during the formation of curvelet bundles. The curvelet bundles with edge from fewer frames are pruned and an enriched edge-map of the object is obtained. Due to the immense difficulty of obtaining ground truth to quantify the performance of the proposed approach, a synthetic sequence is used to provide quantitative analysis which shows significant improvement over single frame segmentations in Figure 1.16. The qualitative comparisons on real video data shows that the resulting composite edge map is significantly better.

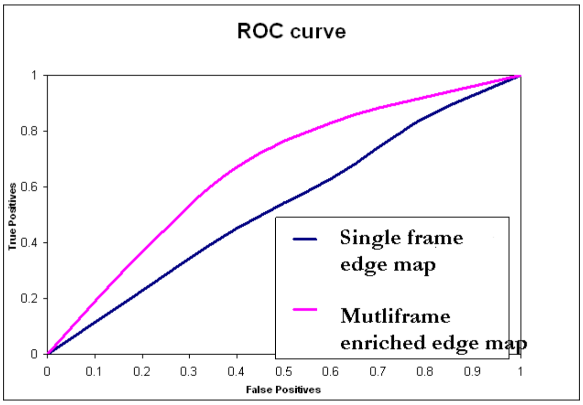


Figure 1.16: ROC curve for comparing figure edge-maps after using multi-frame consistency (shown in Pink) with raw figure-edge maps (shown in blue).

After this stage the limitation remaining is that the object moving along the depth close to the camera cannot be segmented properly. This limitation can be redeemed by estimating 3D motion of the objects and grouping is based on 3D motion.This is further discussed in the next section.

1.3.4 Extension to 3D motion

Since the above system uses motion models in 2D as approximation to 3D motions, they suffer from limitation of breaking up an object close to the camera moving along the depth into different segments. The inevitable solution to this problem would be estimating 3D motion.

The work in this thesis studies the feasibility of the extension of motion model to 3D. First, 3D motion and geometry estimation of curves from in terms of the second-order derivatives of the one-parameter family of curves, $\gamma(s, t)$, which has also been derived by Faugeras[31]. $\gamma(s, t)$ is assumed to observed for a rigidly translating fixed curve relative to the camera which suffices for the scope of this work The magnitude of the translation is not constrained due to unknown depth. The 3D translation direction is estimated as one-parameter family at each edge, γ_0, and its local grouping of edges $\gamma(s, t)$.

At-least two such edges are needed to estimate translation direction. And definitely each object has many more edges available. The lower bound for error in translation direction is estimated as a function of error in measurements. This lower bound was plotted for some typical values of the system. The plots show high amount of errors rendering unreliable estimation of translation direction and hence, unreliable segmentation. This limitation of estimating 3D motion shows that the failure to segment close by objects moving along the depth.

1.4 Contribution

The contributions of this work are

1. Given a high-resolution high frame-rate video sequence of multiple moving objects and a moving camera, segments the objects using curve correspondence across frames and common motion cues, Figure 1.12. This work appeared in Computer Vision And Image Understanding, 2007, [48].

2. Developed a notion of edge-map background model which for the first time used *(i)* edges, and *(ii)* their orientation and *(iii)* sub-pixel position in training the background model from videos of stationary cameras or registered video images. The resulting background model is adaptable to changes in the scene as it can be trained with significantly fewer frames and is substantially more robust to a wide range of illumination changes. This has partially appeared in International Conference on Image Processing, 2007, [47].

3. Developed a multi-frame integration framework to enrich single-frame motion segmentation results, which are typically incomplete with spurious structure due to limitation of motion segmentation algorithms, occlusions, highlights, and other visual transforms. The enrichment is based on the consistency of local differential geometry, based in tangent and curvature of curve bundles at each edge, in several adjacent frames: if the same local structure appears in a majority of frames, the structure is considered non-accidental. This requires a registration of edges in adjacent frames based on a local planarity assumption of 3D structure which is formally shown to lead to a thin plate spline registration scheme so that the curvature can be compared. The resulting enriched motion segmentation eliminates spurious structure and completed gaps, as shown in Figure 4.10.

4. Showed that 3D motion segmentation is required for objects with motion along the depth relative to the depth. It is further show that it is infeasible to recover the full 3D motion of the object.

1.5 Organization of the thesis

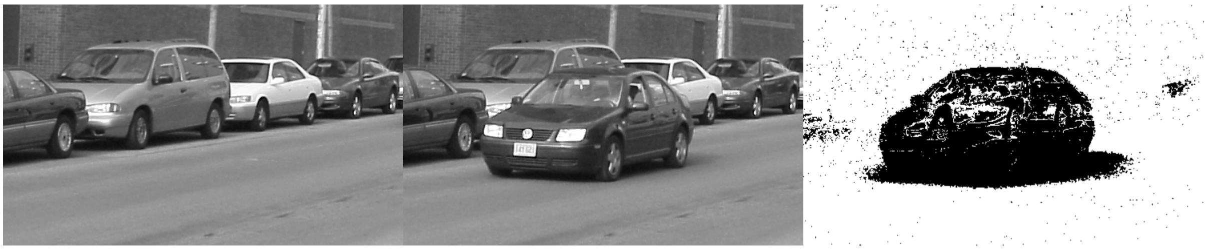
Chapter 2 discusses the existing approaches for motion segmentation and discusses their merits and the limitations. The first component of Stage I, motion segmentation using curves, is discussed in Chapter 3 followed by second component of Stage I, foreground detection using edges, in chapter 4. This is followed by discussion of Stage II approach in Chapter 5. Each of these chapters discusses the approach and performance of each of these stages individually. An attempt to extend the approach to use 3D motion is discussed in Chapter 6.

Chapter 2

Related Work

2.1 Static Camera

Under the simplistic assumption of static camera, the solution to the above problem in the literature exists by modeling the static background and identifying any deviation from the model as the foreground [86, 51, 74, 57, 63]. These approaches are based on modeling intensity of the background and detects the moving object as foreground if its intensity is different from the background. These approaches are robust to gradual change in illumination but misfires due to sudden change in the illumination. This idea of background modeling has been extended to the case of moving camera but requires knowledge of the intrinsic as well as extrinsic parameters [72].



(a) (b) (c)

Figure 2.1: The segmentation of object viewed from a static camera using [86]. (a) shows a typical background frame and (b) shows image (a) with a moving object and the segmentation of the moving object is shown in (c) (in black color).

Algorithm 1: Motion segmentation based on dense image flow

```
1: Compute Dense flow using optical flow approaches.
2: Fit a parameteric model (typically affine model) to the optical flow in local regions.
3: Define the affinity between different regions based on the model as well as other cues.
4: Use a region-growing or graph based approach to merge the regions.
```

Algorithm 2: Motion Segmentation based on Factorization of trajectories of features.

```
1: $W_{2F\times N} \leftarrow$ trajectories of $N$ features over $F$ frames using KLT/SIFT/any other feature.
2: $W$ has a rank of $4n$ where $n$ is the number of different motions (typically affine).
3: Different approaches for segmenting $W$ in this 4D manifold.
```

2.2 Non-static camera: Dense flow based

Since numerous approaches use optical flow to segment independently moving objects, optical flow techniques are briefly reviewed.

2.2.1 Optical Flow

Let $I(\xi(t), \eta(t), t)$ be the intensity of a 3D point $\mathbf{\Gamma}$ at time t, where (ξ, η) are the image coordinates, projection of $\mathbf{\Gamma}$. The total derivative of $I(\xi(t), \eta(t), t)$ is given by

$$\frac{d}{dt}I(\xi(t), \eta(t), t) = \frac{\partial I}{\partial \xi}\frac{\partial \xi}{\partial t} + \frac{\partial I}{\partial \eta}\frac{\partial \eta}{\partial t} + \frac{\partial I}{\partial t} = \nabla I.\boldsymbol{\gamma}_t + \frac{\partial I}{\partial t} \tag{2.1}$$

where $\boldsymbol{\gamma}(t)$ is the image velocity of $\mathbf{\Gamma}$. In order to get a constraint, the intensity is assumed to be constant over infinitesimal range of time, *i.e.*, $I(\xi(t), \eta(t), t) \simeq I(\xi(0), \eta(0), 0)$ or $\frac{d}{dt}I(\xi(t), \eta(t), t) = 0$. This leads to the constraint equation also known as brightness constancy equation given by

$$\nabla I.\boldsymbol{\gamma}_t + \frac{\partial I}{\partial t} = 0. \tag{2.2}$$

But the above equation has two unknowns $\boldsymbol{\gamma}_t$ and only one constraint. The component of $\boldsymbol{\gamma}_t$ along the gradient ∇I is called normal velocity, denoted by $\boldsymbol{\gamma}_t^{\boldsymbol{n}}$ can be recovered, using

$$\nabla I.\boldsymbol{\gamma}_t = \boldsymbol{\gamma}_t^{\boldsymbol{n}} = -\frac{\partial I}{\partial t}/\|\nabla I\| \tag{2.3}$$

The second component of $\boldsymbol{\gamma}_t$ perpendicular to the gradient direction, denoted by $\boldsymbol{\gamma}_t^{\boldsymbol{t}}$ is still free. This is the well-known “aperture problem”. This shows that the second component $\boldsymbol{\gamma}_t^{\boldsymbol{t}}$ of image velocity cannot be computed without additional information or assumptions. The flow obtained using brightness constancy is called optical flow which approximates the true motion field in an image. An important note is that the brightness constancy assumption holds only for translating

Lambertian surface as discussed in [31]. The mainstream solution given in the literature is to resort to smoothness constraints. This assumption of smooth motion field does not hold at the boundary between objects which are either at different depths or moving differently or both. The motion fields can be very different at such boundaries and smoothing leads to the deviation from the actual motion field. The optical flow computation techniques relevant to the scope of this work can be categorized based on their assumptions as *(i)* smoothness-based, *(ii)* parameteric models-based and *(iii)* contour -based.

Smoothness-based: Lucas-Kanade [61] approach also called *local approach* consists in assuming a window W moving uniformly, *i.e.*, all pixels within the window are assumed to have the *same* flow vector, so that an over-constrained system is obtained in two unknowns. A least-squares problem is solved minimizing brightness constancy residual over all pixels of the window.

$$\sum_{x\in R} W^2(\xi,\eta)(\nabla I(\xi,\eta,t).\boldsymbol{\gamma}_t + I_t(\xi,\eta,t))^2 \tag{2.4}$$

where $W(x)$ is the window function, R is the neighborhood and v is estimated in a close form. This least-squares problem is reduced to a linear system of equations but the data matrix can be singular if $I_x = 0$ or $I_y = 0$ (aperture problem) or both (homogeneous region). The problem with the above approach is that a simple motion like translation along optical axis would give widely varying flow vectors and the assumption of flow vector in a window would break down. [14] assumes first-order variation in flow over a small neighborhood and few other variations of the above approach are [98, 17, 84] . These local assumptions fail in presence of multiple motions within the neighborhood.

The Horn-Schunk approach [43] also called *global approach* tries to find a smooth field $\boldsymbol{\gamma}_t(\xi,\eta)$ over the image whose component along the direction of the image gradient is as close as possible to the measured optical flow $\boldsymbol{\gamma}_t^{\boldsymbol{n}}$. This can be expressed mathematically as the following minimization problem [31]

$$\min_{\boldsymbol{\gamma}_t} \int\int \left\{ (\boldsymbol{\gamma}_t \cdot \frac{\nabla I}{\|\nabla I\|} - \boldsymbol{\gamma}_t^{\boldsymbol{n}})^2 + \lambda \mathrm{tr}[D\boldsymbol{\gamma}_t(D\boldsymbol{\gamma}_t)^\top] \right\} d\xi\, d\eta, \tag{2.5}$$

where D is the differential of the function $\boldsymbol{\gamma}_t(\xi,\eta)$ given by $D = \left[\begin{array}{cc} \frac{\partial\boldsymbol{\gamma}_t}{\partial\xi} & \frac{\partial\boldsymbol{\gamma}_t}{\partial\eta} \end{array}\right]$. The criterion (2.5) is the sum of two terms: the first term imposes that the component of the “invented” field along $\boldsymbol{n}$ is as close as possible to the measurements, and the second term controls its smoothness through the parameter λ. This approach is only applicable for scenes with no discontinuities in depth and smooth motion which is not true for a wide range of videos.

Parameteric models-based: Instead of using smoothness constraint several approaches assume the nature of the surface of the objects in the scene such as planar surface or curved surface which gives a parametric model valid in a neighborhood of normal velocities as proposed in [97, 101, 67, 8]. Bergen *et al.*[8] shows image flow to be second order polynomial under the assumption of a planar surface and also extends the model to rigid motion. [101] extends the work of [8] further by assuming the scene to be a quadric surface and fits a cubic polynomial to the motion field $\boldsymbol{\gamma}_t(\xi, \eta)$.

Contour-based: Another interesting class of optical flow methods computed flow on the edges or along contours due to high gradient values and hence higher SNR. Hildreth [42] proposes a smoothness constraint along the contour given by minimizing the following integral

$$\int [(\frac{\partial \boldsymbol{\gamma}_t}{\partial S})^2 + (\boldsymbol{\gamma}_t \cdot \boldsymbol{n} - \beta)^2] dS \tag{2.6}$$

where $\boldsymbol{\gamma}_t$ is the image velocity along the contour S. This idea is similar to [43] except that flow is computed only along the contour and The constraint in Equation 2.5 is applied along a contour so as to avoid problems of low gradient overwhelmed by noise. Another approach in [13] applies spatio-temporal filters to edge-maps in order to measure image velocity. Although they provide better accuracy due to high gradient but the motion field is very sparse.

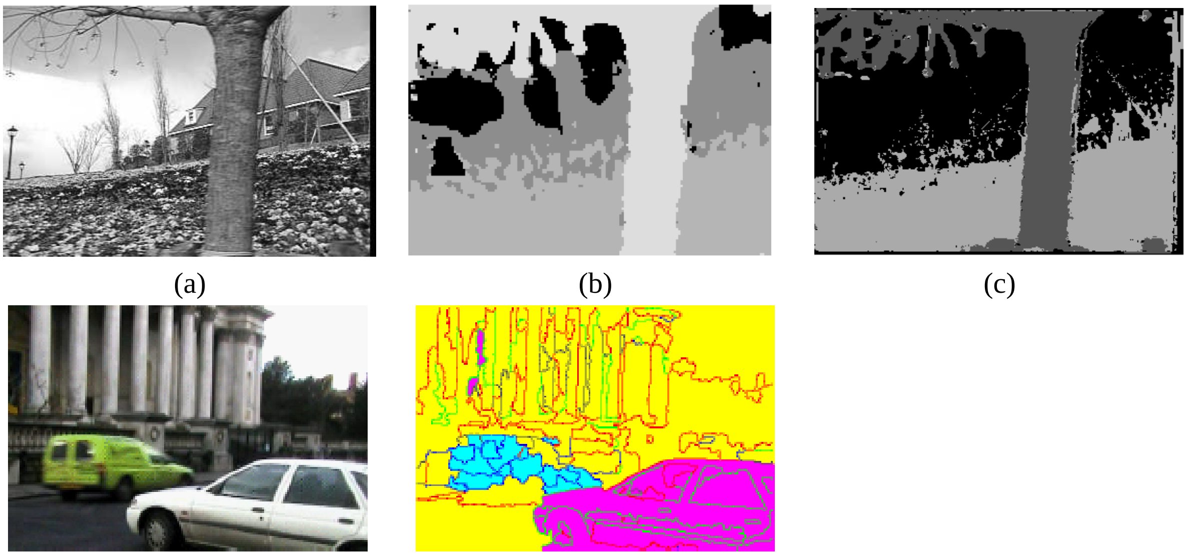
Longuet-Higgins *et al.* [60] do not assume brightness equation but provide relationship between scene geometry, motion of the camera relative to the object and the image flow under the assumption of rigidity,

$$\boldsymbol{\gamma}_t = \frac{\boldsymbol{V}}{\rho} - \frac{\boldsymbol{V}_z}{\rho}\boldsymbol{\gamma} + \Omega_\times \boldsymbol{\gamma} - (\boldsymbol{e}_3^\top \Omega_\times \boldsymbol{\gamma})\, \boldsymbol{\gamma}, \tag{2.7}$$

where $\boldsymbol{V}$ and Ω are the first order derivatives of translation and rotation of the scene, ρ is the depth and $\boldsymbol{\gamma}_t$ is the image velocity. Nagel [69] further showed the difference between optical flow and image flow. But the above approach requires knowledge about 3D scene which is not available generally. Faugeras [31] altogether discards brightness constancy equation and constrain the image flow using second order derivatives of the image and 3D model parameters. The brightness constancy holds for objects with Lambertian surface translating parallel to the image plane which is only an approximation to the true motion field.

2.2.2 Motion Segmentation based 2D Motion

The motion field is further used by numerous approaches to segment images based on their motion into different object. These approaches fit parameteric models to 2D optical flow and group the pixels that best fits a model. Such group of pixels are labeled as a ”layer”. The assumption here is that each ”layer ” would give a different object which is definitely false. It is unclear what is the nature of such segmentation as it breaks down in usual cases like when an object is moving along the depth, dynamic scene with independently moving objects, *etc.* Such algorithms either try to find *dominant motion* and then fit the outliers with different motion model or fit different motion models simultaneously.



(a) (b) (c)

(d) (e)

Figure 2.2: (a) frame from ”Wallflower” sequence, (b) segmentation results from [4], (c)segmentation results from [6] and (d, e) segmentation result from [85].

In[85], the number of different motions, n to be segmented is known and typically $n = 2$. The authors fit one of the n 2D affine models to every edge using an EM optimization. Each edge is assigned probability of belonging to each motion model. The region bounded by edges are labeled based on their collective likelihood of belonging to one of the motion models and are depth-ordered in a Bayesian framework. Similarly [94] fits at most two motion models in each fixed size block or colour based segmented image regions. These regions are grouped according to the similarity of motion models and also their depth ordering is resolved based on the magnitude of motion. [9] uses regions from watershed segmentation. These methods work generally when the objects are not only planar but fronto-parallel as the results shown in Figure2.2 (d,e).

[66] proposes a region-merging algorithm based on spatial and temporal similarity. The regions are obtained from an over-segmented image and connected through a weighted and directed graph. The weights represent spatio-temporal similarity. The clustering is done by first clustering cycles in this graph followed by greedy grouping of the remaining nodes. This algorithm does not need to know about number of motions but instead has a threshold which has a similar effect. Another graph based algorithm proposed [82] forms an undirected weighted graph where weights are the temporal similarity between different regions. N-cuts is used to partition into salient regions recursively. This requires the knowledge of number of motions beforehand .

In [6] the authors estimates multiple motions per region and merges the region with similar model parameters. The approach does not require prior knowledge of number of motions or a threshold but instead a scale parameter which decides the granularity of regions. In effect it scale parameter has to be known. In [4] Adelson and Wang fit 2D affine models in fixed size windows in the image and group the regions which have same 2D affine parameters. This results in segmentation of images based on 2D motion which is deceptive as the whole scene might have the same motion but is segmented into different layers as they had different depths, Figure 2.2. All the above approaches described above do not segment objects moving differently but the segmenting based on the projections of their motion. This leads to a lot of ambiguity as different motions in 3D could lead to similar projections and hence the problem becomes ill-posed. In order to avoid such ill-posedness the scene should be segmented based on their 3D motion.

2.3 Non-static camera: Factorization based

Multibody factorization is referred to motion segmentation from trajectories of tracked features of different moving objects. Specifically, if N features, $p = 1, ..., N$ are tracked in F frames $f = 1, ..., F$ as denoted by $x \in \mathbb{R}^2$, and these features are projection of points X_p, $p = 1, ..., N$ on a rigidly moving object, then under the affine projection model

$$x_{fp} = A_f X_p,$$

where A_f is the affine camera matrix at frame f which in matrix form gives

$$W = MS, \qquad W = [x_{fp}]_{2F \times N}, \quad M = \begin{bmatrix} A_1 \\ . \\ . \\ . \\ A_F \end{bmatrix}_{2F \times 4}, \quad S = [X_1, ..., X_p]_{4 \times N},$$

where S is the *structure matrix* and M is the *motion matrix*. [93] and [12] showed that the rank of W is less than or equal to four. This means the trajectories of features per rigidly moving object span a 4D manifold. This idea was further applied to segmenting multiple moving objects by segmenting the trajectories in these 4D manifolds. In case of multiple n objects W has a rank $r = 4n$ which is decomposed using SVD as

$$W = U \Sigma V^\top.$$

Since the rank of W is $4n$ the matrix Σ has $4n$ non-zero entities. The columns of the matrix V corresponding to the $4n$ largest values in Σ is denoted as V_1 and *shape interaction* matrix Q given by $Q = V_1 V_1^\top$. The elements of this matrix Q has the following property

$$Q_{ij} = 0 \text{ if } i \text{ and } j \text{ belong to different motions.} \tag{2.8}$$

The approach proposed in [23] thresholds Q to cluster these subspaces for motion segmentation . This threshold is quite sensitive to noise as shown in [37]. An algebraic framework GPCA [95, 96] casts the problem in a high-degree nonlinear space, but the number of required sample points grows exponentially with the number of subspaces, thus is not suitable for a large number of non-rigid objects or non-rigid motion. Numerous other approaches attempt to find independent motions but are approximated by affine motions in the image which fails miserably, Figure 6.3, when the variation of motion over the object is not affine.

The limitations posed by the dense-flow based approaches as well as the feature-based approaches motivates motion segmentation based on 3D motion.

Chapter 3

Segmentation using Curves

This chapter presents a method for segmentation of moving objects from monocular video sequences using curves. The approach is based on tracking extracted contour fragments, in contrast to traditional approaches which rely on feature points, regions, and unorganized edge elements. Specifically, a notion of similarity between pairs of curve fragments appearing in two adjacent frames is developed and used to find the curve correspondence. This similarity metric minimizes the extent of the curve along the visual ray and the motion along the depth and in addition takes into account both a novel notion of transitions in curve fragments across video frames. This yields a performance rate of 85% correct correspondence on a manually labeled set of frame pairs. Color/Intensity of the regions on either side of the curve is also used to reduce the ambiguity and improve efficiency of curve correspondence. The recovered curve correspondence is the basis of figure-ground segregation. The main assumption is that curves belonging to the same object transform from one frame to another more similarity to each other than to curves from other objects or background. Specifically, from the inter-frame correspondence between each pair of curves a similarity transformation is recovered and a notion of *transform similarity* is defined between two pairs of curves in a common frame based on how they transform in the next frame. This transform-induced similarity matrix is then converted into clusters which define objects and background in an image. The results on video frames of moving vehicles are very encouraging, as previewed in Figure 3.1, and are illustrated on a number of video sequence of moving vehicles. Results on video sequences of moving vehicles show that using curve fragments for tracking produces a richer segregation of figure from ground than current region or feature-based methods.

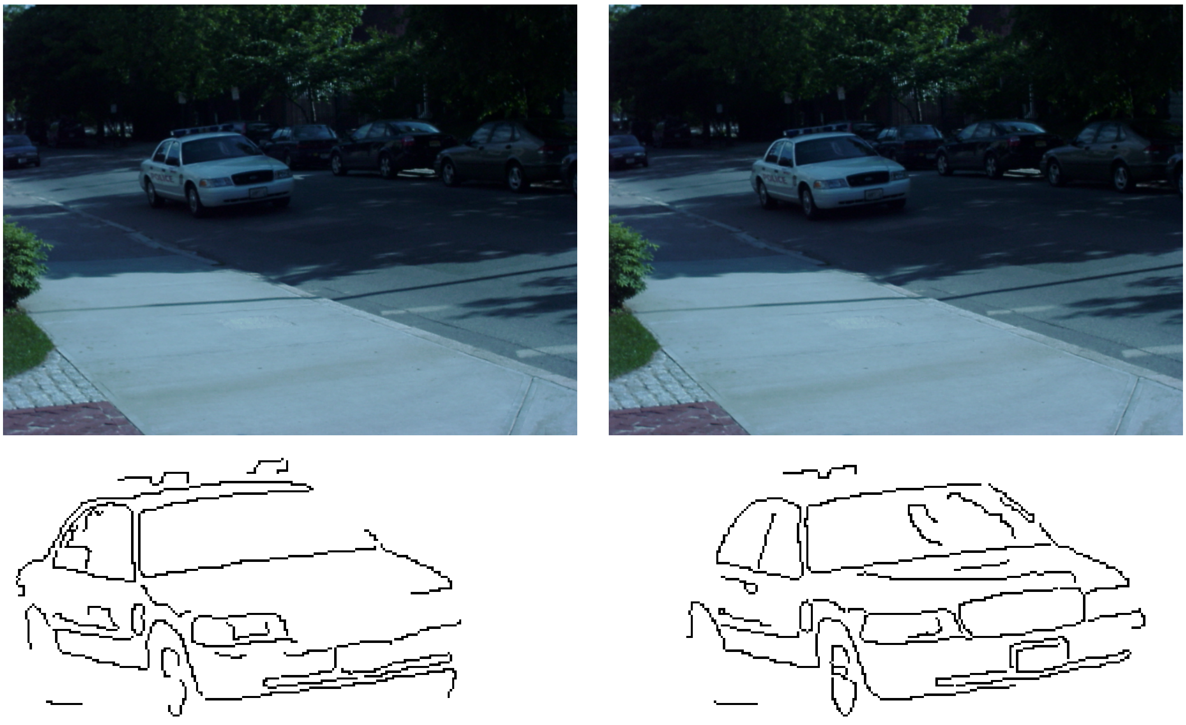


Figure 3.1: The contour fragments of a moving vehicle are segregated from the background using only the two adjacent frames shown.

3.1 Related Work

The approaches for motion segmentation can be loosely organized by the primary spatial dimension of the tracked feature, *i.e.* points [64, 40, 25, 81], curves [53, 44, 35, 45, 29, 36] or regions [56, 39, 71, 22, 33]. Point-based features are typically used in the context of 3D reconstruction, where points are matched across frames on the basis of Euclidean distance and image correlation in a local neighborhood around matching pairs. The epipolar constraint is used to eliminate erroneous matches using robust fitting algorithms such as RANSAC [41]. More recently, there has been considerable interest in regions, where affine invariance derived from intensity operators [58] is used to define salient patches that can be recovered from multiple views of the same surface feature. These *affine patches* can be used for tracking as well as recognition.

There seems to be little work on using geometry of connected edgel chains directly for mtoion segmentation.The edgel chains can be obtained by linking edges obtained from an edge-detector as in [76] or iso-intensity contours [68] with non zero gradient along the contour which are claimed to be robust to illumination changes and do not require any fixed threshold. The closest work is that of Folta *et al.* [35], where they use edge curve matching to form the outlines of moving objects, the key objective of this paper. The algorithmic approach in this chapter is closest to that of Freedman [36] who considered curve tracking as a problem of optimum geometric alignment of detected intensity edges. A key difference from this paper is that Freedman assumes a *model* curve, which is supplied by hand initialization or by learning from a hand-picked set of example curves.

No such assumption is made in the current approach where curves, as segmented, are tracked across frames. In this regard, the tracking process is similar to that for points, *e.g.*, Harris corners, but in addition we exploit the *order* and *continuity* provided by segmented edgel chains. A key reason that edgel curves have not received much attention is that it is difficult to define local correspondences between smooth curve segments. The approach of this work relies on the following key observation: if each extracted curve fragment is sufficiently distinct from other extracted curves in the same frame, and if the inter-frame deformation for each curve fragment is small enough as compared to intra-frame curve differences, then the similarity between curve pairs in the two frames provides a basis for the recovery of curve correspondence by solving an assignment problem. Here the correspondence problem is solved by minimizing the spatial extent of the curve along the visual ray and the motion of the curve along the depth direction using dynamic programming [77]. The choice of similarity metric is discussed further.

3.2 Alignment between two curves

The general form of under linear translation motion equation for a 3D curve both stationary and non-stationary is given by Equation1.13 as

$$\mathbf{\Gamma}(s,t) = \mathbf{\Gamma}(s,0) + \boldsymbol{V}t + \mathbf{\Gamma}^w(s,t)$$

generates one-parameter family of curves in the image given by $\boldsymbol{\gamma}(s,t)$. The $\boldsymbol{\gamma}(s,t)$ can be locally expanded to second order expansion given by

$$\boldsymbol{\gamma}(s,t) = \boldsymbol{\gamma}(s_0,t_0)+\frac{\partial\boldsymbol{\gamma}}{\partial s}(s-s_0)+\frac{\partial\boldsymbol{\gamma}}{\partial t}(t-t_0)+\frac{1}{2}(\frac{\partial^2\boldsymbol{\gamma}}{\partial s^2}(s-s_0)^2+2\frac{\partial^2\boldsymbol{\gamma}}{\partial s\partial t}(s-s_0)(t-t_0)+\frac{\partial^2\boldsymbol{\gamma}}{\partial t^2}(t-t_0)^2) \tag{3.1}$$

where s_0 is the reference point in the reference frame t_0. The image curve in the next frame, t_1, is given by

$$\boldsymbol{\gamma}(s,t_1) = \boldsymbol{\gamma}(s_0,t_0)+\boldsymbol{\gamma}_s(s-s_0)+\boldsymbol{\gamma}_t(t_1-t_0)+\frac{1}{2}(\boldsymbol{\gamma}_{ss}(s-s_0)^2+2\boldsymbol{\gamma}_{st}(s-s_0)(t_1-t_0)+\boldsymbol{\gamma}_{tt}(t_1-t_0)^2) \tag{3.2}$$

and since the curve in frame t_0 can be locally expanded to $\boldsymbol{\gamma}(s,t_0) = \boldsymbol{\gamma}(s_0,t_0) + \frac{\partial\boldsymbol{\gamma}}{\partial s}(s-s_0) + \frac{1}{2}\frac{\partial^2\boldsymbol{\gamma}}{\partial s^2}(s-s_0)^2$, the above equation can be rewritten as

$$\boldsymbol{\gamma}(s,t_1) = \boldsymbol{\gamma}(s,t_0) + \boldsymbol{\gamma}_t(t_1-t_0) + \boldsymbol{\gamma}_{st}(s-s_0)(t_1-t_0) + \frac{1}{2}\gamma_{tt}(t_1-t_0)^2. \tag{3.3}$$

Note that the above equation shows that the transformation between curves in two different frames has a global transformation shared by all the points of the curve in the image, $\boldsymbol{\gamma}_t(t_1-t_0) + \frac{1}{2}\gamma_{tt}(t_1-t_0)^2$, and local transformation given by $\boldsymbol{\gamma}_{st}(s-s_0)(t_1-t_0)$. The global transformation is dependent only on the motion of the 3D curve but the local transformation is a function of the both motion and geometry of the curve. The local correspondence or alignment of the curve is given by this local transformation as

$$\boldsymbol{\gamma}(s,t_1) = \boldsymbol{\gamma}(s,t_0) + \boldsymbol{\gamma}_{st}(s-s_0)(t_1-t_0). \tag{3.4}$$

In absence of any other information, the correspondence, $s_1(s)$ is given by minimizing the difference between $\boldsymbol{\gamma}(s,t)$ and $\boldsymbol{\gamma}(s_1,t_1)$ which means minimizing $\boldsymbol{\gamma}_{st}$ given by

$$\min_{s_1}\int_s |\boldsymbol{\gamma}(s_1,t_1) - \boldsymbol{\gamma}(s,t_0)| = \min_{s_1}\int_s |\boldsymbol{\gamma}_{st}(s-s_0)(t_1-t_0)| \tag{3.5}$$

where l is the length of the curve. Proposition 1.3 shows that $\boldsymbol{\gamma}_{st}$ is given by $g_t\boldsymbol{t} + \varphi_t\boldsymbol{n}$ where g_t is the change in the speed of parameterization (stretching energy) and φ_t is the change in the angle of the tangent of the curve (bending energy). $g_t\Delta t$ is given by $|\frac{ds}{ds} - \frac{ds_1}{ds}|$ where s is the arc-length parameterization in frame t_0. Without loss of generality, $s_0 = 0$ can be assumed. Therefore,

$$\min_{s_1}\int_s |\gamma_{st}(s-s_0)(t_1-t_0)| = \int_{s_1}\left[|1 - \frac{ds_1}{ds}| + |\varphi_t\Delta t|\right] ds \tag{3.6}$$

This minimization gives the the alignment between two curves and once the alignment is known the transformation is computed. This minimization is implemented using dynamic programming as in [77]. To understand the implications of this minimization on 3D motion and structure of the curve, consider expression of γ_{st} related to the 3D shape and motion of the curve. Proposition [] gives

$$\gamma_{st} = (\boldsymbol{V}_z\gamma - \boldsymbol{V})\frac{\rho_s}{\rho^2} - \frac{\boldsymbol{V}_z}{\rho}\gamma_s$$

where $\rho_s = Ge_3^\top\vec{T}$, *i.e.,* third component of 3D tangent of the curve. Minimizing γ_{st} implicates curves with smaller extent along the depth as well $\boldsymbol{V}_z$ smaller motion along the depth relative to the depth of the object. Note that this approximation is good for adjacent frames for faraway objects. The above minimization holds for both stationary and non-stationary curves.

3.3 Curves Transitions

One of the issues of using image curves is their inter-frame transitions such as a curve is split into two curves or vice versa. Numerous transitions are observed from frame to frame which have been illustrated in Figure 3.2. Numerous curves do change their shape slightly which is referred to curve deformation, Figure 3.2(a). And the transitions are classified into simple, Figure 3.2 (b,c,d) and complex transitions Figure 3.2 (e,f,g). The complex transitions are combination of multiple simple transitions. The transitions can be listed as below:

Simple Transitions

(b) A curve splits into two or two curves join into one. This happens mainly due to noise or falling of detection below threshold.

(c) A curve is split into two because of another curve forming a T-junction.

(d) A curve disappears in the next frame due to falling below detection threshold or occlusion.

Complex Transitions

(e) A curve splits into more than two fragments which can be treated as multiple steps of simple transition (b).

(f) When a T-junction curve into 3 pieces of curves and different combination of merging of these pieces causes this transition. This can be modeled as a combination of simple transitions (b) and (c).

(g) This transitions occurs for closed curve as the endpoints are ambiguous. This transition is seen as two steps of (b) by first breaking a curve into two and then merging the two curves resulting in different end-points.

3.4 Curve Tracking via Transition-based Elastic Matching

In this section we describe a similarity-based method for finding the correspondence between contour fragments in two video frames. Specifically, we first describe how contour fragments are extracted from each frame, then describe how a correspondence is obtained from a pairwise elastic similarity of these curve fragments, and finally describe three modifications to induce the notion of transitions and the vanishing point constraint.

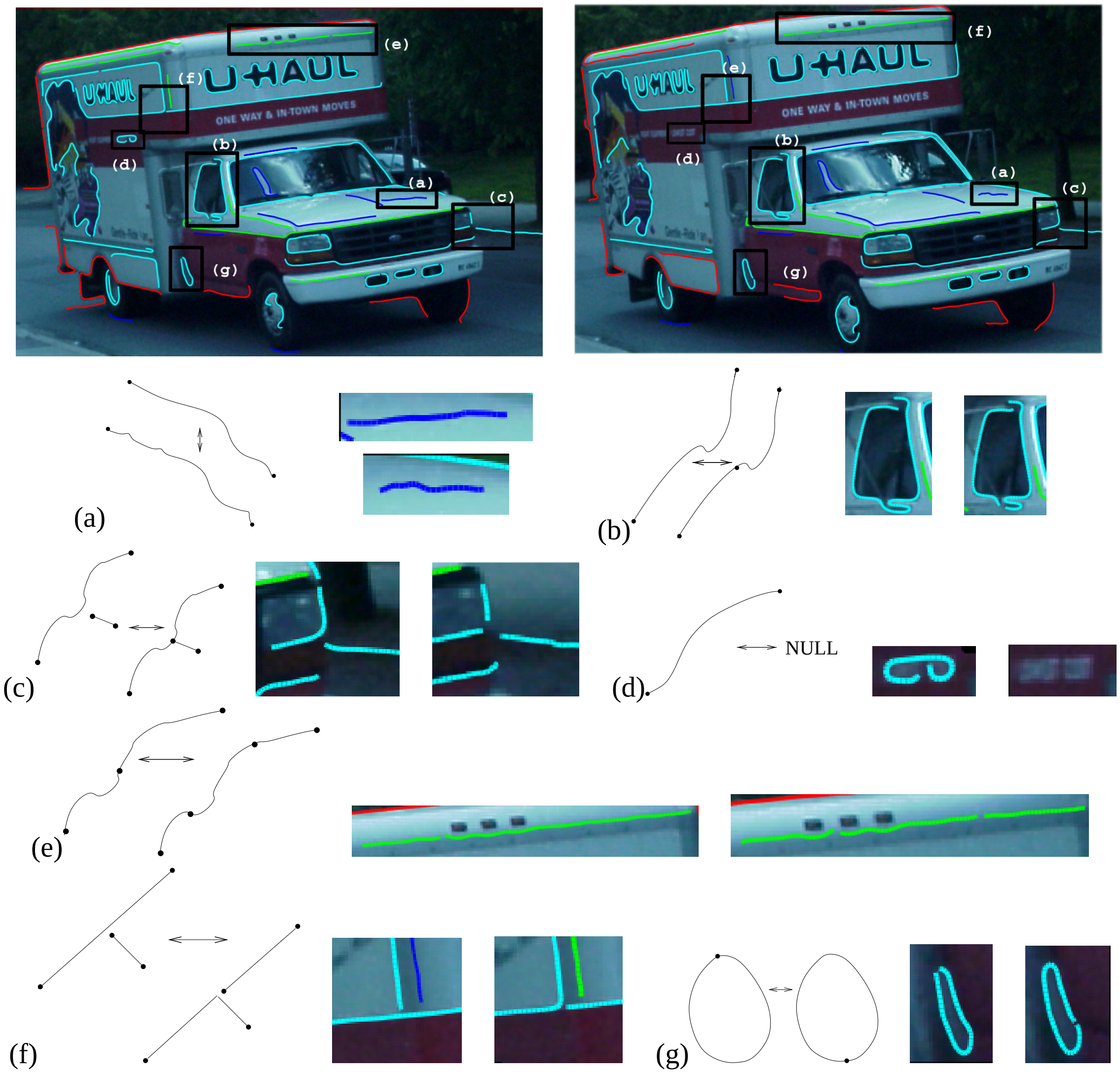


Figure 3.2: Typical changes in curve fragments extracted from two frames of a video sequence using the topologically-driven edge operator [76] are illustrated for two frames of the UHAUL sequence. Typically, about half of curve fragments change smoothly as illustrated in (a). However, the remaining half can be expected to undergo abrupt changes as classified into six *transitions*: (b) a curve fragment can be split into two, or two can be joined into one. (c) The formation or disappearance of a T-junction. (d) The complete disappearance or appearance of a curve. (e) Compound fragmentation when two curve fragments join and split differently, a combination of two transitions of type “b”. (f) Compound T-junction, a combination of transitions “c” and “b”. (g) Compound fragmentation of closed curves, a combination of two transitions of type b.

3.4.1 Extracting Contours

The contour detector used in these experiments is based on a modification of the Canny algorithm [15] as proposed in [76]. As is well known, the performance of the original Canny step edge detector is poor near corners and junctions. The algorithm developed in [76] focuses on extending the edgel chains at corners and junctions so that better topological connections are achieved by relaxing the constraints of the step edge model and searching for paths with the greatest intensity variation. The edges are located to sub-pixel accuracy using weighted parabolic interpolation with respect to the edge direction. Examples of these contour fragments are shown in Figure 3.2.

3.4.2 From a similarity metric to curve correspondence:

The similarity metric S_{nm} which is computed between curve fragment C_n in the first image and curve fragment C_m in the second frame as described further below. The resulting similarity matrix S_{nm} is converted into an *assignment* in a greedy best-first fashion: The highest similarity ranked pair in the matrix is made into a correspondence, and the remaining items in the corresponding row and column are removed to retain a one-to-one mapping. Furthermore, a second similarity metric is used as an additional check to veto those curve pairs which are not sufficiently similar. Specifically, this second metric is based on the Hausdorff distance after the curves have been aligned by the optimal alignment-based similarity transformation between the curves. The process of selecting the most likely corresponding curve pairs eliminates the corresponding rows and columns. This continues until either no rows or no columns remain. This greedy approach can be potentially further improved by achieving a globally optimal assignment, *e.g.*, by using graduated assignment [38, 80], but this is not the focus of this paper.

3.4.3 Transition-sensitive elastic Matching:

We begin with an elastic curve-matching algorithm [78, 105] which minimizes an elastic energy functional over all possible alignments between two curves C and $\bar{C}$, by using an *alignment curve* α mediating between the two curves, Figure 3.3,

$$\alpha(\xi) = (h(\xi), \bar{h}(\xi)),\ \xi \in [0, \tilde{L}],\ \alpha(0) = (0,0),\ \alpha(\tilde{L}) = (L, \bar{L}), \tag{3.7}$$

where ξ is the arc-length along the alignment curve, h and $\bar{h}$ represent arc-lengths on C and $\bar{C}$, respectively, L and $\bar{L}$ represent lengths on C and $\bar{C}$, respectively, and $\tilde{L}$ is the length of the alignment curve α. The alignment curve can be specified by a single function, namely, $\psi(\xi)$, $\xi \in [0, \tilde{L}]$, where

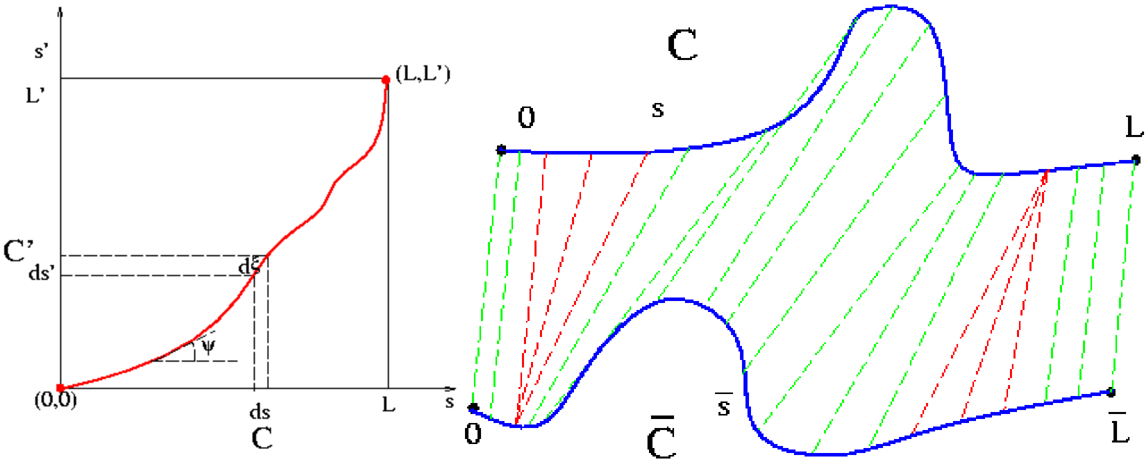


Figure 3.3: From [78] The *alignment curve* (left) represents a correspondence between two curves (right). The notion of an alignment curve allows for predicting correspondences mapping an entire interval to a point; This aspect of the correspondence is crucial here as it does occur in the context of transitions. The optimal alignment curve α is efficiently found by dynamic programming [78].

ψ denotes the angle between the tangent to the alignment curve and the x-axis. The arc-lengths of C and $\bar{C}$ can then be obtained by integration from ψ,

$$h(\xi) = \int_0^{\xi} \cos(\psi(\eta))d\eta,\ \bar{h}(\xi) = \int_0^{\xi} \sin(\psi(\eta))d\eta,\ \xi \in [0, \tilde{L}]. \tag{3.8}$$

The optimal alignment α between the curves can be found by minimizing an energy functional μ,

$$\mu[\psi] = \int [|\cos(\psi) - \sin(\psi)| + R_1|\kappa(h)\cos(\psi) - \bar{\kappa}(\bar{h})\sin(\psi)|]d\xi \tag{3.9}$$

where κ and $\bar{\kappa}$ are the curvatures of the curves. The first term describes differences in the arclength as defined in Eq 3.8 and thus penalizes "stretching". The second term is the difference in the angular extent associated with each infinitesimal pair of corresponding curve pieces and thus penalizes "bending" and R_1 relates the two terms. The "edit distance" between the curves C and $\bar{C}$ is defined as the cost of the optimal alignment given by $d(C, \bar{C}) = \min_{\psi} \mu(\psi)$ which is found by dynamic programming [78].

When this similarity metric is used to rank-order all curves in a frame with respect to a curve in another frame, the top ranking curve typically (72% for our database) yields the right correspondence when only gradual changes are involved. The overall curve correspondence performance is defined as

$$\text{curve correspondence}(\%) = \frac{\text{No of correctly assigned curve pairs}}{\text{Total no of corresponding curve pairs}} \times 100 \tag{3.10}$$

which is measured after the greedy assignment described earlier on a set of four manually labeled pairs of video frames. The overall curve-correspondence performance is 48% with errors arising mainly because transitions mislead the similarity metric, especially when a portion of one curve is matched to an entire curve of which it is a fragment, Figure 3.4(a,d), *e.g.*, as occurs in the fragmentation transition, Figure 3.2(b).

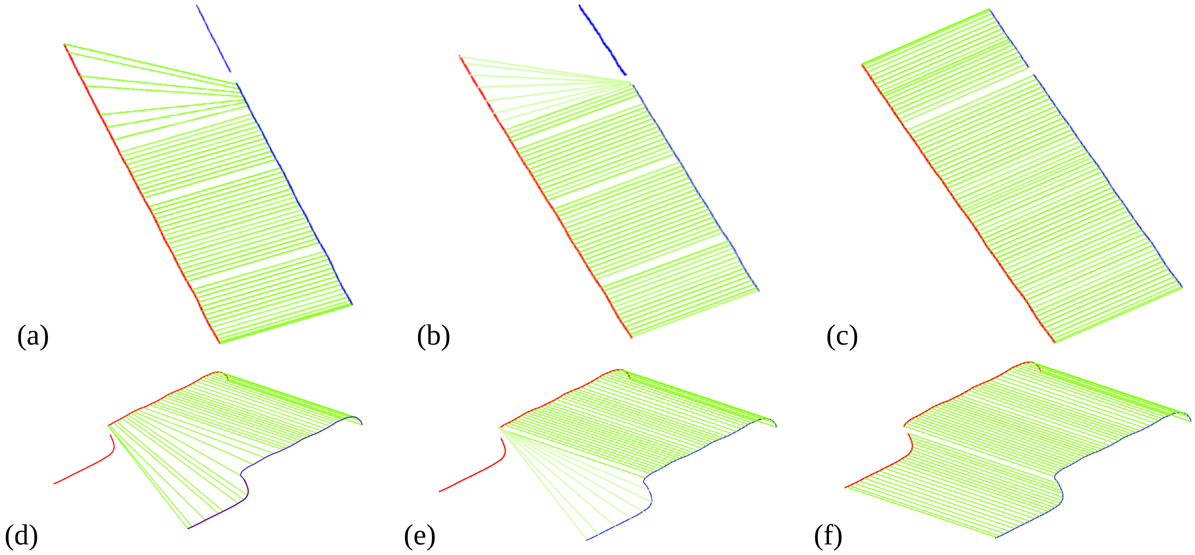
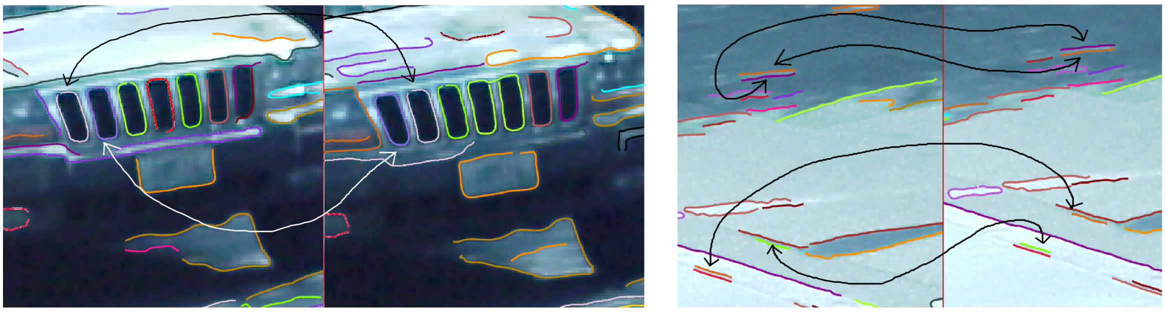


Figure 3.4: (a,d) The elastic curve matching alignments are incorrect in the presence of a transition but modifying the energy function to allow for such cases corrects the alignment (b,e) and furthermore recovers the fragmented “tail”. The alignment when excluding the tail is then used to define a geometric transform (similarity) between the two curves, which in turn is used to find and recover the broken curve fragment (c,f).

Observe, however, that in such cases there often remains sufficient shape similarity information in the remaining portion to correctly identify it as a sub-curve of the other curve. This requires that the energy cost be appropriately modified to allow for the possibility of such transitions. The removal or addition of a contour segment during the match is represented as a vertical or horizontal segment in the beginning or in the end of the alignment curve, since either h is constant and $\bar{h}$ is varying, or vice-versa. To avoid discouraging such alignments, the elastic energy on these segments is diminished by a factor ν ($\nu = 0.3$ for all our experiments). Figure 3.4(b,e) illustrates that the alignment is correctly identified from a sub-curve to an entire curve, and this is typically the case when the fragment has sufficient structure on it. The significance of the above modification is twofold. First, the elastic energy arising from the new corrected alignment results in a corrected similarity measure which more often points to the correct corresponding curve. Second, it allows for a more precise similarity transformation since in the corrected alignment the “tails” mapping an entire segment to a point are discarded from the Hausdorff distance computation, which is more sensitive to the presence of “tails”. This modification of the energy functional aimed at handling sub-curve matching increases the performance from 48% to 56%.

3.4.4 Explicit handling of transitions:

The above modification works well for sub-curves which have sufficient shape content but not so well for smaller sub-curves. Thus, in stage two we incorporate transitions, *e.g.*, as a single curve in one frame breaks into two sub-curves in the second frame, in the matching process. Specifically, assuming that the first stage has been successful in identifying the right correspondence between the original curve and one of the resulting sub-curves involved in the transition, the existence of a “tail” in the alignment is flagged (as shown in Figures 3.4(b,e)) as an indicator that a transition has likely occurred. Recall that the similarity transformation between the two curve fragments is obtained in the verification step without involving the initial “tail”. We now transform the “tail” accordingly in search of a mate in the other frame. If a third curve (the second sub-curve) exists that is sufficiently similar, the two sub-curves in frame two are merged and identified as a single curve. While this can be done iteratively for multiply fragmented curves, Figure 3.2(c, e, f, g), our current implementation only joins two curve fragments. With this improvement correspondence performance increases from 56% to about 68%.



(a) (b)

Figure 3.5: The stage two similarity metric fails to identify the corresponding pair when multiple similar structures exist (a) or when curves do not depict significant structure (b).

3.5 Use of Epipolar constraint to reduce ambiguity:

The above transition-sensitive shape-based similarity fails in two cases. First, when numerous similar structures are present, as in the seven rectangles in the front grill of the vehicle in Figure 3.5(a), the alignment between any pair of curve fragments is excellent and of low energy, so that the intrinsic nature of this shape metric does not significantly differentiate them to rank-order matches according to extrinsic placement. The second case involves contours which do not have significant "shape content", as in the straight lines on the pavement in Figure 3.5(b), so that there are numerous curve fragments with nearly equivalent alignments and energies. In such cases it is useful to introduce an extrinsic measure, namely, the *epipolar constraint*. We assume that within a limited neighborhood of frames, the motion of the object giving rise to the curve can be approximated as a translation, requiring the alignments between projected curves to pass through an epipole **e**, Figure 3.6. This epipole is either available as a vanishing point of the scene or it can be estimated together with the alignment between a pair of curves. In our experiments the camera was fixed, so the edges on the road are used to find the vanishing point manually.

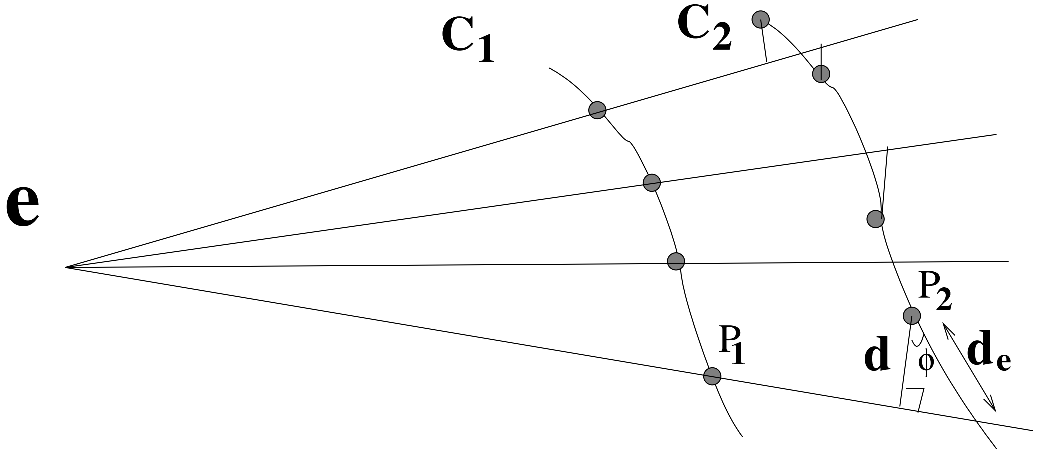


Figure 3.6: Epipolar lines through the sample points of the first curve should pass closely to the corresponding sample points on the second point, and vice versa. The distances between these corresponding points and the lines through he original sample points indicate deviation from the epipolar constraint and is used as an addition clue towards finding the correct curve correspondence.

The epipolar constraint is incorporated in the curve energy using an epipolar term. Consider a point of the alignment curve relating point P_1 on the first curve to the point P_2 on the second curve, Figure 3.6. Then distance of the point P_2 from the epipolar line passing through P_1 is computed along the tangent direction of P_2. The tangential distance d_e is estimated from the perpendicular distance d using $d_e = \frac{d}{cos\phi}$, where ϕ is angle between the tangent at point P_2 and the perpendicular line to the epipolar line as shown in Figure 3.6. Similarly a second estimate of d_e is computed with the role of the points based P_1 and P_2 reversed and the maximum is used as the values of d_e.

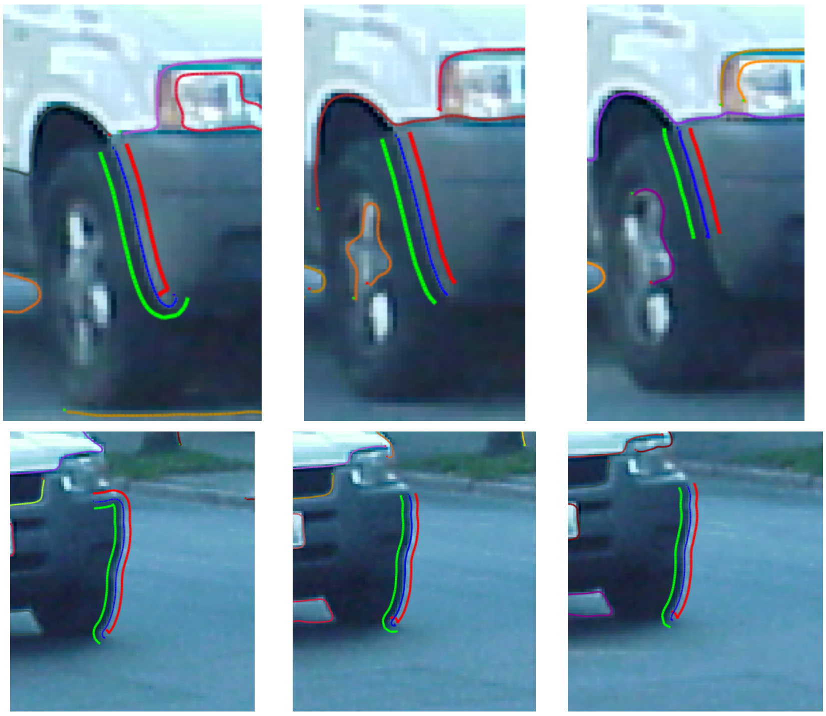
The modified energy then takes the form

$$\mu[\psi] = \int [|\cos(\psi) - \sin(\psi)| + R_1|\kappa(h)\cos(\psi) - \bar{\kappa}(\bar{h})\sin(\psi)| + R_2|(d_e)^p/(1+(d_e)^p)|]d\xi,$$

where $p = 10$ in our experiment. The performance after this stage increases from 68% to 85%.

3.6 Use of color and intensity to reduce ambiguity

Another powerful cue which can drastically reduce ambiguity and improve efficiency is color/intensity. Recall that imposing a maximum speed constraint reduces the number of potential matching curve pairs in two frames by a factor of 10-20 (for example, 35 curve matches may remain out of 700 curves in a second frame). Despite this drastic reduction in number of potential matches, the size of the remaining pool is large enough that the likelihood of a pair of non corresponding curves with similar shapes is not negligible. We now suggest that the use of continuity of color/intensity over time (frames) leads to improved efficiency and reduced ambiguity. The central assumption is that the color/intensity of a narrow region surrounding a curve changes only slightly from one frame to the next, on one side of the curve if it is an occluding contour, and on both sides of the curve otherwise. Figure3.7(a,b) illustrates this point.



(a)

(b)

Figure 3.7: The curve $C(s)$ is shown in blue while $C^{+}(s)$ and $C^{-}(s)$ are show in red and green, respectively. The remaining contours encode other information and should be ignored in this context. (a) non-occluding curve and (b) occluding curve

In the continuous domain, each point on the curve is attributed by color values in some color space, one for the left and one for the right side of the curve. We chose the HSV color space because of its ability to separate intensity from color in an intuitive fashion. Thus, each point on the curve $C(s)$ is attributed with $(H^{+}(s), S^{+}(s), V^{+}(s))$ and $(H^{-}(s), S^{-}(s), V^{-}(s))$ where H is

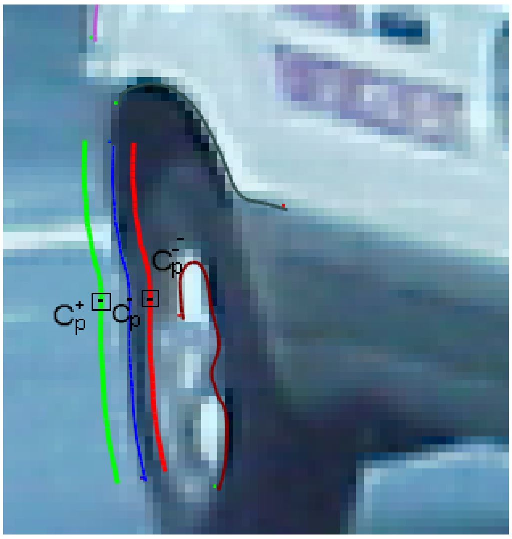


Figure 3.8: The attributes at point p are computed at C_p^+ and C_p^- neighborhoods as indicated.

hue(color), S is saturation and V is value(intensity) as defined by the conversion in [34], and where $+/-$ denote the intensity to the immediate left and the immediate right of the curve respectively. In practice, since a large number of edges are not step edges and the transition across an edge may be gradual, we opt to define

$$\begin{cases} H^{\pm}(s) & = H(C^{\pm}(s)) \\ S^{\pm}(s) & = S(C^{\pm}(s)) \\ V^{\pm}(s) & = V(C^{\pm}(s)), \end{cases} \tag{3.11}$$

where $C^{\pm}(s) = C(s) \pm \delta N(s)$, $N(s)$ is the normal to the curve, and δ is a fixed constant. See Figure 3.8. Observe care should be taken that in case of close-by curves that δ does not exceed the space between curves. Thus, the signed distance transform is used to detect when δ exceeds this limit in which no value is assigned to such points. In addition, to reduce the effect of noise we use the HSV values of a smoothed image by applying a Gaussian kernel to each of the components of HSV space individually.

These new attributes of a curve can now be used to discard unlikely matches, *e.g.*, a curve separating say red and green region in one frame cannot be a match for a curve separating blue and brown regions (we can also use these attributes in the alignment process itself which is work under progress). We propose to summarize the HSV attributes in a coarse fashion using the 3D histogram. $S(r, z, \theta)$ where the height z represents "value", the radius r indicates "saturation" and the angle θ represents "hue" as shown in Figure 3.10(a). The bins are defined with spacing along radius and value to give equal volume bins.

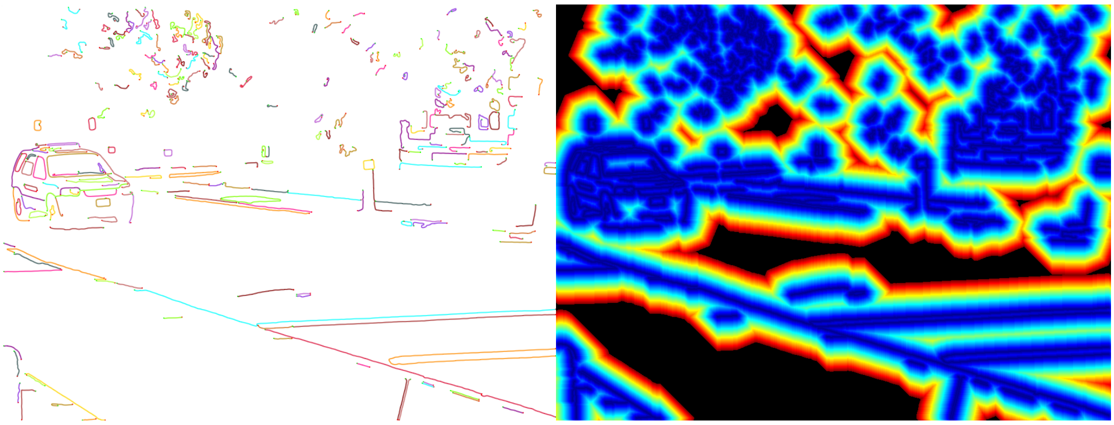

(a) (b)

Figure 3.9: (a) The set of curves in an image and, (b) the distance transform indicating the largest possible distance from each curve.

The three-dimensional histogram $g(\theta, r, z)$ can be used to coarsely compare two curves. We use the Bhattacharya distance to compute the dissimilarity measure between two histograms,

$$d_g(g_1, g_2) = -\ln(\sum_{r,\theta,z} g_1(r, \theta, z) g_2(r, \theta, z)) \tag{3.12}$$

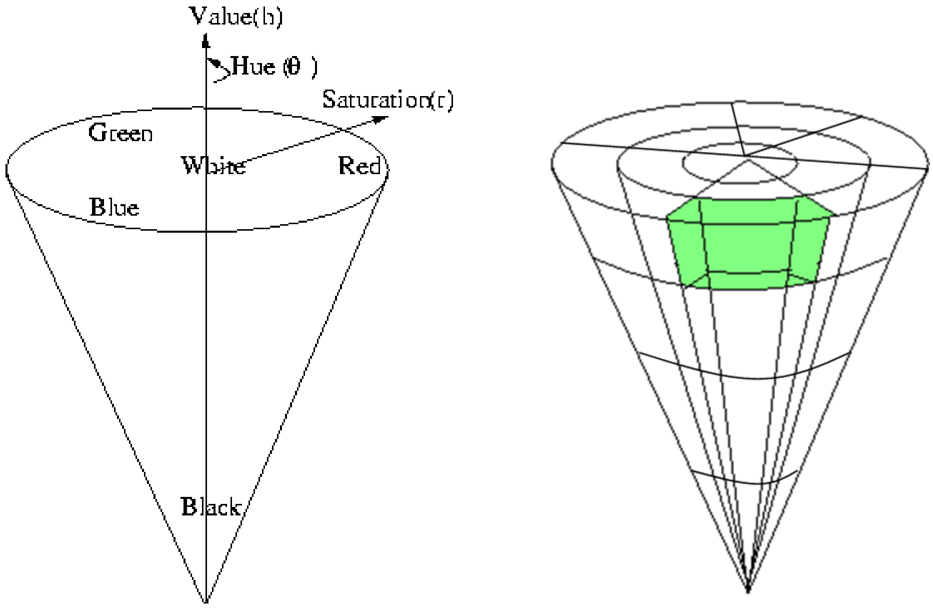
Since an occluding contour may have only one matching side and since the orientation of each curve is arbitrary, we compare four possibilities and assign the minimum dissimilarity as the distance between the two curves

$$d_B(C_1, C_2) = min(d_g(g_1^+, g_2^+), d_g(g_1^-, g_2^+), d_g(g_1^+, g_2^-), d_g(g_1^-, g_2^-)) \tag{3.13}$$

where g_1^+, g_2^+ are the histograms for right side of the curves C_1 and C_2, respectively, and g_1^-, g_2^- are the histograms for left side of the curves C_1 and C_2, respectively. The Bhattacharya distance between coarse HSV histogram of regions flanking a pair of matching candidate curves can then be used to discard unlikely matches by thresholding the distance, *i.e.*, two curves for which

$$d_B(C_1, C_2) < \tau_B \tag{3.14}$$

for some threshold τ_B will be considered further for measuring fine scale shape similarity. We select a very conservative threshold which ensures zero error in our ground truth database ($\tau_B = 0.45$). The improvement in efficiency even with such a very conservative estimate is drastic: about 43% of the matching candidates are discarded. The reduction in ambiguity when using only the maximum speed constraint is roughly 10% improvement in the correct correspondence rate. These



(a) (b)

Figure 3.10: (a) The HSV color space representation, (b) Binning for building a 3D histogram

measurements were not repeated for the epipolar constraint as the latter constraint already removes much ambiguity. However, in cases where this constraint is not expected to hold, the more generic inter-frame continuity plays a significant role.

Results: Figure 3.11 shows several examples of the final curve correspondence for adjacent frames taken from several video sequences. In order to formally evaluate the curve correspondence algorithm, a database of ground truth consisting of four image pairs was manually created. Along with this, a curve is also labeled as foreground or background for verification of results in Section 3.7.As tabulated in Table 3.6, 80%-90% of correspondences are correct in these four frames which, as we shall see in the next section, is sufficiently high to enable reliable figure-ground segregation. We also expect significant improvements when several other constraints are utilized in the similarity measure, including a measure of intensity and color match for each alignment, use of 3D geometric reconstruction, imposing spatial order among the curve fragments to disambiguate correspondences, and in particular when compound transitions are also explicitly handled.

3.7 Transformation-Induced Figure-Ground Segregation

In this section we describe a figure-ground segregation method based on the Gestalt cue of *common fate*. Specifically, since the curve correspondence has established how each curve transforms form

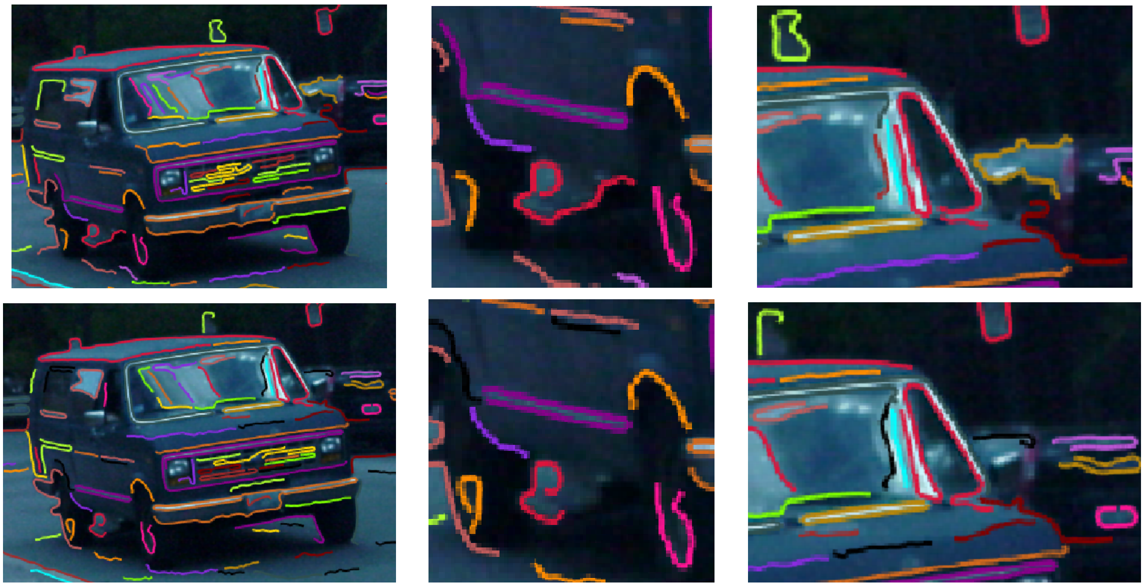


Figure 3.11: Matched curves in a pair of video frames (top and bottom, on the left) and corresponding zoomed areas on the right. Corresponding curve fragments are shown in the same color.

one frame to another, curves with distinctly similar transforms should be grouped. These transforms are characterized in the domain of an expected geometric transform, in our case the similarity transform, although affine or projective transformations can also be used.

While it is tempting to measure the similarity between two transforms by measuring the distance between the parameter vectors describing each transform, it is much more meaningful to measure similarity not in the parameter space, but in the observation space. Specifically, consider a transform $T_1(T_{1x}, T_{1y}, \theta_1, \lambda_1)$ where (T_{1x}, T_{1y}) are translation coordinates and θ_1 is the angle of rotation, and λ_1 is scaling is defined by an inter-frame curve pair $(C_1, \bar{C}_1)$ and similarly $T_2(T_{2x}, T_{2y}, \theta_2, \lambda_2)$ is defined for the $(C_2, \bar{C}_2)$ pair. Rather than rely on differences between the parameter describing T_1

Image-pair	% correct
SUV67-68	73
Police-car 16-17	80
Police-car 21-22	85
Minivan65-66	86

Table 3.1: Overall curve performance for four pair of video frames.

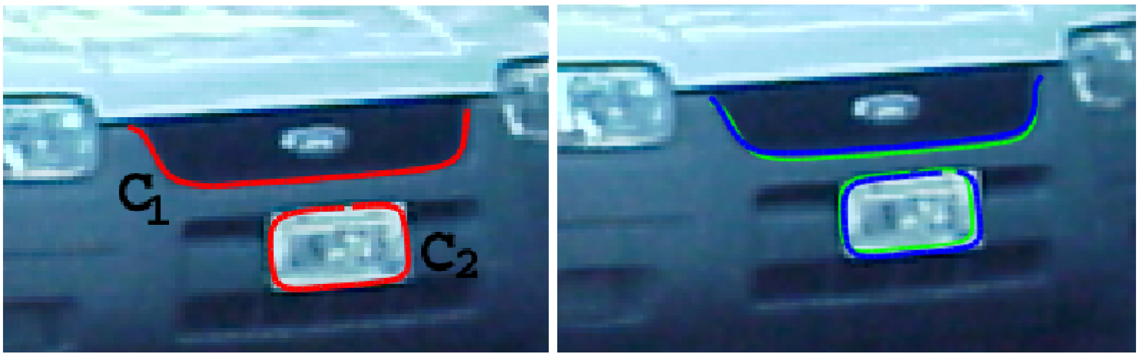


Figure 3.12: Green curves in frame 2 are modeled similarity transformations of the red curves in frame 1 while blue curves are the actual curves in frame 2.

and T_2, we define the similarity of T_1 and T_2 by the extent T_1C_2 is similar as a curve to $\bar{C}_2$, and analogously, T_2C_1 is similar to $\bar{C}_1$, Figure 3.12,

$$d_T(C_1, C_2) = max\{d_H(T_1C_2, \bar{C}_2), d_H(T_2C_1, \bar{C}_1)\}. \tag{3.15}$$

where d_H is the Hausdorff metric between two curves.

This pairwise measure defines the degree by which two curves in one frame have "common fate" with respect to the second frame and is represented by an m x m matrix where m is the number of curves in the first frame. Ideally, a moving object on a stationary background would lead to two distinct clusters in this matrix. However, since background curves can also shift in a wide range of movements resembling some of those on the object, *e.g.*, tree branches moving in the wind, this distinction is smeared.

We adopt a simple clustering technique to determine cluster boundaries, namely, the seeded region growing method used for segmentation of intensity images [3]. Each curve is initialized as a cluster. The distance between two clusters is defined as the median of pairwise distances between their members. An iterative procedure then merges the two closest clusters into one until either the closest distance between clusters exceeds some threshold or the number of clusters falls below a minimum number of expected clusters. An additional spatial constraint is used to rule out clustering of curves which are far in Euclidean space. The clusters only for which inter-cluster Euclidean distance is less than threshold τ_s are considered for clustering.

Figure 3.13 depicts the clusters associated with the foreground for two distinct frame pairs. Figures 3.14 show results for four subsequent frames, in three different videos. Note that the figure ground segregation only based on adjacent pair of frames only. As tabulated in Table 3.7 the segregation includes few non-object contours (5%-10%) for our four ground-truth frame pairs, while capturing a significant collection of the curves on the object.

Image-pair	object curves	false segregation	correctly segregated curves
SUV67-68	84	5	51
Police-car 16-17	38	3	18
Police-car 21-22	31	5	16
Minivan65-66	65	3	29

Table 3.2: Performance of segregating curves in four pair of video frames.

The computational complexity of the approach depends on a number of factors including the number of curves and the number of sample points on each curve. There are generally 600-700 curve fragments per frame. The number of sample points on each curve varies from 40 to 200. The complexity of matching a pair of curve segments is $O(n^2)$ where n is the number of sample points on curve segments but multi-scale approaches can be used to speed this up. The complexity for matching curves in two frame is $O(M^2n^2)$ where M is the number of curves in a frame. The overall analysis takes approximately 2 to 2.5 minutes to process a frame on a Pentium 4, 2 GHz machine.

3.7.1 Comparisons:

We have compared the segmentation of the figure results using our approach with 3 different approaches. First we compare it with the KLT tracker [81], in which robust feature points are tracked. Our segregation is richer than the above technique as evident in Fig 3.15(b,c). Next, we compare it to an optical-flow based approach. The optical flow is computed for the image and the pixels with velocity above a certain threshold are considered as figure. Note that the optical flow segmentation is not robust to noise in the background and also it suffers from the well known aperture problem in uniform regions. As a result the segmentation has holes as shown in Fig 3.15(d). Last, we compare our results with active-contour based tracking methods. A contour in the first frame has been manually initialized which is then snapped onto the object using the geodesic active contour approach [16, 54]. A constant velocity model is then used to propagate the contour to the following frames and used the active contour approach is again applied.

We also tested our approach for performance under occlusion by blocking a portion of the video sequence. As illustrated in Fig 3.16, curves in the non-occluded part of the object are not affected by the occlusion and the resulting figures is a rich description of object.

We emphasize that these results while already very encouraging are only using pairwise comparison of frames and can be potentially significantly improved further. Observe in Figure 3.14 how each frame pair gives a segmentation that has many common curve fragments with its nearby frame pairs, but also feature novel curves not seen before. We have not yet utilized this *multi-frame regularity* which should lead to a dense and complete segmentation after a few frames. Also, the emphasis has not been on using a sophisticated clustering method, although the use of one would certainly improve the results. We expect that the addition of regional motion information will also significantly improve the results. As the comparison in Figure 3.14 shows our curve-based approach is a promising direction for figure-ground segregation and tracking in a wide range of applications.

3.8 System parameters:

A list of parameters for the approach and their effects are given in Table 3.3.

Parameter	**Description**	**Effect**
τ	edge detector threshold for contrast.	increasing the value will increase the selectivity of the detector and reduce the number of edges detected.
e	initial estimate of the epipole obtained by extrapolating sides of the roads in the scene	rough estimate of epipole is needed
R_1 & R_2	constants used in the elastic matching cost function which were estimated empirically	R_1 weighs the bending term and R_2 weighs the epipolar term. $R_1 = 10$ and $R_2 = 3$ is used for all the experiments.
ν	energy factor reduction at the end of the curve to enable the sub-curve matching and handle transitions	lower values favor sub-curve matching.
N	initial number of clusters for clustering curves with similar transformation	larger number than expected number of objects would increase the fragmentation (overfitting) of the figures and fewer number (underfitting) would result in opposite.
d_{min}	minimum inter-cluster distance in agglomerative clustering	large values would lead to conservative clustering
τ_s	threshold for clustering curves which are spatially closer to each other based on their Euclidean distance	the value should be adjusted to the expected size of the object. As increasing the value would allow more false alarms in the segregation.
δ	used in computing $\{$H,S,V$\}$ values at each point on the curve at δ distance from the curve	should be around 2-3 pixel. If its really large we would cross over into regions of other curves and if its really small the intensities would be from the edge region.
τ_B	threshold for comparing curves based on their color values	lower values would be more conservative and higher values would lead to increased number of misses.

Table 3.3: System parameters are listed along with their sensitivities and effects.

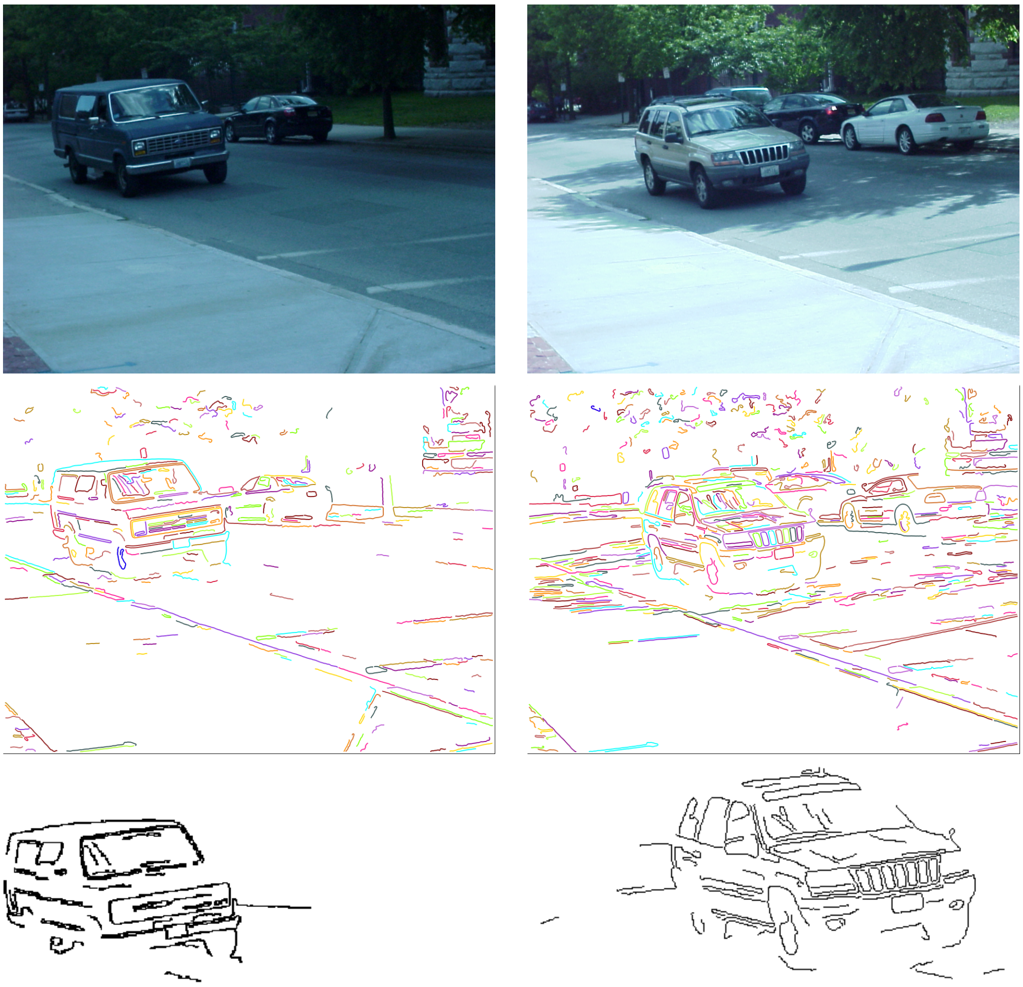


Figure 3.13: Results of Figure-ground segregation based on two adjacent frames for a Van (first frame shown on the left) and an SUV(first frame shown on the right). The top row shows the original image, the second row shows the contours extracted and last row shows the segmented object.

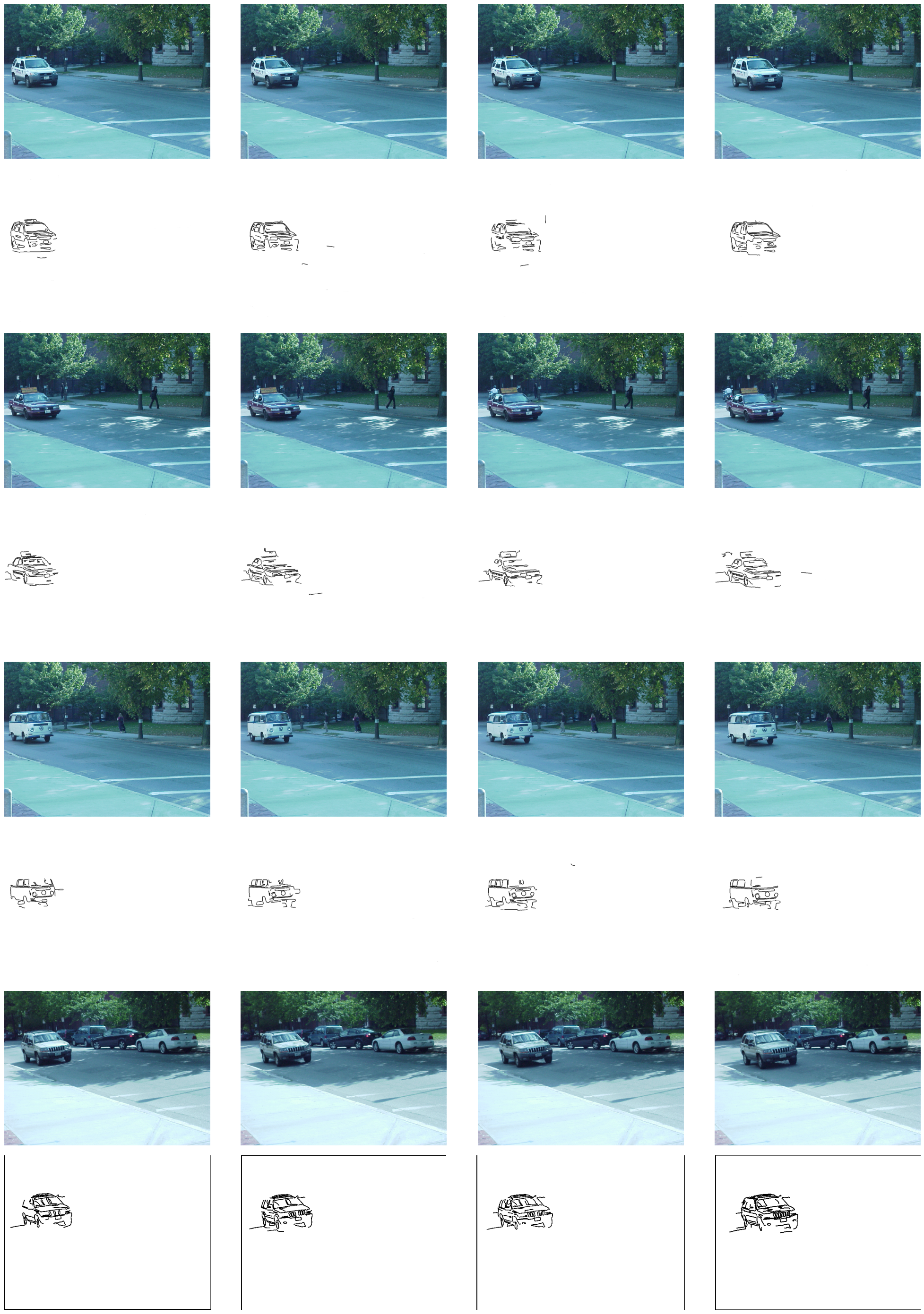


Figure 3.14: Two-frame segregation of moving vehicles in two subsequent video sequences. Observe how our segregation produces a rich description of the figure which can than be used for recognition.

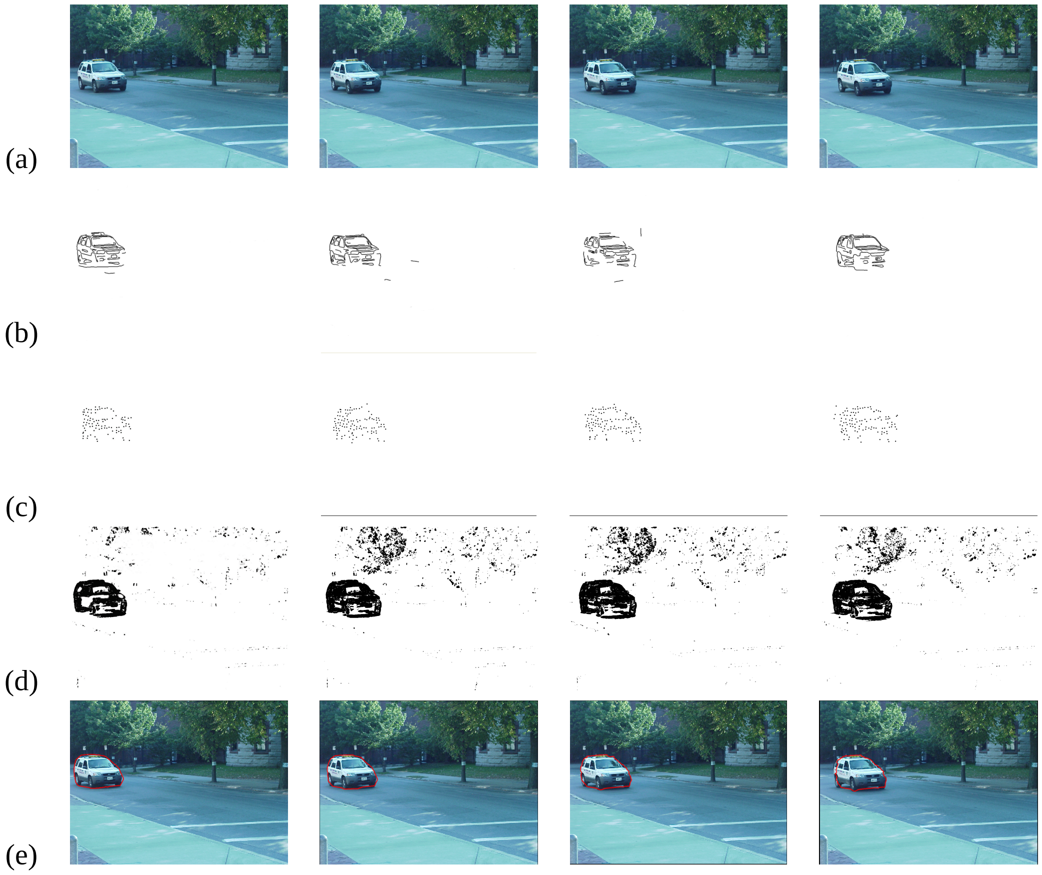


Figure 3.15: (a) original video sequence, (b) segmentation of curves using our approach, (c) segmentation using KLT tracker [81], (d) region-based segmentation using optical flow and (e) edge based tracking using geodesic active contours, which requires a periodic manual initialization.

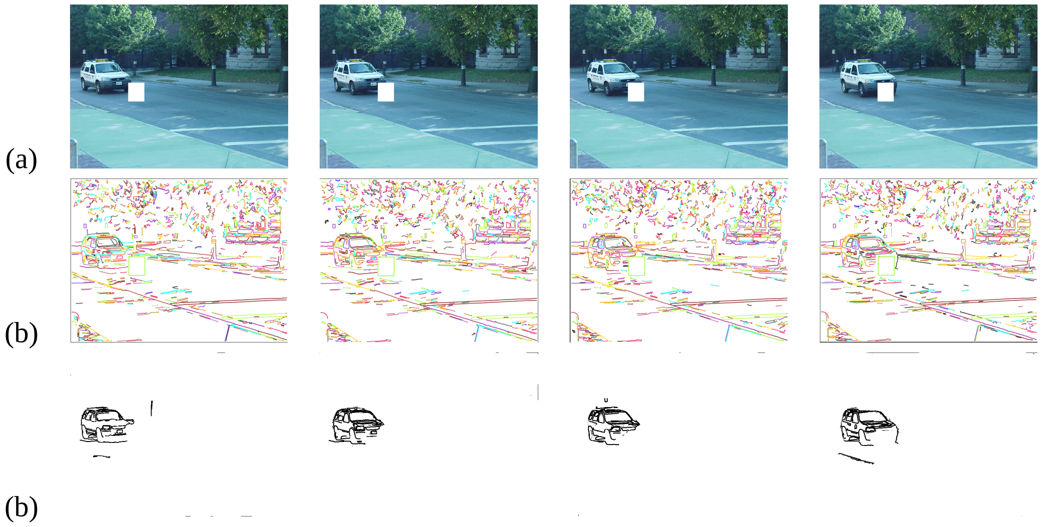


Figure 3.16: (a) original video sequence with an occlusion, (b) edge maps of the above video sequence, and (c) segregation of curves using our approach.

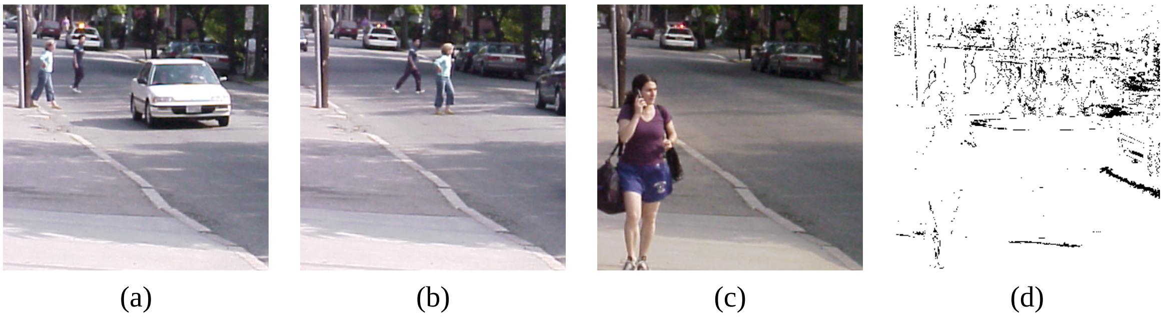
Chapter 4

Edge-based Segmentation of Moving Objects in Videos from Stationary Camera or Registered Images

The curves obtained from aerial videos where size of the objects is small are unstable leading to poor matching of contours across the frames to allow segmentation of objects. Such class of videos are acquired in surveillance applications, *e.g.*, aerial videos where objects of interest like vehicles are way smaller than the background objects. Since the size of the objects is small, the contour detectors do not provide consistent linking throughout the frames and thus the approached described in Section3 would fail to segment objects. But the edge-maps across the frames are stable. The aerial videos are different as they have large background and small foreground objects. This allows the videos to be registered with respect to the background in case of planar scenes [70] as well as in case of non-planar scene [24]. Once the video is registered, background is stationary and then anything moving is detected. This idea of modeling stationary background and detecting everything deviant from the model as a moving object has been widely used for various applications. But most of these approaches model intensity of the background.

4.1 Introduction

Methods for the analysis of moving objects in video and other time sequences obtained from stationary cameras, *e.g.*, for surveillance and monitoring, typically model the stationary background and detect moving objects as those pixels which do not fit this model. Averaging frames over time is a simple method of constructing a background model which is effective if objects move continuously over the scene and lighting does not change rapidly. Background modeling using multiple distributions is effectively used to handle images when the following conditions are met: *(i)* with slowly moving objects, *(ii)* slight lighting variations, and *(iii)* repetitive object movements such as leaves moving because of wind [86, 51, 74, 57, 63]. The most popular schemes use the Mixture of Gaussian (MoG) model for each pixel. The intensity at each pixel is modeled using a fixed number of Gaussians which are updated on every observation. Any pixel which is unlikely to come from the MoG is classified as foreground.



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

Figure 4.1: The effect of sudden illumination change on intensity-based background modeling . (a,b) a pair of typical background images,(c) a frame when the illumination has changed and (d) the corresponding foreground detection. Observe that the foreground, the pedestrian, has not been identified and a large part of the background is labeled as the foreground.

Methods for **modeling background intensity** typically suffer from three limitations. First, they are susceptible to sudden changes in illumination, either global changes, *e.g.*, due to the sun coming out of the clouds, or local changes, *e.g.*, due to partial reflection from a brightly colored objects passing nearby, *etc.* Handling different illumination sources requires either a broader distribution model or adding a new distribution to the mixture, both of which reduce the sensitivity to figure segmentation. Second, these models are susceptible to changes in the camera model. For example, automatic gain control can change the overall intensity distribution as a bright object enters the field of view as illustrated, *e.g.* in Figure 4.5. The third drawback of the intensity-based methods is that numerous observation frames are required for learning the background model, especially (i) when the illumination is changing and (ii) the scene is constantly occupied with moving objects or when objects are moving slowly.

An alternative to modeling background intensities is to model the **background intensity gradient**. Jabri *et al.* [46] augments the traditional intensity background model with models of the intensity gradient magnitude as captured by the Sobel operator responses. Large changes in either intensity or in edges indicate the presence of foreground. While the use of intensity gradients While the use of intensity gradient is diagnostic to detect foregrounds, retaining the use of intensities also retains the sensitivity to sudden changes in illumination. In contrast, Javed *et al.* [49] require significant changes in *both* the intensity and intensity gradient to declare pixel as foreground. However, intensity gradients arising from large illumination changes can still signal a figure when none exists, Figure 4.6(b).

A key limitation of intensity and intensity gradient background models is that background models do not take *spatial interactions* into account: Each pixel is classified independent of its neighboring pixel, without regard to the local geometry of the image. A step in this direction of using the local geometry is to use edges in the modeling of the background. Edge maps tag those background pixels which maximize local gradient in a neighborhood of pixels. The edge-maps are sparse as compared to intensity and gradient map but still contains the same amount of information about the shape of the object as it is shown that image can be constructed from the information available on edges in [28]. Yang and Levine [103] modeled background edge-maps as a spatial binary map using robust statistics on the strength of the edges where the edges diagnosed as outliers correspond to the foreground edges. This approach is not adaptive to changes in the scene as it requires a background map as input and is susceptible to changes in strength of the edge when illumination changes. Kim and Hwang [55] detect the edges of current frame as well as of the difference image of consecutive frames. They compare the edge-locations of both maps with a background edge-map and detect foreground edges for each of the maps individually. The edges common to the two foreground edge-maps are output as foreground edges.

Modeling pixel-level binary edge-maps has two main drawbacks. First, the discretization errors in pixel-based binary edge maps lead to unnecessarily broad background models: a background edge halfway between the pixels will require both pixels modeled as background, thus unnecessarily “blurring” the background model, which in turn reduces sensitivity to detecting figures. Second, edges in previous works are considered as points with spatial coordinates, while they also capture the *orientation* of a local image patch. Discarding orientation information in the background model has the undesirable effect that when a foreground edge happens to fall on a pixel with a declared background edge, it would be misclassified as background. The use of orientation in background modeling would drastically reduce such misclassifications to the case when both position and orientation of a foreground edge match that of a background edge, a much less significant event.

This work proposes a background model based on *sub-pixel* edge-maps where sub-pixel position (x, y) and subsample orientation θ of edges are modeled. This work also shows that sub-pixel edge-map background models attain high precision and accuracy in addition to being invariant to illumination changes and accommodates small translations easily. Another advantage is that the algorithm requires fewer frames to build the background model even in case of slow moving objects and busy scene. The advantage of modeling sub-pixel edges becomes evident in scenes with cluttered backgrounds where edges from a figure can share the same pixel as well as the same orientation as shown in Figure 4.3.

This paper is organized as follows. First, the effect of illumination changes on edges is shown to smaller when compared to intensity and gradient in Section 4.2 . Second, the conventional approach to model the background by MoG is discussed in Section 4.3. The observation variables for modeling sub-pixel edges and modifications to the above approach are discussed in Section 4.4 followed by qualitative and quantitative experimental results in Section 4.5.

4.2 Effect of illumination change

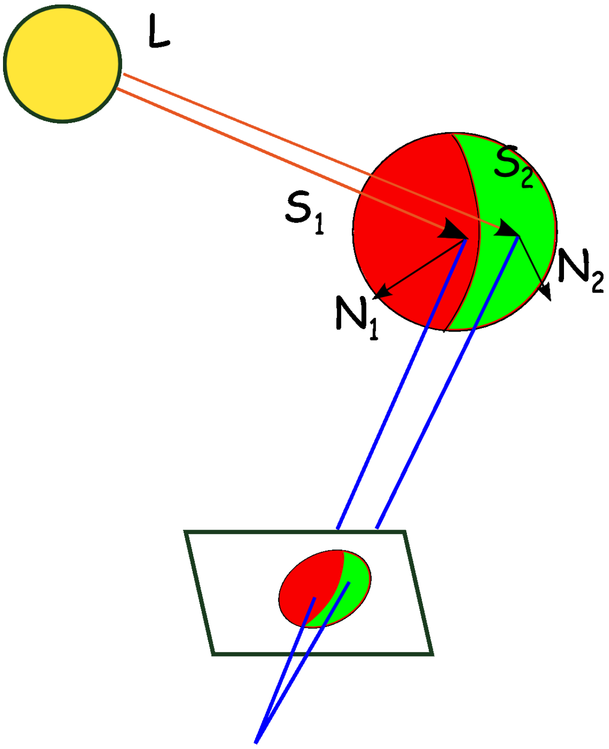


Figure 4.2: This figure shows an image formed by two surfaces S_1 and S_2 and light source L

We now compare the effect of change in illumination on *(i)* intensity, *(ii)* intensity gradient and (iii) edges. Consider a local area in a scene with two Lambertian surface patches S_1 and S_2 with normals N_1 and N_2, and albedos a_1 and a_2, respectively, are mapped to adjacent patch in the images separated by an edge, as shown in Figure 4.2. The surface patches need not be adjacent, as in the case of occluding contours. They also need to have distinct normals as in the case of reflectance edge on a smooth surface patch. The image intensities in each patch, I_1 and I_2 are given by

$$\left\{ \begin{array}{lcl} I_1 & = & a_1 N_1 . L + L_\epsilon \\ I_2 & = & a_2 N_2 . L + L_\epsilon \end{array} \right. \tag{4.1}$$

where L is a point illumination source and L_ϵ is the ambient light.

1. **Intensity**: The differential change in intensity, ΔI due to a change in illumination of the light source ΔL is given by

$$\left\{ \begin{array}{lcl} \Delta I_1 & = & a_1 N_1 . \Delta L \\ \Delta I_2 & = & a_2 N_2 . \Delta L . \end{array} \right. \tag{4.2}$$

2. **Gradient**: The intensity gradient at its peak is computed pixel by using the two adjacent pixels with intensities I_1 and I_2,

$$|\nabla I| = \frac{1}{\Delta x} |I_1 - I_2| = \frac{1}{2} |a_1 N_1 . L - a_2 N_2 . L| \tag{4.3}$$

where $\Delta x = 2$ is the distance between the two adjacent pixels. The differential change in the gradient with respect to an illumination change Δl is given by

$$\begin{array}{rcl} \Delta |\nabla I| & = & \frac{1}{2} |a_1 N_1 . \Delta L - a_2 N_2 . \Delta L| \\ & \leq & \frac{1}{2} |a_1 N_1 . \Delta L| + \frac{1}{2} |a_2 N_2 . \Delta L| \\ & \leq & \frac{1}{2} (\Delta I_1 + \Delta I_2) \end{array} \tag{4.4}$$

Clearly, $\Delta |\nabla I| \leq \Delta I$. In other words, the correlation effect of illumination change on pixels intensities is mostly canceled in the gradient process, leaving behind a much reduced effect.

3. **Edges**: Edges are obtained by localizing at the extrema of gradient and thresholding the gradient, *i.e.*, if $|\nabla I| > \tau$, where τ is the edge contrast threshold. The illumination change has two potential effects. First, it changes the location of the edge slightly. The position of edges are not affected by the change in the magnitude of the illumination, e.g., clouds hiding the sun. However, the position of edges changes due to change in illumination direction, e.g., sun changing position from morning to evening. Second, the change in illumination may change the classification of an edge to a non-edge and vice-versa. Fortunately, the majority of edges are typically sufficient above the threshold such that a slight illumination change would not affect them. Those pixels whose strength is close to threshold could change the status if the illumination change acts in an opposite manner to classification, *i.e.*, if the pixel gradient is above threshold, it remains an edge if the illumination change is positive and may change

status if the illumination change is negative and it is sufficiently strong to reduce the gradient below threshold, *i.e.*, if

$$\nabla I - \Delta|\nabla I| < \tau \tag{4.5}$$

or if ΔL satisfies

$$\nabla I - \frac{1}{2}|a_1 N_1.\Delta L - a_2 N_2.\Delta L| < \tau. \tag{4.6}$$

Similarly, for a non-edge pixel the change in illumination must be positive and must be sufficiently strong to change the status of the pixel, *i.e.*,

$$\Delta I + \frac{1}{2}|a_1 N_1.\Delta L - a_2 N_2.\Delta L| > \tau.$$

In summary, an illumination change ΔL affects two sets of pixels, edge points who switch labels,

$$\begin{array}{ccll} (i) & \nabla|\Delta I|.\nabla I^{\top} & = & 0 \\ (ii) & |\nabla I| & > & \tau \\ (iii) & \nabla I - \frac{1}{2}|a_1 N_1.\Delta L - a_2 N_2.\Delta L| & < & \tau. \end{array}$$

and non edge points which becomes edges requiring

$$\begin{array}{ccll} (i) & \nabla|\Delta I|.\nabla I^{\top} & = & 0 \\ (ii) & |\nabla I| & < & \tau \\ (iii) & \nabla I - \frac{1}{2}|a_1 N_1.\Delta L - a_2 N_2.\Delta L| & > & \tau. \end{array}$$

Clearly, edge maps are by far less sensitive to illumination changes.

From the above study, it is clear that edges are the least susceptible representation to change in illumination while intensity is the most susceptible. The edges can handle the sudden change in magnitude of illumination. However in order to handle the gradual changes in illumination direction which changes the edge position, the background modeling technique in [86, 51], is used as it can adapt to gradual changes. This technique is further discussed in the next section.

4.3 Mixture of Gaussians Background Model

The Mixture of Gaussians (MoG) background model uses a Mixture of Gaussians to model the distribution of χ, a nD random variable representing observations over many frames. Each of the components of MoG is a Gaussian distribution, η_i, with a mean μ_i and covariance Σ_i given by

$$\eta_i(\chi, \mu_i, \Sigma_i) \quad = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\chi-\mu_i)^T \Sigma_i^{-1}(\chi-\mu_i)}. \tag{4.7}$$

The probability of observing $\chi = X_0$ is given by

$$P(\chi = X_0) = \sum_{i=1}^{N} \omega_i \eta_i \left(X_0, \mu_i, \Sigma_i\right) \tag{4.8}$$

where w_i are the weights of each of the individual component of MoG and their sum is 1, *i.e.*, $\sum_{i=1}^{N} w_i = 1$.

Updating the MoG: Consider a MoG model of a scene at time time k described by $\{(w_i^k, \mu_i^k, \Sigma_i^k), i = 1, 2, ..., N\}$, where N is the number of Gaussian distributions. A new observation, χ_{k+1}, affects the existing MoG model in that each of the Gaussian components of the mixture as well as their weight needs to be updated to include the new observation in the model. There exists many variants of the updating rule [51, 86]; in this work the update rule described in [51] is used, namely, the new MoG model $\{(w_i^{k+1}, \mu_i^{k+1}, \Sigma_i^{k+1}), i = 1, 2, ..., N\}$ is computed using

$$\begin{cases} w_i^{k+1} & = & w_i^k + \delta w_i^{k+1} \\ \mu_i^{k+1} & = & \mu_i^k + \frac{\delta w_i^{k+1}}{\delta w_i^{k+1} + w_i^k} (\chi_{k+1} - \mu_i^k) \\ \Sigma_i^{k+1} & = & \Sigma_i^k + \frac{\delta w_i^{k+1}}{\delta w_i^{k+1} + w_i^k} \\ & & \\ & & [(\chi_{k+1} - \mu_i^k)^\top (\chi_{k+1} - \mu_i^k) - \Sigma_i^k] \end{cases} \tag{4.9}$$

where δw_i^{k+1} is computed as

$$\delta w_i^{k+1} = \frac{\eta_i(\chi^{k+1})}{L}, \tag{4.10}$$

where L is the number of "training" frames typically set to $L_{max} = 20$. When the number of observed frames are less than L then $L = k$, *i.e.*, $L = min(k, L_{max})$. The use of a maximum number of frame L_{max} allows for the model to adapt to gradual changes, *i.e.* by taking account only a moving window of say the last 20 frames. These updated weights w_i are normalized to ensure $\sum_{i=1}^{N} w_i = 1$. The N components of the mixture are sorted by the ratio $w_i/|\Sigma_i|$. This ratio is expected to have a high value for the frequently observed distributions which have a lower covariance. The components corresponding to higher ratios are considered to be background as they have been seen very often. Thus the first N_b components having sum of weights greater than a threshold τ_w (set to $\tau_w = 0.5$) are defined as the background components.

Foreground Detection: Given an observation χ^{k+1}, at a pixel p, p will be classified as foreground if χ^{k+1} **does not** lie within d standard deviations of any one of the N_b components, *i.e*,

$$(\chi^{k+1} - \mu_i)\Sigma_i^{-1}(\chi^{k+1} - \mu_i)^T > d^2 \qquad \forall i \in \; N_b \tag{4.11}$$

where typically d is 2.5. On the one hand, if the observation belongs to one of the components then it updates that particular component. On the other hand if it is a foreground then a new Gaussian distribution is initialized with mean χ^{k+1} and a default covariance and weight is initialized to 0.1. And this new distribution replaces the distribution with the lowest ratio $w_i/|\Sigma_i|$ if the number of distributions of MoG is N and is just added if the number of distributions is less than N.

Behavior of components of MoG: Each of the components of MoG represents one class of expected views together with a group of variations in the observations arising from noise or small variations and quantified by the variance. Observe that by design each component cannot handle large variations since as the variance increases, the rank of the distribution component goes down according to $w_i/|\Sigma_i|$ which is then eventually discarded.

The MoG model is initialized at each point of the image grid with zero components and the first image updates this model. Note that the model is continuously updated but generally it takes several frames to build the model and to get reliable foreground detection. The observations for intensity, gradient and edges are discussed below.

4.4 Different variations of Observations

The above machinery for the MoG background modeling is the same whether χ represents intensity, intensity gradient, or edges. We consider four possibilities below.

Intensity: In order to model gray-scale intensities, χ at each pixel is a 1-d variable given by

$$\chi = I(x, y) \;\in [0, 255]. \tag{4.12}$$

Gradient Map: Gradients map are obtained by convolving gray-scale image with a 1D derivative of Gaussian filter. The image is convolved along the rows to obtain I_x and along the columns to obtain I_y. Then the gradient magnitude is obtained by $\|\nabla I\| = \sqrt{I_x^2 + I_y^2}$. Therefore, χ is a 1-d variable at each pixel (x, y) is given by

$$\chi = \|\nabla I\|(x, y) \;\in [0, 255] \tag{4.13}$$

Pixel edges with orientation: Pixel edge maps are computed by non-max suppression on the thresholded gradient map. The pixel edge map is a binary map and has an orientation associated with it, if

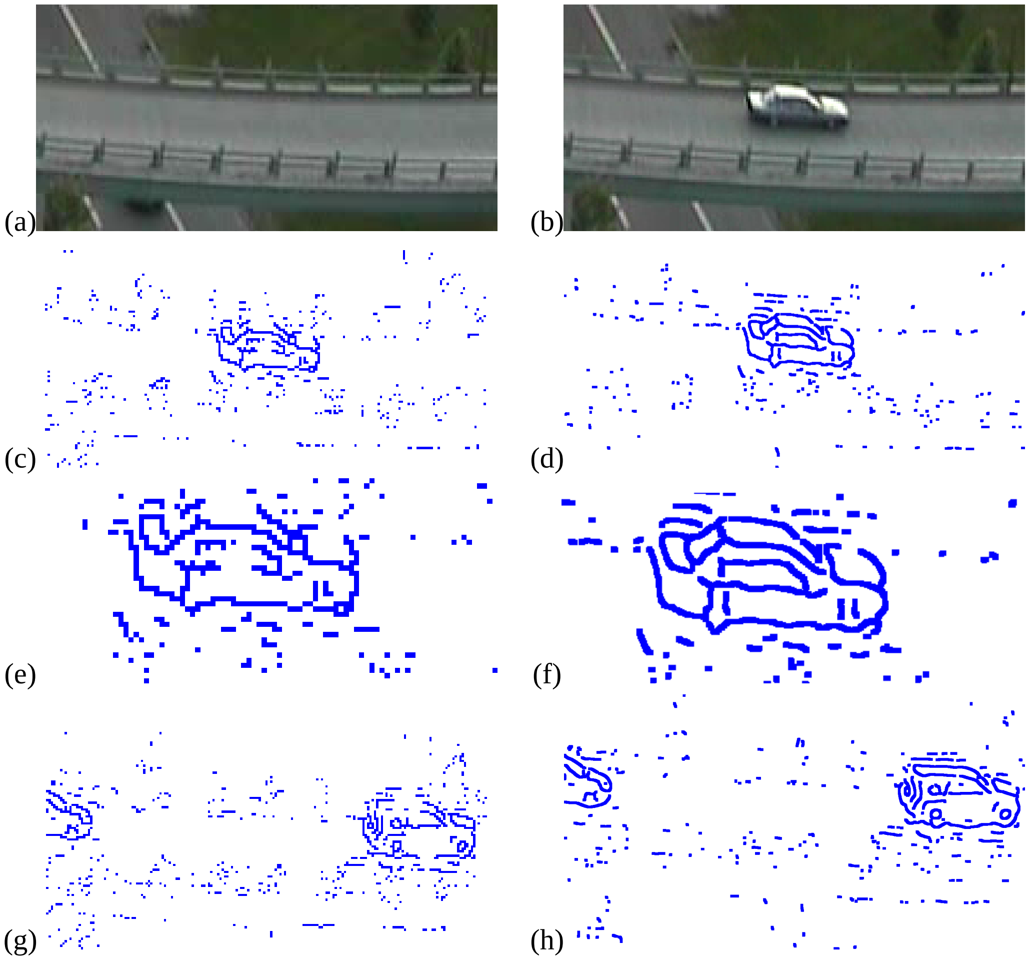


Figure 4.3: This figure compares the accuracy of background modeling using pixel and sub-pixel edge maps. (a) One of the background images, (b) latest observed image, (c) Pixel-edge foreground detection, (d) Sub-pixel edge foreground detection, (e) zoomed image of (c), (f) zoomed image of (d). Note how the roof of the car disappears in the case of pixel-edge modeling as the railing behind the car has the same orientation as of the roof edges. Another example with a jeep against the railing with output based on (g) pixel edges and (h) sub-pixel edges.

the edge is on. The orientation is computed as the normal direction to the gradient, ∇I. In this case, χ at a pixel location (i, j) models the orientation of the edge $\theta \in [0, \pi)$ if (i, j) is an edge otherwise χ is assigned a predefined constant outside the range of θ

$$\chi = \begin{cases} \theta \in [0, \pi) & (i, j) \text{ is an edge} \\ -100 & (i, j) \text{ otherwise.} \end{cases} \tag{4.14}$$

Note that θ needs to be circular, *i.e.*, 0 and 2π are the same. Since θ lies on a circle, the maximum distance should be π. Therefore, the values closer to 2π are closer to zero but a simple difference gives a huge value. This motivates to change the difference function for two variables χ_1 and χ_2 belonging to $[0, 2\pi)$

$$\chi_1 - \chi_2 = \begin{cases} \chi_1 - \chi_2 & for \quad |\chi_1 - \chi_2| \leq \pi \\ \chi_1 - \chi_2 - \frac{\chi_1 - \chi_2}{|\chi_1 - \chi_2|} 2\pi & for \quad |\chi_1 - \chi_2| > \pi \end{cases} \tag{4.15}$$

So, when $|\chi_1 - \chi_2| > \pi$, there can be two cases: *(i)* for $\chi_1 > \chi_2$, it becomes $(\chi_1 - 2\pi) - \chi_2 = \chi_1 - \chi_2 - 2\pi$ and *(ii)* for $\chi_1 < \chi_2$, it becomes $\chi_1 - (\chi_2 - 2\pi) = \chi_1 - \chi_2 + 2\pi$.

Due to the discretization errors during non-max suppression of pixel edge maps lead to unnecessary broad background models: a background edge halfway between the pixels will require both pixels modeled as background, thus unnecessarily "blurring" the background model, which in turn reduces sensitivity to detecting figures. Instead, a background model based on edge-maps with sub-pixel position of the images is proposed.

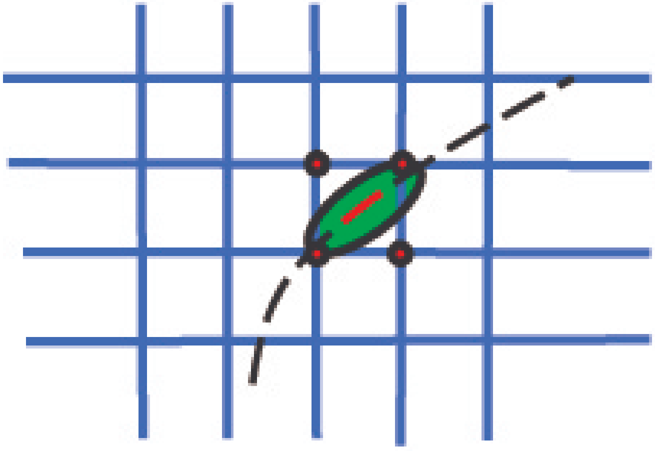


Figure 4.4: Covariance $\Sigma_{x,y}$ (green ellipse) for an edge distribution and four red dots shown represents the sites for the edge under consideration. Any edge which lies in the distribution is considered to be background.

Sub-pixel edges with orientation: Sub-pixel edge-maps attains high precision and accuracy in addition to being invariant to illumination changes and accommodates small translations easily.

The advantage of modeling sub-pixel edges becomes evident in scenes with cluttered backgrounds where edges from a figure can share the same pixel as well as the same orientation as shown in Figure 4.3. Figure 4.3(c) shows output based on pixel edge-maps. Note the roof of the car aligns with the rail in the background and both of them have similar orientation. Since localization of the position for pixel-edge maps is poor, it detects the roof as background. But by modeling sub-pixel edge-maps most of such cases can be resolved and thus, increasing the true positives. However, it becomes difficult to model sub-pixel position as it is no longer fixed to the grid. The authors propose an approach to overcome this problem which is discussed below.

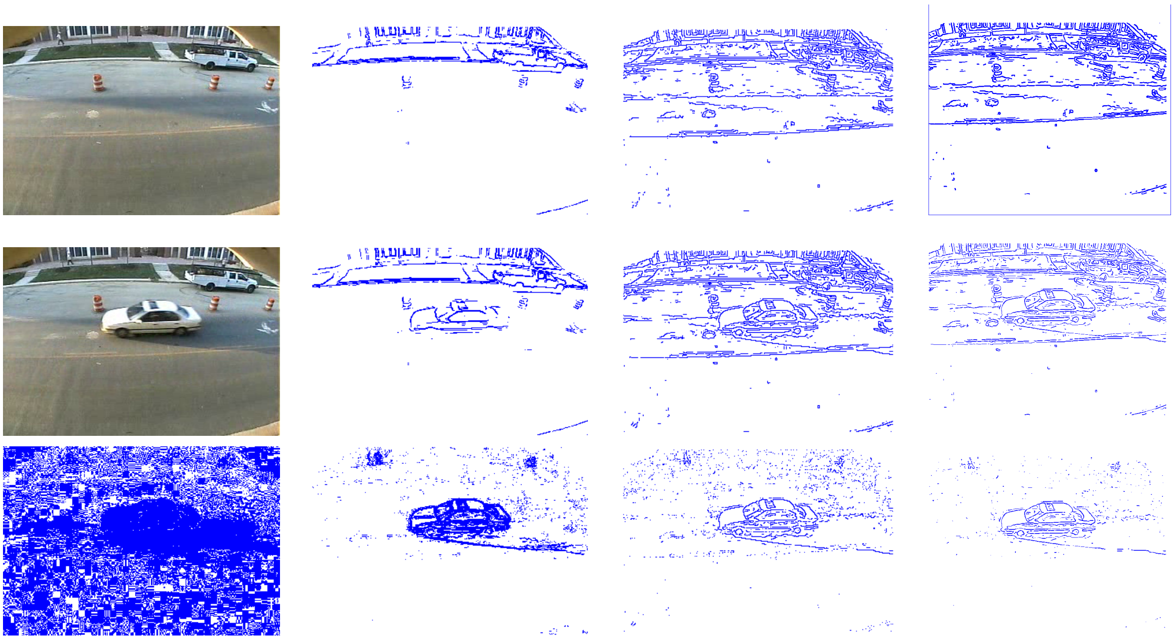
Sub-pixel edge-maps are obtained on a video sequence using a modified Canny edge detector which computes third-order derivatives [88] as represented by a set (x, y, θ) for each edge, where $x, y \in \mathbf{R}$ are the sub-pixel positions and $\theta \in [0, 2\pi)$ is the orientation. Since the edges are sub-pixel, we associate each edge to its corresponding *"sites"* (neighboring pixels) as illustrated by red dots in Figure 4.4. Note "sites" acts as a placeholder for the distribution of χ. $\chi(x, y, \theta)$ is a 3D variable in this case. It allows for sub-pixel accuracy for modeling the distribution of edges across frames. Each of the sites, s, has a MoG for $\chi(x, y, \theta)$ in which i^{th} Gaussian component would have a covariance Σ_i^s and a mean μ_i^s. The current edge observation $\chi(x, y, \theta)$ belongs to the foreground if $\forall s \in S$, Equation 4.11 holds,

$$(\chi^{k+1} - \mu_i^s)\Sigma_i^{s^{-1}}(\chi^{k+1} - \mu_i^s)^T > d^2 \quad \forall i \in \ N_b \ \ \forall s \in S \tag{4.16}$$

where S are the four sites. Most of the edges except spurious ones are samples of a curve in the image, Figure 4.4. Since an edge can slide along the curve, one would expect a large variance in the tangential direction and small variation along normal as shown in Figure 4.4. Empirically, larger variation along the curve was observed.

Parameter	Meaning	Default	Range
τ	threshold for edge detection	1.5	0-100
N	# of components of MoG	4	1-inf
L	moving window size of history of observations	20	1-inf
τ_w	threshold for declaring the components to background	0.5	0-1
d	maximum Mahalanobis distance allowed	2.5	0-inf

Table 4.1: System parameters



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

Figure 4.5: The effect of change in the gain of the camera is depicted for different background modeling schemes. Top Row: a background input image, and its (b) gradient-map, (c) edge-map and (d) sub-pixel edge-map. Middle row: (a) a new input image with a change in the gain of camera, and (b) intensity gradient, (c) edge map and (d) sub-pixel edge-map. Bottom row: Foreground detection results based on (a) intensity, (b) intensity gradient, (c) pixel edge map and (d) sub-pixel edge-map. Observe that the extent of the spurious responses reduces from left to right.

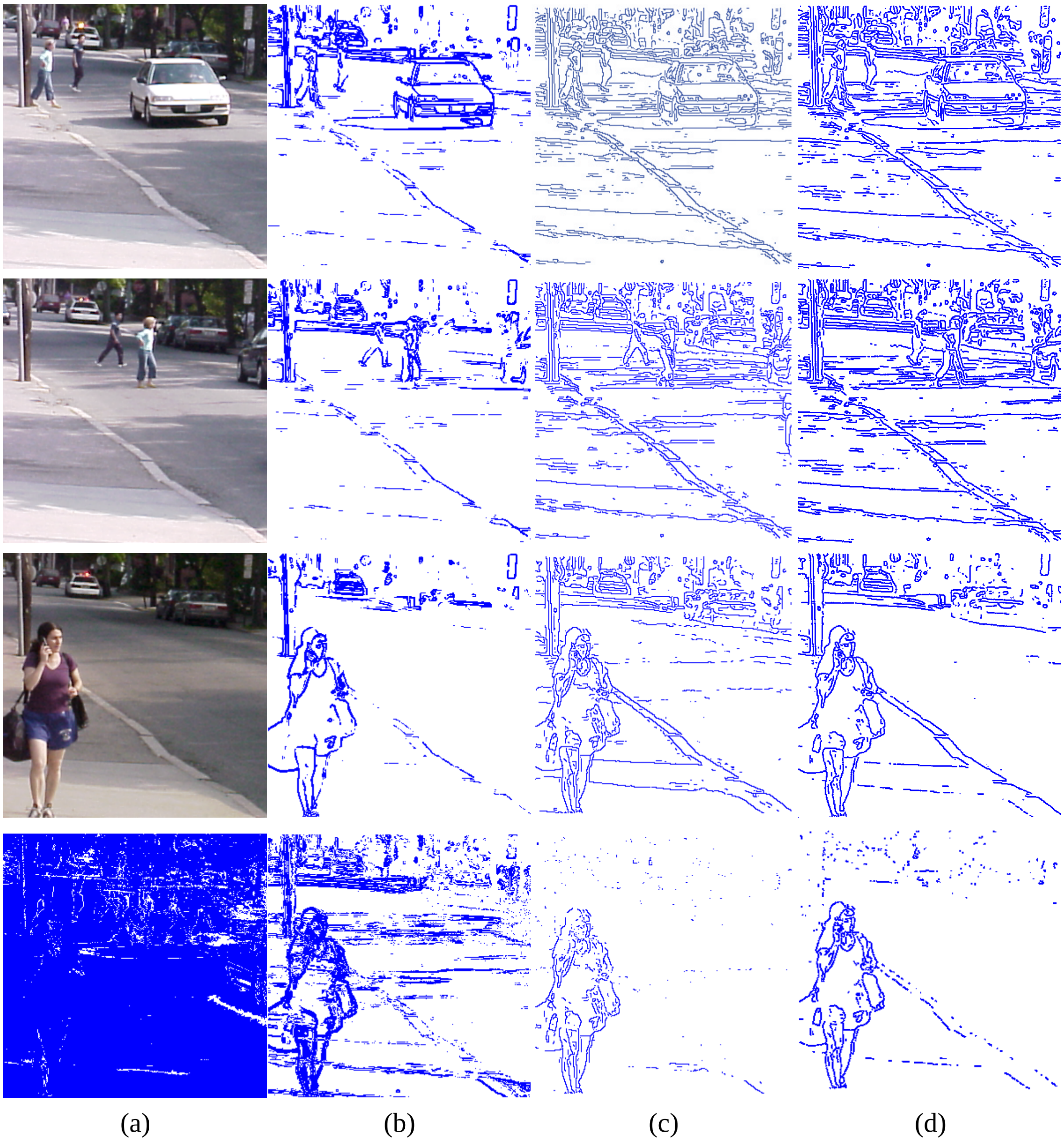


Figure 4.6: The effect of sudden illumination change on different background modeling schemes is illustrated. First and Second row: a pair of typical (a) background images, and their (b) gradient maps, (c) edges and (d) sub-pixel edges. Third row: (a) a frame when the illumination has changed and its (b) intensity gradient, (c) edge map and (d) sub-pixel edgemap. Fourth row: Foreground detection using (a) intensity, (b) intensity gradient, (c) edge map and (d) sub-pixel edgemap.

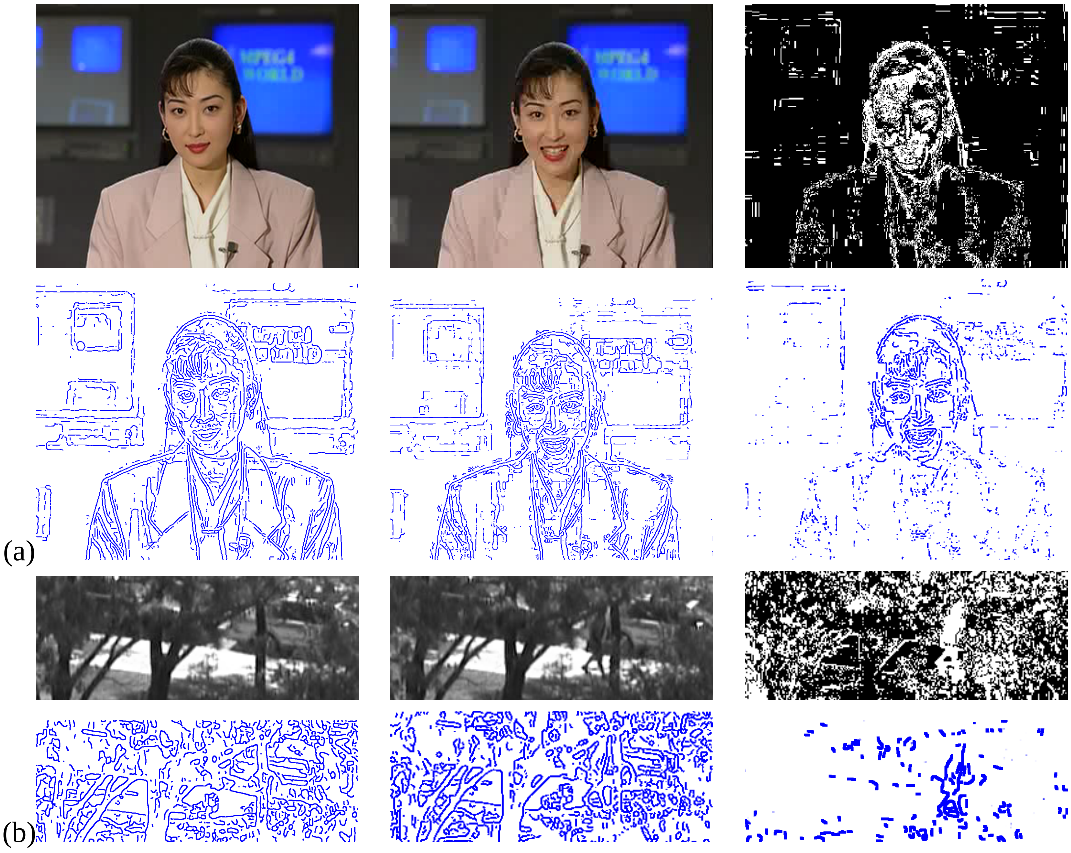


Figure 4.7: A comparison on standard video sequence (a) “Akiyo” and the (b) “occluded pedestrian” video sequence. The first column shows a typical frame and its sub-pixel edgemap used for the background model, the second column shows a new frame for which a figure needs to be segregated and the last column shows the foreground detection. It is clear in all cases the sub-pixel edge is more selective.

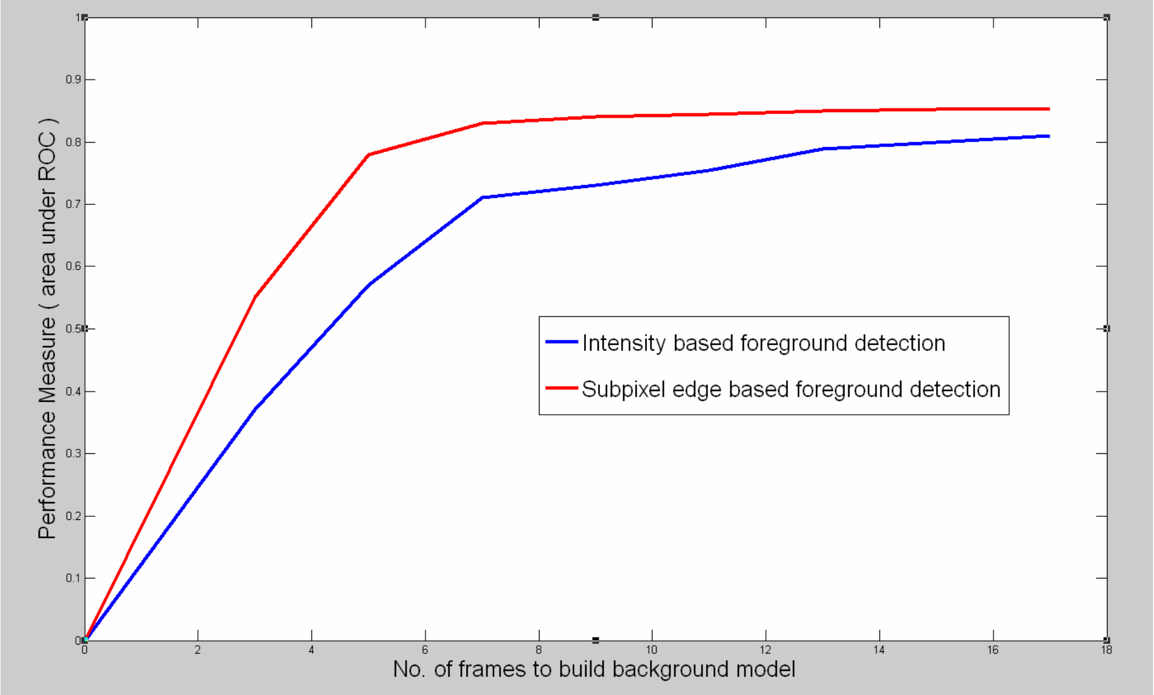


Figure 4.8: Plot of detection rate versus number of frames to build the model on sequence in Figure 4.3 for intensity and sub-pixel edges based method.

4.5 Experiments & Results

The four background models based on: *(i)* intensity, *(ii)* gradient, *(iii)* pixel edges, and *(iv)* sub-pixel edges were compared qualitatively and quantitatively. *Qualitatively,* observe the differences among foreground detected by these models in Figures 4.6 and 4.5. Figure 4.6 shows results on video sequence undergoing sudden illumination change due to clouds in the sky and Figure 4.5 shows results on video sequence undergoing sudden illumination change due to change in the gain of the camera. Figure 4.10 shows results of edges and intensity based modeling on different frames of a video under illumination change. The results on a widely used video sequences Akiyo in multimedia are shown in Figure 4.7. Observe in this sequence that the foreground occupies most of the scene and its very slowly moving, so it becomes really difficult to model the intensity of the background for the video sequence. The sub-pixel edge based method is able to detect foreground as compared to the intensity based methods. The last sequence shows how robust the method is in images with trees and bushes.

Intensity-based methods require a lot of frames to learn the background model especially when a foreground object is moving slowly like Figure 4.7. This happens because of homogeneous regions moving slowly so that a pixel might observe the same intensity for a long duration and model it as a background. But edges have an advantage as they are very sparse. The edges might face a similar problem. A simple example would be a checker-board pattern moving with the displacement equal to the width of each square. Definitely such cases are rarely occur than the situation described for intensity based approaches.

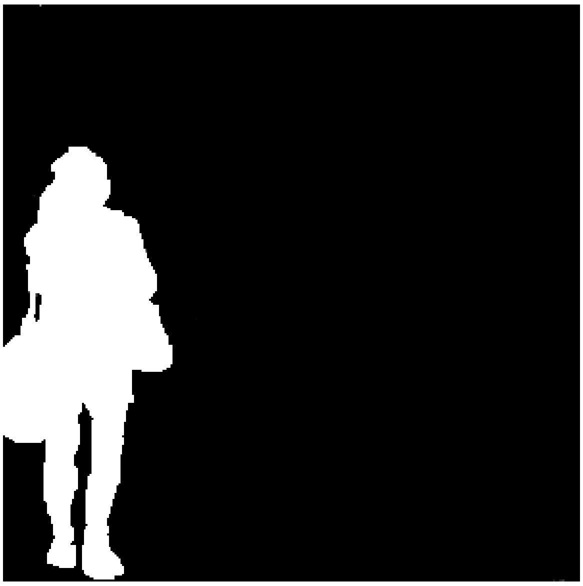
Quantitatively, ROC curve in Figure 4.9(b), quantify the differences in the performance of the four approaches on the sequence shown in Figure 4.6. Ground truth was marked for each of the four methods manually for 5 frames, Figure 4.9(a) . The ground truth is a binary image of foreground object. The false positives and true positives were recorded by varying the detection threshold τ for each of the approaches. An important thing to note is if a video with constant illumination is considered then intensity-based approach and the approach proposed in this work would performing similarly. The scenarios with constant illumination would be mainly indoor and have non-shiny surfaces as shiny surfaces can act as a light source. But for outdoor scenes where illumination changes quite a lot the proposed method outperforms intensity-based approach.

The experiment to compare number of frames required to build a background model for intensity based approach versus edge-based approach shows the edge based approach requires fewer frames to build a background model. A video sequence with no sudden illumination change was used for *(i)* intensity and *(ii)* edges. ROC curves for each of the frames were computed individually and the area under these curves were plotted versus the frame number as shown in Figure 4.8. This plot shows that edge-based model attains high performance in fewer frames than intensity based methods.

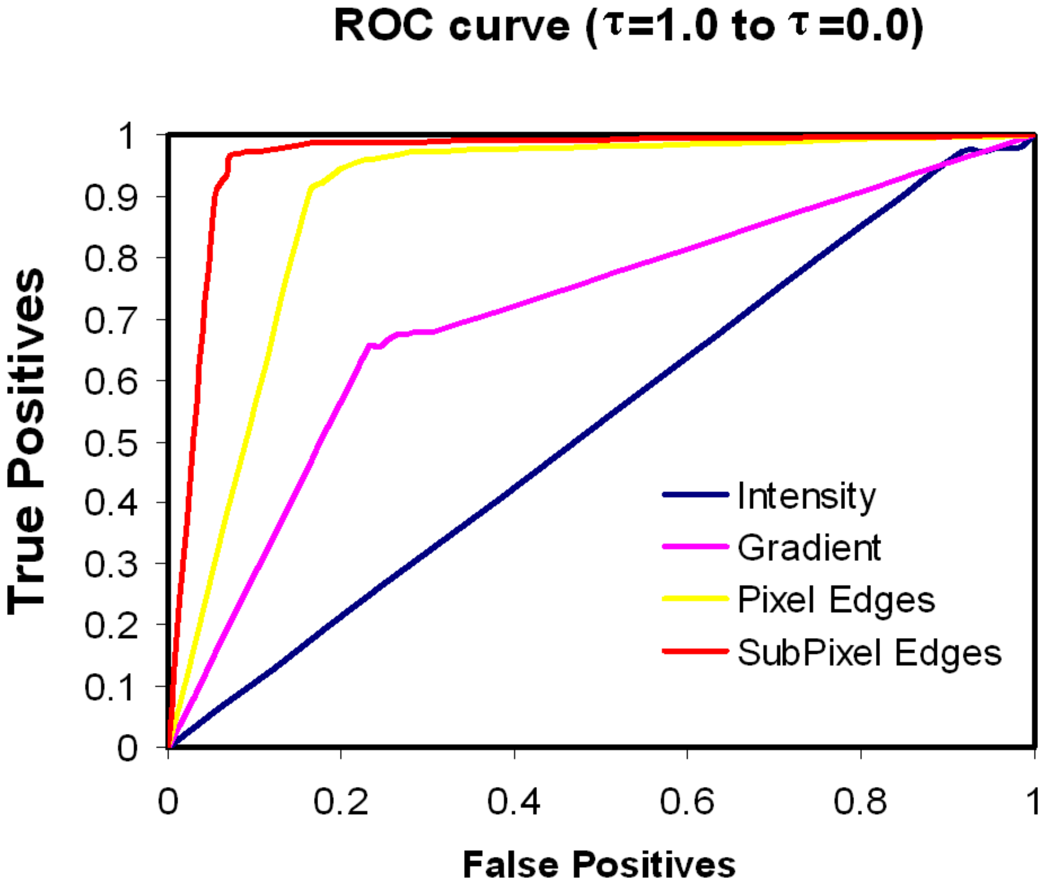
The limitation of the approach proposed is that edges detected as foreground do not give a region or boundary where as in case of intensity pixels of a foreground object are often contiguous. But the edges can be linked but due to misdetections or false alarms it might not output a closed shape. But the approach described in **??** shows how to enrich these segmentation as well as provide local linking of these edges.

4.6 Conclusion

In this work, a novel idea of using edges with their sub-pixel position and intensity to model the background to detect moving objects. This work shows edges are least susceptible to illumination changes and the background modeling technique introduced by [86, 51] requires fewer frames to model the background using edges as compared to intensity.



(a)



(b)

Figure 4.9: (a) An example of the ground truth image. (b) ROC curve obtained for foreground detection using Intensity (Black), Gradient (Pink), Pixel Edges (Yellow) and Subpixel Edges (Red) for sequence Figure 4.6

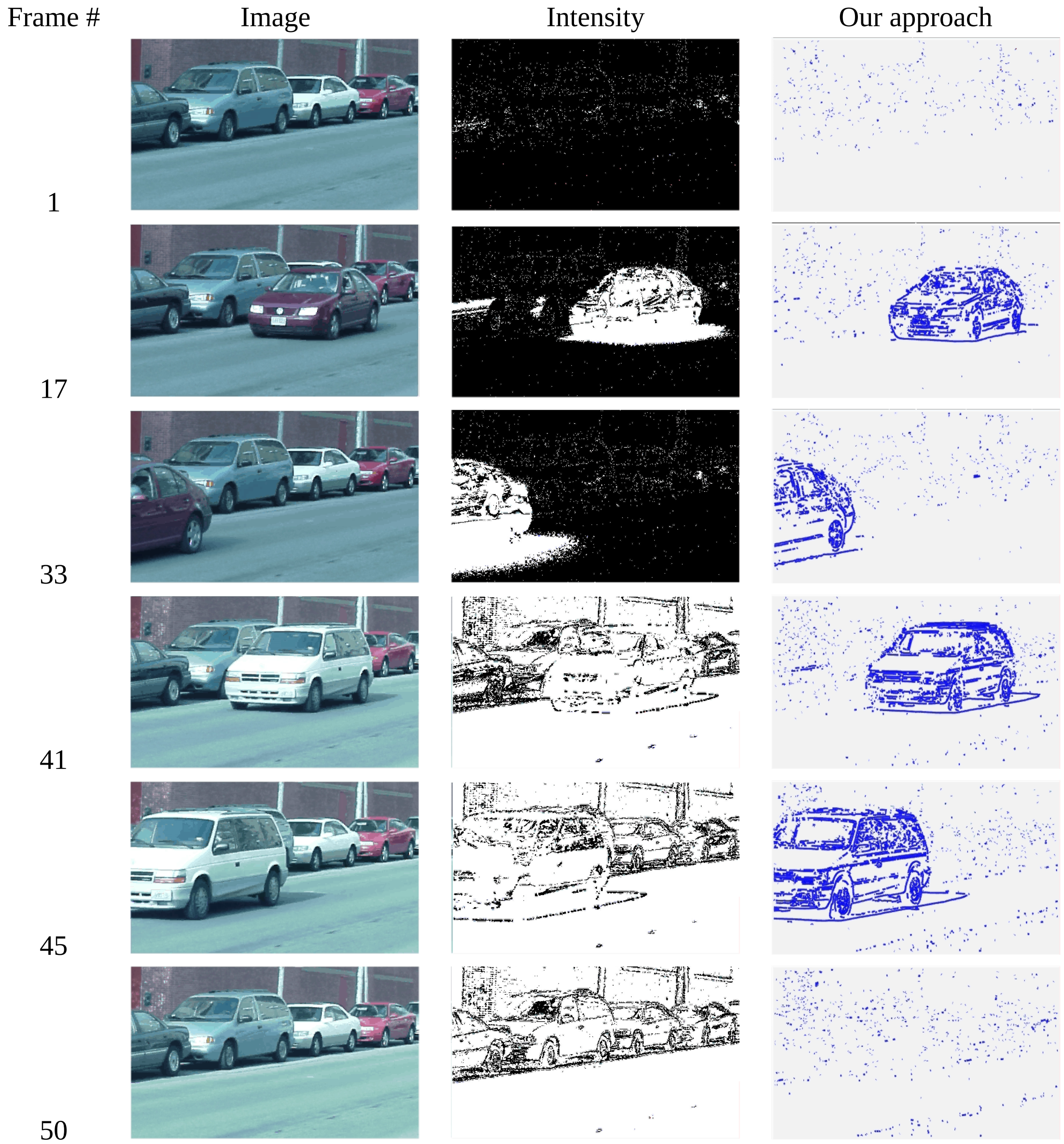


Figure 4.10: This figure shows the results of our approach and intensity based approach on multiple frames of a video sequence where illumination changes at frame 37.

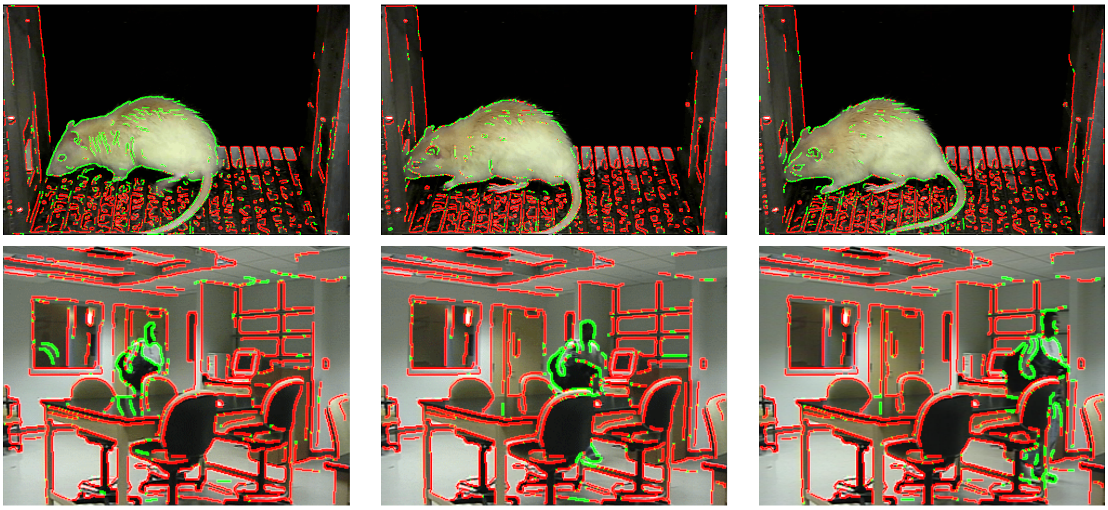


Figure 4.11: Two additional examples highlight the effectiveness of background modeling of edges. Figure and background edges are colored in green and red, respectively. First row: The mouse video is obtained from psychology department at our university where they want to do a behavioral study of the mouse by tracking its head. Second row: a moving person (note the reflections of the person from the window and table are captured).

Chapter 5

Multi-Frame Enrichment of Motion Segmentations

5.1 Introduction

Numerous computer vision applications such as surveillance, automated vehicles navigation, and robotics, among others, require segregated moving objects. However, motion segmentation of objects in real scenes can be challenging due to factors such as changes in illumination, limitation of motion models, the presence of, multiple objects, occlusion, blending of foreground into background, *etc.* These factors result in poor segmentations of moving objects as shown in Figures 5.1 and 5.2 which depicts missing spurious structures from other objects and have some missing structures. These degradations of motion segmentations hold for segmentation using regions, curves, edges or features. We observe, however, that the degradations of missing, and spurious and deforming structures are not consistent over adjacent frames, mainly because they are not structural but only due to temporary alignment or combination of 3D spatial configurations. As these configurations are altered, so are the type and extent of the degradation. What remains invariant is the 3D structure of the objects and the background. The goal of this work is to integrate information from several adjacent frames to improve the quality of segmentation.

Existing approaches for motion segmentation use different representation to establish the correspondence between adjacent frames namely, *(i)* regions (connected set of pixels), *(ii)* features like corners or SIFT, and *(iii)* curves or edges. In all these representations the correspondence required for motion segmentation can break down due to factors such as change in illumination, occlusions, blending of objects, *etc.* Feature-based approaches mainly suffer from instability of feature from frame-to-frame. This instability is caused due to change in viewpoints, specular reflections and

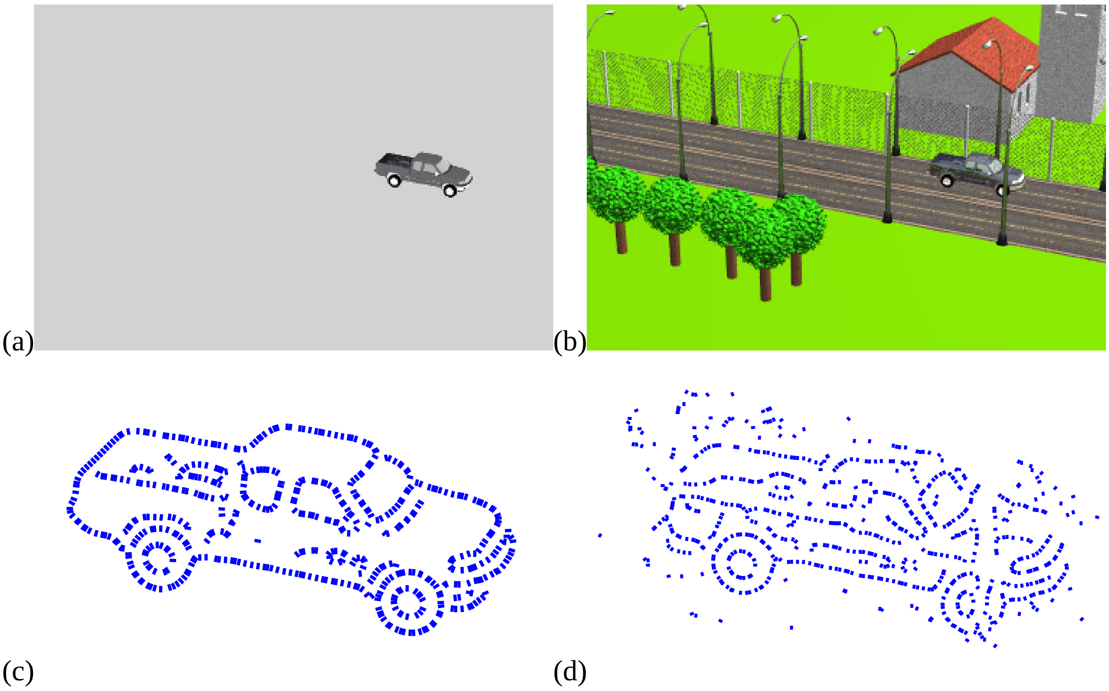


Figure 5.1: This synthetic video illustrates some of the problems of motion segemtnation in a controlled setting. (a) A single frame of a video of a truck moving to the right. (b) A single frame of a video of the same truck moving to the right, but now embedded in a rich scene. (c) The segmentation of the truck in (a) using background modeling of edges [47]. (d) The same approach applied to (b) shows gaps, missing edges, due to partial occlusion, spurious edges, highlight edges due to inter-reflections, *etc.* While the segmentation in (c) can be used for recognition, tracking, *etc.* that in (d) represents a greater challenge. Realistic videos show an even greater degree of complexity of interaction and therefore greater deviation from expected models and a greater degree of degradation.

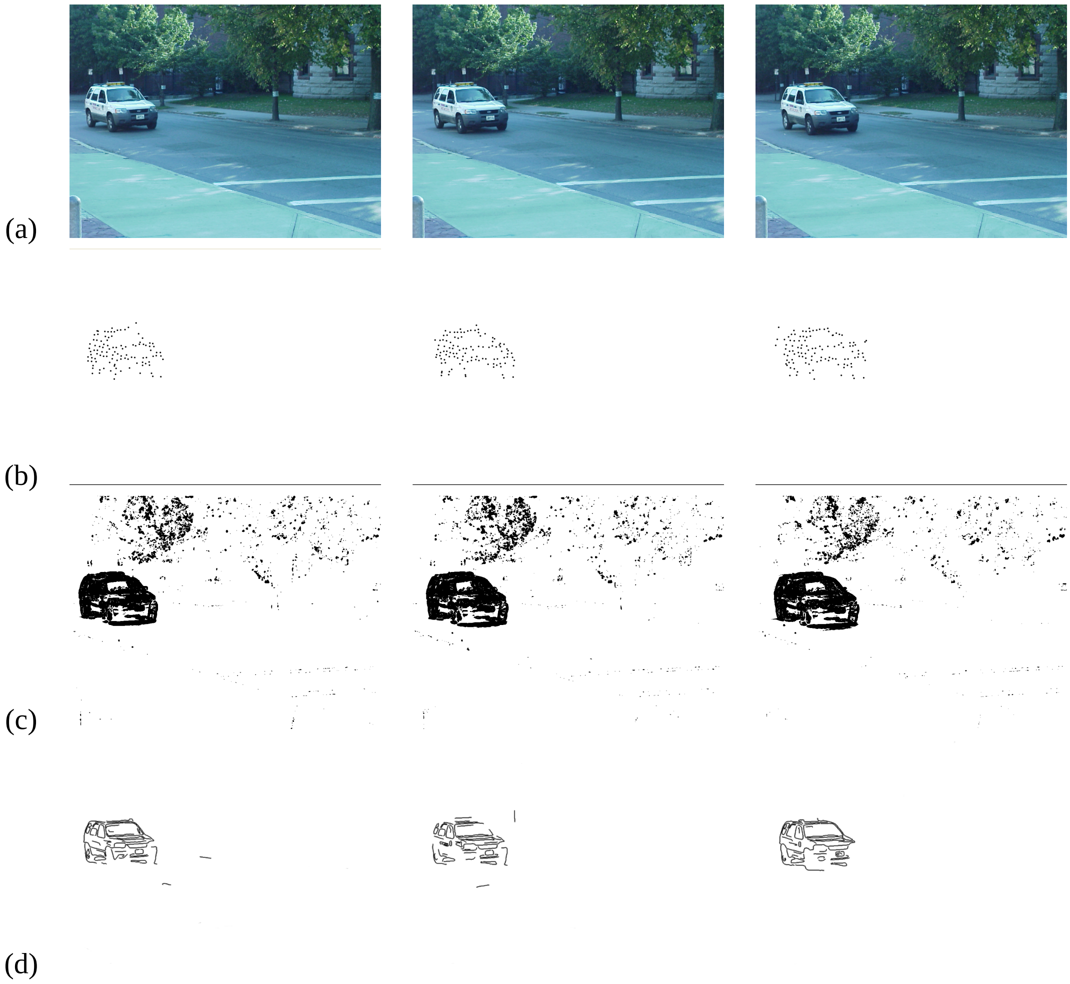


Figure 5.2: This figure shows the motion segmentations from different approaches. (a) Three frames of an outdoor video sequence and the motion segmentation of this video sequence using (b) KLT features based segmentation, (c) region based segmentation and (d) its curve-based object segmentation [48] is shown. Note how all these segmentation have noise and missing gaps.

occlusions and hence poor segmentations, as shown in Figure 5.2(d). For the case of regions, the correspondence between pixels is error-prone due to changes in illumination and inter-reflection from other objects, Figure 5.2(f). A comparison segmentations obtained using edges [47] in a scene with no background and illumination effects, Figure 5.1(a), versus segmentations obtained in a realistic scenes with illumination effects, Figure 5.1(b). Similarly, in the case of segmentation using curves, the curves have wrong correspondences due to transitions of the curves from one frame to another, *e.g.*, breaking of a curve into two in the next frame. This wrong correspondence result in segmentations with missing curves and some curves of different objects as shown in Figure 5.2 (b). Thus, segmentations on a single frame can be noisy and incomplete but they are complimentary over adjacent frames and hence can be used to *enrich* each other.

We now discuss an approach to use multiple adjacent frames to enrich the segmentations by integrating and fusing information across multiple adjacent frames (typically 5 or 7 frames). Note that this does not increase the resolution of the data, rather it enriches the segmentations. The proposed approach requires edge-maps of segmentations of objects from a video sequence as input and output an enriched figure edge/curve map for each frame of the video. The edge maps of the objects are computed from the motion segmentations using regions and features whereas the output of algorithms based on curves/edges [47, 48, 85] is directly feed into our approach. The choice of edges is motivated by *(i)* sparser than pixels in the region but richer than the points/SIFT which provides efficiency yet enough correspondences to find the alignment, *(ii)* robustness of edges to illumination changes and *(iii)* edges enables geometric consistency as compared to regions or features.

An overview of approach can be organized into several steps. First, the edge maps from neighboring frames are aligned onto a central frame so as to obtain a compound edge-map. The alignment which “transports” the temporal information into a common reference is based on a view of an edge as a sample of an underlying curve arising from a local planar patch. This assumption allows to use a Thin Plate Spline model to align the adjacent edge maps onto the central frame. The edge-maps are brought into alignment by minimizing *(i)* the distance between the edges and *(ii)* higher order derivatives of image flow.

Second, the edges from multiple frames need to be integrated. An edge from one frame can fill a gap in another or it can add to the spurious edges content. The decision as to whether an edge is spurious or adds to the geometric content is the notion of **geometric consistency**: an edge that together with other edges can arise from a local curve model (a circular arc in our case) has geometric support and is therefore structural. All other edges are spurious. This retains the edges which are consistent spatially as well as temporally and thereby removing some spurious edges and filling-in some gaps. Quantitative comparisons on synthetic video and qualitative comparisons on real video data shows that the resulting enriched composite edge map is significantly better both on synthetic and on real data.

This work is organized as follows: A brief review of related work is discussed in Section 5.2. The geometric alignment of edge map is presented in Section 5.3 and the geometric consistency is presented in Section **??**. The experimental results are described in Section 5.4.

5.2 Existing Approaches

To the best of our knowledge, the idea of using edge maps both for the alignment or registration of frames and as a main feature for fusion across frames is novel. However, the idea of integrating edge information to fuse image or edge-maps from multiple sensors has been proposed earlier. Abidi and Delcroix [2] proposed an approach to fuse range and intensity edge-maps. They employ two ideas: *(i) principle of token corroboration*: an edge in the final fused edge-map is retained if it is supported by either range and intensity edge-map, and *(ii) principle of belief enhancement/withdrawal* : edge in the final edge map is weighted depending on how similar the edge content in the two edge-maps are. Yocky [104] proposes fusion of multi-sensor images using wavelet transform. The idea is to fuse data which has compression along complementary datasets, *e.g.*, an image with high spatial resolution but low resolution color information and another image with low spatial resolution but high color information. The authors enhance an image from a sensor using high frequency components from the other sensor image.

The work by Yang and Blum [102] proposes a method using multiple neighboring frames for fusion of multi-sensor images. The approach is to use a statistical model for image formation whose parameters, and the final fusion image are unknown. An EM-based iterative algorithm is employed to solve for the parameters and the fused image iteratively. The temporal information or the neighboring frames add a constraint through consistency of parameters. The authors claim temporal information improves the fusion results.

Numerous approaches uses edges for registration of two images. The work by Stewart *et al.*[87] uses edges to register two images. The algorithm uses very high confidence matches as initialization and employs a region growing algorithm. The model selection is allowed from simple image translation to quadratic transformations. This implies that the objects need to be planar which is definitely not the case in general.

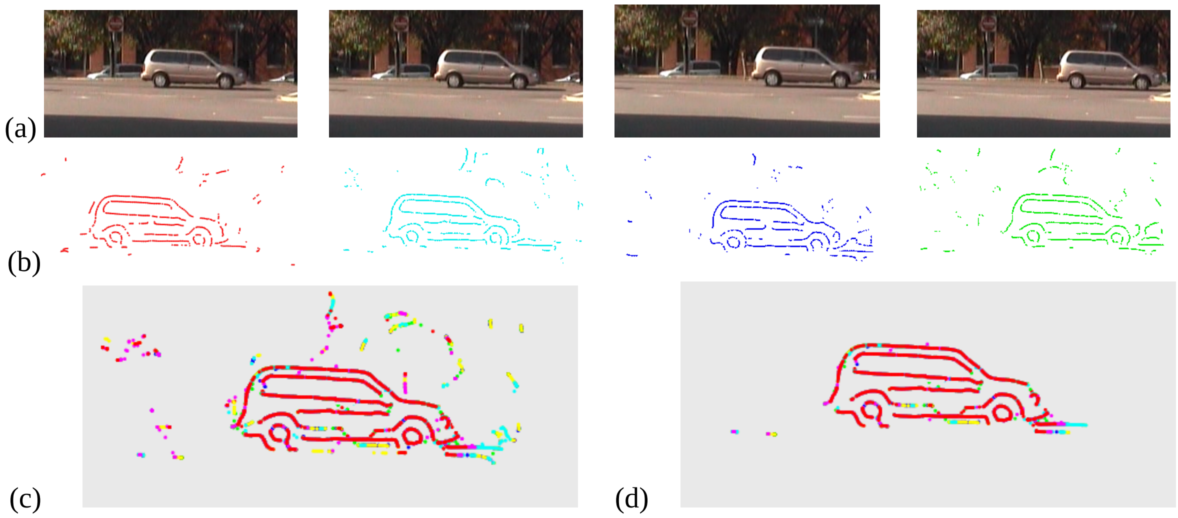


Figure 5.3: (a) A central frame with 3 frames following it; we do not show the 3 preceding frames for clarity of presentation. (b) The edge map of the central frame is red followed by three other edge maps. (c) The composite edge map show superimposition of all the registered edges on a central frame and **(d)** show consistent edges retained. Observe how most of the gaps are completed and the spurious edges are removed.

5.3 Alignment of Edge-Maps

The first stage of our approach aligns the motion segmentation edge maps in the neighboring frames onto a single **composite edge map** in the central frame under consideration. Typically the neighborhood window is 5 or 7 frames and each of these frames is individually registered to the central frame using pairwise alignment between each of the neighboring frames and the central frame. The pairwise alignment between two frames requires a transformation $\chi(\xi, \eta)$ for all (ξ, η) belonging to the object in the central frame, where χ is a vector field. Note that some points may have good correspondence in the other frames which are used to compute the complete χ. Similarly, the point (ξ, η) in the central frame is aligned with the point $(\bar{\xi}, \bar{\eta})$ in an adjacent frame, *i.e.*,

$$\begin{bmatrix} \bar{\xi} \\ \bar{\eta} \end{bmatrix} = \chi(\xi_i, \eta_i) + \begin{bmatrix} \xi_i \\ \eta_i \end{bmatrix}.$$

Since, the measurements of the motion segmentation of an objects are edge maps, consistency of any alignment vector field $\chi(\xi, \eta)$ with data is only at edges. Let $\{e_i, i = 1, ..., L\}$ and $\{\bar{e}_j, j = 1, ..., \bar{L}\}$ represent the edges in the central and adjacent frames respectively. The transformation $\chi(\xi, \eta)$ is constrained by the minimizing the following difference

$$d_c(e_i, \hat{e}_j) = d(e_i, \chi(\hat{e}_j) + \hat{e}_j) \tag{5.1}$$

where $\bar{e}_j = \chi(\hat{e}_j) + \hat{e}_j$ and $d(e_i, \bar{e}_j)$ represents the consistency of two edges in the composite map, to be defined in Section 5.3.1. Since the above constraint is spare, it is not sufficient to estimate χ and additional assumptions are required so that can be densely estimated. A first-order approximation of object surfaces as locally planar implies that $\chi(\xi, \eta)$ is a piecewise linear flow, leading to the minimization of second-order derivatives, *i.e.*,

$$\min_{\chi} \iint_{(\xi,\eta)} \left[|\frac{\partial^2 \chi}{\partial \xi^2}|^2 + 2|\frac{\partial^2 \chi}{\partial \xi \partial \eta}|^2 + |\frac{\partial^2 \chi}{\partial \eta^2}|^2 \right] d\xi d\eta, \tag{5.2}$$

which is justified below in Section 5.3.2. Finally, the optimization to find χ using the above data term and regularization is based on the iterative annealing method of Chui and Rangarajan [19], which is discussed in Section 5.3. We discuss each in turn.

5.3.1 Similarity Between Two Edges

We now discuss the degree of consistency of an edge e_i from one frame with an edge $\bar{e}_i$ from another frame, when superimposed on the central composite edge map, *i.e.*, where $\bar{e}_i$ has been laid out as $\chi(\bar{e}_i) + \bar{e}_i$. In order to understand the relationship between two edges e_i and $\chi(\bar{e}_i) + \bar{e}_i$, consider a 2D curve $\boldsymbol{\gamma}(s, t)$ moving and deforming from one frame to another, where s is the parameter along a curve at constant t where t is the time index. This curve $\boldsymbol{\gamma}(s, t)$ is sampled differently in different frames giving rise to distinct spatial locations for edges in different frames. Thus, when the two edge maps are aligned in the central edge map, the correspondence between edges is no longer one to one: an edge from one curve (red samples) comes from a sample of the same curve in a different frame that is no longer present. This forces a many to many mapping, which is also not accurate, Figure 5.4(a). A Euclidean distance representing "point to point" distance $d_p(i, j) = ||p_i - p_j||$ can lead to multiple correspondences and erroneous distance estimates. We propose to estimate the point to curve distance instead. Since the underlying curve $\boldsymbol{\gamma}(s, t)$ is unavailable and only the trace is available, the best estimate, namely the line extension of the edge, or when curvature information is available a circle is used. Thus, the distance of an edge e_i to the transported edge from another frame $\bar{e}_j$ is the distance between $\bar{e}_j$ and the line or circle extending e_i, Figure 5.4(b).

This algorithm works well in general but (i) it sometimes produces erroneous correspondence due to variation in sampling across curves and (ii) is computationally expensive. We now discuss our modification to address these problems. Specifically, let the edge position of e_i be $p_i = (x_i, y_i)$ and its tangent $t_i = (\cos\theta_i, \sin\theta_i)$ and similarly for $\bar{e}_i$. Then, the distance function between two edges then comprises three terms: (i) the perpendicular distance of an edge to the tangent of the other edge $d_\perp(i, j) = |(p_i - \bar{p}_j) \times \boldsymbol{t}_i|$, (ii) The difference between orientation of edges $d_\theta(\theta_i, \bar{\theta}_j) = |\theta_i - \bar{\theta}_j|_\pi$ and (iii) the Euclidean distance between position of edges $d_e(p_i, \bar{p}_j) = ||p_i - \bar{p}_j||$ to define a local

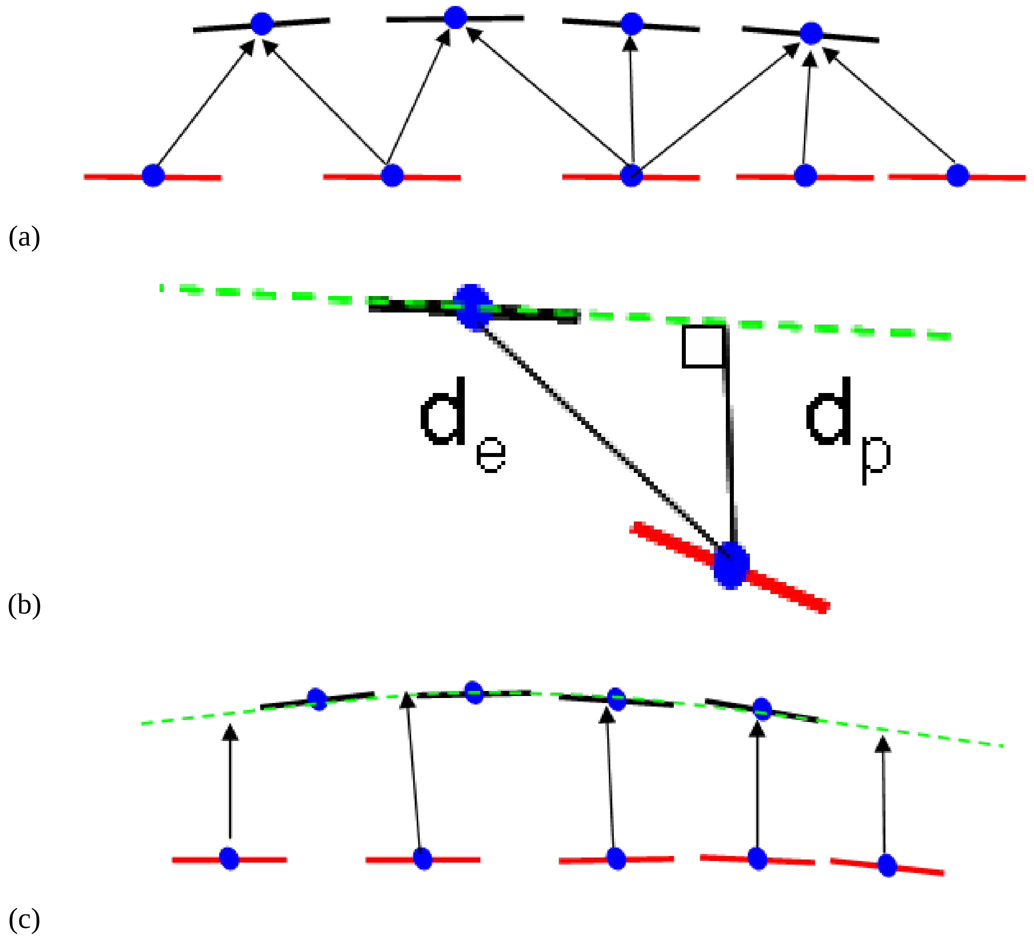


Figure 5.4: This figure illustrates the advantage of using point to curve distance. (a) Point to point distance would be problematic as the sampling of the curve is different as shown in red and blue edges. (b) an estimate of point-curve distance and (c) point to curve distance allows the samples from other frames (red) to align with the underlying curve (shown in green).

neighborhood over which the computation is meaningful. Then the similarity between two edges is represented by

$$d_e(e_i, \bar{e}_j, \sigma_e, \sigma_\theta, \sigma_\perp) = e^{-\frac{||p_i - \bar{p}_j||^2}{2\sigma_e^2}} e^{-\frac{|\theta_i - \bar{\theta}_j||^2}{2\sigma_\theta^2}} e^{-\frac{|(p_i - \bar{p}_j) \times \boldsymbol{t}_i|^2}{2\sigma_\perp^2}}, \tag{5.3}$$

where σ_θ, σ_p and σ_e are the uncertainties associated with each of the distances and are typically assigned as equal to $\sigma_\perp = 2.0$ pixels, $\sigma_\theta = \pi/6$ radians and $\sigma_e = 5.0$ pixels, Table 5.1.

5.3.2 Choice of Transformation

The objects generally considered for the scope of this work are assumed to be locally planar and far from the camera (high frame-rate). The following proposition shows that χ is given by piecewise linear flow using the above two assumptions.

Proposition 5.1. *The image motion for a locally planar point far from the camera can be approximated by linear flow model.*

Proof. Without loss of generality assume a simple camera model with unit focal length. Let (ξ, η) be the image coordinate of the 3D point (X, Y, Z) which can be related by

$$\begin{cases} \xi &= \frac{X}{Z} \\ \eta &= \frac{Y}{Z} \end{cases} \tag{5.4}$$

Differentiating the above equations with time,

$$\begin{cases} \dot{\xi} &= \frac{Z\dot{X} - X\dot{Z}}{Z^2} = \frac{\dot{X}}{Z} - \frac{X}{Z}\frac{\dot{Z}}{Z} = \frac{\dot{X}}{Z} - \xi\frac{\dot{Z}}{Z} \\ \dot{\eta} &= \frac{Z\dot{Y} - Y\dot{Z}}{Z^2} = \frac{\dot{Y}}{Z} - \frac{Y}{Z}\frac{\dot{Z}}{Z} = \frac{\dot{Y}}{Z} - \eta\frac{\dot{Z}}{Z} \end{cases} \tag{5.5}$$

Since (X, Y, Z) lies on a plane, it satisfies a plane equation given by $aX + bY + cZ = 1$. This constraint can be rewritten to obtain $\frac{1}{Z}$,

$$\frac{1}{Z} = a\frac{X}{Z} + b\frac{Y}{Z} + c = a\xi + b\eta + 1 \tag{5.6}$$

Since the object is far, the camera will have a small angle of view and large focal length and thus $\xi \ll 1$ and $\eta \ll 1$. The assumptions of small motion and object far from the camera gives $\xi\frac{\dot{Z}}{Z} \ll \frac{\dot{X}}{Z}$ and $\eta\frac{\dot{Z}}{Z} \ll \frac{\dot{X}}{Z}$. Equation 5.5 can be simplified to

$$\begin{cases} \dot{\xi} &= \frac{\dot{X}}{Z} \\ \dot{\eta} &= \frac{\dot{Y}}{Z} \end{cases} \tag{5.7}$$

Substituting Equation 5.6 in Equation 5.7,

$$\begin{cases} \dot{\xi} &= (a\xi + b\eta + 1)\dot{X} \\ \dot{\eta} &= (a\xi + b\eta + 1)\dot{Y} \end{cases} \tag{5.8}$$

Therefore the above equation is linear in image coordinates. Hence, the proof. ■

The above proposition shows the flow is locally linear. Since the object is piecewise planar, the flow, χ is piecewise linear. The work by Blake & Zisserman [10] showed that the piecewise linear flow, χ is obtained by minimizing the second-order derivatives of χ. The functional is given by

$$I[\chi] = \iint_{(\xi,\eta)} \left[\left(\frac{\partial^2\chi}{\partial\xi^2}\right)^2 + 2\left(\frac{\partial^2\chi}{\partial\xi\partial\eta}\right)^2 + \left(\frac{\partial^2\chi}{\eta^2}\right)^2 \right] \partial\xi\partial\eta, \tag{5.9}$$

whose Euler Lagrange equation is:

$$\frac{\partial^4 \chi}{\partial \xi^4} + 2\frac{\partial^4 \chi}{\partial \xi^2 \partial \eta^2} + \frac{\partial^4 \chi}{\partial \eta 4} = 0. \tag{5.10}$$

This is the well-known bi-harmonic equation, *i.e.*,

$$\Delta^2 \chi = 0,$$

which has several particular solutions including,

$$Ar^2 \ln(r) + Br^2 + C \ln(r) + D,$$

where

$$r = \sqrt{(\xi - \xi_0)^2 + (\eta - \eta_0)^2} = \|(\xi, \eta) - (\xi_0, \eta_0)\|$$

and where A, B, C, D, ξ_0 and η_0 are arbitrary parameters. Among these $\phi(r) = r^2 \ln(r)$ gives a solution.

$$\chi(\xi, \eta) = r^2 \ln(r) = [(\xi - \xi_0)^2 + (\eta - \eta_0)^2] \ln(\sqrt{(\xi - \xi_0)^2 + (\eta - \eta_0)^2}),$$

which is the radial basis function used in finding solutions when it is constrained by a boundary condition

$$(\bar{\xi_0}, \bar{\eta_0}) = (\xi_0, \eta_0) + \chi(\xi_0, \eta_0),$$

which leads to the Thin Plate Spline approach discussed next.

5.3.3 Thin Plate Spline

The biharmonic equation has been used in the Thin Plate Spline (TPS) methodology, introduced in the context of geometric design by Duchon [26] where a thin membrane is bent so that certain control points are moved to a desirable location and where the rest of the plate complies by minimizing its bending energy. This bending energy is exactly of the form of Equation 5.9. Thus, the thin plate spline satisfies control point condition (boundary point condition) while minimizing the bending energy. That the solution has a closed form and has lead to its use in a variety of setting, including image alignment [18, 19] and shape matching [7]. We now use this methodology for edge-based image alignment. where a series of boundary conditions or otherwise known as control points , are given as an edge in one frame at (ξ_i, η_i) mapping to an edge at another frame at $(\bar{\xi_i}, \bar{\eta_i})$

$$(\bar{\xi_i}, \bar{\eta_i}) = (\xi_i, \eta_i) + \chi(\xi_i, \eta_i) \quad i = 1, 2, ..., N, \tag{5.11}$$

Then, since each radial basis function centered at (ξ_i, η_i)

$$\phi(|(\xi, \eta) - (\xi_i, \eta_i)|)$$

is a solution to the biharmonic equation $\Delta^2\chi$, so is

$$\chi(\xi, \eta) = \Sigma_{i=1}^{N} w_i \phi(|(\xi, \eta) - (\xi_i, \eta_i)|).$$

Since a linear function also satisfies the biharmonic equation, and since this allows for a global affine transformation we can include this to get a more general solution

$$\chi(\xi, \eta) = \Sigma_{i=1}^{N} w_i \phi(|(\xi, \eta) - (\xi_i, \eta_i)|) + b^\top + B(\xi, \eta)^\top \tag{5.12}$$

where $b^\top = (\xi_0, \eta_0)$ and the spatial warping affine matrix B is a 2×2 matrix that enables global scaling, shearing and rotation . The free parameters are the weights $w_i, i = 1, ...N$ and the six parameters in b and B for a total of $2N + 6$ parameters. Restricting the solution space to have square integrable second derivatives leads to

$$\sum_{i=1}^{N} w_i = 0 \quad \sum_{i=1}^{N} w_i \xi_i = 0 \quad \sum_{i=1}^{N} w_i \eta_i = 0. \tag{5.13}$$

which provides 6 constraints in addition to the 2N boundary conditions of Equation 5.11. This set of linear equations can be solved in closed form. Specifically, let Φ be a $N \times N$ matrix defined as

$$\Phi_{ij} = \phi(|(\xi,\eta_i) - (\xi_j, \eta_j)|)$$

and let

$$X = \begin{bmatrix} (\xi_1, \eta_1) \\ (\xi_2, \eta_2) \\ \cdot \\ \cdot \\ \cdot \\ (\xi_N, \eta_N) \end{bmatrix}, \quad \bar{X} = \begin{bmatrix} (\bar{\xi}_1, \bar{\eta}_1) \\ (\bar{\xi}_2, \bar{\eta}_2) \\ \cdot \\ \cdot \\ \cdot \\ (\bar{\xi}_N, \bar{\eta}_N) \end{bmatrix}, \quad W = \begin{bmatrix} w_1^\top \\ w_2^\top \\ \cdot \\ \cdot \\ \cdot \\ w_N^\top \end{bmatrix}$$

so that the Equation 5.12 can be written as

$$\chi(\xi_j, \eta_j) = b^\top + B(\xi_j, \eta_j)^\top + \sum_{i=1}^{N} w_i \Phi_{ij}$$

or in matrix form

$$\bar{X} - X = \mathbf{1}_{N\times 1}.b_{1\times 2}^\top + XB^\top + \Phi W,$$

This can be merged with Equation 5.13 so that it can be written in block form

$$\left[\begin{array}{c} (\bar{X}-X)_{N\times 2} \\ \hline 0_{2\times 2} \\ \hline 0_{1\times 2} \end{array}\right] = \left[\begin{array}{c|c|c} \Phi_{N\times N} & X_{N\times 2} & \mathbf{1}_{N\times 1} \\ \hline X^{\top}_{2\times N} & 0 & 0 \\ \hline \mathbf{1}_{1\times N} & 0 & 0 \end{array}\right] \left[\begin{array}{c} W_{N\times 2} \\ \hline B^{\top}_{2\times 2} \\ \hline b^{\top}_{1\times 2} \end{array}\right]$$

This matrix equation can be inverted to solve for W, B and b. More generally, however, since the enforcement of control points *exactly* passing through the desired point has rather harsh implications on the smoothness of the resulting mapping χ, we obtain a smoother solution by allowing the control points to pass near the desired points with some cost in a traditional regularization framework, leading to a minimization of

$$\begin{aligned} E(W,b,B) \quad &= \quad \textstyle\sum_{i=1}^{N} \|(\bar{\xi}_i,\bar{\eta}_i) - ((\xi_i,\eta_i) + \chi(\xi_i,\eta_i))\|^2 + \\ &\quad\ \lambda \textstyle\iint \left[|\frac{\partial^2\chi}{\partial\xi^2}| + 2|\frac{\partial^2\chi}{\partial\xi\partial\eta}| + |\frac{\partial^2\chi}{\partial\eta^2}| \right] d\xi d\eta, \end{aligned} \tag{5.14}$$

where λ is the regularization coefficient balancing the data term and the smoothness term. We use $\lambda = 0.2$ throughout this paper. Now, functions of the form in Equation 5.12 which solve exactly the biharmonic equation are retained to minimize $E(W,b,B)$ which after substituing leads to

$$\begin{aligned} E(W,b,B) \quad &= \textstyle\sum_{j=1}^{N} \|(\bar{\xi}_j,\bar{\eta}_j) - b - B(\xi_j,\eta_j)^{\top} - \sum_{i=1}^{N} w_i\phi_{ij}\|^2 \\ &\quad + \lambda Tr(W^{\top}\Phi W), \end{aligned} \tag{5.15}$$

which can be minimized by solving

$$\left[\begin{array}{c} (\bar{X}-X)_{N\times 2} \\ \hline 0_{2\times 1} \\ \hline 0_{1\times 1} \end{array}\right] = \left[\begin{array}{c|c|c} (\Phi-\lambda I) & X_{N\times 2} & 1_{N\times 1} \\ \hline X^{\top}_{2\times N} & 0 & 0 \\ \hline 1_{1\times N} & 0 & 0 \end{array}\right] \left[\begin{array}{c} W_{N\times 2} \\ \hline B^{\top}_{2\times 2} \\ \hline b^{\top}_{1\times 1} \end{array}\right]$$

This matrix equation can be solved in a closed form by a QR decomposition of

$$[X_{N\times 2} \quad 1_{N\times 1}] = [Q^1_{N\times 3} \quad Q^2_{N\times(N-3)}] \begin{bmatrix} R_{3\times 3} \\ 0_{N-3\times 3} \end{bmatrix}$$

which gives [19]

$$\left\{ \begin{array}{l} W = Q_2(Q_2^\top \Phi Q_2 + \lambda I_{N-3})^{-1} Q_2^\top [\bar{X} \quad 1] \\ \left[\begin{array}{c|c} B & b \\ \hline 0 & 1 \end{array} \right] = R^{-1}(Q_1^\top [X_{N\times 2} \quad 1_{N\times 1}] - \phi W). \end{array} \right. \tag{5.16}$$

5.3.4 Representing Correspondence by Softassign

The previous section assumed that the correspondence between (ξ_i, η_i) and $(\bar{\xi}_i, \bar{\eta}_i)$ is known. However, finding this correspondence is part of the problem. We adopt Chui and Rangarajan's point set matching approach where the correspondence is represented by a matrix M whose elements represent the degree a point on one image matches a point another image. The energy is optimized based on this correspondence which then updates M. The process is repeated to convergence.

The algorithm of [19] is extended/modified for aligning edges by replacing pairwise point-point distance to point-curve distance. The correspondence is represented by a matrix M where rows represent edges in the first edge-map $e = \{e_i = (\xi_i, \eta_i), i = 1, ..., L\}$ and columns represent edges in the second edge-map $\overline{e} = \{\overline{e}_j = (\bar{\xi}_j, \bar{\eta}_j), j = 1, ..., \overline{L}\}$. Note that we are not assuming that $L = \bar{L}$ as in the last section. As in the softassign approach of [19] an additional column and an additional row are also added to represent outliers and missing edges, respectively. This extra column/row takes care of edges which do not have a match. The correspondence matrix has then dimension of $(L+1) \times (\overline{L}+1)$. As a permutation matrix, only one element in each row and each column would be one, indicating a one-to-one correspondence between the edges, that is if the one is not in the last column/row, or otherwise a spurious edge or missing edge, respectively. The matrix, however can accommodate a non-binary fuzzy representation when it is doubly stochastic [19]. This fuzzy correspondence is particularly important in corresponding edges since an edge is a sample of a curve and its true correspondence in another frame typically falls between two edges. Given a correspondence matrix M, a transformation χ can be obtained. Since each edge e_i map to all other edges to various degrees, an average correspondence $\hat{e}_i$ is computed by a weighted average

$$\hat{e}_i = \sum_{j=1} M_{ij} \bar{e}_i = M\bar{X} \tag{5.17}$$

The optimal transformation χ for the correspondences $(e_i, \hat{e}_i), i = 1, ..., N$ in the form of Equation 5.12 is then obtained by solving Equation 5.16 for weights W, b and B. Given a transformation χ, an updated correspondence matrix M can then be obtained. Specifically, each edge e_i is transformed to

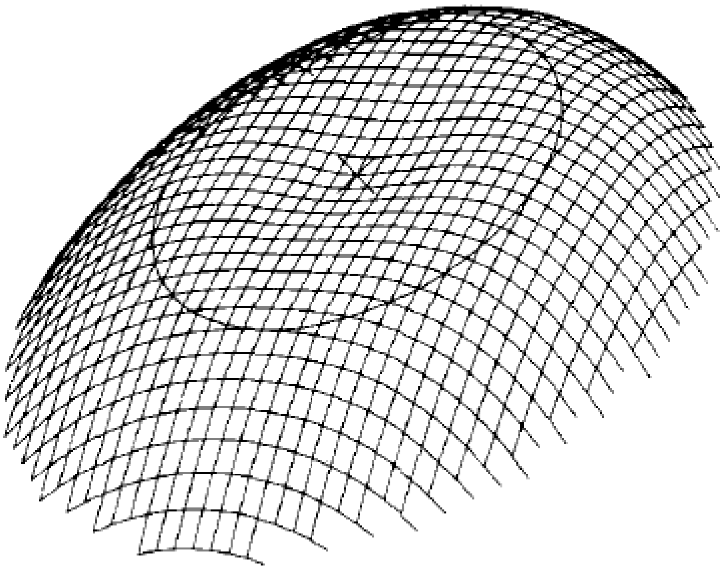
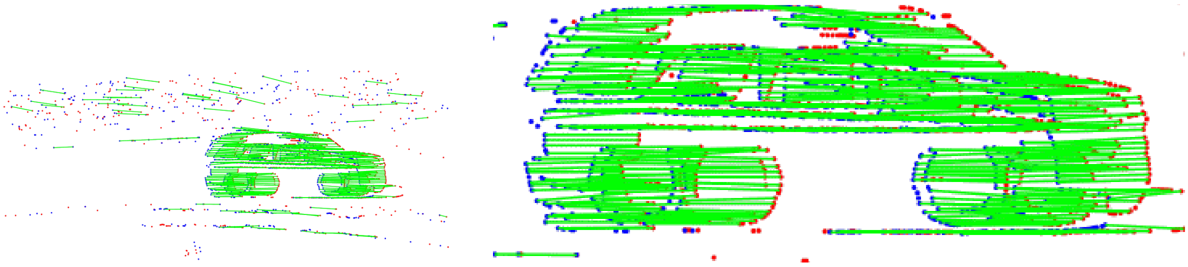


Figure 5.5: This figure shows the plot of $K(r) = -r^2 \log r$ from [11].

$\tilde{e}_i = e_i + \chi(e_i)$. The extent each edge e_i is similar to $\tilde{e}_i$gives the updated degree of correspondence between e_i and $\bar{e}_i$. We use the similarity distance of Equation 5.3 to determine M_{ij} as

$$M_{ij} = d_e(e_i + \chi(e_i), e_j, T\sigma_e, T\sigma_\theta, T\sigma_\perp),$$

where T is the temperature for annealing. The updated M is converted into a doubly stochastic matrix by iterative row and column normalization.



(a) (b)

Figure 5.6: (a) The correspondence between two edge maps is shown as green lines connecting corresponding edge points (red and blue) for two frames of an image sequence. (b) zoomed in (a). Observe that the majority of outliers are correctly deleted.

Initially, at a high temperature the elements of matrix M are assigned uniform values which implies all the pair-correspondences $\{e_i\} \times \{\bar{e}_j\}$ are equally likely. As the temperature is lowered,

M becomes non-uniform. Eventually it approaches a permutation matrix which ensures one-to-one correspondence. The output of this algorithm gives us mapping of $\{e\}$ to edgemap $\{\bar{e}\}$. Figure 5.6 shows the correspondence between two edge maps, where green lines connect points from one edge set (in red) to another edge set (in blue). Observe that in contrast to point-matching, the orientation of the edge allows for a higher degree of selectivity. This optimal correspondence M and its associated transformation χ are used to align edge maps from several frames to form a composite edge map as discussed earlier.

5.3.5 Efficient Alternative for the TPS Model

The closed form solution of the TPS, Equation 5.16, has complexity $O(N^3)$ where N is the number of edges. This can be prohibitive for larger images which can have ten of thousand edges. This motivates search for a more efficient scheme for finding χ without necessarily affecting the performance. We have found that the Clough-Tocher implementation [21] which uses piecewise cubic patch is used to speed up the computation to $O(N \log N)$ does not degrade the resulting transform χ.

The optimal transformation between two frames is now used to transform one edge map onto the other. Specifically, we for each frame we consider a neighborhood of 5 or 7 frames (2 or 3 frame before and after the central frame), and compute the optimal transformation χ between the central frame with every neighboring frame, and transform these edge maps onto a central frame to form a **composite edge map**. Figure 5.3 illustrates this process for a central frame whose edge map is shown in red and 3 subsequent frames following it (we do not show the previous frames for clarity of presentation), each with edge maps of different color. Figure 5.3(c) shows the composite edge map where the edge map of each frame is transported onto the central frame with the corresponding optimal transformation. Other examples of composite edge maps are shown in Figure 5.11(d) and 5.12(d). a comparison of edge maps and composite edge maps shows that composite edge maps are

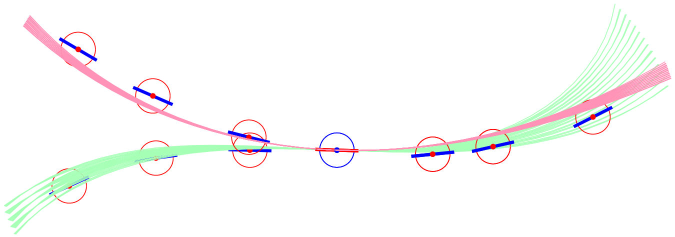


Figure 5.7: Multiple groupings or curvelet bundles from [92] shown in pink and green for an edge (blue circle). Each edge can have multiple groupings.

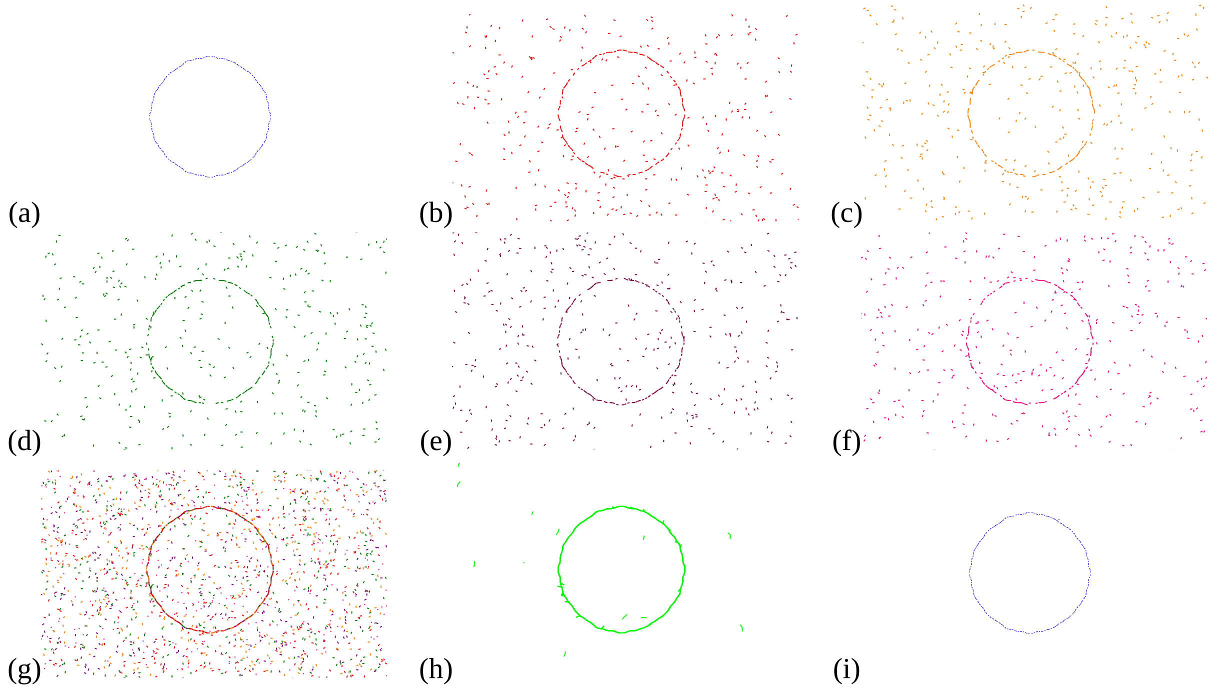


Figure 5.8: A synthetic simulation of a sequence of 5 adjacent frames which are samples from an ideal edge map (a), but where some edges are removed (20%) and where noise is added in the form of random edges (300%). Each of these five edge maps in shown in a different color (b-f). The composite edge map where all five edge maps are aligned and superimposed is shown in (g). The collection of all curve bundles is shown in (h). The edges not participating in the local curve models are labeled as spurious and removed, leaving behind only the structural edges of these only those with significant temporal presence are kept as shown in (i).

significantly richer in structure in that many gaps are appropriately filled. However, it is also evident that the composite edge map inherits the union of all edge maps spurious edges and is significantly noisier. The key to delineating spurious edges is geometric consistency.

Specifically, we define three classes of edges in the composite edge map. First, an edge that is a sample observation of a curve, which may not have been observed in the previous or in the following frames, but which is consistent with a curve constructed from edge samples form the other frames of from the same frame, Figure **??**(a). This is called a **structural edge**. Second, an edge may not have spatial support, but rather has temporal support in that it is consistently observed when the image motion has been taken into account by aligning images into a central frame. This is called a **frequent edge**. Third, an edge that satisfies neither of these constraints is a **spurious edge**, **??**(b). The determination of a structural edge requires that all possible local curves through edges potentially interacting edges from all frames be identified.

While the enumeration of all possible curves through a discrete set of edges is known to be intractable, it was proposed in [89] that a notion of geometric consistency tames the combinatorial explosion. Specifically, given expected edge location and measurement noise, pairs of edges is a small (5×5) or (7×7) neighborhood define a bundle of curves in a low-order geometric Taylor Expansion of curves, typically a circular arc or the Euler Spiral. Pair of pairs edges can potentially form a triplet of geometrically consistent edges if their curve bundles intersect, and so on to n-tuplets, which in our case $n = 6$ or $n = 7$. In this way, the vast majority of discrete combination of edges are discarded early in the process, retain only geometrically, meaningful local groupings, Figure 5.7 illustrates the two septuplet curve bundles formed for a given edge.

Spatiotemporal consistency of edges in multiple frames then translates into discovering geometric consistency in a composite edge map. All edges participating in a viable curve bundle has the potential to have arisen from a curve whose samples are disturbed spatially and/or temporally and as such are structural edges. The remaining edges if they are not temporally consistent are labeled as spurious edges. This idea is demonstrated on a simple circular structure, Figure 5.8. The simulation assumes that the ideal image, Figure 5.8(a), is viewed with a process that generates gaps by eliminating 20% of the edges and introduces spurious edges (100%), as shown in the five sample frames Figure 5.8(b-f), each shown in a distinct color. The pairwise optimal transformation, as described in Section 5.3.2 is used to generate a composite edge map, Figure 5.8(g). The curve bundles for each edge in the composite edge map are drawn in green in Figure 5.8(h).

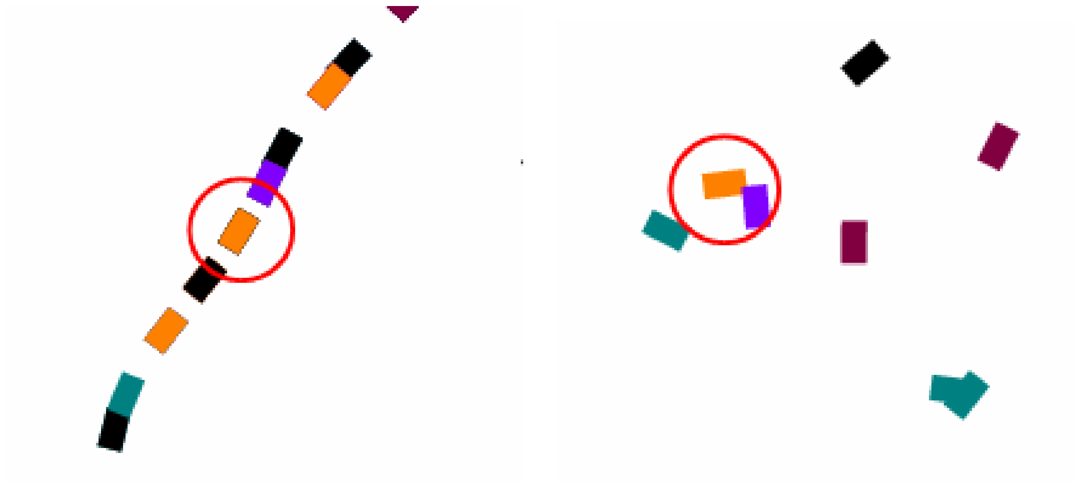

(a) (b)

Figure 5.9: The two types of edges in the composite map: (a) structural edge: an edge form a frame is supported by edges from other frames is that it can be considered as a sample of curve. (b) spurious edge: an edge is spurious when it is neither structure or frequent.

Aside from frequent edges, there can be three scenarios *(i)* edges not participating in any curve bundles; *(ii)* edges with curve bundles which do not have sufficient temporal presence, *i.e.*, the participating edges are from a few frames only. In our system, curve bundles formed from fewer than $\tau = 50\%$ of frames are structural but do not have significant **temporal presence**; *(iii)* edges which are structural and which have temporal presence beyond τ frames, figure 5.8(i). We refer to the latter set of edges the **enriched edge map**.

5.4 Experimental Results

The results of our approach on several synthetic and real videos are demonstrated both qualitatively and quantitatively. Figures 5.11(c) and 5.12(c) represent individual frame edge maps while Figures 5.11(e) and 5.12(e) depict enriched edge maps, respectively. Observe how gaps in individual frames are filled in and how spurious edges are discarded. Figures 5.11(d) and 5.12(d) show the geometric consistency of structural edges in that contours representing the object appear multicolored implying successful integration is possible across frame as validated in Figure 5.11(e) and 5.12(e) respectively. On the other hand, non-interacting edges are successfully discarded.

Observe in particular the success of this approach in dealing with partial occlusion where a vehicle is temporarily occluded by a pole, Figure 5.12(a). In individual frame, the figure edge-map is incomplete and incorrect as it contains edges of the occluding object, Figure 5.12(c). Note how the occluded regions are filled-in and how the enriched edge maps gives a complete figure which remains stable across video frames. Another important observation is that occluder's edges are discarded as the motion of the object and the occluder are different. The proposed approach provides a *more reliable and more complete* foreground edge-maps. Figure 5.16 further shows the comparison of raw figure edge-maps and the enriched edge-maps in detail.

In addition to the above qualitative comparisons, the quantitative performance of our approach is also evaluated. The task is to compare enriched edge maps to the raw edge maps as compared

Parameters	Meaning	Default	Range
$\sigma_\perp$	standard deviation allowed for perpendicular distance	2.0	0.0-inf
σ_e	standard deviation allowed for Euclidean distance	5.0	0.0-inf
σ_θ	standard deviation allowed for difference between orientations	$\pi/6$	0.0-π
λ	weight parameter between data term and regularization term. $\lambda = 0$ uses all data term	0.1	0-inf
τ	threshold for no of frames participating in the geometric consistency (%)	50	40-80%

Table 5.1: System Parameters

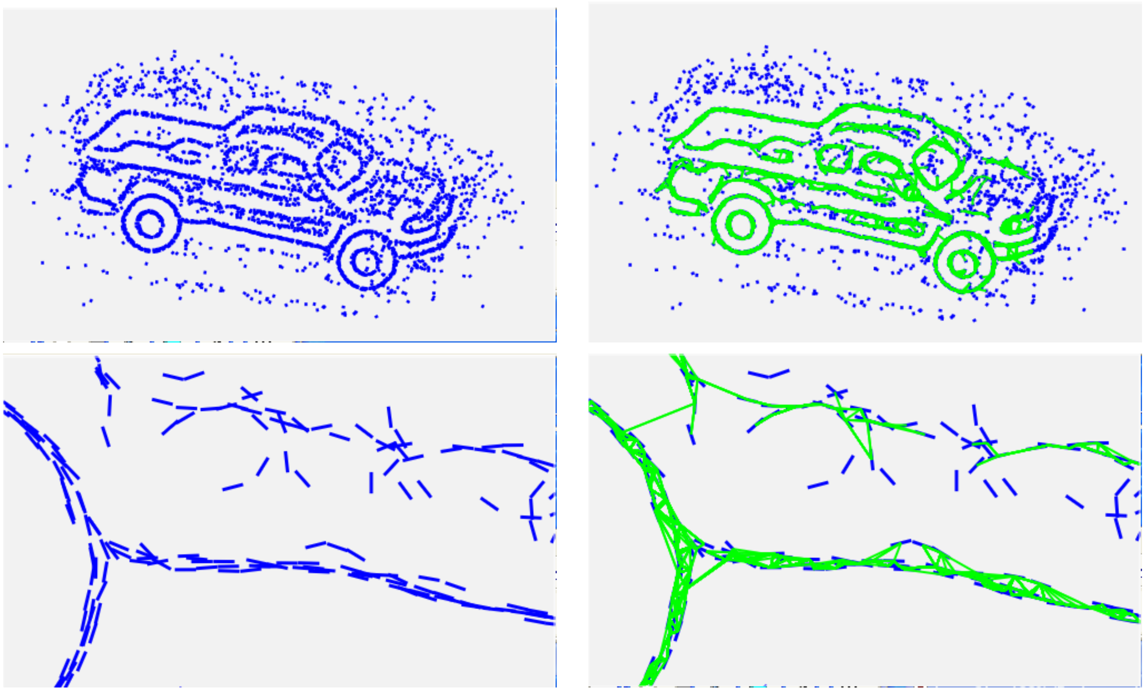


Figure 5.10: The composite edge map top left and curve bundles top right. this is magnified to show how the enriched edge map edges are selected.

to a ground truth edge map. Since it is nearly impossible to manually define a ground truth edge map for a realistic video sequence, we use a controlled rendering approach where an ideal edge map of an object in perfect viewing condition undergoes corruption from illumination changes, inter-reflections from surrounding objects, blending of the object into background and occlusion in some cases. This allows us to control the extent of various factors and the extent by which an ideal edge map, the ground truth, is ”corrupted” by realistic factors.

Specifically, a fairly realistic looking synthetic video is rendered using the 3D rendering software POVRAY [73]. Two video sequences are rendered, *(i)* a vehicle moving along a white background with a simple ambient light model, Figure 5.13(a), and *(ii)* the same vehicle with a complex background with a lighting model allowing for inter-reflections, Figure 5.13(b) . The first video gives us a figure edge-map which is not corrupted by the external factors and is considered to be the ground truth edge map shown in Figure 5.14(b). The edge map of Figure 5.13(b) are shown in Figure 5.14(d). The approach described here is applied against the edge map of the video in Figure 5.13(b) which is magnified in Figure 5.14(d). The enriched edge maps using our multi-frame consistency approach are shown in Figure 5.14(f). These are then compared against the ”ground truth” edge map which is the set of edge maps in Figure 5.14(b). The ROC comparing the two sets is shown in Figure 5.15. Figure 5.16 shows the magnified comparison between single frame

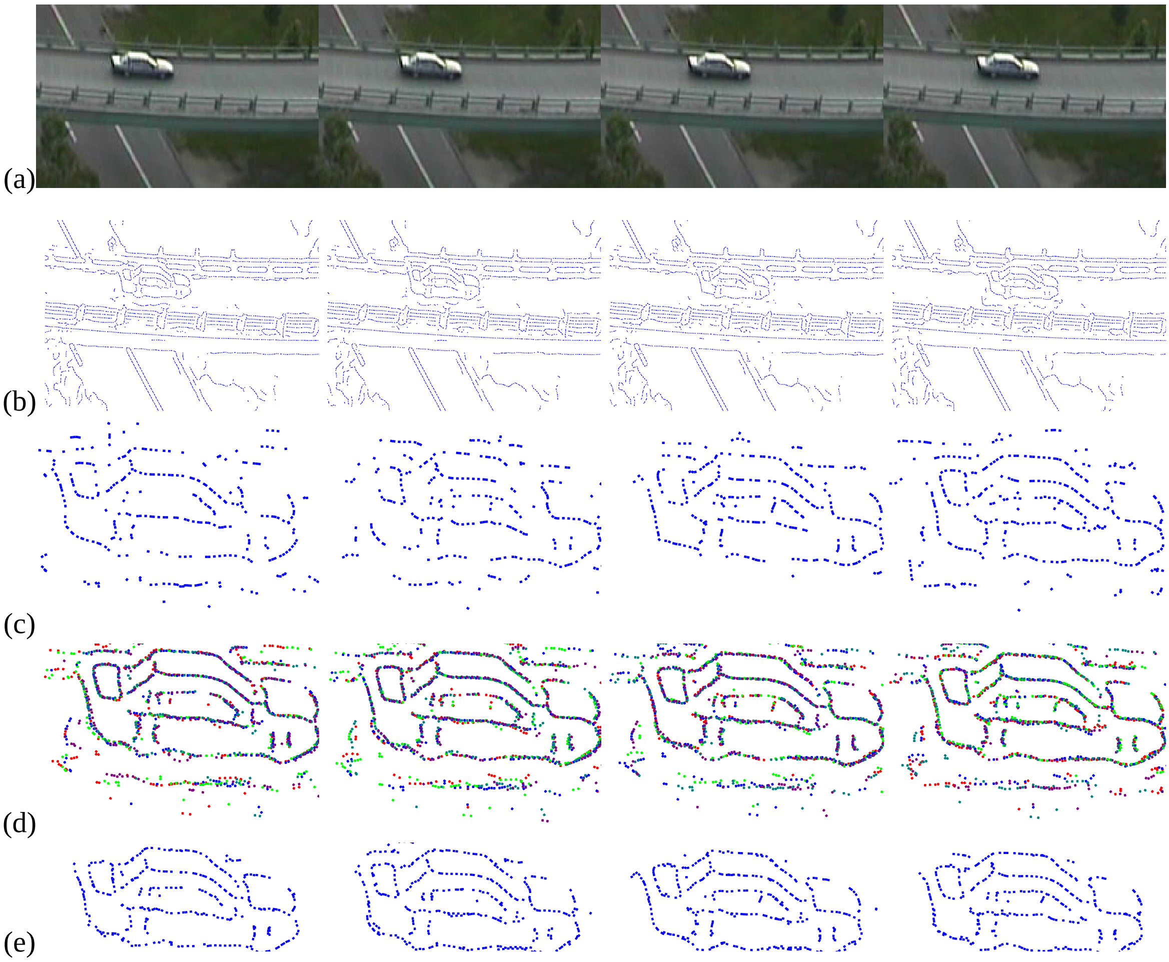


Figure 5.11: Results of our approach on a video sequence. (a) four frames from a video sequence and, (b) corresponding subpixel edge-map of (a) using [91], (c) Foreground edge-map determined by [**?**], (d) the composite edge map is formed by superimposing edge-maps of 5 frames for each frame, (e) the enriched edge map is the set of consistent edges which are retained. Observe the significant difference between the original edge maps in (c) and the enriched edge maps in (e).

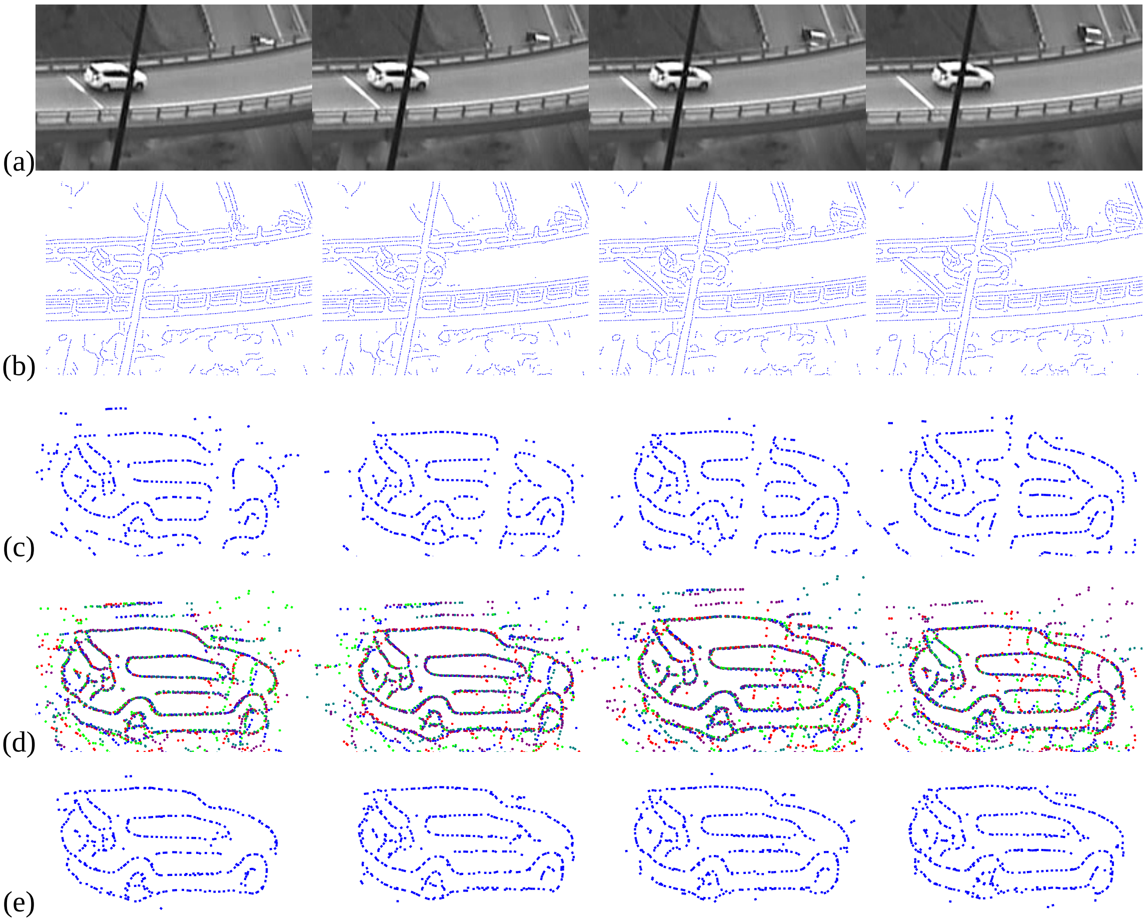


Figure 5.12: Results of our approach on a video sequence. (a) four frames from a video sequence and, (b) corresponding subpixel edge-map of (a) using [91], (c) Foreground edge-map determined by [**?**], (d) the composite edge map is formed by superimposing edge-maps of 5 frames for each frame, (e) the enriched edge map is the set of consistent edges which are retained. Observe the significant difference between the original edge maps in (c) and the enriched edge maps in (e). Observe that when the vehicle undergoes an occlusion an individual frame is severely affected, in contrast to the enriched edge map which is stable and retains the occluded structure.

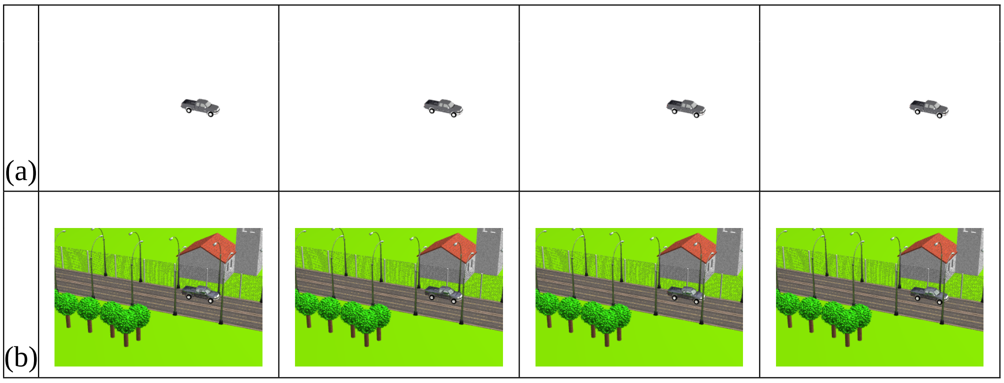


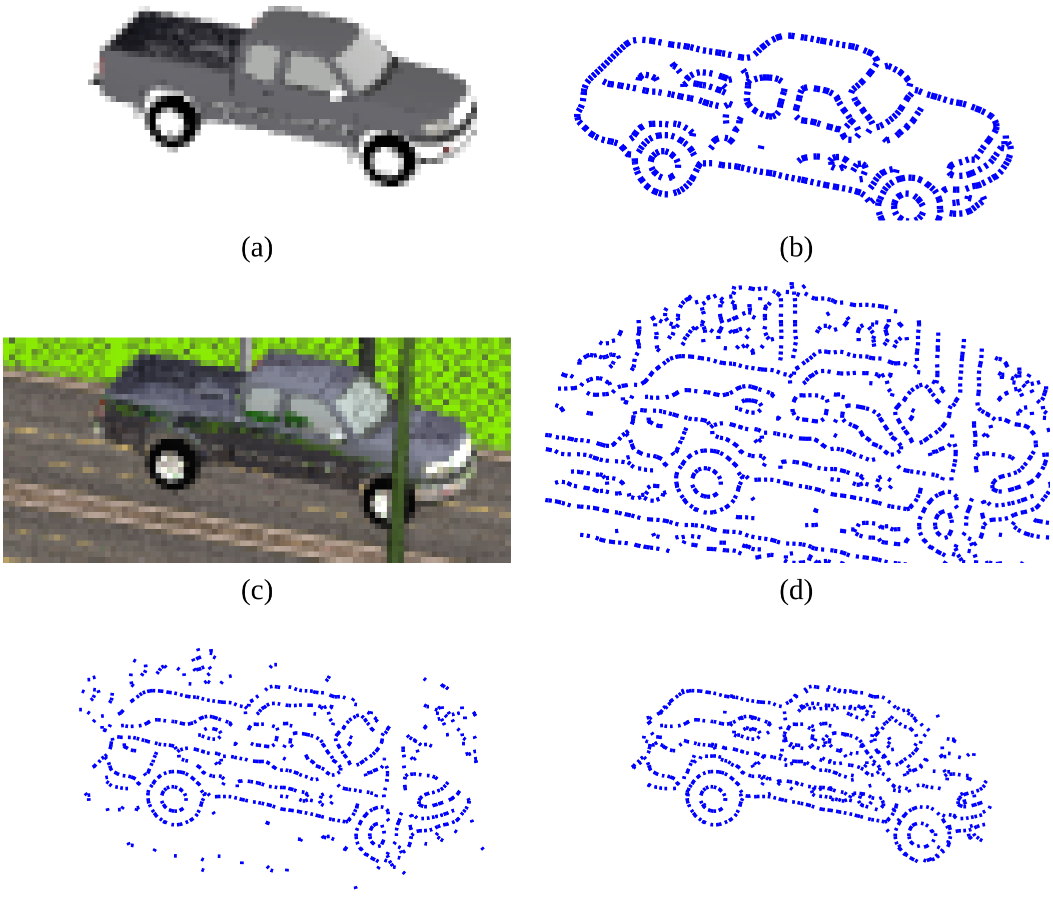
Figure 5.13: This figure shows our synthetic video rendered using a 3D rendering software [73]. (a) shows a vehicle moving against a simple background with ambient light source. (b) The vehicle moving in a scene with a more complex background, with a more complex light source model. The scene contains multiple objects, fences, posts, *etc.* to make it more realistic and model some of the factors like blending of objects into background, inter-reflections, occlusion, *etc.* which are responsible for degradation of the figure edge-maps. Figure 5.1 shows in greater detail the degradation of an ideal edge map under these conditions.

segmentations and the enriched segmentations using our approach.

In order to plot the performance, the threshold to detect the figure edge-maps for background modeling approach [47] was varied. A lower threshold would give many false positives and true positives and as the threshold increases both of these quantities decrease. The comparison between two edge-maps is evaluated over such a variation. The false positives and true positives for both set of edge-maps were plotted in a ROC curve, Figure 5.15. Note that the enriched edge-maps have outperformed the original single frame edge maps, as consistent with the qualitative impressions.

5.5 Conclusion

We have developed an approach for constructing an enriched edge map by integrating the edge maps of several adjacent frames. The methodology brings these frames into register and then uses a notion of geometric consistency to discard spurious edges. The resulting enriched edge map has demonstrated clear qualitative and quantitative advantages over single frame edge maps, which is expected to present a significant advantage for object recognition, object tracking and other higher-level tasks. We can identify two drawbacks of our approach are (i) the method breaks down in low frame rate video if the object undergoes significant change of viewpoint in the adjacent frames, violating an assumption clearly stated earlier (ii) the algorithm is computationally expensive as it takes 30 sec on an Intel Xeon 3.2GHz processor to process one object (approx. 700 edges) with the standard TPS implementation.



(a) (b)

(c) (d)

(e) (f)

Figure 5.14: This figure shows the sequence used for the experiment out on synthetic video to evaluate our approach quantitatively. (a) image and (b) its edge map of the object in simple setting, (c) cropped image and (d) its edge map of the object in more realistic conditions, (c) Foreground edge-detection using [**?**], (d) Enriched edge maps corresponding to edge-map (c).

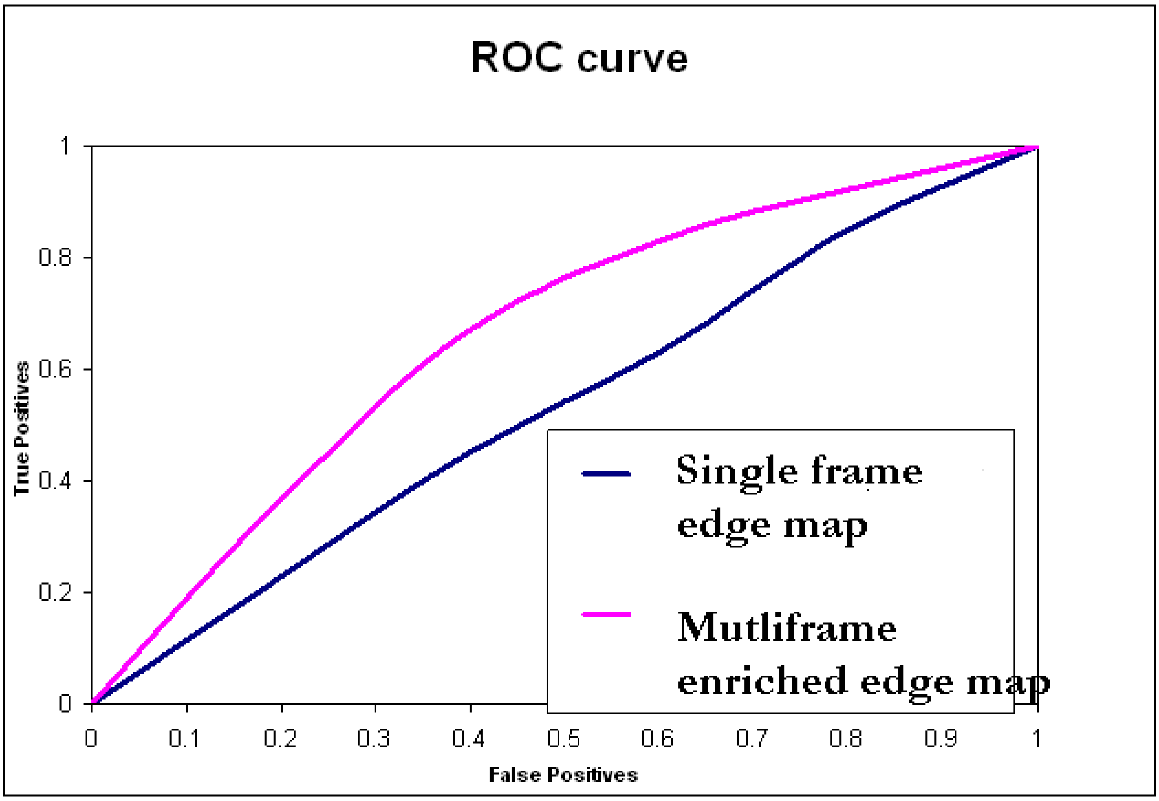


Figure 5.15: ROC curve for comparing figure edge-maps after using multi-frame consistency (shown in Pink) with raw figure-edge maps (shown in blue).

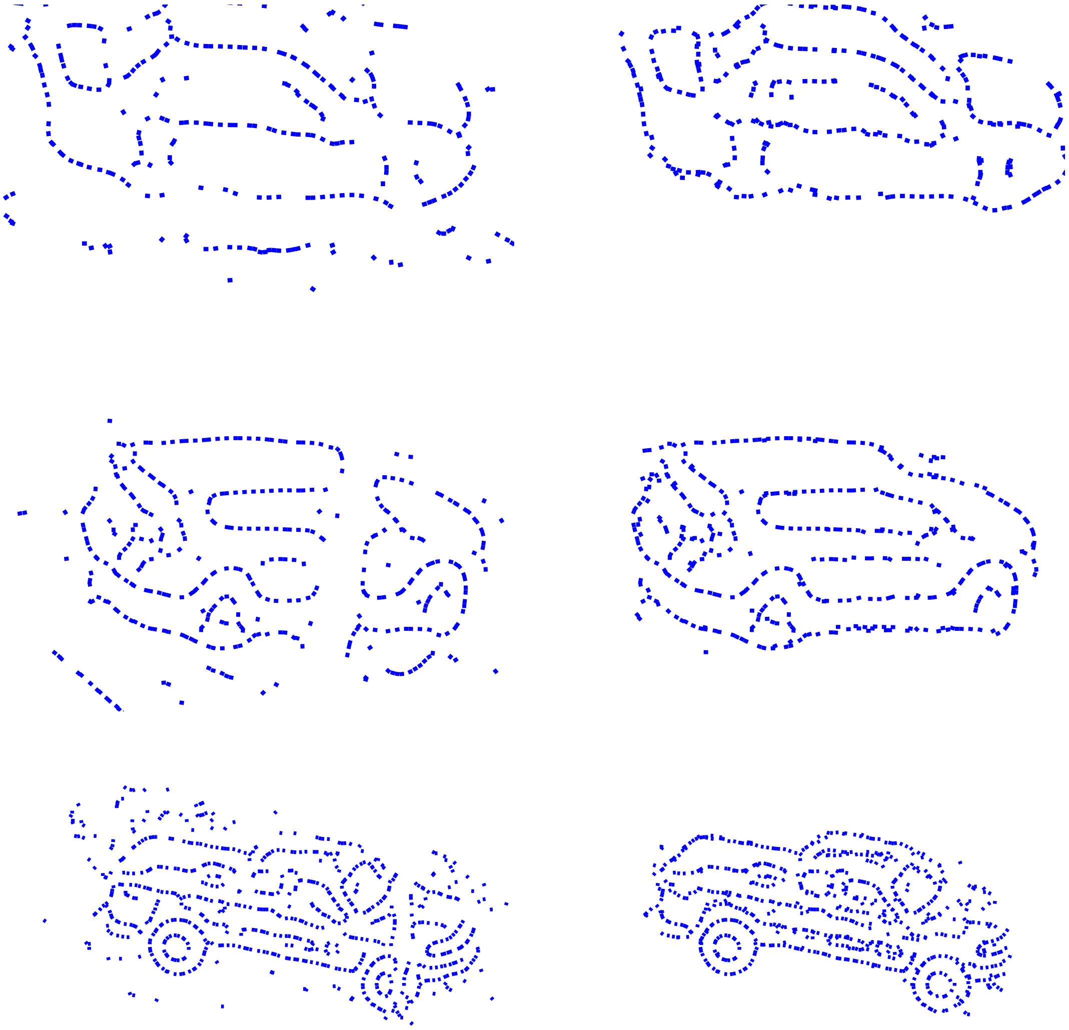


Figure 5.16: This figure shows the zoom in on the results from Figures 5.11, 5.12 and 5.14. The left column shows the single frame foreground detections and the right column shows the multi-frame composite map.

Chapter 6

3D Motion Estimation

6.1 Introduction

A fundamental limitation of motion segmentation approaches which rely on grouping based on *common image motion* is that the object with a significant depth variation project to a range of image motions and can therefore be grouped into multiple segments, some of which may as a result prefer to image with background regions. This is especially true when the elongation is along the optical axis and when the motion has a non-negligible component in depth. This limitation holds regardless of whether dense flow, isolated features, or edges/curves are initially used to establish correspondence as Figure 1.1 illustrates. It reflects a violation of an implicit assumptions; namely, that "objects with similar motion have similar image motions". Rather, common 3D motion **can** lead to a distinct 2D motion either due to a non-negligible component in depth of the object or due to significant component of velocity along the optical axis or both. This motivates an examination of the extent aspects of 3D motion, can be directly estimated and used for motion segmentation. Among the three categories of motion-based segmentations, namely, those based on *(i)* dense flow, *(ii)* features, and *(iii)* curves/edges, the latter give the richest, most representative 2D motion segmentations. Each of the classes of approaches has a fundamental drawback. First, features such as KLT/SIFT are typically sparse and insufficient to estimate motion models (and hence the segmentation) unless the object is rich in texture. This is especially an issue for man-made artifacts, *e.g.*, the office environment, and for low resolution images, *e.g.*, aerial images, Figure 1.1(c-d). Second, the pixel-wise dense computation of flow is ambiguous/erroneous mainly because techniques using brightness constancy have a very low signal to noise ration at low-gradient regions images which comprises a significant portion of the image, *i.e.*, the pixels away from edges. The pixels near or on the edge have higher signal to noise ratio and provide a better estimation of flow. The main difficulty is these approaches is that it is not clear when flow estimation is reliable, Figure 1.4. In contrast the curve/edge-based representation avoid these fundamental limitations. Figures 1.1, 6.1 and 6.2 clerly shows that the curve/edge-based 2D motion segmentation are more selective and more representative of the object structure. Table 6.1 compares the three representations in more detail. We now argue that the curve/edge based representations is also more suitable for 3D motion segmentation while compared to the feature-based and dense-flow based representations. First, observe that the although the correspondence in a feature-based method is typically reliable mainly due to *isolated* features, grouping based on 3D motion would be unreliable precisely because assuming common 3D motion for sparse, isolated features is risky. Even when there are a sufficient number of features, only few remain satisfying the criterion. feature-based methods are therefore not our candidate of choice as representation for 3D motion segmentation.

Second, while in contrast to the sparse feature-based estimation dense flow estimation provides a dense set of correspondences over which the assumption of common 3D motion can be safely made, the correspondence itself is not as reliable as feature-based correspondence due to the low-SNR at low gradient regions of the image. Optical flow methods overcome this by assuming a smooth motion field planar/quadric models. The lack of precise localization (as present in an edge/curve) present two dimensions of ambiguity.

	Features	Pixels	Curves
Correspondence	unambiguous	ambiguous	ambiguous along the curve
Computational Complexity	Low	High	Medium
Illumination changes	moderately invariant	variant	moderately invariant
Segmentation results	Sparse 2D cloud of points	connected set of pixels (region)	Collection of curve fragments.
Nature of Objects	Objects rich in texture	should not be homogeneous	have boundaries and reflectance edges

Table 6.1: This table compares three types of representations, features, pixels, and curves in terms of properties such as density, correspondence estimation, robustness to illumination changes and the delineation of the object boundaries.

The use of curves/edges as a representation for 3D motion estimation represents a middle ground between feature-based and dense-flow based representation, as Table 6.1 suggests. Specifically, consider a moving 3D curve, an occluding contour, reflectance discontinuity, a sharp ridge, *etc.* represented by a $\mathbf{\Gamma}(s,t)$ projecting to $\boldsymbol{\gamma}(s,t)$ in a video sequence. In this paper we examine the relationship between the 3D motion of $\mathbf{\Gamma}(s,t)$ and the observations capturee by $\boldsymbol{\gamma}(s,t)$ as moderated

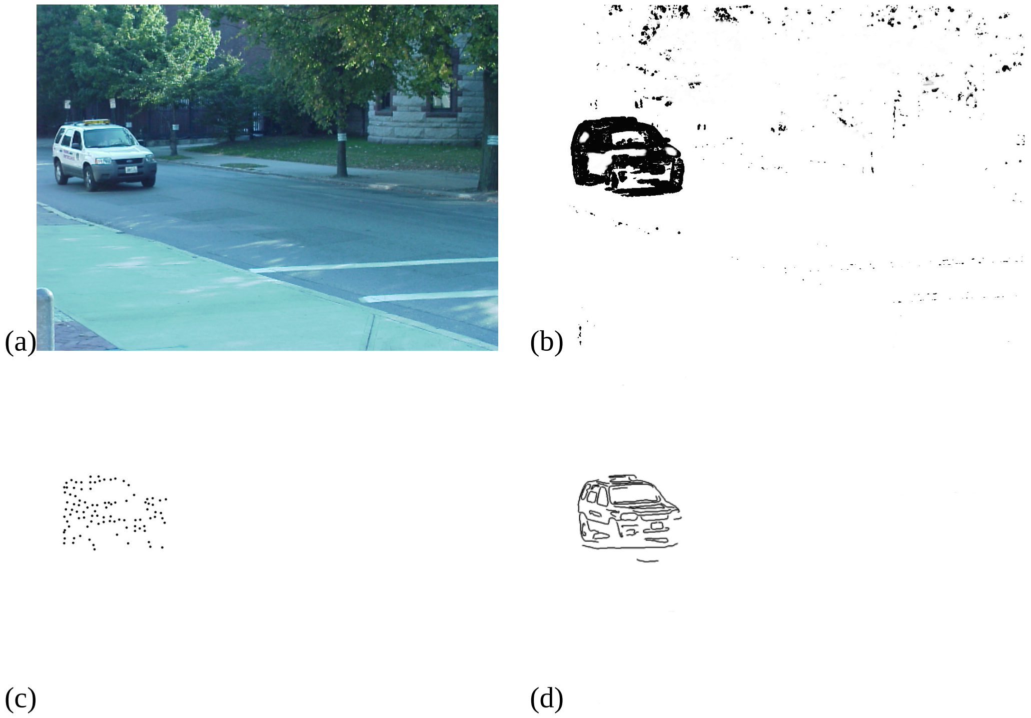


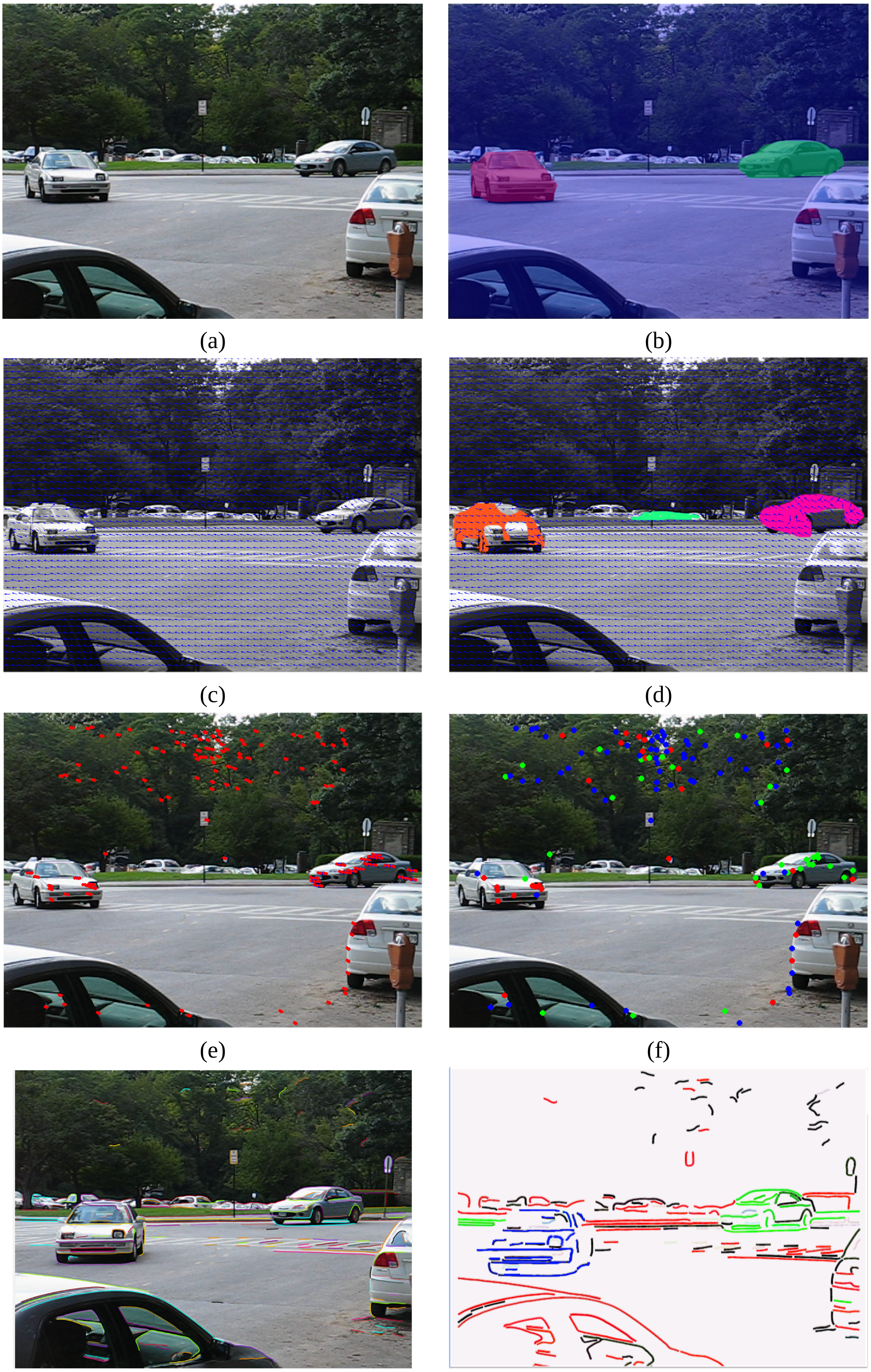
Figure 6.1: The feature-based segmentation of a moving vehicle from a video of sequence of frames one of which is shown in (a) gives a sparse representation in (c) in contrast to curve-based segmentation using the methods of this thesis (b). On the other hand the pixel-based approach gives a dense segmentation than both (b) and (c) but misses a lot of regions due to ambiguous flow in regions of uniform intensity.

by

$$\mathbf{\Gamma}(s,t) = \rho(s,t)\boldsymbol{\gamma}(s,t)$$

where s is the parameterization along the curve and t is the time index and ρ is the depth, Figure 1.7(a). The one-parameter family of curves $\boldsymbol{\gamma}(s,t)$ can be examined in a single central frame, Figure 1.7(b). For example, consider the moving truck Figure 1.8 (a) whose edge maps are superimposed to give rise to this one-parameter family of curves, where each color denotes a separate time sample, Figure 1.8(b) and a zoomed area in Figure 1.8(c).

In this approach, we focus on a camera moving with respect to an object with rotation $R(t)$ and translation $T(t)$ such that in a few local frames. this can be approximated using $\Omega(t) = \frac{dR(t)}{dt}$ and $V(t) = \frac{dT(t)}{dt}$. The shape of the 3D curve $\mathbf{\Gamma}(s,0)$ can also be locally described using a point $\mathbf{\Gamma}_0$, tangent $\vec{T}$, normal $\vec{N}$, speed of parameterization G, and curvature K. Clearly, the desirable unknown are the 3D shape of the curve $\{\mathbf{\Gamma}_0, \vec{T}, \vec{N}, G, K\}$ and the 3D motion of the curve $\{\Omega, \boldsymbol{V}\}$.



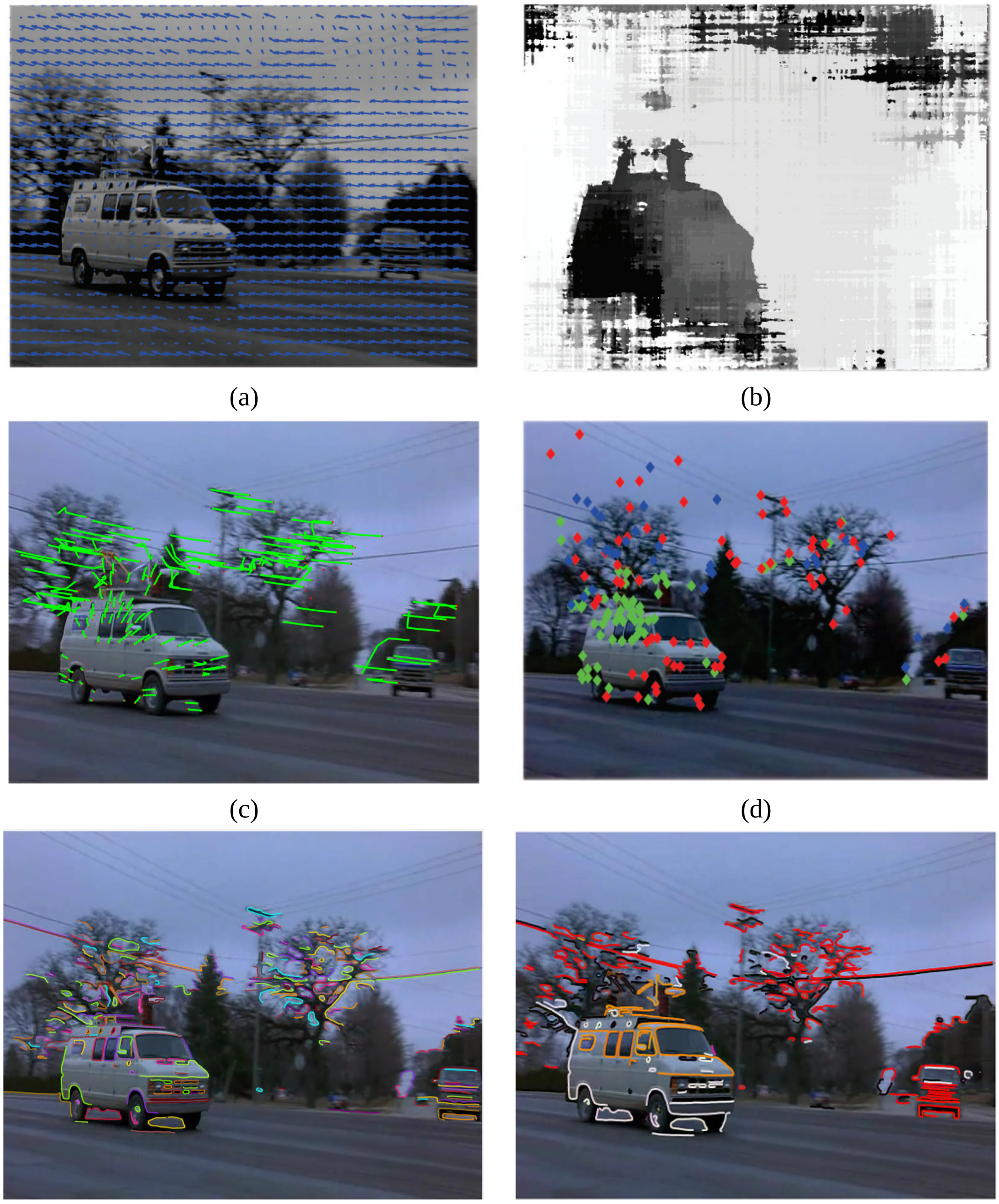
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 6.2: The goal of motion segemtnation is the delineation of moving objects relative to a background from the sequences of frames in a video. A frame of a video is shown in (a) and the ideal results are depicted in (b). (c) dense optical flow estimation and (d) segmentation into component based on fitting a paramteric 2D motion model, shown in pink, green and orange. (e) KLT features and (f) factorization based segmentation shown in red, green and blue.(g) curves superimposed on the image and (h) segmentation based on affine motion model where distinct

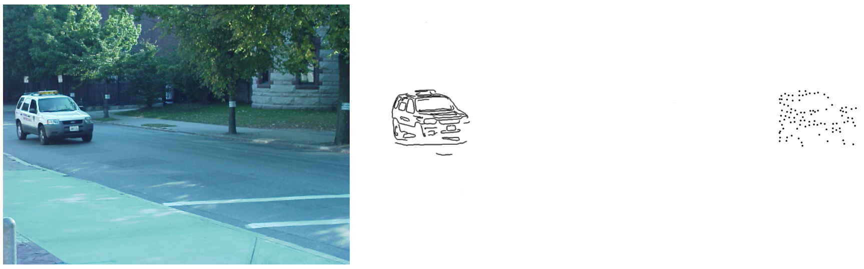


(a) (b)

(c) (d)

(e) (f)

Figure 6.3: (a) A frame from the movie “Groundhog Day” with optical flow superimposed on it. (b) the frame can then be segmented based on 2D parametric motion models and the resulting segmentation. This clearly shows the van is divided into two pieces. Similarly the tracked KLT features of the same frame are shown in (c) and a factorization-based segmentation of these features is shown in (d), where each color represents a different group. Clearly, the van’s features are grouped in the different segments, one of which shares features from background. Finally, the curves on the van (e) are tracked and segmented based on the similarity of 2D motion using an affine model. The resulting segmentation in (f) clearly shows that the van is segmented into two regions.



(a) (b) (c)

Figure 6.4: (b) Curve-based segmentation of moving objects of original sequence in (a) and compare it with sparser set of KLT features segmented.

This work aims at estimating 3D translation from the local groupings of edges called "one-parameter family of curves" and analyzes the lower bound of the uncertainty estimation of translation direction as a function of the uncertainty of measurements. This uncertainty is plotted for system values of a typical system and shown the errors are too high to estimate 3D motion for segmentation of objects. First, 3D motion and geometry estimation of curves from in terms of the second-order derivatives of the one-parameter family of curves, $\gamma(s,t)$, which has also been derived by Faugeras[31]. $\gamma(s,t)$ is assumed to observed for a rigidly translating fixed curve relative to the camera which suffices for the scope of this work The magnitude of the translation is not constrained due to unknown depth. The 3D translation direction is estimated as one-parameter family at each edge, γ_0, and its local grouping of edges $\gamma(s,t)$. At-least two such edges are needed to estimate translation direction. And definitely each object has many more edges available. The lower bound for error in translation direction is estimated as a function of error in measurements. This lower bound was plotted for some typical values of the system. The plots show high amount of errors rendering unreliable estimation of translation direction and hence, unreliable segmentation. The main contribution of this work is to show quantitatively the instability of computation of 3D motion form a video sequence. Before this work is discussed in detailed, existing attempts to recover 3D motions and their qualitative observations which match are quantitative results are discussed.

6.2 3D motion based approaches

The problems associated with computation of 3D motion parameters are two-folds, *(i)* Ambiguities in computing 3D motion due to multiple moving objects [5], and *(ii)* fundamental bias in computation of the translation direction [52, 62, 27, 99, 32, 50, 59, 79].

The work by Weng *et al.* [99] uses motion between two frames computed using a stereo matching algorithm [100] to estimate the 3D motion parameters which is rotation and translation, R and

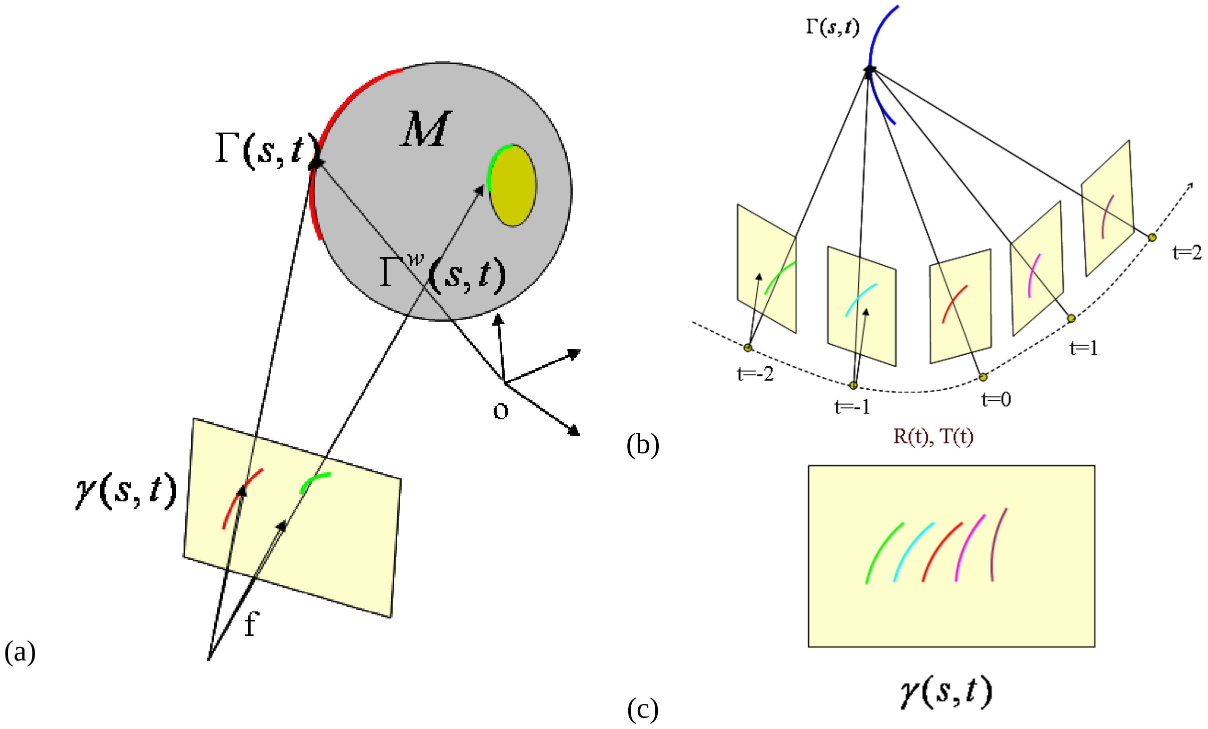


Figure 6.5: This figure shows the 2D curves obtained from a moving 3D curve or moving camera. (a) $\mathbf{\Gamma}(s,t)$ is a 3D stationary curve shown in green and non-stationary curve shown in red and its projection $\boldsymbol{\gamma}(s,t)$, (b) projections of a 3D curve onto different frames which is shown in (c).

T respectively . First, the essential matrix is computed from the correspondences and then the essential matrix is decomposed into R and T. Second, the authors showed large errors in estimation in direction of T for small inter-frame motion. Also, qualitative observations for which the the estimations are are reasonable were reported, which are as follows: *(i)* the scene should be closer to the camera and yield large displacement and *(ii)* translation orthogonal to the image plane allows stable computation. Liu *et al.*[59], overcomes the problem of uncertainty in motion parameters by assuming object of relatively constant depth which is only valid for applications like face and gesture recognition unlike our framework of general segmentation of independently moving objects. Adiv [5] shows the ambiguities of motion field arising from two independently objects can be projected by a single object. The author also shows the SNR is high for the estimation of motion parameters as well as 3D structure for a planar surface.

The survey in [27] summarizes, analyzes and compares three approaches [52, 62, 50] proposed to overcome the bias in the estimation of translation direction. The above three approaches derives a linear constraint $\gamma_t \times \gamma.\boldsymbol{V} = 0$. Then $\tau^i = \gamma_t^i \times \gamma^i$ at different positions i gives us direction of $\boldsymbol{V}$. The direction of $\boldsymbol{V}$ corresponds to the minimum eigenvalue eigenvector of D where D is given

Projection Model	3D curve $\mathbf{\Gamma}(s,0)$ 3D motion $R(t), T(t)$	$\boldsymbol{\gamma}$(s,t) $\frac{d\gamma(s,t)}{dt} = \alpha \boldsymbol{t} + \beta \boldsymbol{n}$ One-parameter family of curves	Depth $\rho(s,t)$
Differential Geometry	3D curve $\mathbf{\Gamma}(s,0)$ $\mathbf{\Gamma}_0, \vec{T}, \vec{N}, G, K$ 3D motion $R(t), T(t)$ $\Omega = \frac{dR(t)}{dt}, \boldsymbol{V} = \frac{dT(t)}{dt}$	point $\boldsymbol{\gamma}_0$ tangent $\boldsymbol{t}, \boldsymbol{n}$ flow α, β $\kappa, \alpha_s \beta_s, \alpha_t, \beta_t$	ρ_0 ρ_s, ρ_t $\rho_{ss}, \rho_{st}, \rho_{tt}$
Observable	None	Theorem: 2D shape $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \kappa$ Normal flow β, β_s, β_t	None
Unknowns	shape $\mathbf{\Gamma}_0, \vec{T}, \vec{N}, G, K$ motion $\Omega, \boldsymbol{V}$	tangential flow $\alpha, \alpha_s, \alpha_t$	depth ρ_0 gradient ρ_s, ρ_t Hessian $\rho_{ss}, \rho_{st}, \rho_{tt}$

Table 6.2: This table summarizes the relationship between 2D and 3D motion and shape parameters and clearly demarcates between shape and motion as well as unknowns and observable.

by

$$D \equiv \sum_{i=1}^{n} \tau^i \tau^{i\prime}.$$

In the case of noisy measurements of $\boldsymbol{\gamma}_t$ and $\boldsymbol{\gamma}$, $\tilde{D}$ is given by

$$\tilde{D} \equiv \sum_{i=1}^{n} (\tau^i + \delta\tau^i)(\tau^i + \delta\tau^i)'.$$

[50] shows the covariance of $\tilde{D}$ is flat along $\boldsymbol{\gamma}$ than along the other directions and has explained this as the reason for bias in the estimation of translation direction.

First, Jepson *et al.*[50] proposes addition of extra noise along $\boldsymbol{\gamma}$ direction. This requires the knowledge of noise variance and would minimize the bias but also allow for more error especially when the motion is along or near the $\boldsymbol{\gamma}$ direction. Second, Kanatani [52] computes the statistical bias by assuming isotropic noise in the flow vectors $\boldsymbol{\gamma}_t$ and subtracts it from covariance of $\tilde{D}$ which removes the bias completely. But this requires estimation of noise at each point accurately which is not practical. Third, MacLean *et al.* [62] overcomes the problem of Kanatani by transforming the $\tilde{D}$ to different space where it is independent of noise variance and then transform its eigenvector back to original space. But the transformation back requires a normalization process which introduces another bias. The results compiled in [27] on a synthetic data shows the effectiveness of the above approaches. The direction of motion considered had a dominant component along z and the results on different motion directions have not been shown. The bias is maximum for motions parallel to the image plane which is shown in the current work.

6.3 Differential 3D structure and motion of a fixed curve from observed differential trace.

This sections shows how to obtain 3D motion and geometry can be obtained from the set of observations of one-parameter family of curves. One important thing to note is that this work considers case of stationary or fixed curves as they suffice for the scope of this work. A second-order local model of a 3D fixed curve $\mathbf{\Gamma}(s,t)$, where s is the parameterization of the curve and t is the different instances of the 3D curve due to motion, is given by

$$\mathbf{\Gamma}(s,t) = \mathbf{\Gamma}(0,0) + \mathbf{\Gamma}_s(0,0)s + \mathbf{\Gamma}_t(0,0)t + \frac{1}{2}\mathbf{\Gamma}_{ss}(0,0)s^2 + \mathbf{\Gamma}_{st}(0,0)st + \frac{1}{2}\mathbf{\Gamma}_{tt}(0,0)t^2. \quad (6.1)$$

This implies if the quantities $\mathbf{\Gamma}(0,0)$, $\mathbf{\Gamma}_s(0,0)$, $\mathbf{\Gamma}_t(0,0)$, $\mathbf{\Gamma}_{ss}(0,0)$, $\mathbf{\Gamma}_{st}(0,0)$ and $\mathbf{\Gamma}_{tt}(0,0)$ are known then the 3D structure and motion of the curve can be recovered. Using the notation, $\mathbf{\Gamma}_s = G\vec{T}$ and $\mathbf{\Gamma}_{ss} = GK\vec{N}$, where $\vec{T}$ and $\vec{N}$ are tangent and normal to the curve in 3D and G is the speed of parameterization. These exhibit the local geometry of the curve. The terms $\mathbf{\Gamma}_t, \mathbf{\Gamma}_{st}, \mathbf{\Gamma}_{tt}$ show variation of $\mathbf{\Gamma}$ due to motion. But only the projection of this 3D curve in multiple frames, $\gamma(s,t)$ is observable. The projection equation of 3D curve is given by

$$\mathbf{\Gamma}(s,t) = \rho(s,t)\gamma(s,t)$$

Therefore all the local derivatives of $\mathbf{\Gamma}(s,t)$ can be expressed in terms of local derivatives of $\rho(s,t)$ and $\gamma(s,t)$ as shown below. The first order derivatives of $\mathbf{\Gamma}$ are given as

$$\left\{ \begin{array}{lcl} \mathbf{\Gamma}_s & = & \rho_s\gamma + \rho\gamma_s \\ \mathbf{\Gamma}_t & = & \rho_t\gamma + \rho\gamma_t \end{array} \right. , \quad (6.2)$$

and the second order derivatives are given by

$$\left\{ \begin{array}{lcl} \mathbf{\Gamma}_{ss} & = & \rho_{ss}\gamma + 2\rho_s\gamma_s + \rho\gamma_{ss} \\ \mathbf{\Gamma}_{st} & = & \rho_{st}\gamma + \rho_s\gamma_t + \rho_t\gamma_s + \rho\gamma_{st} \\ \mathbf{\Gamma}_{tt} & = & \rho_{tt}\gamma + 2\rho_t\gamma_t + \rho\gamma_{tt} \end{array} \right. . \quad (6.3)$$

Second order Model of 3D curve and motion $\boldsymbol{\Gamma}(s,t)$ is given by $\boldsymbol{\Gamma}_0, \vec{T}, G, \vec{N}, K, \boldsymbol{\Gamma}_t, \boldsymbol{\Gamma}_{st}, \boldsymbol{\Gamma}_{tt}$	=	Second order Model of $\gamma(s,t)$ is given by $\gamma_0, \boldsymbol{t}, \boldsymbol{n}, \beta, \alpha, \kappa, \beta_s, \alpha_s, \beta_t, \alpha_t$	+	Second order Model of $\rho(s,t)$ is given by $\rho_0, \rho_s, \rho_t, \rho_{ss}, \rho_{st}, \rho_{tt}$.

Curve Geometry $\boldsymbol{\Gamma}_0, \vec{T}, G, \vec{N}, K$	+	Motion $\boldsymbol{\Gamma}_t, \boldsymbol{\Gamma}_{st}, \boldsymbol{\Gamma}_{tt}$	=	Geometry	+	Motion

Geometry

Observed	$\gamma_0, \boldsymbol{t}, \boldsymbol{n}, \kappa$
Unobserved	$\rho_0, \rho_s, \rho_{ss}$

Motion

Observed	β, β_t, β_s
Unobserved (2D)	$\alpha, \alpha_t, \alpha_s$
Unobserved (3D)	$\rho_t, \rho_{st}, \rho_{tt}$

Table 6.3: Visual description of segregation of 3D Structure and Motion parameters into 2D observables and 2D and 3D unobservables. The first row shows the unknowns of 3D structure and motion can be split into unknowns of one-parameter family of curves and derivatives of depth ρ.

Now the unknowns $\boldsymbol{\Gamma}(0,0)$, $\boldsymbol{\Gamma}_s(0,0)$, $\boldsymbol{\Gamma}_t(0,0)$,$\boldsymbol{\Gamma}_{ss}(0,0)$, $\boldsymbol{\Gamma}_{st}(0,0)$ and $\boldsymbol{\Gamma}_{tt}(0,0)$ translates into ρ and $\boldsymbol{\gamma}$ and their derivatives. Therefore, $\{\rho, \rho_s, \rho_t, \rho_{ss}, \rho_{st}, \rho_{tt}\}$ and $\{\boldsymbol{\gamma}, \boldsymbol{\gamma}_s, \boldsymbol{\gamma}_t, \boldsymbol{\gamma}_{ss}, \boldsymbol{\gamma}_{tt}, \boldsymbol{\gamma}_{st}\}$ determines a general second-order local model of curve and its motion. The first set of variables is unobserved in the image but few components of the variables in the second set are observable in the images. Proposition 1.3 shows $\{\boldsymbol{\gamma}, \boldsymbol{\gamma}_s, \boldsymbol{\gamma}_t, \boldsymbol{\gamma}_{ss}, \boldsymbol{\gamma}_{tt}, \boldsymbol{\gamma}_{st}\}$ can be expressed in terms of $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \kappa, \alpha, \beta, \alpha_s, \alpha_t, \beta_s, \beta_t$. Out of these $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \kappa, \beta, \beta_s, \beta_t$ are observable and $\alpha, \alpha_s, \alpha_t$ are free. Therefore, $\{\rho, \rho_s, \rho_t, \rho_{ss}, \rho_{st}, \rho_{tt}\}$ and $\{\alpha, \alpha_s, \alpha_t\}$ are free parameters need to determine 3D curve and its motion. In other words, given these nine parameters the 3D curve and its motion can be determined. Equation 6.2 and 6.3 can be rewritten in terms of $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \kappa, \alpha, \beta, \alpha_s, \alpha_t, \beta_s, \beta_t$. The first-order derivatives are given by

$$\begin{cases} \boldsymbol{\Gamma}_s &= \rho_s \boldsymbol{\gamma} + \rho \boldsymbol{t} \\ \boldsymbol{\Gamma}_t &= \rho_t \boldsymbol{\gamma} + \rho(\alpha \boldsymbol{t} + \beta \boldsymbol{n}) \end{cases}, \tag{6.4}$$

and the second order derivatives are given by

$$\begin{cases} \boldsymbol{\Gamma}_{ss} &= \rho_{ss}\boldsymbol{\gamma} + 2\rho_s \boldsymbol{t} + \rho\kappa \boldsymbol{n} \\ \boldsymbol{\Gamma}_{st} &= \rho_{st}\boldsymbol{\gamma} + \rho_s(\alpha \boldsymbol{t} + \beta \boldsymbol{n}) + \rho_t \boldsymbol{t} + \rho(\alpha_s - \beta\kappa)\boldsymbol{t} + \rho(\alpha\kappa + \beta_s)\boldsymbol{n} \\ \boldsymbol{\Gamma}_{tt} &= \rho_{tt}\boldsymbol{\gamma} + 2\rho_t(\alpha \boldsymbol{t} + \beta \boldsymbol{n}) + \rho[(\alpha_t - \beta(\alpha\kappa + \beta_s))\boldsymbol{t} + (\alpha(\alpha\kappa + \beta_s) + \beta_t)\boldsymbol{n}] \end{cases}. \tag{6.5}$$

This relation of 3D unknowns to 2D observable and unknowns are summarized in Table 6.3. This is true under any condition which can be subjected to constraints. In this work, the motion of the curve is assumed to constant translation, $\boldsymbol{V}$. Note that the motion of the object can be approximated as piecewise constant translation for an infinitesimal amount of time. This assumption of constant translation constrains the unknowns as shown in the following proposition.

Proposition 6.1. *Given a measurable quantities* $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \kappa, \beta, \beta_s, \beta_t$ *of a trace upto second order for a fixed 3D curve moving with constant velocity* $\boldsymbol{V}$*,*

$$\boldsymbol{\Gamma}(s,t) = \boldsymbol{\Gamma}(s,0) + \boldsymbol{V}t \tag{6.6}$$

the local structure of the curve $\{\rho_s, \rho_t, \rho_{st}, \rho_{tt}\}$ *and* $\{\alpha_s, \alpha_t\}$ *can be estimated as a function of* ρ *and* α *and the observable.*

Proof. First order derivatives of $\boldsymbol{\Gamma}(s,t)$ under first-order translation constraint of Equation 6.6 are given by

$$\left\{ \begin{array}{rcl} \boldsymbol{\Gamma}_s(s,t) & = & \boldsymbol{\Gamma}_s(s,0) \\ \boldsymbol{\Gamma}_t(s,t) & = & \boldsymbol{V} \end{array} \right. \tag{6.7}$$

and the second-order derivatives are given by

$$\left\{ \begin{array}{rcl} \boldsymbol{\Gamma}_{ss}(s,t) & = & \boldsymbol{\Gamma}_{ss}(s,0) \\ \boldsymbol{\Gamma}_{st}(s,t) & = & 0 \\ \boldsymbol{\Gamma}_{tt}(s,t) & = & 0 \end{array} \right. . \tag{6.8}$$

Using the relations from Equation 6.4 and 6.5 to constrain the quantities $\{\rho, \rho_s, \rho_t, \rho_{ss}, \rho_{st}, \rho_{tt}\}$ and $\{\alpha, \alpha_s, \alpha_t\}$

$$\left\{ \begin{array}{rcl} \boldsymbol{\Gamma}_s(s,0) & = & \rho_s \boldsymbol{\gamma} + \rho \boldsymbol{t} \\ \boldsymbol{V} & = & \rho_t \boldsymbol{\gamma} + \rho(\alpha \boldsymbol{t} + \beta \boldsymbol{n}) \end{array} \right. , \tag{6.9}$$

and the second order derivatives are given by

$$\left\{ \begin{array}{rcl} \boldsymbol{\Gamma}_{ss}(s,0) & = & \rho_{ss}\boldsymbol{\gamma} + 2\rho_s \boldsymbol{t} + \rho\kappa\boldsymbol{n} \\ 0 & = & \rho_{st}\boldsymbol{\gamma} + \rho_s(\alpha\boldsymbol{t} + \beta\boldsymbol{n}) + \rho_t\boldsymbol{t} + \rho(\alpha_s - \beta\kappa)\boldsymbol{t} + \rho(\alpha\kappa + \beta_s)\boldsymbol{n} \\ 0 & = & \rho_{tt}\boldsymbol{\gamma} + 2\rho_t(\alpha\boldsymbol{t} + \beta\boldsymbol{n}) + \rho[(\alpha_t - \beta(\alpha\kappa + \beta_s))\boldsymbol{t} + (\alpha(\alpha\kappa + \beta_s) + \beta_t)\boldsymbol{n}] \end{array} \right. . \tag{6.10}$$

There are a total of 15 equations. Now each of the vector equation of Equation 6.9 and 6.10 can be split into 3 equations. First, the first-order derivatives can be split into three components each. Taking the dot product of $\boldsymbol{\Gamma}_s(s,0) = \rho_s\boldsymbol{\gamma} + \rho\boldsymbol{t}$ with e_3 and $\boldsymbol{t}$ is given by

$$\begin{array}{rcl} e_3 \cdot \boldsymbol{\Gamma}_s & = & \rho_s \\ \boldsymbol{\Gamma}_s \cdot \boldsymbol{t} & = & \rho_s\boldsymbol{\gamma} + \rho \end{array} \tag{6.11}$$

Taking the dot product of $\boldsymbol{\Gamma}_t(s,0) = \rho_t\boldsymbol{\gamma} + \rho(\alpha\boldsymbol{t} + \beta\boldsymbol{n})$ with e_3, $U_1 = \boldsymbol{\gamma} \times \boldsymbol{t}$ and $U_2 = \boldsymbol{\gamma} \times \boldsymbol{n}$ is given by

$$\begin{array}{rcl} \boldsymbol{V} \cdot e_3 & = & \rho_t \\ \boldsymbol{V} \cdot U_1 & = & \rho\beta\boldsymbol{n}.U_1 = \rho\beta \\ \boldsymbol{V} \cdot U_2 & = & \rho\alpha\boldsymbol{t}.U_2 = -\rho\alpha \end{array} \tag{6.12}$$

Now moving on to second-order derivatives and taking the dot products with e_3, $\boldsymbol{t}$ and $\boldsymbol{n}$. First with e_3 gives

$$\left\{\begin{array}{ccc} \boldsymbol{\Gamma}_{ss}(s,0)\cdot e_3 & = & \rho_{ss} \\ 0 & = & \rho_{st} \\ 0 & = & \rho_{tt} \end{array}\right. . \tag{6.13}$$

Second the dot product with $\boldsymbol{t}$ gives and using $\rho_{st}=0$ and $\rho_{tt}=0$,

$$\left\{\begin{array}{ccc} \boldsymbol{\Gamma}_{ss}(s,0)\cdot \boldsymbol{t} & = & \rho_{ss}\boldsymbol{\gamma}\cdot\boldsymbol{t}+2\rho_s \\ 0 & = & \rho_s\alpha+\rho_t+\rho(\alpha_s-\beta\kappa) \\ 0 & = & 2\rho_t\alpha+\rho(\alpha_t-\beta(\alpha\kappa+\beta_s)) \end{array}\right. . \tag{6.14}$$

The above equations provides

$$\left\{\begin{array}{ccc} \alpha_s & = & \frac{1}{\rho}(-\rho_s\alpha-\rho_t)+\beta\kappa \\ \alpha_t & = & -\frac{2\rho_t\alpha}{\rho}+\beta(\alpha\kappa+\beta_s) \end{array}\right. \tag{6.15}$$

Third, the dot product with n gives and using $\rho_{st}=0$ and $\rho_{tt}=0$,

$$\left\{\begin{array}{ccc} \boldsymbol{\Gamma}_{ss}(s,0)\cdot \boldsymbol{n} & = & \rho_{ss}\boldsymbol{\gamma}\cdot\boldsymbol{n}+\rho\kappa \\ 0 & = & \rho_s\beta+\rho(\alpha\kappa+\beta_s) \\ 0 & = & 2\rho_t\beta+\rho(\alpha(\alpha\kappa+\beta_s)+\beta_t) \end{array}\right. . \tag{6.16}$$

The above equations provides

$$\left\{\begin{array}{ccc} \rho_s & = & \frac{-\rho}{\beta}(\alpha\kappa+\beta_s) \\ \rho_t & = & -\frac{\rho}{2\beta}(\alpha(\alpha\kappa+\beta_s)+\beta_t) \end{array}\right. . \tag{6.17}$$

Therefore the set of unknowns can be expressed in terms of two unknowns α and ρ and the observable as follows

$$\begin{array}{ccl} \rho_{st} & = & 0 \\ \rho_{tt} & = & 0 \\ \rho_s & = & -\rho\frac{(\alpha\kappa+\beta_s)}{\beta} \\ \rho_t & = & -\rho\frac{\alpha(\alpha\kappa+\beta_s)+\beta_t}{2\beta} \\ \alpha_s & = & \beta\kappa+\frac{\alpha(\alpha\kappa+\beta_s)+\beta_t}{2\beta}+\frac{(\alpha\kappa+\beta_s)}{\beta}\alpha \\ \alpha_t & = & \beta(\alpha\kappa+\beta_s)+\frac{\alpha(\alpha\kappa+\beta_s)+\beta_t}{\beta} \end{array} \tag{6.18}$$

ρ_{ss} is not constrained and hence $\boldsymbol{\Gamma}_{ss}$ cannot be obtained from this second order model. Also $\boldsymbol{V}$ can be written in terms of unknowns α and ρ as

$$\boldsymbol{V}=-\rho[\frac{\alpha(\alpha\kappa+\beta_s)+\beta_t}{2\beta}\boldsymbol{\gamma}+\boldsymbol{\gamma}_t] \tag{6.19}$$

■

The above proposition shows that 3D structure and motion, except ρ_{ss} or $\boldsymbol{\gamma}_{ss}$, can be recovered if α and ρ are known at every point. Since it is impossible to obtain such information at every individual point, a global variable $\boldsymbol{V}$ is estimated as it is the same for all the points of an object. Therefore $\boldsymbol{V}$ from the above proposition has three constraints in α and ρ given as

$$\begin{cases} \boldsymbol{V}.e_3 & = & -\rho\frac{\alpha(\alpha\kappa+\beta_s)+\beta_t}{2\beta} \\ \boldsymbol{V}.U_1 & = & \rho\beta \\ \boldsymbol{V}.U_2 & = & -\rho\alpha \end{cases} \tag{6.20}$$

Since $\boldsymbol{V}$ is a 3-vector, there are 5 unknowns and 3 constraints. By eliminating ρ, the constraints are further reduced to

$$\begin{cases} \boldsymbol{V}.e_3 & = & -\frac{\boldsymbol{V}.U_1}{\beta}\frac{\alpha(\alpha\kappa+\beta_s)+\beta_t}{2\beta} \\ \frac{\boldsymbol{V}.U_1}{\boldsymbol{V}.U_2} & = & -\frac{\beta}{\alpha} \end{cases} \tag{6.21}$$

One important thing to note is that by eliminating ρ, the magnitude of $\boldsymbol{V}$ is also eliminated. This is because only depth could determine how fast or slow an object is moving, *e.g.*, a slowly moving object far-away and fast moving object near by can give rise to same projection on the image. This leads to 3 unknowns, 2 for direction of $\boldsymbol{V}$ denoted by $\hat{\boldsymbol{V}}$ and one for α. Therefore $\hat{\boldsymbol{V}}$ can be expressed as one-parameter family of solutions which is derived in the following proposition.

Proposition 6.2. *The direction of first order translation* $\hat{\boldsymbol{V}} = (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta)$ *where* $-\pi/2 \le \theta \le \pi/2$ *and* $-\pi \le \phi \le \pi$ *of relative motion of the camera is constrained by spatial and temporal derivatives ,* $\boldsymbol{t}, \boldsymbol{n}, \beta, \beta_s, \beta_t, \kappa$ *and* $\boldsymbol{\gamma} = [\xi, \eta, 1]$

$$\begin{cases} \tan\theta & = & \frac{r}{\sqrt{[2\beta(\alpha t_y+\beta t_x)-r\eta]^2+[2\beta(\alpha t_x-\beta t_y)-r\xi]^2}} \\ (\sin\phi, \cos\phi) & = & \frac{(2\beta(\alpha t_y+\beta t_x)-r\eta, 2\beta(\alpha t_x-\beta t_y)-r\xi)}{\sqrt{[(\alpha t_y+\beta t_x)2\beta-r\eta]^2+[2\beta(\alpha t_x-\beta t_y)-r\xi]^2}} \end{cases} \tag{6.22}$$

where $r = \alpha\beta_s + \beta_t + \alpha^2\kappa$ *and* α *is a free parameter.*

Proof. U_1 and U_2 can be computed as

$$\begin{aligned} U_1 &= \boldsymbol{\gamma} \times \boldsymbol{t} = [-t_y, t_x, \xi t_y - \eta t_x] \\ U_2 &= \boldsymbol{\gamma} \times \boldsymbol{n} = [-t_x, -t_y, \xi t_x + \eta t_y] \end{aligned} \tag{6.23}$$

Now rewriting $\alpha = -\frac{V.U_2}{V.U_1}\beta$ as $\alpha\hat{\boldsymbol{V}}.U_1 + \beta\hat{\boldsymbol{V}}.U_2 = 0$, it can be expanded as

$$-\alpha t_y\hat{\boldsymbol{V}}_x + \alpha t_x\hat{\boldsymbol{V}}_y + \alpha(\xi t_y - \eta t_x)\hat{\boldsymbol{V}}_z - \beta t_x\hat{\boldsymbol{V}}_x - \beta t_y\hat{\boldsymbol{V}}_y + \beta(\xi t_x + \eta t_y)\hat{\boldsymbol{V}}_z = 0.$$

Collecting coefficients of $\hat{\boldsymbol{V}}_x, \hat{\boldsymbol{V}}_y$ and $\hat{\boldsymbol{V}}_z$,

$$(-\alpha t_y - \beta t_x)\hat{\boldsymbol{V}}_x + (\alpha t_x - \beta t_y)\hat{\boldsymbol{V}}_y + [\alpha(\xi t_y - \eta t_x) + \beta(\xi t_x + \eta t_y)]\hat{\boldsymbol{V}}_z = 0. \tag{6.24}$$

Equation **??** is given by

$$2\hat{\boldsymbol{V}}_z\beta^2 + (-t_y\hat{\boldsymbol{V}}_x + t_x\hat{\boldsymbol{V}}_y + (\xi t_y - \eta t_x)\hat{\boldsymbol{V}}_z)[\alpha\beta_s + \beta_t + \alpha^2\kappa] = 0, \tag{6.25}$$

with $r = \alpha\beta_s + \beta_t + \alpha^2\kappa$,

$$-t_y\hat{\boldsymbol{V}}_x + t_x\hat{\boldsymbol{V}}_y + (\frac{2\beta^2}{r} + (\xi t_y - \eta t_x))\hat{\boldsymbol{V}}_z = 0 \tag{6.26}$$

The two constraints are given by

$$\begin{cases} (-\alpha t_y - \beta t_x)\hat{\boldsymbol{V}}_x + (\alpha t_x - \beta t_y)\hat{\boldsymbol{V}}_y + [\alpha(\xi t_y - \eta t_x) + \beta(\xi t_x + \eta t_y)]\hat{\boldsymbol{V}}_z = 0. & = 0 \\ -t_y\hat{\boldsymbol{V}}_x + t_x\hat{\boldsymbol{V}}_y + (\frac{2\beta^2}{r} + (\xi t_y - \eta t_x))\hat{\boldsymbol{V}}_z & = 0 \end{cases} \tag{6.27}$$

Let $l = \frac{2\beta^2}{r}$, eliminating V_z we get

$$\frac{[\alpha(\xi t_y - \eta t_x) + \beta(\xi t_x + \eta t_y)]}{l + (\xi t_y - \eta t_x)} = \frac{(-\alpha t_y - \beta t_x)V_x + (wt_x - \beta t_y)V_y}{-t_yV_x + t_xV_y} \tag{6.28}$$

which is expanded as

$$\begin{aligned} -[\alpha(\xi t_y - \eta t_x) + \beta(\xi t_x + \eta t_y)]t_yV_x + [\alpha(\xi t_y - \eta t_x) + \beta(\xi t_x + \eta t_y)]t_xV_y = \\ (-\alpha t_y - \beta t_x)(l + (\xi t_y - \eta t_x))V_x + (\alpha t_x - \beta t_y)(l + (\xi t_y - \eta t_x))V_y \end{aligned}$$

The above equation is simplified to

$$[-\eta + (\alpha t_y + \beta t_x)\frac{l}{\beta}]V_x = [(\alpha t_x - \beta t_y)\frac{l}{\beta} - \xi]V_y \tag{6.29}$$

Since $\hat{\boldsymbol{V}}_x = \cos\theta\cos\phi$ and $\hat{\boldsymbol{V}}_y = \cos\theta\sin\phi$ and $l = 2\beta^2/r$,

$$\tan\phi = \frac{2\beta(\alpha t_y + \beta t_x) - r\eta}{2\beta(\alpha t_x - \beta t_y) - r\xi} \tag{6.30}$$

or,

$$(\sin\phi, \cos\phi) = \frac{(2\beta(\alpha t_y + \beta t_x) - r\eta, 2\beta(\alpha t_x - \beta t_y) - r\xi)}{\sqrt{[2\beta(\alpha t_y + \beta t_x) - r\eta]^2 + [2\beta(\alpha t_x - \beta t_y) - r\xi]^2}} \tag{6.31}$$

Now, $\hat{\boldsymbol{V}}_z$ is given by

$$\hat{\boldsymbol{V}}_z = -\frac{(-\alpha t_y - \beta t_x)\hat{\boldsymbol{V}}_x + (\alpha t_x - \beta t_y)\hat{\boldsymbol{V}}_y}{\alpha(\xi t_y - \eta t_x) + \beta(\xi t_x + \eta t_y)} = \frac{(\alpha t_y + \beta t_x)\hat{\boldsymbol{V}}_x - (\alpha t_x - \beta t_y)\hat{\boldsymbol{V}}_y}{\alpha(\xi t_y - \eta t_x) + \beta(\xi t_x + \eta t_y)} \tag{6.32}$$

and with $\hat{\boldsymbol{V}}_x = \cos\theta\cos\phi$, $\hat{\boldsymbol{V}}_y = \cos\theta\sin\phi$ and $\hat{\boldsymbol{V}}_z = \sin\theta$,

$$\tan\theta = -\frac{(\alpha t_y + \beta t_x)[2\beta(\alpha t_x - \beta t_y) - r\xi] - (\alpha t_x - \beta t_y)[2\beta(\alpha t_y + \beta t_x) - r\eta]}{\alpha(\xi t_y - \eta t_x) + \beta(\xi t_x + \eta t_y)\sqrt{[2\beta(\alpha t_y + \beta t_x) - r\eta]^2 + [2\beta(\alpha t_x - \beta t_y) - r\xi]^2}} \tag{6.33}$$

or,

$$\tan\theta = -\frac{-(\alpha t_y + \beta t_x)r\xi + (\alpha t_x - \beta t_y)r\eta}{[\alpha(\xi t_y - \eta t_x) + \beta(\xi t_x + \eta t_y)]\sqrt{[2\beta(\alpha t_y + \beta t_x) - r\eta]^2 + [2\beta(\alpha t_x - \beta t_y) - r\xi]^2}} \tag{6.34}$$

$$\tan\theta = \frac{r}{\sqrt{[2\beta(\alpha t_y + \beta t_x) - r\eta]^2 + [2\beta(\alpha t_x - \beta t_y) - r\xi]^2}}. \tag{6.35}$$

Hence, the proof. ■

This proposition gives $\hat{\boldsymbol{V}}$ as function of α given the trace at every point. $\hat{\boldsymbol{V}}(\alpha)$ is a1-d trajectory in (θ, ϕ) space. This requires at least two such trajectories to intersect and hence at-least two points and their trace is required to obtain a solution of $\hat{\boldsymbol{V}}$. And more points would make the estimation robust.

6.4 Numerical Estimation of Observable from the trace.

This section describes the algorithm to compute all the observable from a sequence of images. Typically , five or seven frames are considered for measurement of one set of observable. Five frames will be used for illustration throughout this section. Consider an image sequence as shown in Figure 6.7(a).

The third-order edge-detector developed by Tamrakar *et al.*[88] is used to compute edges which provides sub-pixel position, $\boldsymbol{\gamma} = (\xi, \eta)$, and the orientation, $\boldsymbol{t}$, with sub-pixel accuracy, Figure 6.7(b). The tangent $\boldsymbol{t} = (\cos\varphi, \sin\varphi)$ also gives the normal which is given by $\boldsymbol{n} = (\cos\varphi, \sin\varphi)$. The motivation of using [88] is to obtain a better estimate of orientation of edges as the stability of estimation depends on all the observable, which will be further studied in Section6.5. Figure 6.6 shows the better accuracy of orientation of edges (green) of [88] against the one obtained using traditional operator.

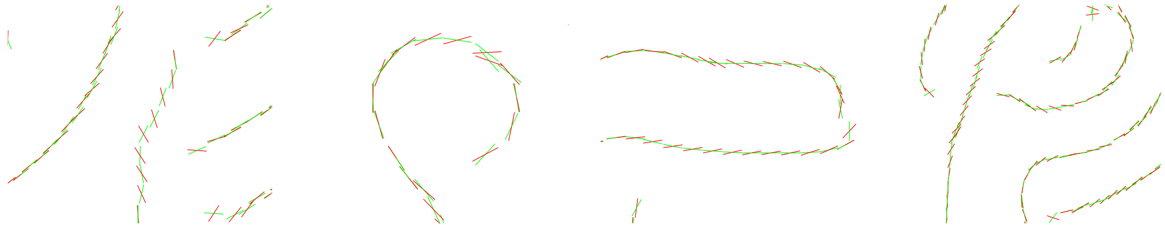


Figure 6.6: The third-order edge [88] detector (shown in green) provides a better estimate of edge-orientation as compared to traditional operators (shown in red).

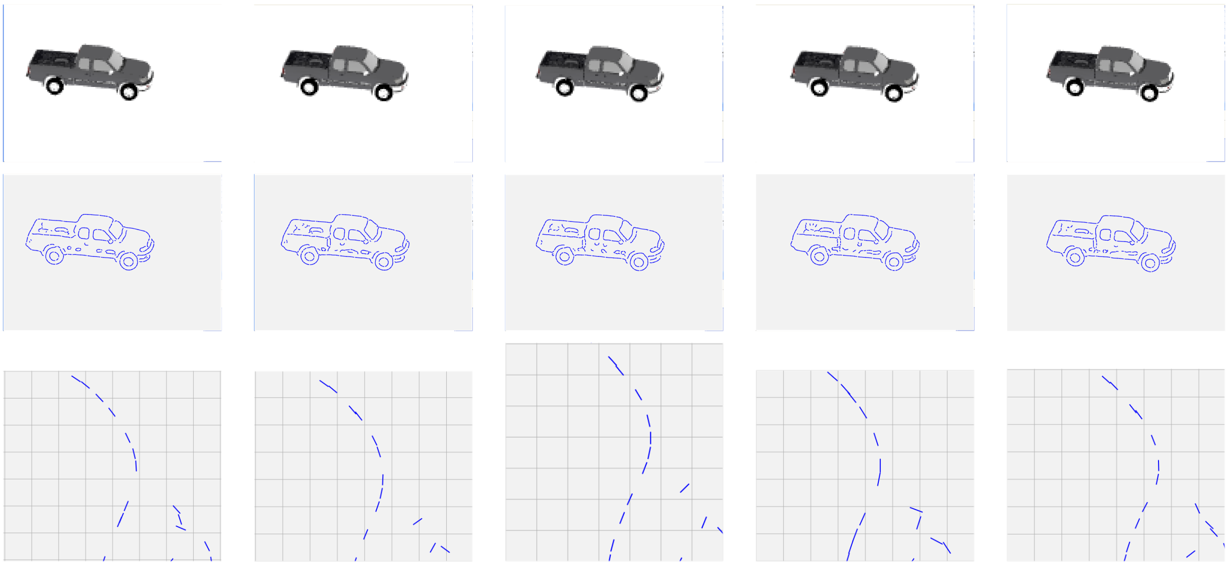


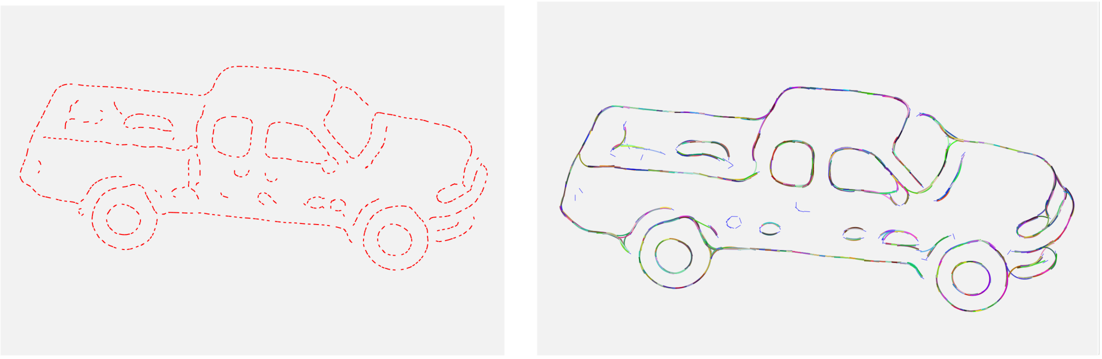
Figure 6.7: Top Row: Shows the image sequence, Middle Row: edge-maps of the above image sequence and Bottom Row: zoomed in edge-maps of neighborhood of 7×7.

The edge-maps are not enough as there are spurious edges and there is no ordering to them which is required to obtain a trace. Figure 6.7(c) shows zoomed in edge-maps and clearly the edges do not have any order. In order to obtain ordering of edges in a local window, Tamrakar *et al.*[88] approach of fitting a circular arc to a local neighborhood of edges is employed to obtain *(i)* curvature, κ, and *(ii)* local groupings of edge, Figure6.8. This method fits a local circular arc model at an edge e with position (ξ, η) and tangent $\boldsymbol{t} = (\cos\varphi, \sin\varphi)$ and $\boldsymbol{n} = -(\sin\varphi, \cos\varphi)$ given by

$$\boldsymbol{\gamma}(s) = e + s\boldsymbol{t} + \frac{1}{2}\kappa s^2 \boldsymbol{n} \tag{6.36}$$

where s is the arc-length parameterization. If the model is supported or passes through some number of edges (typically 7), then the model is kept and the grouping is called “curvelet bundle”. The circular arc model provides curvature. At each edge there can be multiple groupings but mostly there is one grouping per edge. This gives $\boldsymbol{\gamma}_0, \boldsymbol{t}, \boldsymbol{n}, \kappa$ and local trace of curve in each frame. In order to obtain trace of one-parameter family curve, these local traces in each frame needs to be grouped.

Consider one of the curvelet-bundles C^0 in the central frame, $t = 0$. A temporal window of five frames $t = \{-2, -1, 1, 2\}$ is considered and the curvelet-bundles in each of these frame within a neighborhood of size $v_{max}t$ around $\boldsymbol{\gamma}_0$ are considered. This spatio-temporal window in each frame is centered around $\boldsymbol{\gamma}_0$ and its dimension is given by $v_{max}t$. Note that it is possible to get more than one groupings in each of these windows in each frame. But the assumption is that motion between frames is less than the spacing between two groupings which is sufficient for the scope of



(a) (b)

Figure 6.8: This figures shows the pre-processing step for computing the spatial and temporal derivatives. (a) shows edge-map and (b) shows curvelet-map of of (a). Different colors show different local groupings.

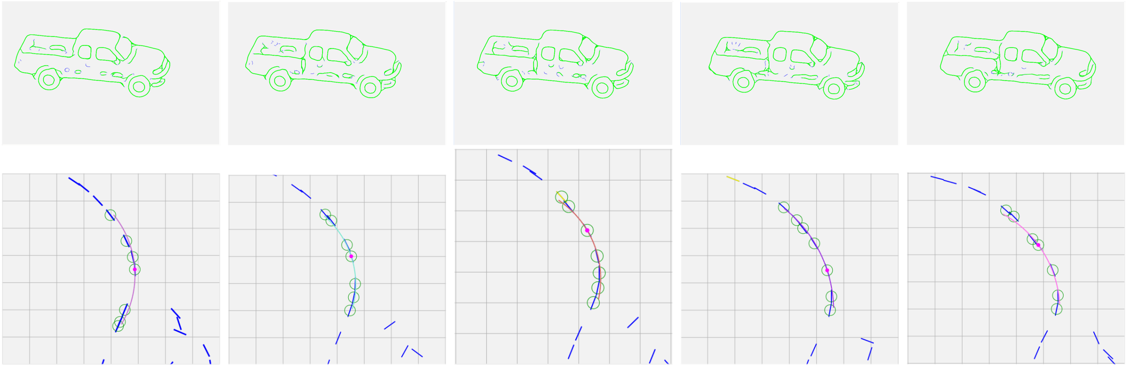


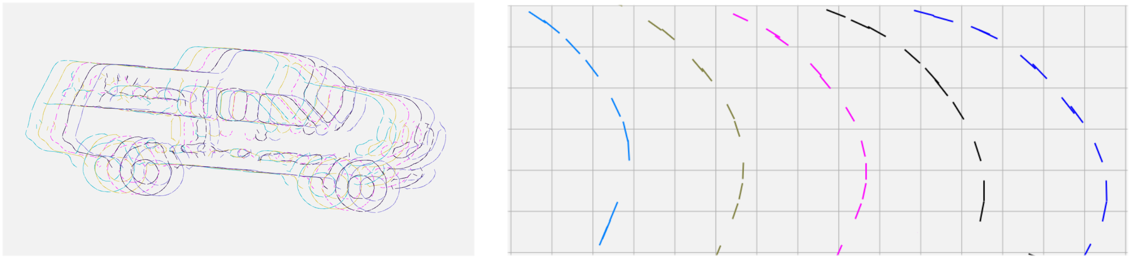
Figure 6.9: Top row: shows curvelet bundles computed on edge-maps of Figure 6.7(b) and bottom row shows the local grouping of edges in different frames.

this work. Definitely, this assumption can be violated but does not affect the result of this work. This groupings considered together gives a trace of one-parameter family of curves, Figure 6.10(b). Now the remaining observables, β, β_s, β_t, can be computed on the trace of one-parameter family of curves as described below, anchored at γ_0.

Definition 6.3. *Forward β-map is a discrete normal field where normal-velocities are computed at discrete position of edges in the current frame to local curve models in the next frame.*

Definition 6.4. *Backward β-map is a discrete normal field where normal-velocities are computed from local curve models in the previous frame to discrete position of edges in the current frame. frame.*

The idea is to compute trajectory of β for the anchor γ_0. This requires computation of *forward*



(a) (b)

Figure 6.10: (a) shows edge-maps from five consecutive frames (in different colors) in a single frame and (b) shows a local window of edge-maps from different frames.

β-map and *backward β-map*. From γ_0, the normal velocity to the next frame is computed and the head of this velocity vector acts as a point in the trajectory, Figure 6.11(a). Similarly normal velocity from this new point is computed to its next frame and so on. This normal velocity from a point to a local grouping is called forward β. For the frames previous to γ_0, normal velocity from the grouping in previous frame to γ_0 is computed and the tail of this velocity is added to the trajectory, Figure 6.11(b). Similarly, normal velocity to this new point from the grouping in previous frame is called backward β. Description of computation of both forward and backward β is discussed below:

1. Computation of forward β-map: From an edge in a frame, t, $e_i^t(p_i, \boldsymbol{t}_i, \boldsymbol{n}_i)$, the intersection of the normal ray with curvelet bundle of an edge, $e_j^{t+1}(p_j, \boldsymbol{t}_j, \boldsymbol{n}_j)$ and curvature κ_j in the $t+1$ frame is given by

$$s_{\{1,2\}} = \frac{-(\boldsymbol{t}_j \times \boldsymbol{n}_i) \pm \sqrt{|(\boldsymbol{t}_j \times \boldsymbol{n}_i)|^2 + 2\kappa_j(\boldsymbol{n}_j \times \boldsymbol{n}_i)((p_i - p_j) \times n_i)}}{\kappa(\boldsymbol{n}_j \times \boldsymbol{n}_i)} \tag{6.37}$$

the smaller of s_1 and s_2 is chosen and β is computed as

$$\beta = (p_j - p_i).\boldsymbol{n}_i + \boldsymbol{t}_j.\boldsymbol{n}_i s + \frac{1}{2}\kappa_j \boldsymbol{n}_j.\boldsymbol{n}_i s^2. \tag{6.38}$$

β is computed at each of the edges in the central frame, Figure6.11(a) . The point of intersection, s, is added to the trajectory and β's for these added points are computed in frame $t = 1$ using frame $t = 2$, which gives us forward chain of normal velocity, Figure6.11, and so on.

2. Computation of backward β-map: This requires computation of normal displacement from a curvelet bundle of an edge e_j^{t-1} to an edge e_i^t which is nothing but closest distance of a point to a circular arc, Figure6.11(e). Therefore, backward β is computed at each edge for the central frame using frame $t = -1$. Similar to the case of forward β computation, the points on the curve of e_j^{-1} are added to the trajectory and backward β's for these added points are computed using frame $t = -2$.

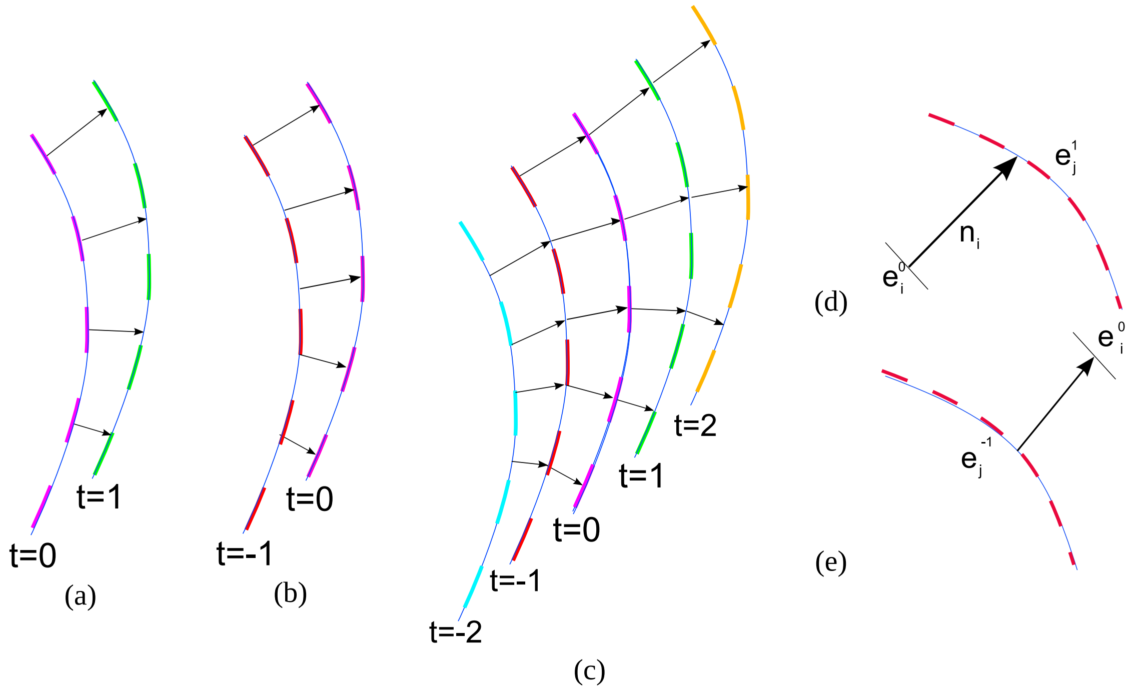


Figure 6.11: The local edge grouping in the central frame, $t = 0$, is shown in magenta. Note the forward normal velocity (a) is computed from edges to local grouping, where as backward normal velocity (b) is computed from a local grouping to an edge. (c) shows backward β-map computed for $t < 0$ frames and forward β-map computed for $t > 0$ frames. (d) shows computation of forward β and (e) backward β.

At each edge in the current frame we obtain a chain of β's as shown in Figure6.11(c). Then an interpolation scheme known as ENO [83] implemented in VXL [1] was used to find β at the central frame and its derivative β_t^* at $t = 0$. This interpolation provides robustness to the computation of β and β_t^*. This gives us β and β_t^* at every edge.

After β is known at every edge of the curvelet bundle in the central frame, same scheme as above to compute derivative along the curvelet bundle is used, *e.g.*, β's along the magenta curve in Figure6.11(c), and the first derivative at the central frame edge gives β_s.

6.5 Stability of $\hat{V}$ as a Function of Observable.

This section discusses the stability of estimation of $\hat{\boldsymbol{V}}$ as function of the measurements. Analytically, Proposition6.2 gives $\hat{\boldsymbol{V}}(\alpha)$ in terms of measurements, Φ, as one parameter curve in (θ, ϕ) space. So at-least two such one parameter curves in (θ, ϕ) space at two different points are required to obtain a solution. This means $\hat{\boldsymbol{V}}$ needs to be computed for different values of α and would intersect at the right $\hat{\boldsymbol{V}}^*$ with the other trajectory. In this section we analyze the stability of $\hat{\boldsymbol{V}}$ assuming a given α with respect to the error in measurements. The error in $\hat{\boldsymbol{V}}$ is estimated as the first-order variation of $\hat{\boldsymbol{V}}$ with respect to the measurements. This error would be computed for any given α and would be shown to be invariant by the value of α. The following proposition gives us a minimum bound on the uncertainty of $\hat{\boldsymbol{V}}$.

Theorem 6.5. *The lower bound of uncertainty in the estimation of* $\hat{\boldsymbol{V}} = \begin{bmatrix} \cos\theta\cos\phi \\ \cos\theta\sin\phi \\ \sin\theta \end{bmatrix}$ *in terms of the uncertainty of the measurements is given as*

$$\delta\hat{\mathbf{V}} \geq \sqrt{A^2 + B^2 + C^2} \tag{6.39}$$

where

$$\begin{aligned} A &= \frac{(\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))\sin\theta\cos\phi)^2+(l\sin(\phi-\mu)\cos\theta\sin\phi)^2}{\sqrt{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)}} \\ B &= \frac{(\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))\sin\theta\sin\phi)^2+(l\sin(\phi-\mu)\cos\theta\cos\phi)^2}{\sqrt{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)}} \\ C &= \frac{\cos^3\theta(1+l\tan\theta\cos(\phi-\mu))}{\sqrt{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)}} \end{aligned} \tag{6.40}$$

Proof. Differentiating $\hat{\boldsymbol{V}}(\theta, \phi) = \begin{bmatrix} \cos\theta\cos\phi \\ \cos\theta\sin\phi \\ \sin\theta \end{bmatrix}$ with respect to (θ, ϕ) gives

$$\frac{\partial\hat{\mathbf{V}}}{\partial\theta} = \begin{bmatrix} -\sin\theta\cos\phi \\ -\sin\theta\sin\phi \\ \cos\theta \end{bmatrix} \quad \frac{\partial\hat{\mathbf{V}}}{\partial\phi} = \begin{bmatrix} -\cos\theta\sin\phi \\ \cos\theta\cos\phi \\ 0 \end{bmatrix}. \tag{6.41}$$

Next differentiate θ and ϕ, from Proposition 6.2 w.r.t zeroth-order measurements ξ and η, first-order measurements β and ϕ (orientation of the tangent of the edge) and second-order measurements β_s, β_t and κ. Note that Since $r = w^2\kappa + w\beta_s + \beta_t$ has all the second order quantities, it will be considered as one variable and $\hat{\boldsymbol{V}}$ will be differentiated with respect to r. The derivatives w.r.t zeroth-order measurements give

$$\left\{ \begin{array}{llllll} \frac{\partial\theta}{\partial\xi} & = & \sin^2\theta\cos\phi & \frac{\partial\phi}{\partial\xi} & = & \tan\theta\sin\phi \\ \frac{\partial\theta}{\partial\eta} & = & \sin^2\theta\sin\phi & \frac{\partial\phi}{\partial\eta} & = & -\tan\theta\cos\phi \end{array} \right., \tag{6.42}$$

and the derivatives w.r.t first-order measurements are given as

$$\left\{ \begin{array}{llllll} \frac{\partial\theta}{\partial\beta} & = & \frac{-2\cos\theta\sin\theta(\sin\phi(w\boldsymbol{t}_y+2\beta\boldsymbol{t}_x)+\cos\phi(w\boldsymbol{t}_x-2\beta\boldsymbol{t}_y))}{\sqrt{[2\beta(wt_y+\beta t_x)-r\eta]^2+[2\beta(wt_x-\beta t_y)-r\xi]^2}} & \frac{\partial\phi}{\partial\beta} & = & \frac{\cos\phi(2w\boldsymbol{t}_y+4\beta\boldsymbol{t}_x)+\sin\phi(2w\boldsymbol{t}_x-4\beta\boldsymbol{t}_y)}{\sqrt{[2\beta(wt_y+\beta t_x)-r\eta]^2+[2\beta(wt_x-\beta t_y)-r\xi]^2}} \\ \frac{\partial\theta}{\partial\varphi} & = & \frac{-2\beta\cos\theta\sin\theta((w\boldsymbol{t}_x-\beta\boldsymbol{t}_y)\sin\phi+(-w\boldsymbol{t}_y-\beta\boldsymbol{t}_x)\cos\phi)}{\sqrt{[2\beta(wt_y+\beta t_x)-r\eta]^2+[2\beta(wt_x-\beta t_y)-r\xi]^2}} & \frac{\partial\phi}{\partial\varphi} & = & \frac{(2\beta(w\boldsymbol{t}_x-\beta\boldsymbol{t}_y)\cos\phi+2\beta(-w\boldsymbol{t}_y-2\beta\boldsymbol{t}_x)\sin\phi)}{\sqrt{[2\beta(wt_y+\beta t_x)-r\eta]^2+[2\beta(wt_x-\beta t_y)-r\xi]^2}} \end{array} \right., \tag{6.43}$$

and the derivatives w.r.t second-order measurements are given as

$$\left\{ \begin{array}{lllll} \frac{\partial\theta}{\partial r} & = & \frac{\cos^2\theta(1+\tan\theta(\sin\phi\eta+\cos\phi\xi))}{\sqrt{[2\beta(wt_y+\beta t_x)-r\eta]^2+[2\beta(wt_x-\beta t_y)-r\xi]^2}} & \frac{\partial\phi}{\partial r} & = & \frac{-\cos\phi\eta+\sin\phi\xi}{\sqrt{[2\beta(wt_y+\beta t_x)-r\eta]^2+[2\beta(wt_x-\beta t_y)-r\xi]^2}} \end{array} \right. . \tag{6.44}$$

Substituting $(\xi,\eta) = l(\cos\mu,\sin\mu)$, $(w,\beta) = v(\cos\nu,\sin\nu)$ and $\boldsymbol{t} = (\cos\varphi,\sin\varphi)$ in Equations 6.42, 6.43 and 6.44,

$$\left\{ \begin{array}{llllll} \frac{\partial\theta}{\partial r} & = & \frac{\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))}{\sqrt{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)}} & \frac{\partial\phi}{\partial r} & = & \frac{l\sin(\phi-\mu)}{\sqrt{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)}} \\ \frac{\partial\theta}{\partial\beta} & = & \frac{-\sin 2\theta(v\sin(\phi+\varphi+\nu)+\beta\sin(\phi-\varphi))}{\sqrt{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)}} & \frac{\partial\phi}{\partial\beta} & = & \frac{2v\cos(\phi+\varphi+\nu)+2\beta\cos(\phi+\varphi))}{\sqrt{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)}} \\ \frac{\partial\theta}{\partial\xi} & = & \sin^2\theta\cos\phi & \frac{\partial\phi}{\partial\xi} & = & \tan\theta\sin\phi \\ \frac{\partial\theta}{\partial\eta} & = & \sin^2\theta\sin\phi & \frac{\partial\phi}{\partial\eta} & = & -\tan\theta\cos\phi \\ \frac{\partial\theta}{\partial\varphi} & = & \frac{-\beta v\sin 2\theta\sin(\phi-\varphi-\nu)}{\sqrt{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)}} & \frac{\partial\phi}{\partial\varphi} & = & \frac{2\beta v\cos(\phi+\varphi+\nu)}{\sqrt{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)}} \end{array} \right. \tag{6.45}$$

Now error in $\delta\theta$ and $\delta\phi$ can be expressed as a linear combination of the uncertainties in the measurement and is given as

$$\left\{ \begin{array}{lll} (\delta\theta)^2 & = & (\frac{\partial\theta}{\partial r})^2(\delta r)^2 + (\frac{\partial\theta}{\partial\beta})^2(\delta\beta)^2 + (\frac{\partial\theta}{\partial\xi})^2(\delta\xi)^2 + (\frac{\partial\theta}{\partial\eta})^2(\delta\eta)^2 + (\frac{\partial\theta}{\partial\varphi})^2(\delta\varphi)^2 \\ (\delta\phi)^2 & = & (\frac{\partial\phi}{\partial r})^2(\delta r)^2 + (\frac{\partial\phi}{\partial\beta})^2(\delta\beta)^2 + (\frac{\partial\phi}{\partial\xi})^2(\delta\xi)^2 + (\frac{\partial\phi}{\partial\eta})^2(\delta\eta)^2 + (\frac{\partial\phi}{\partial\varphi})^2(\delta\varphi)^2 \end{array} \right. . \tag{6.46}$$

Due to extensive calculations for each of the derivatives, the error due to second-order measurements are studied. Note that this does not affect the result of the proposition as the lower bounds need to be estimated. So the $\delta\theta$ and $\delta\phi$ with respect to second-order measurements is given as

$$\left\{ \begin{array}{lll} (\delta\theta)^2 & \geq & (\frac{\partial\theta}{\partial r})^2(\delta r)^2 \\ (\delta\phi)^2 & \geq & (\frac{\partial\phi}{\partial r})^2(\delta r)^2 \end{array} \right. . \tag{6.47}$$

The error in the direction of translation $\hat{\boldsymbol{V}}$ as

$$\delta\hat{\boldsymbol{V}} = \|\frac{\partial\hat{\boldsymbol{V}}}{\partial\theta}\delta\theta + \frac{\partial\hat{\boldsymbol{V}}}{\partial\phi}\delta\phi\| = \sqrt{(\delta\hat{\boldsymbol{V}}_x)^2 + (\delta\hat{\boldsymbol{V}}_y)^2 + (\delta\hat{\boldsymbol{V}}_z)^2} \tag{6.48}$$

where $\hat{\boldsymbol{V}}_x$, $\hat{\boldsymbol{V}}_y$ and $\hat{\boldsymbol{V}}_z$ represents the three components of $\hat{\boldsymbol{V}}$ and their uncertainties are given by

$$\begin{array}{lll} \delta\hat{\boldsymbol{V}}_x & = & (\frac{\partial\hat{\boldsymbol{V}}_x}{\partial\theta})\delta\theta + (\frac{\partial\hat{\boldsymbol{V}}_x}{\partial\phi})\delta\phi \\ \delta\hat{\boldsymbol{V}}_y & = & (\frac{\partial\hat{\boldsymbol{V}}_y}{\partial\theta})\delta\theta + (\frac{\partial\hat{\boldsymbol{V}}_y}{\partial\phi})\delta\phi \\ \delta\hat{\boldsymbol{V}}_z & = & (\frac{\partial\hat{\boldsymbol{V}}_z}{\partial\theta})\delta\theta \end{array} \tag{6.49}$$

Substituting values of $\delta\hat{\boldsymbol{V}}_x$, $\delta\hat{\boldsymbol{V}}_y$ and $\delta\hat{\boldsymbol{V}}_z$ in Equation 6.48 gives

$$\begin{array}{lll} \delta\hat{\boldsymbol{V}} & = & \sqrt{[(\frac{\partial\hat{\boldsymbol{V}}_x}{\partial\theta})\delta\theta + (\frac{\partial\hat{\boldsymbol{V}}_x}{\partial\phi})\delta\phi]^2 + [(\frac{\partial\hat{\boldsymbol{V}}_y}{\partial\theta})\delta\theta + (\frac{\partial\hat{\boldsymbol{V}}_y}{\partial\phi})\delta\phi]^2 + [(\frac{\partial\hat{\boldsymbol{V}}_z}{\partial\theta})\delta\theta]^2} \\ & = & \sqrt{(\frac{\partial\hat{\boldsymbol{V}}_x}{\partial\theta})^2(\delta\theta)^2 + (\frac{\partial\hat{\boldsymbol{V}}_x}{\partial\phi})(\delta\phi)^2 + (\frac{\partial\hat{\boldsymbol{V}}_y}{\partial\theta})^2(\delta\theta)^2 + (\frac{\partial\hat{\boldsymbol{V}}_y}{\partial\phi})^2(\delta\phi)^2 + (\frac{\partial\hat{\boldsymbol{V}}_z}{\partial\theta})^2(\delta\theta)^2} \end{array} \tag{6.50}$$

and further substituting $\delta\theta$ and $\delta\phi$ from Equation 6.47

$$\delta\hat{\boldsymbol{V}} \geq \delta r\sqrt{(\frac{\partial\hat{\boldsymbol{V}}_x}{\partial\theta})^2(\frac{\partial\theta}{\partial r})^2+(\frac{\partial\hat{\boldsymbol{V}}_x}{\partial\phi})(\frac{\partial\phi}{\partial r})^2+(\frac{\partial\hat{\boldsymbol{V}}_y}{\partial\theta})^2(\frac{\partial\theta}{\partial r})^2+(\frac{\partial\hat{\boldsymbol{V}}_y}{\partial\phi})^2(\frac{\partial\phi}{\partial r})^2+(\frac{\partial\hat{\boldsymbol{V}}_z}{\partial\theta})^2(\frac{\partial\theta}{\partial r})^2} \tag{6.51}$$

where

$$\begin{array}{rcl} (\frac{\partial\hat{\boldsymbol{V}}_x}{\partial\theta})^2(\frac{\partial\theta}{\partial r})^2+(\frac{\partial\hat{\boldsymbol{V}}_x}{\partial\phi})^2(\frac{\partial\phi}{\partial r})^2 & = & \frac{(\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))\sin\theta\cos\phi)^2+(l\sin(\phi-\mu)\cos\theta\sin\phi)^2}{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)} \\ (\frac{\partial\hat{\boldsymbol{V}}_y}{\partial\theta})^2(\frac{\partial\theta}{\partial r})^2+(\frac{\partial\hat{\boldsymbol{V}}_y}{\partial\phi})^2(\frac{\partial\phi}{\partial r})^2 & = & \frac{(\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))\sin\theta\sin\phi)^2+(l\sin(\phi-\mu)\cos\theta\cos\phi)^2}{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)} \\ (\frac{\partial\hat{\boldsymbol{V}}_z}{\partial\theta})^2(\frac{\partial\theta}{\partial r})^2\delta r & = & \frac{\cos^3\theta(1+l\tan\theta\cos(\phi-\mu))}{r^2l^2+4\beta^2v^2-4\beta vrl\sin(\varphi+\mu+\nu)} \end{array} \tag{6.52}$$

Hence, the proof. ■

Corollary 6.6. *The minimum bound for* $\delta\hat{\boldsymbol{V}}$ *can be further simplified to*

$$\delta\hat{\boldsymbol{V}} \geq \delta r\sqrt{A_1^2+B_1^2+C_1^2} \tag{6.53}$$

where

$$\begin{array}{rcl} A_1 & = & \frac{(\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))\sin\theta\cos\phi)^2+(l\sin(\phi-\mu)\cos\theta\sin\phi)^2}{||rl|+|2\beta v||} \\ B_1 & = & \frac{(\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))\sin\theta\sin\phi)^2+(l\sin(\phi-\mu)\cos\theta\cos\phi)^2}{||rl|+|2\beta v||} \\ C_1 & = & \frac{\cos^3\theta(1+l\tan\theta\cos(\phi-\mu))}{||rl|+|2\beta v||} \end{array} \tag{6.54}$$

Proof. Now, consider the denominators of A, B and C, $\sqrt{r^2l^2+4\beta^2v^2-4rl\beta v\sin(\varphi+\mu+\nu)}$ is bounded by

$$||rl|-|2\beta v|| \leq \sqrt{r^2l^2+4\beta^2v^2-4rl\beta v\sin(\varphi+\mu+\nu)} \leq ||rl|+|2\beta v|| \tag{6.55}$$

Hence, the proof. ■

The lower bound on $\delta\hat{\boldsymbol{V}}$ is function of $r, l, \beta, v, \theta, \phi, \delta r$. Since these are too many variables and the variation of $\delta\hat{\boldsymbol{V}}$ with respect to $\hat{\boldsymbol{V}}$ and position in the image are more interesting and meaningful, rest of the variables need to be eliminated. This motivates estimation of bounds on r, β, l, v.

Bounds on $r, \beta, l, v, \delta r$**:** The image velocity $\boldsymbol{\gamma}_t$ has to be small of the order of few pixels as the differential quantities like β, β_t, β_s would not hold over large motions. Let us denote maximum value of $\boldsymbol{\gamma}_t$ by v^{max}. This implies β and w are also bounded by v^{max}. Since $\boldsymbol{\gamma}_{tt}$ is given by

$$\boldsymbol{\gamma}_{tt} = \frac{-2\boldsymbol{V}_z}{\rho}\boldsymbol{\gamma}_t \tag{6.56}$$

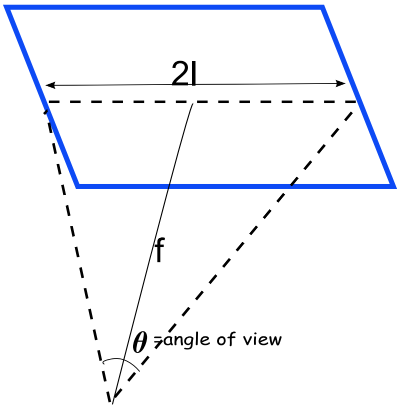


Figure 6.12: This figure shows the relationship between focal length (f), angle of view (θ) and the dimension of the image (l).

γ_{tt} is bounded

$$\gamma_{tt} \leq 2|\frac{\boldsymbol{V}_z}{\rho}|^{max}|\gamma_t|^{max} = 2|\frac{\boldsymbol{V}_z}{\rho}|^{max}v^{max} \tag{6.57}$$

since $r = \boldsymbol{\gamma}_{tt}.\boldsymbol{n}$,

$$r \leq \boldsymbol{\gamma}_{tt}^{max} = 2|\frac{\boldsymbol{V}_z}{\rho}|^{max}v^{max} \tag{6.58}$$

The imaging sensor has typically an angle of view less than 60°. This means dimension of the sensor $2l$ is bounded by $l^{max} \leq f \tan 30^\circ$ where f is the focal length. Let the error in position of an edge be $\delta\boldsymbol{\gamma}$. In the computation of β and its derivatives, the edge in the central frame can be assumed to have zero position error along normal direction but the edges in the neighboring frame would have an error of $\delta\boldsymbol{\gamma}$. Therefore error in β can be bounded by

$$\delta\beta \simeq \delta\boldsymbol{\gamma}$$

and similarly, $\delta\beta_t \simeq 2\delta\boldsymbol{\gamma}$ and $\delta\beta_s \simeq 2\delta\boldsymbol{\gamma}$. With $r = 2w\beta_s + w^2\kappa + \beta_t$, r is linear in β_s, β_t, κ, the error can be estimated by

$$\delta r^2 = (2w\delta\beta_s)^2 + w^4(\delta\kappa)^2 + (\delta\beta_t)^2 \geq (\delta\beta_t)^2 \tag{6.59}$$

Corollary 6.7. *The minimum bound for $\delta\hat{\boldsymbol{V}}$ can be further simplified to*

$$\delta\hat{\boldsymbol{V}} \geq |\delta\beta_t|\sqrt{A_2^2 + B_2^2 + C_2^2} \tag{6.60}$$

where

$$\begin{aligned} A_2 &= \frac{(\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))\sin\theta\cos\phi)^2+(l\sin(\phi-\mu)\cos\theta\sin\phi)^2}{||r^{max}l|+|2(v^{max})^2||} \\ B_2 &= \frac{(\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))\sin\theta\sin\phi)^2+(l\sin(\phi-\mu)\cos\theta\cos\phi)^2}{||r^{max}l|+|2(v^{max})^2||} \\ C_2 &= \frac{\cos^3\theta(1+l\tan\theta\cos(\phi-\mu))}{||r^{max}l|+|2(v^{max})^2||} \end{aligned} \tag{6.61}$$

Proof. Note that r, β and v occur in the denominator and

$$||rl| + |2\beta v|| \leq ||r^{max}l| + |2\beta^{max}v^{max}|| = ||r^{max}l| + |2(v^{max})^2||. \tag{6.62}$$

Further, $\delta r \geq \delta\beta_t$. Hence, the proof. ■

Note that minimum bound of error has been further simplified and is independent of α and only depends on θ, ϕ, l, which are varied to compute $\delta\hat{\boldsymbol{V}}$ in the next section.

6.6 Analysis and Experiments

In this section, the minimum error is computed by varying the values of l from a typical video acquisition system and is varied also for different ϕ and θ. The image velocity is assumed to be maximum of $v_{max} = 3$ pixels/frame on an image of dimensions 512×512 . The maximum velocity of an object w.r.t to the camera is assumed to be 108 Km/h. The distance of the object from the video is assumed to be minimum 6 meters. The bound for $\boldsymbol{\gamma}_{tt}$ is then

$$\boldsymbol{\gamma}_{tt} \leq 2 \times \frac{1}{6} \times 3 = 1 \text{ pixels}$$

This implies $r^{max} = 1$. With normalized focal length $f = 1$, $l^{max} = 0.57$. For high-resolution images the error in localization of edge positions has been observed to be $\delta\boldsymbol{\gamma} = 0.25$ pixels. This implies $\delta\beta_t \geq 0.5$, using Equation 6.59. The r$^{max}, \delta\beta_t, v^{max}$ need to be normalized. Since the image size is 512 which is equal to l^{max}. Therefore, normalized values for r$^{max}, \delta\beta_t, v^{max}$ are given by

$$\begin{cases} r^{max} &= 1 * l^{max}/512 \\ \delta\beta_t &= 0.5 * l^{max}/512 \\ v^{max} &= 3 * l^{max}/512 \end{cases} \tag{6.63}$$

Substituting the values of $r^{max}, \delta\beta_t, v^{max}$ in Corollary 6.7 gives

$$\delta\hat{\boldsymbol{V}} \geq 0.5\sqrt{A_3^2 + B_3^2 + C_3^2} \tag{6.64}$$

where

$$\begin{aligned} A_3 &= \frac{(\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))\sin\theta\cos\phi)^2+(l\sin(\phi-\mu)\cos\theta\sin\phi)^2}{||l^{max}l|+|0.035(l^{max})^2||} \\ B_3 &= \frac{(\cos^2\theta(1+l\tan\theta\cos(\phi-\mu))\sin\theta\sin\phi)^2+(l\sin(\phi-\mu)\cos\theta\cos\phi)^2}{||l^{max}l|+|0.035(l^{max})^2||} \\ C_3 &= \frac{\cos^3\theta(1+l\tan\theta\cos(\phi-\mu))}{||l^{max}l|+|0.035(l^{max})^2||} \end{aligned} \tag{6.65}$$

The uncertainty $\delta\hat{\boldsymbol{V}}$ depends on (θ,ϕ), l and μ. μ is the tangent angle which could be assumed to be zero without loss of generality. Therefore, $\delta\hat{\boldsymbol{V}}$ is plot against the variation of (θ,ϕ) and l. Figure 6.13 (a) shows the plots for the uncertainty for different samples of 3D motion and for three different positions in the image *(i)* near the optical center, $l = 0.05$, *(ii)* in the middle of the image $l = 0.25$ and *(iii)* at the periphery of the image, $l = l^{max}$. Another set of binary plots, Figure 6.13 (in blue and maroon), shows the motion direction (θ,ϕ) for which the error is more than one (in maroon). These area shows that estimation of motion direction for these (θ,ϕ) is merely up to chance. The observations are as follows:

1. Note that the $\hat{\boldsymbol{V}}$ for all (θ,ϕ) decreases as l is varied from the center of the image periphery of the image. The error for pixel position near the center of the image is huge. Also, note the area covered in maroon in second plot of Figure 6.13 (b) for l near the optical axis. It covers almost the entire range of motion direction. Similarly, it covers more than half for $l = l_{max}/2$and at-least one third for $l = l_{max}$.

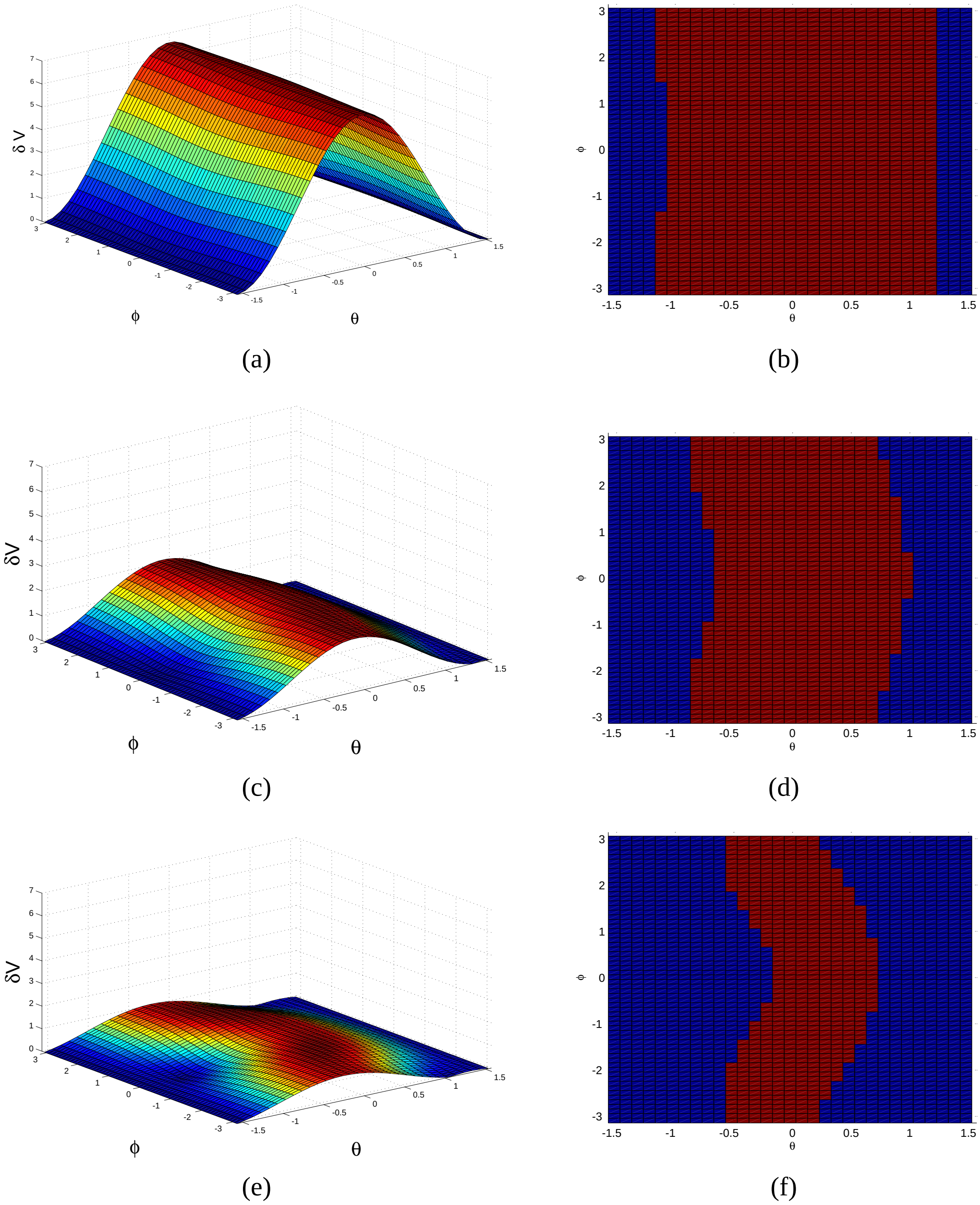
2. $\delta\hat{\boldsymbol{V}}$ is always maximum for θ near or equal to zero. This means that the when the motions are merely parallel to the image plane, they exhibit higher error than when the motion is perpendicular to the plane.

The above observations make it clear that the direction of the motion cannot be estimated for general cases. [100] concluded that the angle of view should be increased which means increase l_{max}. But angle of view cannot be increased in practice for a typical acquisition system. This limitation of high error in $\hat{\boldsymbol{V}}$ renders segmentation based on 3D motion unsolved from a monocular video sequence.

Recovery of some aspects of $\hat{\boldsymbol{V}}$: Figure 6.14 shows the plots of $\delta\hat{\boldsymbol{V}}_x$, $\delta\hat{\boldsymbol{V}}_y$ and $\delta\hat{\boldsymbol{V}}_z$. Note that the error in $\delta\hat{\boldsymbol{V}}_z$ is higher than error in $\delta\hat{\boldsymbol{V}}_x$ and $\delta\hat{\boldsymbol{V}}_y$. Further, the $\delta\hat{\boldsymbol{V}}_x$ and $\delta\hat{\boldsymbol{V}}_y$ does not have large error and hopefully can be recovered. The ratio of $\delta\hat{\boldsymbol{V}}_y/\delta\hat{\boldsymbol{V}}_x$ is given by $\tan\phi$ which is

$$\tan\phi = \frac{2\beta(\alpha t_y + \beta t_x) - r\eta}{2\beta(\alpha t_x - \beta t_y) - r\xi}. \tag{6.66}$$

This above ratio might be recovered and used for segmentation. Note that this is different from using 2D motion, $\boldsymbol{\gamma}_t = \alpha\boldsymbol{t} + \beta\boldsymbol{n}$ as shown in Equation 6.66.



(a) (b)

(c) (d)

(e) (f)

Figure 6.13: Left column shows the plots of $\delta\hat{\boldsymbol{V}}$ for all (θ, ϕ) for *(a)* near the optical center, $l = 0.05$, *(c)* in the middle of the image $l = l^{max}/2$ and *(e)* at the periphery of the image, $l = l^{max}$. Right column shows a binary map of the plots of the left column where blue represents error less than 1 and maroon represents error equal to or greater than 1.

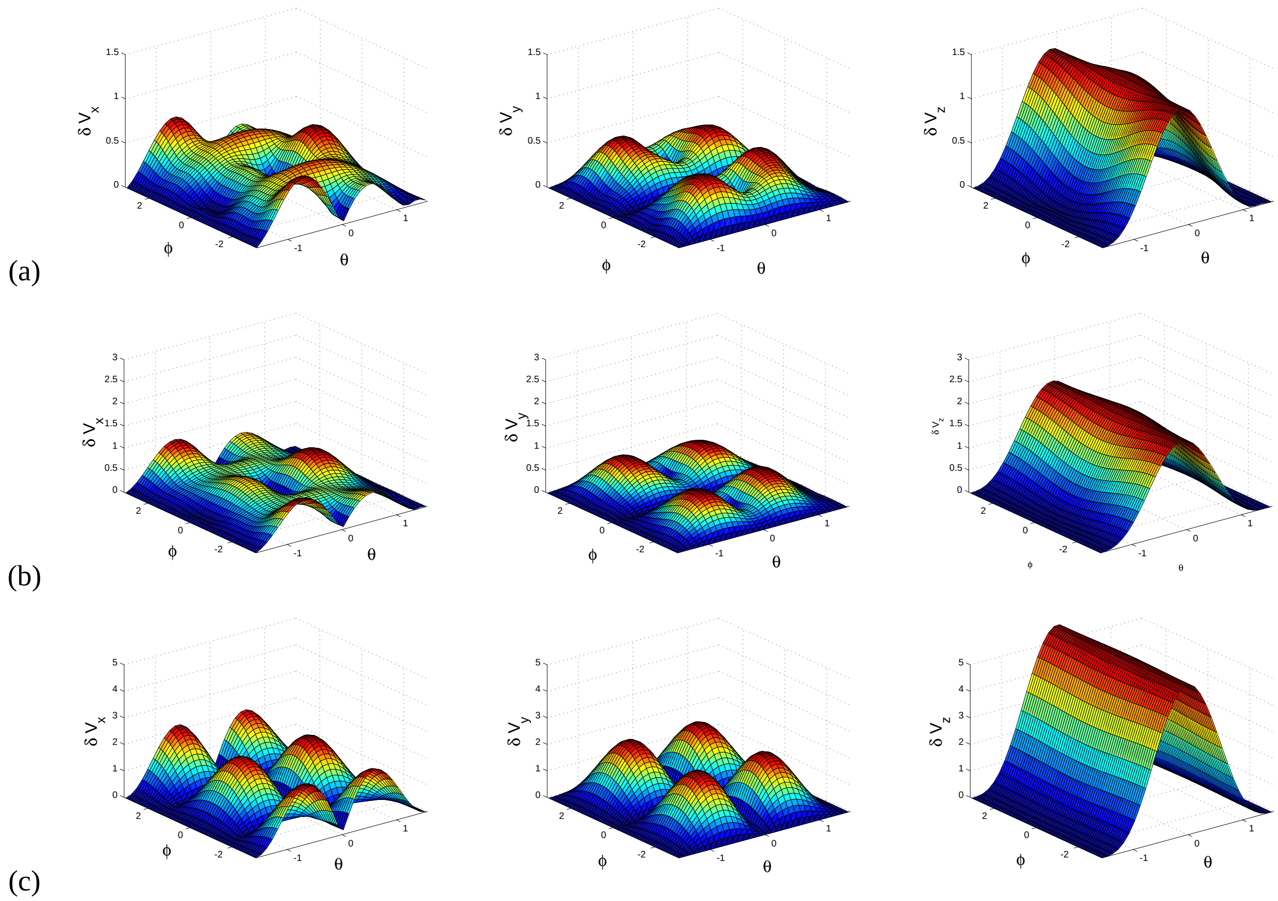


Figure 6.14: This figure shows the plots of x, y and z components of $\delta\hat{\boldsymbol{V}}$ for all (θ, ϕ) for *(a)* near the optical center, $l = 0.05$, *(c)* in the middle of the image $l = l^{max}/2$ and *(e)* at the periphery of the image, $l = l^{max}$. Note that the errors in $\delta\hat{\boldsymbol{V}}_x$ and $\delta\hat{\boldsymbol{V}}_y$ is not large.

Appendix A

Evolution of curves: trace and parameterization

Proposition A.1. *For a one-parameter family of curves* $\boldsymbol{\gamma}(s,t)$ *defined by the evolutionary model by*

$$\begin{cases} \frac{\partial \gamma}{\partial s}(s,t) &= g(s,t)\boldsymbol{t}(s,t) \\ \frac{\partial \gamma}{\partial t}(s,t) &= \alpha(s,t)\boldsymbol{t}(s,t)+\beta(s,t)\boldsymbol{n}(s,t). \end{cases}$$

where $g(s,t)$ *is defined as speed of parameterization,then the trace of* $\boldsymbol{\gamma}(s,t)$ *is given by*

$$\frac{\partial \boldsymbol{\gamma}}{\partial t}(s,t)=\beta(s,t)\boldsymbol{n}(s,t). \tag{A.1}$$

Proof. Consider a different parameterization $\boldsymbol{\gamma}_1(\overline{s},t)=\boldsymbol{\gamma}(s,t)$, where $\overline{s}=\overline{s}(s,t)$. Differentiating both sides of $\boldsymbol{\gamma}(s,t)=\boldsymbol{\gamma}_1(\overline{s},t)$ with respect to t we have

$$\begin{aligned} \frac{\partial \gamma}{\partial t} &= \frac{\partial \gamma_1}{\partial \overline{s}}.\frac{\partial \overline{s}}{\partial t}+\frac{\partial \gamma_1}{\partial t} \\ \alpha \boldsymbol{t}+\beta \boldsymbol{n} &= g_1 \boldsymbol{t}_1 \frac{\partial \overline{s}}{\partial t}+\frac{\partial \gamma_1}{\partial t}. \end{aligned} \tag{A.2}$$

Now to relate $\boldsymbol{t}_1$ to $\boldsymbol{t}$, differentiate $\boldsymbol{\gamma}_1(\overline{s}(s,t))=\boldsymbol{\gamma}(s,t)$ with respect to s so that

$$\begin{aligned} \frac{\partial \gamma}{\partial s} &= \frac{\partial \gamma}{\partial \overline{s}}.\frac{\partial \overline{s}}{\partial s}, \\ g\boldsymbol{t} &= g_1 \boldsymbol{t}_1 \frac{\partial \overline{s}}{\partial s}, \end{aligned} \tag{A.3}$$

which gives

$$\begin{cases} \boldsymbol{t}_1(s,t) &= \boldsymbol{t}(s,t) \\ g_1(\overline{s}(s,t))\frac{\partial \overline{s}}{\partial s}(\overline{s}(s,t)) &= g(s,t) \end{cases} \tag{A.4}$$

and also

$$\boldsymbol{n}_1(\overline{s}(s,t))=\boldsymbol{n}(s,t)$$

Substituting these into Equation A.2 we have

$$\alpha\boldsymbol{t} + \beta\boldsymbol{n} = g\boldsymbol{t}\frac{\frac{\partial\bar{s}}{\partial t}}{\frac{\partial\bar{s}}{\partial s}} + \frac{\partial\gamma_1}{\partial t}, \tag{A.5}$$

or rearranging,

$$\frac{\partial\gamma_1}{\partial t} = (\alpha - g\frac{\frac{\partial\bar{s}}{\partial t}}{\frac{\partial\bar{s}}{\partial s}})\boldsymbol{t} + \beta\boldsymbol{n}.$$

Now, choose $\bar{s}$ such that $\alpha\frac{\partial\bar{s}}{\partial s} = g\frac{\partial\bar{s}}{\partial t}$, we have

$$\frac{\partial\gamma_1}{\partial t} = \beta(s,t)\boldsymbol{n}(s,t).$$

which shows that the trace can be obtained by the above equation. Hence, the proof. ■

Proposition A.2. *For a one-parameter family of curves $\boldsymbol{\gamma}(s,t)$ defined by the evolutionary model by*

$$\begin{cases} \frac{\partial\boldsymbol{\gamma}}{\partial s}(s,t) &= g(s,t)\boldsymbol{t}(s,t) \\ \frac{\partial\boldsymbol{\gamma}}{\partial t}(s,t) &= \alpha(s,t)\boldsymbol{t}(s,t) + \beta(s,t)\boldsymbol{n}(s,t). \end{cases}$$

where $g(s,t)$ is defined as speed of parameterization,then the second order derivatives $\frac{\partial^2\boldsymbol{\gamma}}{\partial s^2}$, $\frac{\partial^2\boldsymbol{\gamma}}{\partial s\partial t}$ and $\frac{\partial^2\boldsymbol{\gamma}}{\partial t^2}$ are given by

$$\begin{cases} \frac{\partial^2\boldsymbol{\gamma}}{\partial s^2} &= g_s\boldsymbol{t} + g^2\kappa\boldsymbol{n} \\ \frac{\partial^2\boldsymbol{\gamma}}{\partial s\partial t} &= (\alpha_s - \beta g\kappa)\boldsymbol{t} + (\alpha g\kappa + \beta_s)\boldsymbol{n} \\ \frac{\partial^2\boldsymbol{\gamma}}{\partial t^2} &= (\alpha_t - \beta(\alpha\kappa + \frac{\beta_s}{g}))\boldsymbol{t} + (\alpha(\alpha\kappa + \frac{\beta_s}{g}) + \beta_t)\boldsymbol{n}. \end{cases} \tag{A.6}$$

Furthermore, assuming arc-length parameterization at $t = 0$, i.e.*, $g(s,0) = 1$ and $g_s(s,0) = 0$, the second order-derivatives are given as*

$$\begin{cases} \frac{\partial^2\boldsymbol{\gamma}}{\partial s^2} &= \kappa\boldsymbol{n} \\ \frac{\partial^2\boldsymbol{\gamma}}{\partial s\partial t} &= (\alpha_s - \beta\kappa)\boldsymbol{t} + (\alpha\kappa + \beta_s)\boldsymbol{n} \\ \frac{\partial^2\boldsymbol{\gamma}}{\partial t^2} &= (\alpha_t - \beta(\alpha\kappa + \beta_s))\boldsymbol{t} + (\alpha(\alpha\kappa + \beta_s) + \beta_t)\boldsymbol{n}. \end{cases} \tag{A.7}$$

Proof. First, lets compute derivatives of $\boldsymbol{t}$ and $\boldsymbol{n}$, where $\boldsymbol{t} = (\cos\theta, \sin\theta)$ and $\boldsymbol{n} = (-\sin\theta, \cos\theta)$ which are given by

$$\begin{aligned} \boldsymbol{t}_s &= (-\sin\theta, \cos\theta)\theta_s &&= \theta_s\boldsymbol{n} \\ \boldsymbol{n}_s &= (-\cos\theta, -\sin\theta)\theta_s &&= -\theta_s\boldsymbol{t} \\ \boldsymbol{t}_t &= (-\sin\theta, \cos\theta)\theta_t &&= \theta_t\boldsymbol{n} \\ \boldsymbol{n}_t &= (-\cos\theta, -\sin\theta)\theta_t &&= -\theta_t\boldsymbol{t} \end{aligned} \tag{A.8}$$

and computing the second order derivatives by differentiating the Equation**??** we get

$$\begin{aligned}\boldsymbol{\gamma}_{ts} &= (g\boldsymbol{t})_s \\ &= g_s\boldsymbol{t} + g\boldsymbol{t}_s \\ &= g_s\boldsymbol{t} + g^2\kappa\boldsymbol{n}\end{aligned}$$

$$\begin{aligned}\boldsymbol{\gamma}_{st} &= (\alpha\boldsymbol{t}(s,t) + \beta\boldsymbol{n}(s,t))_s \\ &= \alpha_s\boldsymbol{t} + \alpha\boldsymbol{t}_s + \beta_s\boldsymbol{n} + \beta\boldsymbol{n}_s \\ &= \alpha_s\boldsymbol{t} + \alpha g\kappa\boldsymbol{n} + \beta_s\boldsymbol{n} - \beta g\kappa\boldsymbol{t} \\ &= (\alpha_s - \beta g\kappa)\boldsymbol{t} + (\alpha g\kappa + \beta_s)\boldsymbol{n}\end{aligned} \tag{A.10}$$

$$\begin{aligned}\boldsymbol{\gamma}_{st} &= (g\boldsymbol{t})_t \\ &= g_t\boldsymbol{t} + g\boldsymbol{t}_t \\ &= g_t\boldsymbol{t} + g\theta_t\boldsymbol{n}\end{aligned} \tag{A.11}$$

Now, due to symmetry $\boldsymbol{\gamma}_{ts} = \boldsymbol{\gamma}_{st}$

$$g_t\boldsymbol{t} + g\theta_t\boldsymbol{n} = (\alpha_s - \beta g\kappa)\boldsymbol{t} + (\alpha g\kappa + \beta_s)\boldsymbol{n} \tag{A.12}$$

which gives

$$\begin{cases} g_t = \alpha_s - \beta g\kappa \\ g\theta_t = \alpha g\kappa + \beta_s. \end{cases} \tag{A.13}$$

Computing $\boldsymbol{\gamma}_{tt}$,

$$\begin{aligned}\boldsymbol{\gamma}_{tt} &= (\alpha\boldsymbol{t} + \beta\boldsymbol{n})_t \\ &= \alpha_t\boldsymbol{t} + \alpha\boldsymbol{t}_t + \beta_t\boldsymbol{n} + \beta\boldsymbol{n}_t \\ &= (\alpha_t - \beta\theta_t)\boldsymbol{t} + (\alpha\theta_t + \beta_t)\boldsymbol{n}\end{aligned} \tag{A.14}$$

Substituting $g\theta_t$ in $\boldsymbol{\gamma}_{tt}$

$$\boldsymbol{\gamma}_{tt} = (\alpha_t - \beta(\alpha\kappa + \tfrac{\beta_s}{g}))\boldsymbol{t} + (\alpha(\alpha\kappa + \tfrac{\beta_s}{g}) + \beta_t)\boldsymbol{n} \tag{A.15}$$

Now, for $g = 1$ and $g_s = 0$

$$\begin{aligned}\boldsymbol{\gamma}_{ss} &= \kappa\boldsymbol{n} \\ \boldsymbol{\gamma}_{st} &= (\alpha_s - \beta\kappa)\boldsymbol{t} + (\alpha\kappa + \beta_s)\boldsymbol{n} \\ \boldsymbol{\gamma}_{tt} &= (\alpha_t - \beta(\alpha\kappa + \beta_s))\boldsymbol{t} + (\alpha(\alpha\kappa + \beta_s) + \beta_t)\boldsymbol{n}\end{aligned} \tag{A.16}$$

Hence, the proof.

■

Proposition A.3. *Given a 1-parameter family of curves under an arbitrary regular parameterization* $\boldsymbol{\gamma}(s,t)$ *where* $\|\frac{\partial\boldsymbol{\gamma}}{\partial s}(s,0)\| = 1$, *there are two first-order intrinsic measure (invariant to parameterization),*

$$\begin{cases} I &= \frac{\partial\boldsymbol{\gamma}}{\partial s} = \boldsymbol{t} \\ II &= \frac{\partial\boldsymbol{\gamma}}{\partial t}\cdot\boldsymbol{n} \end{cases} \tag{A.17}$$

and three second-order intrinsic measures

$$\begin{cases} III &= \boldsymbol{\gamma}_{ss}\cdot\boldsymbol{n}, \\ IV &= \boldsymbol{\gamma}_{st}\cdot\boldsymbol{n} - \kappa\,\boldsymbol{\gamma}_t\cdot\boldsymbol{t}, \\ V &= (\boldsymbol{\gamma}_{st}\cdot\boldsymbol{n})^2 - \kappa\,\boldsymbol{\gamma}_{tt}\cdot\boldsymbol{n}, \end{cases} \tag{A.18}$$

where $\boldsymbol{t}$ *and* $\boldsymbol{n}$ *represent the unit tangent and normal, respectively. In other words, the remaining degrees of freedom from the first-order derivative* $\frac{\partial\boldsymbol{\gamma}}{\partial t}\cdot\boldsymbol{t}$, *and from the second-order derivatives,* $\frac{\partial^2\boldsymbol{\gamma}}{\partial s\partial t}\cdot\boldsymbol{t}$ *and* $\frac{\partial^2\boldsymbol{\gamma}}{\partial t^2}\cdot\boldsymbol{t}$ *are dependent on the choice of parameterization.*

Proof. Denote the parameter indexing the curves into this family of curves as t. Let $\boldsymbol{\gamma}(s,t)$ and $\boldsymbol{\gamma}_1(w,t)$ be two arbitrary parameterizations with the constraint that for the curve at $t=0$ both s and w are arc-length parameters, , $\|\frac{\partial\boldsymbol{\gamma}_1}{\partial w}(w,0)\| = 1$. Thus, for each time t we can write s as a function of w, ,

$$s = f(w,t). \tag{A.19}$$

Then it is clear that

$$\boldsymbol{\gamma}_1(w,t) = \boldsymbol{\gamma}(f(w,t),t), \tag{A.20}$$

since the traces implied by each coincide. Taking the derivatives of (A.20) with respect to t and w and using the chain rule, we have:

$$\begin{cases} \frac{\partial\boldsymbol{\gamma}_1}{\partial w} &= \frac{\partial\boldsymbol{\gamma}}{\partial s}\frac{\partial f}{\partial w} \quad \text{at } (f(w,t),t) \\ \frac{\partial\boldsymbol{\gamma}_1}{\partial t} &= \frac{\partial\boldsymbol{\gamma}}{\partial s}\frac{\partial f}{\partial t} + \frac{\partial\boldsymbol{\gamma}}{\partial t} \quad \text{at } (f(w,t),t) \end{cases} \tag{A.21}$$

Evaluating now at $t=0$ gives

$$\begin{cases} \frac{\partial\boldsymbol{\gamma}_1}{\partial w} &= \boldsymbol{t}, \text{ and } f_w = 1 \text{ at } t=0 \\ \frac{\partial\boldsymbol{\gamma}_1}{\partial t} &= f_t\,\boldsymbol{t} + \frac{\partial\boldsymbol{\gamma}}{\partial t} \quad \text{at } t=0 \end{cases} \tag{A.22}$$

The first equation implies that

$$\boxed{\frac{\partial\boldsymbol{\gamma}_1}{\partial w} = \frac{\partial\boldsymbol{\gamma}}{\partial s}} \tag{A.23}$$

and

$$f(w,0) = w. \tag{A.24}$$

The second equation implies that the velocity measurements in the two parameterizations, , $\frac{\partial \gamma_1}{\partial t}$ and $\frac{\partial \gamma}{\partial t}$ differ by a tangential component which is arbitrary. Thus, the only invariant measurement is $\frac{\partial \gamma}{\partial t} \cdot \boldsymbol{n}$, ,

$$\boxed{\frac{\partial \gamma_1}{\partial t} \cdot \boldsymbol{n} = \frac{\partial \gamma}{\partial t} \cdot \boldsymbol{n}} \tag{A.25}$$

Now, taking second-derivatives of (A.20), and using $f_w(w,0) = 1$ and $f_{ww}(w,0) = 0$, we have:

$$\begin{cases} \frac{\partial^2 \gamma_1}{\partial w^2} &= \frac{\partial^2 \gamma}{\partial s^2} \left(\frac{\partial f}{\partial w}\right)^2 + \frac{\partial \gamma}{\partial s}\frac{\partial^2 f}{\partial w^2}, \\ \frac{\partial^2 \gamma_1}{\partial w \partial t} &= \frac{\partial^2 \gamma}{\partial s^2}\frac{\partial f}{\partial w}\frac{\partial f}{\partial t} + \frac{\partial \gamma}{\partial s}\frac{\partial^2 f}{\partial w \partial t} + \frac{\partial^2 \gamma}{\partial s \partial t}\frac{\partial f}{\partial w}, \\ \frac{\partial^2 \gamma_1}{\partial t^2} &= \frac{\partial^2 \gamma}{\partial s^2}\left(\frac{\partial f}{\partial t}\right)^2 + 2\frac{\partial^2 \gamma}{\partial s \partial t}\frac{\partial f}{\partial t} + \frac{\partial^2 \gamma}{\partial t^2} + \frac{\partial \gamma}{\partial s}\frac{\partial^2 f}{\partial t^2}. \end{cases} \tag{A.26}$$

Evaluating at $t = 0$ where $f_w(w,0) = 1$ we have

$$\begin{cases} \frac{\partial^2 \gamma_1}{\partial w^2} &= \frac{\partial^2 \gamma}{\partial s^2} = \kappa \boldsymbol{n} \\ \frac{\partial^2 \gamma_1}{\partial w \partial t} &= f_{wt}\, \boldsymbol{t} + f_t\, \kappa \boldsymbol{n} + \frac{\partial^2 \gamma}{\partial s \partial t}, \\ \frac{\partial^2 \gamma_1}{\partial t^2} &= f_{tt}\, \boldsymbol{t} + f_t^2\, \kappa \boldsymbol{n} + 2 f_t \frac{\partial^2 \gamma}{\partial s \partial t} + \frac{\partial^2 \gamma}{\partial t^2}. \end{cases} \tag{A.27}$$

It is clear that γ_{1ww} is invariant to parameterization.

$$\boxed{\frac{\partial^2 \gamma_1}{\partial w^2} = \frac{\partial^2 \gamma}{\partial s^2}} \tag{A.28}$$

It is also clear that since f_{wt} and f_{tt} are arbitrary, the two measurement pairs ($\frac{\partial \gamma_1}{\partial w \partial t}$ and $\frac{\partial^2 \gamma}{\partial s \partial t}$) and ($\frac{\partial^2 \gamma_1}{\partial t^2}$ and $\frac{\partial^2 \gamma}{\partial t^2}$) can differ arbitrarily along the tangent direction. We therefore explore a relation along the normal direction by taking the dot product with $\boldsymbol{n}$, but only need the last two equations since there are no unknowns in the first.

$$\begin{cases} \frac{\partial^2 \gamma_1}{\partial w \partial t} \cdot \boldsymbol{n} &= f_t\, \kappa + \frac{\partial^2 \gamma}{\partial s \partial t} \cdot \boldsymbol{n}, \\ \frac{\partial^2 \gamma_1}{\partial t^2} \cdot \boldsymbol{n} &= f_t^2\, \kappa + 2 f_t \frac{\partial^2 \gamma}{\partial s \partial t} \cdot \boldsymbol{n} + \frac{\partial^2 \gamma}{\partial t^2} \cdot \boldsymbol{n}. \end{cases} \tag{A.29}$$

Observe that $f_t = (\frac{\partial \gamma_1}{\partial t} - \frac{\partial \gamma}{\partial t}) \cdot t$ from Equation A.22. Thus,

$$\frac{\partial^2 \gamma_1}{\partial w \partial t} \cdot \boldsymbol{n} = (\frac{\partial \gamma_1}{\partial t} \cdot \boldsymbol{t} - \frac{\partial \gamma}{\partial t} \cdot \boldsymbol{t})\kappa + \frac{\partial^2 \gamma}{\partial s \partial t} \cdot \boldsymbol{n}, \tag{A.30}$$

or arranging it in a symmetric form,

$$\boxed{\frac{\partial^2 \gamma_1}{\partial w \partial t} \cdot \boldsymbol{n} - (\frac{\partial \gamma_1}{\partial t} \cdot \boldsymbol{t})\, \kappa = \frac{\partial^2 \gamma}{\partial s \partial t} \cdot \boldsymbol{n} - (\frac{\partial \gamma}{\partial t} \cdot \boldsymbol{t})\kappa} \tag{A.31}$$

This is clearly an invariant measure! Similarly, the last equation needs to have f_t substituted in and arranged in a symmetric form. We rewrite the last equation as

$$\frac{\partial^2 \gamma_1}{\partial t^2}\cdot \boldsymbol{n} = f_t(f_t\kappa + \frac{\partial^2\gamma}{\partial s\partial t}\cdot\boldsymbol{n}) + f_t\frac{\partial^2\gamma}{\partial s\partial t}\cdot\boldsymbol{n} + \frac{\partial^2\gamma}{\partial t^2}\cdot\boldsymbol{n} \tag{A.32}$$

$$= f_t(\frac{\partial^2 \gamma_1}{\partial w\partial t}\cdot\boldsymbol{n}) + f_t\frac{\partial^2\gamma}{\partial s\partial t}\cdot\boldsymbol{n} + \frac{\partial^2\gamma}{\partial t^2}\cdot\boldsymbol{n}. \tag{A.33}$$

Now, using $\kappa f_t = \frac{\partial^2\gamma_1}{\partial w\partial t}\cdot\boldsymbol{n} - \frac{\partial^2\gamma}{\partial s\partial t}\cdot\boldsymbol{n}$ from the first equation, we have

$$k\frac{\partial^2 \gamma_1}{\partial t^2}\cdot \boldsymbol{n} = (\frac{\partial^2 \gamma_1}{\partial w\partial t}\cdot\boldsymbol{n} - \frac{\partial^2\gamma}{\partial s\partial t}\cdot\boldsymbol{n})(\frac{\partial^2 \gamma_1}{\partial w\partial t}\cdot\boldsymbol{n} + \frac{\partial^2\gamma}{\partial s\partial t}\cdot\boldsymbol{n}) + \kappa\frac{\partial^2\gamma}{\partial t^2}\cdot\boldsymbol{n} \tag{A.34}$$

$$= \left[(\frac{\partial^2 \gamma_1}{\partial w\partial t}\cdot\boldsymbol{n})^2 - (\frac{\partial^2\gamma}{\partial s\partial t}\cdot\boldsymbol{n})^2\right] + \kappa\frac{\partial^2\gamma}{\partial t}\cdot\boldsymbol{n} \tag{A.35}$$

Arranging this in a symmetric form, we have

$$\boxed{(\frac{\partial^2 \gamma_1}{\partial w\partial t}\cdot\boldsymbol{n})^2 - \kappa\frac{\partial^2\gamma_1}{\partial t^2}\cdot\boldsymbol{n} = (\frac{\partial^2\gamma}{\partial s\partial t}\cdot\boldsymbol{n})^2 - \kappa\frac{\partial^2\gamma}{\partial t}\cdot\boldsymbol{n}}, \tag{A.36}$$

which represents another invariant measurement. ■

Corollary A.4. *(Geometric interpretation of invariant measures)*

$$\begin{cases} I & = \boldsymbol{t} \text{ and } \boldsymbol{n} \\ II & = \beta \\ III & = \kappa \\ IV & = \beta_s \\ V & = \alpha\kappa\beta_s + \beta_s^2 - \kappa\beta_{\boldsymbol{t}} \end{cases} \tag{A.37}$$

Proof. Since $I = \frac{\partial\gamma}{\partial s} = \boldsymbol{t}$. This means $\boldsymbol{n}$ is also invariant as it is perpendicular to $\boldsymbol{t}$. Proposition1.3 gives $\frac{\partial\gamma}{\partial t}\cdot\boldsymbol{n} = \beta$, $\frac{\partial^2\gamma}{\partial s^2}\cdot\boldsymbol{n} = \kappa$, $\frac{\partial^2\gamma}{\partial s\partial t}\cdot\boldsymbol{n} - (\frac{\partial\gamma}{\partial t}\cdot\boldsymbol{t})\kappa = \alpha\kappa + \beta_s - \alpha\kappa = \beta_s$ and

$$V = (\boldsymbol{\gamma}_{st}\cdot\boldsymbol{n})^2 - \kappa\boldsymbol{\gamma}_{tt}\cdot\boldsymbol{n} \tag{A.38}$$

$$= (\alpha\kappa + \beta_s)^2 - \kappa(\alpha^2\kappa + \alpha\beta_s + \beta_t) \tag{A.39}$$

$$= \alpha\kappa\beta_s + \beta_s^2 - \kappa\beta_t \tag{A.40}$$

■

Corollary A.4 gives that β, β_t, κ are parameterization independent and can be computed from the trace. Note that β_t is not an invariant. β_t can be expressed as

$$\kappa\beta_t = \alpha\kappa\beta_s + \beta_s^2 - V$$

Now, for normal parameterization,$\alpha = 0$, $V = \beta_s^2 - \kappa\beta_t^*$. Therefore,

$$\kappa\beta_t = \alpha\kappa\beta_s + \beta_s^2 - \beta_s^2 + \kappa\beta_t^*$$

or,

$$\kappa\beta_t = \alpha\kappa\beta_s + \kappa\beta_t^*$$

This gives us

$$\beta_t = \alpha\beta_s + \beta_t^* \tag{A.41}$$

Since, β_t^* can be measured without knowledge of parameterization.

Bibliography

[1] Vxl - c++ libraries for computer vision. http://vxl.sourceforge.net/.

[2] M. A. Abidi and C. J. Delcroix. Analytic fusion of edge maps. In *Proceedings of IEEE Southeast Conference*, volume 2, pages 739–744, April 1989.

[3] R. Adams and L. Bischof. Seeded region growing. *PAMI*, 16(6):641–647, 1994.

[4] Edward H. Adelson and John Y. A. Wang. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3:625–638, 1994.

[5] G. Adiv. Inherent ambiguities in recovering 3-d motion and structure from a noisy flow field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(5):477–489, 1989.

[6] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 777, Washington, DC, USA, 1995. IEEE Computer Society.

[7] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[8] James R. Bergen, P. An, Th J. Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. pages 237–252. Springer-Verlag, 1992.

[9] Lothar Bergen and Fernand Meyer. Motion segmenation and depth ordering based on morphological segmentation. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II*, pages 531–547, London, UK, 1998. Springer-Verlag.

[10] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT, 1987.

[11] F.L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(6):567–585, Jun 1989.

[12] T.E. Boult and L.G. Brown. Factorization-based segmentation of motions. In *Workshop on Visual Motion*, pages 179–186, 1991.

[13] B.F. Buxton and H. Buxton. Monocular depth perception from optical flow by space time signal processing. B-218:27–47, 1983.

[14] M. Campani and A. Verri. Computing optical flow from an overconstrained system of linear algebraic equations. *Third International Conference on Computer Vision, 1990. Proceedings,*, pages 22–26, Dec 1990.

[15] J. Canny. A computational approach to edge detection. *PAMI*, 8:679–698, 1986.

[16] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *ICCV*, pages 156–162, 1995.

[17] C. H. Chu and E. J. Delp. Estimating displacement vectors from an image sequence. *Journal of the Optical Society of America A*, 6:871–878, June 1989.

[18] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages II:44–51, 2000.

[19] Haili Chui and Anand Rangarajan. A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.*, 89(2-3):114–141, 2003.

[20] Roberto Cipolla and Peter Giblin. *Visual motion of curves and surfaces*. Cambridge University Press, New York, NY, USA, 2000.

[21] R. Clough and J. Tocher. Finite element stiffness matrices fr analysis of pplates in bending. In *In Proc. of Conference on Matrix Methods in Structural Analysis*, 1965.

[22] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of nonrigid objects using mean shift. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2:142–149, Hilton Head Island, South Carolina, 2000.

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

[24] D.E. Crispell, J.L. Mundy, and G. Taubin. Parallax-free registration of aerial video. 2008.

[25] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *IJCV*, pages 167–187, 1993.

[26] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. *Lecture Notes in Mathematics*, 571:85–100, Nov. 1977.

[27] A. Mark Earnshaw and Steven D. Blostein. The performance of camera translation direction estimators from optical flow: Analysis, comparison, and theoretical limits. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:927–932, 1995.

[28] James H. Elder. Are edges incomplete? *Int. J. Comput. Vision*, 34(2-3):97–122, 1999.

[29] C. E. Erdem, A. Tekalp, and B. Sankur. Video object tracking with feedback of performance measures. In *Proc. IEEE conference on Computer Vision and Pattern Recognition*, pages 593–600, Dec 2001.

[30] ”Ricardo Fabbri, Vishal Jain, and Benjamin B. Kimia”. ”using differential geometry of curves and surfaces to solve structure and motion in computer vision”,, ”2008”,.

[31] Olivier Faugeras and Inria Sophia Antipolis. A theory of the motion fields of curves. *International Journal of Computer Vision*, 10:125–156, 1993.

[32] Cornelia Fermuller and Yiannis Aloimonos. Observability of 3d motion. *International Journal of Computer Vision*, 37:2000, 2000.

[33] V. Ferrari, T. Tuytelaars, and L. van Gool. Real-time affine region tracking and coplanar grouping. pages 226–233, Kauai, Hawaii, 2001.

[34] J.D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics Principles and Practice*. Addison-Wesley, 2nd edition, 1996.

[35] F. Folta, L. Van Eycken, and Luc van Gool. Shape extraction using temporal continuity. In *Proc. European Workshop on Image Analysis for Multimedia Interactive Services of the IEEE conference on Computer Vision and Pattern Recognition*, pages 69–74, 1997.

[36] D. Freedman. Effective tracking through tree search. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 25, pages 604–615, May 2003.

[37] C. W. Gear. Multibody grouping from motion images. *Int. J. Comput. Vision*, 29(2):133–150, 1998.

[38] S. Gold, A. Rangarajan, and E Mjolsness. Learning with preknowledge:clustering with point and graph matching distance measures. *Neural Computation*, 8(4):787–804, 1996.

[39] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, 1998.

[40] C.G. Harris. Determination of ego-motion from matched points. *International Journal of Computer Vision*, pages 189–192, 1993.

[41] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[42] E. C. Hildreth. The Computation of the Velocity Field. *Royal Society of London Proceedings Series B*, 221:189–220, April 1984.

[43] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[44] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *PAMI*, 15:850–863, 1993.

[45] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, pages 29:2–28, 1998.

[46] Sumer Jabri, Zoran Duric, Harry Wechsler, and Azriel Rosenfeld. Detection and location of people in video images using adaptive fusion of color and edge information. In *ICPR*, pages 4627–4631, 2000.

[47] Vishal Jain, Benjamin B. Kimia, and Joseph L. Mundy. Background modeling based on subpixel edges. In *IEEE International Conference on Image Processing*, volume IV, pages 321–324, San Antonio, TX, USA, September 2007. IEEE.

[48] Vishal Jain, Benjamin B. Kimia, and Joseph L. Mundy. Segregation of moving objects using elastic matching. *Computer Vision and Image Understanding*, 108:230–242, 2007.

[49] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *Motion02*, pages 22–27.

[50] Allan D. Jepson and David J. Heeger. Linear subspace methods for recovering translational direction. In *Spatial Vision in Humans and Robots*, pages 39–62. Cambridge University Press, 1993.

[51] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proc. on Advanced Video Based Surveillance Systems*, Sept 2001.

[52] K. Kanatani. Unbiased estimation and statistical analysis of 3-d rigid motion from two views. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(1):37–50, 1993.

[53] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[54] Satyanad Kichenassamy, Arun Kumar, Peter J. Olver, Allen Tannenbaum, and Anthony J. Yezzi. A geometric snake model for segmentation of medical imagery. *IEEE Trans. Med. Imaging*, 16(2):199–209, 1997.

[55] Changick Kim and Jenq-Neng Hwang. Video object extraction for object-oriented applications. *J. VLSI Signal Process. Syst.*, 29(1-2):7–21, 2001.

[56] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proceedings of the Third European Conference on Computer Vision*, volume I. Springer Verlag, 1994.

[57] Dar-Shyang Lee, Jonathan J. Hull, and Berna Erol. A Bayesian framework for Gaussian mixture background modeling. In *Proc. ICIP*, 2003.

[58] T. Lindenberg. Feature detection with automatic scale detection. *IJCV*, 30(2):77–116, 1998.

[59] Haiying Liu, Rama Chellappa, and Azriel Rosenfeld. A hierarchical approach for obtaining structure from two-frame optical flow. In *MOTION '02: Proceedings of the Workshop on Motion and Video Computing*, page 214, Washington, DC, USA, 2002. IEEE Computer Society.

[60] H.C. Longuet Higgins and K. Prazdny. The interpretation of a moving retinal image. B-208:385–397, 1980.

[61] Bruce D. Lucas. *Generalized Image Matching by the Method of Differences*. PhD thesis, Robotics Institute, Carnegie Mellon University, July 1984.

[62] W.J. MacLean. Removal of translation bias when using subspace methods. *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, 2:753–758 vol.2, 1999.

[63] Nicolas Martel-Brisson and Andre Zaccarin. Moving cast shadow detection from a gaussian mixture shadow model. In *CVPR*, pages 643–648. IEEE Computer Society, 2005.

[64] H. P. Moravec. Visual mapping by a robot rover. In *Proc. of the 6th International Joint Conference on Artificial Intelligence*, pages 598–600, 1979.

[65] G. Mori, Xiaofeng Ren, A.A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:II–326–II–333 Vol.2, June-2 July 2004.

[66] Fabrice Moscheni, Sushil Bhattacharjee, and Murat Kunt. Spatiotemporal segmentation based on region merging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(9):897–915, 1998.

[67] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(2):220–228, 1987.

[68] P. Muse, F. Sur, F. Cao, Y. Gousseau, and J.M. Morel. An a contrario decision method for shape element recognition. 69(3):295–315, September 2006.

[69] H.-H. Nagel. On a constraint equation for the estimation of displacement rates in image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(1):13–30, 1989.

[70] Shobhit Niranjan, Gaurav Gupta, Amitabha Mukerjee, and Sumana Gupta. Efficient registration of aerial image sequences without camera priors. In *ACCV (2)*, pages 394–403, 2007.

[71] N. Paragios and R. Deriche. A PDE-based level set approach for detection and tracking of moving objects. In *Proc. International Conference on Computer Vision*, Bombay,India, Jan 1998.

[72] T. Pollard and J.L. Mundy. Change detection in a 3-d world. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–6, June 2007.

[73] POVRAY. Persistence of vision pty. ltd, persistence of vision raytrace, 2004.

[74] P. W. Power and J.A. Schoonees. Understanding background mixture models for foreground segmentation. *Imaging and Vision Computing New Zealand*, 2002.

[75] X. Ren and J. Malik. Learning a classification model for segmentation. *Proceedings. Ninth IEEE International Conference on Computer Vision, 2003.*, pages 10–17 vol.1, Oct. 2003.

[76] C. Rothwell, J. Mundy, W. Hoffman, and V.-D. Nguyen. Driving vision by topology. In *ISCV*, pages 395–400, 1995.

[77] Thomas Sebastian, Philip Klein, and Benjamin Kimia. Curve matching using alignment curve. Technical Report LEMS 184, LEMS, Brown University, June 2000.

[78] Thomas Sebastian, Philip Klein, and Benjamin Kimia. On aligning curves. *PAMI*, 25(1):116–125, January 2003.

[79] H. Sekkati and A. Mitiche. Joint optical flow estimation, segmentation, and 3d interpretation with level sets. *Computer Vision and Image Understanding*, 103(2):89 – 100, 2006.

[80] Daniel Sharvit, Jacky Chan, Hüseyin Tek, and Benjamin B. Kimia. Symmetry-based indexing of image databases. *Journal of Visual Communication and Image Representation*, 9(4):366–380, December 1998.

[81] J. Shi and C. Tomasi. Good features to track. In *Proc. of the IEEE conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[82] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 1154, Washington, DC, USA, 1998. IEEE Computer Society.

[83] Kaleem Siddiqi, Kaleem Siddiqi, Benjamin B. Kimia, Benjamin B. Kimia, Chi wang Shu, and Chi wang Shu. Geometric shock-capturing eno schemes for subpixel interpolation, computation and curve evolution. In *Graphical Models and Image Processing*, pages 278–301, 1997.

[84] E.P. Simoncelli, E.H. Adelson, and D.J. Heeger. Probability distributions of optical flow. *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pages 310–315, Jun 1991.

[85] Paul Smith, Tom Drummond, and Roberto Cipolla. Layered motion segmentation and depth ordering by tracking edges. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(4):479–494, 2004.

[86] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE TRANS. PAMI*, 22(8):747–757, August 2000.

[87] C.V. Stewart, Chia-Ling Tsai, and B. Roysam. The dual-bootstrap iterative closest point algorithm with application to retinal image registration. *Medical Imaging, IEEE Transactions on*, 22(11):1379–1394, Nov. 2003.

[88] A. Tamrakar and B.B. Kimia. No grouping left behind: From edges to curve fragments. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007.

[89] Amir Tamrakar. Image contour extraction using geometric consistency. Ph.D. dissertation, Division Of Engineering, Brown University, Providence, RI, 02912, September 2008.

[90] Amir Tamrakar and Benjamin B. Kimia. Medial visual fragments as an intermediate image representation for segmentation and perceptual grouping. In *Proceedings of CVPR Workshop on Perceptual Organization in Computer Vision*, page 47, 2004.

[91] Amir Tamrakar and Benjamin B. Kimia. Combinatorial grouping of edges using geometric consistency in a lagrangian framework. In *Proceedings of IEEE Workshop on Perceptual Organization in Computer Vision, POCV*, pages 189–197, 2006.

[92] Amir Tamrakar and Benjamin B. Kimia. No grouping left behind: From edges to curve fragments. Rio de Janeiro, Brazil, October 2007.

[93] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9(2):137–154, 1992.

[94] David Tweed and Andrew Calway. Integrated segmentation and depth ordering of motion layers in image sequences. In *Image and Vision Computing*, page 00. Press, 2000.

[95] R. Vidal, Yi Ma, and J. Piazzi. A new gpca algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 1:I–510–I–517 Vol.1, June-2 July 2004.

[96] R. Vidal, Yi Ma, and S. Sastry. Generalized principal component analysis (gpca). *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1:I–621–I–628 vol.1, June 2003.

[97] Allen M. Waxman and Kwangyoen Wohn. Contour evolution, neighborhood deformation and image flow: textured surfaces in motion. pages 72–98, 1987.

[98] Joseph Weber, Jitendra Malik, S. Devadas, and P. Michel. Robust computation of optical flow in a multi-scale differential framework. *International Journal of Computer Vision*, 14:12–20, 1995.

[99] J. Weng, Thomas S. Huang, and Narendra Ahuja. 3-D motion estimation, understanding, and prediction from noisy image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):370–389, 1987.

[100] Juyang Weng, N. Ahuja, and T.S. Huang. Two-view matching. *Second International Conference on Computer Vision.*, pages 64–73, Dec 1988.

[101] Y. Wexler and A. Shashua. Q-warping: Direct computation of quadratic reference surfaces. In *In Proc. CVPR*, pages 333–338, 1999.

[102] J. Yang and R. S. Blum. Multi-frame image fusion using the expectation-maximization algorithm. In *8th International Conference on Information Fusion*, volume 1, July 2005.

[103] Yee-Hong Yang and Martin D. Levine. The background primal sketch: An approach for tracking moving objects. *MVA*, 5(1):17–34, December 1992.

[104] David A. Yocky. Image merging and data fusion by means of the discrete two-dimensional wavelet transform. *Journal of Optical Society of America*, 12(9):1834–1845, 1995.

[105] Laurent Younes. Computable elastic distance between shapes. *SIAM Journal of Applied Mathematics*, 58:565–586, 1998.