

GOAL-ORIENTED EDGE DETECTION

Ph.D. THESIS

Binnur KURT, M.Sc.

Department : Computer Engineering

Programme : Computer Engineering

OCTOBER 2007

AMACA YÖNELİK AYRIT SAPTAMA

DOKTORA TEZİ

Y. Müh. Binnur KURT

(504972018)

Anabilim Dalı : Bilgisayar Mühendisliği

Programı : Bilgisayar Mühendisliği

EKİM 2007

ACKNOWLEDGEMENTS

The accomplishments in this work were made possible by the help and support of many individuals. First, and foremost, I would like to express my sincere gratitude to my advisor, Professor Muhittin Gökmen, for his continuous guidance, suggestions, patience, and friendship during the course of my work. He has contributed tremendously to the technical content of this dissertation. Perhaps more importantly, he has provided me a perfect example by establishing a productive and enjoyable advisor-student relationship.

I want to thank the members of my dissertation committee for a careful evaluation of my work, and their useful feedback which has significantly improved the quality of this document.

I would like to thank to the members of Computer Vision and Image Processing Lab. at Istanbul chapter 2 Technical University, Abdulkerim Çapar and Fatih Kahraman for their contributions. I would like to thank to Abdulkerim Çapar also for the contributions to Chapter 4.

I would like to thank to Hasan Tahsin Demiral from Omega Information Technologies Training and Consultancy Company for the contributions to Chapter 3.

And finally, I thank my parents for their support throughout my PhD study. I could not finish the degree without their encouragement. I dedicate the study to Mom and Dad, who have always loved and supported me.

July 2007

Binnur KURT

ABBREVIATIONS

1D	: One Dimensional
2D	: Two Dimensional
3D	: Three Dimensional
DoG	: Difference of Gaussian
FD	: Fourier Descriptor
FOM	: Figure Of Merit
GED	: Generalized Edge Detector
LoG	: Laplacian of Gaussian
LPD	: License Plate Detection
LPL	: License Plate Localization
LPR	: License Plate Recognition
MPEG	: Moving Pictures Expert Group
MSE	: Mean Square Error
MSD	: Minimum Square Distance
NMSE	: Normalized Mean Square Error
PDE	: Partial Differential Equation
ROC	: Receiver Operator Characteristics
SNR	: Signal to Noise Ratio
SVM	: Support Vector Machine

SYMBOLS

∇	: Gradient Operator
Δ	: Laplace Operator
$\mathbb{H}$	: Hessian Operator
$\mathbb{R}$	: Real Number
λ	: Smoothing parameter
τ	: Continuity control parameter
T_{low}	: Low threshold in hysteresis thresholding
T_{high}	: High threshold in hysteresis thresholding
η	: Tangent Vector
ξ	: Normal Vector
Z	: Z Transform
tp	: True Positives
fp	: False Positives
ugt	: Unmatched Ground Truth

CONTENTS

	<u>Page No</u>
ACKNOWLEDGEMENTS	iii
ABBREVIATIONS	iv
SYMBOLS	v
TABLE LIST	x
FIGURE LIST	xiv
SUMMARY	xv
ÖZET	xx
1 INTRODUCTION	1
1.1 Goal of the Thesis	1
1.2 Survey of Edge Detectors	3
1.3 Contributions of the Thesis	13
1.4 Software and Data Availability	14
1.5 Organization of the Thesis	14
2 REGULARIZED LINEAR SOLUTION: λ_τ -SPACE	15
2.1 Derivation of Two Dimensional R - and G -Filter	16
2.1.1 Case I: $\Delta > 0$	18
2.1.2 Case II: $\Delta = 0$	20
2.1.3 Case III: $\Delta < 0$	22
2.1.4 Case IV: $A = 0$	25
2.1.5 Case V: $B = 0$	26

2.2	Smoothing Filter: R -Filter	27
2.3	Edge Detection Filter: G -Filter	31
2.4	Serial Implementation	33
2.5	Recursive Implementation	38
2.6	Isotropic Diffusion and Recursive Implementation	42
2.7	Evaluation Based on Empirical ROC Curves	43
2.8	Conclusion	47
3	REGULARIZED NON-LINEAR SOLUTION: ANISOTROPIC $\lambda\tau$ -SPACE	52
3.1	Introduction	52
3.2	Overview	53
3.3	Adaptive Scheme	54
3.4	Implementation and Results	57
3.5	Summary and Conclusions	63
4	GRADIENT-BASED SHAPE DESCRIPTOR	66
4.1	Directional Gradient Extraction Using Steerable Filters	68
4.2	Affine Invariant Shape Descriptor	73
4.3	Experimental Results	75
4.4	Conclusion	82
5	GOAL-ORIENTED EDGE DETECTION	83
5.1	Stage I: Extracting Feature Vector	84
5.2	Stage II: Contour Matching	86
6	SUMMARY AND CONCLUSION	91
Appendix A	PROOF OF CIRCULAR-SHIFTING PROPERTY IN FOURIER DOMAIN	97
Appendix B	RECOGNITION RESULTS FOR GRADIENT BASED DESCRIPTORS	98
Appendix C	GOAL-ORIENTED EDGE DETECTION EXPERIMENTAL RESULTS	102

LIST OF TABLES

	<u>Page No</u>
Table 2.1. Two Dimensional $R(x, y; \lambda, \tau)$ -filters.	27
Table 2.2. Two Dimensional Discrete $R(m, n)$ -filters.	28
Table 2.3. Values of s_1 and s_2 for five cases.	28
Table 2.4. Filter types associated with the selected parameter values.	29
Table 2.5. Two Dimensional G-filter along x direction : $G^{(x)}(x, y; \lambda, \tau)$	30
Table 2.6. Two Dimensional Discrete G-filter along x direction : $G^{(x)}(m, n; \lambda, \tau)$. ..	31
Table 2.7. The normalized mean square error (NMSE) between the reconstructed images and the original image.	33
Table 2.8. Quantitative evaluation of edges from the checkerboard image.	36
Table 2.9. Transfer function $H(z_1, z_2)$ coefficients.	40
Table 2.10. Optimum scale parameters with respect to mean square error (mse) between the smoothed image and the original image.	45
Table 2.11. Optimum detector parameters $(\lambda^*, \tau^*, T_{low}^*, T_{high}^*)$ with respect to true positives (tp) computed between the detected edges and the ground truth edges. fp denotes <i>false positives</i> and ugt denotes <i>unmatched ground truth</i>	47
Table 3.1. Quantitative results for noisy checkerboard images.	58
Table 3.2. Quantitative results for reconstructed noisy checkerboard images.	58
Table 3.3. SNR values for the reconstructed <i>Parrot</i> image.	62
Table 3.4. SNR values for the reconstructed images.	62
Table 4.1. Rotation angle estimation error $\Theta()$ for various M	75
Table 4.2. Recognition rate with respect to descriptor size (M, L)	77
Table 4.3. Recognition rate with respect to descriptor size (M, L) obtained for the license plate digits where poorly detected edges are used.	79
Table 4.4. Recognition rate obtained for the descriptor $\tilde{H}$	80
Table 4.5. Recognition rate with respect to filter scale.	80

Table 4.6. Average distances between the object and its 1° rotated versions (Binary object results are given in the first row, and Grayscale object results are given in the second row).	80
Table 4.7. Recognition rates for the MPEG-7 Core Experiments Shape data set with respect to rank and descriptor size.	81
Table B.1. The effect of training size on recognition rates for the MPEG-7 shape data set.	100
Table B.2. Recognition performance for the Kimia Data Set with respect to rank and descriptor size. Results are given in terms of the number of correctly retrieved shapes.	101
Table B.3. Recognition rates for the occluded objects that are synthetically produced from the MPEG-7 Core Experiments shape data set at various occlusion rates.	101

LIST OF FIGURES

	<u>Page No</u>
Figure 1.1 : (a) Two sample images from two prominent computer vision applications and (b) their “ideal” edges.	3
Figure 1.2 : $\lambda\tau$ -space representation of edges ($\lambda = 1.0, 2.5, 5.0, 10.0, 15.0, 20.0, 25.0, 50.0, 75.0, \tau = 0.0, 0.25, 0.5, 0.75$)	4
Figure 1.3 : $\lambda\tau$ -space representation of edges ($\lambda = 1.0, 2.5, 5.0, 10.0, 15.0, 20.0, 25.0, 50.0, 75.0, \tau = 0.0, 0.25, 0.5, 0.75$)	5
Figure 1.4 : Edge detector space.	6
Figure 1.5 : Sobel edge operators: (a) G_x , (b) G_y	6
Figure 1.6 : Prewitt edge operators at various directions.	7
Figure 1.7 : Laplacian edge operator for a 4-neighborhood system.	7
Figure 1.8 : Laplacian of Gaussian: (a) 3D view (b) Cross-section of (a) at $y = 0$	8
Figure 2.1 : The geometry of level set for R -filters	29
Figure 2.2 : R -filters (τ takes the values 0.0, 0.5, and 1.0 from left to right).	29
Figure 2.3 : $\lambda\tau/(1 - \tau)^2 = \text{constant}$ curves in $\lambda\tau$ -space.	30
Figure 2.4 : G -filters (τ takes the values 0.0, 0.5, and 1.0 from left to right).	32
Figure 2.5 : The noisy bars and checkerboard images (SNR=12dB)	33
Figure 2.6 : $\lambda\tau$ -space Representation of noisy checkerboard image (τ takes the values 0.0, 0.5, and 1.0 from left to right).	34
Figure 2.7 : $\lambda\tau$ -space representation of edges (τ takes the values 0.0, 0.5, and 1.0 from left to right).	35
Figure 2.8 : The original and noisy (SNR=2.4dB) house images	37
Figure 2.9 : $\lambda\tau$ -space Representation of edges for the noisy house image (τ takes the values 0.0, 0.5, and 1.0 from left to right).	37
Figure 2.10 : Architecture of serial implementation.	39
Figure 2.11 : Four regions used to partition the space.	39
Figure 2.12 : Architecture of recursive implementation.	41

Figure 2.13 : Scan order of transfer functions in recursive implementation for Region I (a), Region II (b), Region III (c), and Region IV (d).	41
Figure 2.14 : Search space for the Generalized Edge Detector.	44
Figure 2.15 : Search trajectories for the bars images with various SNR values.	44
Figure 2.16 : Search trajectories for the checkerboard images.	45
Figure 2.17 : <i>nmse</i> surfaces for the bars images with various SNR values.	46
Figure 2.18 : <i>nmse</i> surfaces for the checkerboard images.	46
Figure 2.19 : Florida data set and its ground truth edges [43].	48
Figure 2.20 : Edges detected with optimum parameters $(\lambda^*, \tau^*, T_{low}^*, T_{high}^*)$ for the images in Florida data set.	49
Figure 2.21 : (a) <i>True positives (tp)</i> , (b) <i>False positives (fp)</i> and (c) <i>Unmatched ground truth (ugt)</i> plots obtained for various images in Florida data set. In the plots, the x -axis is $\lambda \in [0.1..1.0]$ and the y -axis is $\tau \in [0.0..1.0]$	51
Figure 3.1 : The original and noisy checkerboard images.	58
Figure 3.2 : Restored checkerboard images and their edges.	59
Figure 3.3 : (a) Original Airfield and bridge images (b) Degraded Airfield and Bridge images with Gaussian noise $\sigma^2 = 451.36$ and $\sigma^2 = 375.73$	59
Figure 3.4 : Reconstruction and edge detection results for the original Airfield image (a) linear R - and G -filter results (b) Results obtained by the proposed adaptive representation.	60
Figure 3.5 : Reconstruction and edge detection results for the noisy Airfield image (a) linear R - and G -filter results (b) Results obtained by the proposed adaptive representation.	61
Figure 3.6 : Reconstruction and edge detection results for the original Bridge image (a) linear R - and G -filter results (b) Results obtained by the proposed adaptive representation.	61
Figure 3.7 : Reconstruction and edge detection results for the noisy Bridge image (a) linear R - and G -filter results (b) Results obtained by the proposed adaptive representation.	62
Figure 3.8 : Linear $\lambda\tau$ -space representation of <i>Parrot</i> image.	63
Figure 3.9 : Adaptive smoothing of <i>Parrot</i> image.	63
Figure 3.10 : Adaptive smoothing of real images (a) <i>Hat</i> image (b) <i>House</i> image (c) <i>Mandril</i> image.	64
Figure 3.11 : The original and noisy <i>Lenna</i> image.	64
Figure 3.12 : Adaptive smoothing of noisy <i>Lenna</i> image.	65

Figure 4.1 : (a) Sample digit “5”, (b) Filter responses on object boundary pixels for each direction, (c) Steerable filter response at one pixel on the curve with respect to steering angle, which corresponds to a vertical cross section of (b). (d) 90 degrees rotated digit (e) Filter responses on object boundary pixels for each angle, (f) Steerable filter response at the same pixel on the curve with respect to rotation angle, which corresponds to a horizontal cross section of (e).	71
Figure 4.2 : Typical segmented license plate characters selected from the database. .	76
Figure 4.3 : Recognition rates for four methods, the proposed descriptor ($M = 8$), centroid distance, boundary curvature, complex coordinates with respect to L where L is the number of Fourier coefficients used.	78
Figure 4.4 : Poorly segmented license plate characters that are selected from the database.	78
Figure 4.5 : Distance error between the object and its rotated versions for $M = 2, 4, 5, 6, 8, 10, 12, 14, 16$	80
Figure 5.1 : Labeling pixels as edge and non-edge pixels (a) Original checkerboard image, (b) Noisy checkerboard image (c) Marked pixels (black pixels are edge points, gray pixels are non-edge points (d) Another marking used in training.	85
Figure 5.2 : $\lambda\tau$ –space sampling scheme.	86
Figure 5.3 : Feature vector scatter diagrams with respect to different $\lambda\tau$ –space sampling schemes when the labels given in Fig. 5.1(c) is used. (a) ($\lambda_1 = 1.0, \tau_1 = 0.5$), ($\lambda_2 = 8.0, \tau_1 = 0.5$), (b) ($\lambda_1 = 1.0, \tau_1 = 0.5$), ($\lambda_2 = 16.0, \tau_1 = 0.5$), (c) ($\lambda_1 = 8.0, \tau_1 = 0.5$), ($\lambda_2 = 16.0, \tau_1 = 0.5$), (d) ($\lambda_1 = 8.0, \tau_1 = 0.5$), ($\lambda_2 = 32.0, \tau_1 = 0.5$), (e) ($\lambda_1 = 1.0, \tau_1 = 0.5$), ($\lambda_2 = 32.0, \tau_1 = 0.5$), (f) ($\lambda_1 = 32.0, \tau_1 = 0.5$), ($\lambda_2 = 64.0, \tau_1 = 0.5$).	87
Figure 5.4 : Feature vector scatter diagrams with respect to different $\lambda\tau$ –space sampling schemes when the labels given in Fig. 5.1(d) is used. (a) ($\lambda_1 = 1.0, \tau_1 = 0.5$), ($\lambda_2 = 8.0, \tau_1 = 0.5$), (b) ($\lambda_1 = 1.0, \tau_1 = 0.5$), ($\lambda_2 = 16.0, \tau_1 = 0.5$), (c) ($\lambda_1 = 8.0, \tau_1 = 0.5$), ($\lambda_2 = 16.0, \tau_1 = 0.5$), (d) ($\lambda_1 = 8.0, \tau_1 = 0.5$), ($\lambda_2 = 32.0, \tau_1 = 0.5$), (e) ($\lambda_1 = 1.0, \tau_1 = 0.5$), ($\lambda_2 = 32.0, \tau_1 = 0.5$), (f) ($\lambda_1 = 32.0, \tau_1 = 0.5$), ($\lambda_2 = 64.0, \tau_1 = 0.5$).	88
Figure 5.5 : Edge detection results (a) Vehicle image (b) Detected edges (c) Detected contours.	90
Figure B.1 : Content-based image retrieval performance of the proposed method. Each line contains a query where the first image is the query image and the remaining images are the query results.	98
Figure B.2 : CBIR performance of the proposed method under occlusion. Each line contains a query where the first image is the occluded query image and the remaining images are the query results.	99

Figure C.1 : Checkerboard images (a) Original checkerboard image, (b) Noisy checkerboard image (25dB) used in training (c) Noisy checkerboard image (18dB) used in testing.	102
Figure C.2 : Bars images (a) Original Bars image, Noisy bars images (b) 25dB, (c) 22dB, (d) 20dB, (e) 17dB, (f) 14dB, (g) 11dB.	102
Figure C.3 : Edge detection test results for the noisy checkerboard image (18dB). The labels given in Fig. 5.1(c) is used. The following $\lambda\tau$ -space sampling schemes are used: (a) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 8.0, \tau_1 = 0.5)$, (b) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 16.0, \tau_1 = 0.5)$, (c) $(\lambda_1 = 8.0, \tau_1 = 0.5)$, $(\lambda_2 = 16.0, \tau_1 = 0.5)$, (d) $(\lambda_1 = 8.0, \tau_1 = 0.5)$, $(\lambda_2 = 32.0, \tau_1 = 0.5)$, (e) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 32.0, \tau_1 = 0.5)$, (f) $(\lambda_1 = 32.0, \tau_1 = 0.5)$, $(\lambda_2 = 64.0, \tau_1 = 0.5)$	103
Figure C.4 : Edge detection test results for the noisy checkerboard image (18dB). The labels given in Fig. 5.1(d) is used. The following $\lambda\tau$ -space sampling schemes are used: (a) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 8.0, \tau_1 = 0.5)$, (b) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 16.0, \tau_1 = 0.5)$, (c) $(\lambda_1 = 8.0, \tau_1 = 0.5)$, $(\lambda_2 = 16.0, \tau_1 = 0.5)$, (d) $(\lambda_1 = 8.0, \tau_1 = 0.5)$, $(\lambda_2 = 32.0, \tau_1 = 0.5)$, (e) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 32.0, \tau_1 = 0.5)$, (f) $(\lambda_1 = 32.0, \tau_1 = 0.5)$, $(\lambda_2 = 64.0, \tau_1 = 0.5)$	104
Figure C.5 : Edge detection test results for the noisy bars images. The labels given in Fig. 5.1(c) is used.	105
Figure C.6 : Edge detection test results for the noisy bars images. The labels given in Fig. 5.1(d) is used.	106
Figure C.7 : Feature vector scatter diagrams with respect to different $\lambda\tau$ -space sampling schemes when the labels given in Fig. 5.1(c) is used. (a) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 2.0, \tau_1 = 0.5)$, $(\lambda_3 = 4.0, \tau_1 = 0.5)$, (b) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 4.0, \tau_1 = 0.5)$, $(\lambda_3 = 8.0, \tau_1 = 0.5)$	107
Figure C.8 : Feature vector scatter diagrams with respect to different $\lambda\tau$ -space sampling schemes when the labels given in Fig. 5.1(d) is used. (a) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 2.0, \tau_1 = 0.5)$, $(\lambda_3 = 4.0, \tau_1 = 0.5)$, (b) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 4.0, \tau_1 = 0.5)$, $(\lambda_3 = 8.0, \tau_1 = 0.5)$	108
Figure C.9 : Edge detection test results for the noisy bars images. Feature space is three dimensional. The labels given in Fig. 5.1(c) is used.	109
Figure C.10 : Edge detection test results for the noisy bars images. Feature space is three dimensional. The labels given in Fig. 5.1(d) is used.	110
Figure C.11 : Edge detection test results for the noisy checkerboard image (18dB). The labels given in Fig. 5.1(c) is used in (a) and (b). The labels given in Fig. 5.1(d) is used in (c) and (d). The following $\lambda\tau$ -space sampling schemes are used: (a),(c) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 2.0, \tau_1 = 0.5)$, $(\lambda_3 = 4.0, \tau_1 = 0.5)$, (b),(d) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 4.0, \tau_1 = 0.5)$, $(\lambda_3 = 8.0, \tau_1 = 0.5)$	111
Figure D.1 : License plate digits selected from the database.	112
Figure D.2 : Labeled contours that belong to the license plate digits selected from the database.	113

GOAL-ORIENTED EDGE DETECTION

SUMMARY

The aim of edge detection is to provide a meaningful description of object boundaries in a scene from intensity surface. These boundaries are due to discontinuities manifesting themselves as sharp variations in image intensities. There are different sources for sharp changes in images which are created by structure (e.g. texture, occlusion) or illumination (e.g. shadows, highlights). Extracting edges from a still image is certainly the most significant stage of any computer vision algorithm requiring high accuracy of location in the presence of noise. In many contour-based vision algorithms, such as shape-based query, curved-based stereo vision, contour-based image compression and edge-based target recognition, edge-based face detection their performance is highly dependent on the quality of the detected edges. Therefore, edge detection is an important area of research in computer vision. Despite considerable work and progress made on this subject, edge detection is still a challenging research problem due to the lack of a robust and efficient general purpose algorithm.

Any edge detector should tackle with the tradeoff between good localization property forcing the location of the detected edges to be close as much as possible to the real edges and good noise rejection property forcing the intensity surface to be smooth. Without a priori assumption, one can not select the best tradeoff. In fact, deciding whether a pixel belongs to a contour is an ill-posed problem. The detection of sharp changes in image intensity requires the computation of derivatives of the noisy image at different orders. As known, the numerical computation of derivatives of the noisy data is an unstable process since it amplifies the noise. To overcome the noise problem, edge detection algorithms first employ a noise suppression process prior to the differentiation operation. This can be performed by smoothing the noisy image by a low-pass filter. Most of the efforts in edge detection have been devoted to the development of an optimum edge detector which can resolve the tradeoff. Furthermore, extracting edges at different scales and combining these edges have attracted a substantial amount of interest. In the course of developing optimum edge detectors that can resolve the tradeoff between localization and detection performances, several different approaches have resulted in either a Gaussian filter or a filter whose shape is very similar to a Gaussian. Furthermore, these filters are very suitable for obtaining scale space edge detection since the scale of the filter can be easily controlled by means of a single parameter. Although these filters are used very widely, it is very difficult to claim that they can provide the desired output for any specific problem. For instance, there are some cases where the improved localization performance is the primary requirement. In these cases, a sub-optimum filter which promotes the localization performance becomes more appropriate.

In this thesis, we extend powerful smoothing and edge filters (R - and G -filters) known as $\lambda\tau$ -space representation to two dimensional space. G -filter is called Generalized Edge Detector which is capable of producing most of the existing edge detectors. It has been shown that the first order R -filter can deliver improved results on checkerboard and bar images as well as some real images for moderate values of signal-to-noise ratio (SNR).

The original problem was formulated on two dimensional hybrid model comprised of the linear combination of membrane and thin-plate functionals:

$$E_m(f) = \iint (f(x, y) - d(x, y))^2 + \lambda (f_x^2(x, y) + f_y^2(x, y)) d\Omega$$

$$E_p(f) = \iint (f(x, y) - d(x, y))^2 + \lambda (f_{xx}^2(x, y) + 2f_{xy}^2(x, y) + f_{yy}^2(x, y)) d\Omega$$

where $E_m(f)$ is the energy functional associated with membrane and $E_p(f)$ is the energy functional associated with thin plate. Hybrid energy functional is defined as

$$E_h(f) = \iint (f(x, y) - d(x, y))^2 + \lambda \left[\begin{aligned} &(1 - \tau) (f_x^2(x, y) + f_y^2(x, y)) + \\ &\tau (f_{xx}^2(x, y) + 2f_{xy}^2(x, y) + f_{yy}^2(x, y)) \end{aligned} \right] d\Omega$$

where λ is the real-valued regularization parameter and $\tau \in [0, 1]$ is the real-valued continuity control parameter. Here, the hybrid functional is used to obtain a smooth transition from the membrane model to the plate model. In these functionals, the first term on the right hand side is a measure of closeness of the solution $f(x, y)$ to the data $d(x, y)$, and the second term, called stabilizer, is a measure of the smoothness. The compromise between these two terms is controlled by the regularization parameter λ . Smoothing problem then reduced to the solution of two dimensional partial differential equation (PDE):

$$f(x, y) - B(f_{xx}(x, y) + f_{yy}(x, y)) + A(f_{xxxx}(x, y) + 2f_{xxyy}(x, y) + f_{yyyy}(x, y)) = d(x, y).$$

For any given $d(x, y)$, the solution $f(x, y)$ can be easily determined by taking the convolution as follows

$$f(x, y) = d(x, y) * R(x, y).$$

In the original study, the filters were obtained for one dimensional case assuming a separable solution. Nevertheless the proposed solution does not satisfy the 2D PDE. We have developed a solution to the 2D PDE. Two dimensional extension of the representation is important because of that the behaviour of images and edges in the space is best modeled by two dimensional smoothing and edge detector filters. Also since GED filters encompass most of the well-known edge detectors, two dimensional version of these filters could be obtained. The derived filters are more robust to noise when compared to the previous one dimensional filtering scheme in the sense of FoM (Figure of Merit) and false alarm characteristics. There are several parameters to tune when dealing with edge detectors. Usually there is no easy way to find the optimal edge detector parameters for an image. In fact, one set of optimal parameters may no longer be optimal for another image. In this thesis, we find optimal GED parameters using receiver operator characteristics (ROC)

for an image when its ideal edges are available using exhaustive search to see how best it achieves.

Unfortunately, computation time in serial implementation depends on scale parameters (λ and τ). Since filter size increases with increasing λ , convolution requires more multiplication per pixel. We have implemented recursive filtering approach to convolution with two dimensional R - and G -Filters. Filtering is performed in constant time now with this recursive implementation. In this study, we also interpreted the recursive implementation and revealed its relation with the forth-order isotropic diffusion equation.

Image representation aims at producing images with different details. This is one of the most challenging issue in computer vision and image processing due to lack of a robust representation. In general, representations can be classified as linear and nonlinear. Linear representations are constructed by convolving the image by linear filters (kernels) with varying scales. For instance, in classical scale-space the kernel is a Gaussian and the scale-space representation is obtained either by convolving the image by a Gaussian with increasing standard deviation or equivalently by solving the linear heat equation in time. This representation is causal, since the isotropic heat equation satisfies a maximum principle. However, the Gaussian scale-space suffers from serious drawbacks such as over-smoothing and location uncertainty along edges at large scales due to interactions between nearby edges and displacements. Smoothing is done in a homogeneous fashion at the same rate in all directions. This results in blurring of edges at the same rate as smoothing of object interiors. Such a filter results in poor edge localization. Although $\lambda\tau$ -space presents richer representation, it suffers from the same problem. Space parameters are constant throughout the image space causing the edges to be blurred while noise is removed elsewhere. This problem is due to the linear nature of the representation. To overcome this problem, we present an anisotropic $\lambda\tau$ -space image and edge representation allowing the smoothing parameter λ to vary through the image respecting the feature directions. We have showed experimentally that our proposed scheme is capable of locating the feature direction robustly when both terms in the hybrid diffusion are used as contrary to the Perona-Malik equation. Using this approach one may obtain surfaces having different characteristics for the choice of different τ .

Shape representation and description have been playing important roles in many areas of computer vision, pattern recognition, and robotics including character recognition, fingerprint matching and industrial inspection. The thesis presents two affine-invariant shape descriptors which could be applied to both binary and grayscale images. Our purpose is to close the gap between signature-based descriptors and shape matching using gradient-based local description by increasing the recognition performance of the signature-based approaches. The local descriptors are increasingly used in image recognition due to their robustness to occlusions and geometrical deformations. The proposed algorithm utilizes gradient based features which are extracted along the object boundaries. We use two-dimensional steerable G -Filters to obtain gradient information at different orientations and scales, and then aggregate the gradients into a shape signature:

$$F(\Gamma) = \begin{bmatrix} D_{(\lambda,\tau)}^{\theta_0}(x_0, y_0) & \cdots & D_{(\lambda,\tau)}^{\theta_{M-1}}(x_0, y_0) \\ D_{(\lambda,\tau)}^{\theta_0}(x_1, y_1) & \cdots & D_{(\lambda,\tau)}^{\theta_{M-1}}(x_1, y_1) \\ \vdots & \ddots & \vdots \\ D_{(\lambda,\tau)}^{\theta_0}(x_{N-1}, y_{N-1}) & \cdots & D_{(\lambda,\tau)}^{\theta_{M-1}}(x_{N-1}, y_{N-1}) \end{bmatrix}$$

where $D_{(\lambda, \tau)}^{\theta_m}(x_k, y_k) = \left(I * G_{(\lambda, \tau)}^{\theta_m}\right)(x_k, y_k)$ and assuming that we use M number of steerable filters whose directions are multiple of π/M such that $\theta_m = m\pi/M$. Another alternative to the previous gradient-based signature definition is to treat the gradients in the perpendicular directions as a complex number rather than treating them as independent data. In this case, the response matrix takes the following form

$$H(\Gamma) = \begin{bmatrix} \left(D^{\theta_0} + jD^{\theta_0 + \frac{\pi}{2}}\right)(x_0, y_0) & \cdots & \left(D^{\theta_{M-1}} + jD^{\theta_{M-1} + \frac{\pi}{2}}\right)(x_0, y_0) \\ \left(D^{\theta_0} + jD^{\theta_0 + \frac{\pi}{2}}\right)(x_1, y_1) & \cdots & \left(D^{\theta_{M-1}} + jD^{\theta_{M-1} + \frac{\pi}{2}}\right)(x_1, y_1) \\ \vdots & \ddots & \vdots \\ \left(D^{\theta_0} + jD^{\theta_0 + \frac{\pi}{2}}\right)(x_{N-1}, y_{N-1}) & \cdots & \left(D^{\theta_{M-1}} + jD^{\theta_{M-1} + \frac{\pi}{2}}\right)(x_{N-1}, y_{N-1}) \end{bmatrix}$$

where $\theta_m = \frac{m\pi}{2M}$. The signature derived from the rotated object is circularly shifted version of the signature derived from the original object. This property is called the *circular-shifting rule*. In the first formulation, $F(\Gamma)$ is a N by M matrix while in the second formulation $H(\Gamma)$ is a N by $M/2$ matrix. On the other hand, $F(\Gamma)$ is a real valued matrix while $H(\Gamma)$ is a complex valued matrix. Note that both matrices $F(\Gamma)$ and $H(\Gamma)$ satisfy the perfect shifting rule under the rotations when θ is exact multiple of $\frac{\pi}{M}$.

The shape descriptors are defined as the Fourier transform of the signatures:

$$\begin{aligned} \tilde{f}_{k,m} &= \frac{1}{N} \sum_{n=0}^{N-1} f_{n,m} \exp\left(\frac{-j2\pi kn}{N}\right) \\ \tilde{h}_{k,m} &= \frac{1}{N} \sum_{n=0}^{N-1} h_{n,m} \exp\left(\frac{-j2\pi kn}{N}\right) \end{aligned}$$

We also provide a distance measure between two descriptors, $\tilde{p}_{k,m}$ and $\tilde{q}_{k,m}$ by taking the circular-shifting rule into account:

$$SD(\tilde{p}, \tilde{q}) = \min_{r \in \{0, \dots, M-1\}} \left\| \tilde{p}_{k, (m+r) \bmod M} - \tilde{q}_{k,m} \right\|_2.$$

The performance of the proposed descriptor is evaluated over two databases; one containing digits taken from vehicle license plates and the other containing MPEG-7 Core Experiment shape data set. The experiments show that the devised method outperforms other well-known Fourier-based shape descriptors such as centroid distance and boundary curvature. Another advantage of the proposed descriptors is that one can use them on both grayscale and binary images whereas other shape descriptors can only be applied to binary objects. The proposed algorithm takes the advantage of local distribution information found in the gradient cue to achieve this high performance. Even though the performance of the proposed descriptors ($\tilde{F}$ and $\tilde{H}$) are comparable, the compactness and the computational efficiency of the descriptor $\tilde{H}$ make this descriptor very attractive for content-based image retrieval systems.

In many vision applications encountered above, there is a great demand for an edge detector which can produce edge maps with very different characteristics in nature, so that one of these edge maps may meet the requirements of the problem under consideration. Unfortunately it is not evident how to choose the desired or the optimum edge maps

from these solutions that the edge detector offers. The proposed solutions are usually too general that cannot be easily adapted to the application needs by tuning edge detection parameters. One edge detector that we have studied in this dissertation is Generalized Edge Detector (GED) which is capable of producing edges with very different characteristics as we change the space parameters known as λ and τ . Although the edge maps based on this representation are reasonable, no one set of scale parameters alone yields a solution close to the desired edges. In this thesis, our aim is to find out powerful edge operators and use them under a goal-based edge detection framework. Proposed framework is a two-stage process. First, user marks some pixels in the database as edge and non-edge pixels. Then feature vectors comprised of filter responses to G-Filters at different scales are extracted at these marked pixels. Edge detection problem is imposed as two-class classification problem. Support vector machine (SVM) is used in the experiments. Classifier itself is not adequate to extract desired edges for the application under consideration. In the second stage continuous edges are treated as one contour. Then contours are matched with the contours in the training set. Only matched contours are kept and the other contours are eliminated.

The purpose of the first stage is to keep only prominent edges and remove irrelevant edges with respect to the application. The classifier decides which discontinuity is prominent or irrelevant. So how the training set is constructed is very important for the performance of the detector. Training set should contain almost equal number of edge and non-edge examples.

In summary, this dissertation proposes a new, effective and efficient framework for edge detection. The use of goal-oriented approach will find its applications in robust, accurate, efficient edge and contour extraction. Consequently, the approach developed here leads to powerful edge/contour-based systems especially for the applications involving face recognition and license plate detection.

AMACA YÖNELİK AYRIT SAPTAMA

ÖZET

Ayrıt saptamada amaç yoğunluk yüzeylerinden nesnelerin sınırlarına karşı düşen ayrıtların üretilmesidir. Üretilen bu ayrıtlar görüntü yüzeyindeki keskin değişimlerin neden olduğu süreksizlikler olarak ortaya çıkar. Görüntü yüzeyindeki bu ani değişimin kaynağı ya nesnenin dış yüzeyinin yapısal özellikleri (örneğin örüntü, örtme) ya da aydınlatma özellikleri (örneğin gölgeler, parlaklık) olabilir. Kaynağı ne olursa olsun, durağan bir görüntüdeki ayrıtların saptanması, yüksek doğruluk gerektiren ayrıt temelli herhangi bir bilgisayar görü algoritmasının önemli bir aşamasını oluşturur. Bir çok çevrit temelli bilgisayar görü uygulamasında (örneğin şekil-tabanlı sorgulama, çevrit-temelli görüntü çifti, çevrit-temelli görüntü sıkıştırma, ayrıt-temelli yüz tanıma ve ayrıt-temelli hedef tanıma) başarımlar büyük oranda saptanan ayrıtların doğruluğuna bağlıdır. Bu nedenle ayrıt saptama, bilgisayar görüde önemli araştırma konularından birini oluşturmaktadır.

Herhangi bir ayrıt saptayıcı, yüksek doğruluk ile gürültüyü bastırma özellikleri arasındaki ödünleşimi çözmelidir. Ne var ki, bir ön bilgi olmaksızın en iyi ödünleşimi belirlemek mümkün değildir. Gerçekten, bir beneğin ayrıt üzerinde bulunup bulunmadığına karar verme problemi kötü konumlandırılmış bir problemdir. Karar verme süreci yoğunluk yüzeyinin çeşitli derecelerden türevinin hesabını gerektirmektedir. Bilindiği gibi, türev alma işlevi gürültüyü kuvvetlendirmektedir. Bu problemin önüne geçmek için türev işlevinden önce düzleştirme işlemi uygulanır. Bu görüntüye alçak geçiren bir süzgeç uygulanması ile sağlanır. Mevcut ayrıt saptayıcıların çoğunluğu bu ödünleşimin en iyi çözümünü hedeflemişlerdir. Ancak en iyi çözümün tek bir ölçekte saptanan ayrıtlardan elde edilemeyeceği gösterilmiş ve farklı ölçeklerdeki ayrıtların saptanarak tümleştirilmesi çözümü önerilmiştir. Bu çalışmaların çoğu ya Gauss süzgeci ya da şekli Gauss süzgecine çok benzeyen süzgeçlerle sonuçlanmıştır. Ayrıca bu süzgeçler, tek bir ölçek parametresi ile yönlendirilebildiklerinden, ayrıtların ve yüzeylerin ölçek uzayının gösteriminin eldesinde kullanılmak için son derece uygundur. Bu süzgeçlerin geniş bir kullanıma sahip olmalarına rağmen belirli bir problem için istenilen sonucu sağladıklarını söylemek mümkün değildir. Örneğin, ayrıtın yerinin doğru olarak saptanmasının daha önemli olduğu durumu ele alalım. Bu durumda, yer bilgisini daha doğru veren eniyi-altı bir süzgeç kullanmak daha uygun olacaktır. Mevcut çözümlerin amaca yönelik ayrıtlar üretmekten uzak olmaları, araştırmacıları genel amaçlı bir ayrıt saptayıcı tasarımına yöneltmiştir.

Bu tez çalışmasında, $\lambda\tau$ -uzayı gösterilimi olarak bilinen düzleştirme ve ayrıt saptama süzgeçleri (R - and G -süzgeçleri) iki boyutlu uzaya genişletilmektedir. G -süzgeci Genelleştirilmiş Ayrıt saptayıcı olarak adlandırılmaktadır ve bir çok mevcut ayrıt saptayıcıyı üretebilmektedir. R -süzgecinin orta seviyede İşaret Gürültü Oranına (İGO) sahip tavla ve bar gibi yapay imgeler ile bazı gerçek imgeler üzerinde yapılan deneysel çalışmalarla iyi sonuçlar verdiği daha önce gösterilmiştir. Buna göre karma model zar ve levha modellerinin doğrusal bileşiminden oluşmaktadır:

$$E_m(f) = \iint (f(x, y) - d(x, y))^2 + \lambda (f_x^2(x, y) + f_y^2(x, y)) d\Omega$$

$$E_p(f) = \iint (f(x, y) - d(x, y))^2 + \lambda (f_{xx}^2(x, y) + 2f_{xy}^2(x, y) + f_{yy}^2(x, y)) d\Omega$$

Burada $E_m(f)$ zar modeline karşı düşen enerji fonksiyoneli ve $E_p(f)$ ise zar modeline karşı düşen enerji fonksiyoneli göstermektedir. Karma enerji fonksiyoneli ise

$$E_h(f) = \iint (f(x, y) - d(x, y))^2 + \lambda \left[\frac{(1 - \tau)(f_x^2(x, y) + f_y^2(x, y)) + \tau(f_{xx}^2(x, y) + 2f_{xy}^2(x, y) + f_{yy}^2(x, y))}{\tau} \right] d\Omega$$

biçiminde tanımlanmaktadır. Burada λ reel değerli düzenleme parametresi ve $\tau \in [0, 1]$ ise yine reel değerli süreklilik denetim parametresi olarak adlandırılmaktadır. Karma model zar modelinden levha modeline geçişteki ara modelleri içermektedir. Bu fonksiyonelerde, ilk terim çözümün $f(x, y)$ giriş verisine $d(x, y)$ uzaklığını tanımlamaktadır. İkinci terim düzenleme terimi olarak adlandırılmaktadır ve çözümdeki değişimin miktarını denetlemektedir. Bu iki terim arasındaki ödünleşim düzenleme katsayısı λ ile ayarlanmaktadır. Düzleştirme problemi daha sonra iki boyutlu kısmi diferansiyel denklemin (KDE) çözümüne indirgenmektedir:

$$f(x, y) - B(f_{xx}(x, y) + f_{yy}(x, y)) + A(f_{xxx}(x, y) + 2f_{xyy}(x, y) + f_{yyy}(x, y)) = d(x, y).$$

Verilen bir veri $d(x, y)$ için, çözüm $f(x, y)$ evrişim integrali hesabı ile elde edilebilmektedir:

$$f(x, y) = d(x, y) * R(x, y).$$

Görüldüğü gibi esas problem iki boyutlu uzayda tanımlanmıştır. Daha önceki özgün çalışmada, süzgeçler tek boyutlu uzayda çözümün ayrıştırılabilir olduğu varsayımı altında elde edilmiştir. İki boyutlu uzaydaki çözümün elde edilmiş olması özellikle iki açıdan önemlidir. Birincisi, ayrıtların ve yüzeylerin $\lambda\tau$ -uzayı gösterimi ve bu uzaydaki davranışları en iyi iki boyutlu uzaydaki çözümle modellenabilir. İkincisi, GAS bilinen bir çok ayrıt saptayıcıyı içerdiğinden, bu ayrıt saptayıcılarının iki boyutlu formları elde edilebilir. Deneysel çalışmalarda, iki boyutlu uzayda türetilen süzgeçlerin bir boyutlu uzayda türetilen süzgeçlere göre gürültüye karşı daha gürbüz oldukları hesaplanan FoM (Figure of Merit) ve yanlış kabul hatası karakteristikleri üzerinden gösterilmiştir. Ayrıt saptayıcılar ile çalışırken ayarlanması gereken çok sayıda parametre ile uğraşmak gerekmektedir. Verilen bir imge için en iyi parametre kümesini bulmanın genel geçer bir yöntemi bulunmaktadır. Gerçekte, bir imge için en iyilenen bir ayrıt saptayıcının parametre kümesi başka bir imge için en iyi olmayacaktır. Bu tez çalışmasında, en iyi GAS parametrelerini verilen bir imge için hesaplanan alıcı işletim eğrisi (AİE) üzerinden hesaplanmıştır. Bunun için özgün ayrıtların verilmiş olması gerekir. Burada amaç GAS'ın başarımının sınırlarını göstermektir. AİE tam kapsamlı arama ile elde edilmiştir.

Ne yazık ki, süzgeçlemenin dizisel uygulanması durumunda işlem süresi ölçek parametlerine (λ ve τ) bağlı olmaktadır. Süzgeç boyutu artan λ ile arttığından, evrişim toplamı her benek için daha fazla çarpma işlemi gerektirecektir. Bu çalışmada iki boyutlu R - ve G -süzgeçleri özyinelemeli süzgeçleme yaklaşımı ile gerçekleştirilmiştir.

Böylelikle süzgeçlemenin ölçek uzayı parametrelerinden bağımsız olarak sabit sayıda çarpma yaparak gerçekleştirilmesi sağlanmıştır. Bu çalışmada ayrıca, özyinelemeli süzgeçleme ile dördüncü dereceden yönbağımsız yayılım denklemi arasındaki ilişki ortaya konmuştur.

İmge gösteriminde amaç imgeyi farklı detay seviyelerinde ifade edebilmektir. Bu gürbüz bir gösterimin eksikliğinden dolayı bilgisayar görü ve görüntü işlemenin önemli problemlerinden biridir. Genel olarak, gösterimler doğrusal ve doğrusal olmayan olmak üzere ikiye ayrılabilir. Doğrusal gösterimler imgenin değişik ölçeklerdeki doğrusal bir süzgeç (çekirdek fonksiyonu) ile evriştirilmesi ile elde edilirler. Örneğin, klasik ölçek uzayı gösteriliminde çekirdek fonksiyonu Gauss süzgecidir ve ölçek uzayı gösterilimi imgenin artan standard sapmalı Gauss ile evriştirilmesi ile elde edilir. Benzer gösterim doğrusal yayılım denkleminin zamanda çözülmesi ile de elde edilebilir. Yöndeğişimsiz yayılım denklemi maksimum prensibini sağlaması nedeni ile bu gösterim aynı zamanda nedensel olarak adlandırılır. Ancak Gauss ile elde edilen ölçek-uzayı gösteriminin birbirine yakın ayrıtlar arasındaki etkileşimden kaynaklanan aşırı bulanıklaştırma ve ayrıtlardaki konumlandırma hataları nedeni ile önemli yitimleri vardır. Düzleştirme tüm doğrultularda aynı oranda türdeş olarak uygulanmaktadır. Bu ise nesne yüzeyleri ile sınır bölgelerinin aynı oranda bulanıklaşmasına neden olmaktadır. $\lambda\tau$ -uzayı gösterimi daha zengin bir gösterim sunsa da, doğrusal gösterimlerdeki aynı sorunu taşımaktadır. Uzay parametrelerinin imgenin her noktasında aynı değere sahip olması gürültü süzülürken ayrıtların da aynı oranda bulanıklaşmasına neden olmaktadır. Bu davranış gösterilimin doğrusal doğasından kaynaklanmaktadır. Bu problemi çözmek için, düleştirme parametresi λ 'yı imge boyunca yerel öznitelikleri ve doğrultuları dikkate alınarak değiştirildiği bir yönbağımlı $\lambda\tau$ -uzayı gösterimi geliştirilmiştir. Deneysel olarak, önerilen yöntemdeki karma yayılım denkleminde, her iki terimin de kullanıldığı durumda elde edilen sonuçların Perona-Malik'in yönbağımlı yayılım denklemi ile elde edilen sonuçlara göre daha başarılı olduğu gösterilmiştir. Önerilen yaklaşımda τ 'nın seçimine göre farklı karakteristikte yüzeyler elde edilebilmektedir.

Şekil betimleyiciler karakter tanıma, parmakizi eşleme, endüstriyel denetim gibi bilgisayar görünüm ve görüntü tanımanın bir çok uygulamasında yoğun olarak kullanılmaktadır. Bu tez çalışmasında iki tane ilgin değişmez şekil betimleyicisi geliştirilmiştir. Bu betimleyiciler daha önceki çalışmaların aksine hem ikili hem de gri-seviyeli imgelere uygulanabilir niteliktedir. Amacımız imza temelli betimleyiciler ile şekil eşleme yöntemleri arasındaki uçurumu imza temelli betimleyicilerin tanıma başarımını artırarak, gradyan temelli yerel betimleyiciler ile kapatmaktır.

Yerel betimleyiciler, örtme ve geometrik bozulmalara karşı gürbüzdür ve bu nedenle imge tanımda yoğun olarak kullanılmaktadır. Bu çalışmada, nesne sınırı boyunca hesaplanan bir dizi yönlendirilebilir süzgeç cevabının kullanıldığı çevrit temelli global şekil betimleyicisi geliştirilmiştir. Bu çalışmada, farklı ölçek ve doğrultularda gradyan vektörünün saptanmasında, yönlendirilebilir G süzgeçleri kullanılmıştır. Daha sonra bu gradyanlardan yararlanarak şekil imzası tanımlanmıştır:

$$F(\Gamma) = \begin{bmatrix} D_{(\lambda,\tau)}^{\theta_0}(x_0, y_0) & \cdots & D_{(\lambda,\tau)}^{\theta_{M-1}}(x_0, y_0) \\ D_{(\lambda,\tau)}^{\theta_0}(x_1, y_1) & \cdots & D_{(\lambda,\tau)}^{\theta_{M-1}}(x_1, y_1) \\ \vdots & \ddots & \vdots \\ D_{(\lambda,\tau)}^{\theta_0}(x_{N-1}, y_{N-1}) & \cdots & D_{(\lambda,\tau)}^{\theta_{M-1}}(x_{N-1}, y_{N-1}) \end{bmatrix}.$$

Burada $D_{(\lambda, \tau)}^{\theta_m}(x_k, y_k) = (I * G_{(\lambda, \tau)}^{\theta_m})(x_k, y_k)$ olarak tanımlıdır ve M adet yönlendirilebilir süzgeç cevabı kullanılmıştır. Bu süzgeçlerin doğrultuları $\theta_m = m\pi/M$ olacak şekilde π/M 'nin tam katıdır. Bir önceki gradyan tabanlı şekil imzasında, birbirine dik doğrultulardaki cevaplar bir karmaşık sayı olarak değerlendirilerek, ikinci bir şekil imzası elde edilmiştir. Bu durumda cevap matrisi aşağıdaki gibi tanımlı olacaktır:

$$H(\Gamma) = \begin{bmatrix} (D^{\theta_0} + jD^{\theta_0 + \frac{\pi}{2}})(x_0, y_0) & \cdots & (D^{\theta_{M-1}} + jD^{\theta_{M-1} + \frac{\pi}{2}})(x_0, y_0) \\ (D^{\theta_0} + jD^{\theta_0 + \frac{\pi}{2}})(x_1, y_1) & \cdots & (D^{\theta_{M-1}} + jD^{\theta_{M-1} + \frac{\pi}{2}})(x_1, y_1) \\ \vdots & \ddots & \vdots \\ (D^{\theta_0} + jD^{\theta_0 + \frac{\pi}{2}})(x_{N-1}, y_{N-1}) & \cdots & (D^{\theta_{M-1}} + jD^{\theta_{M-1} + \frac{\pi}{2}})(x_{N-1}, y_{N-1}) \end{bmatrix}.$$

Burada $\theta_m = \frac{m\pi}{2M}$ olarak tanımlıdır. özgün nesne için elde edilen imza ile nesnenin döndürülmüş sürümü için elde edilen imza arasında *mükemmel çevresel öteleme* adı verilen bir ilişki vardır. Birinci formülasyonda, $F(\Gamma)$ N 'ye M boyutlarında bir matris, ikinci formülasyonda $H(\Gamma)$ ise N 'ye $M/2$ boyutlarında bir matristir. Diğer taraftan, $F(\Gamma)$ reel bileşenli, $H(\Gamma)$ ise karmaşık sayı bileşenli bir matristir. Hem $F(\Gamma)$ hem de $H(\Gamma)$, θ 'nın $\frac{\pi}{M}$ 'nin tam katı olması durumunda, mükemmel çevresel öteleme adı verilen özelliği sağlamaktadır.

Şekil betimleyiciler, imzaların Fourier dönüşüğü olarak tanımlanmışlardır:

$$\begin{aligned} \tilde{f}_{k,m} &= \frac{1}{N} \sum_{n=0}^{N-1} f_{n,m} \exp\left(\frac{-j2\pi kn}{N}\right) \\ \tilde{h}_{k,m} &= \frac{1}{N} \sum_{n=0}^{N-1} h_{n,m} \exp\left(\frac{-j2\pi kn}{N}\right) \end{aligned}$$

Ayrıca, bu tez çalışmasında, $\tilde{f}_{k,m}$ ve $\tilde{p}_{k,m}$ gibi iki betimleyici arasındaki benzerlik oranı için çevresel öteleme özelliğini de dikkate alan bir uzaklık ölçütü tanımlanmıştır:

$$SD(\tilde{f}, \tilde{p}) = \min_{r \in \{0, \dots, M-1\}} \left\| \tilde{f}_{k, (m+r) \bmod M} - \tilde{p}_{k,m} \right\|_2.$$

Yeni şekil betimleyicinin tanıma başarısı iki veri tabanı üzerinde test edilmiştir; gri seviyeli araç plaka görüntülerinden elde edilen rakam veritabanı ve MPEG-7 Core Experiment şekil veri tabanı. Deneysel sonuçlar, geliştirilen yöntemin diğer çok bilinen sınır tabanlı betimleyicilere (merkez uzaklığı, sınır eğriliği gibi) göre çok üstün olduğu gösterilmiştir. Tanıtılan yöntemin diğer bir kazanımı, diğer yöntemlerin aksine, hem ikili hem de gri seviyeli görüntülere uygulanabiliyor olmasıdır. önerilen yöntem bu başarımı gradyan işaretindeki yerel dağılım bilgisine borçludur. Her ne kadar, önerilen betimleyicilerin ($\tilde{F}$ ve $\tilde{H}$) başarımı birbirine yakın olsa da, tıklılık ve işlem maliyeti açısından karşılaştırıldığında, $\tilde{H}$ betimleyicisi içerik tabanlı imge sorgulama sistemleri için daha uygun bir çözümdür.

Yukarıda sayılan birçok bilgisayar görü uygulamasında, farklı özelliklerde ayırt haritası üretebilen ayırt saptayıcılara oldukça fazla ihtiyaç duyulmaktadır. Bu farklı karakteristiklerdeki ayırt haritalarından birinin eldeki uygulamanın ihtiyaç duyduğu ayırtları üreteceği umulmaktadır. Ancak ayırt saptayıcının önerdiği çözümlerden uygulamanın ihtiyaç duyduğu en iyi ayırtların seçimi ile ilgili genel geçer bir çözüm yoktur.

Ayrıt saptama için önerilen yöntemler çoğunlukla uygulamanın ihtiyaç duyduğu ayrıtları üretecek şekilde kolaylıkla uyarlanabilir olmaktan uzaktır. Bu tez çalışmasında üzerinde çalışılan ve kullanılan Genelleştirilmiş Ayrıt Saptayıcısının (GAS) ölçek parametreleri olan λ ve τ 'nin değiştirilmesiyle farklı özelliklerde ayrıtlar üretilebilmesi sağlanmıştır. Üretilen ayrıt haritalarının başarılı olmasına karşın tek bir ölçekte amaca uygun ayrıtlar üretmek mümkün değildir. Tez çalışmasında amacımız, güçlü ayrıt saptayıcıları geliştirmek ve bu saptayıcıları amaca yönelik ayrıt saptama çatısı altında kullanmaktır. Önerilen çatı iki aşamalı bir süreçten oluşmaktadır. İlk olarak kullanıcı, veritabanındaki imgelerdeki beneklerin bir kısmını ayrıt benekleri ve ayrıt olmayan benekler olarak etiketlemektedir. Daha sonra işaretlenen bu benekler üzerinde hesaplanan çeşitli ölçek ve doğrultulardaki G –süzgeç cevapları birleştirilerek öznitelik vektörleri elde edilir.

Böylelikle, ayrıt saptama problemi iki sınıflı sınıflandırma problemine dönüştürülmüş olur. Bu çalışmada sınıflandırıcı olarak Destek Karar Makinalarından (DKM) yararlanılmıştır. Sınıflandırıcı tek başına uygulama için istenilen ayrıtları üretmek için yeterli değildir. İkinci aşamada sürekli ayrıtlar birer çevit olarak değerlendirilir ve ardından bu çevitler eğitim kümesindeki çevitler ile eşlenir. Eşlemede, bu tez çalışmasında geliştirilen gradyan temelli ilgin değişimsiz şekil betimleyicileri kullanılmıştır. Sadece bu eşlemeyi geçen çevitler çevit olarak kabul edilir.

Birinci aşamanın amacı sadece uygulama için anlamlı olabilecek ayrıtları tutmak ve diğer ilgisiz ayrıtları elemektir. Sınıflandırıcı hangi değişimin önemli hangisinin ise ilgisiz olduğuna karar vermektedir. Bu nedenle eğitim kümesinin nasıl oluşturulduğu son derece önemlidir. Eğitim kümesinde yaklaşık olarak eşit sayıda ayrıt ve ayrıt olmayan benek örneği bulunmalıdır. Bu sınıflandırıcının ve birinci aşamanın başarımı için önemlidir.

Özet olarak, bu tez çalışmasında yeni, etkin ve verimli ayrıt saptama çatısı geliştirilmiştir. Önerilen yaklaşım, gürbüz, kesin ve etkin ayrıt ve çevit saptamaya ihtiyaç duyulan bir çok uygulamada yerini bulacağına inanıyoruz. Sonuç olarak, geliştirilen yöntem güçlü ayrıt/çevrit-temelli yüz tanıma ve plaka tanıma gibi bilgisayar görü uygulamalarının geliştirilmesine olanak sağlayacaktır.

1. INTRODUCTION

This dissertation presents a new approach to edge detection. Edge detection is one of the most studied yet remained unsolved problem of image processing and computer vision fields. It has become increasingly popular in recent years in a variety of pattern recognition applications such as face recognition and vehicle license plate recognition systems. The purpose of this chapter is to: 1) introduce the problem addressed in this dissertation; 2) summarize the current state of edge detection, and within this context discuss the needs for new edge detection techniques; 3) provide a concise description of the approach taken in this work by pointing out the main contributions and an outline of the dissertation.

1.1 Goal of the Thesis

The aim of edge detection is to provide a meaningful description of object boundaries in a scene from intensity surface. These boundaries are due to discontinuities manifesting themselves as sharp variations in image intensities. There are different sources for sharp changes in images which are created by structure (e.g. texture, occlusion) or illumination (e.g. shadows, highlights). Extracting edges from a still image is certainly the most significant stage of any computer vision algorithm requiring high accuracy of location in the presence of noise. In many contour-based vision algorithms, such as shape-based query [2], curve-based stereo vision [3], contour-based image compression [4] and edge-based target recognition [5], edge-based face detection [6] their performance is highly dependent on the quality of the detected edges. Therefore, edge detection is an important area of research in computer vision. Despite considerable work and progress made on this subject, edge detection is still a challenging research problem due to the lack of a robust and efficient general purpose algorithm.

Any edge detector has to tackle with the tradeoff between good localization property forcing the location of the detected edges to be as close as possible to the real edges and good noise rejection property forcing the intensity surface to be smooth. Without a priori assumptions, one can not select the best tradeoff. In fact, deciding whether a pixel belongs to a contour is an ill-posed problem. The detection of sharp changes in image intensity requires computation of derivatives of the noisy image at different orders. As

known, numerical computation of derivatives of noisy data is an unstable process since it amplifies the noise. To overcome this problem, edge detection algorithms first employ a noise suppression process prior to the differentiation operation which can be performed by smoothing the noisy image by a low-pass filter.

Most of the efforts in edge detection have been devoted to the development of an optimum edge detector which can resolve the tradeoff [7, 8, 9, 10]. Extracting edges at different scales [7, 8, 11] and combining these edges [8, 12] have attracted a substantial amount of interest. In the course of developing optimum edge detectors that can resolve the tradeoff between localization and detection performances, several different approaches have resulted in either a Gaussian filter or a filter whose shape is very similar to a Gaussian [7, 8, 13]. Furthermore, these filters are very suitable for obtaining scale space edge detection since the scale of the filter can be easily controlled by means of a single parameter. Although these filters are used very widely, it is difficult to claim that they can provide the desired output for any specific problem. For instance, there are some cases where the improved localization performance is the primary requirement. In these cases, a sub-optimum filter which promotes the localization performance becomes more appropriate. It has been shown that the first order R -filter can deliver improved results on checkerboard and bar images as well as on real images with moderate signal-to-noise ratio (SNR) [12].

In many vision applications encountered above, there is a great demand for an edge detector which can produce edge maps with very different characteristics in nature, so that one of these edge maps may meet the requirements of the problem under consideration. Unfortunately it is not evident how to choose the desired or the optimum edge maps from these solutions that the edge detector offers. The proposed solutions are usually too general and cannot be easily adapted to the application needs by tuning edge detection parameters. In order to illustrate the problem, we consider two prominent computer vision applications: human face recognition and traffic license plate recognition. We take two sample images from the application domains as shown in Fig. 1.1. One edge detector that we will study in this dissertation is Generalized Edge Detector (GED) which is capable of producing edges with very different characteristics as we change the space parameters known as λ and τ . The edge maps produced by the detector are given in Fig. 1.2 for the face image and in Fig. 1.3 for the license plate image. Although the edge representation contains reasonable edge maps, none of the scale parameters yield a solution close to the ideal edges given in Fig. 1.1b. In the thesis, our aim is to find out powerful edge operators and use them under a goal-based edge detection framework. Proposed framework is a two-stage process. First, user marks some pixels in the database as edge and non-edge pixels. Then

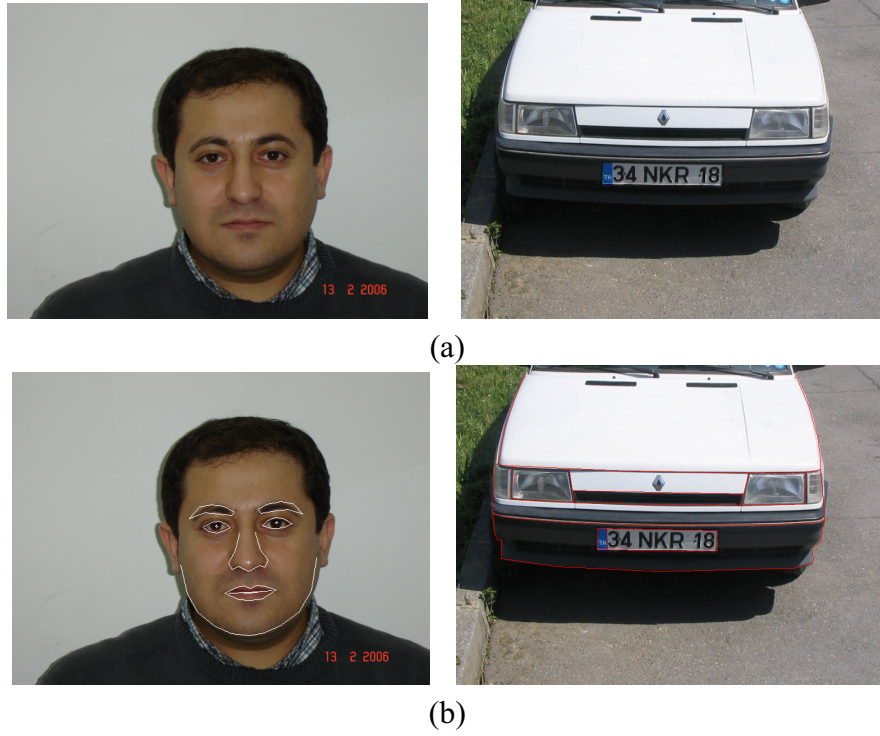


Figure 1.1: (a) Two sample images from two prominent computer vision applications and (b) their “ideal” edges.

feature vectors comprised of filter responses to G-Filters at different scales are extracted at these marked pixels. Edge detection problem is imposed as a two-class classification problem. Support vector machine (SVM) is used to obtain a classifier. Classifier itself is not adequate to extract desired edges for the application under consideration. In the second stage continuous edges are treated as one contour. Then contours are matched with the contours in the training set. Only matched contours are kept and the other contours are eliminated. The purpose of the first stage is to keep only prominent edges and remove irrelevant edges with respect to the application. The classifier decides which discontinuity is prominent or irrelevant. So how the training set is constructed is very important for the performance of the detector. Training set should contain almost equal number of edge and non-edge examples. Target applications that we studied in this thesis are face recognition and license plate detection.

1.2 Survey of Edge Detectors

Any edge detector can be considered as a point in a 3D space whose dimensions are edge operator, kernel and decision in this classification (see Fig. 1.4). The earliest works use small smoothing kernels to enhance the input signal. The edge operators are usually either the first derivative or the second derivative of an image. Examples are Roberts filter, Sobel filter, Prewitt filter, and Laplacian filter [14]. The Robert’s filter is a simple

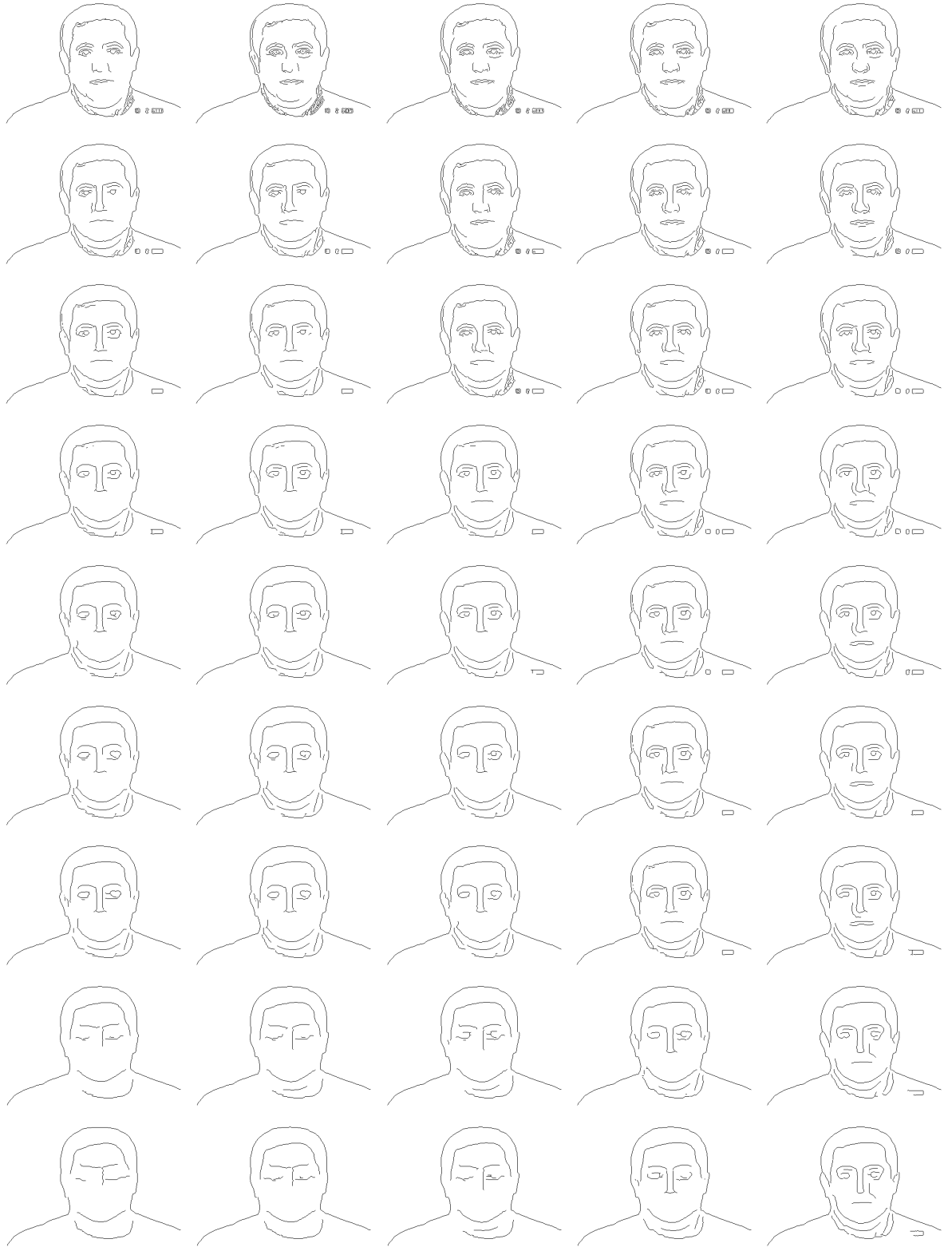


Figure 1.2: $\lambda\tau$ -space representation of edges ($\lambda = 1.0, 2.5, 5.0, 10.0, 15.0, 20.0, 25.0, 50.0, 75.0, \tau = 0.0, 0.25, 0.5, 0.75$)

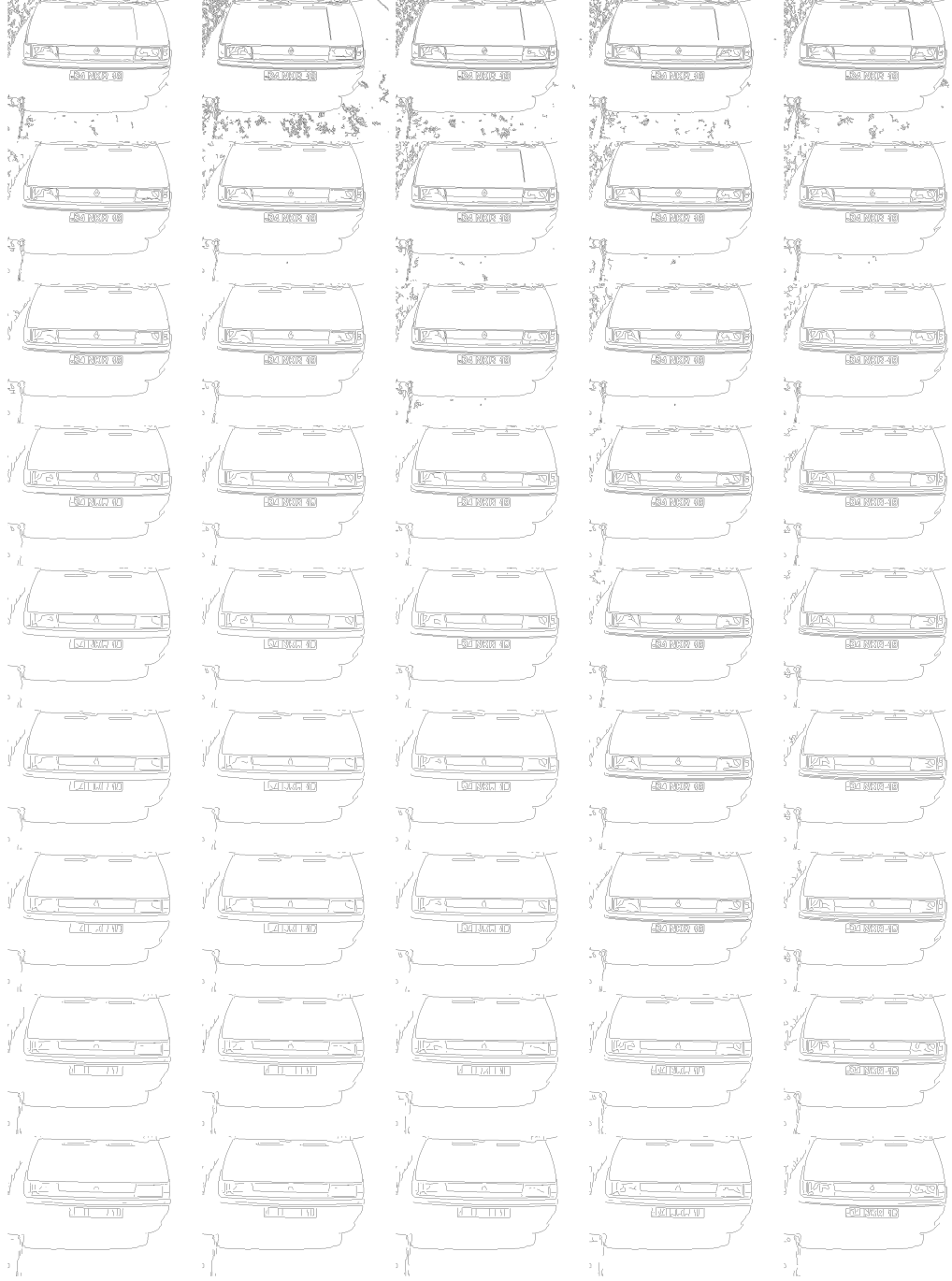


Figure 1.3: $\lambda\tau$ -space representation of edges ($\lambda = 1.0, 2.5, 5.0, 10.0, 15.0, 20.0, 25.0, 50.0, 75.0, \tau = 0.0, 0.25, 0.5, 0.75$)

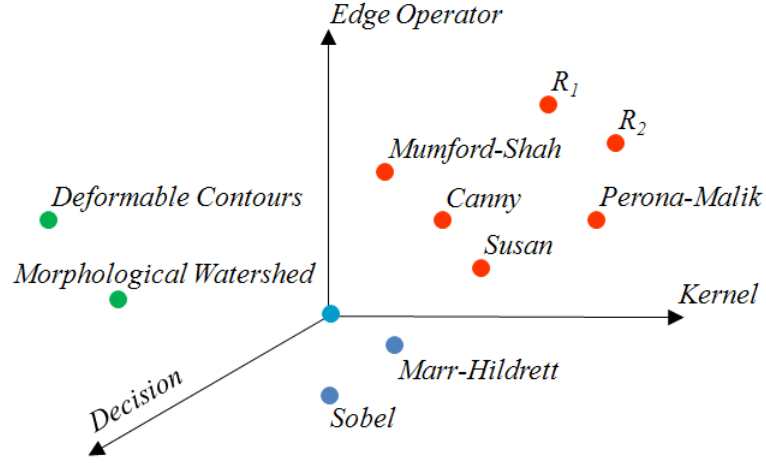


Figure 1.4: Edge detector space.

1	2	1	1	0	-1
0	0	0	2	0	-2
-1	-2	-1	1	0	-1
(a)			(b)		

Figure 1.5: Sobel edge operators: (a) G_x , (b) G_y .

gradient operator based on a 2×2 gradient operator. This operator provides the simplest approximation of the gradient magnitude given as

$$G(f(i, j)) = [f(i, j) - f(i + 1, j + 1)] + [f(i + 1, j) - f(i, j + 1)]$$

Since the kernel is only a 2×2 mask, it is quite sensitive to noise. Sobel filter is a gradient operator defined over 3×3 neighborhood system. The convolution filters for the Sobel operator are defined by the two kernels shown in Fig. 1.5. The kernels are used to extract the vertical and the horizontal components of the gradient as

$$G_x(f(i, j)) = [f(i - 1, j - 1) + 2f(i - 1, j) + f(i - 1, j + 1)] \\ - [f(i + 1, j - 1) + 2f(i + 1, j) + f(i + 1, j + 1)]$$

and

$$G_y(f(i, j)) = [f(i - 1, j - 1) + 2f(i, j - 1) + f(i + 1, j - 1)] \\ - [f(i - 1, j + 1) + 2f(i, j + 1) + f(i + 1, j + 1)].$$

The magnitude of the gradient vector is usually computed as

$$G(f(x, y)) = \sqrt{G_x^2 + G_y^2}.$$

1	1	1
0	0	0
-1	-1	-1

(a)

0	1	1
-1	0	1
-1	-1	0

(b)

-1	0	1
-1	0	1
-1	0	1

(c)

-1	-1	0
-1	0	1
0	1	1

(d)

Figure 1.6: Prewitt edge operators at various directions.

	0	1	0
$\frac{1}{4}$	1	-4	1
	0	1	0

Figure 1.7: Laplacian edge operator for a 4-neighborhood system.

The Prewitt operator is defined by a set of eight kernels, four of which are shown in Fig. 1.6. Other four kernels are generated by rotating these kernels by 90°. The mask that produces maximal response yields the direction of the gradient.

The gradient operators are all anisotropic, which means that they are not rotation invariant. Operators that use odd order derivatives are all anisotropic. On the other hand, operators that uses even order derivatives are all isotropic. If we apply the isotropic operators to an image and then rotate the resultant image, it will yield the same results as rotating the image first and then applying the operator. An example of isotropic operator is Laplacian operator. It is defined in terms of second-order derivatives as

$$G_{xx}(f(i, j)) = f(i, j + 1) - 2f(i, j) + f(i, j - 1)$$

$$G_{yy}(f(i, j)) = f(i + 1, j) - 2f(i, j) + f(i - 1, j)$$

The corresponding kernel which is a combination of responses along x and y directions for a 4-neighborhood system is shown in Fig. 1.7. Since Laplacian operator contains second order derivatives of the input signal, it is more susceptible to noise than the operators that make use of the first-order derivatives.

In decision phase simple or hysteresis thresholding is applied to the output of these filters to identify the edge points. Although the earliest edge detectors are easy to implement

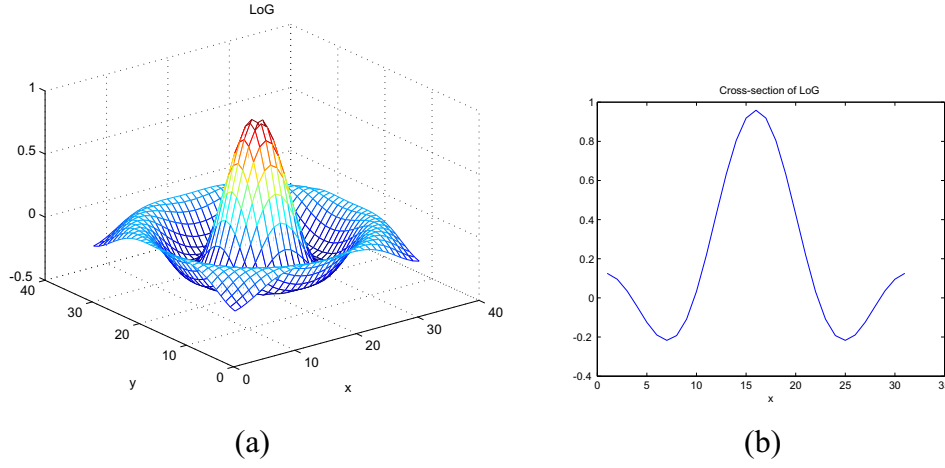


Figure 1.8: Laplacian of Gaussian: (a) 3D view (b) Cross-section of (a) at $y = 0$.

and faster, they provide no means to control the tradeoff between edge localization and smoothing. Therefore, these filters are very noise-sensitive.

Marr and Hildreth [7] have proposed an edge detector where the edge operator is zero-crossings and the kernel is the Laplacian of Gaussian (LoG). LoG performs first the Gaussian smoothing, then the Laplacian operation. Laplacian operator in continuous domain is defined as

$$\Delta f(x, y) = \nabla^2 f(x, y) = f_{xx}(x, y) + f_{yy}(x, y)$$

and one of its discretized version is given in Fig. 1.7. Laplacian operator is susceptible to noise. Applying the convolution with Gaussian reduces the noise and the combined mask minimizes the probability of detecting false edges. The LoG operator is defined in continuous domain as

$$LoG(x, y) = \frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (1.1)$$

The 3D plot of the $LoG(x, y)$ function looks like a Mexican hat (Fig. 1.8a). As σ increases, we have a wider convolution mask. The first zero-crossing of the LOG function takes place at $\sqrt{2}\sigma$ and the two dimensional cross section of the zero-crossing of the Mexican hat is shown in Fig. 1.8b. A step edge corresponds to a sudden change in the image, so the first derivative of the image should have a maximum at the edge point, and so the second derivative should be zero at the same pixel. Usually it is easier to find a zero-crossing than a local maximum in a 2D signal. On the other hand, the higher-order derivatives are also known to be more sensitive to noise. In order to reduce noise, the image has to be smoothed. When choosing a smoothing filter, two criteria should be fulfilled [7]. First, the filter should be smooth and roughly band-limited in the frequency domain to reduce

the number of frequencies at which image function changes can take place. Second, the constraint of spatial localization requires the response of a filter to be from nearby points in the image. These two criteria are conflicting, and the Gaussian filter ($e^{-\frac{x^2+y^2}{2\sigma^2}}$) is an compromise between spatial and frequency criteria.

Poggio et al. [13] introduced a cubic spline filter. The image is smoothed by a cubic spline filter before differentiation. It was shown that the results of the cubic spline filter are very similar to Gaussian smoothing. Canny described a widely used edge detection algorithm [8] which is optimal to step edges corrupted by noise. Three criteria were defined for the optimality of edge detection:

- **Good Detection:** The detector must minimize the probability of false edges caused by noise, as well as missing real edges.
- **Good Localization:** The edges detected must be as close as possible to the true edges.
- **Single Response:** The detector must return one point only for each true edge point.

First two criteria are then developed quantitatively into a set of functions with minimal and maximal constraints. A closed form solution was found using variational calculus. Without the third criterion, the optimal detector for a step edge

$$G(x) = \begin{cases} 0 & x < 0 \\ A & x \geq 0 \end{cases}$$

is $f(x) = -G(x)$ in $[-W, W]$, assuming the filter width is $2W$. Therefore, the optimal, one-dimensional step edge detector is a truncated step. Unfortunately, this filter contains a very high bandwidth and tends to produce many maxima with noisy step edges. If the third criterion is added, the optimal solution may be found by numerical method. The result filter can be approximated with error less than 20% by the first derivative of a Gaussian smoothing filter. This is similar to Marr-Hildreth edge detector [7] which is based on the Laplacian of a Gaussian. The detector is then generalized to two-dimensions. Non-maximum suppression is then applied on the results of the filter to thin wide edges in order to produce 1-pixel wide edges. It is done by finding local maxima in the direction perpendicular to the edges. Finally, weak edges are removed using thresholding. The thresholding is applied with hysteresis. Edges contours are processed as units, and two threshold values are defined; each contour must have at least one point with gradient magnitude above the higher threshold, while all points in the contours must not go below the lower threshold. Canny also proposed feature synthesis which is a multiple-scale

approach where the standard deviation of the Gaussian are used as the scaling factor. All significant edges from the operator with the smallest scale are marked first, and the edges for larger scale are synthesized from these marked edges and then compared to the actual detector output. Additional edges are marked only if they have a significantly stronger response than that predicted by the synthetic response. Many researchers were inspired by Canny's work. For example, Deriche [9] extended Canny's initial filter to two-dimensions and implemented it using recursive filtering. Petrou and Kittler [15] derived another optimal detector for a blurred step edge model using criteria similar to those by Canny.

Canny's feature synthesis is not the first attempt to detect edges using different scale parameters. Marr and Hildreth [7] has suggested to obtain a description of an image at different scales by applying a feature detector at different scales and combining the edge information. Another alternative to Gaussian filtering is to solve the linear heat equation in time

$$\frac{\partial f(x, y, t)}{\partial t} = K \nabla^2 f(x, y) = K \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right) \quad (1.2)$$

where $f(x, y, 0) = f_0(x, y)$ is the original noisy image and the constant K is called the diffusion rate. $f(x, y, t)$ represents a solution of the equation (1.2) at the time instance t . It is equivalent to convolving the initial image with the Gaussian filter with the variance σ . The relationship between σ , K and t is given by [16, 17]

$$\sigma^2 = 4\pi Kt \quad (1.3)$$

One disadvantage of linear isotropic diffusion is that the diffusion would blur features as well as smooth noise. An isotropic diffusion method was introduced by Perona and Malik [18] where the diffusion rate is controlled by the function $c(x, y, t)$:

$$\begin{aligned} \frac{\partial f(x, y, t)}{\partial t} &= \nabla(c(x, y, t) \nabla f(x, y, t)) \\ &= \nabla c \nabla f + c \nabla^2 f \end{aligned} \quad (1.4)$$

$c(x, y, t)$ is chosen as $c(x, y, t) = g(\nabla f(x, y, t))$ and g is a nonnegative monotonically decreasing function with $g(0) = 1$. A different function g would generate different scale-spaces, though they turn out to be perceptually similar. This way, the points with strong features, i.e., with high $|\nabla f|$ values, would have less diffusion effect than other points, thus retain most of the features in the image. To make the diffusion anisotropic, the function $c(x, y, t)$ should be directional and is perpendicular to the orientation of the gradient of the image. The nonlinear diffusion methods process an image with different smoothing parameters such that it could blur noise and at the same time keep the edges

sharp. Another approach of edge detection is parametric fitting. This involves fitting the image with an edge model with parameters, then finding the parameters that minimize the fitting error. The Hueckel edge detection [19] is an example of the parametric fitting method. A two-dimensional step edge model in a circular window is fitted to each image pixel. The parameters of the edge model are the gray scale values, the edge orientation, and the distance from the center of the window to the edge. An edge is detected if the fit is accurate enough. Accuracy of edge fitting is measured in terms of the mean square error. Hueckel introduced a polar Fourier expansion and used the first eight Fourier coefficients in the fitting process in order to reduce the computation. Experimental results indicated that Hueckel operator performs well in noisy and highly textured environments, though no analysis of the operator in presence of noise is documented.

Some other detectors that can be placed in the parametric fitting category are proposed by O’Gorman [20], Hummel [17], Hartley [21], Nalwa and Binford [22], and Lyvers et al. [23]. Moreover, this approach is also used to detect other features. Hueckel [19] used the same model to detect line edges, and Rohr [24] and Parida et al. [25] used it to detect corners. Since this approach uses a rich description of the image structure, edge detectors using this approach could provide edge attributes such as position with subpixel accuracy, contrast, blur, and width. In other words, this approach could provide more thorough description of the edges. One problem of this approach is that it is not easy to provide a two dimensional edge model with an arbitrary-shaped curve. Most edge detectors use a straight edge model. For some complex features such as a corner, or more than one edges within the window, the detectors can still pick the accurate parameter set that minimizes the edge fitting error. Therefore the value of the edge fitting error will be increased, and the edge might be rejected once the error exceeds the default threshold.

Edge detection is an ill-posed problem. One solution is to convert the edge detecting problem into energy functionals imposing some constraint over the intensity surface usually in the form of its derivatives. Then the solution is obtained by minimizing the energy functions. We selected two examples in this class of techniques. These are deformable contours [26], and the Mumford- Shah theory [29].

The deformable contour technique employs energy minimization to shift an initial line segment, which is called the deformable contour, the active contour, or the snake, into a curve that would adhere to an edge of the given image. For edge detecting purposes, two forms of energy are involved: an internal energy that generates a force keeping the curve continuous and smooth, and an image energy that move the curve towards edge points on the image. The Euler equation is used to minimize the energy function. When the

energy is minimized, the two forms of energy are at balance. Therefore, the deformable contour is attached to an edge, and is smoothed so it is not zigzag due to digitization or noise. Deformable contours are usually used in motion analysis, and stereopsis, not in single stationary image. The reason is the technique requires a proper initial line segment or closed curve to begin with. Without some external input, this technique does not work for a single image. However, in a motion picture, or in an image pair, outlines of objects can usually be detected by finding the difference of two images. Those outlines can be used as the initial line segments for deformable contours. There are different extensions of this model [27, 28].

Mumford and Shah [29] proposed another form of energy minimizing to detect boundary of noisy images. For a given image $g : R \rightarrow \mathbb{R}$, the energy functional E for a boundary B and smoothed image f is given by

$$E(f, B) = \mu^2 \int_R (f - g)^2 + \underbrace{\int_{R-B} \|\nabla f\|^2}_{\text{SmoothnessConstraint}} + \underbrace{\nu |B|}_{\text{EdgePenalty}} \quad (1.5)$$

where R is the region of the image, $f : R \rightarrow \mathbb{R}$ is the smoothed ideal image, $B : R \rightarrow \{0, 1\}$ is the binary edge process, μ and ν are scalar parameters. $|B|$ is the volume of B . For two-dimensional images, $|B|$ is the length of the edges and $R - B$ denotes the region R excluding the edges.

The Mumford-Shah energy $E(f, B)$ formalizes a tradeoff between noise removal and edge detection. However, this functional lacks a practical means to find the smoothed image f , and the binary edge process B that minimize the energy function $E(f, B)$. Many methods have been proposed to simplify the minimization of the functional. In one of them, [30], the binary edge process, B , of the energy function (1.5) is replaced by a continuous edge field. The energy functional would become

$$E(f, v) = \int_R \underbrace{\mu (f - g)^2}_{\text{Data Fidelity}} + \underbrace{\alpha (1 - v)^2 \|\nabla f\|^2}_{\text{Smoothness Constraint}} + \underbrace{\frac{\beta}{2} \|\nabla v\|^2 + \frac{v^2}{2\beta}}_{\text{Edge Penalty}} \quad (1.6)$$

which is called the Ambrosio-Tortorelli functional. The $g : R \rightarrow \mathbb{R}$ denotes the given image, $f : R \rightarrow \mathbb{R}$ the piecewise smoothed image, $v : R \rightarrow [0, 1]$ the corresponding continuous edge strength of the image, R the image domain, α , β and μ are scalar parameters. This functional could be minimized using the Euler equation.

1.3 Contributions of the Thesis

The first major contribution of this dissertation is the formulation of edge detection within a goal-based, regularized, shape-aware framework. This framework provides mechanisms for edge detection without requiring any thresholding or parameter tuning. It also takes the contour geometry into account.

The second contribution consists of two gradient-based affine-invariant shape descriptors denoted as $\tilde{F}$ and $\tilde{H}$ which could be applied to both binary and grayscale images. We use two-dimensional steerable G –Filters to obtain gradient information at different orientations and scales, and then aggregate the gradients into a shape signature. The signature derived from the rotated object is circularly shifted version of the signature derived from the original object. This property is called the circular-shifting rule. The shape descriptor is defined as the Fourier transform of the signature. We also provided a distance measure for the proposed descriptor by taking the circular-shifting rule into account.

The third contribution of this dissertation is the extension of Generalized Edge Detector (GED) to two dimensional space. Two dimensional extension of the representation is important because of that the behaviour of images in the space is best modeled by two dimensional smoothing and edge detector filters. Also since GED filters encompass most of the well-known edge detectors, two dimensional version of these filters could be obtained. The derived filters are more robust to noise when compared to the previous one dimensional filtering scheme in the sense of FoM (Figure of Merit) and false alarm characteristics.

The final contribution of this dissertation is a practical one. We have implemented recursive filtering framework to include two dimensional R – and G –Filters. Filtering is possible now in constant time with this recursive implementation.

Overall, this dissertation proposes a new, effective and efficient framework for edge detection. The use of goal-based approach results in edges which lend themselves to robust, accurate, efficient edge and contour extraction. Consequently, the approach developed here leads to powerful edge/contour-based systems especially for applications involving face recognition and license plate detection.

1.4 Software and Data Availability

The software that is to achieve the results in this thesis, the experimental data, as well as the various tools developed to visualize and analyze them, are available on the web at <http://www3.itu.edu.tr/~bkurt/dissertation> as MATLAB file and MS Visual C++ project. These files and projects are tested on MATLAB R2007a and MS Visual Studio 6. Electronic copy of the thesis in PDF format is available at the following URL: <http://www3.itu.edu.tr/~bkurt/dissertation/thesis.pdf>.

1.5 Organization of the Thesis

This thesis is organized as six chapters. We propose two dimensional generalized edge detectors (2D-GED) in Chapter 2. Classical implementation of 2D-GED has iterative nature. The main drawback of iterative approach is that the computational cost is highly dependent on the scale parameters (λ and τ). We also investigated a recursive implementation where computational cost is constant. Recursive implementation is presented also in Chapter 2. Surface and edge representation presented in Chapter 2 has a linear nature which means that the space parameters (λ, τ) are constant throughout the image. We let the space parameter λ to vary through the image respecting the feature directions. Also we apply iterative smoothing scheme in which an image is smoothed at each iteration by controlling the space parameters also in time. The adaptive approach is introduced in Chapter 3. Chapter 4 presents two affine-invariant shape descriptors which could be applied to both binary and grayscale images. The proposed algorithm utilizes gradient based features which are extracted along the object boundaries. We used two-dimensional steerable G -Filters explained in Chapter 2 to obtain gradient information at different orientations and scales, and then aggregate the gradients into a shape signature. Chapter 5 presents a new edge detection formulation of edge detection within a goal-based, regularized, shape-aware framework. This framework provides mechanisms for achieving detection without requiring any thresholding or parameter tuning, also taking the contour geometry into account. Finally, Chapter 6 provides a summary and conclusion.

2. REGULARIZED LINEAR SOLUTION: $\lambda\tau$ -SPACE

In the course of developing the generalized edge detector, the authors in [1] explored the relationship between regularization theory and convolution with filters. Regularization theory is a general framework used to convert an ill-posed problem to well-posed by restricting the class of admissible solutions using the constraints such as smoothness [31, 32]. A problem is said to be well-posed if its solution exists, is unique and continuously depends on the data. Since edge detection violates the continuity requirement of the well-posed problems, the regularization theory could be used to transform it to well-posed problem. This is achieved by imposing constraints on the solution in the variational form such as minimizing a cost functional. Smoothness is used widely as a physical constraint because the coherence of matter tends to give rise to smoothly varying intensity values. In regularization theory, the smoothness can be imposed on the solution by minimizing an energy functional containing derivatives of the solution. Given noisy data $d(x, y)$, the regularized solution $f(x, y)$ can be found by minimizing

$$E_m(f) = \iint (f(x, y) - d(x, y))^2 + \lambda (f_x^2(x, y) + f_y^2(x, y)) d\Omega \quad (2.1)$$

where $E_m(f)$ is the energy functional associated with membrane. Another energy functional used to obtain a regularized solution contains the second derivatives of the solution. In this case, a problem is posed finding an f that minimizes

$$E_p(f) = \iint (f(x, y) - d(x, y))^2 + \lambda (f_{xx}^2(x, y) + 2f_{xy}^2(x, y) + f_{yy}^2(x, y)) d\Omega \quad (2.2)$$

where $E_p(f)$ is the energy functional associated with thin plate. In these functionals, the first term on the right hand side is a measure of closeness of the solution $f(x, y)$ to the data $d(x, y)$, and the second term, called stabilizer, is a measure of the smoothness. The compromise between these two terms is controlled by the regularization parameter λ . It was shown that [12, 33, 34] a multi-scale representation can be obtained by carrying out the smoothing process using regularization instead of filtering with a Gaussian. Each level of resolution is obtained by minimizing, for example, the functional (2.1). The scale parameter, in this case, is the regularization parameter λ , so that the coarse levels of representation are obtained by minimizing the membrane functional with a large λ , while fine levels are obtained by using a small λ . A multi-scale edge representation can be

obtained by locating the local maxima in the magnitude of the gradient of these regularized solutions. The relationship between the regularization and filtering process is established by solving the Euler-Lagrange equation associated with the functional. Let us consider the membrane functional. The associated Euler-Lagrange equation will be as follows

$$f - \lambda \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right) = d \quad (2.3)$$

with the boundary conditions $\lim_{x,y \rightarrow \pm\infty} f(x,y) = 0$. Suppose that the solution of the PDE given above is denoted by the function $R(x,y)$ when $d(x,y) = \delta(x,y)$. For given any $d(x,y)$, the solution $f(x,y)$ can be easily determined by taking the convolution as follows

$$f(x,y) = d(x,y) * R(x,y) \quad (2.4)$$

The properties of R-filters for membrane and thin plate functionals will be studied later in detail. The goal of the hybrid model is to provide the smooth transitions between these two groups of filters. For this purpose, linear combination of the membrane and the thin plate functionals is obtained as follows

$$E_h(f) = \iint (f(x,y) - d(x,y))^2 + \lambda \left[\begin{aligned} &(1 - \tau) (f_x^2(x,y) + f_y^2(x,y)) + \\ &\tau (f_{xx}^2(x,y) + 2f_{xy}^2(x,y) + f_{yy}^2(x,y)) \end{aligned} \right] d\Omega \quad (2.5)$$

where λ is the real-valued regularization parameter and $\tau \in [0,1]$ is the real-valued continuity control parameter. Here, the hybrid functional is used to obtain a smooth transition from membrane model to the plate model. Note that, for $\tau = 0$, the functional reduces to the membrane model, for $\tau = 1$ it reduces to the plate model, and for intermediate values of this parameter we obtain hybrid surfaces. In order to find a smooth transition between two filter groups, we need the solution of the Euler-Lagrange equation associated to the hybrid functional. In the next section, the R-filter corresponding to the hybrid functional will be derived and analyzed.

2.1 Derivation of Two Dimensional R- and G-Filter

We now consider the hybrid energy functional and try to find out a function $f(x,y)$ minimizing the functional. As a first step, we obtain the Euler-Lagrange equation

associated with the hybrid functional

$$\begin{aligned} f(x, y) - B(f_{xx}(x, y) + f_{yy}(x, y)) + \\ A(f_{xxx}(x, y) + 2f_{xxy}(x, y) + f_{yyy}(x, y)) = d(x, y) \end{aligned} \quad (2.6)$$

on $\Omega = \{(x, y) | x \geq 0, y \geq 0\}$ where $A = \lambda\tau$ and $B = \lambda(1 - \tau)$ with the mixed boundary conditions

$$\begin{aligned} \lim_{x, y \rightarrow \infty} f(x, y) &= 0, \\ \lim_{x, y \rightarrow \infty} \frac{\partial f(x, y)}{\partial x} &= 0, \quad \lim_{x, y \rightarrow \infty} \frac{\partial f(x, y)}{\partial y} = 0, \\ \lim_{x, y \rightarrow 0} \frac{\partial f(x, y)}{\partial x} &= 0, \quad \lim_{x, y \rightarrow 0} \frac{\partial f(x, y)}{\partial y} = 0, \end{aligned} \quad (2.7)$$

and

$$\begin{aligned} \iint_{\Omega} f(x, y) dx dy &= \frac{1}{4}, \\ f(x, y) &= f(y, x). \end{aligned} \quad (2.8)$$

The partial differential equation (2.6) can be written in terms of differential operators as

$$\begin{aligned} \mathcal{L}f &= d, \\ L &= A(D_{xxx} + 2D_{xxy} + D_{yyy}) - B(D_{xx} + D_{yy}) + 1 \end{aligned} \quad (2.9)$$

where $D_{xx} = \frac{\partial^2}{\partial x^2}$, $D_{yy} = \frac{\partial^2}{\partial y^2}$, $D_{xxx} = \frac{\partial^3}{\partial x^3}$, $D_{xxy} = \frac{\partial^3}{\partial x^2 \partial y}$, and $D_{yyy} = \frac{\partial^3}{\partial y^3}$. Note that $D_{xxx} + 2D_{xxy} + D_{yyy} = (D_{xx} + D_{yy})(D_{xx} + D_{yy}) = (D_{xx} + D_{yy})^2$. The differential operator, $\mathcal{L}$, becomes

$$\mathcal{L} = A(D_{xx} + D_{yy})^2 - B(D_{xx} + D_{yy}) + 1 \quad (2.10)$$

We may express (2.10) as

$$\begin{aligned} \mathcal{L} &= A\mathcal{L}_1\mathcal{L}_2 = A(D_{xx} + D_{yy} - \eta)(D_{xx} + D_{yy} - \xi) \\ \mathcal{L}_1 &= (D_{xx} + D_{yy} - \eta) \\ \mathcal{L}_2 &= (D_{xx} + D_{yy} - \xi) \end{aligned} \quad (2.11)$$

where η and ξ are the roots of the polynomial, $As^2 - Bs + 1$. Here, the operators $\mathcal{L}_1$ and $\mathcal{L}_2$ are known as the two-dimensional diffusion equation. Five cases arise with respect to $A = 0$, $B = 0$, and the sign of $\Delta = B^2 - 4A$. The values of η and ξ are given in Table.1 for these cases. The solutions (R -filter) corresponding to each case are summarized in Table.2. The detail of the derivations is given in the following subchapters.

2.1.1 Case I: $\Delta > 0$

The characteristic polynomial is given by $As^2 - Bs + 1$ according to (2.10). The roots of the polynomial are both real: $s_{1,2} = \frac{B \pm \sqrt{\Delta}}{2A}$. The corresponding differential operators for these roots are given as follows

$$\mathcal{L} = A\mathcal{L}_1\mathcal{L}_2 = A \left(D_{xx} + D_{yy} - \frac{B + \sqrt{\Delta}}{2A} \right) \left(D_{xx} + D_{yy} - \frac{B - \sqrt{\Delta}}{2A} \right) \quad (2.12)$$

$$\begin{aligned} \mathcal{L}_1 &= \left(D_{xx} + D_{yy} - \frac{B + \sqrt{\Delta}}{2A} \right) \\ \mathcal{L}_2 &= \left(D_{xx} + D_{yy} - \frac{B - \sqrt{\Delta}}{2A} \right) \end{aligned} \quad (2.13)$$

The equation (2.12) has a solution of the form

$$R(x, y) = P(x)Q(y) + S(x)T(y) \quad (2.14)$$

where

$$\begin{aligned} a &= \frac{B + \sqrt{\Delta}}{2A}, \\ b &= \frac{B - \sqrt{\Delta}}{2A}. \end{aligned} \quad (2.15)$$

Note that a and b are both positive constants since $B > \sqrt{B^2 - 4A}$. Plugging the equation (2.15) into (2.14) and dividing the both sides to $P(x)Q(y)$ yields

$$\begin{aligned} \frac{1}{P} \frac{d^2 P}{dx^2} + \frac{1}{Q} \frac{d^2 Q}{dy^2} &= a^2, \\ \frac{1}{S} \frac{d^2 S}{dx^2} + \frac{1}{T} \frac{d^2 T}{dy^2} &= b^2. \end{aligned} \quad (2.16)$$

The left hand equation depends only on x and right hand side depends only on y . Since x and y are independent variables, two ratios must equal to some constants

$$\begin{aligned} \underbrace{\frac{1}{P} \frac{d^2 P}{dx^2}}_{constant} &= a^2 - \underbrace{\frac{1}{Q} \frac{d^2 Q}{dy^2}}_{constant}, \\ \frac{1}{P} \frac{d^2 P}{dx^2} &= \pm \alpha^2, \\ \frac{1}{Q} \frac{d^2 Q}{dy^2} &= \pm \beta^2, \\ \pm \alpha^2 \mp \beta^2 &= a^2. \end{aligned} \quad (2.17)$$

We now see that the original problem has been reduced to the solving of two ordinary differential equations. In order to determine the non-trivial solution of $R(x, y)$, we need to

find the values of α and β for which the mixed boundary value problem (2.12)-(2.15) has a non-trivial solution. We consider three cases.

- Case 1: $\alpha^2 - \beta^2 = a^2$

Here the roots of the auxiliary equations are $\pm\alpha, \pm\beta i$. So a general solution is of the following form: $(C_1 e^{-\alpha x} + C_2 e^{\alpha x})(D_1 \cos(\beta y) + D_2 \sin(\beta y))$. The boundary condition (2.7) enforces the solution to be zero at infinity $\lim_{x,y \rightarrow \infty} R(x, y) = 0$. This is true only when $C_2 = D_1 = D_2 = 0$. So there is no non-trivial solution for this case.

- Case 2: $-\alpha^2 + \beta^2 = a^2$

The conclusion from the **Case 1** also applies to this case.

- Case 3: $\alpha^2 + \beta^2 = a^2$

The roots of the auxiliary equations are $\pm\alpha, \pm\beta$. A general solution is of the following form: $(C_1 e^{-\alpha x} + C_2 e^{\alpha x})(D_1 e^{-\beta y} + D_2 e^{\beta y})$. The boundary conditions (2.7) give $C_2 = D_2 = 0$. Therefore the non-trivial solutions (eigenfunctions) corresponding to the eigenvalues are given by

$$\begin{aligned} P(x) Q(y) &= e^{-(\alpha x + \beta y)}, \\ S(x) T(y) &= e^{-(\mu x + \nu y)}. \end{aligned} \quad (2.18)$$

Combining the equation (2.18) with the equation (2.14) we obtain the functions

$$R(x, y) = K_1 e^{-(\alpha x + \beta y)} + K_2 e^{-(\mu x + \nu y)} \quad (2.19)$$

which are solutions of the fundamental partial differential equation (2.12). The solution must satisfy the following boundary condition

$$\begin{aligned} \lim_{x,y \rightarrow 0} \frac{\partial f(x,y)}{\partial x} &= 0, \quad \lim_{x,y \rightarrow 0} \frac{\partial f(x,y)}{\partial y} = 0, \\ \alpha K_1 + \mu K_2 &= 0, \\ \beta K_1 + \nu K_2 &= 0. \end{aligned} \quad (2.20)$$

The general solution now becomes

$$\begin{aligned} R(x, y) &= K (\mu \nu e^{-(\alpha x + \beta y)} - \alpha \beta e^{-(\mu x + \nu y)}), \\ \alpha^2 + \beta^2 &= a^2, \\ \mu^2 + \nu^2 &= b^2. \end{aligned} \quad (2.21)$$

Since we are looking for a solution such that $R(x, y) = R(y, x)$, then $\alpha = \beta = \frac{a}{\sqrt{2}}$ and $\mu = \nu = \frac{b}{\sqrt{2}}$. Finally the solution becomes

$$R(x, y) = \frac{K}{\sqrt{2}} \left(b e^{-\frac{a(x+y)}{\sqrt{2}}} - a e^{-\frac{b(x+y)}{\sqrt{2}}} \right). \quad (2.22)$$

K can be determined from the following integral

$$\begin{aligned} \iint f(x, y) &= 1, \\ \frac{K}{\sqrt{2}} \int_0^\infty \int_0^\infty b e^{-\frac{a(x+y)}{\sqrt{2}}} - a e^{-\frac{b(x+y)}{\sqrt{2}}} &= 1, \\ K &= \frac{ab}{b^2 - a^2}. \end{aligned} \quad (2.23)$$

The only non-trivial solution for the **Case I** is

$$\begin{aligned} R(x, y) &= \frac{1}{\sqrt{2}} \frac{ab}{b^2 - a^2} \left(b e^{-\frac{a(x+y)}{\sqrt{2}}} - a e^{-\frac{b(x+y)}{\sqrt{2}}} \right) \\ R(x, y) &= \frac{1}{\left(\frac{1}{a/\sqrt{2}} \right)^2 - \left(\frac{1}{b/\sqrt{2}} \right)^2} \left(\frac{1}{a/\sqrt{2}} e^{-\frac{a}{\sqrt{2}}(x+y)} - \frac{1}{b/\sqrt{2}} e^{-\frac{b}{\sqrt{2}}(x+y)} \right) \end{aligned} \quad (2.24)$$

2.1.2 Case II: $\Delta = 0$

The characteristic polynomial is given by $As^2 - Bs + 1$ according to (2.10). The roots of the polynomial are both real: $s_1 = s_2 = \frac{B}{2A}$. The corresponding differential operators for these roots are given as follows

$$\mathcal{L} = A \left(D_{xx} + D_{yy} - \frac{B}{2A} \right)^2 \quad (2.25)$$

The equation (2.25) has a solution of the form

$$R(x, y) = (M_1 x + M_0) P(x) (N_1 y + N_0) Q(y) \quad (2.26)$$

Plugging the equation (2.26) into (2.25) and dividing the both sides to $P(x)Q(y)$ yields

$$\frac{1}{P} \frac{d^2 P}{dx^2} + \frac{1}{Q} \frac{d^2 Q}{dy^2} = a^2 \quad (2.27)$$

where $a = \sqrt{\frac{B}{2A}}$. The left hand equation depends only on x and right hand side depends only on y . Since x and y are independent variables, two ratios must equal some constants.

$$\begin{aligned} \underbrace{\frac{1}{P} \frac{d^2 P}{dx^2}}_{\text{constant}} &= a^2 - \underbrace{\frac{1}{Q} \frac{d^2 Q}{dy^2}}_{\text{constant}}, \\ \frac{1}{P} \frac{d^2 P}{dx^2} &= \pm \alpha^2, \\ \frac{1}{Q} \frac{d^2 Q}{dy^2} &= \pm \beta^2, \\ \pm \alpha^2 \mp \beta^2 &= a^2. \end{aligned} \tag{2.28}$$

We now see that the original problem has been reduced to the solving of two ordinary differential equations. In order to determine the non-trivial solution of $R(x, y)$, we need to find the values of α and β for which the mixed boundary value problem (2.12)-(2.15) has a non-trivial solution. We consider three cases.

- Case 1: $\alpha^2 - \beta^2 = a^2$

The roots of the auxiliary equations are $\pm \alpha, \pm \beta i$. So a general solution is of the following form: $(C_1 e^{-\alpha x} + C_2 e^{\alpha x}) (D_1 \cos(\beta y) + D_2 \sin(\beta y))$. The boundary condition (2.7) enforces the solution to be zero at infinity $\lim_{x, y \rightarrow \infty} R(x, y) = 0$. This is true only when $C_2 = D_1 = D_2 = 0$. So there is no non-trivial solution for this case.

- Case 2: $-\alpha^2 + \beta^2 = a^2$

The conclusion from the **Case 1** also applies to this case.

- Case 3: $\alpha^2 + \beta^2 = a^2$

The roots of the auxiliary equations are $\pm \alpha, \pm \beta$. A general solution is of the following form: $(C_1 e^{-\alpha x} + C_2 e^{\alpha x}) (D_1 e^{-\beta y} + D_2 e^{\beta y})$. The boundary conditions (2.7) give $C_2 = D_2 = 0$. Therefore the non-trivial solutions (eigenfunctions) corresponding to the eigenvalues are given by

$$P(x) Q(y) = e^{-(\alpha x + \beta y)} \tag{2.29}$$

Since we are looking for a solution such that $R(x, y) = R(y, x)$, then $\alpha = \beta = \frac{a}{\sqrt{2}}$, and $M_1 = N_1, M_0 = N_0$. The solution must also satisfy the following boundary condition

$$\lim_{x, y \rightarrow 0} \frac{\partial f(x, y)}{\partial x} = 0, \lim_{x, y \rightarrow 0} \frac{\partial f(x, y)}{\partial y} = 0 \tag{2.30}$$

$$\begin{aligned}
\lim_{x,y \rightarrow 0} \frac{\partial f(x,y)}{\partial x} &= \lim_{x,y \rightarrow 0} \left(\left(M_1 (M_1 y + M_0) - \frac{a}{\sqrt{2}} (M_1 x + M_0) (M_1 y + M_0) \right) e^{-\frac{a}{\sqrt{2}}(x+y)} \right) \\
&= M_1 M_0 - \frac{a}{\sqrt{2}} M_0^2 = 0 \\
M_1 &= \frac{a}{\sqrt{2}} M_0
\end{aligned} \tag{2.31}$$

Finally the solution becomes

$$R(x, y) = M_0^2 \left(\frac{a}{\sqrt{2}} x + 1 \right) \left(\frac{a}{\sqrt{2}} y + 1 \right) e^{-\frac{a}{\sqrt{2}}(x+y)} \tag{2.32}$$

M_0^2 can be determined from the following integral equation

$$\begin{aligned}
\int_0^\infty \int_0^\infty R(x, y) dx dy &= 1, \\
M_0^2 \int_0^\infty \int_0^\infty \left(\frac{a}{\sqrt{2}} x + 1 \right) \left(\frac{a}{\sqrt{2}} y + 1 \right) e^{-\frac{a}{\sqrt{2}}(x+y)} &= 1, \\
M_0^2 \left(\frac{a+\sqrt{2}}{a} \right)^2 &= 1.
\end{aligned} \tag{2.33}$$

The only non-trivial solution for the **Case II** is

$$R(x, y) = \frac{\frac{a}{\sqrt{2}}}{\frac{a}{\sqrt{2}} + 1} \left(\frac{a}{\sqrt{2}} \|x\| + 1 \right) \left(\frac{a}{\sqrt{2}} \|y\| + 1 \right) e^{-\frac{a}{\sqrt{2}}(\|x\| + \|y\|)} \tag{2.34}$$

2.1.3 Case III: $\Delta < 0$

The characteristic polynomial is given by $As^2 - Bs + 1$ according to (2.10). The roots of the polynomial are both real: $s_{1,2} = \frac{B \pm i\sqrt{|\Delta|}}{2A}$. The corresponding differential operators for these roots are given as follows

$$\begin{aligned}
\mathcal{L} &= A\mathcal{L}_1\mathcal{L}_2 = A \left(D_{xx} + D_{yy} - \frac{B+i\sqrt{|\Delta|}}{2A} \right) \left(D_{xx} + D_{yy} - \frac{B-i\sqrt{|\Delta|}}{2A} \right), \\
\mathcal{L}_1 &= \left(D_{xx} + D_{yy} - \frac{B+i\sqrt{|\Delta|}}{2A} \right), \\
\mathcal{L}_2 &= \left(D_{xx} + D_{yy} - \frac{B-i\sqrt{|\Delta|}}{2A} \right).
\end{aligned} \tag{2.35}$$

The equation (2.12) has a solution of the form

$$R(x, y) = P(x) Q(y) + S(x) T(y) \tag{2.36}$$

where

$$\begin{aligned} a^2 &= \frac{B+i\sqrt{|\Delta|}}{2A}, \\ b^2 &= \frac{B-i\sqrt{|\Delta|}}{2A}. \end{aligned} \quad (2.37)$$

Plugging the equation (2.36) into (2.35) and dividing the both sides to yields

$$\begin{aligned} \frac{1}{P} \frac{d^2 P}{dx^2} + \frac{1}{Q} \frac{d^2 Q}{dy^2} &= a^2, \\ \frac{1}{S} \frac{d^2 S}{dx^2} + \frac{1}{T} \frac{d^2 T}{dy^2} &= b^2. \end{aligned} \quad (2.38)$$

The left hand equation depends only on x and right hand side depends only on y . Since x and y are independent variables, two ratios must be equal to some constants.

$$\begin{aligned} \underbrace{\frac{1}{P} \frac{d^2 P}{dx^2}}_{\text{constant}} &= a^2 - \underbrace{\frac{1}{Q} \frac{d^2 Q}{dy^2}}_{\text{constant}}, \\ \frac{1}{P} \frac{d^2 P}{dx^2} &= \alpha^2, \\ \frac{1}{Q} \frac{d^2 Q}{dy^2} &= \beta^2, \\ \alpha^2 + \beta^2 &= a^2. \end{aligned} \quad (2.39)$$

We now see that the original problem has been reduced to the solving of two ordinary differential equations. In order to determine the non-trivial solution of $R(x, y)$, we need to find the values of α and β for which the mixed boundary value problem (2.10)-(2.8) has a non-trivial solution. Since a is a complex number, the roots of the auxiliary equations, α and β , are both complex numbers. So the general solution is of the following form

$$\begin{aligned} P(x)Q(y) = & \left(\begin{aligned} &C_1 e^{-\|\alpha\| \cos(\varphi)x} (M_1 \cos(\sin(\varphi) \|\alpha\| x) - iM_2 \sin(\sin(\varphi) \|\alpha\| x)) + \\ &C_2 e^{\|\alpha\| \cos(\varphi)x} (V_1 \cos(\sin(\varphi) \|\alpha\| x) + iV_2 \sin(\sin(\varphi) \|\alpha\| x)) \end{aligned} \right) \\ & \left(\begin{aligned} &D_1 e^{-\|\beta\| \cos(\phi)y} (N_1 \cos(\sin(\phi) \|\beta\| y) - iN_2 \sin(\sin(\phi) \|\beta\| y)) + \\ &D_2 e^{\|\beta\| \cos(\phi)y} (W_1 \cos(\sin(\phi) \|\beta\| y) + iW_2 \sin(\sin(\phi) \|\beta\| y)) \end{aligned} \right) \end{aligned} \quad (2.40)$$

where $(\|\alpha\|, \varphi)$ and $(\|\beta\|, \phi)$ are the polar coordinate representation of α and β . The boundary condition (2.7) enforces the solution to be zero at infinity, $\lim_{x,y \rightarrow \infty} R(x, y) = 0$. This is true only when $C_2 = D_2 = 0$. Therefore the non-trivial solutions (eigenfunctions) corresponding to the eigenvalues are given by

$$\begin{aligned} P(x)Q(y) &= K e^{-(\|\alpha\| \cos(\varphi)x + \|\beta\| \cos(\phi)y)} \\ &\times (M_1 \cos(\sin(\varphi) \|\alpha\| x) - iM_2 \sin(\sin(\varphi) \|\alpha\| x)) \\ &\times (N_1 \cos(\sin(\phi) \|\beta\| y) - iN_2 \sin(\sin(\phi) \|\beta\| y)) \end{aligned} \quad (2.41)$$

Since we are looking for a solution such that $f(x, y) = f(y, x)$, then $(\|\alpha\|, \phi) = (\|\beta\|, \varphi)$, $\|\alpha\| = \|\beta\| = \frac{1}{2} \sqrt[4]{\frac{B^2 + |\Delta|}{A^2}}$, $\phi = \varphi = \frac{1}{2} \arctan\left(\frac{\sqrt{\Delta}}{B}\right)$, $M_1 = N_1, M_2 = N_2$.

$$P(x)Q(y) = Ke^{-\|\alpha\| \cos(\varphi)(x+y)} \begin{pmatrix} (\cos(\sin(\phi)\|\alpha\|x) \cos(\sin(\phi)\|\alpha\|y) - \\ (\sin(\sin(\phi)\|\alpha\|x) \sin(\sin(\phi)\|\alpha\|y)) \end{pmatrix} \quad (2.42)$$

Then $R(x, y)$ becomes

$$R(x, y) = K_1 e^{-\|\alpha\| \cos(\varphi)(x+y)} \cos(\sin(\phi)\|\alpha\|(x+y)) \quad (2.43)$$

We find $S(x)T(y)$ following similar steps given above as

$$S(x)T(y) = K_2 e^{-\|\alpha\| \cos(\varphi)(x+y)} \sin(\sin(\phi)\|\alpha\|(x+y)) \quad (2.44)$$

The solution must satisfy the following boundary condition

$$\lim_{x,y \rightarrow 0} \frac{\partial R(x,y)}{\partial x} = 0, \quad \lim_{x,y \rightarrow 0} \frac{\partial R(x,y)}{\partial y} = 0 \quad (2.45)$$

Combining the equation (2.41) with the equation (2.36) we obtain the functions

$$R(x, y) = Ke^{-\|\alpha\| \cos(\varphi)(x+y)} (\cos(\sin(\phi)\|\alpha\|(x+y)) + \sin(\sin(\phi)\|\alpha\|(x+y))) \quad (2.46)$$

which are solutions of the fundamental partial differential equation (2.35). K can be determined from the following integral

$$\int_0^\infty \int_0^\infty R(x, y) dx dy = 0.25, \quad (2.47)$$

The only non-trivial solution for the **Case III** is

$$R(x, y) = K \exp\left(-\frac{\cos(\theta)(|x|+|y|)}{\sqrt[4]{4A}}\right) (\cos(\theta) \cos(\varphi(|x|+|y|)) + \sin(\theta) \sin(\varphi(|x|+|y|))). \quad (2.48)$$

where

$$\begin{aligned} K &= \frac{4}{\sqrt{A} \cos^3(\theta) \sin^2(2\theta)}, \\ 2\theta &= \arctan \frac{\sqrt{|\Delta|}}{B}, \\ \varphi &= \frac{1}{\sqrt[4]{4A}} \sin(\theta). \end{aligned} \quad (2.49)$$

2.1.4 Case IV: $A = 0$

The characteristic polynomial is given by $1 - Bs$ according to (2.10). The root of the polynomial is $s = \frac{1}{B}$. The corresponding differential operators for these roots are given as follows

$$\mathcal{L} = (-B(D_{xx} + D_{yy}) + 1) \quad (2.50)$$

The equation (2.50) has a solution of the form

$$R(x, y) = P(x)Q(y) \quad (2.51)$$

Plugging the equation (2.51) into (2.50) and dividing the both sides to $P(x)Q(y)$ yields

$$\frac{1}{P} \frac{d^2 P}{dx^2} + \frac{1}{Q} \frac{d^2 Q}{dy^2} = a^2 \quad (2.52)$$

where $a = \sqrt{\frac{1}{B}}$. The left hand equation depends only on x and right hand side depends only on y . Since x and y are independent variables, two ratios must equal some constants.

$$\begin{aligned} \underbrace{\frac{1}{P} \frac{d^2 P}{dx^2}}_{\text{constant}} &= a^2 - \underbrace{\frac{1}{Q} \frac{d^2 Q}{dy^2}}_{\text{constant}}, \\ \frac{1}{P} \frac{d^2 P}{dx^2} &= \pm \alpha^2, \\ \frac{1}{Q} \frac{d^2 Q}{dy^2} &= \pm \beta^2, \\ \pm \alpha^2 \mp \beta^2 &= a^2. \end{aligned} \quad (2.53)$$

We now see that the original problem has been reduced to the solving of two ordinary differential equations. In order to determine the non-trivial solution of $R(x, y)$, we need to find the values of α and β for which the mixed boundary value problem (2.12)-(2.15) has a non-trivial solution. We consider three cases.

- Case 1: $\alpha^2 - \beta^2 = a^2$

The roots of the auxiliary equations are $\pm \alpha, \pm \beta i$. So a general solution is of the following form: $(C_1 e^{-\alpha x} + C_2 e^{\alpha x})(D_1 \cos(\beta y) + D_2 \sin(\beta y))$. The boundary

condition (2.7) enforces the solution to be zero at infinity $\lim_{x,y \rightarrow \infty} R(x, y) = 0$. This is true only when $C_2 = D_1 = D_2 = 0$. So there is no non-trivial solution for this case.

- Case 2: $-\alpha^2 + \beta^2 = a^2$

The conclusion from the **Case 1** also applies to this case.

- Case 3: $\alpha^2 + \beta^2 = a^2$

The roots of the auxiliary equations are $\pm\alpha, \pm\beta$. A general solution is of the following form: $(C_1 e^{-\alpha x} + C_2 e^{\alpha x})(D_1 e^{-\beta y} + D_2 e^{\beta y})$. The boundary conditions (2.7) give $C_2 = D_2 = 0$. Therefore the non-trivial solutions (eigenfunctions) corresponding to the eigenvalues are given by

$$P(x)Q(y) = K e^{-(\alpha x + \beta y)} \quad (2.54)$$

Since we are looking for a solution such that $R(x, y) = R(y, x)$, then $\alpha = \beta = \frac{a}{\sqrt{2}}$. The solution must also satisfy the following integral equation

$$\begin{aligned} \iint f(x, y) &= 1, \\ K \int_0^\infty \int_0^\infty e^{-\frac{a}{\sqrt{2}}(x+y)} &= 1, \\ K &= \frac{2}{a^2}. \end{aligned} \quad (2.55)$$

Finally the solution becomes

$$R(x, y) = M_0^2 \left(\frac{a}{\sqrt{2}}x + 1 \right) \left(\frac{a}{\sqrt{2}}y + 1 \right) e^{-\frac{a}{\sqrt{2}}(x+y)} \quad (2.56)$$

The only non-trivial solution for the **Case IV** is

$$R(x, y) = 2B \exp\left(-\sqrt{\frac{1}{2B}}(|x| + |y|)\right) \quad (2.57)$$

2.1.5 Case V: $B = 0$

Case V is a special case of Case III where $\theta = \frac{\pi}{2}$. The solution is simply written as

$$\begin{aligned} R(x, y) &= K \exp\left(-\frac{(|x|+|y|)}{2\sqrt{A}}\right) \\ &(\cos(\varphi(|x| + |y|)) + \sin(\varphi(|x| + |y|))). \end{aligned} \quad (2.58)$$

Table 2.1: Two Dimensional $R(x, y; \lambda, \tau)$ -filters.

Case		$R(x, y)$
I	$\Delta > 0$	$\frac{2A}{\sqrt{\Delta}} [a \exp(-b(x + y)) - b \exp(-a(x + y))],$ $a = \sqrt{\frac{B + \sqrt{\Delta}}{4A}},$ $b = \sqrt{\frac{B - \sqrt{\Delta}}{4A}}$
II	$\Delta < 0$	$K \exp\left(-\frac{1}{\sqrt[4]{4A}} \cos(\theta)(x + y)\right) \times$ $(\cos(\theta) \cos(\varphi(x + y)) + \sin(\theta) \sin(\varphi(x + y)))$ $K = \frac{4}{\sqrt{A} \cos^3(\theta) \sin^2(2\theta)},$ $2\theta = \arctan \frac{\sqrt{ \Delta }}{B},$ $\varphi = \frac{1}{\sqrt[4]{4A}} \sin(\theta)$
III	$\Delta = 0$	$\frac{1}{4B} \left(\frac{1}{\sqrt{B}} x + 1\right) \left(\frac{1}{\sqrt{B}} y + 1\right) \exp\left(-\sqrt{\frac{1}{B}}(x + y)\right)$
IV	$A = 0$	$2B \exp\left(-\sqrt{\frac{1}{2B}}(x + y)\right)$
V	$B = 0$	$\frac{8}{\sqrt{\frac{A}{2}}} \exp\left(-\frac{(x + y)}{2\sqrt[4]{A}}\right) (\cos(\varphi(x + y)) + \sin(\varphi(x + y)))$ $\varphi = \frac{1}{2\sqrt[4]{A}}$

where

$$\begin{aligned}
K &= \frac{8}{\sqrt{A}}, \\
\varphi &= \frac{1}{2\sqrt[4]{A}}.
\end{aligned} \tag{2.59}$$

2.2 Smoothing Filter: R -Filter

In the previous section we derived two dimensional smoothing filters by solving several partial differential equations of different types. These filters are summarized in Table 2.1. R -filters are not circular symmetric on contrary to the Gaussian filter. The level set geometry of the filters is given in the Fig. 2.1. This causes the response to the R -filter at the local features such as edge primitive, line or corner varies with the orientation. Fig. 2.4 shows the R -filter ($R(x, y; \lambda, \tau)$) for the specified values of λ and τ including all five cases. The parameter values and the corresponding cases are shown in Table 2.4. Although only nine discrete values of the parameters are used, the above formulation allows us to obtain the $R(x, y; \lambda, \tau)$ filters as λ and τ values vary continuously in the range $\lambda > 0$ and $0 \leq \tau \leq 1$. Fig. 2.3 shows a sketch of the $\lambda\tau$ -space, where the curves specified by $\lambda\tau/(1 - \tau)^2 = k$ are shown for various values of k . The thick curve with $\lambda\tau/(1 - \tau)^2 = 4$ derived from $\Delta = 0$ divides the space, limited by $\tau = 0$ (the membrane model) and $\tau = 1$ (the plate model) lines, into two regions, say Regions I and II. The analysis of this space reveals that the shape of the filter does not change as we move along any $\lambda\tau/(1 - \tau)^2 = \text{constant}$ curve. On the other hand, moving across these curves will

Table 2.2: Two Dimensional Discrete $R(m, n)$ -filters.

Case		$R(m, n)$
I	$\Delta > 0$	$H(a \exp(-b(m + n)) - b \exp(-a(m + n))),$ $H = \frac{a(1-e^{-a})^2(1+e^{-b})^2 - b(1+e^{-a})^2(1-e^{-b})^2}{(1-e^{-a})^2(1-e^{-b})^2}$ $a = \sqrt{\frac{B+\sqrt{\Delta}}{4A}},$ $b = \sqrt{\frac{B-\sqrt{\Delta}}{4A}}$
II	$\Delta < 0$	$H(\cos(\theta) \cos(\varphi(m + n)) + \sin(\theta) \sin(\varphi(m + n)))$ $\times \exp(-(\frac{\cos(\theta)(m + n)}{\sqrt[4]{4A}}))$ $H = \frac{4}{\sqrt{A} \cos^3(\theta) \sin^2(2\theta)},$ $2\theta = \arctan\left(\frac{\sqrt{ \Delta }}{B}\right), \varphi = \frac{1}{\sqrt[4]{4A}} \sin(\theta)$
III	$\Delta = 0$	$H(\frac{1}{\sqrt{B}} m + 1)(\frac{1}{\sqrt{B}} n + 1)e^{-\frac{1}{\sqrt{B}}(m + n)}$ $H = \frac{\left(1 - e^{-\frac{1}{\sqrt{B}}}\right)^4}{\left(1 + \frac{2}{\sqrt{B}}e^{-\frac{1}{\sqrt{B}}} - e^{-\frac{2}{\sqrt{B}}}\right)^2}$
IV	$A = 0$	$\left(\frac{1 - \exp(-\frac{1}{\sqrt{2B}})}{1 + \exp(-\frac{1}{\sqrt{2B}})}\right)^2 \exp\left(-\left(\frac{ m + n }{\sqrt{2B}}\right)\right)$
V	$B = 0$	$\frac{8}{\sqrt{A}} \exp\left(-\frac{(m + n)}{2\sqrt[4]{A}}\right) (\cos(\varphi(m + n)) + \sin(\varphi(m + n)))$ $\varphi = \frac{1}{2\sqrt[4]{A}}$

Table 2.3: Values of s_1 and s_2 for five cases.

Case		s_1	s_2
I	$\Delta > 0$	$\frac{B+\sqrt{\Delta}}{2A}$	$\frac{B-\sqrt{\Delta}}{2A}$
II	$\Delta < 0$	$\frac{B+i\sqrt{ \Delta }}{2A}$	$\frac{B-i\sqrt{ \Delta }}{2A}$
III	$\Delta = 0$	$\frac{B}{2A}$	$\frac{B}{2A}$
IV	$A = 0$	$\frac{1}{B}$	-
V	$B = 0$	$i\frac{\sqrt{A}}{2}$	$-i\frac{\sqrt{A}}{2}$

considerably change the shape of the corresponding filter. One of the important differences between filters in Region I and in Region II is that the R -filter does not have negative values for any λ and τ in Region I, where as it will have two negative lobes at both sides of the central positive lobe for λ and τ in Region II. These negative values are the primary reasons for the overshoots around discontinuities at the filtered images. Thus one can observe that in Region I, the first order R -filter is gradually transformed to a Gaussian like filter as τ increases while in region II, the Gaussian-like filter is transformed to the second order R -filter as τ increases.

Table 2.4: Filter types associated with the selected parameter values.

	τ		
λ	0.0	0.5	1.0
1	IV	II	I
8	IV	III	I
16	IV	I	I

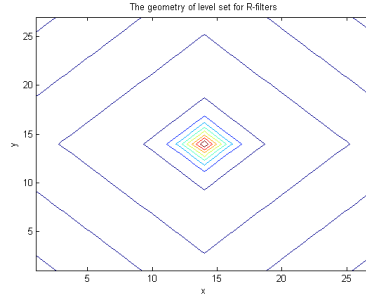
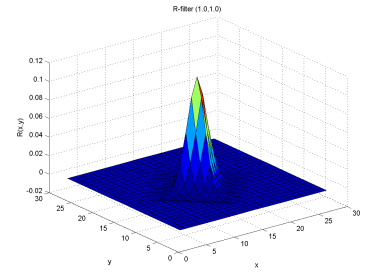
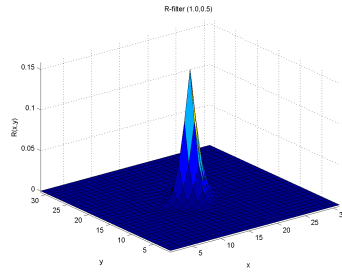
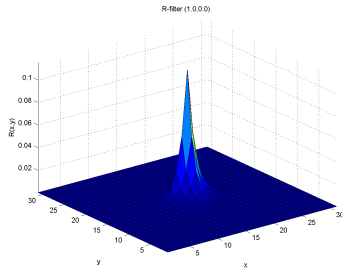
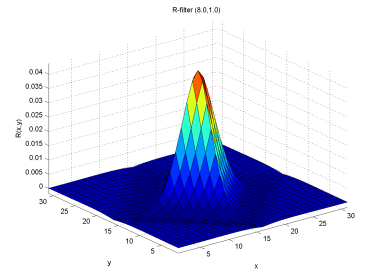
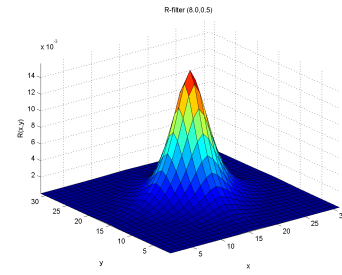
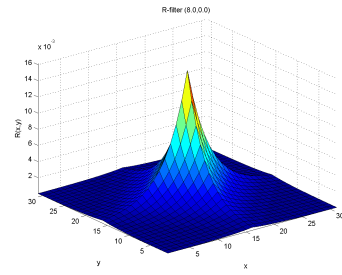


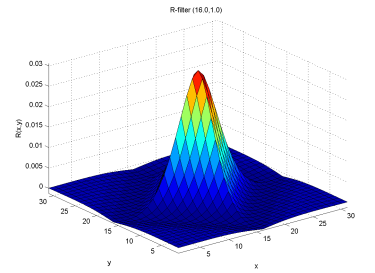
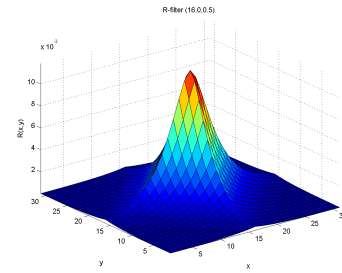
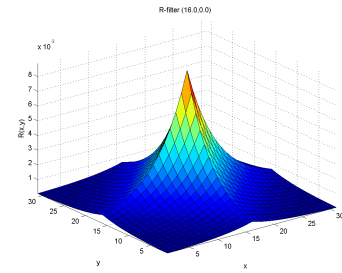
Figure 2.1: The geometry of level set for R -filters



(a) $\lambda = 1.0, \tau = 0.0, 0.5, 1.0$



(b) $\lambda = 8.0, \tau = 0.0, 0.5, 1.0$



(c) $\lambda = 16.0, \tau = 0.0, 0.5, 1.0$

Figure 2.2: R-filters (τ takes the values 0.0, 0.5, and 1.0 from left to right).

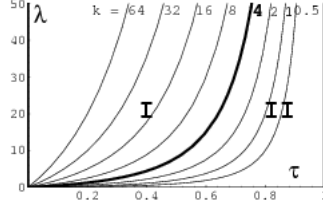


Figure 2.3: $\lambda\tau/(1-\tau)^2 = \text{constant}$ curves in $\lambda\tau$ -space.

Table 2.5: Two Dimensional G-filter along x direction : $G^{(x)}(x, y; \lambda, \tau)$.

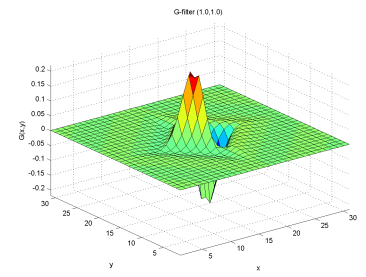
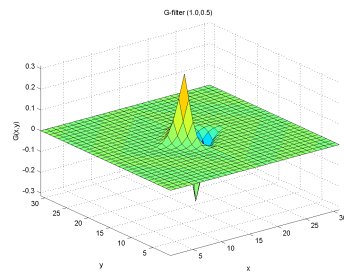
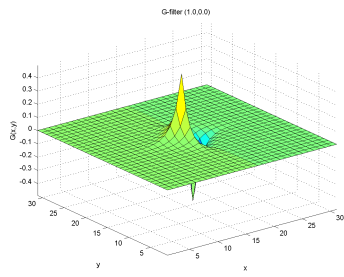
Case		$G^{(x)}(x, y)$
I	$\Delta > 0$	$\frac{2A}{\sqrt{\Delta}} \text{sgn}(x) a b [\exp(-a(x + y)) - \exp(-b(x + y))],$ $a = \sqrt{\frac{B+\sqrt{\Delta}}{4A}},$ $b = \sqrt{\frac{B-\sqrt{\Delta}}{4A}}.$
II	$\Delta < 0$	$-K \frac{1}{\sqrt[4]{4A}} \text{sgn}(x) \exp\left(-\frac{1}{\sqrt[4]{4A}} \cos(\theta)(x + y)\right) \times$ $[\cos(2\theta) \cos(\varphi(x + y)) + \sin(2\theta) \sin(\varphi(x + y))],$ $K = \frac{4}{\sqrt{A} \cos^3(\theta) \sin^2(2\theta)},$ $2\theta = \arctan \frac{\sqrt{ \Delta }}{B},$ $\varphi = \frac{1}{\sqrt[4]{4A}} \sin(\theta).$
III	$\Delta = 0$	$\frac{-1}{4B^2} x \text{sgn}(x) \left(\frac{1}{\sqrt{B}} y + 1\right) \exp\left(-\sqrt{\frac{1}{B}}(x + y)\right)$
IV	$A = 0$	$-\sqrt{2B} \text{sgn}(x) \exp\left(-\sqrt{\frac{1}{2B}}(x + y)\right)$
V	$B = 0$	$\frac{8}{\sqrt{\frac{A}{2}}} \exp\left(-\frac{(x + y)}{2\sqrt[4]{A}}\right) (\cos(\varphi(x + y)) + \sin(\varphi(x + y)))$ $\varphi = \frac{1}{2\sqrt[4]{A}}.$

2.3 Edge Detection Filter: G -Filter

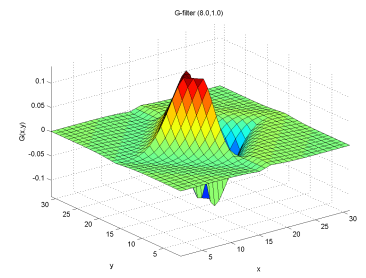
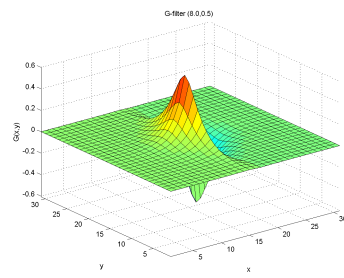
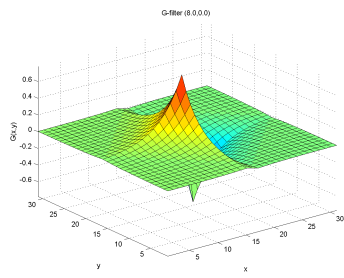
The G -filter is obtained by taking the derivative of R -filter with respect to x and y . G -filter along x direction denoted by $G^x(x, y)$ is given in Table 2.4. Note that the R -filters are symmetric while the G -filters are anti-symmetric. Both filters do not have circular symmetry. In order to implement the scale-space filtering, we discretized the $G(x, y)$ filters in such a way that the discretized form still satisfy that $\sum_m \sum_n \|G(m, n)\| = 1$ and $\sum_m \sum_n G(m, n) = 0$. The discretized version is given in Table. 2.5. Fig. 2.2 shows the G -filter ($G(m, n; \lambda, \tau)$) for the specified values of λ and τ . The parameter values and the corresponding cases are shown in Table 2.4. Although only nine discrete values of the parameters are used, the above formulation allows us to obtain the $G^x(m, n; \lambda, \tau)$ filters as λ and τ values vary continuously in the range $\lambda > 0$ and $0 \leq \tau \leq 1$.

Table 2.6: Two Dimensional Discrete G -filter along x direction : $G^{(x)}(m, n; \lambda, \tau)$.

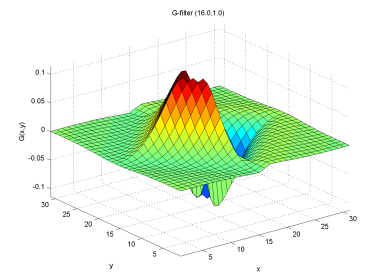
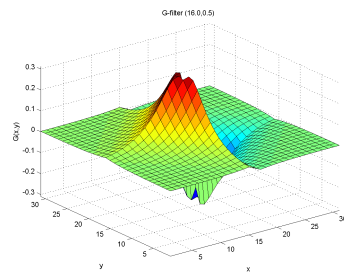
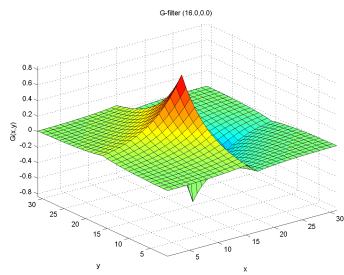
Case		$G^{(x)}(m, n)$
I	$\Delta > 0$	$\frac{2A}{\sqrt{\Delta}} \operatorname{sgn}(m) a b [\exp(-a(m + n)) - \exp(-b(m + n))],$ $a = \sqrt{\frac{B + \sqrt{\Delta}}{4A}},$ $b = \sqrt{\frac{B - \sqrt{\Delta}}{4A}}.$
II	$\Delta < 0$	$-K \frac{1}{\sqrt[4]{4A}} \operatorname{sgn}(m) \exp\left(-\frac{1}{\sqrt[4]{4A}} \cos(\theta)(m + n)\right) \times$ $[\cos(2\theta) \cos(\varphi(m + n)) + \sin(2\theta) \sin(\varphi(m + n))],$ $K = \frac{4}{\sqrt{A} \cos^3(\theta) \sin^2(2\theta)},$ $2\theta = \arctan \frac{\sqrt{ \Delta }}{B},$ $\varphi = \frac{1}{\sqrt[4]{4A}} \sin(\theta).$
III	$\Delta = 0$	$\frac{-1}{4B^2} x \operatorname{sgn}(m) \left(\frac{1}{\sqrt{B}} n + 1\right) \exp\left(-\sqrt{\frac{1}{B}}(m + n)\right)$
IV	$A = 0$	$-\sqrt{2B} \operatorname{sgn}(m) \exp\left(-\sqrt{\frac{1}{2B}}(m + n)\right)$
V	$B = 0$	$\frac{8}{\sqrt{\frac{A}{2}}} \exp\left(-\frac{(m + n)}{2\sqrt[4]{A}}\right) \operatorname{sgn}(m) (\cos(\varphi(m + n)) + \sin(\varphi(m + n)))$ $\varphi = \frac{1}{2\sqrt[4]{A}}.$



(a) $\lambda = 1.0$



(b) $\lambda = 8.0$



(c) $\lambda = 16.0$

Figure 2.4: G-filters (τ takes the values 0.0, 0.5, and 1.0 from left to right).

Table 2.7: The normalized mean square error (NMSE) between the reconstructed images and the original image.

$\tau \Rightarrow$			SNR=12dB			SNR=8.5dB			SNR=5.5dB		
			0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0
Checkerboard	λ	1.0	31.0	26.8	29.5	31.8	28.7	31.5	32.7	30.9	33.6
		8.0	54.5	48.5	35.9	54.8	48.6	36.3	54.9	48.8	36.9
		16.0	64.2	54.5	39.5	65.3	54.5	39.9	75.6	54.7	40.2
Bars	λ	1.0	47.5	35.7	38.7	48.0	37.2	39.8	48.6	38.9	41.5
		8.0	77.2	74.8	54.6	77.3	74.8	55.0	77.3	74.8	55.5
		16.0	84.2	77.7	60.3	84.2	77.7	60.5	84.2	77.8	60.9

2.4 Serial Implementation

In order to obtain a scale space representation and edge detection, one should first discretized the R- and G-Filters. After discretization such that R-filter coefficients sum to unit and G-filter coefficients sum to zero. Does the hybrid model provide any improved results as compared to the membrane and plate functionals? To answer this question we consider a checkerboard image (Fig. 2.5) and then analyze the reconstructed images obtained with $\tau = 0, 0.5$, and 1, which are shown in Fig. 2.6. We see that each model has its own difficulty in reconstructing the original image. On the other hand, the hybrid solution obtained with $\tau = 0.5$ does not suffer from over smoothed discontinuities of the membrane solution or the overshoots of the plate solution. As compared to these two standard models, the hybrid model provides a better replica of the original surface. In order to evaluate the reconstructed surfaces quantitatively, we calculate the mean square error (*nmse*) between the original image and the reconstructed image, shown in Table 2.7. The hybrid model results in a smaller *nmse* as compared to the membrane and plate model for the moderate λ values, while the *nmse* value is the smallest for the membrane model for large λ values in which case none of the results are satisfactory. Besides these quantitative results, the plate model causes overshoots around discontinuities on these surfaces are the major drawbacks of this model. For example, when the reconstructed surfaces are used in classifying the surfaces based on the curvature, these overshoots will create misclassified surface patches around discontinuities.

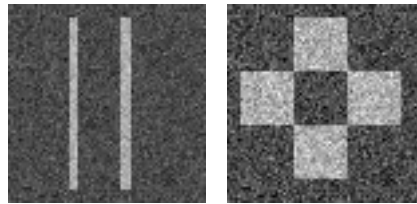


Figure 2.5: The noisy bars and checkerboard images (SNR=12dB)

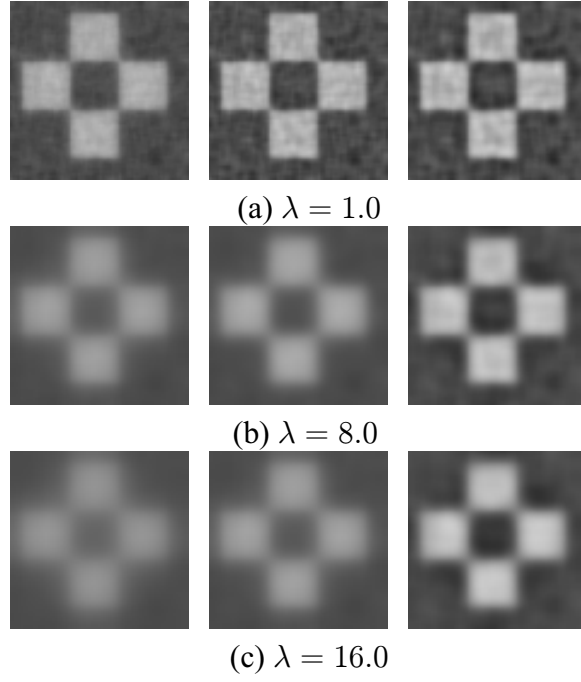


Figure 2.6: $\lambda\tau$ -space Representation of noisy checkerboard image (τ takes the values 0.0, 0.5, and 1.0 from left to right).

Similarly, when a Laplacian-based edge detection algorithm is used, these overshoots will give rise to spurious edges due to the sign change at these points. The solution provided by the hybrid model can be considered as the best choice for these problems. The quantitative and qualitative results show experimentally that improved results may be obtained by means of the hybrid model with $0 < \tau < 1$ and it justifies sweeping of τ values between 0 and 1 as well as sweeping λ in continuum to obtain the so called $\lambda\tau$ -space representation of images. In order to expose the properties of the generalized edge detection algorithm and the advantages of using such a filter for different purposes, we have considered two different set of images. The first set of images shown in Fig. 2.5 consists of noisy checkerboard and bar images. These synthetic images are useful to determine how well the detector can handle the tradeoff between the detection and localization performances and to evaluate the results quantitatively as well. The $\lambda\tau$ -space edge representations for checkerboard and bar images are shown in Fig. 2.6. To quantitatively evaluate the results, we calculated the conditional probability of a detected edge pixel given an ideal edge pixel, $Pr(DE/IE)$, the conditional probability of an ideal edge pixel given a detected edge pixel, $Pr(IE/DE)$, the Pratt's Figure of Merit (FoM), and the mean square distance (MSD) between ideal and detected edge pixels [12]. The mean square distance and the figure of merit are defined as

$$MSD = \frac{1}{n_d} \sum_{i=1}^{n_d} d_i^2 \quad (2.60)$$

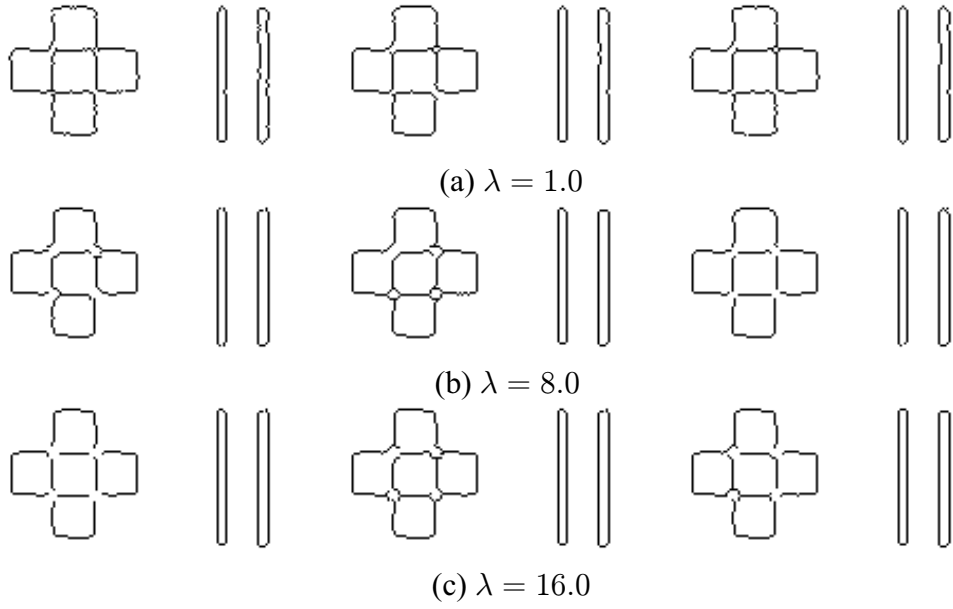


Figure 2.7: $\lambda\tau$ -space representation of edges (τ takes the values 0.0, 0.5, and 1.0 from left to right).

and

$$FOM = \frac{1}{\max\{n_d, n_t\}} \sum_{i=1}^{n_d} \frac{1}{1 + 0.9d_i^2}, \quad (2.61)$$

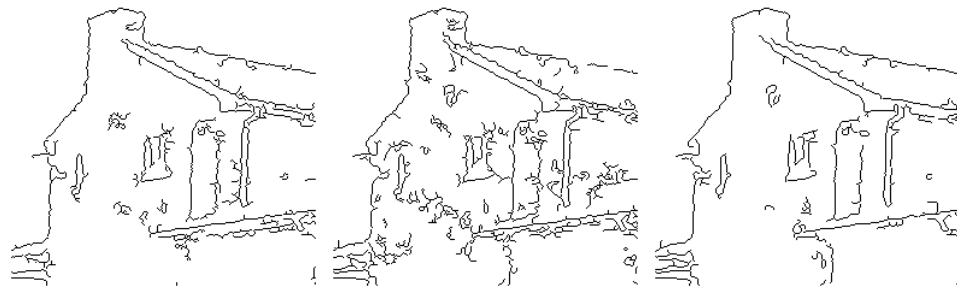
where n_d and n_t are the number of detected and ideal edge points, respectively, and d_i is the edge deviation for the i^{th} detected edge pixel. Edge detectors should maximize the probability measures $Pr(DE/IE)$ and $Pr(IE/DE)$ as well as FoM while minimizing the error distance. The quantitative results are shown in Table 2.8. As another test image, we have considered the noisy house image in Fig. 2.8 where the standard deviation of the additive Gaussian noise is 80. The $\lambda\tau$ -space edge representation obtained with the generalized edge detector is shown in Fig. 2.9. In this case, the first-order R -filter results in noisy edge fragments as shown in the left column. Note that even with a large value of the smoothing parameter λ , these noisy and disconnected edges can not be eliminated. For this image, the edges obtained with $\tau = 0.5$ are better. These two examples show that different problems require different edge detectors and the generalized edge detector is capable of producing these different results as samples of the scale-continuity ($\lambda\tau$) space representation.

Table 2.8: Quantitative evaluation of edges from the checkerboard image.

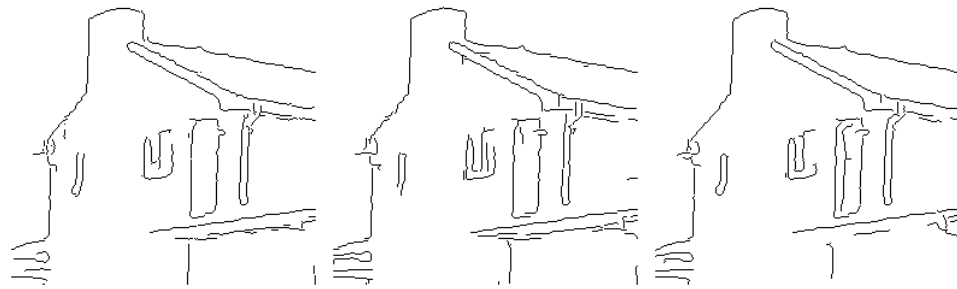
λ	$\tau =$	SNR=12dB			SNR=8.5dB			SNR=5dB		
		0	0.5	1	0	0.5	1	0	0.5	1
1.0	P(DE/IE)	0.58	0.51	0.52	0.56	0.48	0.47	0.52	0.46	0.46
	P(IE/DE)	0.59	0.50	0.52	0.57	0.49	0.48	0.58	0.52	0.48
	MSD	0.64	0.77	0.75	0.66	0.77	0.77	0.64	0.69	0.75
	FOM	0.60	0.52	0.53	0.58	0.51	0.50	0.53	0.48	0.48
8.0	P(DE/IE)	0.79	0.73	0.46	0.73	0.57	0.46	0.62	0.57	0.45
	P(IE/DE)	0.81	0.79	0.47	0.75	0.64	0.47	0.63	0.62	0.45
	MSD	0.45	0.49	0.82	0.53	0.72	0.86	0.70	0.82	0.88
	FOM	0.80	0.74	0.49	0.74	0.58	0.49	0.63	0.58	0.47
16.0	P(DE/IE)	0.79	0.45	0.44	0.79	0.47	0.44	0.73	0.44	0.42
	P(IE/DE)	0.90	0.50	0.48	0.88	0.50	0.46	0.83	0.45	0.45
	MSD	0.36	0.71	0.83	0.34	0.77	0.89	0.57	0.89	0.88
	FOM	0.79	0.47	0.46	0.79	0.49	0.46	0.74	0.47	0.44



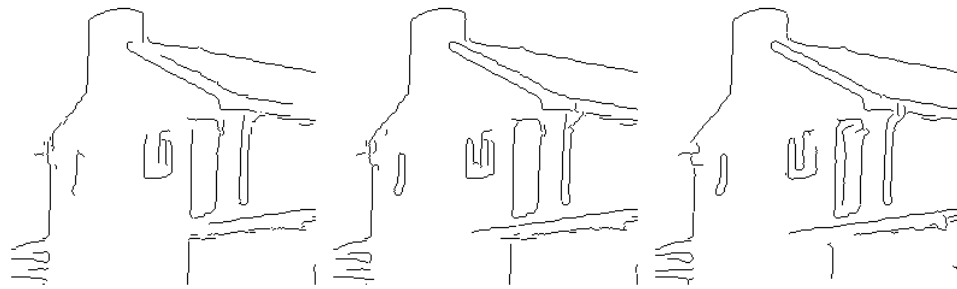
Figure 2.8: The original and noisy (SNR=2.4dB) house images



(a) $\lambda = 1.0$



(b) $\lambda = 8.0$



(c) $\lambda = 16.0$

Figure 2.9: $\lambda\tau$ -space Representation of edges for the noisy house image (τ takes the values 0.0, 0.5, and 1.0 from left to right).

2.5 Recursive Implementation

Serial implementation (Fig. 2.10) explained above has one major drawback: filtering with R -filter requires $O(L^2)$ multiplications where

$$L = \max\left(\frac{1}{e_1}, \frac{1}{e_2}\right) \times \ln\left(\frac{1}{\varepsilon}\right)$$

and

$$e_1 = \sqrt{\frac{b + \sqrt{\Delta}}{4a}}, e_2 = \sqrt{\frac{b - \sqrt{\Delta}}{4a}}.$$

Computational complexity increases as λ increases. One solution to keep the complexity constant is to use recursive implementation where two dimensional space is partitioned into four regions in such a way that the final system is causal and the filter is rewritten as the sum of continuous functions which are partially defined over these regions:

$$R(m, n) = \begin{cases} R_I(m, n) & ; m, n \geq 0 \\ R_{II}(m, n) & ; m < 0, n > 0 \\ R_{III}(m, n) & ; m, n \leq 0 \\ R_{IV}(m, n) & ; m > 0, n < 0 \end{cases} \quad (2.62)$$

The boundaries of these four regions are shown in Fig. 2.11. The Z transform is applied to $R(m, n)$ in each region such that

$$H_i(z_1^{-1}, z_2^{-1}) = Z\{R_i(m, n)\}; i = I, II, III, IV \quad (2.63)$$

and

$$H(z_1^{-1}, z_2^{-1}) = H_I(z_1^{-1}, z_2^{-1}) + H_{II}(z_1, z_2^{-1}) + H_{III}(z_1, z_2) + H_{IV}(z_1^{-1}, z_2). \quad (2.64)$$

$H(z_1^{-1}, z_2^{-1})$ is called transfer function. The transfer function has the following general form

$$H(z_1^p, z_2^q) = H_{00} \frac{\sum_{m=0}^2 \sum_{n=0}^2 c_{mn} z_1^{mp} z_2^{nq}}{\sum_{m=0}^2 \sum_{n=0}^2 d_{mn} z_1^{mp} z_2^{nq}} \quad (2.65)$$

where $p, q = \pm 1$. The coefficients in the equation (2.65) are calculated for each $R(m, n)$ filters of five cases and are summarized in Table.2.9. The architecture of recursive implementation is given in Fig. 2.12. The relationship between input (i.e. $x(m, n)$) and

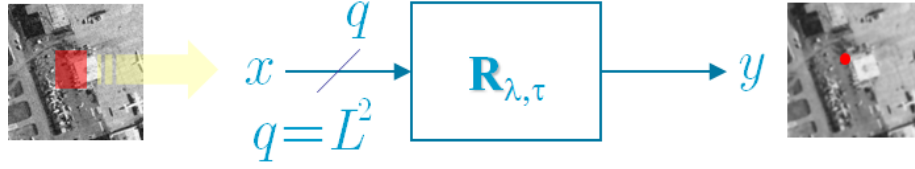


Figure 2.10: Architecture of serial implementation.

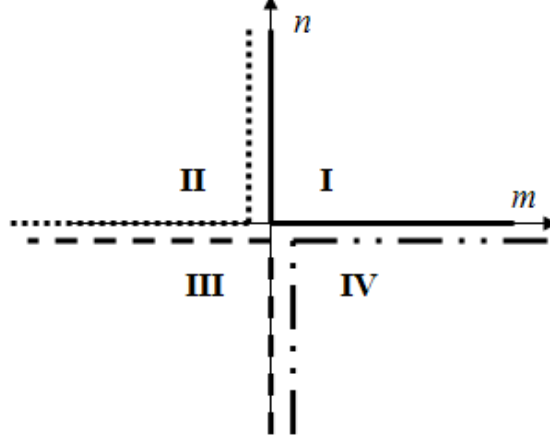


Figure 2.11: Four regions used to partition the space.

output (i.e. $y(m, n)$) is given by the following equation

$$\begin{aligned}
 y(m, n) = H_{00} & \left[\begin{aligned} & c_{00}x(m, n) + c_{10}x(m - p, n) + c_{01}x(m, n - q) \\ & + c_{11}x(m - p, n - q) + c_{20}x(m - 2p, n) + c_{02}x(m, n - 2q) \\ & + c_{21}x(m - 2p, n - q) + c_{12}x(m - p, n - 2q) \\ & + c_{22}x(m - 2p, n - 2q) \end{aligned} \right] \\
 & - d_{10}y(m - p, n) - d_{01}y(m, n - q) - d_{11}y(m - p, n - q) \\
 & - d_{20}y(m - 2p, n) - d_{02}y(m, n - 2q) - \\
 & - d_{21}y(m - 2p, n - q) - d_{12}y(m - p, n - 2q) - d_{22}y(m - 2p, n - 2q)
 \end{aligned} \tag{2.66}$$

Since the transfer function has four terms (H_I, H_{II}, H_{III} , and H_{IV}), the output $y(m, n)$ is a sum of four respective outputs (y_I, y_{II}, y_{III} , and y_{IV}) such that

$$y(m, n) = y_I(m, n) + y_{II}(m, n) + y_{III}(m, n) + y_{IV}(m, n). \tag{2.67}$$

Scan order is important in applying the transfer functions given in Table 2.9 the output at each region. The scan orders are collectively given in Fig. 2.13. Please note that the recursive architecture (Fig. 2.12) requires constant number of multiplications to compute $y(m, n)$ when $x(m, n)$ is given. Whatever scale parameters (λ and τ) are chosen, recursive implementation needs 52 multiplications for the cases $\Delta > 0$, $\Delta < 0$, and $\Delta = 0$, and 20 multiplications for the case $A = 0$.

Table 2.9: Transfer function $H(z_1, z_2)$ coefficients.

Case	$R(m, n)$	
$\Delta > 0$	$d_{00} = 1, d_{10} = d_{01} = -(e^{-a} + e^{-b})$ $d_{11} = (e^{-a} + e^{-b})^2, d_{20} = d_{02} = e^{-(a+b)}$ $d_{21} = d_{12} = -e^{-(a+b)}(e^{-a} + e^{-b}), d_{22} = e^{-2(a+b)}$	
	I	$c_{00} = (b - a), c_{10} = c_{01} = be^{-b} - ae^{-a}$ $c_{11} = ae^{-2a} - be^{-2b}$
	II	$c_{10} = (ae^{-b} - be^{-a}), c_{11} = c_{20} = e^{-(a+b)}(b - a)$ $c_{21} = (ae^{-a} + be^{-b})e^{-(a+b)}$
	III	$c_{01} = (ae^{-b} - be^{-a}), c_{11} = c_{02} = e^{-(a+b)}(b - a)$ $c_{12} = (ae^{-a} + be^{-b})e^{-(a+b)}$
	IV	$c_{11} = (ae^{-2b} - be^{-2a}), c_{21} = c_{12} = (be^{-2a} - ae^{-2b})$ $c_{22} = (a - b)e^{-2(a+b)}$
$\Delta = 0$	$d_{00} = 1, d_{10} = d_{01} = -2e^{-\frac{1}{\sqrt{B}}}, d_{11} = 4e^{-\frac{2}{\sqrt{B}}}$ $d_{20} = d_{02} = e^{-\frac{2}{\sqrt{B}}}, d_{21} = d_{12} = -2e^{-\frac{3}{\sqrt{B}}}, d_{22} = e^{-\frac{4}{\sqrt{B}}}$	
	I	$c_{00} = 1, c_{10} = c_{01} = \left(\frac{1}{\sqrt{B}} - 1\right)e^{-\frac{1}{\sqrt{B}}}, c_{11} = e^{-\frac{2}{\sqrt{B}}}\left(\frac{1}{\sqrt{B}} - 1\right)^2$
	II	$c_{10} = \left(\frac{1}{\sqrt{B}} + 1\right)e^{-\frac{1}{\sqrt{B}}}, c_{11} = e^{-\frac{2}{\sqrt{B}}}\left(\frac{1}{B} - 1\right)$ $c_{20} = -e^{-\frac{2}{\sqrt{B}}}, c_{21} = -e^{-\frac{3}{\sqrt{B}}}\left(\frac{1}{\sqrt{B}} - 1\right)$
	III	$c_{01} = \left(\frac{1}{\sqrt{B}} + 1\right)e^{-\frac{1}{\sqrt{B}}}, c_{11} = e^{-\frac{2}{\sqrt{B}}}\left(\frac{1}{B} - 1\right)$ $c_{02} = -e^{-\frac{2}{\sqrt{B}}}, c_{12} = -e^{-\frac{3}{\sqrt{B}}}\left(\frac{1}{\sqrt{B}} - 1\right)$
	IV	$c_{11} = \left(\frac{1}{\sqrt{B}} + 1\right)^2 e^{-\frac{2}{\sqrt{B}}}, c_{21} = c_{12} = -\left(\frac{1}{\sqrt{B}} + 1\right)e^{-\frac{3}{\sqrt{B}}}$ $c_{22} = e^{-\frac{4}{\sqrt{B}}}$
$\Delta < 0$	$d_{00} = 1, d_{10} = d_{01} = -2\cos(w), d_{11} = 4\cos^2(w)$ $d_{20} = d_{02} = e^{-2\alpha}, d_{21} = d_{12} = -2e^{-2\alpha}\cos(w), d_{22} = e^{-4\alpha}$	
	I	$c_{00} = \cos(\theta), c_{10} = c_{01} = e^{-\alpha}\cos(w - \theta)$ $c_{11} = e^{-2\alpha}\cos(2w - \theta)$
	II	$c_{00} = \cos(\theta), c_{10} = -e^{-\alpha}\cos(w - \theta)$ $c_{01} = -e^{-\alpha}\cos(w + \theta), c_{11} = e^{-2\alpha}\cos(\theta)$
	III	$c_{00} = \cos(\theta), c_{10} = -e^{-\alpha}\cos(w - \theta)$ $c_{01} = -e^{-\alpha}\cos(w + \theta), c_{11} = e^{-2\alpha}\cos(\theta)$
	IV	$c_{00} = \cos(\theta), c_{10} = c_{01} = e^{-\alpha}\cos(w - \theta)$ $c_{11} = e^{-2\alpha}\cos(2w - \theta)$
$A = 0$	$d_{00} = 1, d_{10} = d_{01} = -e^{-\frac{1}{\sqrt{2B}}}, d_{11} = e^{-\frac{2}{\sqrt{2B}}}$	
	I	$c_{00} = 1$
	II	$c_{10} = e^{-\frac{1}{\sqrt{2B}}}$
	III	$c_{01} = e^{-\frac{1}{\sqrt{2B}}}$
	IV	$c_{11} = e^{-\frac{1}{\sqrt{2B}}}$

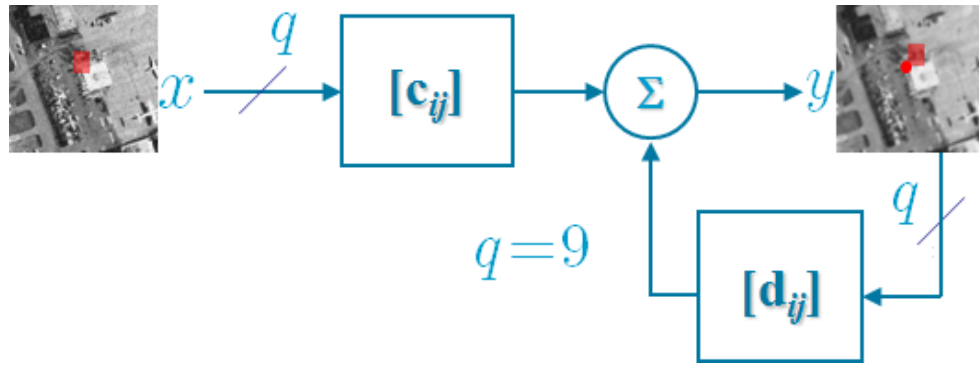


Figure 2.12: Architecture of recursive implementation.

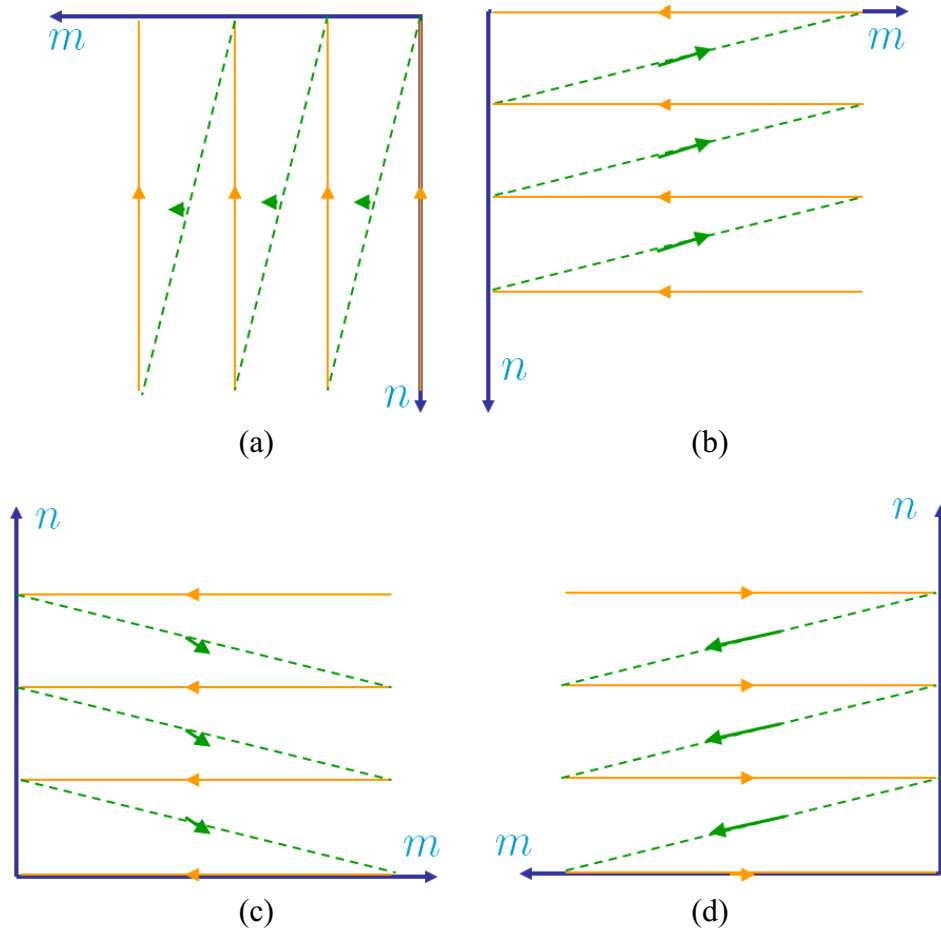


Figure 2.13: Scan order of transfer functions in recursive implementation for Region I (a), Region II (b), Region III (c), and Region IV (d).

2.6 Isotropic Diffusion and Recursive Implementation

In this study, we also interpreted the recursive implementation and revealed its relation with the forth-order isotropic diffusion equation. Forth-order diffusion equation is the minimizer of the following energy functional

$$E(f) = \int \int_{\Omega} \lambda(1 - \tau) \|\nabla f\|^2 + \lambda\tau \|Hf\|^2 d\Omega. \quad (2.68)$$

The corresponding partial differential equation is easily obtained by the Euler-Lagrange equation follows

$$\frac{\partial f}{\partial t} = (1 - \tau) \lambda \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right) + \tau \lambda \left(\frac{\partial^4 f}{\partial x^4} + 2 \frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4} \right) \quad (2.69)$$

If the above partial differential equation is discretized using the forward and backward differences by replacing the Laplacian and the Hessian operators with their discrete equivalents

$$\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right) \approx f_{m+1,n}^{k+1} + f_{m,n+1}^{k+1} + f_{m-1,n}^{k+1} + f_{m,n-1}^{k+1} - 4f_{m,n}^{k+1} \quad (2.70)$$

$$\begin{aligned} \left(\frac{\partial^4 f}{\partial x^4} + 2 \frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4} \right) \approx & f_{m+2,n}^{k+1} + f_{m,n+2}^{k+1} + f_{m-2,n}^{k+1} + f_{m,n-2}^{k+1} - \\ & 8(f_{m+1,n}^{k+1} + f_{m,n+1}^{k+1} + f_{m-1,n}^{k+1} + f_{m,n-1}^{k+1}) + \\ & 2(f_{m+1,n+1}^{k+1} + f_{m-1,n+1}^{k+1} + f_{m+1,n-1}^{k+1} + f_{m-1,n-1}^{k+1}) + 20f_{m,n}^{k+1} \end{aligned} \quad (2.71)$$

then the partial differential equation becomes

$$f_{m,n}^{k+1} - f_{m,n}^k = (1 - \tau) \lambda_r (f_{m+1,n}^{k+1} + f_{m,n+1}^{k+1} + f_{m-1,n}^{k+1} + f_{m,n-1}^{k+1} - 4f_{m,n}^{k+1}) + \tau \lambda_s \left(\begin{aligned} & f_{m+2,n}^{k+1} + f_{m,n+2}^{k+1} + f_{m-2,n}^{k+1} + f_{m,n-2}^{k+1} - \\ & 8(f_{m+1,n}^{k+1} + f_{m,n+1}^{k+1} + f_{m-1,n}^{k+1} + f_{m,n-1}^{k+1}) + \\ & 2(f_{m+1,n+1}^{k+1} + f_{m-1,n+1}^{k+1} + f_{m+1,n-1}^{k+1} + f_{m-1,n-1}^{k+1}) + 20f_{m,n}^{k+1} \end{aligned} \right). \quad (2.72)$$

The transfer function of the system whose input-output relation is governed by the equation (2.72) is given by

$$\hat{H}(z_1, z_2; \lambda, \tau) = \frac{1}{\left((1 + 4(1 - \tau)\lambda - 20\tau\lambda) - \right.} \quad (2.73)$$

$$\left. \left(((1 - \tau)\lambda + 8\tau\lambda)(z_1 + z_2 + z_1^{-1} + z_2^{-1}) - \right. \right.$$

$$\left. \left. \tau\lambda \left(z_1^2 + z_1^{-2} + z_2^2 + z_2^{-2} + \right. \right. \right.$$

$$\left. \left. \left. 2(z_1 z_2 + z_1^{-1} z_2 + z_1 z_2^{-1} + z_1^{-1} z_2^{-1}) \right) \right) \right)$$

The transfer function given by the equation (2.65) is a special case of the transfer function given by the equation (2.72). So recursive implementation is equivalent to the forth-order isotropic diffusion.

2.7 Evaluation Based on Empirical ROC Curves

There are several parameters to tune when dealing with edge detectors. Usually there is no easy way to find the optimal edge detector parameters for an image. In fact, one set of optimal parameters may not be optimal for another image. In this section we will find optimal edge detection parameters using receiver operator characteristics (ROC) for an image when its ideal edges are available using exhaustive search. For smoothing filter we have 2-dimensional search space where the dimensions are λ and τ . For Generalized Edge Detector (GED) we have 4-dimensional search space where the dimensions are λ , τ , T_{low} , and T_{high} . The search space is given in Fig. 2.14.

First the optimum scale parameters (λ^*, τ^*) in smoothing with $R(x, y)$ -filters are searched where the noisy and the original images are available. Checkerboard and bar images are used in searching. Image pairs and the optimum scale parameters are listed in Table 2.10. Search trajectories for the images given in Table 2.10 are plotted in Fig. 2.15 and Fig. 2.16. MSE surfaces for the same images are given in Fig. 2.17 and Fig. 2.18. These experiments verify the previous conclusion that the hybrid solution obtained with τ around 0.5 does not suffer from over smoothed discontinuities of the membrane solution or the overshoots of the plate solution. As compared to these two standard models, the hybrid model provides a better replica of the original surface.

Then the optimum edge detection parameters $(\lambda^*, \tau^*, T_{low}^*, T_{high}^*)$ in edge detection with $G(x, y)$ -filters are searched where the ideal edges are available. Florida database ([43]) is used in the experiments. There are 50 natural images and 10 aerial images in the database. The ground-truth edges are obtained by averaging edges marked by users. Pixels are

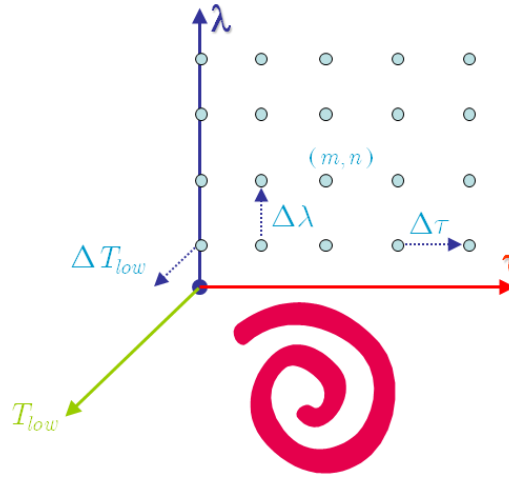


Figure 2.14: Search space for the Generalized Edge Detector.

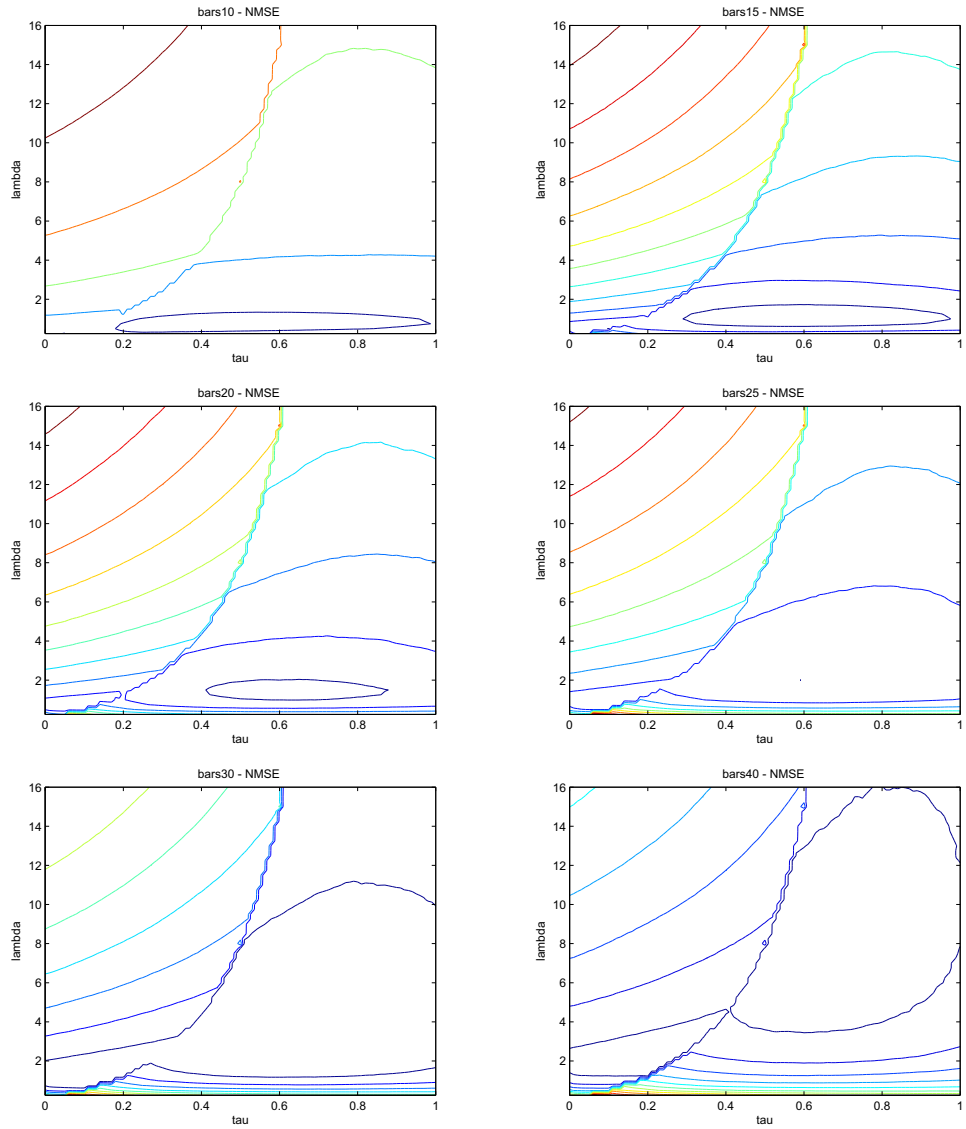


Figure 2.15: Search trajectories for the bars images with various SNR values.

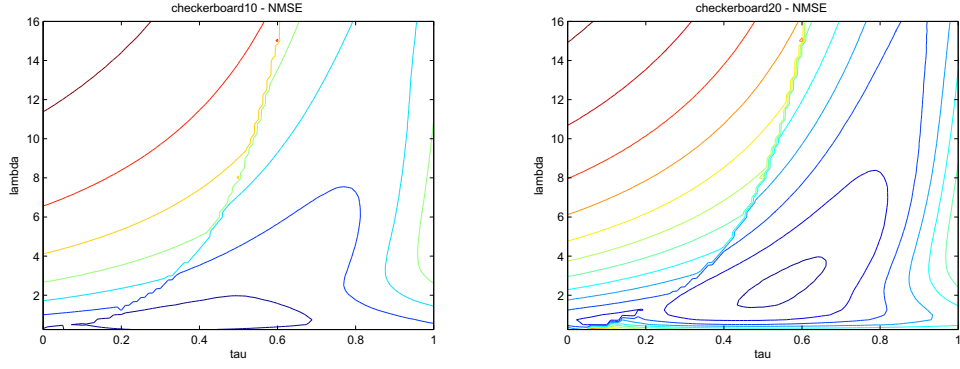


Figure 2.16: Search trajectories for the checkerboard images.

Table 2.10: Optimum scale parameters with respect to mean square error (mse) between the smoothed image and the original image.

Image	dB	λ^*	τ^*	mse
Checkerboard	10	0.75	0.38	237028
	20	2.75	0.56	393060
Bars	10	0.75	0.55	90234
	15	1.0	0.59	111856
	20	1.5	0.62	135814
	25	2.0	0.59	159999
	30	2.75	0.65	181256
	40	2.25	0.28	210896

classified into three classes as edges represented by 0, non-edges represented by 1, and don't care pixels represented by 255. Images and their ground truth edges in the data set are given in Fig. 2.19. An edge detector can report an edge where none exists; this can be due to noise, or simply poor design or thresholding, and is called a *false positive*. In addition, an edge detector could fail to report an edge pixel that does exist. This is called a *false negative* or *unmatched ground truth*. Correct edges are called *true positives*. Fig. 2.20 presents edges detected with the optimum edge detector parameters ($\lambda^*, \tau^*, T_{low}^*, T_{high}^*$) for the images in Florida data set with respect to maximum true positives. The optimum edge detector parameters ($\lambda^*, \tau^*, T_{low}^*, T_{high}^*$) are given in Table 2.11. True positive values (**tp**) are around 0.99 for almost all images in the data set. It shows that G -filters (GED) can detect edges successfully when the optimum parameters are available. But this is not the case in practical applications. Finally, we provide ROC curves used to find the optimum detector parameters. The curves are given in Fig. 2.21. We may conclude that each image has different ROC's and finding optimum detector parameters for a given image almost impossible.

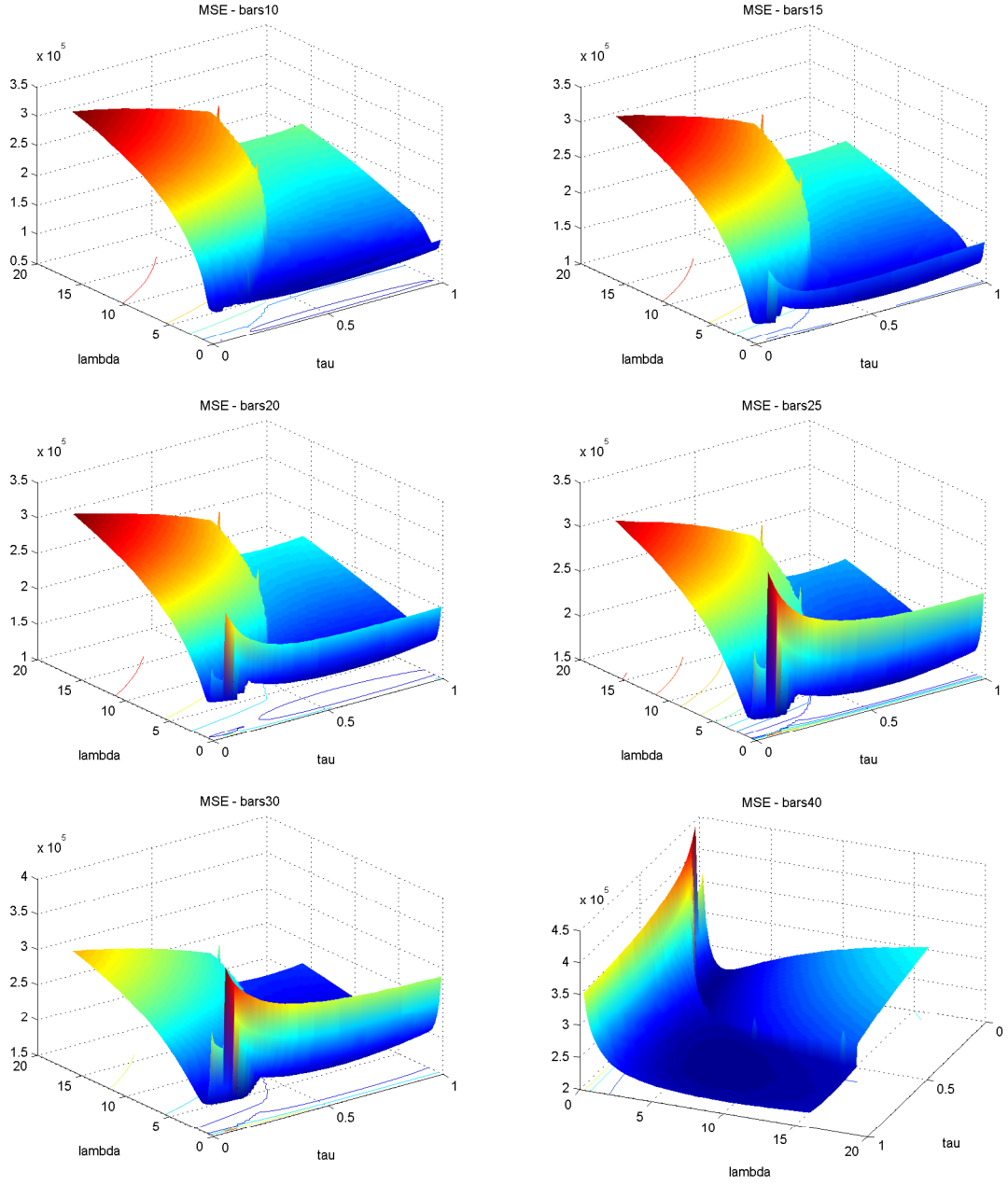


Figure 2.17: *nmse* surfaces for the bars images with various SNR values.

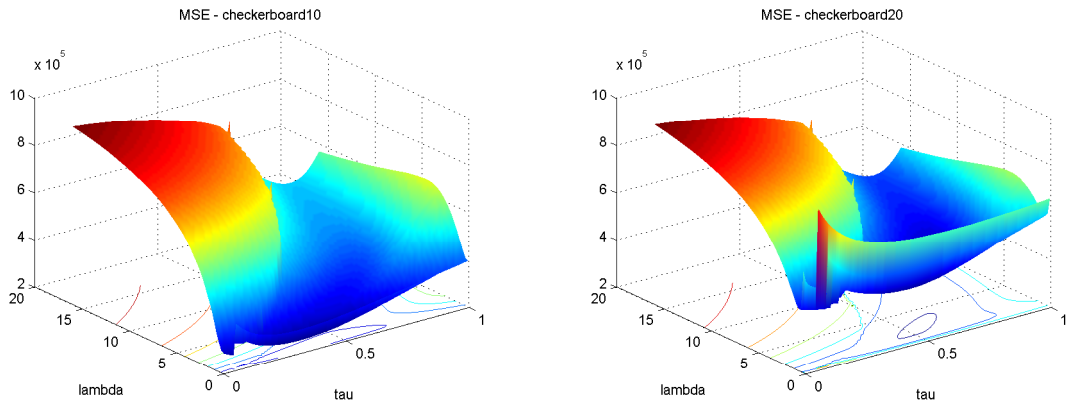


Figure 2.18: *nmse* surfaces for the checkerboard images.

Table 2.11: Optimum detector parameters $(\lambda^*, \tau^*, T_{low}^*, T_{high}^*)$ with respect to true positives (**tp**) computed between the detected edges and the ground truth edges. **fp** denotes *false positives* and **ugt** denotes *unmatched ground truth*.

Image	tp	fp	ugt	λ^*	τ^*	T_{low}^*	T_{high}^*
101	0.97	0.02	0.09	0.2	0.1	0.02	0.1
103	0.97	0.02	0.09	0.1	0.9	0.01	0.1
104	0.99	0.007	0.17	0.3	0.5	0.01	0.2
105	0.99	0.007	0.02	0.1	0.1	0.02	0.15
106	0.98	0.013	0.02	0.1	0.1	0.02	0.1
108	0.99	0.003	0.02	0.1	0.1	0.05	0.15
109	0.98	0.014	0.10	0.1	0.9	0.03	0.1
110	0.99	0.004	0.04	0.1	0.9	0.01	0.2
111	0.99	0.009	0.04	0.3	0.0	0.05	0.2
125	0.98	0.010	0.013	0.1	0.4	0.03	0.1
126	0.98	0.012	0.0	0.1	0.3	0.04	0.1
130	0.99	0.006	0.01	0.1	0.9	0.01	0.1
131	0.98	0.020	0.03	0.2	0.8	0.01	0.1
132	0.98	0.016	0.00	0.2	1.0	0.01	0.1
133	0.98	0.021	0.30	0.4	0.1	0.03	0.2
134	0.99	0.011	0.00	0.1	0.3	0.03	0.15
137	0.99	0.008	0.00	0.1	0.3	0.02	0.1
138	0.99	0.013	0.01	0.3	0.3	0.03	0.15
143	0.98	0.015	0.03	0.1	0.4	0.03	0.1
144	0.98	0.023	0.03	0.1	0.2	0.02	0.1
146	0.98	0.017	0.01	0.2	0.9	0.02	0.1
202	0.99	0.004	0.01	0.3	0.6	0.04	0.2
203	0.99	0.002	0.00	0.1	0.3	0.01	0.15
204	0.99	0.003	0.01	0.2	0.4	0.01	0.1
207	0.99	0.014	0.00	0.1	0.2	0.05	0.1
214	0.98	0.019	0.05	0.1	0.9	0.02	0.1
215	0.99	0.011	0.01	0.1	0.2	0.03	0.1
217	0.99	0.010	0.16	0.2	0.3	0.03	0.2

2.8 Conclusion

We have presented two dimensional edge detector by using the hybrid model derived from the membrane and plate models. Compared to the scale space representation, this representation can decompose the image/surface into a larger number of descriptions, each of which exhibits different characteristics of the image or surface.

After finding the relationship between the minimization of the hybrid functional and convolution of an image with a filter, we can regard this representation as an extended scale space representation where the shape characteristic of the filter as well as its scale change. The hybrid model and the edge detection have been examined on various types of data obtained from different cues. We have experimentally shown that the hybrid model



Figure 2.19: Florida data set and its ground truth edges [43].

provides additional information compared to the two extreme constituent models, *i.e.*, the membrane and plate models. We believe that the proposed representation will be applicable to a wide variety of problems.

We have derived a general-purpose edge detector from the hybrid model, which encompasses most of the well-known edge detectors under a unified framework. The edge representation generated by this operator contains very different edge descriptions, so the desired edge description required for a given problem can be found in this representation. We believe that this generalization and flexibility is very suitable for developing successful vision systems and that the generalized edge detector will find many applications.

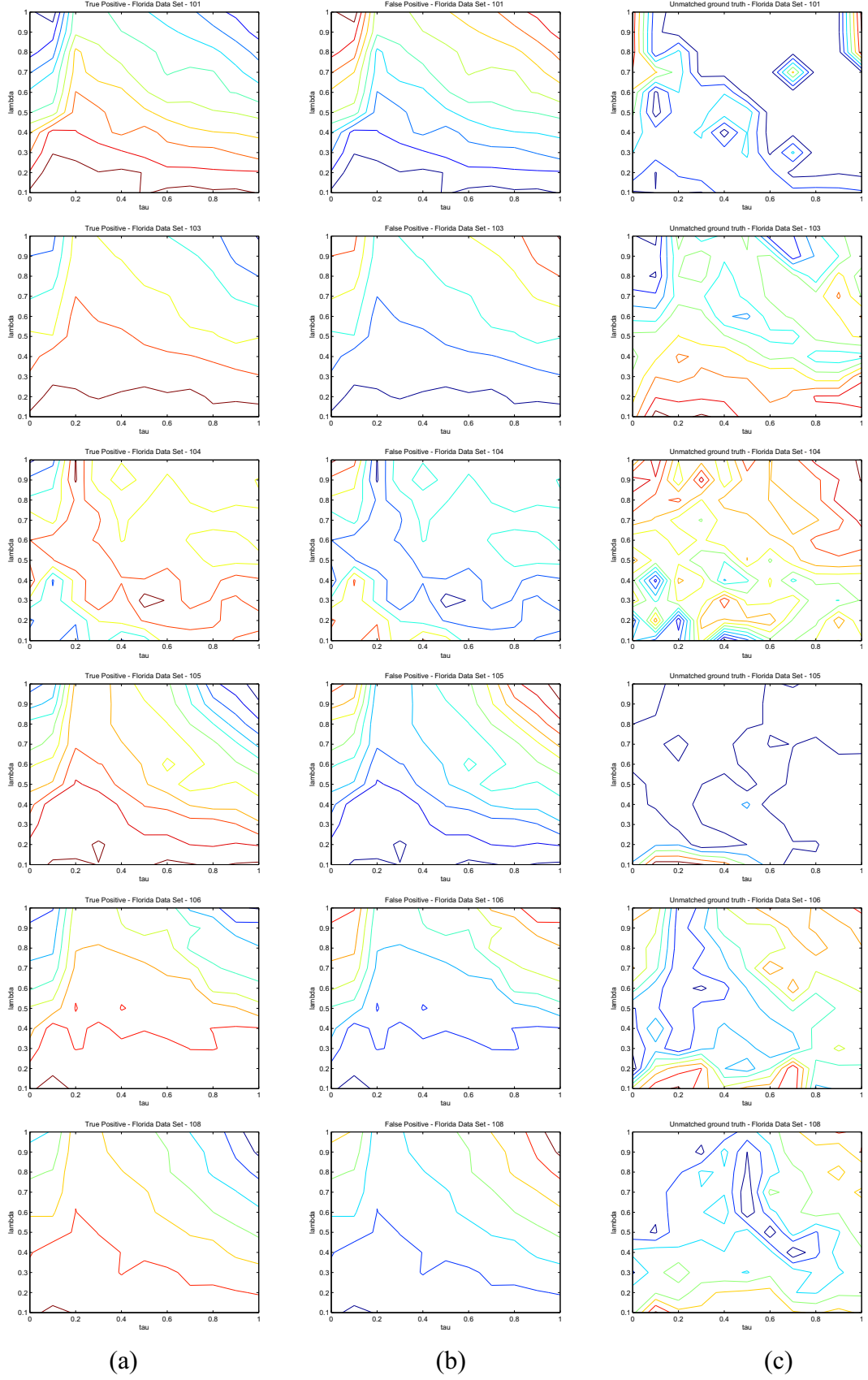


Figure 2.21: (a) *True positives (tp)*, (b) *False positives (fp)* and (c) *Unmatched ground truth (ugt)* plots obtained for various images in Florida data set. In the plots, the x -axis is $\lambda \in [0.1..1.0]$ and the y -axis is $\tau \in [0.0..1.0]$.

3. REGULARIZED NON-LINEAR SOLUTION: ANISOTROPIC $\lambda\tau$ -SPACE

In this section, we propose an adaptive scheme for the representation of images and edges in $\lambda\tau$ -space explained in the previous chapter. The space is comprised of two dimensions: the scale dimension and the continuity dimension, controlling the smoothness and the continuity of the surface, respectively. The representation is obtained simply by filtering the image with the filter denoted as $R(x, y; \lambda, \tau)$. We have shown that the representation is richer than the classical scale-space representation. In our proposed scheme, we let the space parameters, λ and τ , to vary through the image respecting the feature directions. Also we apply iterative smoothing scheme in which an image is smoothed at each iteration by controlling the space parameters also in time. The first aspect of the algorithm is connected with the robustness to noise. The second aspect of the algorithm concerns the way it treats the direction of the edges. The relation between the proposed algorithm and the anisotropic diffusion is also established in this study.

3.1 Introduction

Our purpose in this study is to introduce an adaptive scheme for $\lambda\tau$ -space representation using regularization and partial differential equation framework. Nonlinear partial differential equations have attracted great interest in solving many low-level image processing and computer vision problems including image restoration [44, 45], edge detection [46, 47], stereo [48] and curve evaluation [49, 50]. This is due to their ability of modeling the image in continuous domain and high accuracy and stability properties when discretized.

Partial differential equations mostly arise from regularization stemming from the formulation of image processing problems as energy functionals. Regularization is a general framework used to convert an ill-posed problem to well-posed by restricting the class of admissible solutions using constraints such as smoothness. Smoothness could be imposed in the form of derivatives of the solution. Gökmen and Jain [1] presented a method based on regularization with filters in which regularization term consists of first and second order partial derivatives of the solution. First order derivatives correspond to membrane model and second order derivatives correspond to thin plate model. The smoothing

filter, $R(x, y)$, associated to the functional is obtained by solving Euler-Lagrange equation resulting in forth-order partial differential equation. These filters and their properties are investigated in [4]. $\lambda\tau$ -space representation of images and edges are formed by convolving the image with the filters denoted as $R(x, y)$.

One major drawback of the algorithm is that space parameters are constant throughout the image space causing the edges to be blurred while noise is removed elsewhere. This problem is due to the linear nature of the representation. To overcome this problem, we present an anisotropic $\lambda\tau$ -space image and edge representation allowing the space parameters, λ and τ , to vary through the image respecting the feature directions.

3.2 Overview

Image representation aims at producing images with different details. This is one of the most challenging issue in computer vision and image processing due to lack of a robust representation. In general, representations can be classified as linear and nonlinear. Linear representations are constructed by convolving the image by linear filters (kernels) with varying scales. For instance, in classical scale-space the kernel is a Gaussian and the scale-space representation is obtained either by convolving the image by a Gaussian with increasing standard deviation or equivalently by solving the linear heat equation in time. This representation is causal, since the isotropic heat equation satisfies a maximum principle. However, the Gaussian scale-space suffers from serious drawbacks such as over-smoothing and location uncertainty along edges at large scales due to interactions between nearby edges and displacements. Smoothing is done in a homogeneous fashion at the same rate in all directions. This results in blurring of edges at the same rate as smoothing of object interiors. Such a filter results in poor edge localization. A representation, which is known to be richer than the Gaussian scale-space, is $\lambda\tau$ -space representation in which the kernel is derived from the hybrid functional

$$E(u) = \int \int_{\Omega} (u - d)^2 + \lambda[(1 - \tau)(u_x^2 + u_y^2) + \tau(u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2)]d\Omega \quad (3.1)$$

where $d(x, y)$ is the given image, λ is the real-valued regularization parameter and $\tau \in [0, 1]$ is the real-valued continuity control parameter. Here, the hybrid functional is used to obtain a smooth transition from membrane model to the plate model. Note that, for $\tau = 0$, the functional reduces to the membrane model, for $\tau = 1$ it reduces to the plate model, and for the intermediate values of this parameter we obtain hybrid surfaces. In order to obtain these hybrid surfaces, the solution of the Euler-Lagrange equation associated the

hybrid functional is needed. The derivation of these solutions and their analysis are given in Chapter 2.

The $\lambda\tau$ -space representation is capable of producing images each of which have different details. However, the use of constant scale parameters throughout the image causes the edges to get blurred for large values regardless of the dimension. Next section deals with this problem, introducing anisotropic $\lambda\tau$ -space representation.

3.3 Adaptive Scheme

Recall that the hybrid functional is comprised of two terms containing first- and second-order derivatives of the solution along x and y directions. We denote these terms as

$$E_1(u) = \int \int_{\Omega} \|\nabla u\|^2 d\Omega \quad (3.2)$$

$$E_2(u) = \int \int_{\Omega} \|Hu\|^2 d\Omega \quad (3.3)$$

where H designates the *Hessian* operator. The hybrid functional is defined as a linear combination of these two terms

$$E(u) = (1 - \tau)E_1(u) + \tau E_2(u) \quad (3.4)$$

Minimization of the functionals (3.2) and (3.3) separately yields the following partial differential equations

$$\frac{\partial u}{\partial t} = \nabla(\lambda(\|\nabla(u)\|))\nabla^T(u) \quad (3.5)$$

$$\frac{\partial u}{\partial t} = \nabla\nabla^T(\lambda(\|\nabla(u)\|)\nabla\nabla^T(u)) \quad (3.6)$$

where ∇ is the gradient operator defined as follows

$$\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}] \quad (3.7)$$

such that

$$\nabla(g) = [\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}] \hat{=} \nabla g, (\nabla(g))^T = \nabla^T(g) \quad (3.8)$$

$$\nabla\left(\begin{bmatrix} f \\ g \end{bmatrix}\right) = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} \quad (3.9)$$

$$\Delta = \nabla \nabla^T \quad (3.10)$$

$$H = \nabla^T \nabla \quad (3.11)$$

and $u(x, y; 0) = d(x, y)$.

The quasi-linear partial differential equation given in the equation (3.5) is the anisotropic (inhomogeneous) diffusion proposed by Perona and Malik [5]. λ is called the diffusion coefficient which is a function of the gradient of the solution, $f(x, y; t)$, and is chosen to be a decreasing function of the gradient, $\|\nabla f\|$, in such a way that a higher rate of diffusion occurs for low values of $\|\nabla f\|$ and vice-versa. This encourages smoothing within a region in preference to smoothing across edges, thereby also performing image enhancement.

The minimization of the hybrid functional (3.4) can be easily derived from the partial differential equations (3.5) and (3.6) as

$$\begin{aligned} \frac{\partial u}{\partial t} = (1 - \tau) \nabla(\lambda_m(\|\nabla u\|) \nabla^T(u)) + \\ \tau \nabla \nabla^T(\lambda_p(\|\Lambda u\|) \nabla \nabla^T(u)) \end{aligned} \quad (3.12)$$

Equation (3.12) consists of the linear combination two diffusion processes whose fluxes are controlled by the λ_m and λ_p which are functions of gradient. When the image is corrupted by noise of high variance, higher gradient terms are required to robust estimation of the local image features such as edges and corners. In order to control the direction of diffusion allowing minimal smoothing in the direction across to the image features, and maximal smoothing in the direction normal to the image features, we express the hybrid diffusion in $\eta\xi$ -form such that new directional operators, ∇_η and ∇_ξ , are defined as

$$\nabla_\eta \hat{=} \eta \nabla$$

$$\nabla_\xi \hat{=} \xi \nabla$$

where

$$\eta(u) = \frac{[u_x, u_y]}{\sqrt{u_x^2 + u_y^2}}, \xi(u) = \frac{[-u_y, u_x]}{\sqrt{u_x^2 + u_y^2}} \quad (3.13)$$

Also note that

$$\nabla(u) = \nabla_\eta(u)\eta + \nabla_\xi(u)\xi \quad (3.14)$$

$$\nabla(\nabla^T(u)) = \nabla_\eta(\nabla_\eta(u)) + \nabla_\xi(\nabla_\xi(u)) \quad (3.15)$$

Rewriting the hybrid diffusion given in the equation (3.12) in terms of the directional operators yields

$$\begin{aligned} \frac{\partial u}{\partial t} = (1 - \tau) \{ & [\lambda_m + \|\nabla u\| \lambda'_m] \nabla_\eta(\nabla_\eta^T(u)) + \\ & \lambda'_m \Lambda_\xi(\Lambda_\xi^T(u)) \} + \\ & \tau \{ \nabla \nabla^T(\lambda_p(\|\Lambda u\|)) \nabla \nabla^T(u) + \\ & \lambda_p(\|\nabla u\|) \nabla \nabla^T(\nabla \nabla^T(u)) \} \end{aligned} \quad (3.16)$$

or in a simpler form

$$\begin{aligned} \frac{\partial u}{\partial t} = (1 - \tau) \{ & [\lambda_m + \|\Lambda u\| \lambda'_m] u_{\eta\eta} + \lambda_m u_{\xi\xi} \} + \\ & \tau \{ [\lambda_p + \|\Lambda u\|^2 \Delta \lambda_p] u_{\eta\eta\eta\eta} + \lambda_p u_{\xi\xi\xi\xi} \} \end{aligned} \quad (3.17)$$

where

$$\begin{aligned} u_{\eta\eta} &= \frac{u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}}{(u_x^2 + u_y^2)} \\ u_{\xi\xi} &= \frac{u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}}{(u_x^2 + u_y^2)} \end{aligned} \quad (3.18)$$

$$\begin{aligned} u_{\eta\eta\eta\eta} &= \frac{u_x^2 u_{xxxx} + 2u_x u_y u_{xxxy} + (u_x^2 + u_y^2) u_{xxyy} + 2u_x u_y u_{xyyy} + u_y^2 u_{yyyy}}{(u_x^2 + u_y^2)} \\ u_{\xi\xi\xi\xi} &= \frac{u_y^2 u_{xxxx} - 2u_x u_y u_{xxxy} + (u_x^2 + u_y^2) u_{xxyy} - 2u_x u_y u_{xyyy} + u_x^2 u_{yyyy}}{(u_x^2 + u_y^2)} \end{aligned} \quad (3.19)$$

The functions λ_m and λ_p control the amount and the direction of smoothness such that for large values of the gradient ($\|\nabla f\|$) we allow less diffusion and vice-versa, and in the

gradient direction (i.e. η) we have always smaller amount of diffusion due to the additive term inversely proportional to the gradient. Note that for $\tau = 0$, the diffusion reduces to the Perona-Malik equation, for $\tau = 1$ it reduces to the second-order diffusion, and for intermediate values of this parameter we obtain the surfaces corresponding to the hybrid diffusion.

Next section first deals with the discretization issues of the hybrid diffusion, and then presents computational results.

3.4 Implementation and Results

In this section, we briefly mention on the discretization of the partial differential equation given in (3.17) by using finite differences. At each instance of time, the function $u(x, y; t + 1)$ is evaluated through the forth-order procedure:

$$\frac{\partial u(x, y; t)}{\partial t} = V^{(t)}(u(x, y; t))(k) - V^{(t)}(u(x, y; t))(k - 1) \quad (3.20)$$

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= \frac{\partial}{\partial x}(V(u)(i, j) - V(u)(i - 1, j)) \\ &= V\left(\frac{\partial u}{\partial x}\right)(i, j) - V\left(\frac{\partial u}{\partial x}\right)(i - 1, j) \end{aligned} \quad (3.21)$$

$$\begin{aligned} \frac{\partial^4 u}{\partial x^4} &= \frac{\partial^2}{\partial x^2}(V(u)(i + 1, j) + V(u)(i - 1, j) - 2V(u)(i, j)) \\ &= V\left(\frac{\partial^2 u}{\partial x^2}\right)(i + 1, j) + V\left(\frac{\partial^2 u}{\partial x^2}\right)(i - 1, j) - 2V\left(\frac{\partial^2 u}{\partial x^2}\right)(i, j) \end{aligned} \quad (3.22)$$

where $T(\cdot)(k)$ denotes the value of the function of t at k and $V(\cdot)(i, j)$ denotes the value of the function of (x, y) at (i, j) . The functions λ_m and λ_p are chosen as

$$\lambda_m(\|\nabla(u)\|; t) = \frac{1}{1 + \left(\frac{\|\nabla(u)\|}{K(t)}\right)^2} \quad (3.23)$$

or

$$\lambda_p(\|\nabla(u)\|; t) = \exp\left(-\left(\frac{\|\nabla(u)\|}{K(t)}\right)^2\right) \quad (3.24)$$

where the function $K(t)$ is a decreasing function of time. The effect of $K(t)$ on the solution is studied in [51]. We have used a linear function, $K(t) = K_{max} - \alpha t$, in our experiments where $\alpha = (K_{max} - K_{min})/T$ and T is called the diffusion time.

Table 3.1: Quantitative results for noisy checkerboard images.

MSE	41.06			30.47			25.43		
τ	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0
$Pr(IE DE)$	0.94	0.97	0.95	0.974	0.97	0.99	0.99	0.99	0.99
$Pr(IE DE)$	0.95	0.98	0.97	0.985	0.99	0.99	0.99	0.99	0.99
MSD	0.22	0.13	0.17	0.13	0.11	0.06	0.08	0.06	0.06
FOM	0.94	0.98	0.95	0.97	0.97	0.99	0.99	0.99	0.99

Table 3.2: Quantitative results for reconstructed noisy checkerboard images.

MSE	41.06			30.47			25.43		
τ	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0
MSE	14.4	13.0	13.6	8.7	9.4	7.6	5.8	4.4	4.2
$NMSE$	0.31	0.28	0.30	0.195	0.20	0.16	0.12	0.09	0.09

The algorithm was first tested on 8-bit gray level images. We now present these results on synthetic and real images. In order to evaluate detection and localization performance, we used noisy checkerboard image with $nmse = 41.06$. The original and noisy checkerboard images are shown in Fig. 3.1. Surfaces and detected edges for three τ values 0.0, 0.5, and 1.0 are shown in Fig. 3.2. The conditional probability of a detected edge pixel given an ideal edge pixel, $Pr\{DE|IE\}$, the conditional probability of an ideal edge pixel given a detected edge pixel, $Pr\{IE|DE\}$, the mean square distance (MSD) between ideal and detected edge pixels and the Pratt's Of Merit (FOM). These performance indices for checkerboard images degraded at different amounts are given in Table 3.1. For highly noisy images the detected edges for $\tau = 0.5$ are always better in the sense of the performance indices introduced above while in other cases the results for $\tau = 1.0$ are better.

Table 3.2 contains mean square error (MSE) and normalized mean square error ($NMSE$) between the reconstructed and the original surfaces. The reconstructed surfaces obtained for $\tau = 0.5$ result in smaller MSE for highly noisy images. As compared to the Perona-Malik's diffusion [46], hybrid diffusion always results in better replica of the original image.

Finally, we present a comparison of the proposed scheme and linear $\lambda\tau$ -space representation and generalized edge detector (GED). We see that for the case of no noise (Fig.3.4 and Fig.3.6), the proposed algorithm performs as well as GED and in the presence

**Figure 3.1:** The original and noisy checkerboard images.

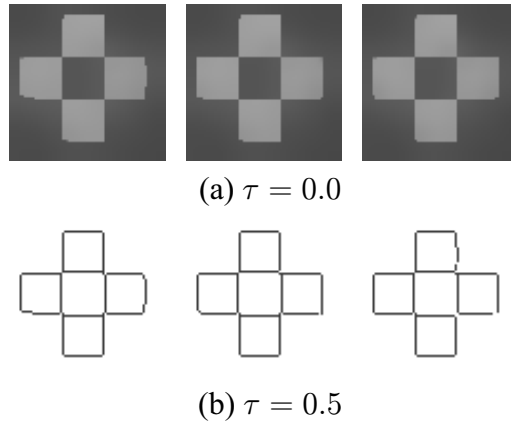


Figure 3.2: Restored checkerboard images and their edges.

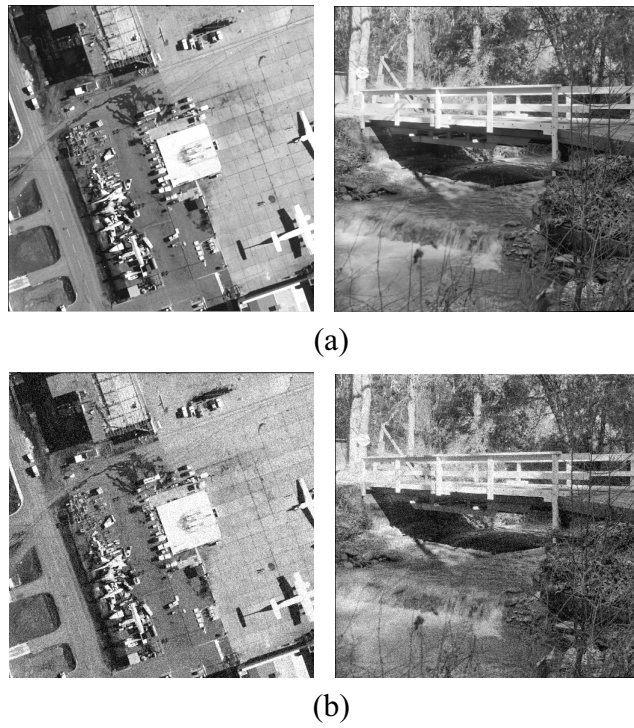


Figure 3.3: (a) Original Airfield and bridge images (b) Degraded Airfield and Bridge images with Gaussian noise $\sigma^2 = 451.36$ and $\sigma^2 = 375.73$.

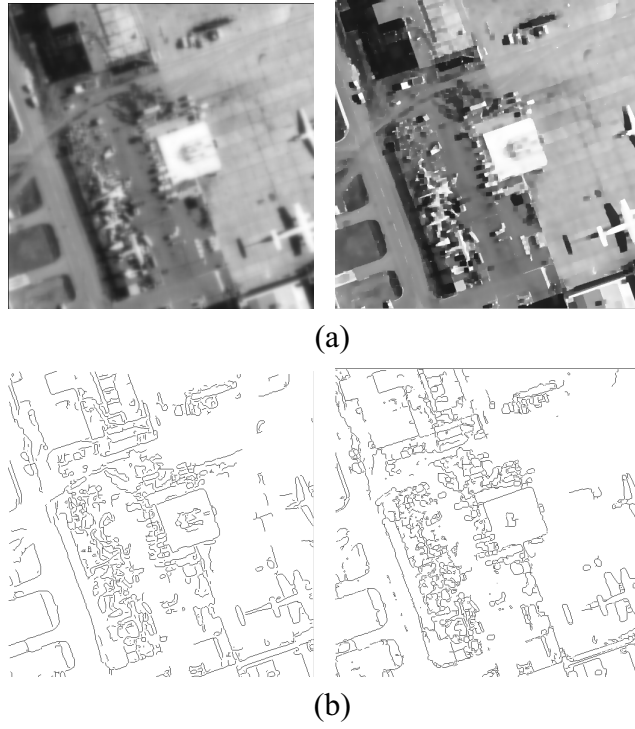


Figure 3.4: Reconstruction and edge detection results for the original Airfield image (a) linear R - and G -filter results (b) Results obtained by the proposed adaptive representation..

of high noise (Fig.3.5 and Fig.3.7), it outperforms GED. The original and noisy Airfield and Bridge images are given in Fig.3.3.

The algorithm is also tested on various real 24-bit color images. We now present these results. We compare the success of the adaptive representation over the linear representation on clear and noisy images. Fig.3.10 contains the linear $\lambda\tau$ -space representation of clear Parrot image for $\lambda = 1, 4, 16$ and $\tau = 0.0, 0.5, 1.0$. The SNR values corresponding to these images are also given in Table 3.3. The linear algorithm always degrades the image structure the least for $\tau = 0.5$. Adaptive representation of the same image for $\tau = 0.0, 0.5, 1.0$ is given in Fig. 3.9. The SNR values for the adaptive scheme are given in Table 3.4. On contrary to the linear space, the image structure is much better preserved, corresponding to the larger SNR values, when τ is close to 1. Fig. 3.10 presents the adaptive smoothing performance for three clear natural images. Fig. 3.12 shows the performance in the case of noise for *Lenna* image (Fig. 3.11). The best representation is achieved when τ is closer to 1.0.

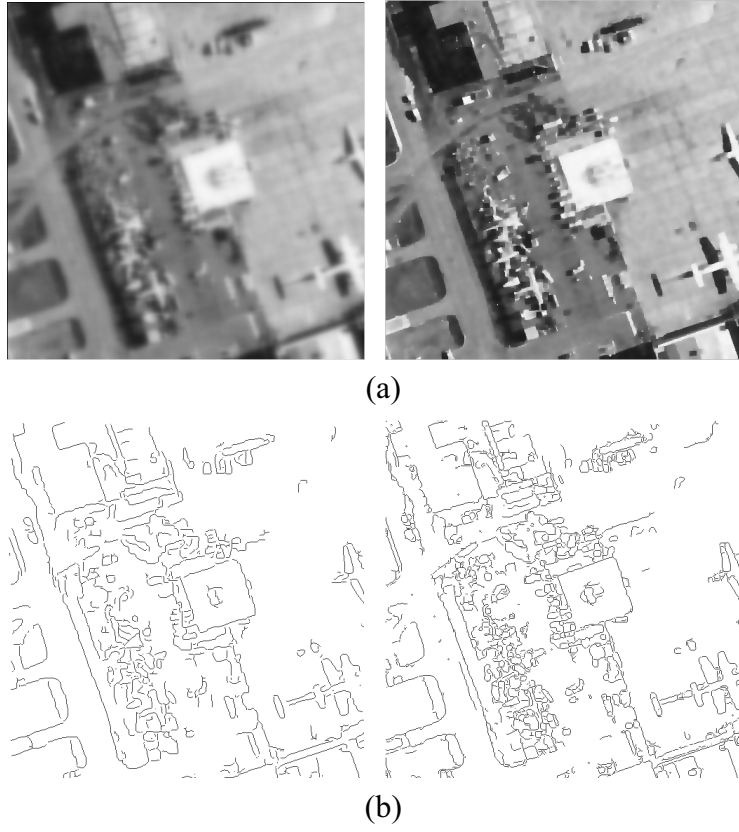


Figure 3.5: Reconstruction and edge detection results for the noisy Airfield image (a) linear R - and G -filter results (b) Results obtained by the proposed adaptive representation.

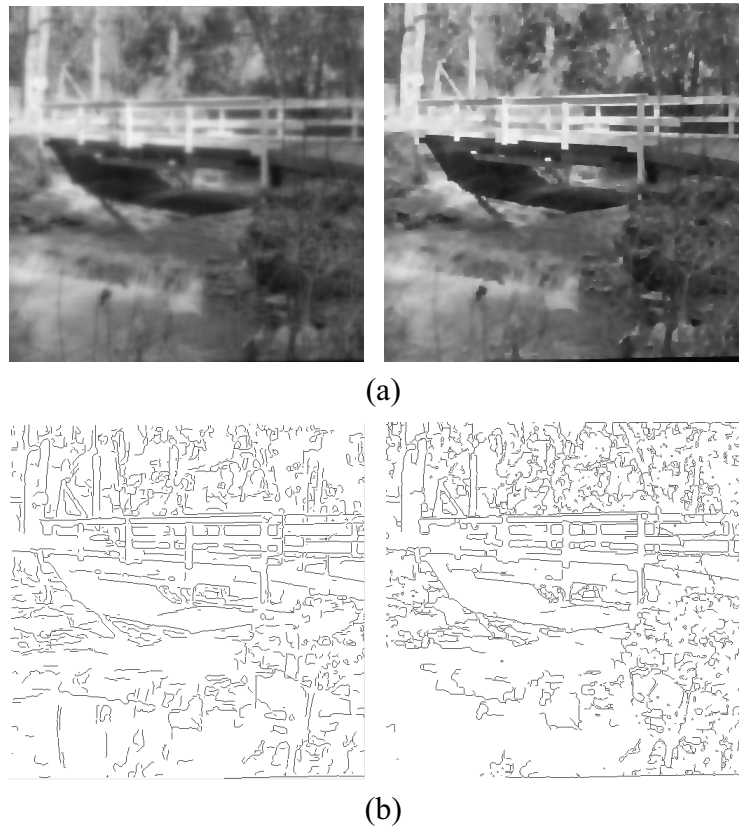


Figure 3.6: Reconstruction and edge detection results for the original Bridge image (a) linear R - and G -filter results (b) Results obtained by the proposed adaptive representation.

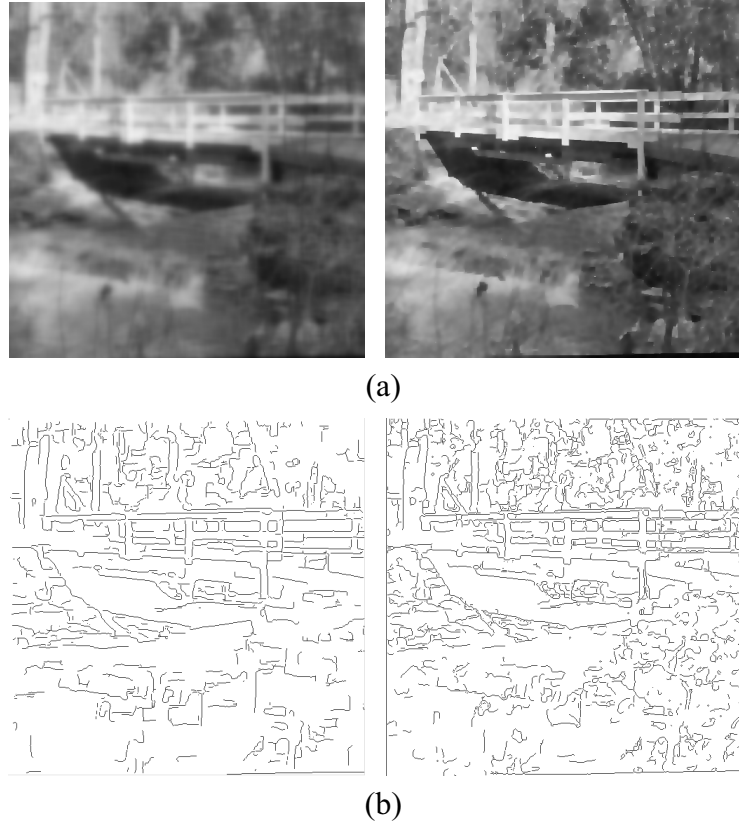


Figure 3.7: Reconstruction and edge detection results for the noisy Bridge image (a) linear R - and G -filter results (b) Results obtained by the proposed adaptive representation.

Table 3.3: SNR values for the reconstructed *Parrot* image.

λ/τ	0.0	0.5	1.0
1	12.92	15.43	14.52
4	11.06	13.63	13.27
16	8.20	9.62	12.12
<i>Proposed</i>	18.10	18.53	19.03

Table 3.4: SNR values for the reconstructed images.

Image/ τ	0.0	0.5	1.0
<i>Hats</i>	17.62	18.10	18.76
<i>House</i>	19.24	19.57	19.87
<i>Mandril</i>	15.30	16.20	17.46

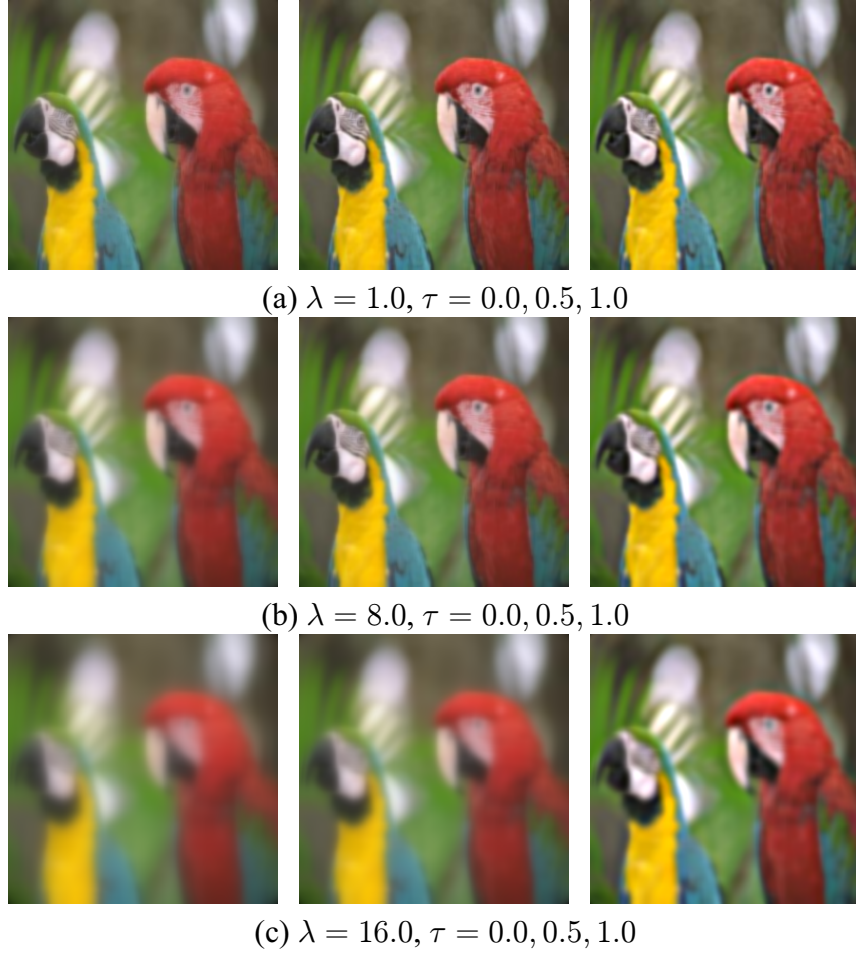


Figure 3.8: Linear $\lambda\tau$ -space representation of *Parrot* image.

3.5 Summary and Conclusions

We have presented nonlinear image and edge representation derived from the $\lambda\tau$ -space representation in which the space parameters are adaptively determined controlling the amount and the direction of smoothness. We have showed experimentally that our proposed scheme is capable of locating the feature direction robustly when the both terms in the hybrid diffusion are used contrary to the Perona-Malik equation. Using this approach one may obtain surfaces having different characteristics for the choice of different τ .

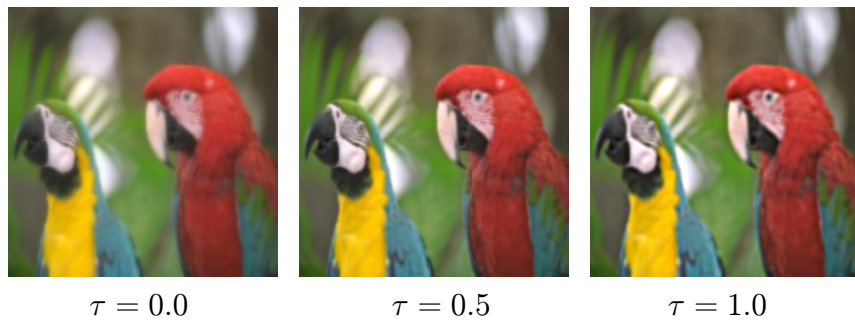


Figure 3.9: Adaptive smoothing of *Parrot* image.

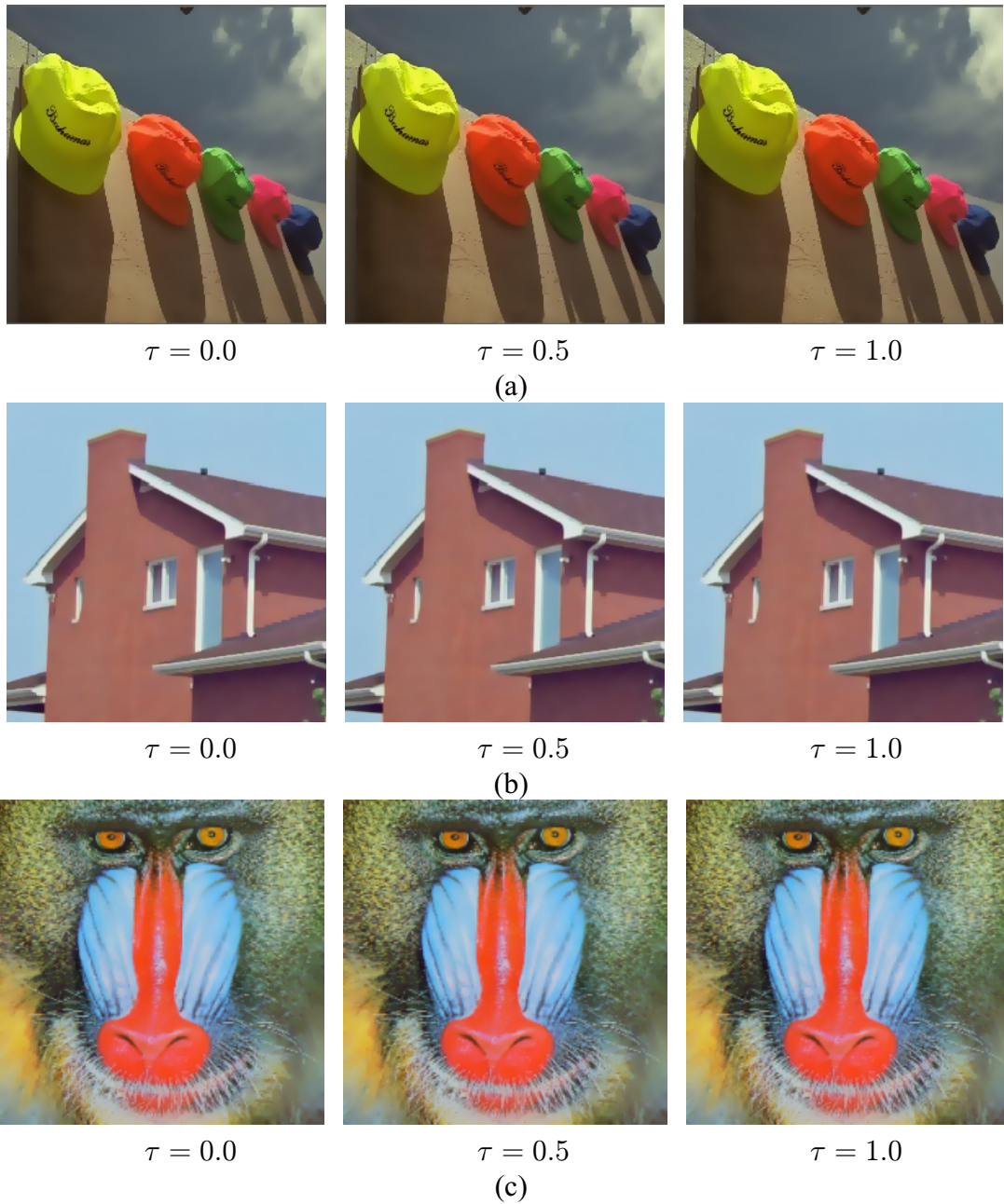


Figure 3.10: Adaptive smoothing of real images (a) *Hat* image (b) *House* image (c) *Mandrill* image.

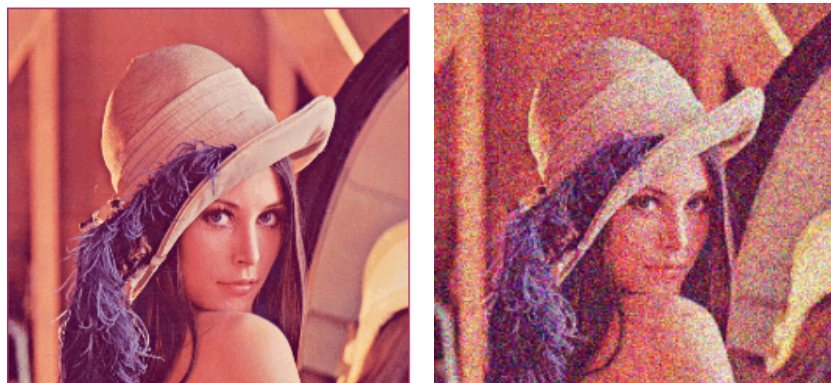
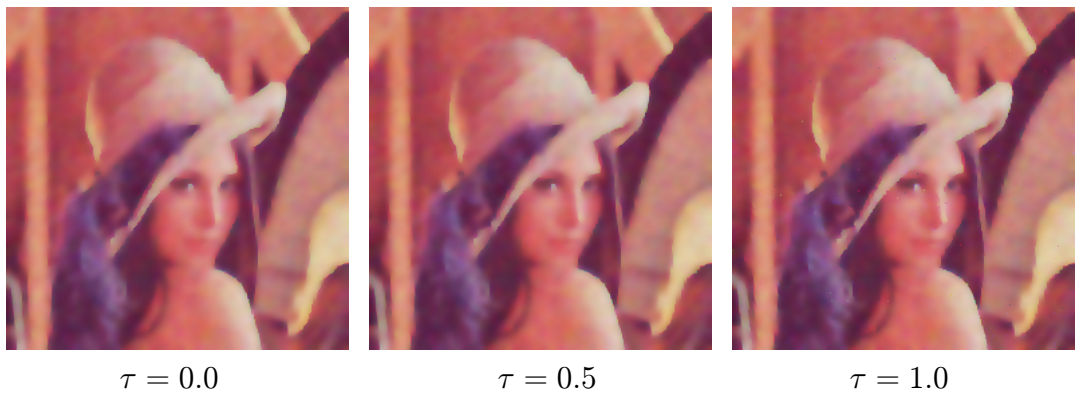


Figure 3.11: The original and noisy *Lenna* image.



$\tau = 0.0$

$\tau = 0.5$

$\tau = 1.0$

Figure 3.12: Adaptive smoothing of noisy *Lenna* image.

4. GRADIENT-BASED SHAPE DESCRIPTOR

Shape representation and description have been playing important roles in many areas of computer vision, pattern recognition, and robotics including character recognition, fingerprint matching and industrial inspection [52]. There are two recent tutorials on the shape description and matching techniques. Veltkamp and Hagedoorn [53] investigated the shape matching methods in four parts: global image transformations, global object methods, voting schemes and computational geometry. They also worked on shape dissimilarity measures. Another review on shape representation methods is accomplished by Zhang and Lu [54]. They first classified the solutions into two categories: contour-based and region-based methods, also referred to as external and internal techniques, respectively. The classification is based on whether shape features are extracted only from the contour or they are extracted from the whole shape region. Under each class, the different methods are further divided as structural approaches and global approaches. This sub-classification is based on whether the shape is represented as a whole or by segments/sections called primitives.

Our study falls into global contour-based shape description class utilizing only object boundaries. Two approaches most related to this study are shape signatures and shape matching. A shape signature represents a shape by a function extracted from object boundary points. Examples of shape signatures include complex coordinates, centroid distance and curvature. Shape signatures are usually needed to be normalized to satisfy scale and translation invariancy. Mostly shift matching is required to compensate the rotation variations between two shapes. Shape signatures are also sensitive to noise, local shape deformations and occlusion. In order to decrease the sensitivity to noise and local shape deformations, spectral transformations such as Fourier Transform and Wavelet Transform are applied to the signatures, and Fourier descriptors and Wavelet Descriptors are extracted. Matching is done in transform space in this case. For shape description, there is always a compromise between accuracy and efficiency. For efficiency the descriptor should be as much compact as possible to have fast retrieval and indexing. This leads to inaccurate shape descriptors. On the other hand, shape descriptors are expected to be precise to have high hit ratios in shape retrieval.

In contrast to the previous approach, shape matching works in the spatial domain and measures the point-to-point similarity between two shapes. Classical example of shape matching is Hausdorff distance [54]. One advantage of the Hausdorff distance is that it could make partial match. On the other hand, it is not translation, scale, and rotation invariant. Belongie et. al. [55] proposed a shape matching approach attaching a feature called shape context to each point on the boundary. Then they solved one-to-one correspondence problem assuming that corresponding points have the similar shape context. Tu and Yuille [56] presented an algorithm for shape matching based on generative model for how one shape can be generated by the other. The matching process is formulated by the expectation maximization. Although shape matching algorithms yield precise retrieval results, their computational cost due to matching is unacceptable for online shape retrieval systems.

In this study, our purpose is to close the gap between signature-based descriptors and shape matching using gradient-based local description by increasing the recognition performance of the signature-based approaches. The local descriptors are increasingly used in image recognition due to their robustness to occlusions and geometrical deformations. Yokono and Poggio [57] investigated the performance of local descriptors using various combinations of Gaussian derivatives with different orientations and scales for an object recognition task. They compared the performances in terms of selectivity and invariance to several affine transformations such as rotation, scale changes, brightness changes, and viewpoint changes. They reported that the Gaussian derivative descriptor outperformed other Gauss-like filter descriptors.

In this work, we propose two contour-based global shape description schemes using responses to a set of steerable filters along the object boundary. On contrary to Yokono and Poggio’s study [57] we use the gradient information to describe object shape instead of object texture.

In the proposed schemes, in order to capture the shape information, we extract local image gradient on the boundary while we trace it. The gradient feature is a two-dimensional vector which may have various orientation and magnitude depending on the local intensity distribution. Magnitude and orientation are of high importance when they are used as shape clues. The gradients are arranged into a shape clue matrix. This shape clue satisfies a rule called shifting property. The rule implies that when an object rotates around its center, the columns of the matrix rotate with respect to both the amount and the direction of rotation. Such discovery leads us to rotation-invariant shape signatures. The proposed affine-invariant shape signatures are explained in Section 2. The steerable G -filter kernels

are employed to obtain the local image gradient data in this study. Steerable filters are frequently used to detect local shape features in many region-based texture and shape description techniques. They can provide orientation information, which are sensitive to the presence of edges, bars and other simple image structures [59, 60, 61]. It has been shown that the generalized edge detector based on G -filter can deliver improved results when compared to first order Gaussian derivative [1]. In addition, G -filter is capable of producing different edge detectors including the first order Gaussian derivative as we change its two parameters (i.e. λ and τ). We are interested in the filter responses only along object boundaries and these responses are aggregated into a one-dimensional shape signature. Fourier Transform is then applied to the proposed shape signature in order to provide a representation invariant to starting point and to have compact description. Fourier Descriptor (FD) is one of the most widely used shape descriptor due to its low computation complexity, clarity and coarse to fine description capability [62, 63, 64, 65].

Different image signatures are reported in the literature based on “color”, “texture” or “shape”. Since we are only interested in shape based signatures, we consider the well-known “shape” signatures in our investigation such as complex coordinates, centroid distance, curve bending angles and curvature signature. Although boundary-based shape signatures described above provide a means by which the shape of the object is described, they have limited selectivity feature especially used with the content-based image retrieval systems. In order to solve this problem, we suggest an approach where we can manage the selectivity by scale and direction. Another advantage of the proposed method is that it can be applied to both grayscale and binary images. This is especially important when it has been used in content-based image retrieval systems due to the fact that the database are mostly comprised of color images.

The rest of the chapter is structured as follows: Chapter 4.1 introduces the directional gradient extraction methodology. Chapter 4.2 is for the proposed gradient based shape descriptors. The experimental results and conclusions are presented in Chapter 4.3 and 4.4, respectively.

4.1 Directional Gradient Extraction Using Steerable Filters

In this study, we are only interested in boundary based shape descriptors and assume that objects always have closed boundaries. Many shape descriptors exist in the literature and most of these descriptors are not able to address varieties of shape variations in nature such as rotation, scale, skew, stretch, and noise. We propose an affine invariant shape descriptor in this study, which handles rotation, scale and skew transformations. The basic idea of

the proposed approach is to use gradient information at the boundaries rather than using the boundary locations alone. We used 2D Generalized Edge Detector [1] to obtain object boundary, but other well-known edge operators like Canny's edge detector [66] can also be used. We then trace the detected boundary pixels along the clockwise direction to attain the locations of the neighboring boundary pixels denoted as (x_i, y_i) . So the object boundary is easily obtained in the form of a matrix Γ of size n -by-2

$$\Gamma = \{[x_i, y_i]^T; i = 1, 2, \dots, n\}, \quad (4.1)$$

where n is the length of the contour, i.e. $n = |\Gamma|$. Since we are interested in the directed gradients at these boundary locations, we utilize steerable G -Filters to obtain the gradient at certain directions and scales as

$$D_{(\lambda, \tau)}^\theta (I, x_i, y_i) = (I * G_{(\lambda, \tau)}^\theta) (x_i, y_i), \quad (4.2)$$

where I is the image intensity. Steerable $G_{(\lambda, \tau)}^\theta$ filter is defined in terms of $G_{(\lambda, \tau)}^{\theta=0}$ as

$$\begin{aligned} G_{(\lambda, \tau)}^\theta (x', y') &= G_{(\lambda, \tau)}^{\theta=0} (x, y) \\ [x', y']^T &= R(\theta) [x, y]^T, \end{aligned} \quad (4.3)$$

where $R(\theta)$ is the rotation matrix, i.e.

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (4.4)$$

Detailed analysis of these filters ($G_{(\lambda, \tau)}^{\theta=0}$) is given in [1]. In case G is not available, one can use the first derivative of Gaussian as G filter. Let us denote the response matrix as $F(\Gamma) = [f_{k,m}]$ where $f_{k,m}$ is equal to $D_{(\lambda, \tau)}^{\theta_m} (x_k, y_k) = (I * G_{(\lambda, \tau)}^{\theta_m}) (x_k, y_k)$ as

$$F(\Gamma) = \begin{bmatrix} D_{(\lambda, \tau)}^{\theta_0} (x_0, y_0) & \cdots & D_{(\lambda, \tau)}^{\theta_{M-1}} (x_0, y_0) \\ D_{(\lambda, \tau)}^{\theta_0} (x_1, y_1) & \cdots & D_{(\lambda, \tau)}^{\theta_{M-1}} (x_1, y_1) \\ \vdots & \ddots & \vdots \\ D_{(\lambda, \tau)}^{\theta_0} (x_{N-1}, y_{N-1}) & \cdots & D_{(\lambda, \tau)}^{\theta_{M-1}} (x_{N-1}, y_{N-1}) \end{bmatrix} \quad (4.5)$$

assuming that we use M number of steerable filters whose directions are multiple of π/M such that $\theta_m = m\pi/M$. In this case, the size of F is $M |\Gamma|$. As we increase the number of filters M , hence decrease the angles between directions, the local shape is better described by the filter responses.

Another alternative to the previous gradient-based signature definition is to treat the gradients in the perpendicular directions as a complex number rather than treating them as independent data. In this case, the response matrix takes the following form

$$H(\Gamma) = \begin{bmatrix} (D^{\theta_0} + jD^{\theta_0 + \frac{\pi}{2}})(x_0, y_0) & \cdots & (D^{\theta_{M-1}} + jD^{\theta_{M-1} + \frac{\pi}{2}})(x_0, y_0) \\ (D^{\theta_0} + jD^{\theta_0 + \frac{\pi}{2}})(x_1, y_1) & \cdots & (D^{\theta_{M-1}} + jD^{\theta_{M-1} + \frac{\pi}{2}})(x_1, y_1) \\ \vdots & \ddots & \vdots \\ (D^{\theta_0} + jD^{\theta_0 + \frac{\pi}{2}})(x_{N-1}, y_{N-1}) & \cdots & (D^{\theta_{M-1}} + jD^{\theta_{M-1} + \frac{\pi}{2}})(x_{N-1}, y_{N-1}) \end{bmatrix} \quad (4.6)$$

where $\theta_m = \frac{m\pi}{2M}$.

When the object is rotated α degrees around the center of gravity, the columns of the matrix F is circularly shifted to left or right depending on whether the objects rotate clockwise or counterclockwise direction. The relationship between the rotation angle and amount of shifting can be stated as follows assuming that the rotation angle is multiple of $\frac{\pi}{M}$,

$$F(\Gamma^\alpha) = [f'_{k,m}], \alpha = s \frac{\pi}{M}, \quad (4.7)$$

$$f'_{k, (m+s) \bmod M} = f_{k,m}.$$

where $\Gamma^\alpha = \{R(\alpha)[x_i, y_i]^T\}$. This property is called as perfect circular-shifting rule.

In the first formulation, $F(\Gamma)$ is a matrix of N -rows and M -columns while in the second formulation $H(\Gamma)$ is a matrix of N -rows and $M/2$ -columns. On the other hand, $F(\Gamma)$ is a real valued matrix while $H(\Gamma)$ is a complex valued matrix. Note that both matrices $F(\Gamma)$ and $H(\Gamma)$ satisfy the perfect shifting rule under the rotations when θ is exact multiple of $\frac{\pi}{M}$.

In order to demonstrate the circular-shifting rule, we obtain the filter response matrix (F) at different steering angles both for the object (the digit “5”) and its rotated version. We first apply the filter at each pixel on the boundary of the sample object given in Fig. 4.1a and then apply it on the boundary of its 90° rotated version given in Fig 4.1d. The steering angle is changed from 0° to 180° with 1° increments. The response matrices are plotted in Fig. 4.1b and in Fig. 4.1e for 0° and 90° respectively. Fig. 4.1c shows how the steerable filter response changes with the rotation angle at one pixel on the boundary. In other words, Fig. 4.1c is obtained by taking the vertical cross-section of Fig. 4.1b. The cross-section of F for the rotated object on the same pixel is shown in Fig. 4.1f. It can be easily verified from the figures Fig. 4.1c and Fig. 4.1f that they satisfy the equation (4.7) where $s = 90$ and $M = 180$. The experimental results are consistent with the circular-shifting rule.

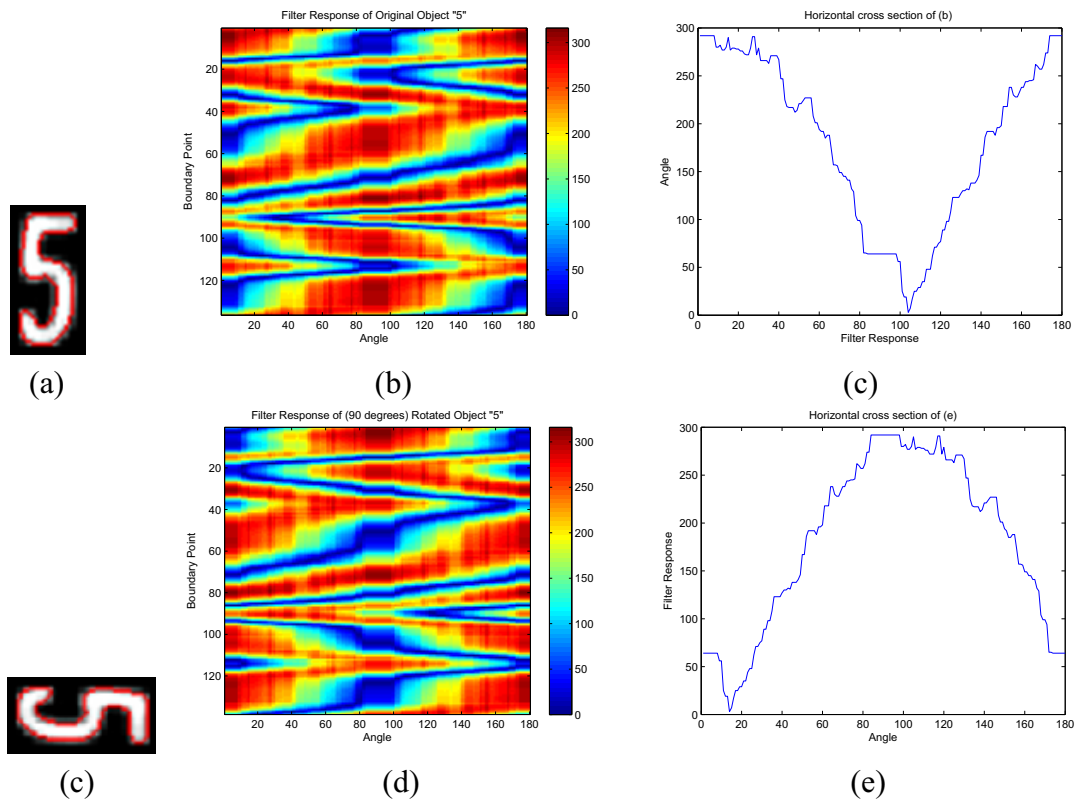


Figure 4.1: (a) Sample digit "5", (b) Filter responses on object boundary pixels for each direction, (c) Steerable filter response at one pixel on the curve with respect to steering angle, which corresponds to a vertical cross section of (b). (d) 90 degrees rotated digit (e) Filter responses on object boundary pixels for each angle, (f) Steerable filter response at the same pixel on the curve with respect to rotation angle, which corresponds to a horizontal cross section of (e).

We also analyze how the circular-shifting rule is affected when the rotation angle is not exact multiple of $\frac{\pi}{M}$. Let us assume that rotation angle is $s\frac{\pi}{M} + \phi$ where $0 \leq \phi < \frac{\pi}{M}$ and we have M steerable filters $\{G^{\theta_k}; k = 0, 1, \dots, M-1\}$. Any steerable filter response in one direction can be expressed as a linear combination of M different G^{θ_k} filter responses as [57]

$$G^\varphi = \sum_{k=0}^{M-1} c_k(\varphi) G^{\theta_k}, \quad (4.8)$$

$$c_k(\varphi) = \cos\left(\varphi - \frac{k\pi}{M}\right) = \cos(\varphi - \theta_k).$$

The difference between the signatures obtained for the same object, one is rotated $\alpha = s\frac{\pi}{M}$ degrees and the other is rotated $\alpha + \phi = s\frac{\pi}{M} + \phi$ degrees can be found as

$$\begin{aligned} \|f_{k,m}\|^2 &= \|I^\alpha * G^{\theta_m}\|^2 \\ \|\hat{f}_{k,m}\|^2 &= \|I^{\alpha+\phi} * G^{\theta_m}\|^2 = \|I^\alpha * G^{\theta_m+\phi}\|^2 \\ \|f_{k,m} - \hat{f}_{k,m}\|^2 &= \|I^\alpha * G^{\theta_m} - I^\alpha * G^{\theta_m+\phi}\|^2 \\ \|f_{k,m} - \hat{f}_{k,m}\|^2 &= \|I^\alpha * (G^{\theta_m} - G^{\theta_m+\phi})\|^2 \\ \|f_{k,m} - \hat{f}_{k,m}\|^2 &\leq \|I^\alpha\|^2 \|G^{\theta_m} - G^{\theta_m+\phi}\|^2. \end{aligned} \quad (4.9)$$

Now the equation (4.8) could be used to compute the difference between the responses obtained for θ_m and $\theta_m + \phi$ degrees rotated filter kernels as

$$\|G^{\theta_m} - G^{\theta_m+\phi}\|^2 = \|G^{\theta_m} - \sum_{k=0}^{M-1} \cos(\theta_m + \phi - \theta_k) G^{\theta_k}\|^2. \quad (4.10)$$

This formulation helps us to decide how many directions we should use, hence to determine the size of the descriptor. Our aim is to keep the difference ($\epsilon = \|f_{k,m} - \hat{f}_{k,m}\|$) as small as possible. For a given ϵ , one can easily obtain M using the above equation. ϵ can be considered as desired description level. Shape is better described as we decrease ϵ . There is a tradeoff between description level and computational cost. The computation cost increases as the descriptor size increases. Experimental results reveal that we don't need to use large numbers of filters for $F(\Gamma)$, and even $M = 2$ is sufficient when $H(\Gamma)$ is used as signature. Next section will discuss how to use Fourier Descriptor applied to the gradient signature that we discussed above and how to define a distance function between two descriptors.

4.2 Affine Invariant Shape Descriptor

Fourier Descriptors (FDs) are mostly employed for boundary shape description. Zhang and Lu [65] compared shape retrieval using FDs derived from different shape signatures and from different Fourier invariants in terms of computational complexity, robustness, convergence speed, and retrieval performance. They reported that the centroid distance shape signature outperforms other signature methods in terms of above criteria. Selecting the shape signature is the most critical step for FDs. Various signature models were proposed in the literature such as complex boundary coordinates, centroid distance and boundary curvatures. These signatures are summarized below.

Complex Coordinate Signature: The boundary curve (Γ) of the form $(x(t), y(t))$ is combined into a complex function as

$$z(t) = (x(t) - x_c) + j(y(t) - y_c) \quad (4.11)$$

where (x_c, y_c) is the centroid of the shape, defined as

$$x_c = \frac{1}{N} \sum_{i=0}^{N-1} x(i), \quad y_c = \frac{1}{N} \sum_{i=0}^{N-1} y(i). \quad (4.12)$$

Here the boundary curve is assumed to be periodic

$$z(t) = z(t + np), p = |\Gamma|, n \in \mathbb{Z} \quad (4.13)$$

Centroid Distance Signature: Centroid distance function is defined by the distance of the contour points to the centroid of the object

$$r(t) = [(x(t) - x_c(t))^2 + (y(t) - y_c(t))^2]^{1/2} \quad (4.14)$$

Curvature Signature: Curvature of a contour at a point is represented in terms of the first and the second derivatives of coordinate functions as

$$\kappa(t) = \frac{x'(t)y''(t) - y'(t)x''(t)}{(x'(t)^2 + y'(t)^2)^{3/2}} \quad (4.15)$$

where $x'(t), y'(t)$ and $x''(t), y''(t)$ are the first and second derivatives of coordinate functions. In order to use the curvature signature for shape representation, a smoothing process should be applied first. We utilized a Gaussian smoothing kernel on coordinate

functions as

$$x^s(t) = x(t) * G(\mu, \sigma), \quad y^s(t) = y(t) * G(\mu, \sigma) \quad (4.16)$$

where $x_s(t)$ and $y_s(t)$ are smoothed coordinate functions and $G(\mu, \sigma)$ is the Gaussian kernel.

The Proposed Shape Descriptors: We apply Fourier transform to the gradient based shape signatures described in the previous section in order to obtain a compact descriptor. We denote the descriptors as $\tilde{F} = \left[\left\| \tilde{f}_{k,m} \right\| \right]$ and $\tilde{H} = \left[\left\| \tilde{h}_{k,m} \right\| \right]$ where the coefficients are computed as follows

$$\begin{aligned} \tilde{f}_{k,m} &= \frac{1}{N} \sum_{n=0}^{N-1} f_{n,m} \exp\left(\frac{-j2\pi kn}{N}\right) \\ \tilde{h}_{k,m} &= \frac{1}{N} \sum_{n=0}^{N-1} h_{n,m} \exp\left(\frac{-j2\pi kn}{N}\right) \end{aligned} \quad (4.17)$$

where N is the length of the curve (Γ) and $k \in [1..L]$. We take the magnitude of the Fourier coefficients to keep the descriptor invariant to starting point on the boundary ([68]). Note also that both $\tilde{F}$ and $\tilde{H}$ still satisfy the circular-shifting rule when we rotate the object as

$$\begin{aligned} \tilde{F}(\Gamma) &= \left[\tilde{f}_{k,m} \right] \\ \tilde{F}(\Gamma_\alpha) &= \left[\tilde{f}'_{k,m} \right], \alpha = s \frac{\pi}{M} \\ \tilde{f}'_{k,m} &= \tilde{f}_{k,(m+s) \bmod M}, \end{aligned} \quad (4.18)$$

which also applies to $\tilde{H}$ and $\tilde{h}_{k,m}$. The proof is given in the Appendix A.

Scale invariance of the descriptor is achieved by dividing the magnitudes by the DC component, i.e., $\tilde{f}_{0,m}$ ([68]). Finally we present a distance function to compare two descriptors. Assume that we are given two descriptors such as $\tilde{f}_{k,m}$ and $\tilde{p}_{k,m}$. Then we define the distance between two descriptors considering the shifting-property as

$$SD(\tilde{f}, \tilde{p}) = \min_{r \in \{0, \dots, M-1\}} \left\| \tilde{f}_{k,(m+r) \bmod M} - \tilde{p}_{k,m} \right\|_2. \quad (4.19)$$

The distance between two descriptors (e.g. $\tilde{f}_{k,m}$ and $\tilde{f}'_{k,m}$) that belong to one object and its s -rotated form (corresponding to the angle $s \frac{\pi}{M}$) is given by

$$SD(\tilde{f}, \tilde{f}') = \min_{r \in \{0, \dots, M-1\}} \left\| \tilde{f}_{k,(m+r) \bmod M} - \tilde{f}'_{k,m} \right\|_2 = 0. \quad (4.20)$$

Table 4.1: Rotation angle estimation error $\Theta()$ for various M .

M	Estimation Error – $\Theta()$
2	15.18°
4	11.45°
6	7.61°
8	5.76°
10	4.69°
12	3.95°
14	3.50°
16	3.05°

where $\tilde{f}'_{k,m} = \tilde{f}_{k,(m+s) \bmod M}$ due to circular-shifting rule. In addition, an estimate of rotation can be computed as

$$\Theta(\tilde{f}, \tilde{p}) = \left(\arg \min_{r \in \{0, \dots, M-1\}} \left\| \tilde{f}_{k,(m+r) \bmod M} - \tilde{p}_{k,m} \right\|_2 \right) \frac{\pi}{M} \quad (4.21)$$

assuming that $\tilde{f}$ and $\tilde{p}$ are in the same object class. The estimation error is upper bounded by $\frac{\pi}{2M}$. Several experiments are performed again on the same digit “5” to study the actual estimation error for various M values (2, 4, 6, 8, 10, 12, 14, 16). The results are given in Table.4.1. It shows that the actual error is much smaller than the theoretical upper-bound.

The computational complexity of the equation (4.19) is $M \times (M \times L)$ where M is the number of filter kernels used and L is the number of Fourier coefficients used.

4.3 Experimental Results

The performance of the proposed descriptors is evaluated over two databases; one containing digits taken from vehicle license plates and the other containing MPEG-7 Core Experiment shape data set. First we demonstrate the performance of our algorithm on the license plate characters. Fig. 4.2 shows typical segmented license plate characters selected from the database which contains 8321 grayscale digit characters segmented from the real vehicle license plates. About the half of the characters in the database, namely 4121 digits, is used in training, and the remaining 4200 digits are kept for testing. In practice, although there are 10 digits appear on the license plate images, we only labeled 9 classes because one can not distinguish between 6 and 9 under rotation assuming that no prior information is available.

In the first group of experiments, we explored how the recognition rate changes with the size of descriptor. The descriptors have two dimensions: number of kernel filters denoted

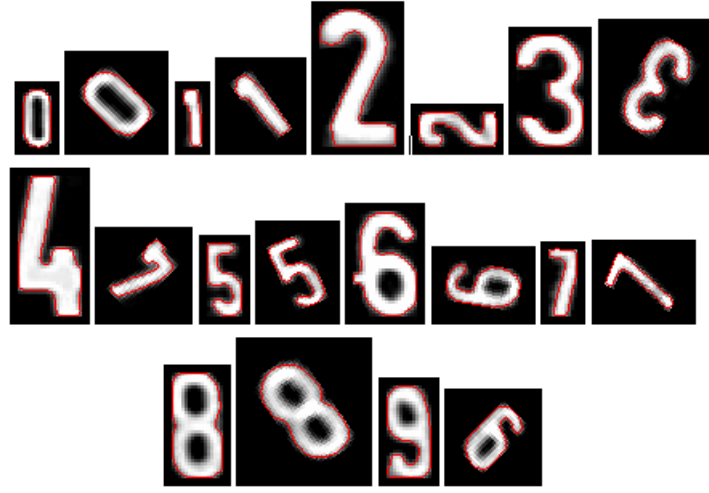


Figure 4.2: Typical segmented license plate characters selected from the database.

as M and the number of Fourier coefficients denoted as L . The results are summarized in Table 4.2. The proposed descriptor $\tilde{F}$ results in 99% recognition rate for $M > 6$ and $L > 3$. So the optimum value of L can be selected as 4. When compared to centroid distance with a performance of at most 90% for $L = 15$, this is a big improvement. Another observation is that the performance of centroid distance drops dramatically when L gets smaller. For example, the recognition performance is 60% for $L = 3$ whereas the performance of the proposed descriptor which is 95% is much higher, for $M = 8$ and $L = 3$. In Fig. 4.3, we plot the recognition rates, as a function of L for four methods, i.e. the proposed descriptors ($\tilde{F}(M = 8)$, $\tilde{H}(M = 2)$), centroid distance, boundary curvature, and complex coordinates. It shows a considerable improvement achieved by the proposed methods. We presented only one result for the descriptor $\tilde{H}(M = 2)$ in Table 4.2 since the recognition rate performance obtained for $M > 2$ is very close to the performance obtained for $M = 2$ as shown in Table 4.4.

The fact that the performance of the descriptor $\tilde{H}$ does not change considerable as M increases is very important in several aspects. First of all, for $M = 2$ the descriptor $\tilde{H}$ has only one column. In this special case, the corresponding signature vector can be easily computed by using horizontal and vertical components of gradient operator. Secondly the distance measure in (4.19) reduces to Euclidean distance since shifting and minimum operations are not required in this case. Another advantage is that the distance measure is a metric now.

The recognition performance of the descriptor $\tilde{F}$ is more sensitive to M and almost constant along the dimension L . On the other hand, the descriptor $\tilde{H}$ is sensitive to L and almost constant along the dimension M . Since the computational complexity is given

Table 4.2: Recognition rate with respect to descriptor size (M, L) .

Method	Number of coefficients					
	15	10	7	5	3	2
Centroid distance	90.21	90.14	84.54	76.11	59.90	52.95
Curvature	89.5	88.88	88.45	85.45	67.54	55.28
Complex coordinates	93.66	89.28	91.54	73.19	50.42	35.9
$\tilde{F}(M = 2)$	57.40	54.04	47.59	44.97	32.54	24.02
$\tilde{F}(M = 3)$	82.69	81.40	76.83	73.40	59.26	42.61
$\tilde{F}(M = 4)$	91.71	89.52	86.11	82.19	67.71	54.38
$\tilde{F}(M = 5)$	96.14	95.64	93.57	90.35	81.81	71.45
$\tilde{F}(M = 6)$	97.66	97.69	96.02	95.21	88.88	79.42
$\tilde{F}(M = 8)$	99.28	99.47	99.14	98.73	95.73	88
$\tilde{F}(M = 12)$	99.04	99.09	99.19	99.23	98	94.97
$\tilde{F}(M = 16)$	99.16	99.42	99.33	99.11	98	96
Complex gradient ($\tilde{H}$)	98.54	93.54	80.73	55.0	35.34	28.57

by M^2L , the descriptor $\tilde{H}$ manages the tradeoff between the descriptor size (ML) and the computational cost better.

We now consider the case that the edges are detected somewhat far away from their ideal locations. We demonstrate the recognition results over the same data set, but this time we use poorly detected edges that are obtained at the early stages of a fast marching algorithm [67] without using global shape information (Fig. 4.4). The recognition results are given in Table 4.3. With regard to the results shown for the ideal edges and poorly detected edges in Table 4.2 and Table 4.3, the superiority of the proposed descriptor over the centroid distance and curvature is now much more evident than the case with the ideal edges. The results show that the proposed method is more robust to the errors introduced during the contour detection stage as compared to other methods.

In the second group of experiments, we explored how the recognition rate changes with the scale parameters, i.e. (λ, τ) , of the filter kernel. We change the filter size from 7 to 15, hence the scale, λ , from 1.5 to 3.0. The results are summarized in Table 4.5. In this experiment, the highest recognition rate is achieved for $\lambda = 2.5$.

In the last group of experiments over the license plate characters, we explored how M and ϕ effect the shifting property of the descriptor. We rotate the same object by 1° angles and compute the distance between the original object and its rotated versions by using Equation 4.19. For the sake of brevity only the average distances are presented in Table 4.6. We apply the same method to both gray level and binary objects. The first row of the Table 4.6 holds the average distances for 9 different kernel sets applied to binary objects. As M increases we describe the shape much better as indicated by (4.10). Results from

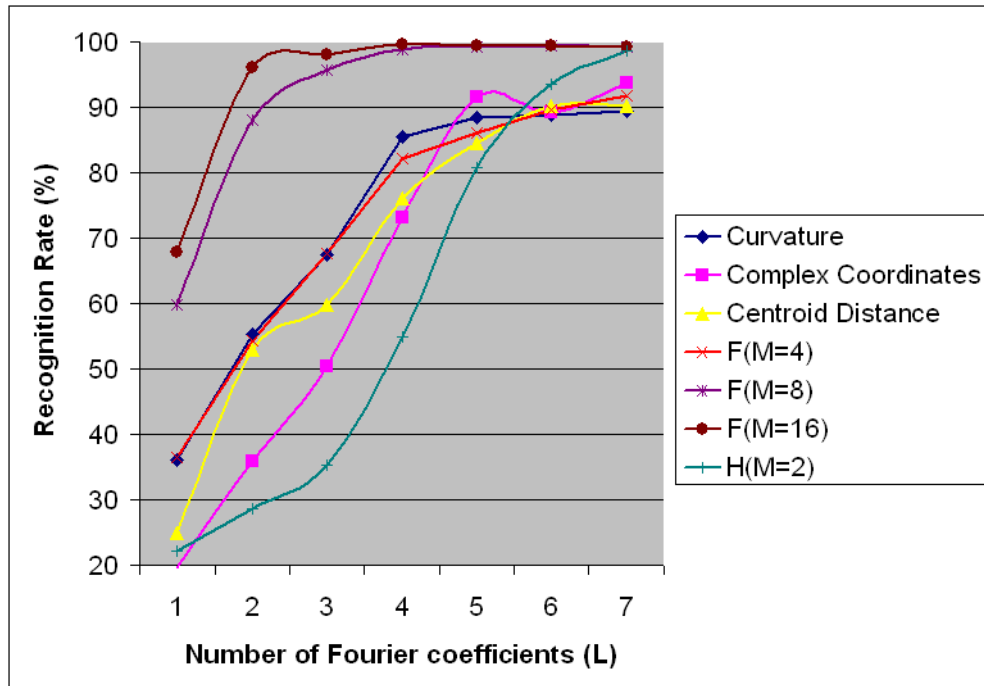


Figure 4.3: Recognition rates for four methods, the proposed descriptor ($M = 8$), centroid distance, boundary curvature, complex coordinates with respect to L where L is the number of Fourier coefficients used.

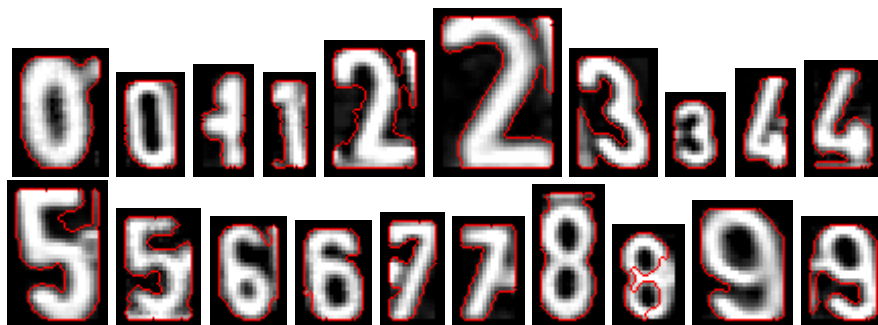


Figure 4.4: Poorly segmented license plate characters that are selected from the database.

Table 4.3: Recognition rate with respect to descriptor size (M, L) obtained for the license plate digits where poorly detected edges are used.

Method	Number of coefficients					
	15	10	7	5	3	2
Centroid distance	88.66	87.74	85	76.72	61.13	48.12
Curvature (5)	86.3	85.28	83.6	79.31	69.25	57.23
Complex coordinates	88.61	68.88	49.09	33.31	24.12	20.91
$\tilde{F}(M = 2)$	93.77	93.2	91.61	88.72	72.49	55.74
$\tilde{F}(M = 3)$	96.07	95.68	94.88	92.63	83.39	71.22
$\tilde{F}(M = 4)$	96.81	96.7	96.38	95.45	89.53	80.73
$\tilde{F}(M = 5)$	97.15	96.94	96.62	96.13	93.18	87.81
$\tilde{F}(M = 6)$	97.47	97.3	97.09	96.7	95	91.31
$\tilde{F}(M = 8)$	97.32	97.33	96.98	96.58	94.64	91.54
$\tilde{F}(M = 12)$	97.40	97.53	96.87	96.38	94.41	91.52
$\tilde{F}(M = 16)$	97.17	96.98	96.83	96.36	94.41	91.29
Complex gradient ($\tilde{H}$)	90.88	81.6	59.8	47.63	30.15	23.84

gray-level objects are given in the second row of this table. The results are collectively plotted in Fig. 4.5 for different values of $M \{2, 4, 5, 6, 8, 10, 12, 14, 16\}$. As M increases, the rotation invariancy property of the defined distance measure becomes more evident.

We have tested our algorithm on a subset of MPEG-7 Core Experiments Shape-1 Part B which contains 20 images for each 42 binary objects. Table 4.3 summarizes the recognition rates with respect to the both rank and descriptor size for the proposed descriptors and the well-known Fourier-based shape descriptors including centroid distance and boundary curvature. As seen, the proposed method with $M \geq 4$ outperforms the other methods for all rank values. In the experiments, we keep 10 images for training and 10 images for testing. We have also performed another series of experiments to asses the effect of training size on the recognition rate. The results are given in Table 4.5 for $\tilde{F}(M = 7)$. We present the rank 10 query results in Fig. B.1 where the first image is the query image and the remaining images are the first 10 images retrieved from the database. We used the same descriptor $\tilde{F}(M = 7)$ again in this experiment. It is seen that the recognition rate increases as the training size increases for this data set.

In addition to these data sets, we evaluate the performance of the proposed descriptors on the Kimia data set [69] containing 99 binary images of 9 objects. The numbers of correctly retrieved objects are given in Table.B.2 for the proposed descriptors ($\tilde{F}$ and $\tilde{H}$) and the other contour-based methods. As seen, the proposed descriptors outperform the other contour-based methods on this database as well.

Table 4.4: Recognition rate obtained for the descriptor $\tilde{H}$.

Method	Number of coefficients					
	15	10	7	5	3	2
Centroid distance	88.66	87.74	85	76.72	61.13	48.12
Curvature(5)	86.3	85.28	83.6	79.31	69.25	57.23
Complex coordinates	88.61	68.88	49.09	33.31	24.12	20.91
$\tilde{H}(M = 2)$	98.54	96.76	93.4	68.61	34.64	20.23
$\tilde{H}(M = 4)$	98.61	96.59	93.23	68.26	34.71	19.76
$\tilde{H}(M = 6)$	98.64	96.45	93.73	68.02	34.4	20.69
$\tilde{H}(M = 8)$	98.59	96.35	93.09	68.42	33.8	19
$\tilde{H}(M = 12)$	98.67	96.54	93.66	68.04	34.23	20.9
$\tilde{H}(M = 16)$	98.54	96.42	93.23	68.57	34.88	20.8

Table 4.5: Recognition rate with respect to filter scale.

	(Filter Size, λ, τ)			
	(7, 1.5, 0.225)	(9, 2, 0.268)	(11, 2.5, 0.3)	(13, 3, 0.33)
The proposed descriptor ($M = 8, L = 5$)	97.19	98.73	99.59	99.54

Table 4.6: Average distances between the object and its 1° rotated versions (Binary object results are given in the first row, and Grayscale object results are given in the second row).

$M = 2$	$M = 4$	$M = 5$	$M = 6$	$M = 8$	$M = 10$	$M = 12$	$M = 14$	$M = 16$
0.018	0.013	0.013	0.013	0.013	0.012	0.012	0.012	0.012
0.012	0.007	0.007	0.006	0.006	0.005	0.005	0.004	0.004

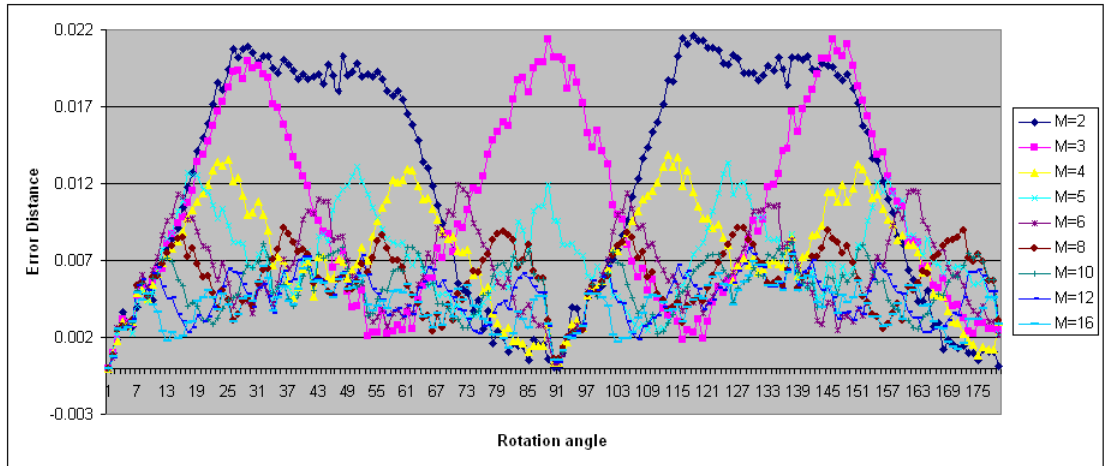
**Figure 4.5:** Distance error between the object and its rotated versions for $M = 2, 4, 5, 6, 8, 10, 12, 14, 16$.

Table 4.7: Recognition rates for the MPEG-7 Core Experiments Shape data set with respect to rank and descriptor size.

Method	Rank									
	1	2	3	4	5	6	7	8	9	10
$\tilde{F}(M=2)$	77.8	82.81	84.96	86.39	87.35	89.22	90.21	90.69	90.93	92.36
$\tilde{F}(M=3)$	84.96	89.26	90.69	92.84	93.07	93.79	94.27	94.99	95.22	95.22
$\tilde{F}(M=4)$	88.06	93.55	94.74	95.7	95.94	96.18	96.18	96.42	97.13	97.13
$\tilde{F}(M=5)$	88.3	93.31	94.03	95.22	95.94	96.42	96.89	97.37	97.85	97.85
$\tilde{F}(M=6)$	89.26	93.07	95.22	97.61	97.61	97.85	98.09	98.32	98.32	98.32
$\tilde{F}(M=7)$	90.93	95.22	96.65	97.61	98.09	98.32	98.32	98.32	98.56	98.56
$\tilde{F}(M=8)$	89.73	93.79	95.94	96.65	96.89	97.61	97.61	98.09	98.32	98.32
$\tilde{F}(M=9)$	90.69	95.22	96.42	97.13	97.37	98.09	98.56	98.8	98.8	98.8
$\tilde{F}(M=10)$	89.02	94.98	96.65	97.85	98.09	98.32	98.32	98.32	98.32	98.32
$\tilde{F}(M=11)$	90.93	94.98	97.13	97.61	98.09	98.09	98.32	98.32	98.32	98.32
$\tilde{F}(M=12)$	89.97	94.27	96.89	97.61	97.85	97.85	98.09	98.32	98.32	98.56
$\tilde{F}(M=13)$	90.45	94.03	96.42	97.61	97.85	97.85	98.09	98.56	98.56	98.56
$\tilde{F}(M=14)$	88.78	93.31	95.22	96.89	97.37	97.61	97.85	97.85	98.32	98.32
$\tilde{F}(M=15)$	89.73	94.03	95.7	97.13	97.61	97.85	97.85	97.85	98.09	98.09
$\tilde{F}(M=16)$	89.26	94.27	96.65	97.61	98.32	98.32	98.32	98.32	98.32	98.32
$\tilde{H}$	89.97	94.03	95.22	95.7	95.94	85.94	96.18	96.65	96.89	97.13
curvature	67.78	72.31	74.94	77.56	79.71	81.62	82.81	84	85.91	87.81
centroid	83.77	87.58	90.69	92.36	93.31	93.79	94.27	94.51	94.74	95.46
complex cord.	84.48	89.26	91.88	92.12	93.31	93.55	95.94	96.18	96.42	96.55

Finally we demonstrate the ability of our algorithm to deal with occlusion on a database comprised of synthetically occluded objects. The objects are selected from MPEG-7 Shape data set that we used in the previous experiments and the occluded version of an object is obtained by erasing the boundary at various rates (e.g. 1%-20%). The recognition performance for the synthetically generated objects is given in Table B.3. The proposed descriptor can tolerate occlusion at most 5% where the recognition rate is still greater than 80%. On the other hand, the other methods can tolerate occlusion at most 2%. It is known that boundary-based shape descriptors have limited performance under occlusion. When this is considered, it is a significant improvement. We also present the rank 10 query results in Fig. B.2, where the first image is the synthetically produced query image and the remaining images are the first 10 images retrieved from the database. In the experiments, the query objects have occlusion rates of 6% on the average.

These experiments demonstrate that the proposed descriptors have the best performances on these binary and grayscale databases. Even though shape matching approaches can achieve higher performance on the Kimia database as reported by [56], with the expense of high matching cost, a cost-efficient invariant descriptors, such as given here, with a high recognition rate will probably be more preferable for most of the content-based retrieval systems.

4.4 Conclusion

In this study, we present new affine invariant object shape descriptors, employing steerable gradient filters and Fourier Descriptors. The proposed system utilizes not only the boundary point coordinates of the objects, but also the filter responses along the boundaries. We compare the recognition performance of new shape descriptors with well-known boundary based shape descriptors on the databases including grayscale license plate characters and binary objects. The experimental results show that the proposed descriptors drastically outperforms other shape descriptors. Another advantage of the proposed descriptors is that one can use them on both grayscale and binary images whereas other shape descriptors can only be applied to binary objects. The proposed algorithm takes the advantage of local distribution information found in the gradient cue to achieve this high performance. Even though the performance of the proposed descriptors ($\tilde{F}$ and $\tilde{H}$) are comparable, the compactness and the computational efficiency of the descriptor $\tilde{H}$ make this descriptor very attractive for content-based image retrieval systems.

5. GOAL-ORIENTED EDGE DETECTION

In many vision applications, there is a great demand for an edge detector which can produce edge maps with very different characteristics in nature, so that one of these edge maps may meet the requirements of the problem under consideration. Unfortunately it is not evident how to choose the desired or the optimum edge maps from these solutions that the edge detectors offer. The proposed solutions are usually too general that cannot be easily adapted to the application needs by tuning edge detection parameters. One edge detector that we have studied in this dissertation is Generalized Edge Detector (GED) which is capable of producing edges with very different characteristics as we change the space parameters known as λ and τ . Although the edge maps based on this representation are reasonable, no one set of scale parameters alone yields a solution close to the desired edges. In this thesis, our aim is to find out powerful edge operators and use them under a goal-oriented edge detection framework.

Proposed framework is a two-stage process. First, user marks some pixels in the database as edge and non-edge pixels. Then feature vectors comprised of filter responses to G -Filters at different scales are extracted at these marked pixels. Edge detection problem is imposed as two-class classification problem. Support vector machine (SVM) is used as a classifier. Classifier itself is not adequate to extract desired edges for the application under consideration. In the second stage continuous edges are treated as one contour. Then contours are matched with the contours in the training set. Only matched contours are kept and the other contours are rejected.

The purpose of the first stage is to keep only prominent edges and remove irrelevant edges with respect to the applications requirements. The classifier decides which discontinuity is prominent or irrelevant. So how the training set is constructed is very important for the performance of the detector. Training set should contain almost equal number of edge and non-edge examples.

5.1 Stage I: Extracting Feature Vector

One of the key element in the framework is feature extraction. A user guides the algorithm by labeling the pixels as edge pixels and non-edge pixels. Since edges are discontinuity points, user should be careful in marking these edge pixels. Non-edge points usually belong to regions where intensity values are homogeneously distributed. When compared to An example is given in Fig. 5.1. Labeling is performed over noisy checkerboard image (Fig. 5.1b). Edge pixels are colored as black and non-edge pixels are colored as gray (Fig. 5.1c) and Fig. 5.1d). Non-edge points are labeled as blobs rather than labeling as pixel. Totally 502 edge pixels and 614 non-edge pixels are marked in Fig. 5.1c and 502 edge pixels and 1257 non-edge pixels are marked in Fig. 5.1d.

We first discretized $\lambda\tau$ - space by taking samples at regular intervals (λ_i, τ_i) as shown in Fig. 5.2 where $\lambda_i = \lambda_0 + i * \Delta\lambda$ and $\tau_j = \tau_0 + j * \Delta\tau, i = 0, 1, \dots, I$, and $j = 0, 1, \dots, J$. The corresponding edge detection filters are given by $G_x(\lambda_i, \tau_j)$ and $G_y(\lambda_i, \tau_j)$. These filters are applied to the training images denoted by $f_k(m, n)$ only at labeled edge pixels ($edge_{f_k}(r) : \mathbb{R}^2 \times \mathbb{N} \rightarrow \mathbb{N}^2$) and non-edge pixels ($none_{f_k}(r) : \mathbb{R}^2 \times \mathbb{N} \rightarrow \mathbb{N}^2$). Filter responses are then aggregated into two feature vectors denoted by u and v ; one for edge samples and one for non-edge samples. Formal definitions of the feature vectors are given as follows

$$\begin{aligned} u_{i,j,r}(f_k) &= (\| [G_x(\lambda_i, \tau_j) * f_k](edge_{f_k}(r)) \|), \\ v_{i,j,s}(f_k) &= (\| [G_y(\lambda_i, \tau_j) * f_k](none_{f_k}(s)) \|) \end{aligned} \quad (5.1)$$

In order to study how sampling the $\lambda\tau$ -space effects the feature vector distribution in the feature space, we set I to 2 and J to 1. In this case, the feature space is two dimensional where the feature vectors are reduced to the following form

$$\begin{aligned} \mathbf{u}_r(\mathbf{f}_k) &= (u_{1,1,r}, u_{2,1,r}), \\ \mathbf{v}_s(\mathbf{f}_k) &= (v_{1,1,s}, v_{2,1,s}). \end{aligned} \quad (5.2)$$

We have used six different sampling and two labeling schemes in our experiments for the two dimensional feature space. The corresponding λ and τ parameters used in the experiments and the feature space distributions are given in Fig. 5.3 and Fig. 5.4. Although noisy checkerboard image (Fig. 5.1b) is used in feature vector extraction, the vector distribution is very appropriate for classification for all sampling and marking schemes. Please note that the original checkerboard image (Fig. 5.1a) contains only step transitions and there are several topological structures in the intensity surface such as

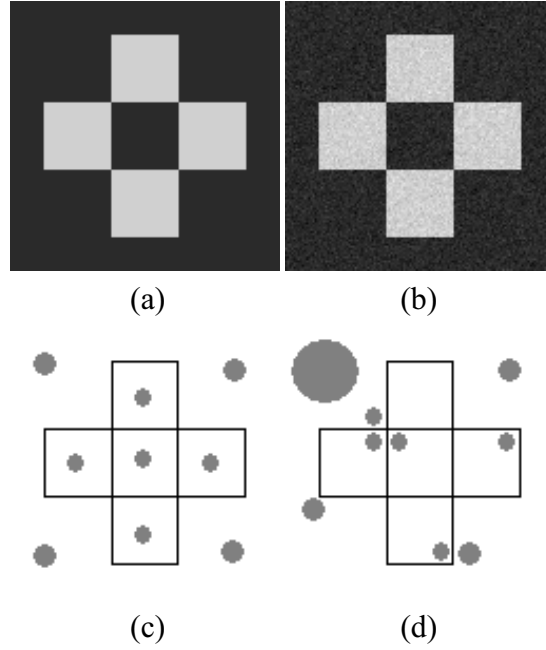


Figure 5.1: Labeling pixels as edge and non-edge pixels (a) Original checkerboard image, (b) Noisy checkerboard image (c) Marked pixels (black pixels are edge points, gray pixels are non-edge points) (d) Another marking used in training.

corners and crossings. Since the marking in the Fig. 5.1d contains non-edge labels near these topological structures, we may expect that the edge detector trained with this labels yields better results compared to the edge detector trained with the other one. Let us call the edge detector trained with the labels in Fig. 5.1c as *Detector-1* and the edge detector trained with the labels in Fig. 5.1d as *Detector-2*.

The noisy checkerboard image shown in Fig. 5.1b is used in training, whereas the noisy checkerboard image with smaller SNR value (18dB) shown in Fig. 5.1c is used in testing. Edge detection results are shown in Fig. Appendix C and Fig. C.4. Experimental results verify that marking process is the heart of the algorithm. Whereas *Detector-1* fails in detecting edges in the noisy checkerboard image with $SNR = 18dB$, *Detector-2* successfully captures the step transitions. On the other hand, *Detector-1* can detect edges in noisy bars images given in Fig. C.2 whereas *Detector-2* fails at detecting edges. The edge detection results for *Detector-1* and *Detector-2* are given in Fig. C.5 and Fig. C.6.

Another issue related to the proposed approach is the sampling scale space parameters (λ and τ). In two dimensional feature space case, detectors obtained with distant and large scale parameters such as $(\lambda_1 = 32.0, \tau_1 = 0.5)$, $(\lambda_2 = 64.0, \tau_1 = 0.5)$ and $(\lambda_1 = 8.0, \tau_1 = 0.5)$, $(\lambda_2 = 16.0, \tau_1 = 0.5)$ have poor detection performance.

We then set I to 3 and J to 1 and repeat the similar experiments with the case $I = 3, J = 1$. The feature space is three dimensional now and the feature vectors reduce to the following

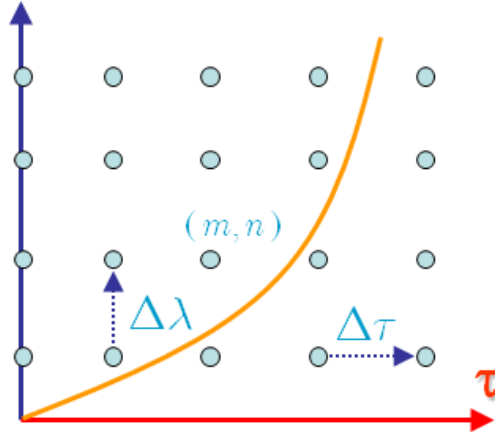


Figure 5.2: $\lambda\tau$ –space sampling scheme.

form

$$\begin{aligned} \mathbf{u}_r(\mathbf{f}_k) &= (u_{1,1,r}, u_{2,1,r}, u_{3,1,r}), \\ \mathbf{v}_s(\mathbf{f}_k) &= (v_{1,1,s}, v_{2,1,s}, v_{3,1,s}). \end{aligned} \quad (5.3)$$

Here we have used two different sampling and two labeling schemes in our experiments for the this case. The corresponding λ and τ parameters used in the experiments and the feature space distributions are given in Fig. C.7 and Fig. C.8. Edge detection results are shown in Fig. C.11. Three dimensional vector space performs better than the two dimensional space. Note that no post processing such as thinning or hysteresis thresholding is applied to the edge detection results. Let us call the edge detector trained with the labels in Fig. 5.1c as *Detector-3* and the edge detector trained with the labels in Fig. 5.1d as *Detector-4*.

Detector-3 can detect edges in noisy bars images given in Fig. C.2 whereas *Detector-4* fails at detecting edges. The edge detection results for *Detector-3* and *Detector-4* are given in Fig. C.9 and Fig. C.10.

5.2 Stage II: Contour Matching

Proposed framework has two steps. In the first, user marks some pixels in the database as edge and non-edge pixels. Then feature vectors comprised of filter responses to G –Filters at different scales are extracted at these labeled pixels. Edge detection problem is imposed as two-class classification problem. Classifier itself is not adequate to extract desired edges for the application under consideration. In the second step continuous edges are treated as one contour. Then contours are matched with the contours in the training set. Only matched contours are kept and the other contours are rejected. In contour matching we have used the algorithm developed in this study. The algorithm is explained in Chapter 4.

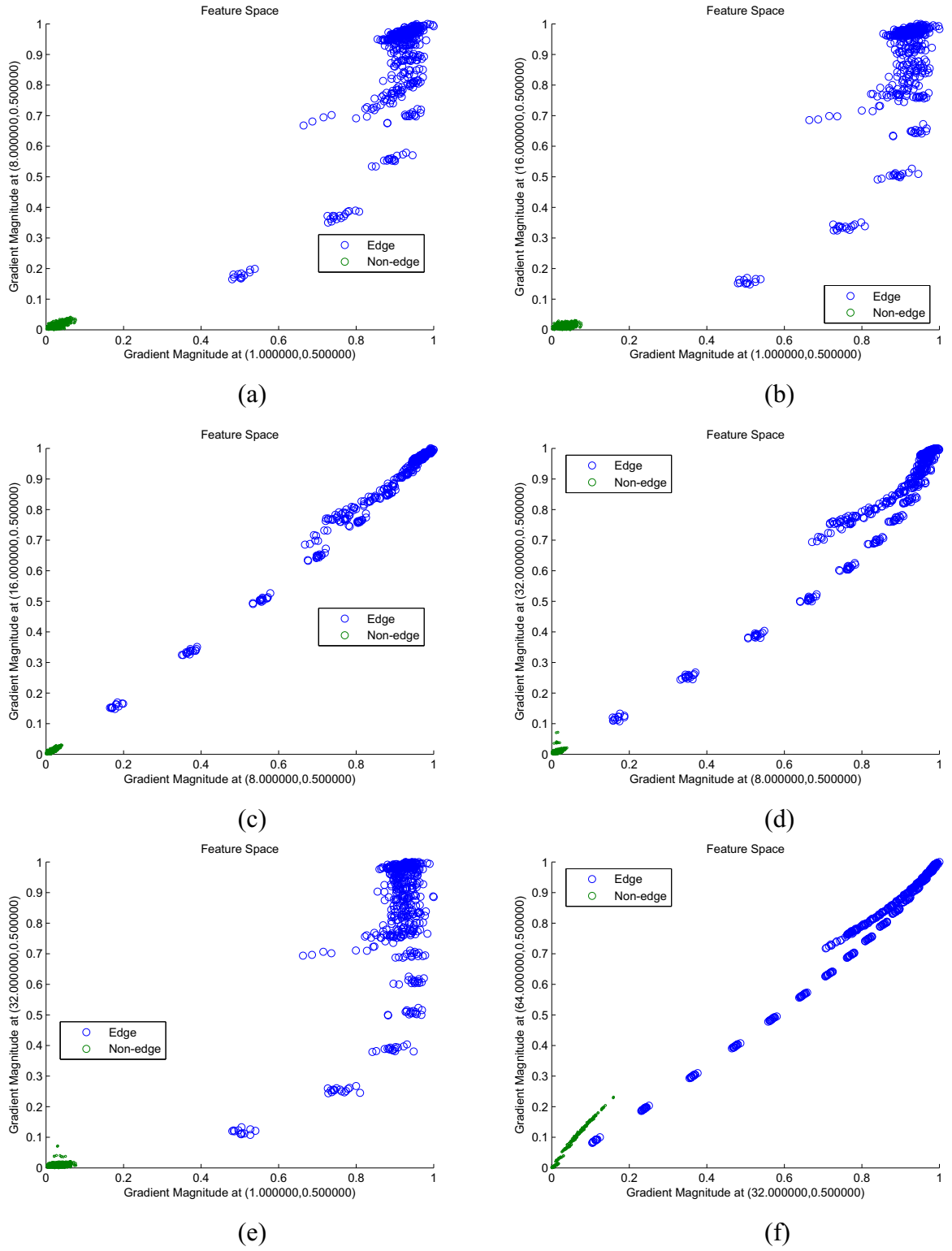


Figure 5.3: Feature vector scatter diagrams with respect to different $\lambda\tau$ -space sampling schemes when the labels given in Fig. 5.1(c) is used. (a) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 8.0, \tau_1 = 0.5)$, (b) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 16.0, \tau_1 = 0.5)$, (c) $(\lambda_1 = 8.0, \tau_1 = 0.5)$, $(\lambda_2 = 16.0, \tau_1 = 0.5)$, (d) $(\lambda_1 = 8.0, \tau_1 = 0.5)$, $(\lambda_2 = 32.0, \tau_1 = 0.5)$, (e) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 32.0, \tau_1 = 0.5)$, (f) $(\lambda_1 = 32.0, \tau_1 = 0.5)$, $(\lambda_2 = 64.0, \tau_1 = 0.5)$.

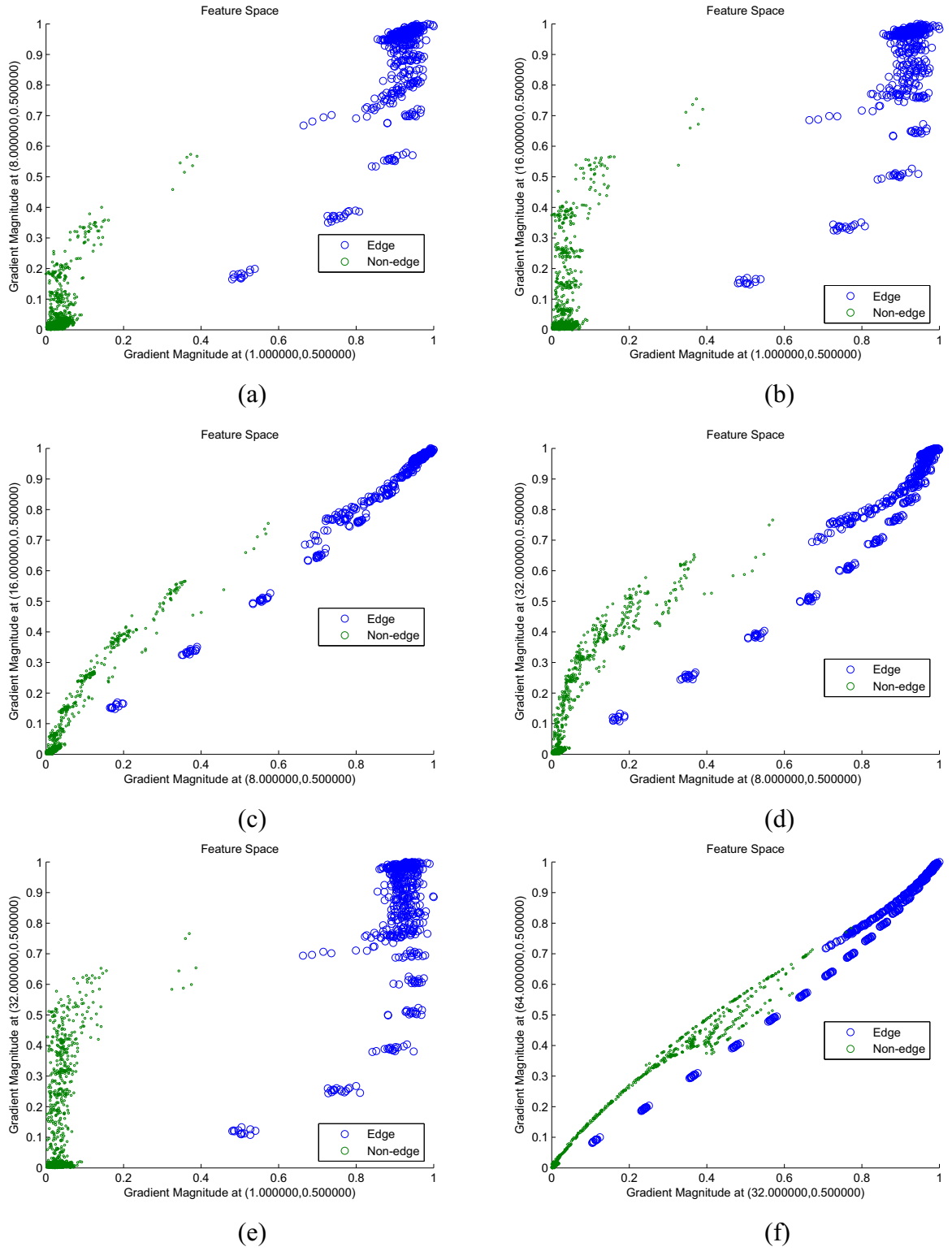


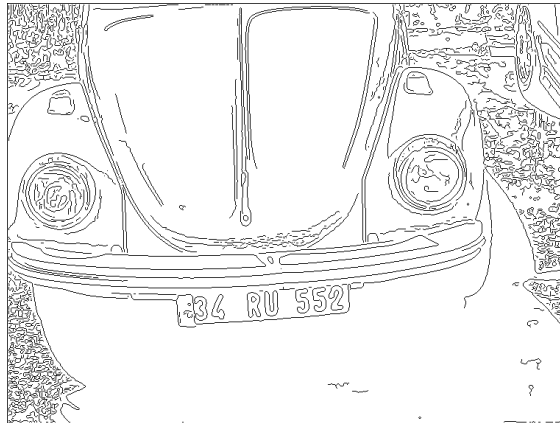
Figure 5.4: Feature vector scatter diagrams with respect to different $\lambda\tau$ -space sampling schemes when the labels given in Fig. 5.1(d) is used. (a) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 8.0, \tau_1 = 0.5)$, (b) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 16.0, \tau_1 = 0.5)$, (c) $(\lambda_1 = 8.0, \tau_1 = 0.5)$, $(\lambda_2 = 16.0, \tau_1 = 0.5)$, (d) $(\lambda_1 = 8.0, \tau_1 = 0.5)$, $(\lambda_2 = 32.0, \tau_1 = 0.5)$, (e) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 32.0, \tau_1 = 0.5)$, (f) $(\lambda_1 = 32.0, \tau_1 = 0.5)$, $(\lambda_2 = 64.0, \tau_1 = 0.5)$.

Contour matching stage will be explained using license plate recognition system. Applications such as traffic measurement and management, traffic surveillance, car park automation require successful license plate recognition (LPR) system. A typical LPR system is comprised of two parts: license plate detection (LPD) and license plate localization (LPL), each targets different aims. LPD aims at detecting the presence of plate in an image, while LPL aims at extracting the actual boundary and segmenting the plate characters which are ready for the optical character recognition.

Fig. [D.1](#) shows typical segmented license plate characters selected from the database which contains 8321 grayscale digit characters segmented from the real vehicle license plates. Fig. [D.2](#) shows the boundaries of the segmented license plate digits. Edge detector explained in the previous section is trained with the digit characters and their boundaries in the database. Fig. [5.2a](#) contains a vehicle image. After the first stage the edges given in Fig. [5.2b](#) are detected. Contour matching is applied to the detected contour. Since non-digit contours are not matched, they are eliminated. After the contour elimination, the contours given in Fig. [5.2c](#) are obtained.



(a)



(b)

34 552

(c)

Figure 5.5: Edge detection results (a) Vehicle image (b) Detected edges (c) Detected contours.

6. SUMMARY AND CONCLUSION

This dissertation presents a new approach to edge detection. We presented a new edge detection formulation within a goal-based, regularized, shape-aware framework. This framework provides mechanisms for achieving detection without requiring any thresholding or parameter tuning, also taking the contour geometry into account. We proposed a two-stage framework. In the first stage, a user marks some pixels in the database as edge and non-edge pixels as the underlying application requires. Then feature vectors comprised of filter responses to G-Filters at different scales are extracted at these marked pixels. Edge detection problem is imposed to two-class classification problem. Classifier itself is not adequate to extract desired edges for the application under consideration. In the second stage continuous edges are treated as one contour. Then contours are matched with the contours in the training set. Only matched contours are kept and the other contours are eliminated.

The purpose of the first stage is to keep only prominent edges and remove irrelevant edges with respect to the application. The classifier decides which discontinuity is prominent or irrelevant. So how the training set is constructed is very important for the performance of the detector. Training set should contain almost equal number of edge and non-edge examples.

In the thesis, we also extended powerful smoothing filter and edge filters (R - and G -filters) known as $\lambda\tau$ -space representation to two dimensional space. G -filter is called Generalized Edge Detector which is capable of producing most of the existing edge detectors. Two dimensional extension of the representation is important because of that the behaviour of images and edges in the space is best modeled by two dimensional smoothing and edge detector filters. Also since GED filters encompass most of the well-known edge detectors, two dimensional version of these filters could be obtained. We have experimentally showed that the derived filters are more robust to noise when compared to the previous one dimensional filtering scheme in the sense of FOM (Figure Of Merit) and false alarm characteristics. We speed up the filtering by implementing recursive approach. Unfortunately computation time in serial implementation depends on scale parameters (λ and τ). Since filter size increases with increasing λ , convolution requires more multiplication per pixel. We have implemented recursive filtering approach

to convolution with two dimensional R - and G -Filters. Filtering is performed in constant time now with this recursive implementation. In this study, we also interpreted the recursive implementation and revealed its relation with the forth-order isotropic diffusion equation.

Although $\lambda\tau$ -space presents richer representation, it suffers from the same problem. Space parameters are constant throughout the image space causing the edges to be blurred while noise is removed elsewhere. This problem is due to the linear nature of the representation. To overcome this problem, we present an anisotropic $\lambda\tau$ -space image and edge representation allowing the smoothing parameter λ to vary through the image respecting the feature directions. We have showed experimentally that our proposed scheme is capable of locating the feature direction robustly when the both terms in the hybrid diffusion are used as contrary to the Perona-Malik equation. Using this approach one may obtain surfaces having different characteristics for the choice of different τ .

The second contribution of this dissertation is two gradient-based affine-invariant shape descriptors denoted as $\tilde{F}$ and $\tilde{H}$ which could be applied to both binary and grayscale images. We use two-dimensional steerable G -Filters to obtain gradient information at different orientations and scales, and then aggregate the gradients into a shape signature. The signature derived from the rotated object is circularly shifted version of the signature derived from the original object. This property is called the circular-shifting rule. The shape descriptor is defined as the Fourier transform of the signature. We also provided a distance measure for the proposed descriptor by taking the circular-shifting rule into account. We compared the recognition performance of new shape descriptors with well-known boundary based shape descriptors on the databases including grayscale license plate characters and binary objects. The experimental results show that the proposed descriptors drastically outperforms other shape descriptors. Another advantage of the proposed descriptors is that one can use them on both grayscale and binary images whereas other shape descriptors can only be applied to binary objects. The proposed algorithm takes the advantage of local distribution information found in the gradient cue to achieve this high performance. Even though the performance of the proposed descriptors ($\tilde{F}$ and $\tilde{H}$) are comparable, the compactness and the computational efficiency of the descriptor $\tilde{H}$ make this descriptor very attractive for content-based image retrieval systems.

Overall, this dissertation proposes a new, effective and efficient framework for edge detection. The use of goal-oriented approach will find its applications in robust, accurate, efficient edge and contour extraction. We believe that the approach developed here leads to powerful edge/contour-based systems especially for applications involving face recognition and license plate detection.

BIBLIOGRAPHY

- [1] **Gökmen, M. and Jain, A.K.**, 1997. $\lambda\tau$ -Space Representation of Images and Generalized Edge Detector, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, 545-563.
- [2] **Günsel, B. and Tekalp, M.**, 1998. Shape Similarity Matching for Query-By-Example, *Pattern Recognition*, **31**, 931-944.
- [3] **Nasrabadi, N.M.**, 1992. A Stereo-Vision Technique using Curve Segments and Relaxation Matching, *IEEE Transactions On Pattern Analysis and Machine Intelligence*, **14**, 566-572.
- [4] **Kurt, B. and Gökmen, M.**, 2005. Image Compression Based On Centipede Model, *ARI, The Bulletin of the Istanbul Technical University*, **54**, 35-43.
- [5] **Olson, C.F. and Huttenlocher, D.P.**, 1997. Automatic Target Recognition by Matching Oriented Edge Pixels, *IEEE Transactions on Image Processing*, **6**, 103-113.
- [6] **Yılmaz, A. and Gökmen, M.**, 1999. Face Recognition Based on Eigenhills, *7th Conference on Signal Processing and Applications*, SIU'99.
- [7] **Marr, D. and Hildreth, E.**, 1980. Theory of Edge Detection, *Proceedings of Royal Society, Sec B*, **207**, 187-217.
- [8] **Canny, J.F.**, 1986. A Computational Approach to Edge Detection, *IEEE Transactions On Pattern Analysis and Machine Intelligence*, **8**, 679-698.
- [9] **Deriche, R.**, 1987. Optimal Edge Detection using Recursive Filtering, *Proceedings of IEEE First International Conference on Computer Vision*, 501-505.
- [10] **Shen, J. and Castan, S.**, 1986. An Optimal Linear Operator for Edge Detection, *Proceedings of IEEE First Conference on Pattern Recognition*, 109-114.
- [11] **Witkin, A.P.**, 1983. Scale-space Filtering, *Proceedings of 8th Int. Joint Conf. on Artificial Intelligence*, 1019-1022.
- [12] **Gökmen, M. and Li, C.**, 1992. Multiscale Edge Detection using first-order R-filter, *Proceedings of International Conference on Pattern Recognition*, 307-310.
- [13] **Poggio, T., Voorhees, H. and Yuille, A.**, 1985. A Regularized Solution to Edge Detection, *Technical Report, MIT AI Lab., AI Memo 833*.
- [14] **Acharya, T. and Ray A.K.**, 2005. Image Processing: Principles and Applications, John Wiley.
- [15] **Petrou, M. and Kittler, J.**, 1991. Optimal edge detector for ramp edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13 (5)**.
- [16] **Koenderink, J.**, 1984. The structure of images. *Biological Cybernetics*, **50**.
- [17] **Hummel, A.**, 1987. Representations based on zero-crossings in scale-space. *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, 204-209.

- [18] **Perona, P. and Malik, J.**, Scale-space and Edge Detection using Anisotropic Diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 629-639.
- [19] **Hueckel, M.**, 1973. A local visual operator which recognizes edges and lines. *Journal of the ACM*, **20**, 634-647.
- [20] **Gorman, O.**, 1978. Edge detection using walsh functions. *Artificial Intelligence*, **10**, 215-233.
- [21] **Hartley, R.**, 1985. A gaussian-weighted multiresolution edge detector. *Computer Vision, Graphics, and Image Processing*, **30(1)**.
- [22] **Nalwa, V.S. and Binford, T.O.**, 1986. On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8(6)**, 699-714.
- [23] **Lyvers, E.P., Michell, O.R., Akey, M.L. and Reeves, A.P.**, 1986. Subpixel measurement using a moment-based edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11 (12)**, 1293-1309.
- [24] **Rohr, K.**, 1992. Recognizing corners by fitting parametric models. *International Journal of Computer Vision*, **9 (3)**, 213-230.
- [25] **Parida, L., Geiger, D. and Hummel, R.**, 1998. Junctions: Detection, Classification, and Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20 (7)**, 687-698.
- [26] **Kass, M., Witkin, A. and Terzopoulos, D.**, 1988. Snakes, active contour models. *International Journal of Computer Vision*.
- [27] **Pentland, A.P. and Horowitz, B.**, 1991. Recovery of nonrigid motion and structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13(7)**.
- [28] **Pentland, A.P. and Williams, J.R.**, 1989. Good vibrations: Modal dynamics for graphics and animation, *Proceedings of ACM SIGGRAPH Conference*.
- [29] **Mumford, M. and Shah, J.**, 1985. Boundary detection by minimizing functionals. *IEEE Conf. Computer Vision and Pattern Recognition*, 22-26.
- [30] **Ambrosio, L. and Tortorelli, V.M.**, 1990. Approximation of functionals depending on jumps by elliptic functional via γ -convergence. *Comm. Pure and Applied Math*, **43 (8)**.
- [31] **Bertero, M., Poggio, T. and Torre, T.**, 1997. Ill-posed Problems in Early Vision, *Technical Report*, MIT AI Lab., AI Memo 924.
- [32] **Tikhonov, A.N. and Arsenin, V.Y.**, 1977. Solution of Ill-posed Problems. Washington D.C.: Winstons and Sons.
- [33] **Gökmen, M.**, 1990. Edge Detection using Regularization Theory, *PhD Thesis*, Dept. Of Elec. Electrical Engineering, Univ. of Pittsburgh.
- [34] **Blake, A. and Zisserman, A.**, 1987. Visual Reconstruction, MIT Press.
- [35] **Kurt, B. and Gökmen, M.**, 1999. Two Dimensional Generalized Edge Detector, *International Conference On Image Analysis and Processing*, ICIAP'99.
- [36] **Torkomani-Azar, F. and Tait, K.E.**, 1997. Image Recovery Using the Anisotropic Diffusion Equation, *IEEE Transactions On Image Processing*, **11 (2)**, 1573-1577.

- [37] **Carmona, R.A. and Zhong, S.**, 1998. Adaptive Smoothing Respecting Feature Directions, *IEEE Transactions On Image Processing*, **7 (3)**, 353-358.
- [38] **F. Catte, P.-L. Lions, J.-M. Morel, and T. Coll**, 1992. Image Selective Smoothing and Edge Detection by Nonlinear Diffusion, *SIAM J. Numerical Analysis*, **29**, 182-193.
- [39] **Faugeras, O. and Keriven, R.**, 1998. Variational Principles, Surface Evaluation, PDE's and the Stereo Problem, *IEEE Transactions On Image Processing*, **7(3)**, 336-344.
- [40] **Moisan, L.**, 1998. Affine Plane Curve Evolution: A Fully Consistent Scheme, *IEEE Transactions On Image Processing*, **7 (3)**, 411-420.
- [41] **Li, X. and Chen, T.**, 1994. Nonlinear Diffusion with Multiple Edginess Thresholds, *Pattern Recognition*, **27 (8)**, 1029-1037.
- [42] **Gökmen, M. and Li, C.**, 1992. Multiscale Edge Detection using first-order R-filter, *Proceedings of International Conference on Pattern Recognition*, 307-310.
- [43] **Bowyer, K.W., Kranenburg, C. and Dougherty, S.**, 2001. Edge detector evaluation using empirical ROC curves, *Computer Vision and Image Understanding*, **84 (1)**, 77-103.
- [44] **Torkomani-Azar, F. and Tait, K.E.**, 1997. Image Recovery Using the Anisotropic Diffusion Equation, *IEEE Transactions On Image Processing*, **11 (2)**, 1573-1577.
- [45] **Carmona, R.A. and Zhong, S.**, 1998. Adaptive Smoothing Respecting Feature Directions, *IEEE Transactions On Image Processing*, **7 (3)**, 353-358.
- [46] **Perona, P. and Malik, J.**, 1990. Scale-space and Edge Detection using Anisotropic Diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 629-639.
- [47] **Catte, F., Lions, P.L., Morel, J.M. and Coll, J.M.**, 1992. Image Selective Smoothing and Edge Detection by Nonlinear Diffusion," *SIAM Journal on Numerical Analysis*, **29**, 182-193.
- [48] **Faugeras, O. and Keriven, R.**, 1998. Variational Principles, Surface Evaluation, PDE's and the Stereo Problem, *IEEE Transactions On Image Processing*, **7 (3)**, 336-344.
- [49] **Shah, J.**, 1996. A Common Framework for Curve Evaluation, Segmentation and Anisotropic Diffusion, *Proceedings of Conference on Computer Vision and Pattern Recognition*, 136-142.
- [50] **Moisan, L.**, 1998. Affine Plane Curve Evolution: A Fully Consistent Scheme, *IEEE Transactions On Image Processing*, **7(3)**, 411-420.
- [51] **Li, X. and Chen, J.**, 1994. Nonlinear Diffusion with Multiple Edginess Thresholds, *Pattern Recognition*, **27(8)**, 1029-1037.
- [52] **Costa, L. F. and Cesar, R.M.**, 2001. Shape Analysis and Classification: Theory And Practice, CRC Press New York.
- [53] **Veltkamp, R. and Hagedoorn, M.**, 1999. State-of-the-art in Shape Matching, *Technical Report*, UU-CS-1999.
- [54] **Zhang, D. and Lu, G.**, 2004. Review of shape representation and description techniques, *Pattern Recognition*, **37**, 1-19.

- [55] **Belongie, S., Malik J. and Puzicha, J.**, 2002. Shape Matching and Object Recognition Using Shape Contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24 (24)**, 509-522.
- [56] **Zhuowen, T. and Yuille, A.**, 2004. Shape Matching and Recognition–Using Generative Models and Informative Features, *8th European Conference on Computer Vision*.
- [57] **Yokono, J.J. and Poggio, T.**, 2004. Oriented Filters for Object Recognition: an Empirical Study, *Automatic Face and Gesture Recognition*, 755-760.
- [58] **Freeman, W.T. and Adelson, E.H.**, 1991. The Design and Use of Steerable Filters, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13 (9)**, 891-906.
- [59] **Balard, D.H. and Wixson, L.E.**, 1993. Object recognition using steerable filters at multiple scales, Qualitative Vision, *Proceedings of IEEE Workshop*, 2-10.
- [60] **Talleux, S., Tavşanoğlu, V. and Tufan, E.**, 1998. Handwritten Character Recognition using Steerable Filters and Neural Networks, *IEEE Proceedings of International Symposium on Circuits and Systems (ISCAS-98)*, 341-344.
- [61] **Li, S. and Shawe-Taylor, J.**, 2005. Comparison and fusion of multiresolution features for texture classification, *Pattern Recognition Letters*, **26 (5)**, 633-638.
- [62] **Zhang, D. and Lu, G.**, 2003. A comparative study of curvature scale space and Fourier descriptors for shape-based image retrieval, *Journal of Visual Communication and Image Representation*, **14 (1)**, 39-57.
- [63] **Rafiei, D. and Mendelzon, A.O.**, 2002. Efficient retrieval of similar shapes, *The VLDB Journal*, **11**, 17-27.
- [64] **Phokharatkul, P. and Kimpan, C.**, 2003. Handwritten Thai Character Recognition Using Fourier Descriptors and Genetic Neural Network, *Computational Intelligence*, **18 (3)**, 270-293.
- [65] **Zhang, D. and Lu, G.**, 2005. Study and evaluation of different Fourier methods for image retrieval, *Image and Vision Computing*, **23 (1)**, 33-49.
- [66] **Canny, J.F.**, 1986. A computational approach to edge detection, *IEEE Transactions to Pattern Analysis and Machine Intelligence*, **8 (6)**, 112-131.
- [67] **Çapar, A. and Gökmen, M.**, 2006. Concurrent Segmentation and Recognition with Shape-Driven Fast Marching Methods," International Conference on Pattern Recognition (ICPR'06), 155-158.
- [68] **Gonzalez, R.C. and Woods, R.E.**, 2002. Digital Image Processing, Second Edition, Prentice Hall.
- [69] **Sharvit, D., Chan, J., Tek, H. and Kimia, B.**, 1998. Symmetry-Based Indexing of Image Database, *Journal on Visual Communication and Image Representation*, **9 (4)**, 366-380.

Appendix A . PROOF OF CIRCULAR-SHIFTING PROPERTY IN FOURIER DOMAIN

Assume that the descriptor obtained for an object is denoted as $f_{k,m}$ and the descriptor obtained for the α degrees rotated form is denoted as $f'_{k,m}$. By the definition of circular-shifting property the descriptors are related to each other as

$$f'_{k,(m+s) \bmod M} = f_{k,m} \quad (\text{A.1})$$

assuming that the rotation angle is exact multiple of $\frac{\pi}{M}$, namely $\alpha = s \frac{\pi}{M}$. Taking the Fourier transform of the descriptors ($f_{k,m}$ and $f'_{k,m}$) yields

$$\tilde{f}_{k,m} = \frac{1}{N} \sum_{n=0}^{N-1} f_{n,m} \exp\left(\frac{-j2\pi kn}{N}\right) \quad (\text{A.2})$$

$$\tilde{f}'_{k,m} = \frac{1}{N} \sum_{n=0}^{N-1} f'_{n,m} \exp\left(\frac{-j2\pi kn}{N}\right). \quad (\text{A.3})$$

Substituting (A.1) into (A.2) results in

$$\tilde{f}_{k,m} = \frac{1}{N} \sum_{n=0}^{N-1} f'_{n,(m+s) \bmod M} \exp\left(\frac{-j2\pi kn}{N}\right)$$

$$\tilde{f}_{k,m} = \tilde{f}'_{k,(m+s) \bmod M}$$

which means that circular-shifting applies also to $\tilde{f}_{k,m}$ as well as to $f_{k,m}$ $\square$

Appendix B. RECOGNITION RESULTS FOR GRADIENT BASED DESCRIPTORS

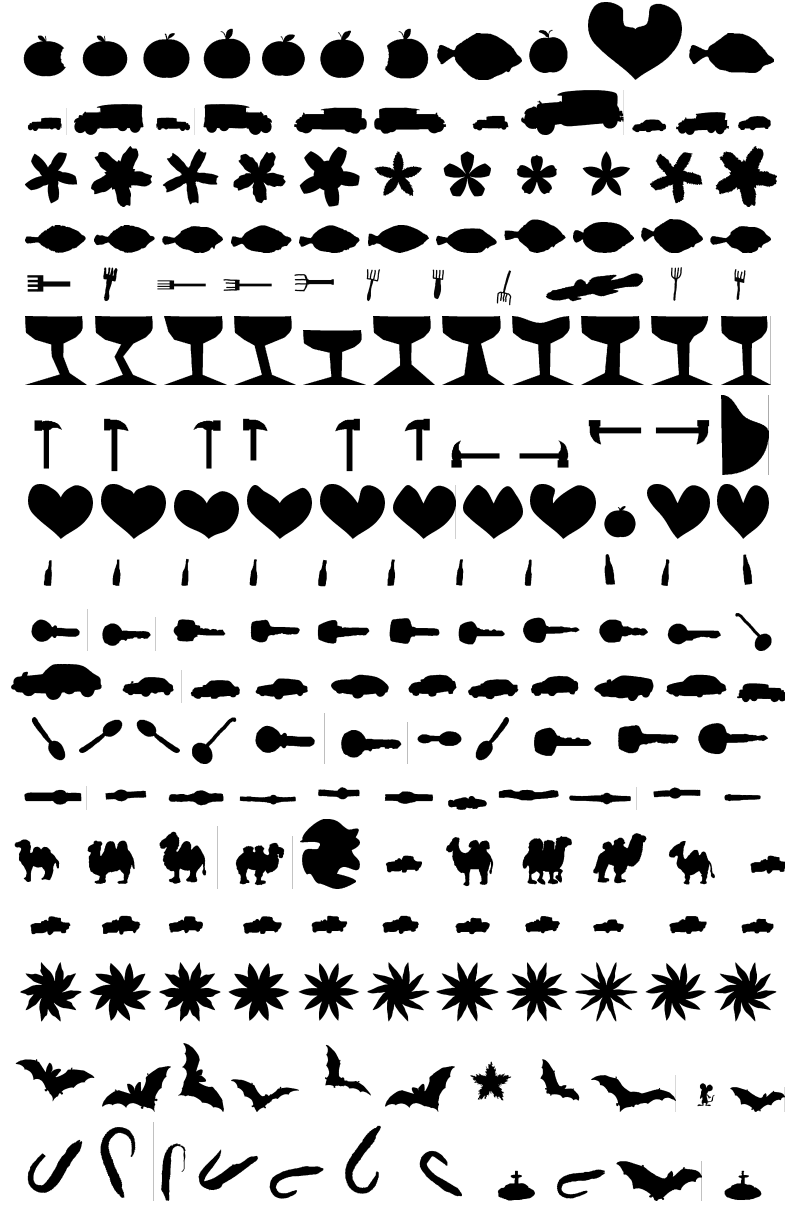


Figure B.1: Content-based image retrieval performance of the proposed method. Each line contains a query where the first image is the query image and the remaining images are the query results.

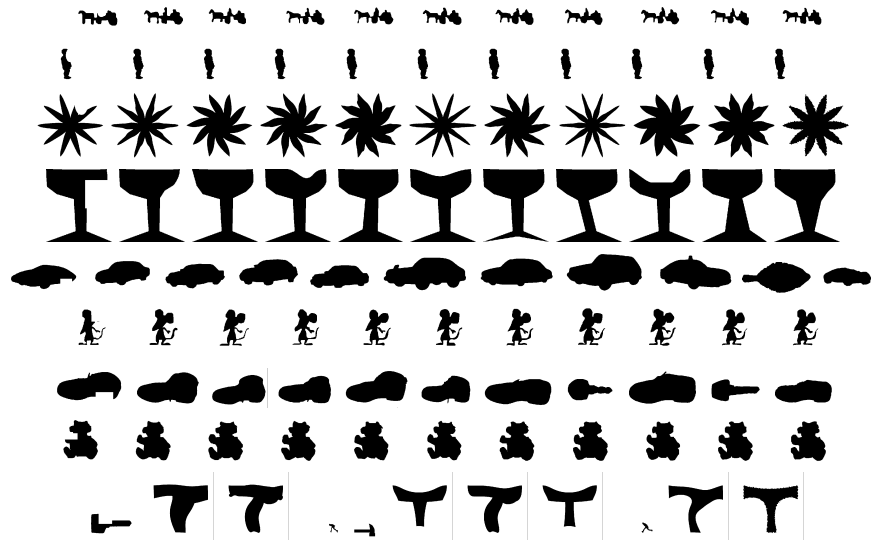


Figure B.2: CBIR performance of the proposed method under occlusion. Each line contains a query where the first image is the occluded query image and the remaining images are the query results.

Table B.1: The effect of training size on recognition rates for the MPEG-7 shape data set.

Training size	Rank									
	1	2	3	4	5	6	7	8	9	10
1	74.15	80.92	84.81	87.57	89.83	91.96	93.47	94.1	94.1	94.47
3	80.36	86.53	90.32	92	92.84	94.24	94.95	95.09	95.51	95.93
5	85.53	91.41	93.64	94.75	95.86	96.5	96.82	96.82	96.97	96.97
7	88.44	92.66	94.31	95.96	96.33	96.88	97.43	97.61	97.61	97.61
9	90.02	95.01	96.52	97.18	98.04	98.04	98.04	98.48	98.69	98.69

Table B.2: Recognition performance for the Kimia Data Set with respect to rank and descriptor size. Results are given in terms of the number of correctly retrieved shapes.

Method	Rank									
	1	2	3	4	5	6	7	8	9	10
$\tilde{F}(M = 2)$	70	79	83	84	86	91	91	92	94	95
$\tilde{F}(M = 3)$	78	84	87	90	92	92	92	92	93	93
$\tilde{F}(M = 4)$	82	89	92	94	97	98	98	98	98	98
$\tilde{F}(M = 5)$	86	90	93	96	98	98	99	99	99	99
$\tilde{F}(M = 6)$	84	94	95	97	98	98	98	98	98	98
$\tilde{F}(M = 7)$	89	95	97	97	99	99	99	99	99	99
$\tilde{F}(M = 8)$	87	93	94	95	97	97	97	98	98	99
$\tilde{F}(M = 9)$	89	93	93	94	96	98	98	98	98	99
$\tilde{F}(M = 10)$	89	94	94	96	98	98	98	99	99	99
$\tilde{F}(M = 11)$	89	94	94	94	96	97	98	99	99	99
$\tilde{F}(M = 12)$	88	93	96	96	98	98	98	99	99	99
$\tilde{F}(M = 13)$	88	94	95	98	98	98	98	99	99	99
$\tilde{F}(M = 14)$	88	92	94	97	97	97	98	99	99	99
$\tilde{F}(M = 15)$	90	93	95	96	98	98	98	99	99	99
$\tilde{F}(M = 16)$	87	91	95	97	98	98	99	99	99	99
$\tilde{H}(M = 2)$	90	93	95	96	97	97	97	97	97	97
Centroid	84	92	94	94	95	96	96	97	97	98
Curvature	52	62	65	70	75	77	77	79	82	87
Comp. Coord.	89	90	93	93	93	93	93	94	96	98

Table B.3: Recognition rates for the occluded objects that are synthetically produced from the MPEG-7 Core Experiments shape data set at various occlusion rates.

Method	Occlusion rate (%)							
	0	1	2	5	8	10	15	20
$\tilde{F}(M = 7)$	90.93	90.69	91.4	82.34	62.52	55.88	34.41	20.76
$\tilde{F}(M = 8)$	89.73	89.74	89.49	82.33	59.9	51.79	32.69	19.57
$\tilde{H}(M = 2)$	89.97	90.69	90.69	79.71	63.48	51.07	22.43	11.93
curvature	67.78	67.54	63	48.44	31.74	26.97	13.84	6.66
centroid distance	83.77	83.53	82.33	68.97	53.69	43.67	21.47	13.84
complex cord.	84.48	84.48	83.77	73.5	63.48	57.27	33.17	20.28

Appendix C. GOAL-ORIENTED EDGE DETECTION EXPERIMENTAL RESULTS

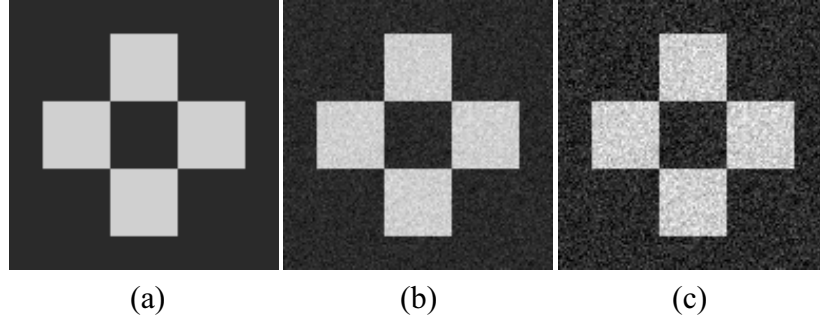


Figure C.1: Checkerboard images (a) Original checkerboard image, (b) Noisy checkerboard image (25dB) used in training (c) Noisy checkerboard image (18dB) used in testing.

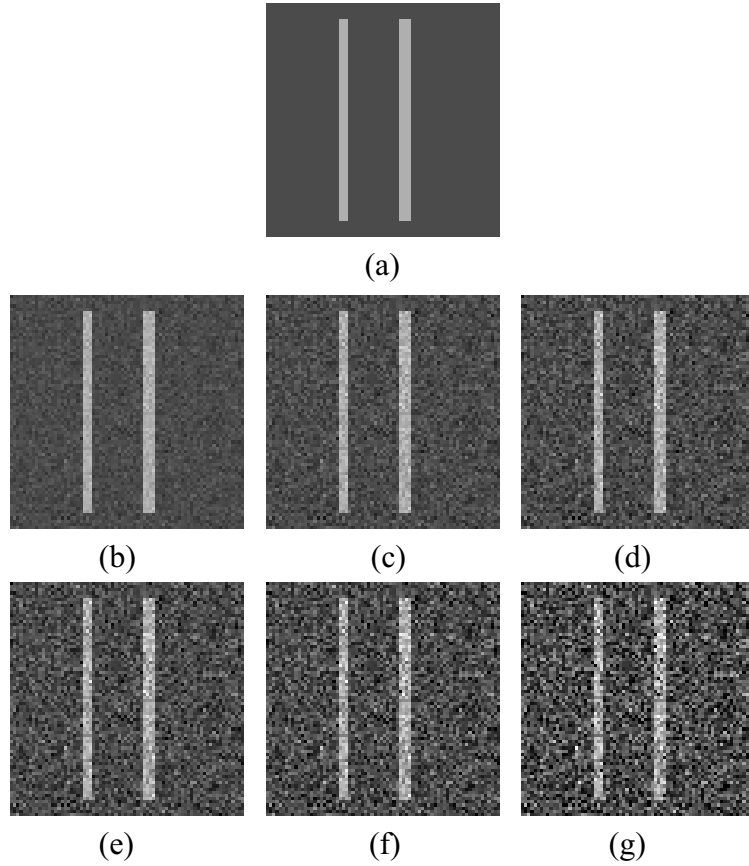


Figure C.2: Bars images (a) Original Bars image, Noisy bars images (b) 25dB, (c) 22dB, (d) 20dB, (e) 17dB, (f) 14dB, (g) 11dB.

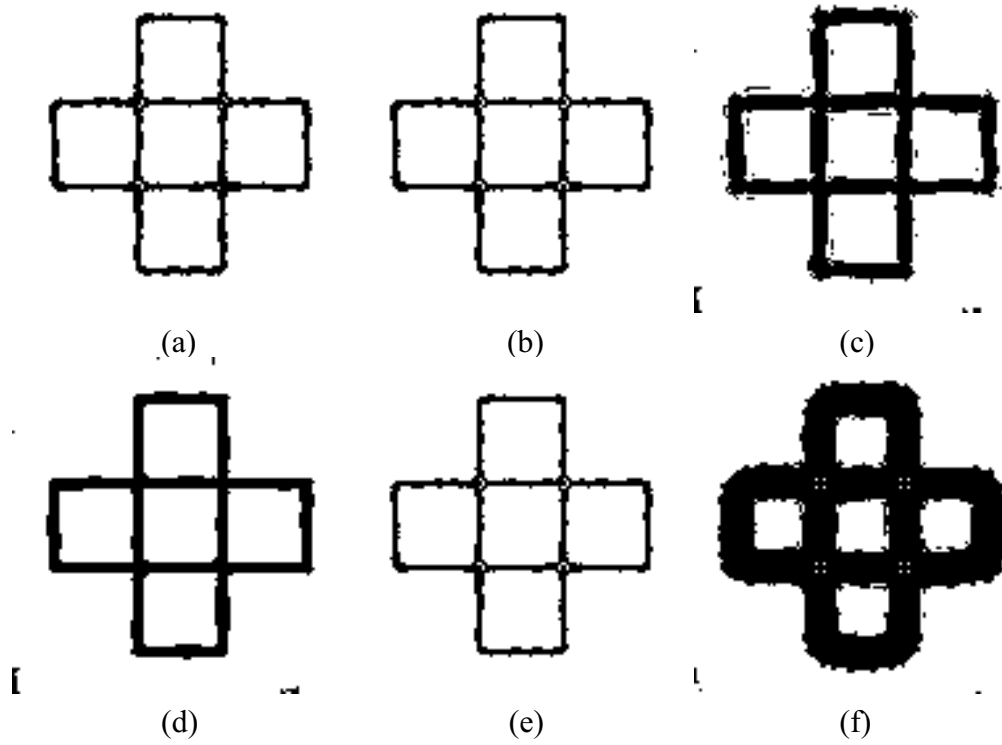


Figure C.3: Edge detection test results for the noisy checkerboard image (18dB). The labels given in Fig. 5.1(c) is used. The following $\lambda\tau$ -space sampling schemes are used: (a) $(\lambda_1 = 1.0, \tau_1 = 0.5), (\lambda_2 = 8.0, \tau_1 = 0.5)$, (b) $(\lambda_1 = 1.0, \tau_1 = 0.5), (\lambda_2 = 16.0, \tau_1 = 0.5)$, (c) $(\lambda_1 = 8.0, \tau_1 = 0.5), (\lambda_2 = 16.0, \tau_1 = 0.5)$, (d) $(\lambda_1 = 8.0, \tau_1 = 0.5), (\lambda_2 = 32.0, \tau_1 = 0.5)$, (e) $(\lambda_1 = 1.0, \tau_1 = 0.5), (\lambda_2 = 32.0, \tau_1 = 0.5)$, (f) $(\lambda_1 = 32.0, \tau_1 = 0.5), (\lambda_2 = 64.0, \tau_1 = 0.5)$.

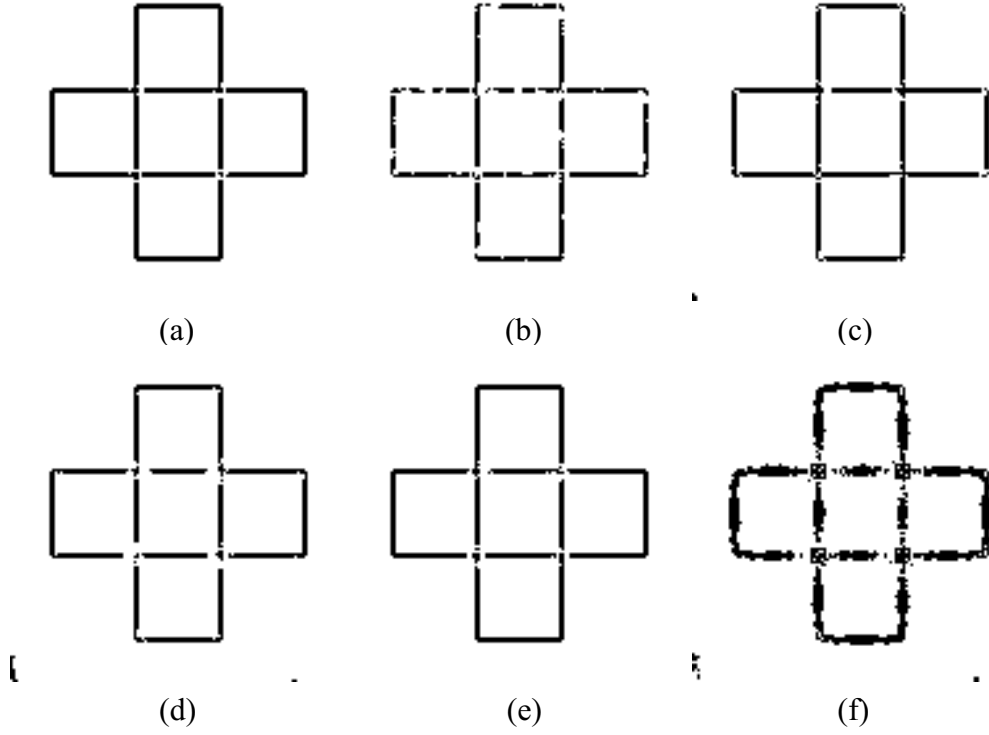


Figure C.4: Edge detection test results for the noisy checkerboard image (18dB). The labels given in Fig. 5.1(d) is used. The following $\lambda\tau$ -space sampling schemes are used: (a) $(\lambda_1 = 1.0, \tau_1 = 0.5), (\lambda_2 = 8.0, \tau_1 = 0.5)$, (b) $(\lambda_1 = 1.0, \tau_1 = 0.5), (\lambda_2 = 16.0, \tau_1 = 0.5)$, (c) $(\lambda_1 = 8.0, \tau_1 = 0.5), (\lambda_2 = 16.0, \tau_1 = 0.5)$, (d) $(\lambda_1 = 8.0, \tau_1 = 0.5), (\lambda_2 = 32.0, \tau_1 = 0.5)$, (e) $(\lambda_1 = 1.0, \tau_1 = 0.5), (\lambda_2 = 32.0, \tau_1 = 0.5)$, (f) $(\lambda_1 = 32.0, \tau_1 = 0.5), (\lambda_2 = 64.0, \tau_1 = 0.5)$.

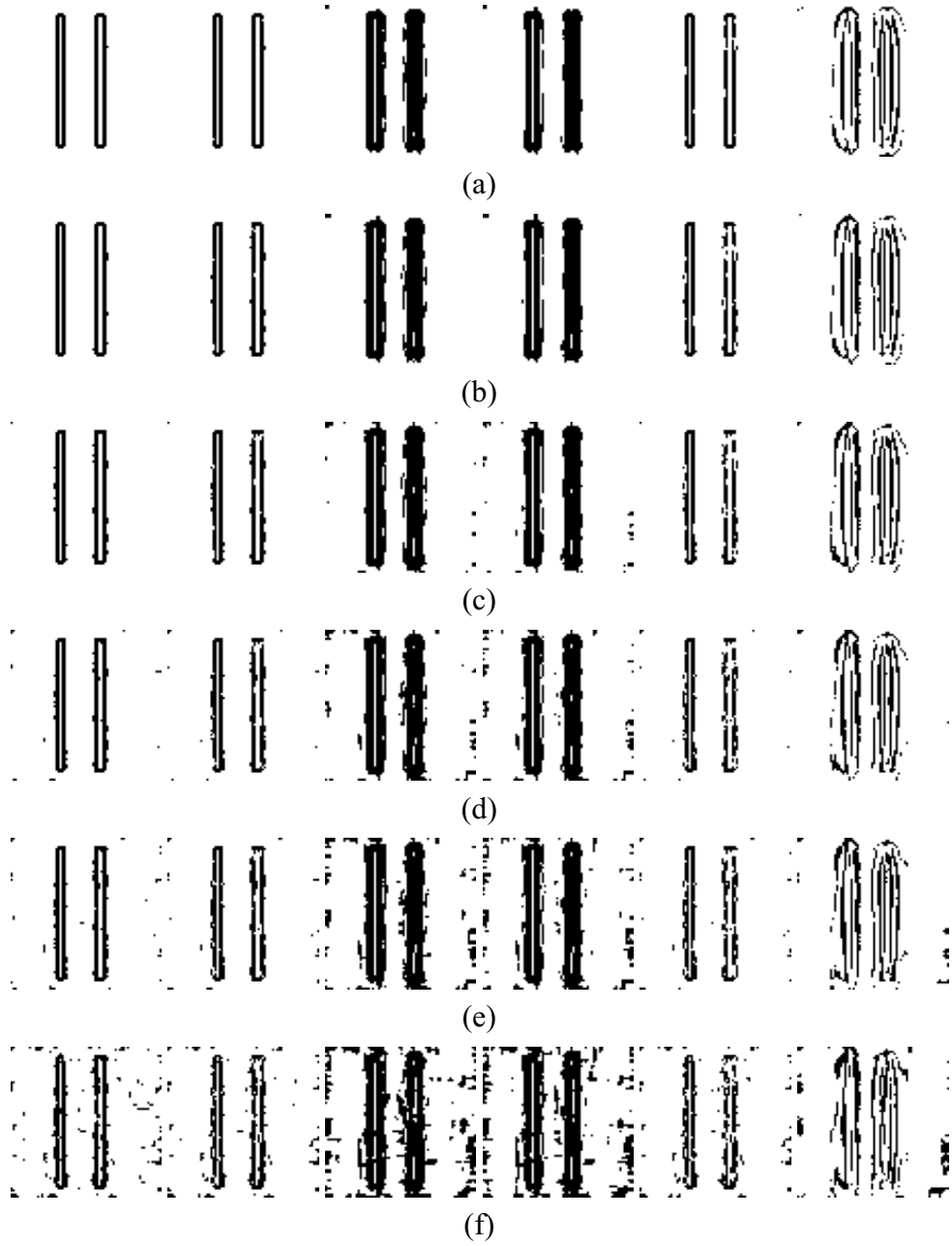


Figure C.5: Edge detection test results for the noisy bars images. The labels given in Fig. 5.1(c) is used.

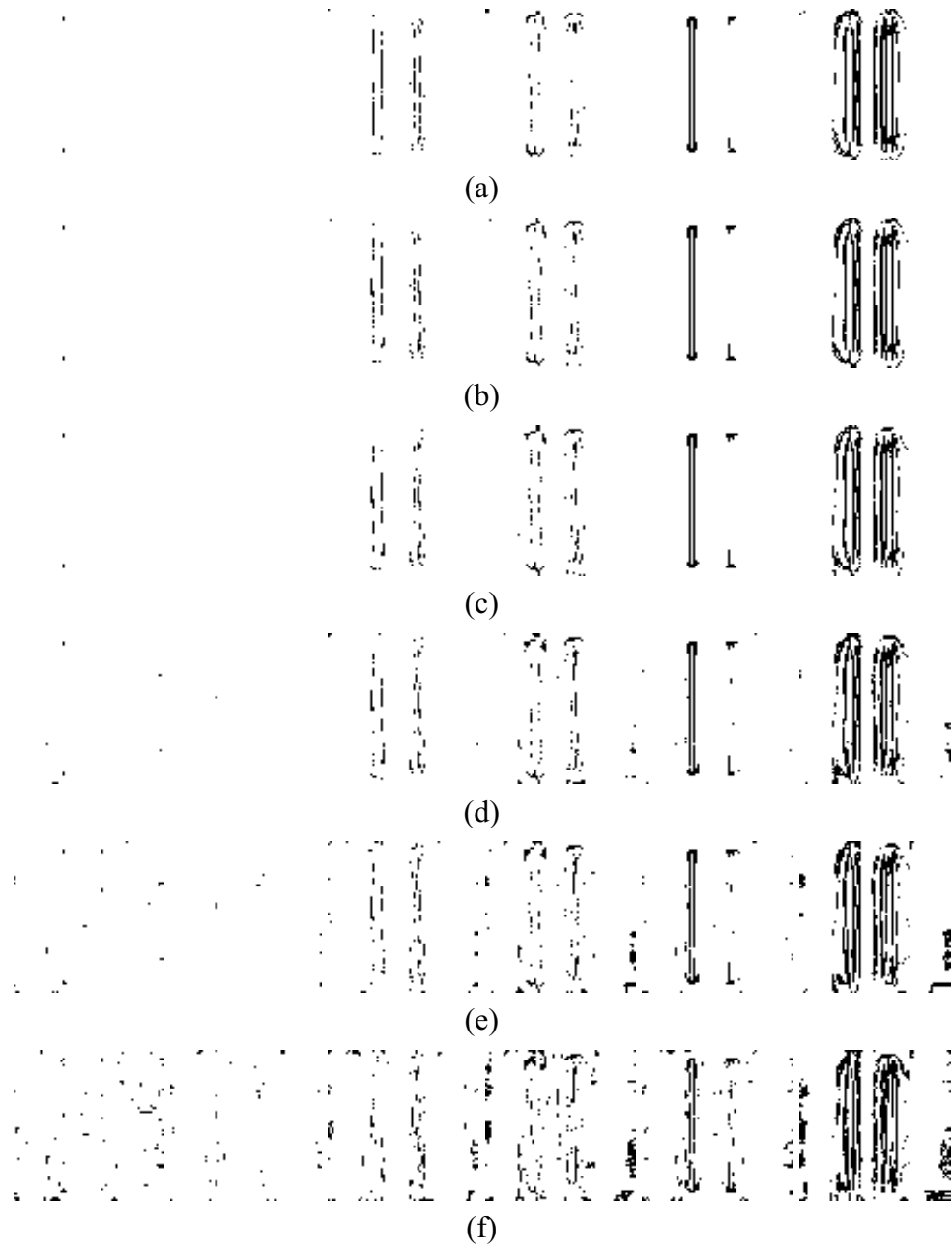
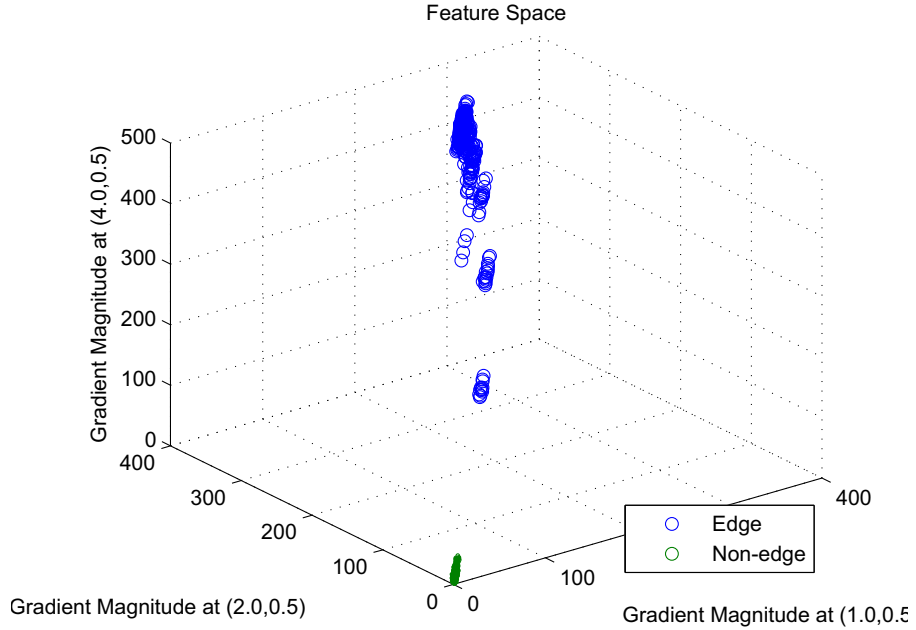
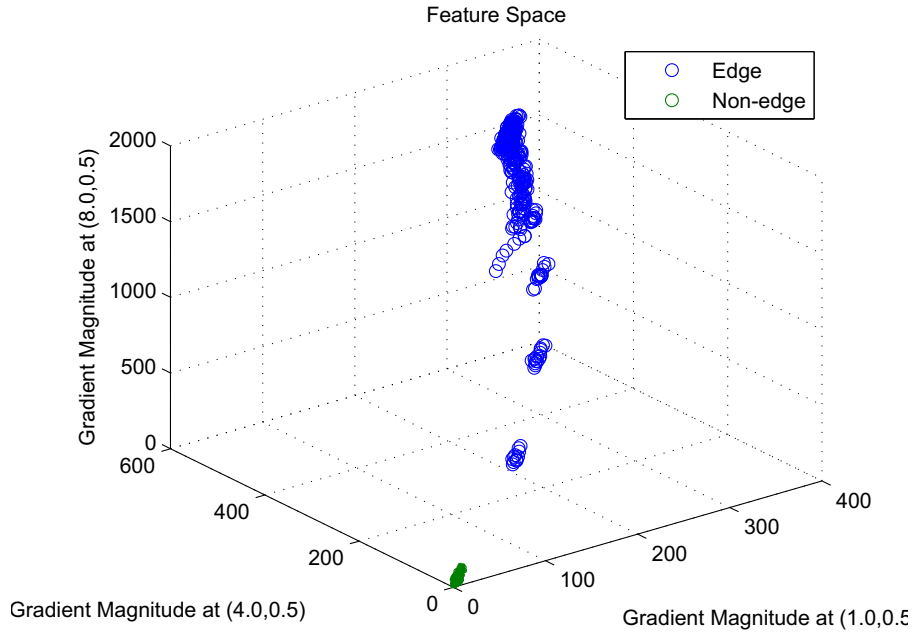


Figure C.6: Edge detection test results for the noisy bars images. The labels given in Fig. 5.1(d) is used.

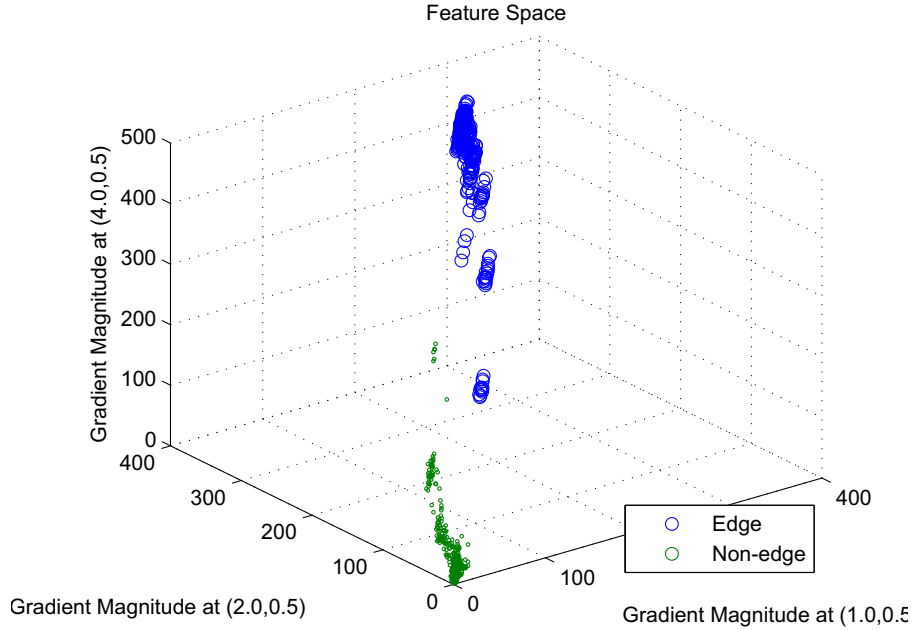


(a)

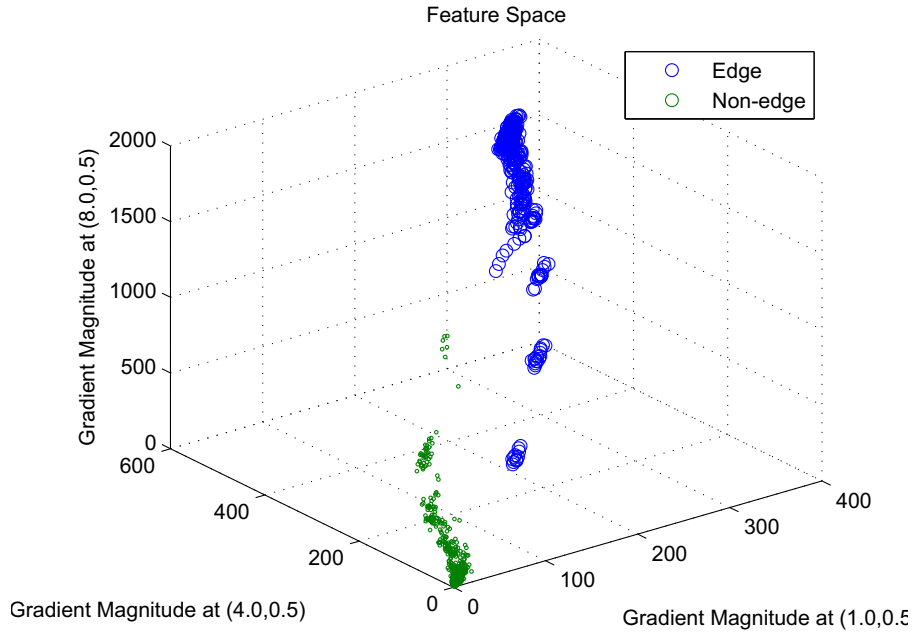


(b)

Figure C.7: Feature vector scatter diagrams with respect to different $\lambda\tau$ -space sampling schemes when the labels given in Fig. 5.1(c) is used. (a) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 2.0, \tau_1 = 0.5)$, $(\lambda_3 = 4.0, \tau_1 = 0.5)$, (b) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 4.0, \tau_1 = 0.5)$, $(\lambda_3 = 8.0, \tau_1 = 0.5)$.



(a)



(b)

Figure C.8: Feature vector scatter diagrams with respect to different $\lambda\tau$ -space sampling schemes when the labels given in Fig. 5.1(d) is used. (a) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 2.0, \tau_1 = 0.5)$, $(\lambda_3 = 4.0, \tau_1 = 0.5)$, (b) $(\lambda_1 = 1.0, \tau_1 = 0.5)$, $(\lambda_2 = 4.0, \tau_1 = 0.5)$, $(\lambda_3 = 8.0, \tau_1 = 0.5)$.

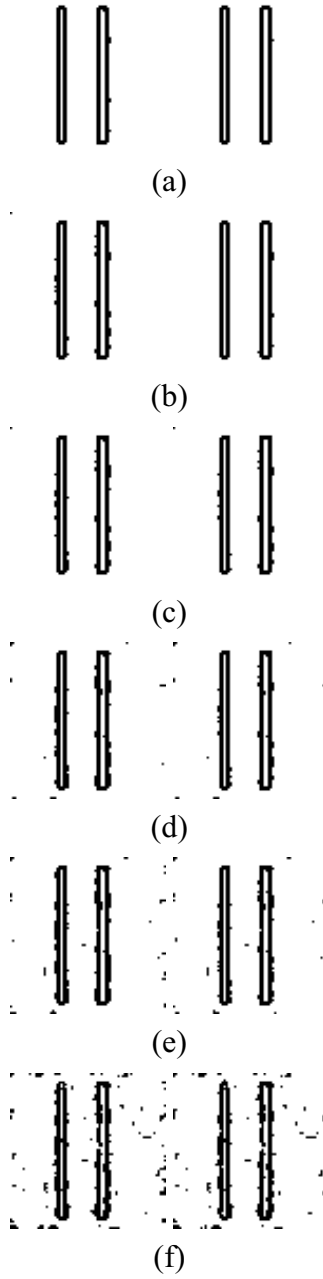


Figure C.9: Edge detection test results for the noisy bars images. Feature space is three dimensional. The labels given in Fig. 5.1(c) is used.



Figure C.10: Edge detection test results for the noisy bars images. Feature space is three dimensional. The labels given in Fig. 5.1(d) is used.

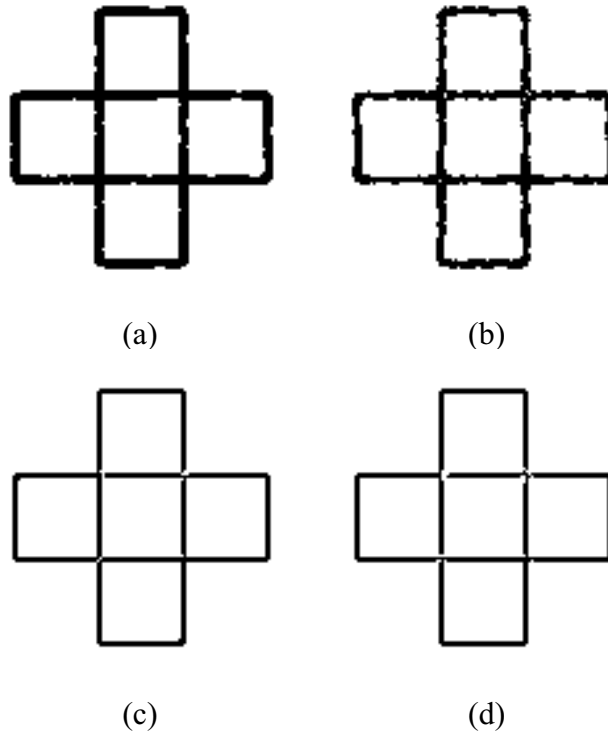


Figure C.11: Edge detection test results for the noisy checkerboard image (18dB). The labels given in Fig. 5.1(c) is used in (a) and (b). The labels given in Fig. 5.1(d) is used in (c) and (d). The following $\lambda\tau$ -space sampling schemes are used: (a),(c) $(\lambda_1 = 1.0, \tau_1 = 0.5), (\lambda_2 = 2.0, \tau_1 = 0.5), (\lambda_3 = 4.0, \tau_1 = 0.5)$, (b),(d) $(\lambda_1 = 1.0, \tau_1 = 0.5), (\lambda_2 = 4.0, \tau_1 = 0.5), (\lambda_3 = 8.0, \tau_1 = 0.5)$.

Appendix D. LICENSE-PLATE DIGIT DATABASE

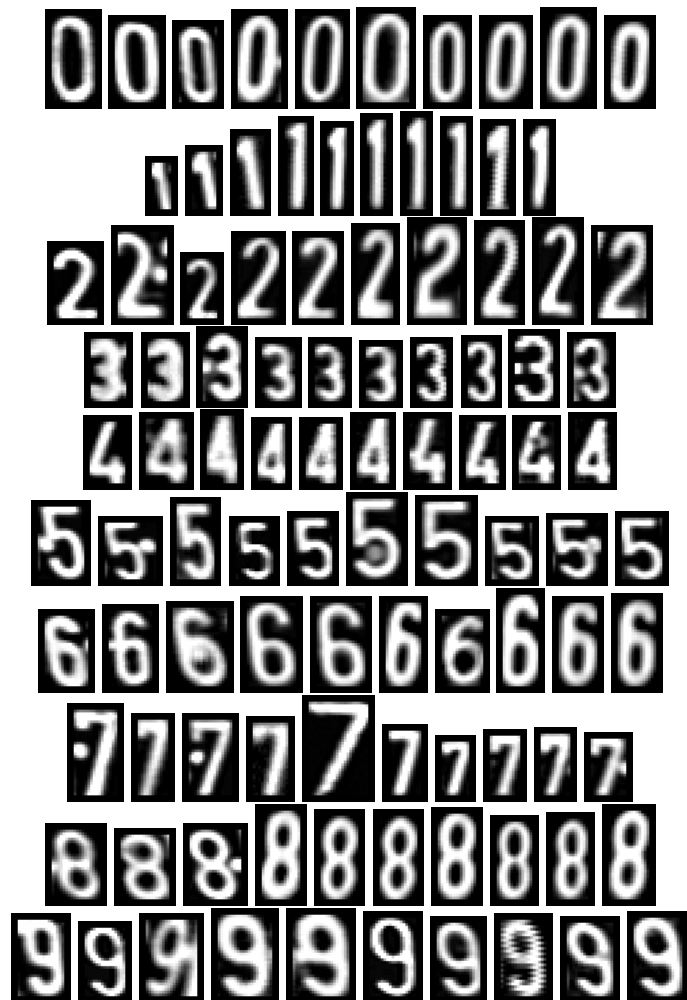


Figure D.1: License plate digits selected from the database.

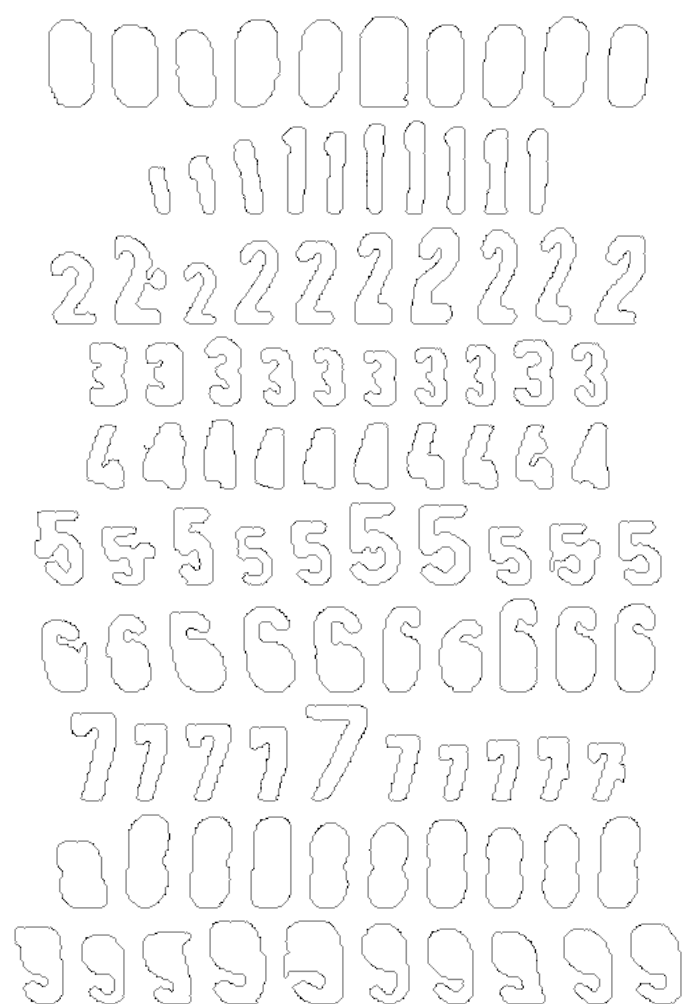


Figure D.2: Labeled contours that belong to the license plate digits selected from the database.

BIOGRAPHY

Binnur KURT graduated from Üsküdar Halide Edip Adıvar Lisesi in 1990. He received his B.Sc. in 1995 from the Control and Computer Engineering Department of Istanbul Technical University with honor, and his M.Sc. in 1997 from the Control and Computer Engineering programme at the Institute of Science and Technology of Istanbul Technical University. He worked as a research and teaching assistant at the Computer Engineering Department of Istanbul Technical University between the years 1995 and 2003. He has been working as a lecturer at the same department since 2004. He enrolled in the Ph.D. program of the Institute of Science and Technology of the same university in 1997.

During his work on Ph.D. thesis, he published the following related papers:

Journal Papers

- A.Çapar, B.Kurt, M.Gökmen, "Affine-Invariant Gradient Based Shape Descriptor," accepted to Machine Vision and Applications, 2007.
- F.Kahraman, B.Evrin Demiröz, B.Kurt, M.Gökmen, "View Independent Robust License Plate Recognition System," Industry and Automation, No. 106, pp. 28-32, January 2006.

Book Chapters

- B.Kurt, H.T.Demiral, "Adaptive $\lambda\tau$ -Space Representation of Images and Edges," Recent Advances in Signal Processing and Communications, pp.53-59, Part I: Signal Processing, World Scientific and Engineering Society Press, 1999.

International Conferences

- F. Kahraman, B.Kurt, M.Gökmen, "Robust Face Alignment for Illumination and Pose Invariant Face Recognition," IEEE Computer Vision and Pattern Recognition Workshop on Biometrics, Minneapolis 2007.
- A.Çapar, B.Kurt, M.Gökmen, "Affine-Invariant Gradient Based Shape Descriptor," Lecture Notes in Computer Science, *International Workshop on Multimedia Content Representation, Classification and Security*, pp.514-521, 2006.
- M. Gökmen, F. Kahraman, B. Kurt, "Face Alignment and Recognition Using Prominent Facial Edges", BioSec 4th Workshop (Biometrics in a global environment), Brussels, Belgium, 2005.
- B.Kurt, M.Gökmen, "Color and Shaped Based Indexing Using Scale-Space Representations," *Proceedings of the International Symposium on Artificial Intelligence and Neural Networks*, pp.30, 2-4 July 2003.
- F.Kahraman, B.Kurt, M.Gökmen, "License Plate Character Segmentation Based on the Gabor Transform and Vector Quantization," Lecture Notes in Computer Science, 18. Int. Conf. on Computer and Information Sciences-ISCIS2003, pp.381-388, 2003.
- B.Kurt, "Direction Selective Smoothing of Color Images in $\lambda\tau$ -Space," *Proceedings of 15nd International Conference on Computer and Information Sciences*, ISCIS'2000, pp.202-210, Istanbul, Turkey, October 2000.

- B.Kurt, M.Gökmen, "Two Dimensional Generalized Edge Detector," *Proceedings of International Conference on Image Analysis and Processing, ICIAP'99*, pp.148-153, Venice Italy, 1999.
- B.Kurt, H.T.Demiral, "Adaptive $\lambda\tau$ -Space Representation of Images and Edges," *Proceedings of 3rd International Conference on Communications, IMACS/IEEE CSCC'99*, pp.3011-3017, Athens GREECE, July 1999.

National Conferences

- B.Kurt, A. Çapar, M.Gökmen, "Affine Invariant Shape Descriptors," (in Turkish) *15th IEEE Conference on Signal Processing and Its Applications*, IEEE SIU'2007, 2007.
- F.Kahraman, B.Kurt, M.Gökmen, "Three-Band Modeling Using Prominent Edges for Face Alignment," (in Turkish) *14th IEEE Conference on Signal Processing and Its Applications*, IEEE SIU'2006, 2006.
- F. Kahraman, B.E. Demiröz, B. Kurt, M. Gökmen, "View Independent Robust License Plate Recognition System", (in Turkish) *13th Conference on Signal Processing and Its Applications*, SIU'2005.
- B.Kurt, M.Gökmen, "Ölçek Uzayı Gösterimine Dayalı Renk ve Şekil Tabanlı İndeksler," (in Turkish) *Proceedings of 11th Conference on Signal Processing and Its Applications*, SIU'2003, pp.805-808, İstanbul, 2003.
- B.Kurt, M.Gökmen, "İmge ve Videoların İndeksleme Amaçlı Ölçek-uzayı Analizi," (in Turkish) *Proceedings of 10th Conference on Signal Processing and Its Applications*, SIU'2002, pp.335-340, Pamukkale, 2002.
- B.Kurt, M.Gökmen, "Yinelemeli ve Uyarlanırlı Ayırıcı Saptayıcı Süzgeçleri," (in Turkish) *Proceedings of 10th Conference on Signal Processing and Its Applications*, SIU'2001, pp.410-415, Gazi Magusa, Northern Cyprus, April 2001.
- B.Kurt, H.T.Demiral, M.Gökmen, "Sahne Geçişlerinin Geometri Tabanlı Olarak Saptanması," (in Turkish) *Proceedings of 8th IEEE Conference on Signal Processing and Its Applications*, IEEE SIU'2000, 2000.
- B.Kurt, "İletim Katsayısının Koloni Sistemine Dayalı Olarak Öğrenilmesi," (in Turkish) *Proceedings of 8th IEEE Conference on Signal Processing and Its Applications*, SIU'2000, 2000.
- B.Kurt, M.Gökmen, "İki Boyutlu Genelleştirilmiş Ayırıcı Saptayıcı," (in Turkish) *Proceedings of 7th IEEE Conference on Signal Processing and Its Applications*, IEEE SIU'99, pp.481-486, Ankara Turkey, 1999.
- B.Kurt, "Yönsel Anisotropik Difüzyon ile Görüntü İyileştirme," (in Turkish) *Proceedings of 6th Conference on Signal Processing and Its Applications*, SIU'98, Vol. I, pp. 137-142, 1998.