FEATURE EXTRACTION VIA HEAT FLOW ANALOGY

By

Cem Direkoğlu

A thesis submitted for the degree of Doctor of Philosophy

School of Electronics and Computer Science,

University of Southampton,

United Kingdom.

July, 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

FEATURE EXTRACTION VIA HEAT FLOW ANALOGY

by Cem Direkoğlu

Feature extraction is an important field of image processing and computer vision. Features can be classified as low-level and high-level. Low-level features do not give shape information of the objects, where the popular low-level feature extraction techniques are edge detection, corner detection, thresholding as a point operation and optical flow estimation. On the other hand, high-level features give shape information, where the popular techniques are active contours, region growing, template matching and the Hough transform.

In this thesis, we investigate the heat flow analogy, which is a physics based analogy, both for low-level and high-level feature extraction. Three different contributions to feature extraction, based on using the heat conduction analogy, are presented in this thesis. The solution of the heat conduction equation depends on properties of the material, the heat source as well as specified initial and boundary conditions. In our contributions, we consider and represent particular heat conduction problems, in the image and video domains, for feature extraction. The first contribution is moving-edge detection for motion analysis, which is a low-level feature extraction. The second contribution is shape extraction from images which is a high-level feature extraction. Finally, the third contribution is silhouette object feature extraction for recognition purpose and this can be considered as a combination of low-level and high-level feature extraction.

Our evaluations and experimental results show that the heat analogy can be applied successfully both for low-level and for high-level feature extraction purposes in image processing and computer vision.

Contents

List of Figures

List of Tables

List of Abbreviations

ACIVS	Advanced Concepts for Intelligent Vision Systems
ACWE	Active Contour Without Edges
ADP	Anisotropic Diffusion Pyramid
BCS	British Computer Society
CCR	Correct Classification Rate
CE	Core Experiments
CF	Control Function
CPU	Central Processing Unit
CSS	Curvature Scale Space
D	Density of Salt and Pepper Noise
EFD	Elliptic Fourier Descriptors
FT	Fourier Transform
FPT	Fourier Polar Transform
FLPT	Fourier Log-Polar Transform
GAC	Geometric Active Contour
GB	Gigabyte
GFD	Generic Fourier Descriptors
GHF	Geometric Heat Flow
GHT	Generalized Hough Transform
GHz	Gigahertz
GVF	Gradient Vector Flow
GVFS	Gradient Vector Flow Snake
HFO	Heat Flow Out
HPGF	High-Pass Gaussian Filter
HT	Hough Transform
ICSP	International Conference on Signal Processing
ICVGIP	Indian Conference on Computer Vision, Graphics and Image Processing
IEEE	Institute of Electrical and Electronics Engineers
ISVC	International Symposium on Visual Computing
LNCS	Lecture Notes in Computer Science

LPGF	Low-Pass Gaussian Filter
LPT	Log-Polar Transform
MCC	Multiscale Convexity Concavity
MPEG	Moving Pictures Expert Group
OD	Object Descriptors
PAC	Parametric Active Contour
RAM	Random Access Memory
RMS	Root Mean Square
RMSE	Root Mean Square Error
SNR	Signal to Noise Ratio
SSE	Sum of Squared Error
TF	Temperature Front
ZM	Zernike Moments
1-D	One Dimension
2-D	Two Dimension

List of Symbols

a_n	Fourier coefficient of the Fourier cosine series with discrete variable
$a(\beta)$	Fourier coefficient of the Fourier cosine integral with continuous variable
A	Area
Ad	Average distance
b_f	Balloon force term
b_n	Fourier coefficient of the Fourier sine series with discrete variable
$b(\beta)$	Fourier coefficient of the Fourier sine integral with continuous variable
$\mathbf{B}(\mathbf{x})$	Binary image
$\mathbf{B}_x$ and $\mathbf{B}_y$	First order derivatives of the binary image in x- and y- directions
$\mathbf{B}_{xx}$ and $\mathbf{B}_{yy}$	Second order derivatives of the binary image in x- and y- directions
$\mathbf{B}_{\eta\eta}$ and $\mathbf{B}_{\tau\tau}$	Second order derivatives of the binary image in normal and tangent directions of the edges
c	Specific heat of the material
c_T	Scale factor for the threshold selection of the heat flow out (HFO) image
c_o	Total number of correctly classified objects
$\mathbf{CF}(\mathbf{x},t)$	Control Function
(c_x, c_y)	Actual circle centre parameters
$C \times E$	Size of the polar mapped image
div	Divergence operator
$\mathbf{E}_n$	Sobel edge mapped image for the frame n
(e_x, e_y)	Extracted circle centre parameters
$Ed^s(\mathbf{T},\mathbf{D})$	Euclidean distance between the test image descriptors and the database image descriptors at scale index s

$F(\mathbf{x})$	Initial temperature inside the body
f	Heat flow rate
f_x	Heat energy in at the left face
f_{x+dx}	Heat energy out at the right face
$\mathbf{FT}(u,v)$	Fourier transform image
$\mathbf{FPT}^s(k,l)$	Fourier Transform of the polar mapped image at scale index *s*
$F^{in}(C)$	Fitting term inside the contour
$F^{out}(C)$	Fitting term outside the contour
$g(\nabla\mathbf{I})$	Stopping function derived from the gradient of the image
$\mathbf{G}$	Grey-level image
$G(x,y,x',y',\tau)$	Greens function
$G_\sigma(\mathbf{x})$	Gaussian filter with standard deviation σ
h_A	Step space parameter of the ACWE
$H(z)$	Heaviside function
$H_\varepsilon(z)$	Regularized version of the Heaviside function
$\mathbf{I}(\mathbf{x},t)$	Image as a function of position and time
I^{in}	Piecewise constant intensity of the object region
I^{out}	Piecewise constant intensity of the background region
k	Thermal conductivity
k_N	Conductivity coefficient in the north direction
k_S	Conductivity coefficient in the south direction
k_E	Conductivity coefficient in the east direction
k_W	Conductivity coefficient in the west direction
κ	Curvature term
L	Length of the one dimensional body
$M \times N$	Size of the 2-D image
$\vec{n}$	Normal vector of the level set function
$N_d(\mu,\sigma^2)$	Normal distribution with mean μ and standard deviation σ
$\mathbf{O}(\mathbf{x})$	Actual segmented image (ground truth)
$\mathbf{OD}^s$	Object descriptors at scale index *s*

$\mathbf{OD}_{\mathbf{T}}^{s}$	Object descriptors of the test image at scale index *s*
$\mathbf{OD}_{\mathbf{D}}^{s}$	Object descriptors of the image from database at scale index *s*
ρ	Density of the material
$\mathbf{P}^{s}(r,\theta)$	Polar mapped image at scale index *s*
q	Heat source
q_c	Constant heat source
Q	Heat source term
R	The rate of decay of the exponential function
(r,θ)	Polar Coordinate systems, *r* defines radial distance and θ is angle
s	Scale index
s_T	Scale factor for the threshold selection of the 2-D Sobel edge mapped image
$\mathbf{S}(\mathbf{x})$	Binary form of the final segmentation
t	Time (iteration)
t_s	Number of iterations for geometric heat flow
t_o	Total number of classified objects
T	Temperature
T_H	Higher threshold of the Hystresis thresholding
T_L	Lower threshold of the Hystresis thresholding
U	Speed term in level sets
v_A	Area parameter of the ACWE
$\mathbf{V}(\mathbf{x}) = (\mathbf{u}(\mathbf{x}), \mathbf{v}(\mathbf{x}))$	Gradient vector field
$\mathbf{x} = (x, y)$	Space vector for the two-dimension
(x_c, y_c)	Centre of the Fourier magnitude image
Y	Number of scales
α	Thermal diffusivity
α_G	Smoothness parameter of the GVFS
β_G	Rigidity parameter of the GVFS
σ	Standard deviation
σ_s	Scale parameter (standard deviation) at scale index *s*
$\sigma_{in}(\mathbf{x},t)$	Weighted square of the difference between the intensity value and

	the mean of the intensity values inside the contour, at each point and time.
$\sigma_{out}(\mathbf{x},t)$	Weighted square of the difference between the intensity value and the mean of the intensity values outside the contour, at each point and time.
λ	Parameter for the rate of anisotropic diffusion in 2-D
λ_n and β_m	Eigenvalues
λ_{in} and λ_{out}	Regional statistic parameters
μ	Mean of the normal distribution
μ_A	Length parameter of the ACWE
μ_G	Regularization parameter in diffusion of the gradient vectors
μ_{in}	Mean inside the contour
μ_{out}	Mean outside the contour
γ_G	External energy parameter of the GVFS
φ	Angle (which a point may be rotated in the image domain)
$\Phi(\mathbf{x},t)$	Temperature at the boundary layer of the body
$\phi(x,y,t)$	Level set function
ξ	Energy functional
$\xi_{ext}(C)$	External energy function
$\delta(z)$	Dirac delta function
$\delta_\varepsilon(z)$	Regularized version of the Dirac delta function
η	Direction normal to the feature boundary
τ	Tangent direction to the feature boundary
ε	Parameter for the Heaviside and Dirac delta functions in the ACWE
∇	Gradient operator
∇_N	Gradient operator in the north direction
∇_S	Gradient operator in the south direction
∇_E	Gradient operator in the east direction
∇_W	Gradient operator in the west direction
Δ	Laplacian operator

Declaration of Authorship

I, *Cem Direkoğlu* declare that the thesis entitled *Feature Extraction via Heat Flow Analogy* and the work presented in it are my own. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- part of this work have been published as:

  - Cem Direkoğlu and Mark S. Nixon. Low Level Moving-Feature Extraction via Heat Flow Analogy, *International Symposium on Visual Computing (ISVC 2006),* LNCS 4291: 243-252, November 2006.
  - Cem Direkoğlu and Mark S. Nixon. Shape Extraction via Heat Flow Analogy, *International conference on Advanced Concepts for Intelligent Vision Systems (ACIVS 2007),* LNCS 4678: 553-564, August 2007.
  - Cem Direkoğlu and Mark S. Nixon. Shape Classification using Multiscale Fourier-based Description in 2-D Space, *IEEE International Conference on Signal Processing (ICSP 2008)*, 1: 820-823, October 2008.
  - Cem Direkoğlu and Mark S. Nixon. Image-based Multiscale Shape Description using Gaussian Filter, *IEEE Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2008),* pages: 673-678, December 2008.
  - Mark S. Nixon, Xin U. Liu, Cem Direkoğlu and David J. Hurley. On Using Physical Analogies for Feature and Shape Extraction in Computer Vision, *British Computer Society Visions of Computer Science Conference (BCS 2008),* pages: 163-177, September 2008.

Signed: Cem Direkoğlu

Date: 15/6/2009

Acknowledgements

First, I would like to thank my supervisor Prof Mark Nixon for his guidance and feedbacks throughout this research, for his efforts in revising papers, reports and thesis, for supporting me to attend computer vision summer school and conferences, as well as for his conversations and companies.

I would like to acknowledge Prof David Nunn for his time to answer my questions about the heat analogy. I thank to Dr. John Carter for giving me important feedbacks in my research. I also thank to John Wynn for his system support during my PhD.

My thanks also go to my friends in ISIS; Bassam, Shodhan, Daniel, Hidayah, Salih, Ramanan, Aly, Xin, Ke, Ni, Iain and Richard for their conversations and companies and who made my stay in the university of Southampton a rich experience.

Finally, I would like to express my gratitude to my family and my fiancée Melike for always being there to support and believe in me, and their motivation throughout my PhD.

This thesis is dedicated to my family.

Chapter 1 Introduction

Feature extraction is an area of image processing and computer vision, which involves algorithms to detect and isolate various desired portions of a digital image or video sequence. Feature extraction largely concerns finding shapes or their descriptors. By feature extraction, the input data is transformed into a small set of features, which are reduced representation of the original data. Features can be used for different purposes, such as classifying a detected object into different catagories or estimation of application dependent parameters (pose and size of the object). Feature extraction can be classified as low-level or high-level [1]. Low-level features are defined as basic features that can be extracted automatically from an image without any shape information. Well known low-level feature extraction techniques are thresholding (as a point operation), edge detection, corner (curvature) detection and optical flow estimation. Naturally, all of these techniques can be used within high level feature extraction. High-level feature extraction desires to find shapes. Well known high-level feature extraction techniques include the Hough Transform (HT), Generalized Hough Transform (GHT), pre-defined template matching, deformable template matching, region growing and active contours (snakes).

Physics based analogies can be used for feature extraction in image processing and computer vision. Hurley et al. [2, 3] introduced a feature extraction for ear biometrics by using gravitational force field model. Liu and Nixon [4, 5] proposed a water flow model to extract complex shapes from still images. Recently, Nixon et al. [6, 7] described how physics based analogies can be used for low-level and high-level feature analysis, which also includes this research. In this research, we investigate heat flow as a new analogy for feature extraction and introduce new techniques. The first technique is moving-edge detection that is low-level feature extraction in motion analysis. The second technique is shape extraction from still images, which is high-level feature extraction. The final technique is silhouette object feature extraction for the purpose of

recognition, which can be considered as both low-level and high-level feature extraction.

1.1 Using Heat Flow in Image Analysis

The heat flow analogy has been deployed in various ways in image processing and computer vision. It has been used for: image smoothing and enhancement; region-based image segmentation; thinning; active contours; motion analysis and graph theory. Anisotropic diffusion, which was introduced to computer vision by Perona and Malik [8], is the state-of-art image enhancement technique. In [9], the anti-geometric heat flow model was introduced for the segmentation of regions. Here, anti-geometric heat flow is represented as diffusion through the normal direction of edges. In [10], a new thinning algorithm was introduced based on time-reversed isotropic heat flow. A geometric active contour [11, 12] is based on a curve moving in normal direction with its curvature dependent speed, which is also called level set method [13]. The curve movement in the level set approach is achieved with geometric heat flow [14]. In motion analysis, Makrogiannis and Bourbakis [15] proposed a spatio-temporal anisotropic heat diffusion for motion activity measurement. An extensive survey about the applications of heat flow in image processing and computer vision is given in Chapter 2.

In addition, the Fourier series, which is actively involved in feature extraction and image processing, was first introduced for the purpose of solving the heat equations [16]. This connection also allows us to investigate Fourier theory for feature extraction.

1.2 Contributions to Feature Extraction using Heat Flow Analogy

We have three different contributions to feature extraction based on using the heat flow analogy.

1.2.1 Moving-edge Detection

In this research, a moving object edge detection algorithm is proposed based on the heat flow analogy. This algorithm starts with anisotropic heat diffusion in the spatial domain to remove noise and sharpen region boundaries for the purpose of obtaining

high quality edge maps. Once the enhanced edge maps are observed in three or more consecutive frames, isotropic and linear heat diffusion is applied in the temporal domain to calculate the total amount of heat flow. The moving-edge map is represented as the total amount of heat flow out (-) from the reference frames. The overall process is completed by non-maxima suppression for thinning and then hysteresis thresholding to obtain binary moving-edges. Evaluation results indicate that this approach has advantages in handling noise in the temporal domain because of the averaging inherent of isotropic heat flow. Results also show that this technique can detect moving-edges in image sequences without using the background image. This work has been published by International Symposium on Visual Computing in 2006 (ISVC 2006) [17].

1.2.2 Shape Extraction

In this research, we introduce a novel evolution-based segmentation algorithm based on the heat flow analogy. Our model is new and different because of the evolution technique and organization of segmentation. The evolution is achieved with the heat conduction equations to gain practical advantages. Our model is also organized effectively to improve the segmentation.

The proposed algorithm consists of two parts. In the first part, we represent a particular heat conduction problem in the image domain to roughly segment objects of interest. In this problem, we consider a two-dimensional conductive solid body that has uniform conductivity within an isotropic medium. Initial and boundary conditions are given, respectively, by $T(\mathbf{x}, t=0)=0$ and $T(\mathbf{x}, t)=0$, where T represents the temperature at position $\mathbf{x}=(x, y)$ and time t. The given conditions mean that the temperature is initially zero inside the body and the boundary condition is "Dirichlet" that has specified temperature, zero, at the boundary layer for all time. If we initialize a continuous heat source (a positive constant) at any point inside the body, there will be heat diffusion to the other points from the source position as time passes and this will cause temperature increase within the body, except at the boundary layer. This concept is represented in the image domain by using a control function in the heat conduction equation. The control function is obtained from the region's statistics of the source location, since we propose to segment the source located region. However, in noisy conditions, we can observe irregular boundaries and holes inside the segmented region.

These problems are solved in the second part of the algorithm, which is geometric heat flow. In this part, the segmented image is first converted to binary form and then geometric heat flow is applied to reduce curvature in the boundary, as well as to remove holes inside the segmented region. After a specified number of iterations, the resultant image is thresholded and the final segmentation is obtained. Experimental results indicate that the proposed algorithm works well in noisy conditions without pre-processing. It can detect multiple objects simultaneously, if a heat source is located at the background. It is also computationally more efficient and easier to control and implement in comparison to active contour models. As such, by using physics based analogies, we can control the segmentation process so as to achieve a result which offers improved segmentation, by a better fit to the image data. This work has been published by international conference on Advanced Concepts for Intelligent Vision Systems in 2007 (ACIVS 2007) [18].

1.2.3 Object Recognition

Solution of heat conduction problems can be achieved by using Fourier series or Fourier integrals [16, 19, 20]. Furthermore, the solution of linear and homogeneous heat equation in an infinite medium can be viewed as smoothing a signal with a Gaussian filter [19, 20], as we show in Chapter 2. This allows us to investigate Fourier series and Gaussian filtering for feature extraction as well. In our research, we use Fourier theory and the Gaussian filtering approach to introduce new and efficient silhouette object descriptors for recognition purposes.

In shape recognition, a multiscale description provides more information about the object, increases discrimination power and immunity to noise. We develop new multiscale Fourier-based object descriptors in 2-D space. We investigate both the LPGF and the HPGF based multiscale representation separately. The LPGF applies smoothing to the object and as scale (standard deviation) decreases, it causes loss of boundary and exterior regions. Therefore using the LPGF at different scales focuses on the inner and central part more than on the boundary of an object. On the other hand, using the HPGF at different scales emphasizes the boundary and exterior parts of an object more than the central part. Our algorithm is organized to achieve size, translation and rotation invariance. First we apply size normalization of the silhouette image and we then compute a Fourier magnitude image that is translation invariant. At this stage,

a LPGF or a HPGF with a selection of different scale parameters is applied to the Fourier magnitude image and produces different Fourier magnitude images. The Fourier magnitude images obtained at each scale vary with rotation and are not convenient for matching. To give rotation invariance, each Fourier magnitude image is polar mapped to convert rotation to translation and then Fourier transform of polar image is computed to obtain the Fourier magnitude. Finally, the obtained Fourier magnitude image is size, translation and rotation invariant and provides descriptors of an object at that scale. For classification, the Euclidean distance is calculated separately at each scale and then the average distance is computed for each object. By classifying with average distance, increase immunity to noise as well as increase correct classification rate is observed. Evaluation indicates that the HPGF based multiscale representation performs better than the LPGF based multiscale representation, and in comparison to Zernike moments and elliptic Fourier descriptors with respect to increasing noise. Multiscale description using HPGF in 2-D also outperforms Wavelet transform based multiscale contour Fourier descriptors and performs similar to the perimeter descriptors without any noise. Part of this work, which uses HPGF for multiscale generation, has been published by IEEE International Conference on Signal Processing in 2008 (ICSP 2008) [21]. Than the extended version, which includes both LPGF and HPGF based multiscale description, has been published by IEEE Indian Conference on Computer Vision, Graphics and Image Processing in 2008 (ICVGIP 2008) [22].

1.3 Thesis Overview

This thesis describes three different contributions to feature extraction based on using the heat flow analogy. Chapters are organized as follows: Chapter 2 explains the basic concepts of heat flow and applications of heat flow in image processing and computer vision. Chapter 3 introduces novel low-level moving feature extraction that is moving-edge detection based on heat flow. Chapter 4 introduces the new shape extraction technique by using the heat flow analogy. Chapter 5 introduces the new object recognition technique using the multiscale Fourier-based description in 2-D space. Finally, Chapter 6 is overall conclusions. A proof of the resulting shape extraction, related to Chapter 4, is given at the end of this report in Appendix A. Appendix B explains the active contour models, which are compared with our shape extraction technique in Chapter 4.

1.4 List of Publications related to this thesis

We have six publications related to our contributions. The first contribution is moving-edge detection, which is low-level feature extraction in motion analysis. The related publication of this work:

- Cem Direkoğlu and Mark S. Nixon. Low Level Moving-Feature Extraction via Heat Flow Analogy, *International Symposium on Visual Computing (ISVC 2006),* LNCS 4291: 243-252, November 2006.

The second contribution is shape extraction (segmentation) from images that is high-level feature extraction. The related publication is

- Cem Direkoğlu and Mark S. Nixon. Shape Extraction via Heat Flow Analogy, *International conference on Advanced Concepts for Intelligent Vision Systems (ACIVS 2007),* LNCS 4678: 553-564, August 2007.

The final contribution is Multiscale Fourier-based object description using Gaussian filter in 2-D space. We first propose HPGF based multiscale description and the related publication is

- Cem Direkoğlu and Mark S. Nixon. Shape Classification using Multiscale Fourier-based Description in 2-D Space, *IEEE International Conference on Signal Processing (ICSP 2008)*, 1: 820-823, October 2008.

Than we extend this work with LPGF based multiscale description and compare it with the HPGF based multiscale description. The publication related to this extended version is given below,

- Cem Direkoğlu and Mark S. Nixon. Image-based Multiscale Shape Description using Gaussian Filter, *IEEE Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2008),* pages: 673-678, December 2008.

Later, we describe how physical analogies based on gravitational force, water flow and heat can be deployed to achieve feature extraction. This work also includes our research. The related publication is

- Mark S. Nixon, Xin U. Liu, Cem Direkoğlu and David J. Hurley. On Using Physical Analogies for Feature and Shape Extraction in Computer Vision, *British Computer Society Visions of Computer Science Conference (BCS 2008),* pages: 163-177, September 2008.

The publication given above is also identified as being one of the best papers that were accepted by the conference *BCS 2008,* and has been selected for submission to a special

issue of The Computer Journal. The Computer Journal is the BCS's 50-year-old scientific journal. The submitted work is given below

- Mark S. Nixon, Xin U. Liu, Cem Direkoğlu and David J. Hurley, On Using Physical Analogies for Feature and Shape Extraction in Computer Vision, *British Computer Society, The Computer Journal,* Submitted.

Chapter 2 Heat Flow in Image Processing and Computer Vision

This Chapter explains the basic concepts of heat flow in Physics and then discusses the applications of heat flow in image processing and computer vision.

2.1 Basic Concept of Heat Flow

In Physics, heat is the form of energy transferred due to the temperature difference within or between bodies. Temperature is the measure of hotness or coldness of the body. Conduction, convection and radiation are three different modes of heat flow. Here, we chose to investigate use of a conduction model, which we found to operate well. Conduction is the flow of heat energy from high- to low- temperature regions due to the presence of a thermal gradient in a body [23]. According to Fourier's law of heat conduction, the heat flow rate per unit area is,

$$f = -k\left(\partial T/\partial x\right) \tag{2.1}$$

Where, f represents the heat-flow rate, k is positive constant that is called the thermal conductivity of a material, $\partial T/\partial x$ is the temperature gradient and the minus sign indicates that heat flows in the opposite direction to the temperature gradient, satisfying the second principle of thermodynamics.

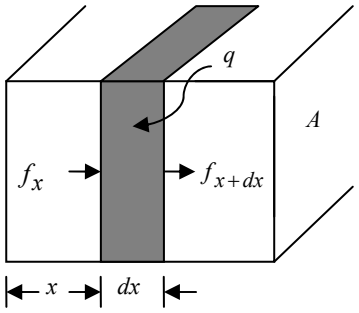


Figure 2.1 One-dimensional heat conduction

If we consider a one-dimensional heat conduction system for an element of thickness dx and area A as shown in Figure 2.1, by the conservation of energy, the energy conducted in at the left face plus the energy generated within element should be equal to the change in internal energy plus the energy conducted out at the right face. These energy quantities are represented mathematically as follows:

$$\text{Energy in at the left face} \quad = f_x = -kA\left(\partial T/\partial x\right) \tag{2.2}$$

$$\text{Energy generated within element} \quad = qAdx \tag{2.3}$$

$$\text{Change in internal energy} \quad = \rho cA\left(\partial T/\partial t\right)dx \tag{2.4}$$

$$\text{Energy out at the right face} \quad = f_{x+dx} = -kA\left(\partial T/\partial x\right)\Big]_{x+dx} \tag{2.5}$$

Where, q is a heat source, which generates energy within element. c is the specific heat of the material and ρ is its density. Combining these relations gives

$$-kA\frac{\partial T}{\partial x} + qAdx = \rho cA\frac{\partial T}{\partial t}dx - A\left[k\frac{\partial T}{\partial x} + \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)dx\right] \tag{2.6}$$

and after simplification, we obtain the general heat conduction or diffusion equation,

$$\partial T/\partial t = \alpha\left(\partial^2 T/\partial x^2\right) + Q \tag{2.7}$$

Where, $\alpha = k/\rho c$ is called thermal diffusivity of the material and a larger values of α indicate faster heat diffusion through the material. $Q = q/\rho c$ is the source term that

applies internal heating as a function of heat source. It can be uniformly or non-uniformly distributed over material body.

The two-dimensional heat conduction equation can also be derived by the process explained above and obtained as

$$\partial T/\partial t = \alpha\left(\partial^2 T/\partial x^2 + \partial^2 T/\partial y^2\right) + Q = \alpha\Delta T + Q \tag{2.8}$$

where Δ represents the spatial Laplace operator. Here, we want to note that Equations 2.7 and 2.8 are obtained by assuming the material has constant conductivity, k, which is called uniform conductivity and in this case the heat equation is *linear*. If the conductivity of a material varies from point to point as a function of position or temperature, the material has non-uniform conductivity and the heat equation is *non-linear* [24]. Furthermore, if a material has conductivity which is same in all directions the material medium is called *isotropic*. Alternatively, a material medium is called *anisotropic,* which means there exists directional variation of conductivity. Equations 2.7 and 2.8 are obtained by assuming that material has uniform conductivity and is an isotropic medium. The solution of Equation 2.8 provides the temperature distribution over the material body and it depends on time, distance, heat source, properties of material, as well as on specified initial and boundary conditions.

Initial conditions specify the temperature distribution in a body, as a function of space coordinates, at the origin of the time coordinate $(t = 0)$. Initial conditions are represented as follows,

$$T(\mathbf{x}, t = 0) = F(\mathbf{x}) \tag{2.9}$$

where, $\mathbf{x} = (x, y)$ is the space vector for the two-dimensional case and $F(\mathbf{x})$ is the function that specifies the initial temperature inside the body.

Boundary conditions specify the temperature or the heat flow at the boundaries of the body. There are three general types of boundary conditions: Dirichlet, Neuman and Robin. In the Dirichlet condition, temperature is specified along the boundary layer. It can be a function of space and time, or constant. The Dirichlet condition is represented as follows,

$$T(\mathbf{x},t)=\Phi(\mathbf{x},t) \tag{2.10}$$

where $\Phi(\mathbf{x},t)$ is the function that specifies the temperature at the boundary layer.

In the Neuman condition, the normal derivative of temperature, heat flow, is prescribed at the boundary surface, and it can be a function of space and time, or constant. The Neuman condition is given in the form

$$\frac{\partial T(\mathbf{x},t)}{\partial n}=\Phi(\mathbf{x},t) \tag{2.11}$$

where, $\Phi(\mathbf{x},t)$ is the function that specifies the normal derivative of temperature at the boundary layer.

In the Robin boundary conditions, a linear combination of the temperature and its normal derivative is prescribed at the boundary surface. These are usually used when the body is in a moving fluid. It is represented as follows,

$$hT(\mathbf{x},t)+z\frac{\partial T(\mathbf{x},t)}{\partial n}=\Phi(\mathbf{x},t) \tag{2.12}$$

where, h and z are non-zero constants and $\Phi(\mathbf{x},t)$ is the function that represents the values of this linear combination.

The heat equation or boundary condition is called *homogeneous* if its terms are the first degree of the function $T(\mathbf{x},t)$, derivatives of $T(\mathbf{x},t)$ or zero itself. Thus the heat equation, in Equation 2.7 and in Equation 2.8, is homogeneous, if it does not have source term. Equations 2.10, 2.11 and 2.12, which represent boundary conditions, are also homogeneous, if $\Phi(\mathbf{x},t)=0$. The heat conduction problem is referred to as a homogeneous problem when both the heat equation and the boundary conditions are homogeneous.

The analytical solution of the linear homogeneous heat conduction problems can be achieved using Fourier series or Fourier integrals [19, 20]. The Fourier series was first introduced to solve heat conduction problems by Joseph Fourier [16]. Although his

motivation was heat conduction, the Fourier series is now applied in many mathematical and physical problems, including signal and image processing. The Fourier series is a method that represents periodic functions as a weighted sum of sine and cosine waves, where the weights are called Fourier coefficients. Given the initial and the boundary conditions of the linear homogeneous heat equation, the Fourier series or integral solution can be achieved by first applying separation of variables method and then applying the principle of superposition [19, 20].

Consider a one-dimensional conductive solid body in $0 \le x \le L$, without any heat generation (heat source). Initial and boundary conditions respectively given by $T(x,t=0)=F(x)$ and $T(0,t)=T(L,t)=0$, which means that the temperature is initially $F(x)$ inside the body and the boundary condition is Dirichlet that has specified temperature (zero) at the boundaries. The solution of this problem is achieved with Fourier sine series as given below,

$$T(x,t=0)=F(x)=\sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{L}\right), \quad T(x,t)=\sum_{n=1}^{\infty} b_n \exp\left(-\frac{\alpha n^2\pi^2 t}{L^2}\right)\sin\left(\frac{n\pi x}{L}\right) \tag{2.13}$$

$$b_n=\frac{2}{L}\int_0^L F(x)\sin\left(\frac{n\pi x}{L}\right)dx \tag{2.14}$$

Here, $F(x)$ is represented with Fourier sine series with respect to Fourier coefficients b_n. $T(x,t)$ is the solution that satisfies all conditions and it is again obtained with Fourier series, but with respect to Fourier coefficients weighted by exponential function, $\exp\left(-\alpha n^2\pi^2 t/L^2\right)$, of time parameter t.

If we consider same solid with initial and boundary conditions respectively given by $T(x,t=0)=F(x)$ and $\partial T(0,t)/\partial n=\partial T(L,t)/\partial n=0$, which means that the temperature is initially $F(x)$ inside the body and the boundaries are insulated with Neuman boundary condition. The solution of this problem is achieved by Fourier cosine series as given below,

$$T(x,t=0)=F(x)=\sum_{n=0}^{\infty}a_n\cos\left(\frac{n\pi x}{L}\right),\ T(x,t)=\sum_{n=0}^{\infty}a_n\exp\left(-\frac{\alpha n^2\pi^2 t}{L^2}\right)\cos\left(\frac{n\pi x}{L}\right) \quad (2.15)$$

$$a_n=\frac{2}{L}\int_0^L F(x)\cos\left(\frac{n\pi x}{L}\right)dx \quad (2.16)$$

where, $F(x)$ is represented with Fourier cosine series with respect to Fourier coefficients a_n and $T(x,t)$ is the solution that satisfies all conditions and again Fourier coefficients are weighed by $\exp\left(-\alpha n^2\pi^2 t/L^2\right)$ to obtain the temperature distribution at specified time t.

We now consider same type of solid in $-L\le x\le L$ with initial condition $T(x,t=0)=F(x)$ and boundary conditions $T(-L,t)=T(L,t)$, $\partial T(-L,t)/\partial n=\partial T(L,t)/\partial n$. These types of boundary conditions are called Periodic boundary conditions. The left boundary tends to be $x=-L$ instead of $x=0$. These boundary conditions are also homogeneous, which can be seen better by rewriting as $T(-L,t)-T(L,t)=0$ and $\partial T(-L,t)/\partial n-\partial T(L,t)/\partial n=0$. The solution of this problem is given below,

$$T(x,t=0)=F(x)=\sum_{n=0}^{\infty}a_n\cos\left(\frac{n\pi x}{L}\right)+\sum_{n=1}^{\infty}b_n\sin\left(\frac{n\pi x}{L}\right) \quad (2.17)$$

$$T(x,t)=\sum_{n=0}^{\infty}a_n\exp\left(-\frac{\alpha n^2\pi^2 t}{L^2}\right)\cos\left(\frac{n\pi x}{L}\right)+\sum_{n=1}^{\infty}b_n\exp\left(-\frac{\alpha n^2\pi^2 t}{L^2}\right)\sin\left(\frac{n\pi x}{L}\right) \quad (2.18)$$

$$a_0=\frac{1}{2L}\int_{-L}^{L}F(x)dx,\quad a_n=\frac{1}{L}\int_{-L}^{L}F(x)\cos\left(\frac{n\pi x}{L}\right)dx,\quad b_n=\frac{1}{L}\int_{-L}^{L}F(x)\sin\left(\frac{n\pi x}{L}\right)dx \quad (2.19)$$

Here, $F(x)$ is represented by full Fourier series with respect to coefficients a_n and b_n. $T(x,t)$ is the solution that satisfies all conditions, and both of the Fourier coefficients, a_n and b_n, are weighed by $\exp\left(-\alpha n^2\pi^2 t/L^2\right)$ to obtain the temperature distribution at specified time t.

Finally, if we consider the one-dimensional solid in an infinite medium, $-\infty\le x\le\infty$, which is initially at a temperature $T(x,t=0)=F(x)$. No boundary conditions are

specified for the problem since the medium extends to infinity in both directions. The solution of this heat conduction problem is given below,

$$F(x) = T(x, t = 0) = \int_{\beta=0}^{\infty} [a(\beta)\cos(\beta x) + b(\beta)\sin(\beta x)]d\beta \tag{2.20}$$

$$T(x,t) = \int_{\beta=0}^{\infty} \exp(-\alpha\beta^2 t)[a(\beta)\cos(\beta x) + b(\beta)\sin(\beta x)]d\beta \tag{2.21}$$

Where,

$$a(\beta) = \frac{1}{\pi}\int_{-\infty}^{\infty} F(x')\cos(\beta x')dx',\ b(\beta) = \frac{1}{\pi}\int_{-\infty}^{\infty} F(x')\sin(\beta x')dx' \tag{2.22}$$

β is continuous variable and has values from zero to infinity. Here, $F(x)$ is represented by the Fourier integral formula with respect to coefficients $a(\beta)$ and $b(\beta)$. $T(x,t)$ is the solution that satisfies all conditions, and both of the Fourier coefficients, $a(\beta)$ and $b(\beta)$, are weighed by $\exp(-\alpha\beta^2 t)$ to obtain the temperature distribution at specified time t. After substituting Equation 2.22 to the Equation 2.20 and rearrangements [19, 20], the solution alternatively can be written as follows,

$$T(x,t) = \frac{1}{(4\pi\alpha t)^{1/2}}\int_{-\infty}^{\infty} F(x')\exp\left[-\frac{(x-x')^2}{4\alpha t}\right]dx' \tag{2.23}$$

From this equation, we can observe that solution of heat equation in infinite space, $-\infty \le x \le \infty$, can be viewed as convolving a signal $F(x)$ with a Gaussian function and the standard deviation of the Gaussian function is represented as $\sqrt{2\alpha t}$.

There are many heat conduction problems which do not have an analytical solution. These problems usually involve geometrical shapes that are mathematically unsuited to representing initial and boundary conditions. However, numerical techniques exist, such as finite differences and finite elements, which are able to handle almost all problems with any complex shapes. The numerical methods yield numerical values for temperatures at selected discrete points within the body and only at discrete time

intervals. Thus answers are obtained only for a given set of conditions, a given set of discrete points and discrete time intervals.

The numerical heat conduction problem can be investigated in the image domain, since the image is formed by a set of points, as well as since the image is convenient for the finite difference technique. We will not consider the finite element methods further, since our aim is to use the heat analogy for feature extraction. As such, the finite element methods would become rather complex for this aim.

2.2 Applications of Heat Flow in Image Processing and Computer Vision

The heat flow analogy has been deployed in various ways in image processing and computer vision. Six applications are briefly surveyed here: image smoothing and enhancement; region-based image segmentation; thinning; active contours; motion analysis and graph theory.

2.2.1 Image Smoothing and Enhancement

Heat flow has first been used for image smoothing. Witkin [25] introduced scale-space theory which involves generating coarser resolution images by convolving the original image with a Gaussian kernel. Then Koenderink [26] and Hummel [27] pointed out that the family of derived images may be equivalently viewed as the solution of heat conduction or diffusion equation based on several criteria: causality, homogeneity and isotropy. According to homogeneity and isotropy, blurring is required to be spatially invariant which makes it difficult to obtain accurately the location of edges at coarse scales. Then, Perona and Malik [8] introduced anisotropic heat flow for selective image smoothing that avoids blurring and localization problems of the edges. In this process, the diffusion coefficient is allowed to vary according to the magnitude of the local image gradient in four nearest direction. In this way, high quality edge detection is observed. After that, many approaches and models have been developed alternative to Perona and Malik's work. Some of them are: robust anisotropic diffusion [28], geometry driven heat flow [14], graph spectral model [29, 30], probabilistic view [31], regularization method [32], discrete image flux conduction model [33] and a global approach for solving evolutive heat transfer in images [34]. Recently, the geometrically stabilized reverse heat equation model has also been introduced for restoration of

blurred images [35]. The heat flow analogy is also used for vector field smoothing to recover surface normals and surface heights in shape-from-shading [36, 37, 38].

2.2.2 Region-based Image Segmentation

In [39], the anisotropic diffusion pyramid (ADP) was introduced for region-based segmentation. The pyramid is constructed using the scale space representation of anisotropic diffusion. Since anisotropic diffusion preserves edge locations as the scale increases, region boundaries in the coarse to fine ADP segmentation are accurately delineated. Recently, Manay and Yezzi [9] have proposed the anti-geometric heat flow model for adaptive thresholding and segmentation of regions. Here, anti-geometric heat flow is represented as diffusion through the normal direction of edges that smears rather than preserves them. As a result of this, regions on the opposite sides of prominent edges are captured in greyscale images.

2.2.3 Thinning

In [10], a new thinning algorithm was introduced based on time-reversed isotropic heat flow. Given an image, which is viewed as a thermal conductor, first the heat flow direction map is computed, then time-reversed heat conduction is simulated to get thinned a pattern. This algorithm can be applied to gray-scale or binary images.

2.2.4 Snake or Active Contours

Model-free active contours can be classified as Parametric Active Contours and Geometric Active Contours (Level Sets). The parametric active contour (PAC) is the first snake model, introduced by Kass et al. [40]. Problems associated with a PAC are initialization and poor convergence to concave regions. These problems were largely solved with the development of new external force model which is called Gradient Vector Field (GVF) [41]. It is computed as a diffusion of the gradient vectors of the grey level or binary edge map. This diffusion process arises from the heat conduction model. A geometric active contour (GAC) [11, 12] is based on a curve moving in normal direction with its curvature dependent on speed. This phenomenon is tackled with a level set method [13] in higher dimension by viewing the curve as the zero level set. The curve movement on the level set is achieved using geometric heat flow [14].

2.2.5 Motion Analysis

Horn and Schunk [42] developed a method for optical flow (velocity vectors) computation from sequence of images. The concept includes two constraints which are change of brightness and smoothness of the velocity flow. An equation was developed whose progress is similar to the propagation effects in the solution of heat conduction equation. However, this method does not preserve optical flow discontinuities on the motion boundary because of the isotropy property of the smoothness constraint. Then some extensions have modified the smoothness constraint and observe anisotropic behaviour to preserve motion boundaries [43, 44]. Makrogiannis and Bourbakis [15] were the first to propose a spatio-temporal anisotropic heat diffusion for motion activity measurement. The motion activity measure is derived from the total amount of diffusion in the spatio-temporal domain. Then, this process is completed by kernel based density estimation and watershed-based segmentation of regions.

2.2.6 Graph Theory

The heat analogy has also been integrated with graph theory in computer vision for the purpose of graph and shape clustering [45, 46, 47]. Xiao and Hancock [46] use the spectrum of the Laplacian matrix to capture graph structure. The Laplacian spectrum has close relationship with the heat equation, which can be used to specify the flow of information with time across a network. The solution to the heat equation, or heat kernel, is obtained by exponentiation of Laplacian eigensystem over time. The solution is related to the distribution of path lengths between nodes. Then the heat content, which is the sum of the entries of the heat kernel over the nodes of the graph, is expanded as a polynomial in time. Finally, graph-clustering is performed by applying principal components analysis to vectors constructed from the polynomial coefficients. In [48], the heat kernel is obtained, similar to the explained before, to evolve minimum spanning tree of a graph for clustering purpose. In [49], a new probabilistic relaxation labelling was developed using the heat flow on graphs.

2.3 Conclusions

In this Chapter, we have reviewed the basic concepts of heat flow and applications of heat flow in image processing and computer vision. It is observed that solution of the

heat equation depends on properties of material, heat source as well as specified initial and boundary conditions. The heat flow analogy has been applied in various ways in image processing and computer vision. Six applications were briefly surveyed here: image smoothing and enhancement; region-based image segmentation; thinning; active contours; motion analysis and graph theory.

An analytic solution of the linear and homogeneous heat conduction problem can be achieved using Fourier series theory. In addition, solving the linear and homogeneous heat equation in infinite medium can be viewed as smoothing a signal with a Gaussian filter. This relation also allows us investigate the Fourier series and Gaussian filtering for feature extraction. In Chapter 5, we introduce new and efficient silhouette object descriptors by using the Fourier series and Gaussian filtering for the purpose of recognition.

Analytic solution techniques for the heat conduction problems are limited, because it is difficult to represent initial and boundary conditions of geometrically complex shapes. Numerical techniques, such as finite differences, are able to handle all problems with any complex shape. In Chapter 4, we extract objects shapes by representing particular heat conduction problems in the image domain, and the proposed heat conduction problems are solved using finite difference technique. First, we solve a linear and non-homogeneous heat equation in the image domain to roughly segment objects of interest. Then anisotropic diffusion (geometric heat flow) is applied to smooth extracted boundaries and remove possible noise inside the prior segmented region.

In the next Chapter, we introduce a novel moving object edge detection technique. Here, we first solve anisotropic heat conduction problem in the spatial domain to remove noise and sharpen region boundaries. Then, linear and isotropic heat flow is applied in the temporal domain, with the proposed initial and boundary conditions, to calculate moving-edges. This problem is also solved using finite difference technique. As such, the heat equations described earlier are applied in a new way to detect the edges of moving object.

Chapter 3 Low Level Moving-Feature Extraction via Heat Flow Analogy

In this Chapter, a new intelligent and automatic moving object edge detection algorithm is proposed, based on using the heat flow analogy. This algorithm starts with anisotropic heat diffusion in the spatial domain, to remove noise and sharpen region boundaries for the purpose of obtaining high quality edge data. Then, isotropic and linear heat diffusion is applied in the temporal domain to calculate the total amount of heat flow. The moving-edges are represented as the total amount of heat flow out from the reference frame. The overall process is completed by non-maxima suppression and hysteresis thresholding to obtain binary moving-edges. Evaluation results indicate that this approach has advantages in handling noise in the temporal domain because of the averaging inherent of isotropic heat flow. Results also show that this technique can detect moving-edges in image sequences, without background image subtraction.

This Chapter is organized as follows: Section 3.1 explains related works. Section 3.2 discusses anisotropic heat flow for edge map enhancement. Section 3.3 introduces our novel moving-edge detection method. Section 3.4 concerns evaluation and experimental results, prior to conclusions.

3.1 Related Works

Segmenting moving objects is a challenging and important task in computer vision. It has many applications such as surveillance, video communication, traffic monitoring,

people tracking and content-based image coding. There are many moving object segmentation techniques, which are based on moving-edge detection.

Kim and Hwang [50] segment moving objects based on extracted moving-edges. Moving-edge points are generated based on processing the frames difference edge map, current frame edge map, and background edge map. These spatial domain edge maps are obtained using the Canny edge detector, which involves Gaussian convolution to suppress noise.

Ahn et al. [51] introduced a moving-edge detection algorithm to implement a home security system. Their algorithm extracts edge segments from the current image and eliminates the background edge segments by matching them with a reference edge list, where the reference edge list is generated from the set of training background images.

Myerscough and Nixon [52] developed a moving-edge detector by extracting feature points and feature velocities from a sequence of images based on using phase congruency. The benefits of using phase congruency are its illumination invariance and good localization. In addition, since phase congruency is illumination invariant, there is no need for threshold selection in edge detection.

Kim [53] proposed a moving-edge detection algorithm by using entropy and cross-entropy approaches. They calculate entropy in the current frame to determine spatial edges. Then, they apply cross-entropy between current and previous images to measure the dissimilarity. Finally, entropy and cross-entropy values are compared and thresholded to detect moving-edges.

Zhang and Zhao [54] proposed a moving-edge detection algorithm using wavelets to overcome noise. First, they find change detection mask in the wavelet domain. Then they apply Canny edge detection on the change detection mask to obtain the difference edge map. Finally, the difference edge map is compared with current frame edge map, background edge map and previous frame's moving-edges to obtain the current frame's moving-edges.

Recently, Dewan et al. [55] introduced an algorithm, which utilizes the three most recent consecutive frames to isolate moving-edges for moving object detection. They first calculate two difference images using those three consecutive frames. Then, the Canny edge detection algorithm is applied to generate difference edge maps. In the difference edge maps, edge pixels are grouped together to represent edge segments. Finally, since the moving-edges of the middle frame are common in both difference edge maps, they apply matching algorithm by using the size, shape and position features of the edge segments.

3.2 Anisotropic Heat Diffusion and Edge Enhancement

Perona and Malik [8], proposed anisotropic diffusion for selective image smoothing that avoids blurring and localization problems of the edges. The anisotropic heat diffusion problem is given below,

$$\begin{aligned} \frac{\partial \mathbf{I}}{\partial t} &= div\left(k(\mathbf{x},t)\nabla \mathbf{I}(\mathbf{x},t)\right) \\ \mathbf{I}(\mathbf{x},t=0) &= \mathbf{F}(\mathbf{x}), \quad \textit{initial condition} \\ \partial \mathbf{I}(\mathbf{x},t)/\partial n &= 0, \quad \textit{boundary condition} \end{aligned} \tag{3.1}$$

Where, *div* represents divergence operator, ∇ is a spatial gradient operator, $\mathbf{I}(\mathbf{x},t)$ is the grey level image that is a function of position $\mathbf{x} = (x, y)$ and time t, $k(\mathbf{x},t) \in [0,1]$ is the conductivity function, which depends on direction. The original image, $\mathbf{F}(\mathbf{x})$, indicates the initial condition. The boundary of the image is insulated, which is represented with homogeneous Neuman condition**.** In this boundary condition, there is no heat flow in or out of the image from the boundary. The solution to Equation 3.1 can be discretized using four nearest neighbours of the Laplacian operator as given below,

$$\mathbf{I}_{x,y}^{t+1} = \mathbf{I}_{x,y}^{t} + \lambda\left[k_N \cdot \nabla_N \mathbf{I} + k_S \cdot \nabla_S \mathbf{I} + k_E \cdot \nabla_E \mathbf{I} + k_W \cdot \nabla_W \mathbf{I}\right]_{x,y}^{t} \tag{3.2}$$

where, λ is a constant and $0 \le \lambda \le 0.25$ for the numerical scheme to be stable in the two-dimensional case [23]. ∇_N, ∇_S, ∇_E and ∇_W indicate nearest neighbor differences respectively in the direction north, south, east and west as shown below,

$$\nabla_N \mathbf{I}^t_{x,y} \equiv \mathbf{I}^t_{x,y-1} - \mathbf{I}^t_{x,y}$$
$$\nabla_S \mathbf{I}^t_{x,y} \equiv \mathbf{I}^t_{x,y+1} - \mathbf{I}^t_{x,y}$$
$$\nabla_E \mathbf{I}^t_{x,y} \equiv \mathbf{I}^t_{x-1,y} - \mathbf{I}^t_{x,y}$$
$$\nabla_W \mathbf{I}^t_{x,y} \equiv \mathbf{I}^t_{x+1,y} - \mathbf{I}^t_{x,y} \tag{3.3}$$

The nearest neighbour differences are calculated for each image pixel position (x, y) and at every iteration t. k_N, k_S, k_E and k_W are conductivity coefficients respectively in the north, south, east and west direction. The conductivity coefficients are also updated, for each position and at every iteration, as a function of the brightness gradient in each direction as given below,

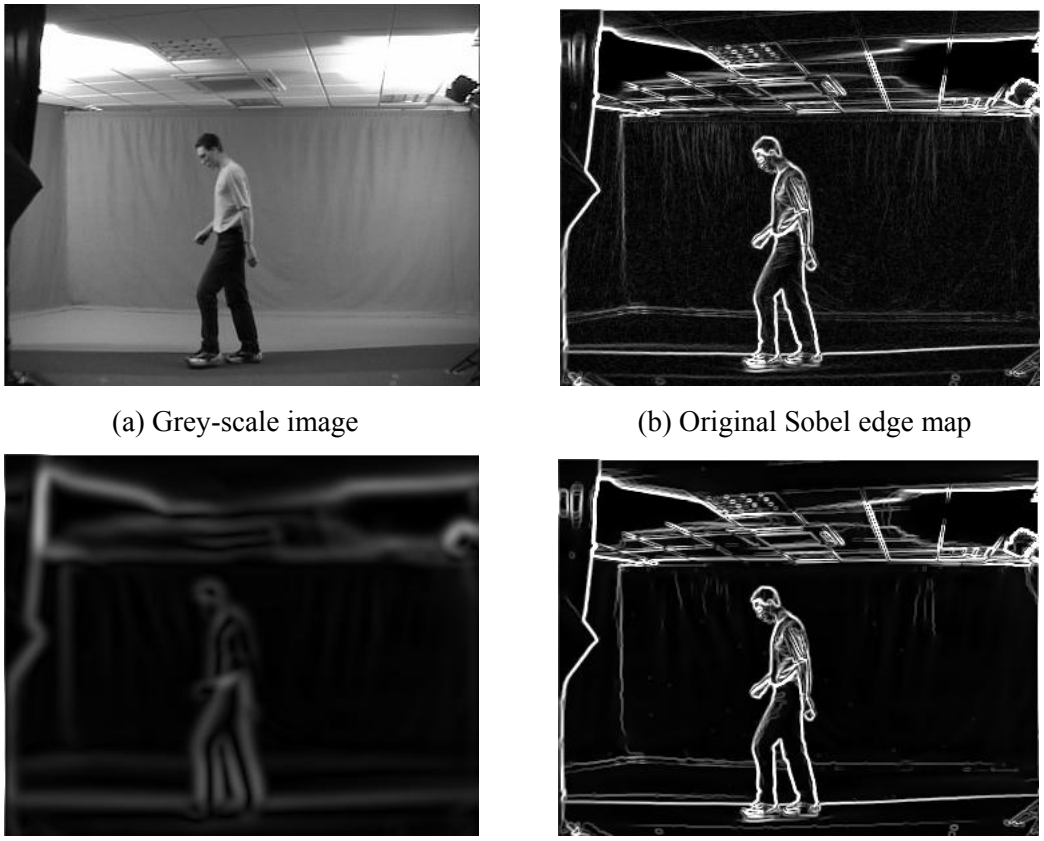
$$k^t_{N_{x,y}} = g\left(\left|\nabla_N \mathbf{I}^t_{x,y}\right|\right)$$
$$k^t_{S_{x,y}} = g\left(\left|\nabla_S \mathbf{I}^t_{x,y}\right|\right)$$
$$k^t_{E_{x,y}} = g\left(\left|\nabla_E \mathbf{I}^t_{x,y}\right|\right)$$
$$k^t_{W_{x,y}} = g\left(\left|\nabla_W \mathbf{I}^t_{x,y}\right|\right) \tag{3.4}$$

Different functions are used for $g\left(\left|\nabla \mathbf{I}\right|\right)$, depending on the chosen aim. In our application, an exponential type is used, (see Equation 3.5), which prefers high-contrast edges to low-contrast ones.

$$g\left(\nabla \mathbf{I}\right) = e^{-\left(\left|\nabla \mathbf{I}\right| / R\right)^2} \tag{3.5}$$

R determines the rate of decay of the exponential function, and thus the rate of smoothing. Note that, if $g\left(\left|\nabla \mathbf{I}\right|\right)$ is constant (at all image locations), this leads to isotropic heat diffusion. In Figure 3.1, we illustrate the difference between isotropic and anisotropic diffusion operations. Figure 3.1(a) is a grey-scale image and Figure 3.1(b) is its Sobel edge map without any diffusion. Figure 3.1(c) is the Sobel edge map after isotropic diffusion, which causes loss of edge information. On the other hand, Figure 3.1(d) is the Sobel edge map of the anisotropic diffused image with the conductivity function given by Equation 3.5 and it can easily be observed that high contrast edges

are enhanced while removing-edges due to noise, and thus important detail is preserved.



(a) Grey-scale image (b) Original Sobel edge map

(c) Sobel edge map after isotropic diff. (d) Sobel edge map after anisotropic diff.

Figure 3.1: Difference between isotropic and anisotropic diffusion.

3.3 Isotropic Heat Flow in Temporal Domain

Here, we introduce a novel moving-edge detection technique. Assume that we have three or more consecutive frames. Once the enhanced Sobel edge maps are obtained in space, the isotropic and linear heat equation is applied in the temporal domain to calculate the total amount of heat flow. Assuming we have L consecutive frames, the discrete formulation of the proposed heat conduction problem is given below,

$$\mathbf{E}_n^t = \mathbf{E}_n^{t-1} + \alpha \Delta \mathbf{E}_n^{t-1} = \mathbf{E}_n^{t-1} + \alpha(\mathbf{E}_{n+1}^{t-1} + \mathbf{E}_{n-1}^{t-1} - 2\mathbf{E}_n^{t-1})$$

$$\mathbf{E}_n^{t=0} = \mathbf{F}_n \qquad \textit{Initial condition}$$

$$\mathbf{E}_{n=0}^t = \mathbf{E}_0^{t=0}, \quad \mathbf{E}_{n=L-1}^t = \mathbf{E}_{L-1}^{t=0} \qquad \textit{Boundary Conditions} \tag{3.6}$$

Where, $\mathbf{E}_n^{t-1}$, $\mathbf{E}_{n-1}^{t-1}$ and $\mathbf{E}_{n+1}^{t-1}$ are Sobel edge mapped images respectively for the reference frame (n), the previous frame $(n-1)$ and the next frame $(n+1)$ at iteration t. Δ is Laplacian operator. α is thermal diffusivity and $0 \le \alpha \le 0.5$ for the numerical scheme to be stable in the one-dimensional case [23]. The original image sequences, $\mathbf{F}_n$, indicates the initial condition and the proposed boundary condition is Dirichlet with constant temperatures. In this boundary condition, the boundary frames always remain at the initial values and there will be heat flow in and out of the image sequences. Since we assume there are moving objects in the consecutive images, moving-edge positions in any interior frame are expected to have higher brightness than the same positions at the boundary frames. As a result of this, moving-edges diffuse out from the reference frame to the neighbour frames and then out of image sequences from the boundary frames.

The diffusion (iteration) stops, when we reach the steady-state solution in the system, when there is no more heat flow. The total amount of heat flow, for each frame, is calculated as follows. Assume that the initial scale is 0(zero) and final scale is t, then Equation 3.6 can be described as,

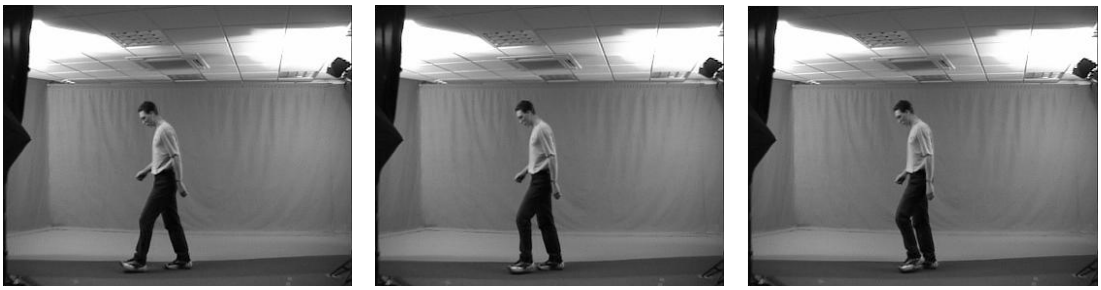
$$\mathbf{E}_n^t = \mathbf{E}_n^0 + \alpha \sum_{i=0}^{t-1} \Delta \mathbf{E}_n^i \tag{3.7}$$

Then, the total amount of heat flow from the initial state to the steady-state is

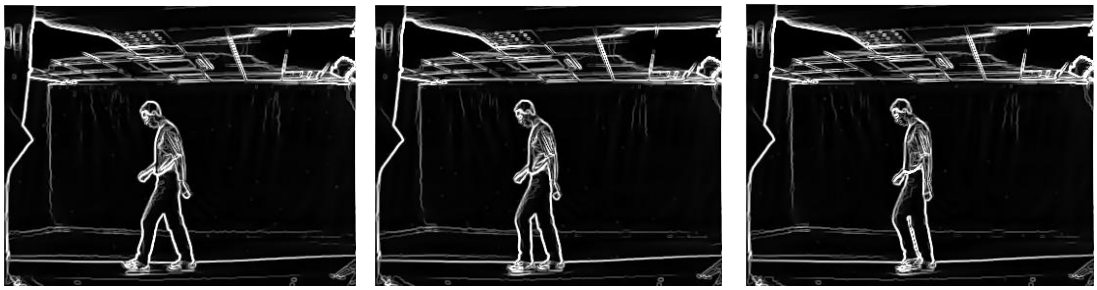
$$\left| \mathbf{E}_n^t - \mathbf{E}_n^0 \right| = \alpha \sum_{i=0}^{t-1} \left| \Delta \mathbf{E}_n^i \right| \tag{3.8}$$

However, this gives us total *heat in* (+) and *heat out* (-) together during diffusion. We are interested in total *heat flow out* $(\mathbf{HFO})$ from the reference frame, $\mathbf{E}_n$, which gives us the moving-edge map. This is obtained as,

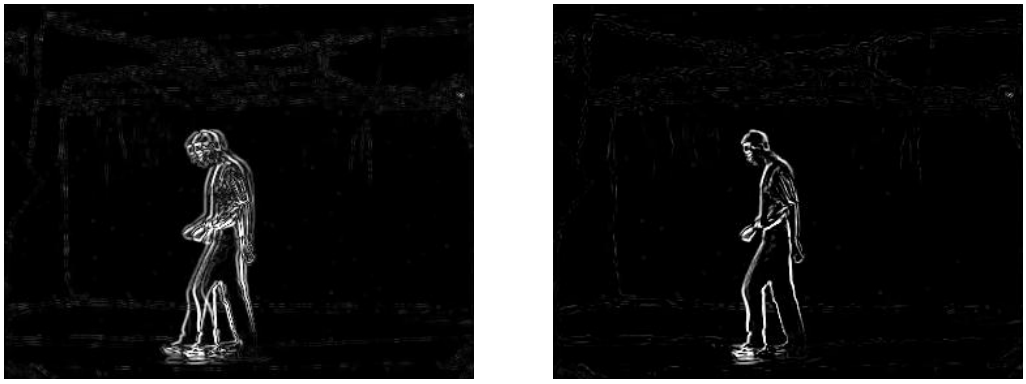
$$\mathbf{HFO} = \alpha \sum_{i=0}^{t-1} \left| \Delta \mathbf{E}_n^i \right| \ , \quad \forall \Delta \mathbf{E}_n^i < 0 \tag{3.9}$$



(a) $Frame_{n+1}$ (b) $Frame_n$ (c) $Frame_{n-1}$



(d) $\mathbf{E}_{n+1}$ (e) $\mathbf{E}_n$ (f) $\mathbf{E}_{n-1}$



(g) Total heat flow (h) Heat flow out $(\mathbf{HFO})$

Figure 3.2: Moving-edge map extraction.

Figure 3.2 illustrates the process for three consecutive frames, where Figures 3.2 (a-c) show a walking human subject. First, anisotropic diffusion is applied in space with parameters values $\lambda = 0.2$, $R = 4$ and for 50 iterations. Figures 3.2(d-f) show enhanced Sobel edge maps of these frames after anisotropic diffusion. Then, isotropic diffusion is applied in temporal domain to calculate total amount of heat flow. This diffusion continues until we observe steady-state with $\alpha = 0.45$, which needs 10 iterations in this experiment. Figure 3.2(g) shows total amount of heat flow in the reference frame, $\mathbf{E}_n$, and Figure 3.2(h) is total heat flow out from $\mathbf{E}_n$, which gives the moving-edge map. Only the moving-edges of the human subject and some slight shadows remain, whilst largely removing the edges introduced by the static background.

The overall process is completed by non-maxima suppression (thinning) and hysteresis thresholding to observe the binary moving-edges. The non-maxima suppressed and hysteresis thresholded images are shown respectively in Figures 3.3(a) and (b). Non-maxima suppression has been developed by Canny [56]. In this approach, the edge point is defined to be a point whose strength is locally maximum in the direction of gradient. Non-maxima suppression ends up with an image settled zero except local maxima points. Local maxima points preserve their values. Hystresis thresholding involves two thresholds, *upper* (T_H) and *lower* (T_L), to convert the image into binary form. The pixels with a value higher then T_H are set to one, the pixels with value between T_H and T_L are set to one, if they are connected to the pixels higher than T_H. The pixels with value lower than T_L are zero.

a) Non-maxima suppressed image (b) Hysteresis thresholded image

Figure 3.3: Binary moving-edge observation.

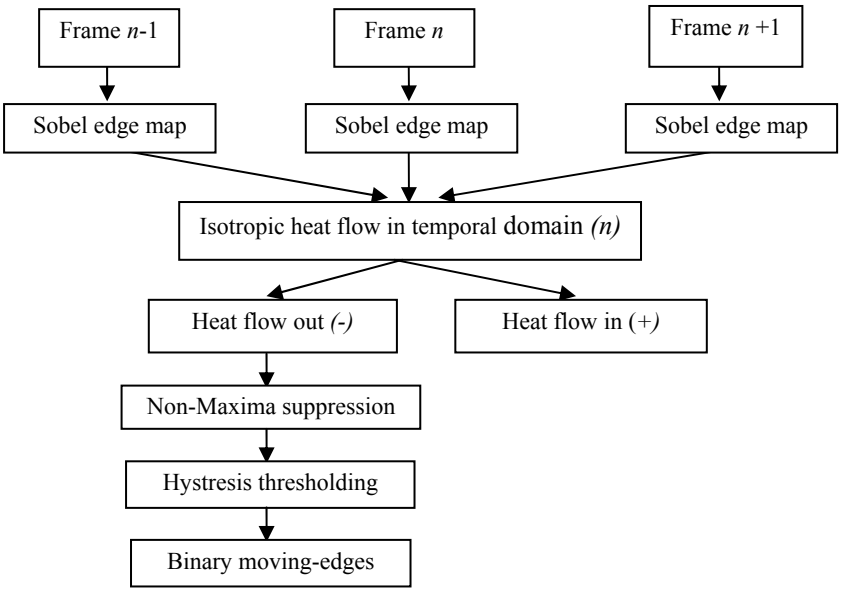


Figure 3.4: Moving-edge detection

3.4 Evaluation and Experimental Results

Performance evaluation is employed by comparing moving-edge detection with 2-D Sobel edge detection. However, anisotropic heat diffusion in the spatial domain is omitted in our algorithm to balance the 2-D Sobel and moving-edge detection algorithms. Figure 3.4 shows the proposed algorithm to obtain the moving-edges in the evaluation. Evaluation is performed using a white circle moving on a black background with varying normally distributed noise $N_d\left(\mu,\sigma^2\right)$. The Hough Transform (HT) is applied to the binary edge images to extract the circle centre parameters. A root mean square error (RMSE) is then employed to quantify the performance of each algorithm.

$$RMSE = \sqrt{\left(\left(e_x - c_x\right)^2 + \left(e_y - c_y\right)^2\right)/2} \tag{3.10}$$

Where, $\left(e_x, e_y\right)$ are the extracted circle centre parameters and $\left(c_x, c_y\right)$ are the actual circle centre parameters. The quantity of noise is considered in terms of standard deviation σ with zero mean. Hysteresis thresholding is used both for 2-D Sobel and moving-edges, after non-maxima suppression (thinning). The thresholds for the 2-D Sobel, to obtain the binary image, are determined by a root mean square (RMS) estimate of the noise. In this process, the gradient magnitude image is thresholded by

its scaled mean value that is proportional to signal to noise ratio (SNR). The upper and the lower thresholds are determined as given below,

$$T_H = s_T \times \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} \mathbf{F}_{x,y} , \qquad T_L = T_H / 4 \tag{3.11}$$

Where, $\mathbf{F}$ is the gradient magnitude reference frame of size $M \times N$ and s_T is a scale factor for threshold selection, which is a positive constant with value 3 in this evaluation. The scale factor, s_T, was determined experimentally to achieve the best performance of the algorithm. The ratio between high, T_H, and low, T_L, thresholds is 4. On the other hand, the thresholds of the moving-edge detection algorithm are based on mean heat flow out $(\mathbf{HFO})$ from the reference frame,

$$T_H = c_T \times \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} \mathbf{HFO}_{x,y} , \qquad T_L = T_H / 4 \tag{3.12}$$

Where, c_T is a scale factor for threshold selection with value 11 and the ratio between the high, T_H, and the low, T_L, thresholds is again 4. Here, the scale factor, c_T, was also determined experimentally to achieve the best performance of the proposed algorithm. Figure 3.5 shows performance of moving-edge detection and 2-D Sobel algorithms. The graphs show error bars representing the mean and standard deviation obtained over 5 applications of moving-edge detection and 2-D Sobel operator. In each application, the obtained graphs are smoothed by applying local averaging to the data points in 6 nearest neighbourhoods. RMSE of the 2-D Sobel increases slowly until $\sigma \cong 130$ and after that we observe rapid increase in RMSE. On the other hand, RMSE of the moving-edges increases slowly until $\sigma \cong 120$ and after that it increases rapidly. It is observed that, the moving-edge detection technique has better performance than 2-D Sobel at all noise levels, which appears due to the averaging inherent in the new operator. Figure 3.6 shows some of the results for moving-edges (second row) and 2-D Sobel (third row). To visual inspection, the input images in Figure 3.6(c) and (d), are very noisy indeed.

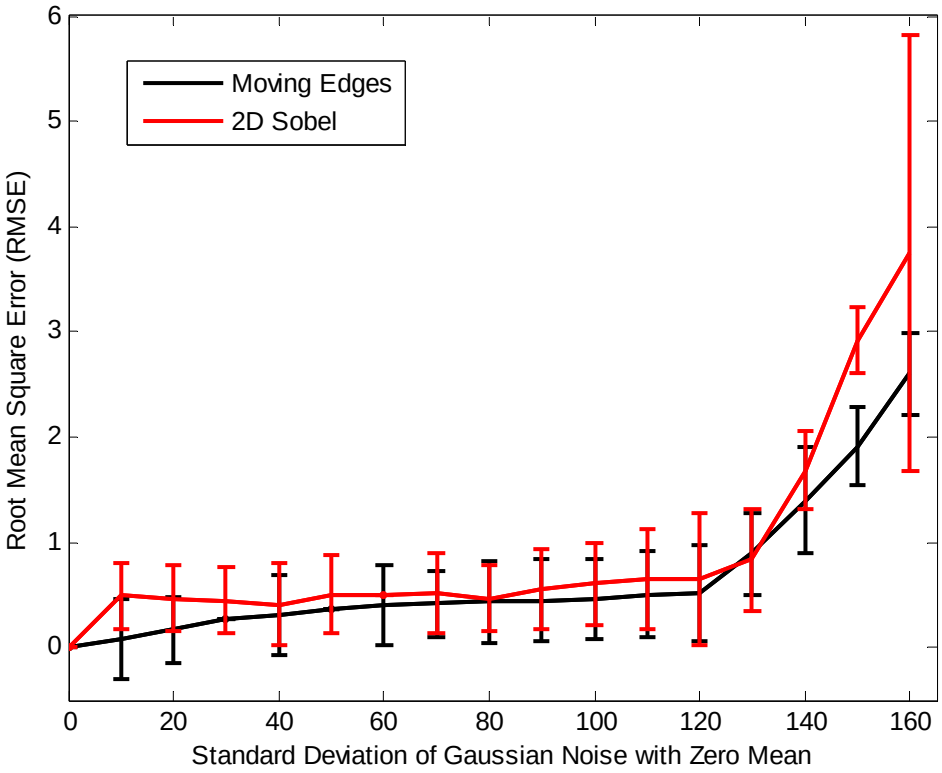


Figure 3.5: Performances of moving-edges and 2-D Sobel with respect to normal distributed noise trials. RMSE is plotted with mean and standard deviation values using error bars.

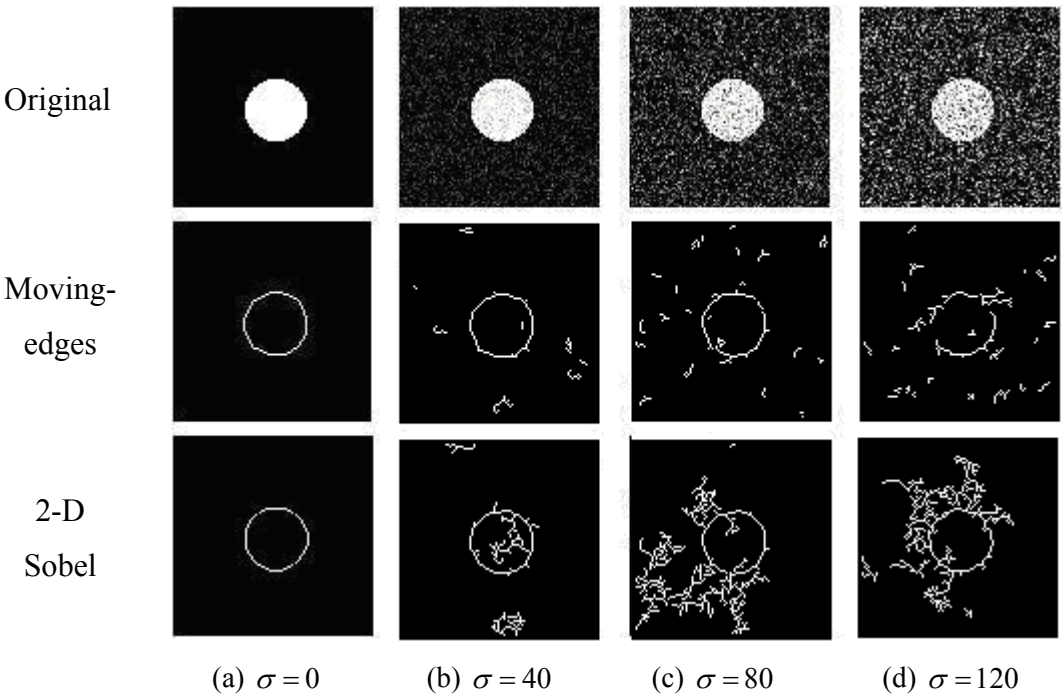


Figure 3.6: Results for moving-edges (second row) and 2-D Sobel (third row) with respect to increasing Gaussian noise.

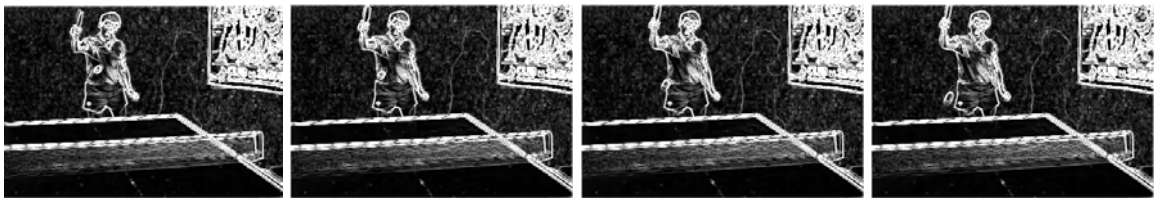
Simulation results also show that our algorithm can detect moving-edges in image sequences, as shown in Figure 3.7. Figure 3.7(a) has eight consecutive frames and shows a human subject, while playing table tennis. Figure 3.7(b) shows Sobel edge maps, $\mathbf{E}$, after the anisotropic diffusion in space. The selected parameters for anisotropic diffusion: $\lambda = 0.2$, $R = 4$ and the number of iterations is 20. It is observed that region boundaries are sharpened after anisotropic diffusion. Although almost all noise has been removed on the table, the wall behind the player still appears noisy. The reason for this is the textured structure of the wall and there are strong edges as well. The textured structure of the wall can bee seen in Figure 3.8(a), since this image is enlarged and also belongs to the same table tennis sequence. Figure 3.7(c) is the total amount of heat flow out $(\mathbf{HFO})$ from each frame, after isotropic diffusion in the temporal domain. This diffusion needs 215 iterations with $\alpha = 0.45$ to achieve steady-state. Here, the first and eighth frames are omitted since they are the boundary frames and their edge map values are constant, during the diffusion, to achieve the given Dirichlet boundary conditions. It is observed that almost all edges introduced by the static background and the noise arising from the textured wall is removed, while retaining the moving-edges of the human subject and the tennis ball. There are also slight edges of the human's shadow in the **HFO** images. Figure 3.7(d) shows the binary moving-edges at each reference frame after applying non-maxima suppression and hysteresis thresholding. It is seen that moving-edges were indeed detected.

Figure 3.8 also shows some simulation results. Figures 3.8(a) shows the reference frame from table tennis (indoor) sequence and moving-edges of the arm and of the ball were detected. We should note that the upper part of the arm and the table are static and this is why they were not detected. Figures 3.8(b) is a reference frame from the flower garden (outdoor) sequence, where the camera is in motion. It is seen again that most of the moving-edges were detected. Some edges, such as part of the house roofs, were not detected, since the camera was moving along those edges.

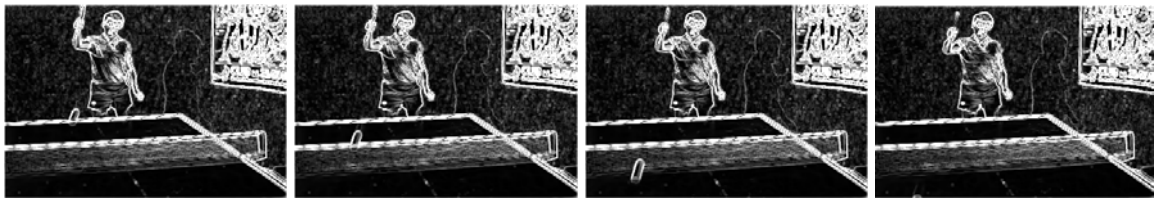
$frame_1$ $frame_2$ $frame_3$ $frame_4$

$frame_5$ $frame_6$ $frame_7$ $frame_8$

(a) Input frames

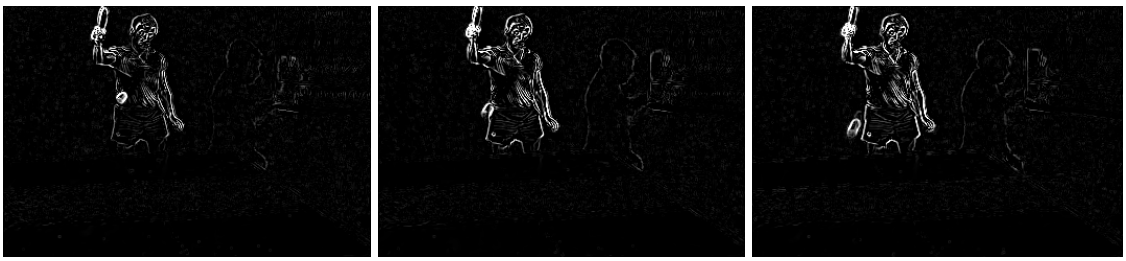


$\mathbf{E}_1$ $\mathbf{E}_2$ $\mathbf{E}_3$ $\mathbf{E}_4$

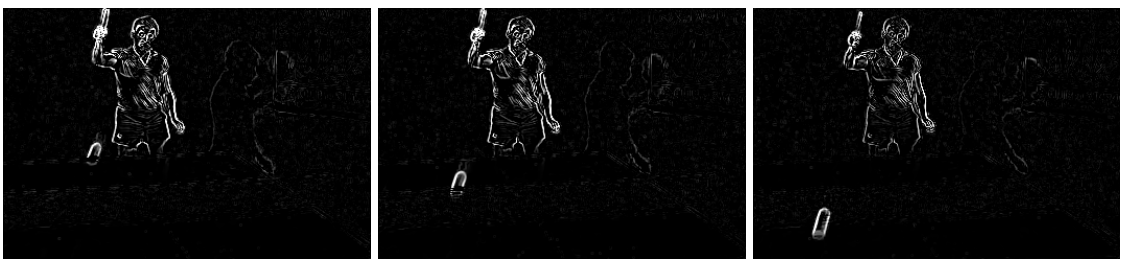


$\mathbf{E}_5$ $\mathbf{E}_6$ $\mathbf{E}_7$ $\mathbf{E}_8$

(b) Enhanced Sobel edge maps after anisotropic diffusion



$\mathbf{HFO}_2$ $\mathbf{HFO}_3$ $\mathbf{HFO}_4$



$\mathbf{HFO}_5$ $\mathbf{HFO}_6$ $\mathbf{HFO}_7$

(c) Heat flow out (-) from reference frames until the steady-state

$Binary\_Edge_2$ $Binary\_Edge_3$ $Binary\_Edge_4$

$Binary\_Edge_5$ $Binary\_Edge_6$ $Binary\_Edge_7$

(d) Binary moving-edges after non-maxima suppression and hysteresis thresholding

Figure 3.7: Moving-edge detection in the eight consecutive frames

(a) Table tennis (indoor) image

(b) Flower garden (outdoor) image, where the camera is in motion.

Figure 3.8: Some of the simulation results for new operator on indoor and outdoor images.

3.5 Conclusions

We have presented a novel low level moving-feature extraction technique based on using the heat flow analogy. Firstly, high quality Sobel edge maps are obtained based on anisotropic heat diffusion, in space. The conductivity function is the key point in this stage; we have chosen an exponential function which enhances high contrast edges and removes edges due to noise. In the next stage, isotropic and linear heat diffusion is applied in the temporal domain to determine the moving-edge map in the reference frame. To do this, the total amount of heat flow is calculated and then separated into the heat in (+) and heat out (-) parts, where the heat out (-) is the moving-edge map. Finally, non-maxima suppression and hysteresis thresholding is applied to obtain binary moving-edges. Evaluation indicates that this technique is better than 2-D Sobel at all levels of noise corrupted image, without anisotropic heat diffusion in space. This result appears to be due to the averaging inherent in the new operator. Results also show that this technique can detect moving-edges in image sequences, without using the background detection.

Chapter 4 Shape Extraction via Heat Flow Analogy

In this Chapter, we introduce a novel evolution-based segmentation algorithm by using the heat flow analogy to gain practical advantage. The proposed algorithm consists of two parts. In the first part, we represent a particular heat conduction problem in the image domain to roughly segment the region of interest. Then we use geometric heat flow to complete the segmentation, by smoothing extracted boundaries and removing noise inside the prior segmented region. The proposed algorithm is compared with active contour models and is tested on synthetic and medical images. Experimental results indicate that our approach works well in noisy conditions without pre-processing. It can detect multiple objects simultaneously. It is also computationally more efficient and easier to control and implement in comparison with active contour models.

The Chapter is organized as follows: Section 3.1 discusses related works. Section 3.2 represents the proposed heat conduction problem in the image domain, which is the first part of our algorithm. Section 3.3 explains the geometric heat flow. Section 3.4 concerns evaluation and experimental results and finally Section 3.5 is conclusions.

4.1 Related Works

There are two main types of shape extraction method that evolve to the target solution: active contours and region growing techniques.

4.1.1 Active Contours

In this section, we summarize the existing active contour models with their advantages and disadvantages. Extensive explanations of the commonly used active contour models are also given in Appendix B. Active contours (snakes) are curves that evolve to recover object shapes. Active contours can be classified as Parametric Active Contours (PAC) and Geometric Active Contours (GAC) according to their representation. These models mainly differ in their ability to handle multiple object detection, computational efficiency and complexity of implementation.

Parametric Active Contours (PAC)

A PAC is a parametric curve which is represented explicitly. There are internal and external forces acting on curve. The internal forces control the smoothness and rigidity of the curve with respect to selected parameters. The external force is image dependent and attracts the curve to the object boundary. The first PAC model was introduced by Kass et al. [40]. In this, segmentation is achieved by using gradient vectors of an edge map as an external force. Problems associated with this model are initialization and poor convergence to concave regions. A constant force, which is called the balloon force [57], was added with direction normal to curve, to accelerate the motion so that the initial curve can be placed far away from the desired object boundary. However, choice of the balloon force is a problem. If the balloon force is large, the contour can pass through weak object boundaries. If the balloon force is small, the contour may not move on narrow regions of the object. Initialization and convergence to concave region problems were largely solved with the development of a new external force model, which is called Gradient Vector Flow (GVF) [41]. GVF is computed as a diffusion of the gradient vectors of an edge map. However, PAC models can have difficulty with simultaneous detection of multiple objects, since they cannot handle the topological changes such as merging or splitting of the moving curve, because of the explicit representation. To solve this problem, GAC models have been introduced, where the curve is represented implicitly in a level set function (the zero set).

Geometric Active Contours (GAC)

GAC use the level set method [13] for curve evolution. A level set is a real valued function of the two-dimensional space variables. The contour is represented implicitly in a level set function, which is the zero set. Caselles et al. [11] and Malladi et al. [12] proposed the first GAC model, which uses gradient-based information for segmentation. In this model, the curve evolves in the direction normal with its curvature and balloon force dependent speed, and stops depending on the gradient information obtained from the image. The curvature dependent speed has the effect of smoothing the curve, while the balloon force accelerates the motion. The gradient-based GAC can detect multiple objects simultaneously but it has other important problems, which are boundary leakage, noise sensitivity, computational inefficiency and complexity of implementation. Some formulations, [58] [59], have been introduced to solve problems with boundary leakage and noise sensitivity by improving gradient-based information. However, they can just increase the tolerance, since gradient-based information is always limited by noise. Xie and Mirmehdi [60] apply region segmentation to the original image and then find the Gradient Vector field (GVF) of the segmented image. This force is added to the GAC, but it can only provide more tolerance to toward weak edges. Several numerical schemes have been proposed to improve the computational efficiency of the level set method, including narrow band [61], fast marching [62] and additive operator splitting [63]. Despite substantial improvements in efficiency, they can be difficult to implement.

Chen and Vese [64] introduced a new GAC model without using edge information. Their model is a particular case of the Mumford-Shah functional [65] and uses regional statistics for segmentation. In this work, the image is divided into two regions, interior and exterior, separated by a curve. The model minimizes the sum of the squares of the differences between the intensity values and the mean of the intensity values inside and outside of the surface of desired object. Their approach especially works well for bimodal images. It is good at handling initialization, noise and boundary leakage but still suffers from computational complexity and difficulty in implementation, because of the level set method. Later, Chen and Vese [66] extended this approach to the multiphase level set framework to segment more than two-phase (binary segmentation).

However, they need more level set functions for this process, which in turns increases complexity and difficulty.

Different types of image forces can be combined with PAC or GAC (level set) models to overcome limitations and improve segmentation. In [67], the Mumford-Shah functional [65] based region force was adapted by PAC to handle noise and initialization problems. Paragios and Deriche [68] unified boundary- and region-based forces and implemented using level sets to improve segmentation and solve initialization and noise problems. Recently, Xie and Mirmehdi [69] proposed a magnetic force based on magnetostatic theory using level sets to handle problems with initialization and convergence to concave regions.

4.1.2 Region Growing

Region growing is a procedure that groups pixels or sub-regions into larger regions based on predefined similarity criteria for region growth. The basic approach starts with a seed point and merges neighboring pixels that have pre-defined properties similar to the seed, such as intensity [70] or texture [71]. Region growing was also combined with edge detection for segmentation [72]. Although, region growing techniques can detect multiple objects simultaneously and can be more efficient than active contour models, the main problem is selection of the similarity criteria. They also have to use connectivity information to define the neighboring pixels in each step of growth. In addition, they can achieve region segmentation with irregular boundaries and holes in the presence of high noise, since they omit smoothing.

4.2 Proposed Heat Conduction Problem and Representation in Image Domain

Consider a two-dimensional conductive solid body that has uniform conductivity and is an isotropic medium. Initial and boundary conditions respectively given by $T(\mathbf{x},t=0)=0$ and $T(\mathbf{x},t)=0$, where T represents the temperature at position $\mathbf{x}=(x,y)$ and time t. The given conditions mean that the temperature is initially zero inside the body and the boundary condition is Dirichlet that has a specified temperature (zero) at the boundaries. If we initialize a continuous heat source, which is a positive

constant, at a point inside the body, there will be heat diffusion to the other points from the source position. As a result of this, all the points inside the body will have temperature values exceeding zero, except the boundary points. This is then an ideal approach for object segmentation in computer images. Let us investigate the proposed problem on a square object that is inside the grey-level image $(\mathbf{G})$, as shown in Figure 4.1(a). Assume that all the temperature values of the objects and the background are stored in another image, which is represented by $\mathbf{I}$, and the initial condition of whole image is zero, $\mathbf{I}(\mathbf{x}, t=0)=0$. This assumption means that all objects have temperature initially zero inside, as well as at the boundaries. When we initialize a heat source at any pixel inside the square object, as shown in Figure 4.1(a), there will be heat diffusion to the other pixels from the source position, which will cause temperature to increase. However the temperature at the boundary layer must be kept at zero all the time to obtain the Dirichlet condition, where the boundary layer is defined at the external side of an object as shown in Figure 4.1(b). To achieve this, we use a control function in the heat conduction equation as given below,

$$\frac{\partial \mathbf{I}(\mathbf{x},t)}{\partial t} = \mathbf{CF}(\mathbf{x},t)(\alpha \Delta \mathbf{I}(\mathbf{x},t)) + \mathbf{Q}(\mathbf{x}) \tag{4.1}$$

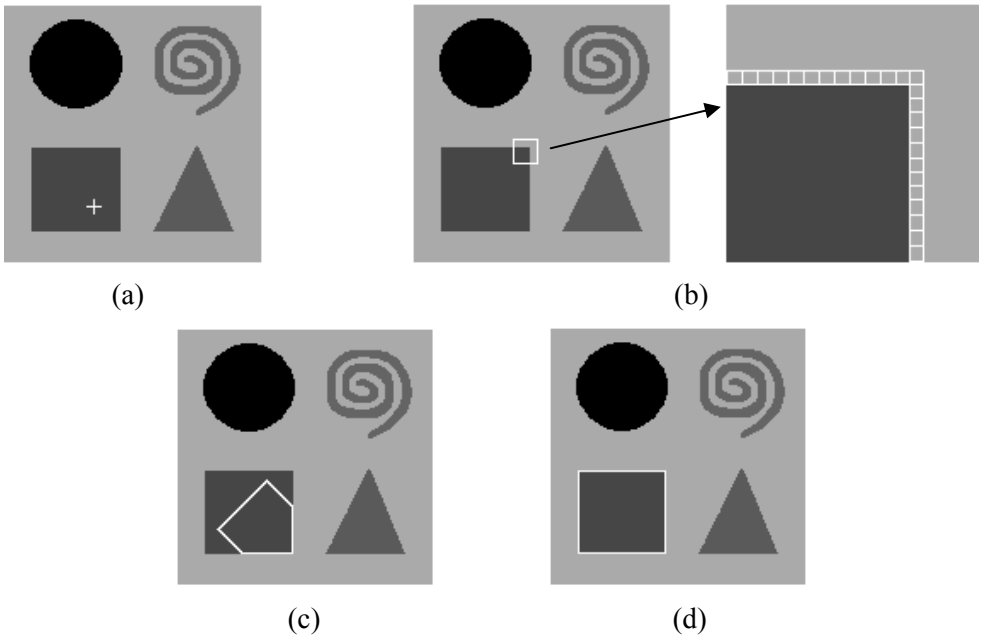


Figure 4.1: Heat conduction modeling in image domain of size 150×150. (a) Source position at $t=0$. (b) Boundary layer illustration. (c) TF at $t=30$ (iterations). (d) Final TF at $t=69$.

where $\mathbf{I}(\mathbf{x},t)$ represents an image pixel value in terms of temperature at each point and time, α is the thermal diffusivity and $0 \le \alpha \le 0.25$ for the numerical scheme to be stable in two-dimensional system [23], Δ is Laplace operator for the spatial variables, $\mathbf{Q}(\mathbf{x})$ is the source term and $\mathbf{CF}(\mathbf{x},t)$ is the control function. The control function is obtained from the region statistics of source location on a given grey-level image. The proposed region statistics model is similar to the one used by Chen and Vese [64]. In this model, the image is divided into two regions, interior and exterior, separated by a contour and the model minimizes the sum of the squares of the differences between the intensity values and the mean of the intensity values inside and outside of the surface of desired object. In our model, the contour is represented by a Temperature Front (TF), where the TF is the boundary of the region that has temperature values exceeding zero. The control function, $\mathbf{CF}(\mathbf{x},t)$, is formulated as follows,

$$\sigma_{in}(\mathbf{x},t) = \lambda_{in}\left|\mathbf{G}(\mathbf{x}) - \mu_{in}\right|^2 \tag{4.2}$$

$$\sigma_{out}(\mathbf{x},t) = \lambda_{out}\left|\mathbf{G}(\mathbf{x}) - \mu_{out}\right|^2 \tag{4.3}$$

where, $\mathbf{G}(\mathbf{x})$ is the given grey-level image, $\sigma_{in}(\mathbf{x},t)$ is the weighted square of the difference between the intensity value and the mean, μ_{in}, of the intensity values inside the TF, at each point and time. $\sigma_{out}(\mathbf{x},t)$ is the weighted square of the difference between the intensity value and the mean, μ_{out}, of the intensity values outside the TF, at each point and time. $\lambda_{in} > 0$ and $\lambda_{out} > 0$ are fixed parameters (constants) for regional statistics inside and outside the TF respectively. Then, the following logical decision is applied at each position and at each time interval.

$$\mathbf{CF}(\mathbf{x},t) = \begin{cases} 1, & \sigma_{in}(\mathbf{x},t) \le \sigma_{out}(\mathbf{x},t) \\ 0, & otherwise \end{cases} \tag{4.4}$$

Therefore, the control function allows heat diffusion inside the object of interest and achieves the proposed Dirichlet condition on the boundary layer by keeping the temperature value at zero. However, it is better to start this process after a short diffusion time by assuming $\mathbf{CF}(\mathbf{x},t) = 1$ at all points. This increases the effective area of initialization, thereby better handling noisy images. In addition, the heat source must

be initialized onto a smooth surface of the object, since the source localization to the edge pixel will give the wrong region statistic for our purpose. Figure 4.1(c) and (d) respectively show the evolution and the final position of the TF. However, there is no need to continue diffusion, after the TF reaches its final position. For this reason, the position of the TF is controlled at each specified time interval and when there is no movement, diffusion is terminated. Here, we also note that the regional statistics parameters will be represented with their ratio $\lambda_{in}/\lambda_{out}$ in our experiments, and in this experiment (Figure 4.1) $\lambda_{in}/\lambda_{out}$=1. The analytical solution of the proposed heat conduction problem, on the square object shown in Figure 4.1, is given in Appendix A. Comparison of analytical and image domain results are also included in Appendix A. The main difference between [64] and our model, in using region statistics, we attempt to segment the region containing the source instead of whole image.

One difficulty arises when the region containing the source intersects the image boundary. This problem can be solved by assuming that image is surrounded by a boundary layer, at its periphery, which has temperature value zero for all time (Dirichlet). Figure 4.2 shows the evolution and the final position of the TF, which has a source location within the background. The result, in Figure 4.2, also shows that multiple object detection can be achieved and the heat can diffuse through the narrow regions within the spiral object. As such, heat has been used to detect the background and therefore boundaries of objects.

(a) Source position (b) $t = 110$ (c) $t = 221$ (final)

Figure 4.2: TF is moving on background in the image of size 150×150, and $\lambda_{in}/\lambda_{out} = 1$.

It is also required to consider the control function when the given image is bimodal as shown in Figure 4.3(a). In this case, the control function attempts to segment the whole image while the TF segments the source located region. This is because, the control function assigns unity to the pixels that are similar to the inside of the TF, and assigns

zero to the pixels that are dissimilar. The process is illustrated in Figure 4.3. Figure 4.3(a) shows the source position, Figure 4.3(b) is the final position of the TF and Figure 4.3(c) is the control function at the end of the diffusion.

(a) Source position (b) Final TF at $t = 69$ (c) Final $\mathbf{CF}(\mathbf{x})$

Figure 4.3: Illustration of the control function, $\mathbf{CF}(\mathbf{x})$, in a bimodal image of size 150×150, and $\lambda_{in}/\lambda_{out} = 1$.

All the results so far have been on synthetic images without added noise. If we simulate this algorithm on noisy medical images, such as a magnetic resonance image of the left ventricle of a human heart shown in Figure 4.4(a) with the heat source location, we observe some drawbacks in segmentation. The drawbacks are irregular boundaries and holes inside the segmented region, as shown in Figure 4.4(b), where the $\lambda_{in}/\lambda_{out} = 1$ in this experiment. These problems are solved by using the heat flow analogy again as described in the next section.

4.3 Geometric Heat Flow

Geometric Heat Flow (GHF) is a kind of anisotropic diffusion and is widely used for image denoising and enhancement [14]. It diffuses along the boundaries of image features, but not across them. It derives its name from the fact that, under this flow, the feature boundaries of the image evolve in the normal direction in proportion to their curvature. Thus GHF decreases the curvature of shapes while removing noise. GHF equation is obtained with the following considerations.

Edge directions are related to the tangents of the feature boundaries of an image $\mathbf{B}$. Let η denote the direction normal to the feature boundary through a given point (the gradient direction), and let τ denote the tangent direction. These directions can be written in terms of the first derivatives of the image, $\mathbf{B}_x$ and $\mathbf{B}_y$, as

$$\eta = \frac{\left(\mathbf{B}_x, \mathbf{B}_y\right)}{\sqrt{\mathbf{B}_x^2 + \mathbf{B}_y^2}}, \qquad \tau = \frac{\left(-\mathbf{B}_y, \mathbf{B}_x\right)}{\sqrt{\mathbf{B}_x^2 + \mathbf{B}_y^2}} \tag{4.5}$$

Since η and τ constitute orthogonal directions, the rotationally invariant Laplacian operator can be expressed as the sum of the second order spatial derivatives, $\mathbf{B}_{\eta\eta}$ and $\mathbf{B}_{\tau\tau}$, in these directions and the linear heat conduction equation can be written without using the source term,

$$\frac{\partial \mathbf{B}}{\partial t} = \alpha \Delta \mathbf{B} = \alpha\left(\mathbf{B}_{\eta\eta} + \mathbf{B}_{\tau\tau}\right) \tag{4.6}$$

Omitting the normal diffusion, while keeping the tangential diffusion yields the GHF equation as

$$\frac{\partial \mathbf{B}}{\partial t} = \alpha \mathbf{B}_{\tau\tau} = \alpha \frac{\left(\mathbf{B}_{xx}\mathbf{B}_y^2 - 2\mathbf{B}_{xy}\mathbf{B}_x\mathbf{B}_y + \mathbf{B}_{yy}\mathbf{B}_x^2\right)}{\left(\mathbf{B}_x^2 + \mathbf{B}_y^2\right)} \tag{4.7}$$

The equation above can also be written as follows to show that feature boundaries of the image evolve in the direction normal in proportion to their curvature κ,

$$\frac{\partial \mathbf{B}}{\partial t} = \alpha \mathbf{B}_{\tau\tau} = \alpha \frac{\left(\mathbf{B}_{xx}\mathbf{B}_y^2 - 2\mathbf{B}_{xy}\mathbf{B}_x\mathbf{B}_y + \mathbf{B}_{yy}\mathbf{B}_x^2\right)}{\left(\mathbf{B}_x^2 + \mathbf{B}_y^2\right)^{3/2}} \sqrt{\left(\mathbf{B}_x^2 + \mathbf{B}_y^2\right)} = \kappa \left|\nabla \mathbf{B}\right| \tag{4.8}$$

In our model, GHF is used to decrease the curvature for the purpose of obtaining smooth boundaries and removing holes that appear because of noise. This is achieved as follows. Firstly, a segmented region is converted to a binary form as given below and also shown in Figure 4.4(c),

$$\mathbf{F}(\mathbf{x}) = \begin{cases} 1, & \mathbf{I}(\mathbf{x}) > 0 \\ 0, & \mathbf{I}(\mathbf{x}) = 0 \end{cases} \tag{4.9}$$

where $\mathbf{I}(\mathbf{x})$ is the temperature distribution after terminating diffusion and $\mathbf{F}(\mathbf{x})$ is the binary form of the segmented image that assigns unity to the region of interest. This binary form represents initial condition for GHF, $\mathbf{B}(\mathbf{x}, t=0) = \mathbf{F}(\mathbf{x})$. The boundaries of the image are insulated with homogeneous Neuman condition, $d\mathbf{B}/dn = 0$, which means there is no heat flow in, or out, from the boundaries. The GHF problem is defined below,

$$\begin{aligned} \frac{\partial \mathbf{B}}{\partial t} &= \alpha \frac{\left(\mathbf{B}_{xx}\mathbf{B}_y^2 - 2\mathbf{B}_{xy}\mathbf{B}_x\mathbf{B}_y + \mathbf{B}_{yy}\mathbf{B}_x^2\right)}{\left(\mathbf{B}_x^2 + \mathbf{B}_y^2\right)} \\ \mathbf{B}(\mathbf{x}, t=0) &= \mathbf{F}(\mathbf{x}), \quad \textit{initial condition} \\ \frac{\partial \mathbf{B}(\mathbf{x}, t)}{\partial n} &= 0, \quad \textit{boundary condition} \end{aligned} \tag{4.10}$$

GHF is applied to the $\mathbf{B}(\mathbf{x})$ until the specified time (number of iterations) and finally the resulting image is thresholded to obtain the final segmentation. The process is formulated below,

$$\mathbf{S}(\mathbf{x}) = \begin{cases} 1, & GHF(\mathbf{B}(\mathbf{x}), t_s) \geq 0.5 \\ 0, & GHF(\mathbf{B}(\mathbf{x}), t_s) < 0.5 \end{cases} \tag{4.11}$$

where, t_s is the number of iterations and $\mathbf{S}(\mathbf{x})$ is the binary form of the final segmentation, which assigns unity to the region of interest. The final segmentation is shown in Figure 4.4(d) and (e), where $t_s = 50$ for this illustration. The selection of t_s depends on the user and it is determined due to the noise level of the image. However, as t_s increases, extracted shape evolves to a circle, then to a point and then it is lost. For this reason, we should avoid using large values for t_s.

Since the illustrated human heart image seems bimodal, we can also consider the final form of the control function as shown in Figure 4.4(f). To smooth boundaries and remove holes, we simply continue with Equation 4.10 and observe the result in Figure 4.4(g) and (h).

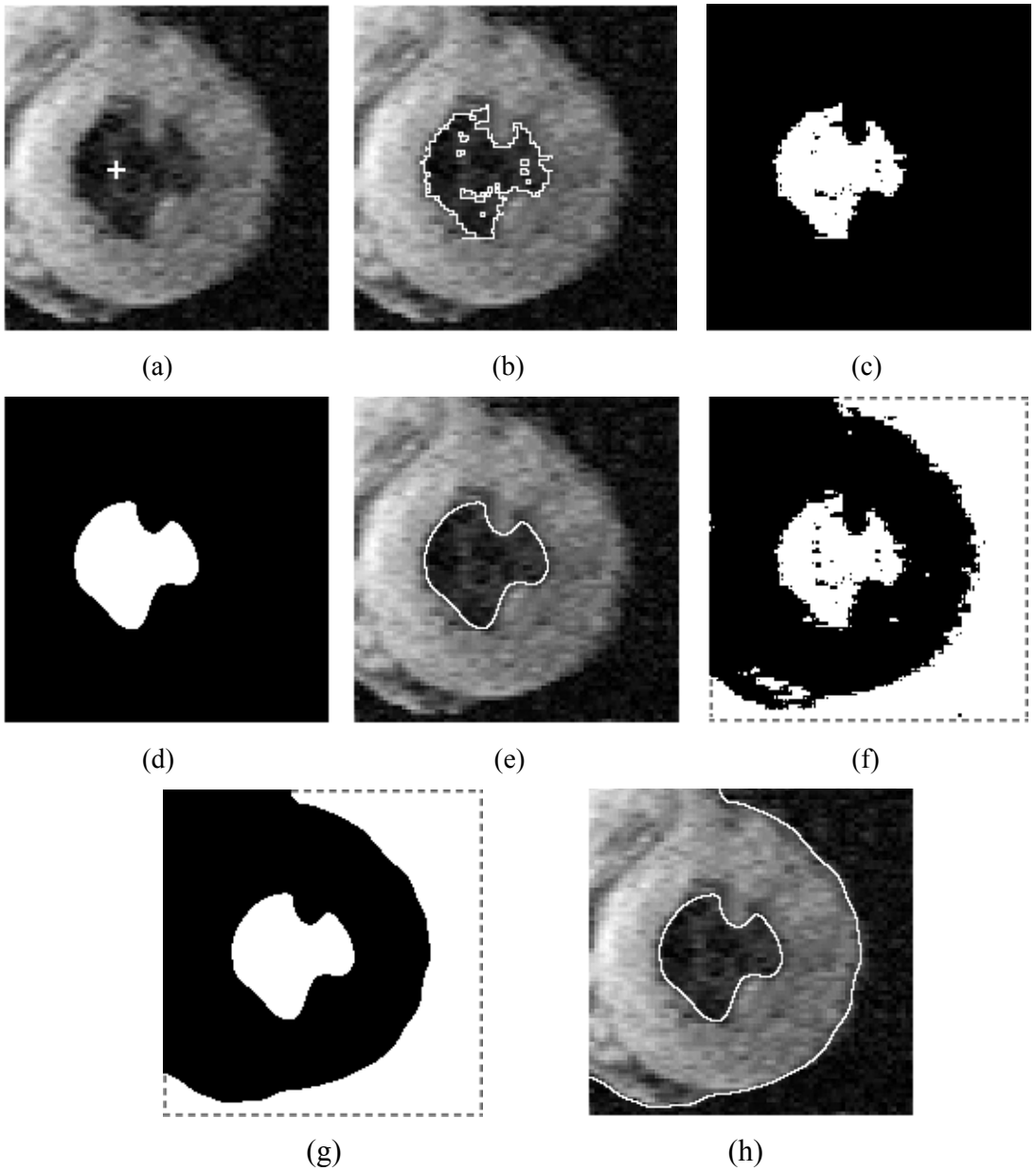


Figure 4.4: A magnetic resonance image of the left ventricle of a human heart and illustration of Geometric Heat Flow (GHF) for the purpose of obtaining smooth boundaries and removing holes inside the prior segmented regions. GHF is applied both to the binary form of the Temperature Front (TF) segmentation, $\mathbf{B}(\mathbf{x})$, and to the control function $\mathbf{CF}(\mathbf{x})$. The size of the image is 177×178, $\lambda_{in}/\lambda_{out} = 1$ and $t_s = 50$. (a) Source position. (b) Final position of the TF at $t = 59$. (c) Binary form of the segmentation by TF, $\mathbf{B}(\mathbf{x})$. (d) Binary form of the final segmentation, $\mathbf{S}(\mathbf{x})$, after GHF. (e) Final shape after GHF. (f) Final form of the control function, $\mathbf{CF}(\mathbf{x})$. (g) Control function, $\mathbf{CF}(\mathbf{x})$, after GHF. (h) Final shape after GHF.

4.4 Evaluation and Experimental Results

In evaluation, our model is denoted by TF+GHF, since the TF roughly segments the region of interest and then the GHF is used to complete the segmentation. Similarly, the segmentation by the control function (CF) followed by the GHF is denoted by CF+GHF. Performance evaluation is employed both for segmentation by TF+GHF and segmentation by CF+GHF. In this section, we first present the evaluation of TF+GHF and CF+GHF on segmentation, then the evaluation for the computational efficiency of our algorithm; finishing with some illustrative examples on medical images. Note that in this section we are using many acronyms, so for convenience a list of abbreviations is given at the beginning of this thesis.

Segmentation by TF+GHF is compared with the Active Contour Without Edges (ACWE) [64] and Gradient Vector Flow Snake (GVFS) [41]. The evaluation is done on a "harmonic" shape object [60] (the object boundary is a harmonic curve) and star shape object with varying normal distributed noise $N_d\left(\mu,\sigma^2\right)$, as shown respectively at the top rows in Figure 4.6 and 4.8. The sum of squared error (SSE) is employed to quantify the performance of each algorithm.

$$SSE=\sum_{x=1}^{M}\sum_{y=1}^{N}\left(\mathbf{S}_{x,y}-\mathbf{O}_{x,y}\right)^2 \tag{4.12}$$

Where, $\mathbf{S}$ is the binary segmented image and $\mathbf{O}$ is the actual (ground truth) binary segmented image of size $M\times N$. The quantity of noise is considered in terms of standard deviation σ with zero mean.

ACWE is a region-based GAC model that is implemented by a level set function [64]. It applies global minimization to segment bimodal images as a whole. However, in this evaluation, we choose the biggest segmented region, since we are concerned with the harmonic shape object segmentation. Otherwise, it will cause very high errors in noisy conditions because of the noise outside the harmonic object. To evaluate ACWE, we use a Matlab implementation given in [73], and we note that this is a non-optimal Matlab framework. In this evaluation, the selected parameter values for ACWE are: $\lambda_{in}=\lambda_{out}=1$ (parameters for regional statistics), $v_A=0$ (the area parameter), $h_A=1$

(the step space), $\Delta t = 0.1$ (the time space), $\varepsilon = 1$ (the parameter for the Heaviside and Dirac delta functions) and $\mu_A = 0.01 * 255^2$ (the length parameter).

GVFS is a gradient-based PAC model that uses GVF as an external force [41]. To evaluate GVFS, we use Matlab implementation given in [74], and this is also a non-optimal Matlab framework. In this evaluation, the selected parameter values for GVFS are: $\alpha_G = 0.25$ (smoothness parameter of the contour), $\beta_G = 0$ (rigidity parameter of the contour), $\gamma_G = 0.6$ (external energy parameter of the contour), and $\mu_G = 0.2$ (in diffusion of gradient vectors), $\Delta t = 1$ (the time interval). In addition, we use 80 iterations to diffuse gradient vectors.

Extensive explanations about ACWE and GVFS, which are compared with our model, are given in Appendix B.

In our algorithm, we use an explicit scheme of finite differences in the first and in the second part. Our finite difference formulation can be found in Appendix A. We use Matlab for the implementation as well. In this evaluation, the selected parameter values for TF+GHF are: $\alpha = 0.25$ (thermal diffusivity), $\lambda_{in}/\lambda_{out}$ =1 (ratio of the parameters for regional statistics), $Q = 5$ (the energy generated from the source position per unit time interval), $\Delta t = 1$ (the time interval), $\Delta x = \Delta y = 1$ (the spatial intervals), $t_s = 10$ (specified time for GHF). In addition, we start to use regional statistics after $t = 10$ to increase the number of samples inside the TF and in each 10 iterations we control the movement of TF to determine the termination of the first part.

In the evaluations for TF+GHF, the contours and the heat source are initialized inside the objects. Figure 4.5 shows the performance of TF+GHF, ACWE and GVFS for the harmonic object segmentation. The graphs are obtained over five applications of each algorithm, where the rectangle represents standard deviation from the mean value and error bar represents minimum and maximum values at each data point. In each application, the obtained graphs are smoothed by applying local averaging to the data points in 6 nearest neighbourhoods. It is observed that TF+GHF and ACWE perform much better than GVFS. The main reason for this is that TF+GHF and ACWE use region-based algorithms, on the other hand GVFS uses a gradient-based algorithm,

which is very sensitive to the noisy conditions. TF+GHF and ACWE can also handle topological changes, which GVFS cannot. When we compare TF+GHF and ACWE, ACWE performs better than TF+GHF until $\sigma \cong 40$. This appears to be due to the smoothing operation in TF+GHF. GHF attempts to smooth the original shape and cause errors in TF+GHF, when there is no noise or low noise in the image, since t_s is fixed in the evaluation. However, from $\sigma \cong 40$ to $\sigma \cong 80$, TF+GHF segments better than ACWE. The main reason is again the smoothing operation. TF+GHF applies smoothing after rough segmentation without any relation to the regional statistic constraints, while ACWE uses smoothness constraint with regional statistic constraints during the segmentation. After $\sigma \cong 80$, it is seen that ACWE shows better performance than TF+GHF. Because, ACWE segments many regions outside the harmonic region in the presence of high noise and then some of the segmented noise remains connected to the original region when we select the biggest region.

GVFS has the highest variation of the SSE at the data points and the SSE starts to vary after $\sigma \cong 20$. For ACWE, we observe variation in the SSE after $\sigma \cong 30$, and the variation looks similar between $\sigma \cong 45$ and $\sigma \cong 100$. The variations of SSE of ACWE are less than other algorithms at the highest noise levels. For TF+GHF, the SSE starts to vary after $\sigma \cong 45$, and in general the SSE increases as noise increases. Figure 4.6 shows some of the results for TF+GHF (second row), ACWE (third row) and GVFS (fourth row).

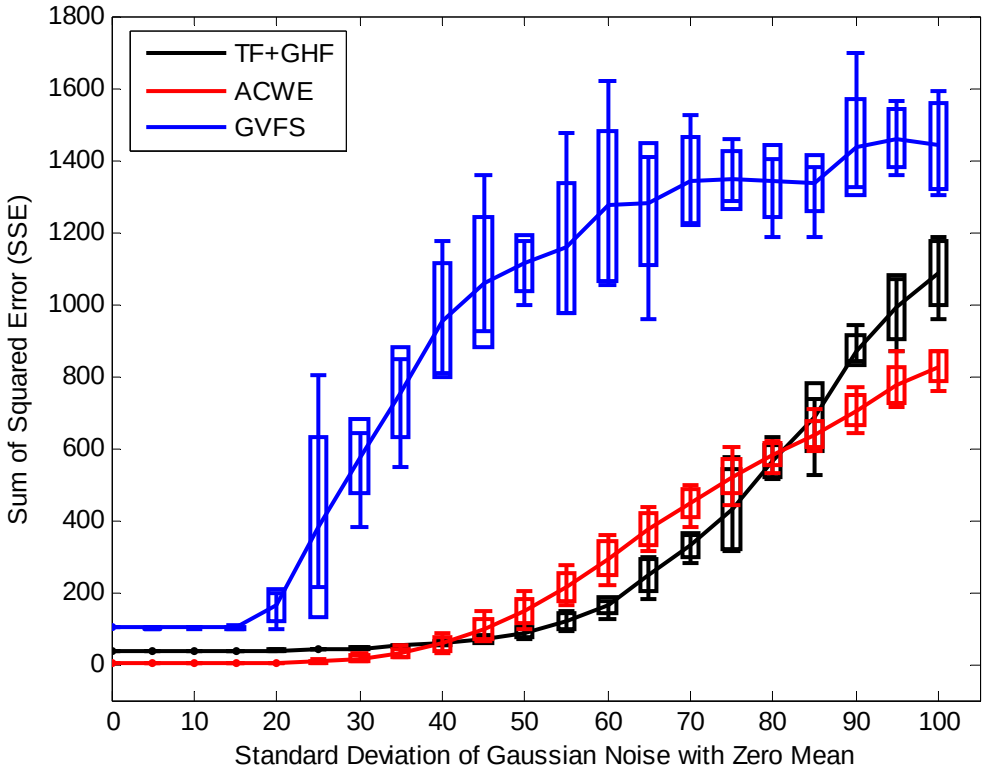


Figure 4.5: Performance of TF+GHF, ACWE and GVFS on the harmonic shape object. The graphs show the rectangle representing standard deviation from the mean value and error bar representing minimum and maximum values of the SSE.

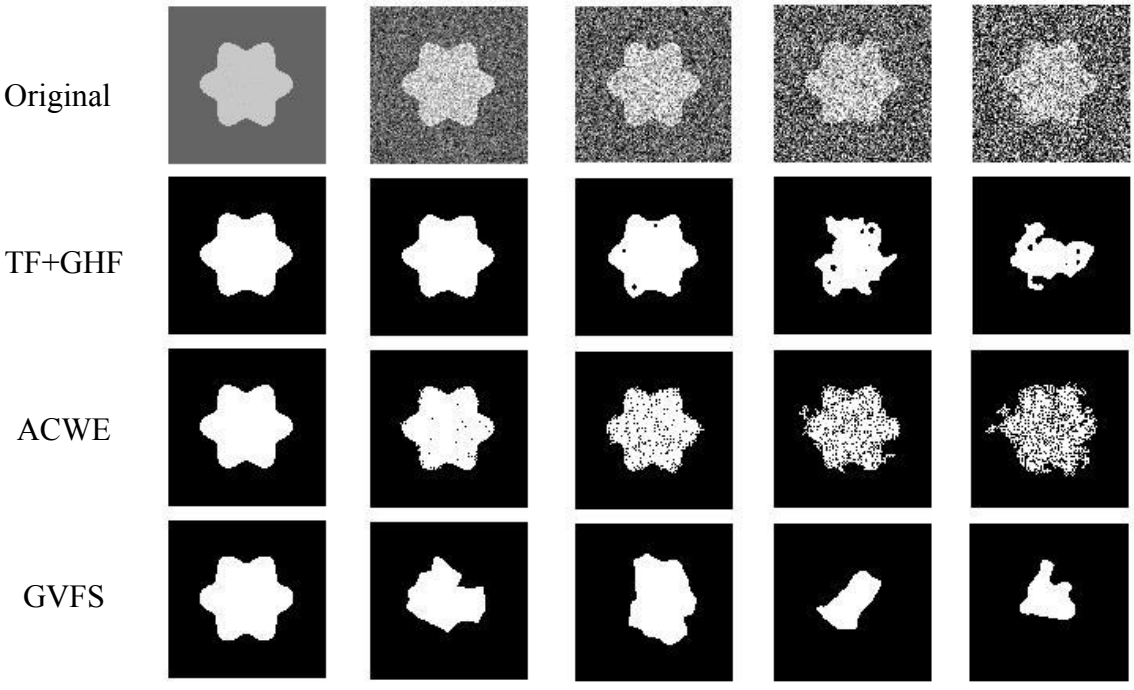


Figure 4.6: Results for TF+GHF (second row), ACWE (third row) and GVFS (fourth row) with respect to increasing Gaussian noise in the image of size 100×100.

Figure 4.7 shows the performance of TF+GHF, ACWE and GVFS for the segmentation of a star object. The graphs are obtained over five applications of each algorithm, where the rectangle represents standard deviation from the mean value and error bar represents minimum and maximum values at each data point. In each application, the obtained graphs are smoothed by applying local averaging to the data points in 6 nearest neighbourhoods. The star object has branches, which narrow until they are one pixel wide. Half of the star object has different intensity, to create weak edges inside the object. It is again observed that TF+GHF and ACWE perform much better than GVFS. The reason for this is that TF+GHF and ACWE use region-based algorithms, on the other hand GVFS uses a gradient-based algorithm, which is very sensitive to the conditions there. It is also observed that GVFS has worse performance in the segmentation of the star object comparing to the segmentation of the harmonic object, because, GVFS cannot flow into the narrow regions of the star object. TF+GHF and ACWE can flow into the narrow regions. ACWE performs better than TF+GHF until $\sigma \cong 30$. This appears to be due to the smoothing operation in TF+GHF. GHF attempts to smooth the original shape and cause errors in TF+GHF, when there is no noise or low noise in the image, since t_s is fixed in the evaluation. However, from $\sigma \cong 30$ to $\sigma \cong 80$, TF+GHF segments better than ACWE. The main reason is again the smoothing operation. TF+GHF applies smoothing after rough segmentation without any relation to the regional statistic constraints, while ACWE uses smoothness constraint with regional statistic constraints during the segmentation. After $\sigma \cong 80$, it is seen that ACWE shows better performance than TF+GHF. Because, ACWE segments many regions outside the star region in the presence of high noise and then some of the segmented noise remains connected to the original region when we select the biggest region.

The SSE of GVFS starts to vary after $\sigma \cong 5$, and the variations are very high between $\sigma \cong 10$ and $\sigma \cong 45$ in comparison to the other algorithms. For ACWE, we observe variation in the SSE after $\sigma \cong 25$, and the variations look similar at all the remaining noise levels. The variations of SSE of ACWE are less than other algorithms at high noise levels, which is consistent with the analysis in Figure 4.5. For TF+GHF, the SSE varies slightly between $\sigma \cong 40$ and $\sigma \cong 60$. After $\sigma \cong 65$, we can observe significant

variation in the SSE. Figure 4.8 shows some of the results for TF+GHF (second row), ACWE (third row) and GVFS (fourth row).

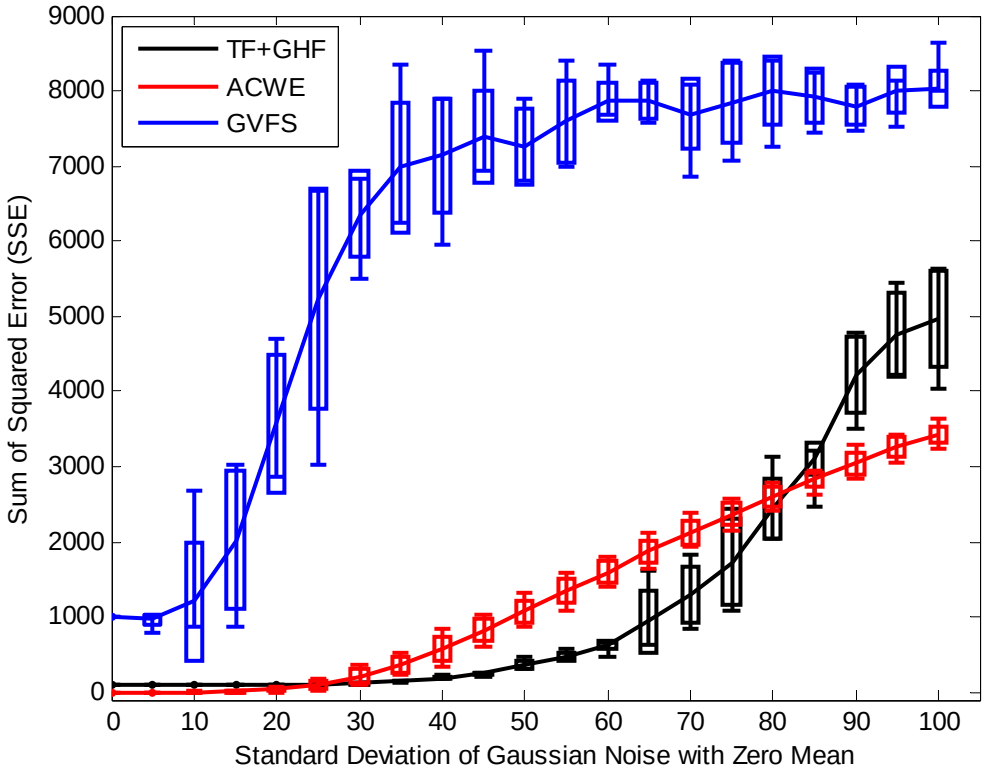
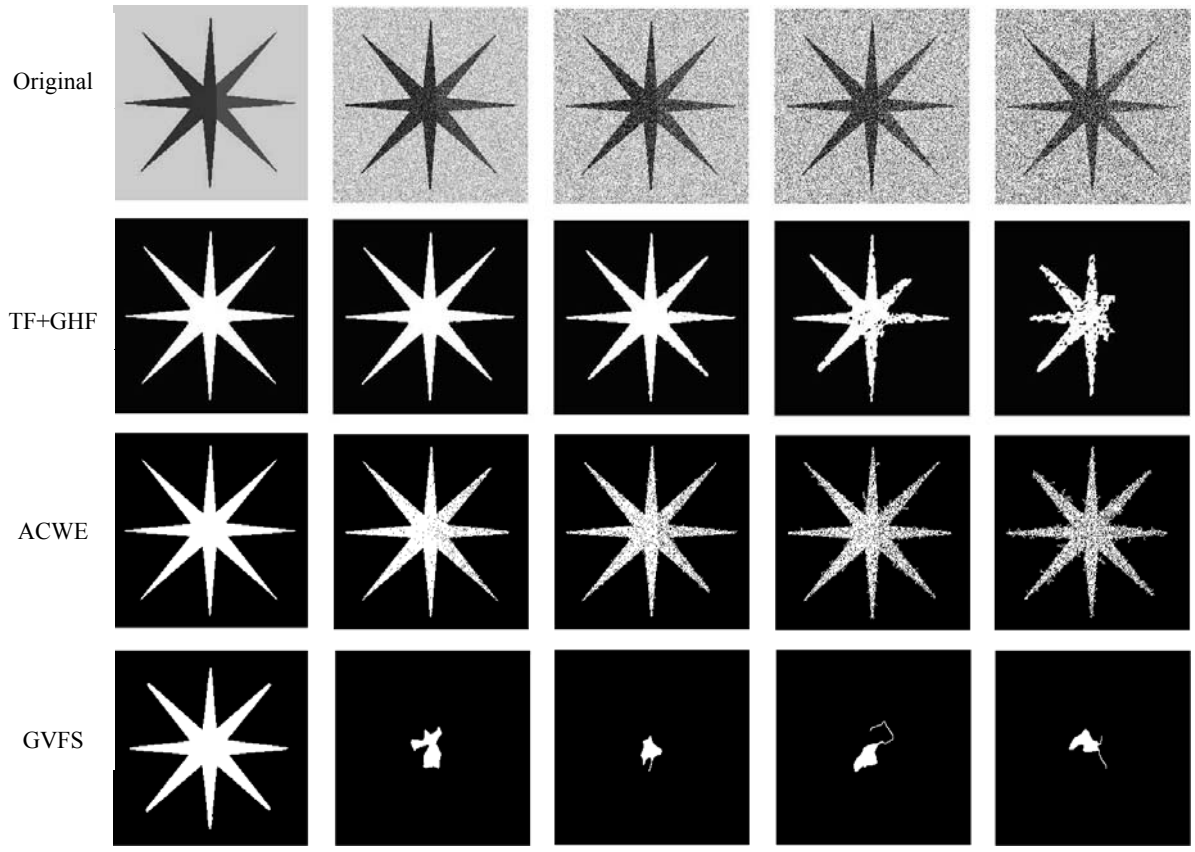


Figure 4.7: Performance of TF+GHF, ACWE and GVFS on the star shape object. The graphs show the rectangle representing standard deviation from the mean value and error bar representing minimum and maximum values of the SSE.



(a) $\sigma = 0$ (b) $\sigma = 40$ (c) $\sigma = 60$ (d) $\sigma = 80$ (e) $\sigma = 100$

Figure 4.8: Results for TF+GHF (second row), ACWE (third row) and GVFS (fourth row) with respect to increasing Gaussian noise in the image of size 256×256.

Segmentation by CF+GHF is evaluated by comparing with ACWE. Since both attempt to segment bimodal images as a whole, the evaluation is done on an image that has two harmonic objects with varying normal distributed noise, as shown at the top row in Figure 4.10. As before, SSE is employed to quantify the performance of each algorithm. We use the parameters as given before, however we do not choose the biggest segmented regions for ACWE. In this evaluation, the contour and the heat source are initialized into one of the harmonic objects. Figure 4.9 shows the performance of CF+GHF and ACWE. The graphs are obtained over five applications of each algorithm, where the rectangle represents standard deviation from the mean value and error bar represents minimum and maximum values at each data point. In each application, the obtained graphs are smoothed by applying local averaging to the data points in 6 nearest neighbourhoods. It is observed that until $\sigma \cong 40$ ACWE performs better than CF+GHF, which appears because of the smoothing operation in CF+GHF. As explained before, GHF attempts to smooth original shape, when there is no noise or low noise in the image, since t_s is fixed in the evaluation. However, after $\sigma \cong 40$,

CF+GHF performs better than ACWE. The main reason is the smoothing operation, since the smoothing in CF+GHF is applied after the rough segmentation without any relation to the regional statistic constraints, while the smoothness constraint in ACWE is applied together with the regional statistic constraints during the segmentation. The SSE of ACWE starts to vary after $\sigma \cong 30$, and the variations look similar after $\sigma \cong 40$ at each data point. The variations of SSE of CF+GHF are less than those for ACWE and the SSE starts to vary after $\sigma \cong 35$ slightly. Figure 4.10 shows some of the results for CF+GHF (second row) and ACWE (third row).

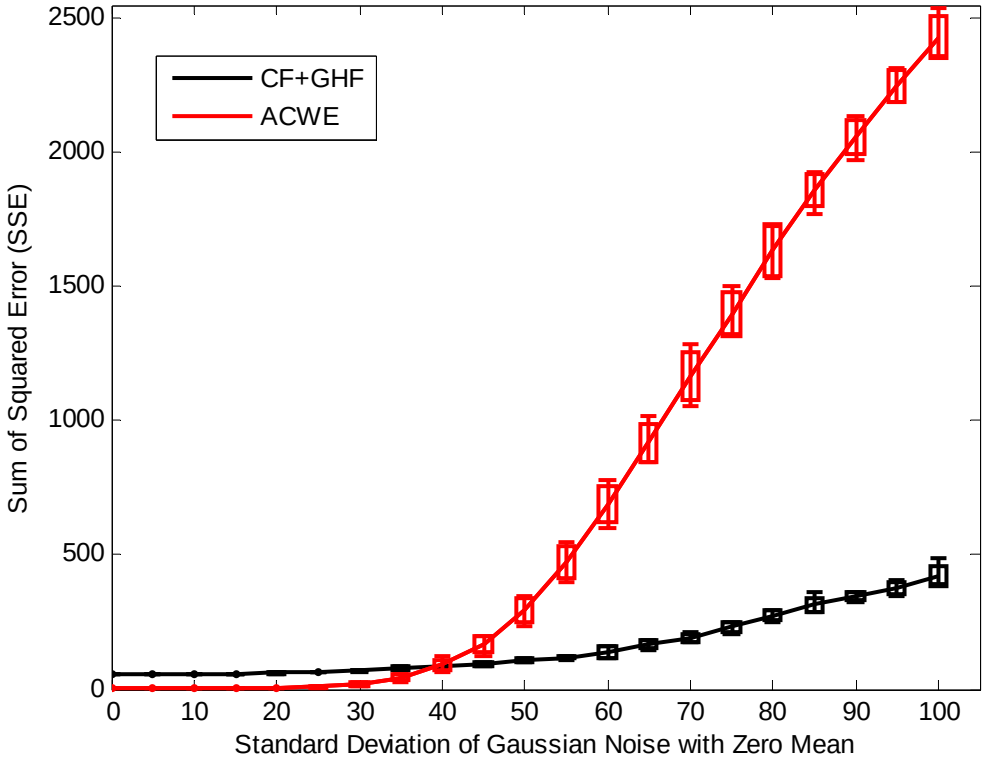


Figure 4.9: Performance of CF+GHF and ACWE. The graphs show the rectangle representing standard deviation from the mean value and error bar representing minimum and maximum values of the SSE.

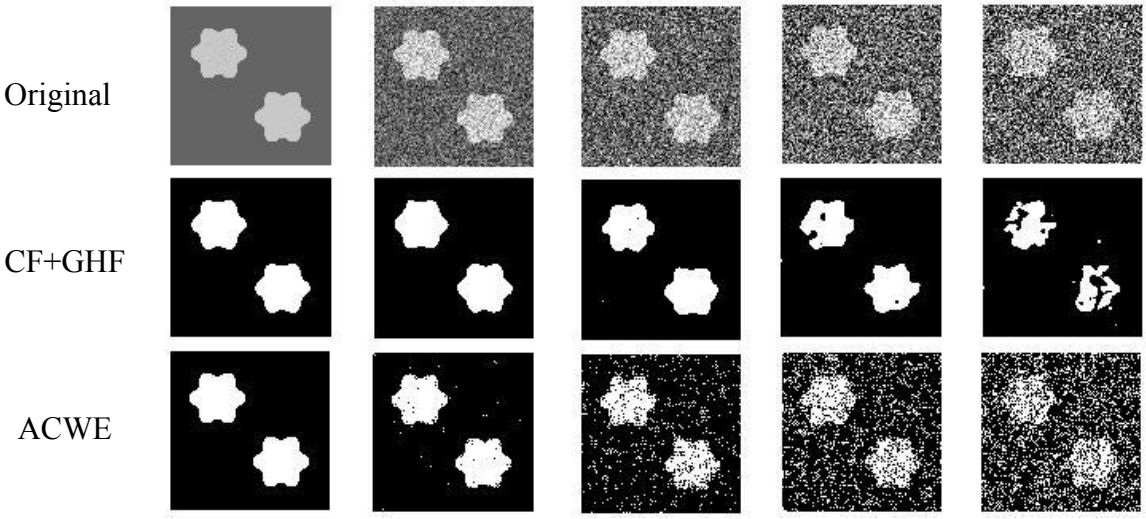


Figure 4.10: Results for CF+GHF (second row) and ACWE (third row) with respect to increasing Gaussian noise in the image of size 100×100.

Results also show the effectiveness and the computational efficiency of our algorithm in comparison to the GVFS and ACWE. The following evaluations results are obtained by using MATLAB 7.0 on a Pentium IV computer, which runs Windows XP operating system with 3.2 GHz CPU and 3GB RAM.

First we investigate how the SSE of TF+GHF, of ACWE and of GVFS changes, on the original (no noise) and on the noisy star object, as iteration number increases (as the contour evolves). We initialize contours at the centre of the star object and arrange their sizes to have SSE $\cong 8000$ at the beginning. Figure 4.11 shows SSE of TF+GHF, of ACWE and of GVFS with respect to increasing iteration numbers on the original star object, shown in Figure 4.8 (a). Table 4.1 also shows the number of iterations required to stabilize SSE, remaining SSE after stabilization and CPU time required for stabilization for each algorithm on the original star object. The final segmentation by each algorithm can be seen in Figure 4.8 (a) as well. We observe that GVFS stabilizes at iteration $\cong 170$ with SSE $\cong 1000$ and CPU time=57.63 seconds. For further iterations GFVS remains at that SSE, because it cannot move through the narrow regions and can not achieve complete segmentation. ACWE achieves complete segmentation at iteration $\cong 890$ with SSE $= 0$ and CPU time=5.92 minutes. Finally, TF+GHF achieves segmentation at iteration $\cong 180$ with SSE $\cong 93$ and CPU time=7.17 seconds. There remains some error, because of the smoothing operation in TF+GHF. GHF is applied, at the end of first part in our algorithm, with fixed iteration number (10 iterations) and

causes errors when there is no noise on the object. It is also observed that TF+GHF achieves segmentation with the lowest CPU time. Although GVFS segments a smaller region, it requires CPU time more than TF+GHF. It must be noted that GVFS also needs 80 iterations to diffuse gradient vectors of the image, which is the image-dependent force it uses, before contour evolution. There is an also big difference in CPU times of ACWE and TF+GHF. ACWE is computationally expensive since it uses the level set method for curve evolution.

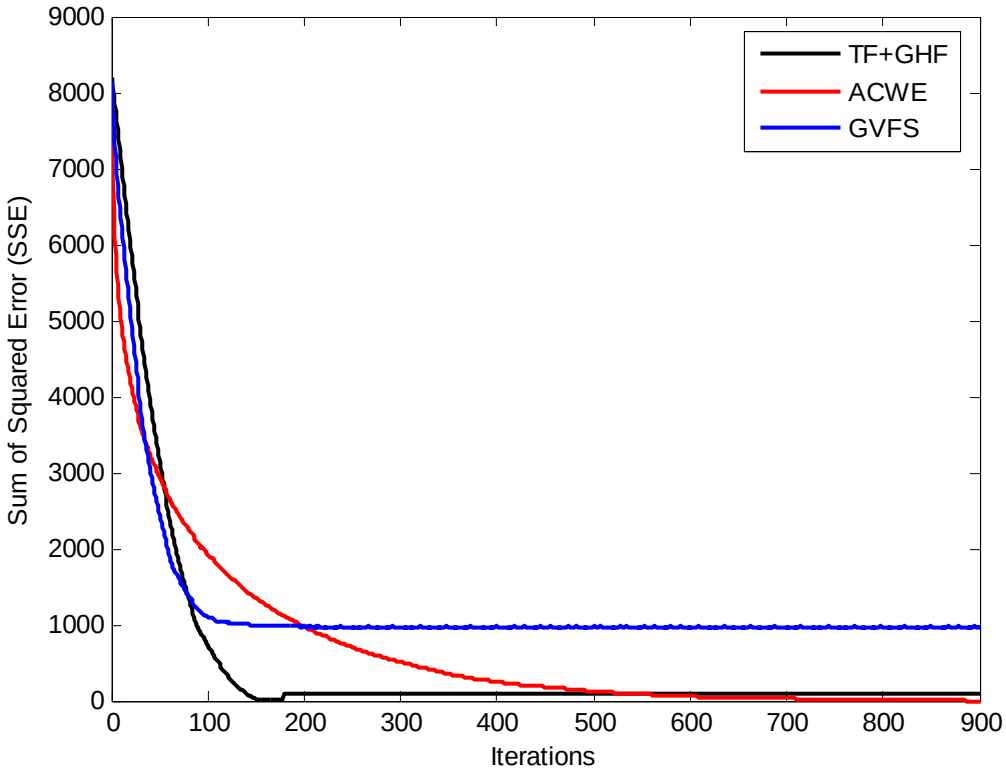


Figure 4.11: SSE for TF+GHF, ACWE and GVF with respect to increasing iteration numbers from the initial positions to the final segmentations on the original star object.

Table 4.1: Number of iterations required to stabilize, remaining SSE after stabilization and CPU time required to stabilize for TF+GHF, ACWE and GVFS on the original star object.

Original Star Object (no noise)			
Algorithm	Num. of iteration to stabilize	Remaining SSE	CPU Time
TF+GHF	180	93	7.17 seconds
ACWE	890	0	5.92 minutes
GVFS	170	1000	57.63 seconds

Figure 4.12 shows SSE of TF+GHF, of ACWE and of GVFS with respect to increasing iteration numbers on the noisy star object, shown in Figure 4.8 (b), which is corrupted by Gaussian noise of $\sigma = 40$. Table 4.2 also shows the number of iterations required to stabilize, remaining SSE after stabilization and CPU time required for each algorithm on the noisy star object. The final segmentation by each algorithm can be seen in Figure 4.8 (b) as well. It is observed that as the iteration number increases, GVFS stabilizes at iteration $\cong 18$ with SSE $\cong 6950$ and CPU time 10.57 seconds. It achieves poor segmentation because of the limitation of the gradient-based information to the noise. ACWE achieves segmentation at iteration $\cong 1100$ with SSE $\cong 570$ and CPU time=7.29 minutes. Finally, TF+GHF achieves segmentation at iteration $\cong 180$ with SSE $\cong 188$ and CPU time=7.26 seconds. In TF+GHF, we observe sudden decrease in SSE before stabilizing. This is because, we apply a fixed number of iterations of GHF (10 iterations) to remove noise and smooth shape at the end of our algorithm. It is observed that TF+GHF achieves segmentation with minimum SSE and again with the lowest CPU time. Although GVFS has poor performance because of noise and segments small region, TF+GHF has slightly smaller CPU time than GVFS. There is again a big difference in CPU times of ACWE and TF+GHF, where TF+GHF performs much faster than ACWE. ACWE also has higher SSE than TF+GHF, which means TF+GHF performs better than ACWE in this segmentation.

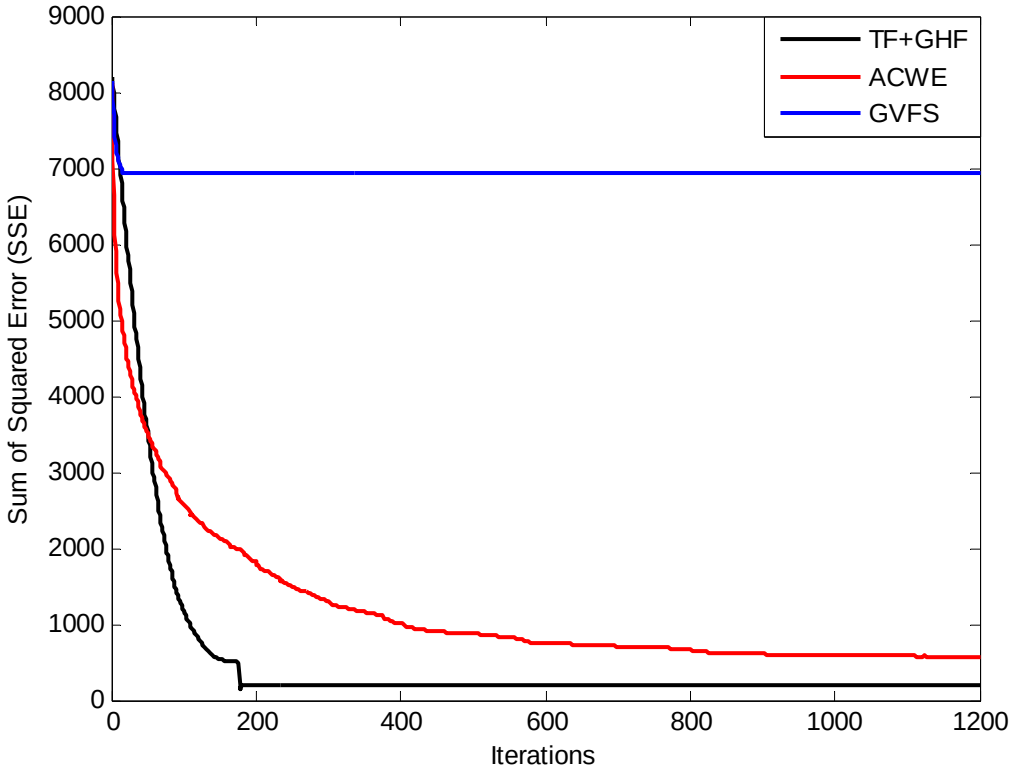


Figure 4.12: SSE for TF+GHF, ACWE and GVF with respect to increasing iteration numbers from the initial positions to the final segmentations on the noisy star object.

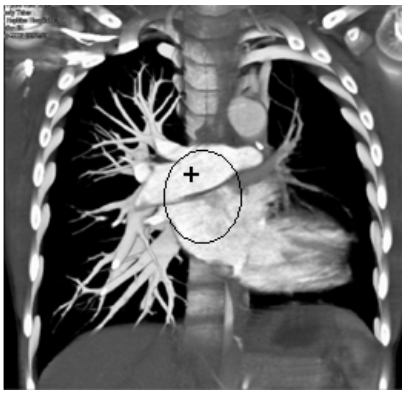
Table 4.2: Number of iterations required to stabilize, remaining SSE after stabilization and CPU time required to stabilize for TF+GHF, ACWE and GVFS on the noisy star object.

Noisy Star Object ($\sigma = 40$)			
Algorithm	Num. of iteration to stabilize	Remaining SSE	CPU Time
TF+GHF	180	188	7.26 seconds
ACWE	1100	570	7.29 minutes
GVFS	18	6950	10.57 seconds

It is important to note that rest of the experiments, in this Chapter, are conducted by same computer and using Matlab 7.0, as described before, except the RAM is 1GB.

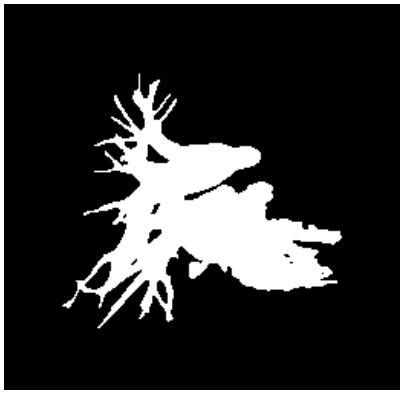
Figure 4.13 shows the segmentation of pulmonary arterial branches in the chest image by TF+GHF, GVFS and ACWE. The initial contour for GVFS and ACWE, and the source position for TF+GHF are shown in Figure 4.13(a). Figure 4.13(b) shows the segmentation by TF+GHF in the given image with black contour, however the

segmented arterial branches are not visible with this illustration and the segmentation is also shown with the binary form in Figure 4.13(c). Figure 4.13(d) and (e) show segmentation by GVFS respectively with the result superimposed as the black contour in the image and the binary form. Figure 4.13(f) and (g) show segmentation by ACWE respectively with black contour superimposed and the binary form. It is observed that TF+GHF segments the desired arterial branches better than GVFS. By this result, TF+GHF can easily handle topological changes and flow into the arterial branches with CPU=7.85s. However, GVFS cannot handle topological changes and cannot flow into the arterial branches. Although GVFS segments a smaller region than TF+GHF, the CPU=9.23s, which is more than for TF+GHF. If we look at segmentation by ACWE, we see that ACWE attempts to segment whole image. Here we focus on segmentation of desired region, which is pulmonary arterial branches, and compare with the TF+GHF. TF+GHF and ACWE have similar segmentation results on the desired region. However, ACWE achieves this segmentation in 24.75 minutes, while TF+GHF achieves in 7.85 seconds. This big difference in CPU time appears because of the computational complexity of ACWE that is implemented with level sets. The CPU times of the algorithms for the pulmonary arterial branches are also shown in Table 4.3.



(a)

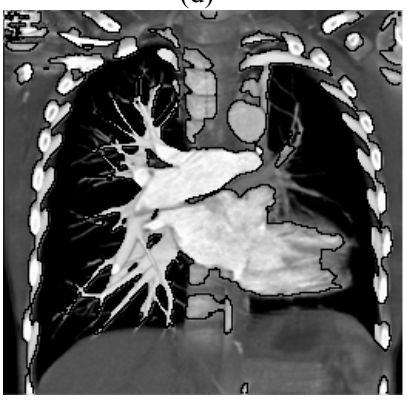
(b)



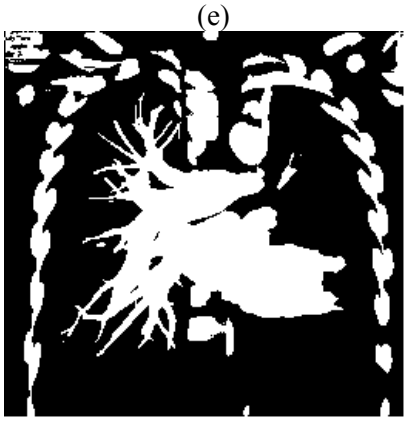
(c)

(d)

(e)



(f)



(g)

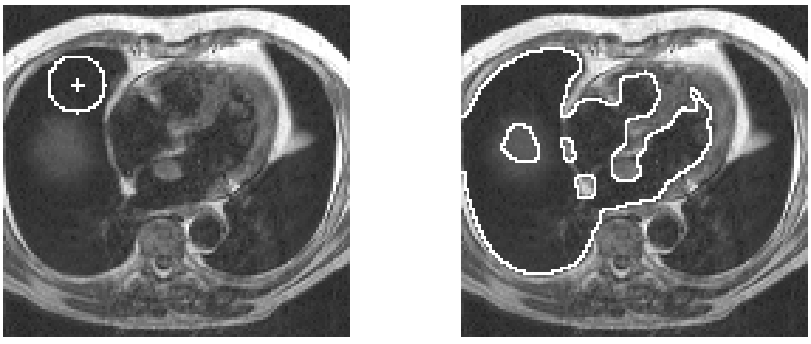
Figure 4.13: Segmentation of pulmonary arterial branches in the chest image of size 259×250 by TF+GHF, GVFS and ACWE. (a) Initial contour and the source position. (b) Segmentation

by TF+GHF is shown by the black contour on the image. All the parameters are same as in evaluation except $t_s = 5$. (CPU=7.85 seconds). (c) Segmentation by TF+GHF is in binary form. (d) Segmentation by GVFS is shown with black contour on the image. All the parameters are same as in evaluation except the iteration to diffuse gradient vectors is 70. (CPU=9.23 seconds). (e) Segmentation by GVFS is in binary form. (f) Segmentation by ACWE is shown with black contour on the image. All the parameters are same as in evaluation except the length parameter $\mu = 0.08 \times 255^2$. (CPU=24.75 minutes). (g) Segmentation by ACWE is in binary form.

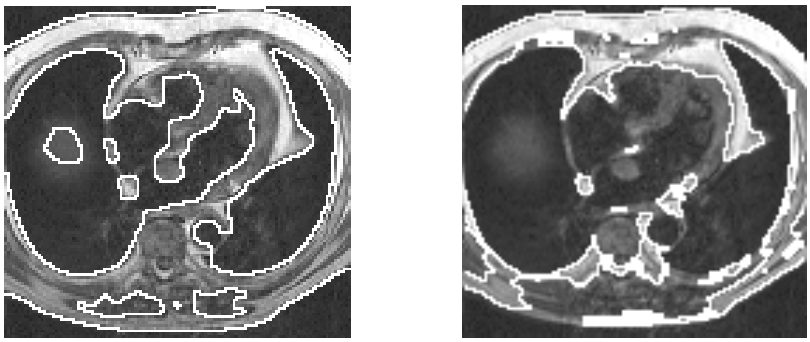
Table 4.3: The CPU times of the algorithms for the pulmonary arterial branches in the chest image

Pulmonary Image	
Algorithm	CPU Time
TF+GHF	7.85 seconds
GVFS	9.23 seconds
ACWE	24.75 minutes

Figure 4.14 shows the segmentation of a bimodal human lung image by TF+GHF, CF+GHF and ACWE, where the initial contour for ACWE and the source position for TF+GHF are shown in Figure 4.14(a). Figure 4.14(b) and (c) respectively show the segmentation by TF+GHF and CF+GHF with white contour in the image. Figure 4.14(d) shows the segmentation by ACWE. It is observed that TF+GHF and CF+GHF achieves segmentation with CPU=1.96 seconds and ACWE achieves with CPU=15.92 minutes. This big difference in CPU time appears because of the computational complexity of ACWE that is implemented with level sets. The CPU times of our algorithm and ACWE for the human lung image are also shown in Table 4.4. It is also observed that CF+GHF can extract feature boundaries better than ACWE especially at the middle and at the bottom of the lung image.



(a) (b)



(c) (d)

Figure 4.14: Segmentation of human lung image of size 123×118 by TF+GHF, CF+GHF and ACWE. (a) Initial contour and the source position. (b) Segmentation by TF+GHF is shown with white contour on the image. All the parameters are same as in evaluation except $t_s = 15$ (CPU=1.96 seconds). (c) Segmentation by CF+GHF. $t_s = 15$. (CPU=1.96 seconds). (d) Segmentation by ACWE. All the parameters are same as in evaluation except the length parameter $\mu = 0.08\times 255^2$. (CPU=15.92 minutes).

Table 4.4: The CPU times of the algorithms for the human lung image

Human lung image	
Algorithm	CPU Time
TF+GHF and CF+GHF	1.96 seconds
ACWE	15.92 minutes

We also discuss how the different regional statistic parameters $(\lambda_{in}/\lambda_{out})$ and the different iteration number for GHF (t_s) effect the segmentation. Figure 4.15 shows the segmentation of the magnetic resonance image of the left ventricle of a human heart by TF+GHF and CF+GHF with respect to the increasing $\lambda_{in}/\lambda_{out}$, while keeping $t_s = 50$. The heat source is located at the same position, as shown in Figure 4.4 (a). Earlier,

$\lambda_{in}/\lambda_{out}$ =1 in the experiment shown in Figure 4.4. In this experiment, the selected values for $\lambda_{in}/\lambda_{out}$ = 0.25, 0.5, 2, 4 and the segmentation results by TF+GHF and CF+GHF are given in Figures 4.15 (a), (b), (c) and (d) respectively. We observe that as $\lambda_{in}/\lambda_{out}$ increases, the selectivity for segmentation increases with respect to the region of source location.

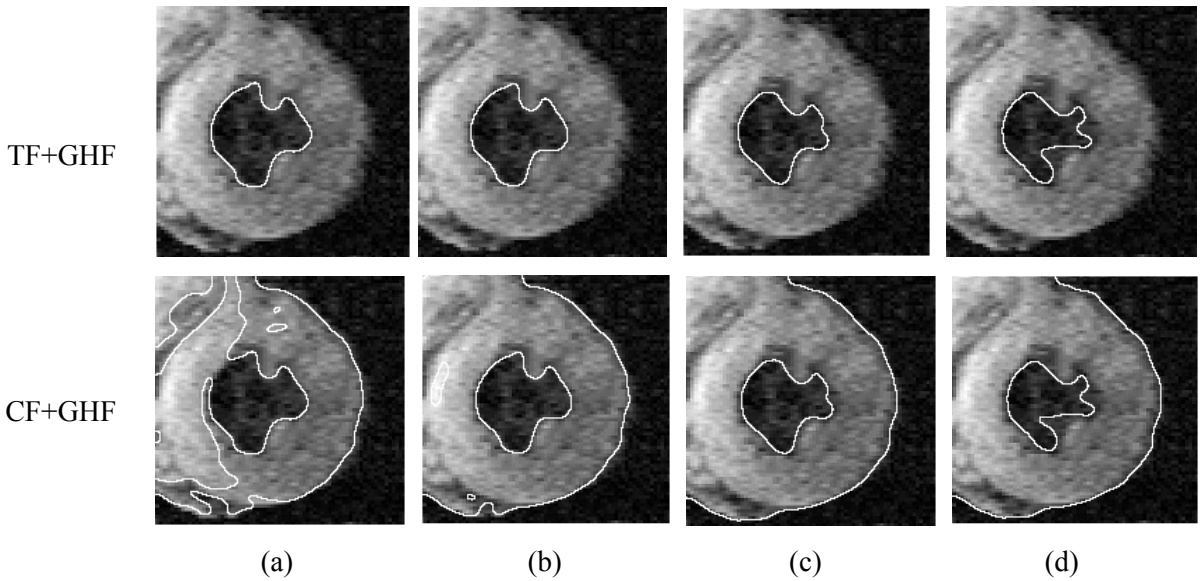


Figure 4.15: A magnetic resonance image of the left ventricle of a human heart and illustration of the segmentation by TF+GHF and CF+GHF with respect to the increasing $\lambda_{in}/\lambda_{out}$ (ratio of regional statistics parameters). (a) $\lambda_{in}/\lambda_{out}$ =0.25, (b) $\lambda_{in}/\lambda_{out}$ =0.5, (c) $\lambda_{in}/\lambda_{out}$ =2, (d) $\lambda_{in}/\lambda_{out}$ =4.

Figure 4.16 shows the segmentation of the magnetic resonance image of the left ventricle of a human heart by TF+GHF and CF+GHF with respect to the increasing t_s, while keeping $\lambda_{in}/\lambda_{out}$ =1. The heat source is located at the same position shown in Figure 4.4 (a) and t_s =50 in that experiment. In this experiment, the selected values for t_s = 20, 40, 60, 80 and the segmentation results by TF+GHF and CF+GHF are illustrated in Figures 4.16 (a), (b), (c) and (d) respectively. It is observed that as t_s increases, we obtain smoother segmentation and remove smaller regions.

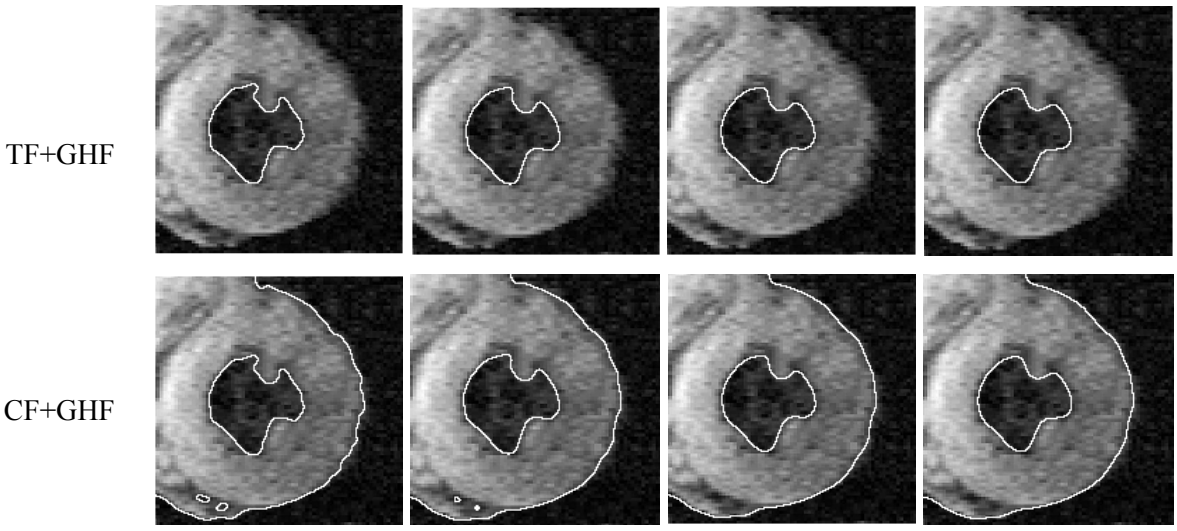


Figure 4.16: A magnetic resonance image of the left ventricle of a human heart and the segmentation by TF+GHF and CF+GHF with respect to the increasing t_s (iteration number of GHF). (a) t_s=20, (b) t_s=40, (c) t_s=60, (d) t_s=80.

Figure 4.17 shows the segmentation of the human lung image by TF+GHF and CF+GHF with respect to the increasing $\lambda_{in}/\lambda_{out}$, while keeping t_s=15. The heat source is located at the same position shown in Figure 4.14 (a) and $\lambda_{in}/\lambda_{out}$=1 in that experiment. In this experiment, the selected values for $\lambda_{in}/\lambda_{out}$ = 0.25, 0.5, 2, 4, 6 and the segmentation results by TF+GHF and CF+GHF are given in Figures 4.17 (a), (b), (c), (d) and (e) respectively. In Figure 4.17 (a), it is observed that TF+GHF moves out of desired region since selecting $\lambda_{in}/\lambda_{out}$=0.25 is too small with respect to region of source location. As $\lambda_{in}/\lambda_{out}$ increases from 0.5 to 6, the selectivity for segmentation increases, and TF+GHF does not flow outside desired region.

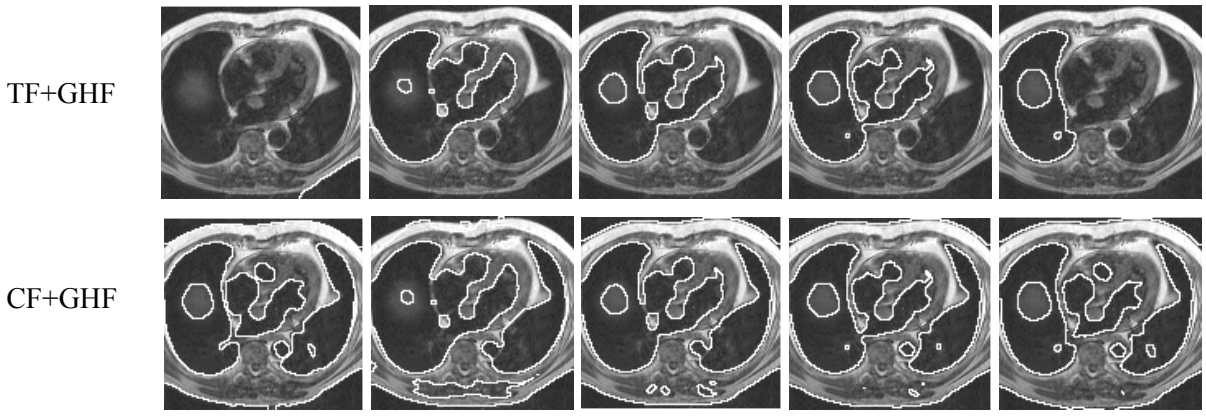


Figure 4.17: Segmentation of human lung image by TF+GHF and CF+GHF with respect to the increasing $\lambda_{in}/\lambda_{out}$ (ratio of regional statistics parameters). (a) $\lambda_{in}/\lambda_{out}$ =0.25, (b) $\lambda_{in}/\lambda_{out}$ =0.5, (c) $\lambda_{in}/\lambda_{out}$ =2, (d) $\lambda_{in}/\lambda_{out}$ =4, (e) $\lambda_{in}/\lambda_{out}$ =6 .

Figure 4.18 shows the segmentation of the human lung image by TF+GHF and CF+GHF with respect to the increasing t_s , while keeping $\lambda_{in}/\lambda_{out}$ =1. The heat source is located at the same position shown in Figure 4.14 (a) and t_s =15 in that experiment. In this experiment, the selected values for t_s = 5, 10, 20, 25 and the segmentation results by TF+GHF and CF+GHF are shown in Figures 4.18 (a), (b), (c) and (d) respectively. It is observed that as t_s increases, we obtain smoother segmentation, as well as remove noise and smaller regions.

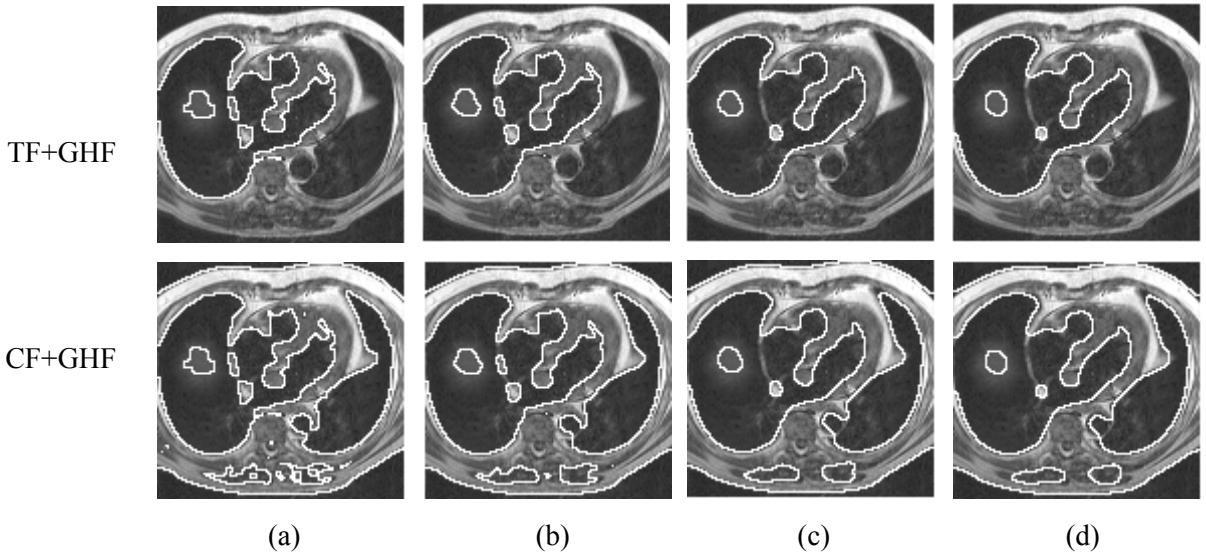
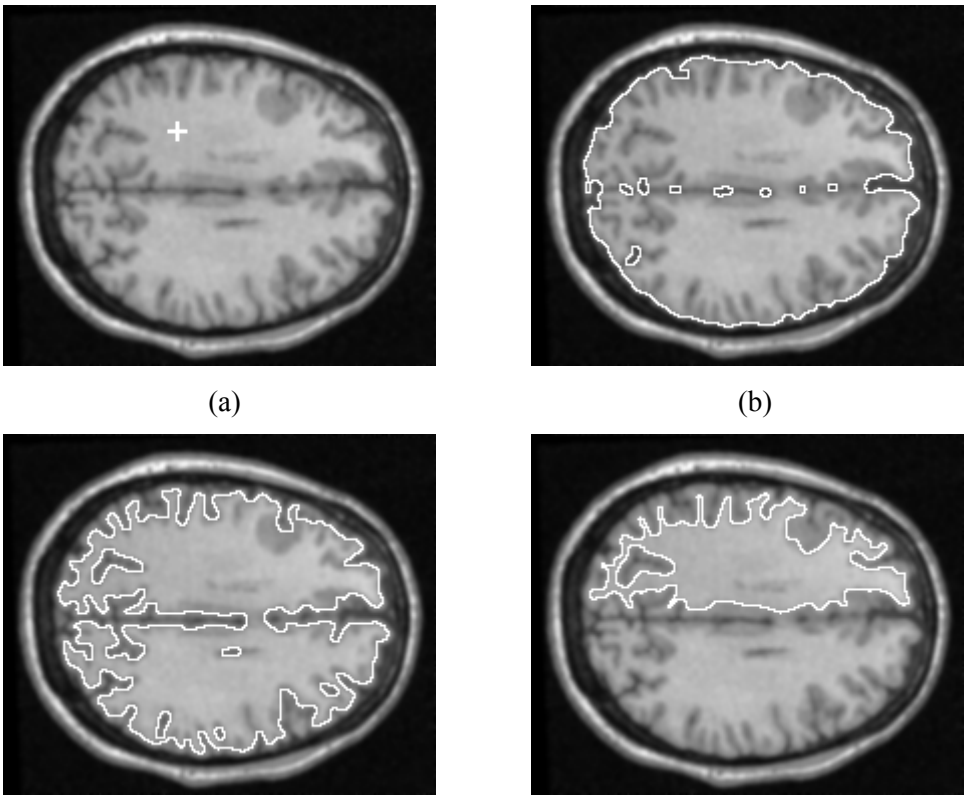


Figure 4.18: Human lung image and segmentation by TF+GHF and CF+GHF with respect to the increasing t_s (iteration number of GHF). (a) t_s =5, (b) t_s =10, (c) t_s =20, (d) t_s =25.

Finally, Figure 4.19 shows the segmentation in a human brain image by TF+GHF with respect to the different ratio of the regional statistic parameters. Figure 4.19 (a) shows the heat source position. Figure 4.19 (b) illustrates the segmentation result for $\lambda_{in}/\lambda_{out}$=1. In Figure 4.19 (c), $\lambda_{in}/\lambda_{out}$=5, which provides more selective segmentation in the brain image and in Figure 4.19 (d), $\lambda_{in}/\lambda_{out}$=8.3, which increases the selectivity comparing to in Figure 4.19 (c). Note that t_s=7 in this experiment.



(a) (b)

(c) (d)

Figure 4.19: Segmentation of human brain image of size 216×180 by TF+GHF with different regional statistic parameters. All the parameters are same as in evaluation except the regional statistic ones and $t_s = 7$. (a) Initial source position. (b) $\lambda_{in}/\lambda_{out} = 1$. (c) $\lambda_{in}/\lambda_{out} = 5$. (d) $\lambda_{in}/\lambda_{out} = 8.3$.

4.5 Conclusions

We have presented a novel segmentation algorithm based on using the heat flow analogy. In the first part of the algorithm, we roughly extract the desired feature boundaries by representing a chosen heat conduction analysis in the image domain. The representation in image domain is achieved by using a control function (CF) in the heat

conduction equation. This formulation also provides advantage when the given image is bimodal, since CF attempts to segment whole image in this case. In the second part, we use geometric heat flow (GHF) to tune the curvature of the extracted feature boundaries and remove possible noise that arises from the first part of the segmentation. Evaluation results indicate that Temperature Front (TF) + GHF has better performance than gradient vector flow snake (GVFS) and active contour without edges (ACWE) with respect to increasing Gaussian noise. For the bimodal images, CF+GHF has better performance than ACWE. These improvements are achieved by effective organization of our algorithm, TF+GHF segments better than GVFS because of using region-based information instead of gradient-based, which is sensitive to noise. TF+GHF and CF+GHF segment better than ACWE because of the smoothing operation of shape. TF+GHF and CF+GHF apply smoothing after rough segmentation without any relation to the regional statistic constraints, while ACWE uses smoothness constraint with regional statistic constraints during the segmentation. In addition, TF+GHF and CF+GHF are computationally more efficient and effective than both GVFS and ACWE based on the simulation results. Especially, there are big differences between our model and ACWE in computational efficiency. The main reason is the complexity of the level set method in ACWE. As such, the heat analogy can be deployed with success for shape extraction in images.

Chapter 5 Shape Classification using Multiscale Fourier-based description in 2-D Space

The solution to analysis of heat conduction can be achieved by using Fourier series or Fourier integrals [16, 19, 20]. Furthermore, the solution of linear and homogeneous heat equation in infinite medium can be viewed as smoothing a signal with a Gaussian filter [19, 20], as shown in Equation 2.23 in Chapter 2. In our research, we use Fourier theory and the Gaussian filtering approach to introduce new and efficient silhouette object descriptors for recognition purposes.

In shape recognition, a multiscale description provides more information about the object, increases discriminatory power and immunity to noise. In this Chapter, we develop a new multiscale Fourier-based object description in 2-D space using a low-pass Gaussian filter (LPGF) and a high-pass Gaussian filter (HPGF), separately. Using the LPGF at different scales (standard deviation) represents the inner and central part of an object more than the boundary. On the other hand using the HPGF at different scales represents the boundary and exterior parts of an object more than the central part. Our algorithms are also organized to achieve size, translation and rotation invariance. Evaluation indicates that representing the boundary and exterior parts more than the central part using the HPGF performs better than the LPGF based multiscale representation, and in comparison to Zernike moments and elliptic Fourier descriptors with respect to increasing noise. Multiscale description using HPGF in 2-D also outperforms Wavelet transform based multiscale contour Fourier descriptors and performs similar to the perimeter descriptors without any noise.

This Chapter is organized as follows: Section 5.1 discusses related works. Section 5.2 explains the proposed multiscale Fourier-based object description with respect to LPGF and HPGF in 2-D space. Section 5.3 represents classification with multiscale Fourier-based object description. Section 5.4 concerns evaluation and experimental results and finally Section 5.5 is conclusions.

5.1 Related Works

Silhouette based object description and recognition is an important task in computer vision. The descriptor must be invariant to size, translation and rotation, and it must be effective in adverse conditions such as noise and occlusion. There are two main types of shape description methods: boundary-based methods and region-based methods.

5.1.1 Boundary-based Shape Descriptors

In boundary-based methods only the boundary pixels of a shape are taken into account to obtain the shape representation. Boundary-based techniques have some limitations. First, they are generally sensitive to noise and variations of shape, since they only use boundary information. Second, in many cases, the object boundary is not complete with disjoint regions or holes. Region-based methods can overcome these limitations. The most common boundary-based shape descriptors are chain codes [75], Fourier descriptors [1], wavelet descriptors [76], Wavelet-Fourier descriptors [77, 78] and Curvature Scale Space (CSS) [79].

Chain codes [75] are one of the oldest techniques in computer vision introduced for shape description. The algorithm starts with isolating the boundary pixels from a region. Given the boundary, which are connected points, start from one of the boundary pixel and go to the neighbour boundary pixel clockwise that is in one of the major compass direction. Directions can be in 4-way or 8 way connectivity depending on contour isolation. Each direction represents a number that becomes an element of chain code. This is repeated for each pixel until the start point is reached, when the closed shape is completely analysed. By encoding relative direction, rather than the position of the contour pixels, the representation becomes translation invariant. However, the code will be different when the start point changes. Start point invariance can be achieved by considering the code to constitute the digits in an integer. The digits are shifted

cyclically until the smallest integer is obtained. The smallest integer represents the start point invariant chain code. This code is also rotation and scale variant, which means not convenient for matching yet. Rotation invariance is achieved by expressing the code as a difference of chain code. Scale invariance is achieved by resampling the boundary before coding. The chain code is sensitive to noise that may appear because of the segmentation or resampling the boundary.

Shape representation using Fourier descriptors is easy to compute and more robust. Fourier Descriptors are obtained from the Fourier transform on a shape signature. The shape signature is a 1-D function that represents the shape derived from the boundary points of a 2-D binary image. Many shape signatures exist such as, centroid distance, complex coordinates (position function), curvature and cumulative angle [80, 81]. Geometric invariance can be achieved at the shape signature extraction stage or after the Fourier transform by normalizing Fourier coefficients appropriately. This depends on the choice of shape signature type. The Fourier descriptors represent the shape of the object in a frequency domain. The lower frequency descriptors contain information about the general features of the shape and the higher frequency descriptors contain finer details of the shape. In general lower frequency components are selected for description to reduce dimension and since the higher frequency components do not add much to the shape description and are susceptible to noise.

Wavelet descriptors are derived from wavelet transform on a 1-D shape signature. The wavelet transform can be considered as a signal decomposition onto a set of basis functions. It has multiresolution, denoising and feature extraction capabilities. Chang and Kuo [76] used 1-D discrete periodized wavelet transform to describe shapes. However, the matching schema was more complicated than for Fourier descriptors. Kunttu [77, 78] introduces multiscale Fourier descriptors using wavelet and Fourier transforms. The multiscale contour Fourier descriptors are obtained by applying the Fourier transform to the coefficients of the multiscale complex wavelet transform.

Recently, McNeil and Vijayakumar [82] introduced perimeter and radial descriptors. In this work, shapes are represented by a large number of points from their boundaries. These points are selected at fixed intervals in terms of distance along the boundary (perimeter distance) or radial angle. Then, a probabilistic correspondence-based

algorithm, which also incorporates with scale, translation and rotation invariance, is applied for shape matching. They note that suitability of the perimeter distance or radial angle for description depends on the classes in the dataset and these two descriptions can also be combined to improve classification in some datasets. Later on, McNeil and Vijayakumar [83] improved their algorithm by segment-based shape matching, which can overcome limitations of global shape matching such as independent movement of parts or smooth deformations.

Multiscale shape description is the most promising approach for recognition. Different features of the shape can be obtained at different scales and the combination of these features can increase discrimination power, so increasing the correct classification rate. In addition, it is more robust to noise since the dominant features are those which persist across scales. There are many boundary-based multiscale description techniques [77, 79, 84, 85, 86].

One of the most influential techniques is curvature scale space (CSS) introduced by Mokhtarian et al. [79]. This method uses the scale space framework in 1-D space [25]. The boundary of a shape is filtered by LPGF of varying scales (standard deviation). For each specific scale, the locations of those curvatures zero crossings designated as one and otherwise as zero. As the scale decreases, the shape becomes smoother. When the scale is small enough, there will be no curvature zero crossing any more. The binary CSS image is generated by the location and scale in the horizontal and vertical axes respectively. Finally, this binary image is used for matching.

Adamek and O'Connor [84] proposed a multiscale representation for a single closed contour that makes use of both concavities and convexities of all contour points. It is called multiscale convexity concavity (MCC) representation, where different scales are obtained by smoothing the boundary with LPGF of different scales. Then, a new measure for the curvature was proposed that is based on the relative displacement of a contour point with respect to its position in the preceding scale level. This approach is motivated by the observations that when smoothing a closed contour, convex and concave points are moved inside and outside the contour, respectively. The matching is done using a dynamic programming approach.

There are also other boundary-based mutiscale description techniques such as graph-based approach [85] and triangle-area based approach [86].

5.1.2 Region-based Shape Descriptors

In region-based methods, all the pixels within a shape are used to obtain the shape representation. Popular region-based shape descriptors include moments [1, 87] and generic Fourier descriptors (GFDs) [88]. There are different types of moments and they can be classified as non-orthogonal and orthogonal moments depending on the basis function used. Geometric moments [89] are the first and simplest type of moments, which has been used for character recognition. They use non-orthogonal basis functions called a monomial. Low order moments capture global description, while as the order increases, more detail is captured. The main problem with Geometric moments is the high degree of information redundancy, because of the non-orthogonal basis function used. If the basis functions are orthogonal then each moment should highlight independent features. Teague [90] proposed Legendre moments that use Legendre polynomials as basis functions. These polynomials are orthogonal and cause Legendre moments to extract independent features within the image, with no information redundancy. This property also provides good reconstruction capability. These moments are based on Cartesian coordinates but the image function has to be mapped to a specific range of values. Zernike moments were also first proposed by Teague [90] and are based on the complex valued Zernike polynomials. These polynomials are defined in polar coordinates, which help to achieve rotation invariance. Zernike moments were found to be the best performing type of moment in image analysis and description task in terms of noise resilience, information redundancy and reconstruction capabilities [91].

Generic Fourier descriptors (GFD) [88] are other popular region-based shape descriptors. A 2-D Fourier transform is applied on a polar raster sampled shape image. The obtained polar Fourier coefficients are translation invariant. Then rotation and scale invariance are achieved by normalizing these coefficients. GFDs capture features of the shape in both polar and radial directions. GFDs are simple to compute and efficient.

Although many boundary-based multiscale description techniques exist, there is no region-based multiscale description technique in the image space. It is important to note that moments and GFDs are multiscale approach in the feature space, but not in the image space. In our work, we introduce image-based multiscale description using LPGF and HPGF, separately.

5.2 Fourier-based Description with Multiscale Representation in 2-D space

We produce multiscale Fourier-based object descriptors in 2-D space. For this purpose, we investigate the LPGF and the HPGF, separately. The new algorithm starts with size normalization of an object using bilinear interpolation in an image. The object size (the total number intensity value) and the image size are determined experimentally depending on the database to locate each object in the image without occlusion.

Interpolation is the problem of approximating the value for a new data point in some space, within the range of a discrete set of known data points. There are three common image interpolation techniques: Nearest neighbour, bilinear and bicubic interpolation. The nearest neighbour algorithm selects the value of the nearest point, and does not process the values of other neighboring points. It is the simplest algorithm to compute. In bilinear interpolation, image values of the four nearest points are weighted differently based on the distance from the desired point and average of those weighted values give the image value of that desired point. Bilinear interpolation achieves more accurate scaling and can produce smoother edges than nearest neighbour interpolation. However, bilinear interpolation is a little more complex than nearest neighbour algorithm. In bicubic interpolation, output pixel values are calculated from a weighted average of pixels in the sixteen closest neighbourhoods. Bicubic interpolation achieves the best scaling and produces the smoothest edges. However, it is more complex than others and not convenient, when the speed is issue. In our algorithm we choose bilinear interpolation, since it scales better than nearest neighbour interpolation and it is faster than bicubic interpolation.

We also note that it is optional to centralize object in the image, since the next step is 2-D Fourier transform, as given in Equation 5.1, which provides translation invariance.

$$\mathbf{FT}(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\mathbf{I}(x,y)e^{[-j2\pi(ux/M+vy/N)]} \tag{5.1}$$

Where $\mathbf{FT}(u,v)$ is Fourier transform of the silhouette image $\mathbf{I}(x,y)$. $M \times N$ is the size of the silhouette image.

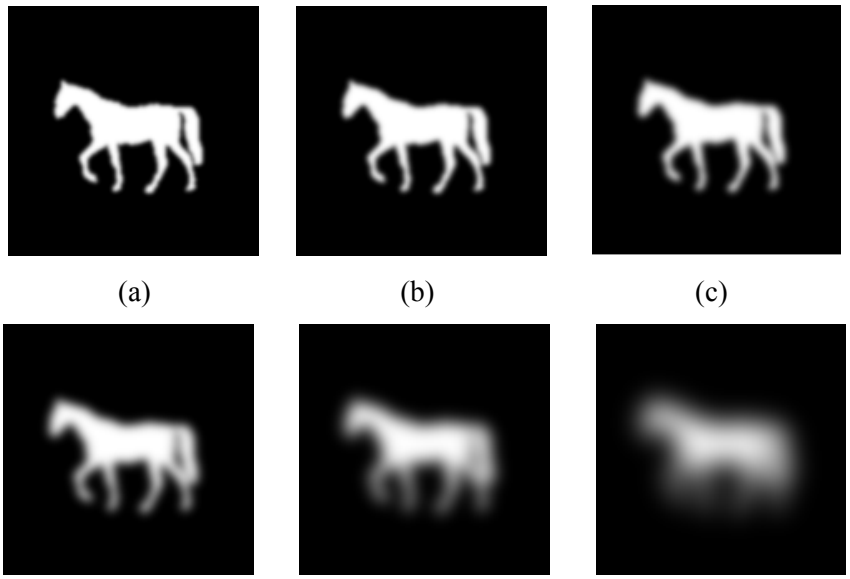
We also note that there is no "windowing" operation before the Fourier transform. The Fourier transform treats an image as it is part of a periodically repeated set of images extending horizontally and vertically to infinity, which can cause strong edges between the neighbours of the periodic image. Therefore, the Fourier transform is the combination of the actual Fourier transform of the given image and that caused by the edge effects at image neighbours. These edge effects can be significantly reduced by using "windowing" operations, which in general makes image values zero towards edges. In our application, the given image is a pre-segmented object on a zero-valued background. Since the object does not occlude image edges, the image values are already zero towards image edges, and there is no need for a "windowing" operation.

In general, the result of the Fourier transform is a complex number and the transform can be represented in terms of its magnitude and phase. The magnitude describes the amount of each frequency component and the phase describes timing, when the frequency components occur. Here, we choose to use the Fourier magnitude image, which is translation invariant. However, the phase also carries considerable information that is discarded here. Oppenheim and Lim [92] showed that if we construct synthetic images from the magnitude information of one image and the phase information of another, we perceive mostly the image corresponding to the phase data. We leave investigation of the phase information as future work and continue with the magnitude information.

The computed Fourier magnitude image, $|\mathbf{FT}(u,v)|$, is translation invariant, however it retains rotation. Given the shift operation (the zero-frequency components are at the centre), multiscale generation is achieved at this stage. To represent the inner and central part of an object more than the boundary, a LPGF with a selection of scale parameters (standard deviation) is applied to the Fourier magnitude image as shown in Equation 5.2.

$$\left|\mathbf{FT}(u,v)\right|^{s} = \left|\mathbf{FT}(u,v)\right|\left(e^{-\left(u^2+v^2\right)/2\sigma_s^2}\right) \tag{5.2}$$

Where $\left|\mathbf{FT}(u,v)\right|^{s}$ and σ_s are Fourier magnitude and scale parameter of scale index s, respectively. This method is generating the scale space [25] of the object in 2-D as shown in Figure 5.1. It is observed that the LPGF smoothes the object and as scale decreases, it causes loss of the boundary and exterior regions. The LPGF emphasizes lower frequency components, but retains some contribution of higher frequency components.



(a) (b) (c)

(d) (e) (f)

Figure 5.1: Horse object filtered by LPGF with respect to decreasing scale. (a) $\sigma_1 = 20$, (b) $\sigma_2 = 15$, (c) $\sigma_3 = 11$, (d) $\sigma_4 = 8$, (e) $\sigma_5 = 5$, (f) $\sigma_6 = 3$.

On the other hand, to represent the boundary and exterior parts of an object more than the central part, a HPGF with a selection of scale parameters (standard deviation) is similarly applied to the Fourier magnitude image as shown in Equation 5.3.

$$\left|\mathbf{FT}(u,v)\right|^{s} = \left|\mathbf{FT}(u,v)\right|\left(1 - e^{-\left(u^2+v^2\right)/2\sigma_s^2}\right) \tag{5.3}$$

Filtering with the HPGF at different scales is illustrated in Figure 5.2. It is observed that the HPGF detects the object boundary and as scale decreases, it represents exterior regions. The HPGF emphasizes higher frequency components, but retains a slight contribution of lower frequency components.

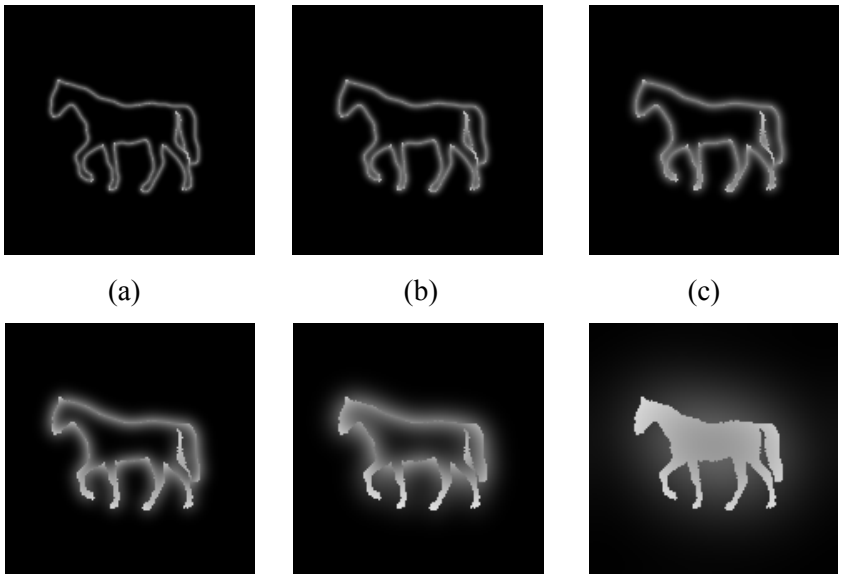


Figure 5.2: Horse object filtered by HPGF with respect to decreasing scale. (a) $\sigma_1 = 15$, (b) $\sigma_2 = 11$, (c) $\sigma_3 = 8$, (d) $\sigma_4 = 5$, (e) $\sigma_5 = 3$, (f) $\sigma_6 = 1$

The obtained Fourier magnitude images are not convenient for matching at this stage, since they still vary with rotation. To remove rotation variance, the coordinates of each Fourier magnitude image are polar mapped to make rotations appear as translations in the new image.

Consider the polar coordinate system (r, θ), where $r \in \Re$ denotes radial distance from the center of the Fourier magnitude image (x_c, y_c) and $0 \le \theta \le 2\pi$ denotes angle. Any point $(x, y) \in \Re^2$ can be represented in polar coordinates as follows

$$\begin{aligned} r &= \sqrt{(x - x_c)^2 + (y - y_c)^2} \\ \theta &= \tan^{-1}\left(\frac{y - y_c}{x - x_c}\right) \end{aligned} \tag{5.4}$$

Equation 5.4 describes conversion from Cartesian to polar coordinates. The reverse process, which is the polar to Cartesian coordinates transform, is defined below,

$$\begin{aligned} x &= r * \cos(\theta) \\ y &= r * \sin(\theta) \end{aligned} \tag{5.5}$$

For every point (x, y), there is a unique point (r, θ). Rotating the Cartesian coordinate system about an origin, while preserving position and size, can be written with the following matrix notation,

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \tag{5.6}$$

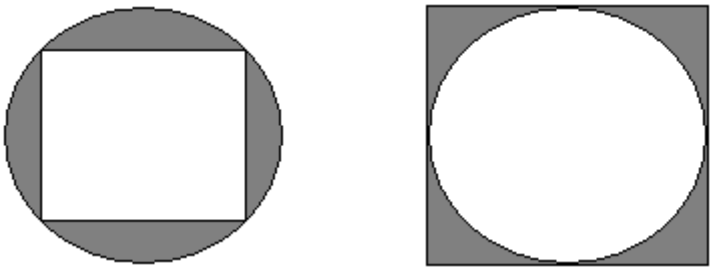
Where, (x_1, y_1) is the point before rotation and (x_2, y_2) is the point after rotation by the angle φ. Assuming that $(x_1, y_1) = (r * \cos(\theta), r * \sin(\theta))$ and after substitution to Equation 5.6, we obtain the new coordinates as

$$\begin{aligned} x_2 &= r * \cos(\theta + \varphi) \\ y_2 &= r * \sin(\theta + \varphi) \end{aligned} \tag{5.7}$$

Here, we can observe that rotation in Cartesian coordinates causes translation in polar coordinates,

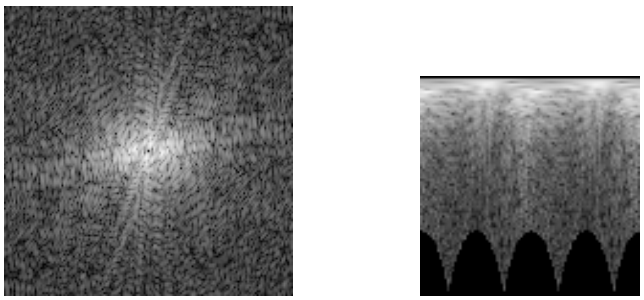
$$\begin{aligned} (x_1, y_1) &\leftrightarrow (r, \theta) \\ (x_2, y_2) &\leftrightarrow (r, \theta + \varphi) \end{aligned} \tag{5.8}$$

There are two principal methods for mapping a rectangular image to a circle in polar transform. The image can either be fitted within the circle as shown in Figure 5.3(a) or the circle can be fitted within the boundaries of the image as shown in Figure 5.3(b). The main problem with fitting circle within the boundaries of the image is losing the information in the corners. Since we want to use all information in the Fourier magnitude image, we use the method that fits the image within a circle. In this method, all pixels will be taken into account but some invalid pixels will also be included, which fall inside the circle but outside the image. In our algorithm these invalid pixel values are set to zero. Figure 5.4 shows the polar transform of a Fourier magnitude image.



(a) (b)

Figure 5.3: Alternative approaches for mapping a square image to the circle. (a) Fitting the image into the circle, where the shaded area shows parts of the circle ignored in the mapping process. (b) Fitting the circle to the square image, where shaded areas represent parts of the image lost in mapping.



(a) (b)

Figure 5.4: Cartesian to polar transform with fitting the image into the circle. (a) Fourier magnitude image of the horse object filtered by HPGF ($\sigma = 3$) and the image size is 151×151. (b) Polar transformed Fourier magnitude image of size 90×90, the invalid pixels are zero.

Finally, another 2-D Fourier transform is applied, as given in Equation 5.9, to compute Fourier magnitude, which removes these translations.

$$\mathbf{FPT}^s(k,l) = \frac{1}{EF} \sum_{r=0}^{C-1} \sum_{\theta=0}^{E-1} \mathbf{P}^s(r,\theta) e^{[-j2\pi(kr/C + l\theta/E)]} \tag{5.9}$$

Where $\mathbf{FPT}^s(k,l)$ is the Fourier transform of the polar mapped image $\mathbf{P}^s(r,\theta)$ of size $C \times E$ and at scale index s. Note that there is no "windowing" operation before the Fourier transform. Although it can remove edge effect between the neighbours of the periodic image, it may also cause loses of some important information in the polar mapped image.

The resultant Fourier magnitude image, $|\mathbf{FPT}(k,l)|^s$, is translation, size and rotation invariant and represents object descriptors $\mathbf{OD}^s$ of a shape at scale index *s*. Figure 5.5 shows the proposed algorithm to obtain multiscale Fourier-based object descriptors.

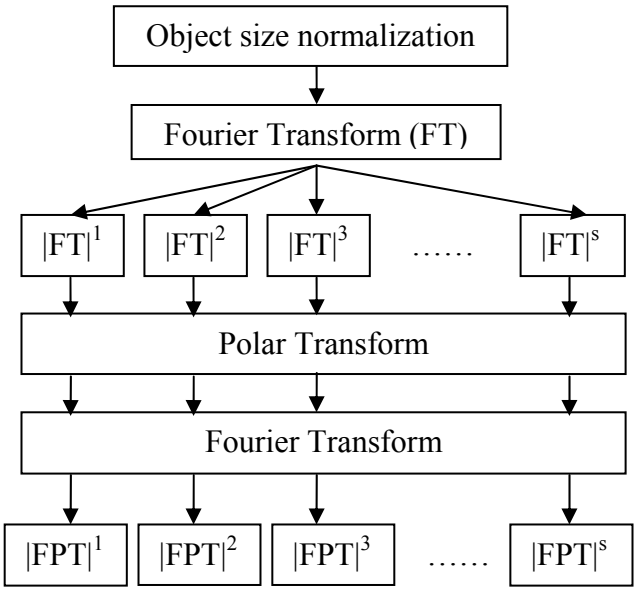


Figure 5.5: Producing the proposed multiscale Fourier-based object descriptors

The Fourier-Mellin transform is similar to our algorithm in terms of achieving rotation, size and translation invariance. The Fourier-Mellin transform is a method for rotation, size and translation invariant image feature extraction in 2-D space [93]. The first stage is a 2-D Fourier transform to calculate the Fourier magnitude image (|FT|), which removes translation variance while keeping scale and rotation variances, then the coordinates are log-polar transformed (LPT) to make scaling and rotation appear as translations, and finally another 2-D Fourier transform is applied to compute Fourier magnitude image (|FLPT|), which remove these translations. Figure 5.6 shows the Fourier-Mellin transform to obtain rotation, size and translation invariant image features. In the log-polar transform, converting scale change to translation is achieved by logarithmic scaling the radius coordinate of the polar map image [94]. The difference from our new approach is that we now have a filtering approach to create a multiscale representation, which must be applied to the objects of same size. Because of this, object size is normalized in the first step and we do not apply logarithmic scaling to the radius coordinate of the polar transformed image.

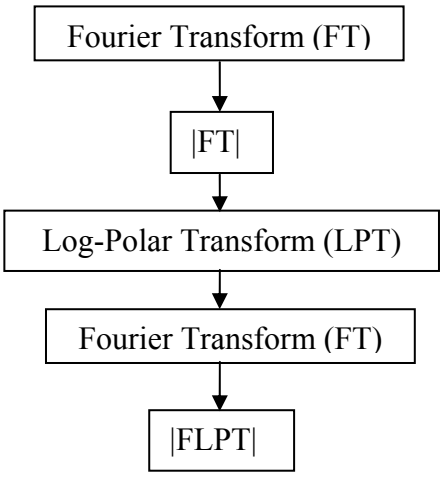


Figure 5.6: Fourier-Mellin transform to produce rotation, size and translation invariant image features.

5.3 Classification with Multiscale Fourier-based Description

Classification is achieved using the nearest neighbour algorithm. We are using a standard approach to allow comparison, though other classifiers are equally appropriate. Euclidean distance (*Ed*) is used to measure similarity between objects and is computed separately in each scale as given below,

$$Ed^{s}(\mathbf{T},\mathbf{D}) = \sqrt{\sum_{x=1}^{C}\sum_{y=1}^{E}\left(\mathbf{OD}_{\mathbf{T}}^{s}(x,y) - \mathbf{OD}_{\mathbf{D}}^{s}(x,y)\right)^{2}} \tag{5.10}$$

Where $Ed^{s}(\mathbf{T},\mathbf{D})$ is the Euclidean distance between the object descriptors, $\mathbf{OD}_{\mathbf{T}}^{s}$, of the test image **T** and object descriptors $\mathbf{OD}_{\mathbf{D}}^{s}$ of an image from database **D**, at scale index s. Then average distance (*Ad*) is computed for each object.

$$Ad = \frac{1}{Y}\sum_{s=1}^{Y} Ed^{s} \tag{5.11}$$

Where Ad represents average distance and Y is the number of scales. Classifying with average distance, instead of single scale distance, increase correct classification as well as increase immunity to noise.

5.4 Evaluations and Experimental Results

For evaluation, we use MPEG-7 CE-Shape-1 Part B database. There are 1400 images in this dataset, which are pre-segmented and are in binary form. The objects are divided into 70 classes with 20 images in each class. The object classes are shown in Figure 5.7. The appearance of these silhouettes changes due to

- viewpoint with respect to objects (size, translation and rotation variance),
- non-rigid object motion (e.g. people walking and fish swimming),
- noise inside shape (e.g. digitization and segmentation noise).

Some objects variations are shown in Figure 5.8. Leave-one-out cross-validation is applied to validate classification. The correct classification rate (CCR%) is measured as follows,

$$CCR(\%) = \frac{c_o}{t_o} \times 100 \tag{5.12}$$

where c_o is the total number of correctly classified objects and t_o is the total number of classified objects.

In evaluation, first we investigate and compare single scales (filtering at different scales) and average distance (with the method given in section 5.3) results of LPGF and HPGF based representation without any noise in silhouette images. Single scales and average distance results are also compared with the original (no filtering) result. Second, we experiment with the original, single scales and average distance performances with respect to increasing noise in the dataset. Finally, LPGF and HPGF based multiscale description (average distance performances) are compared with other object description techniques.

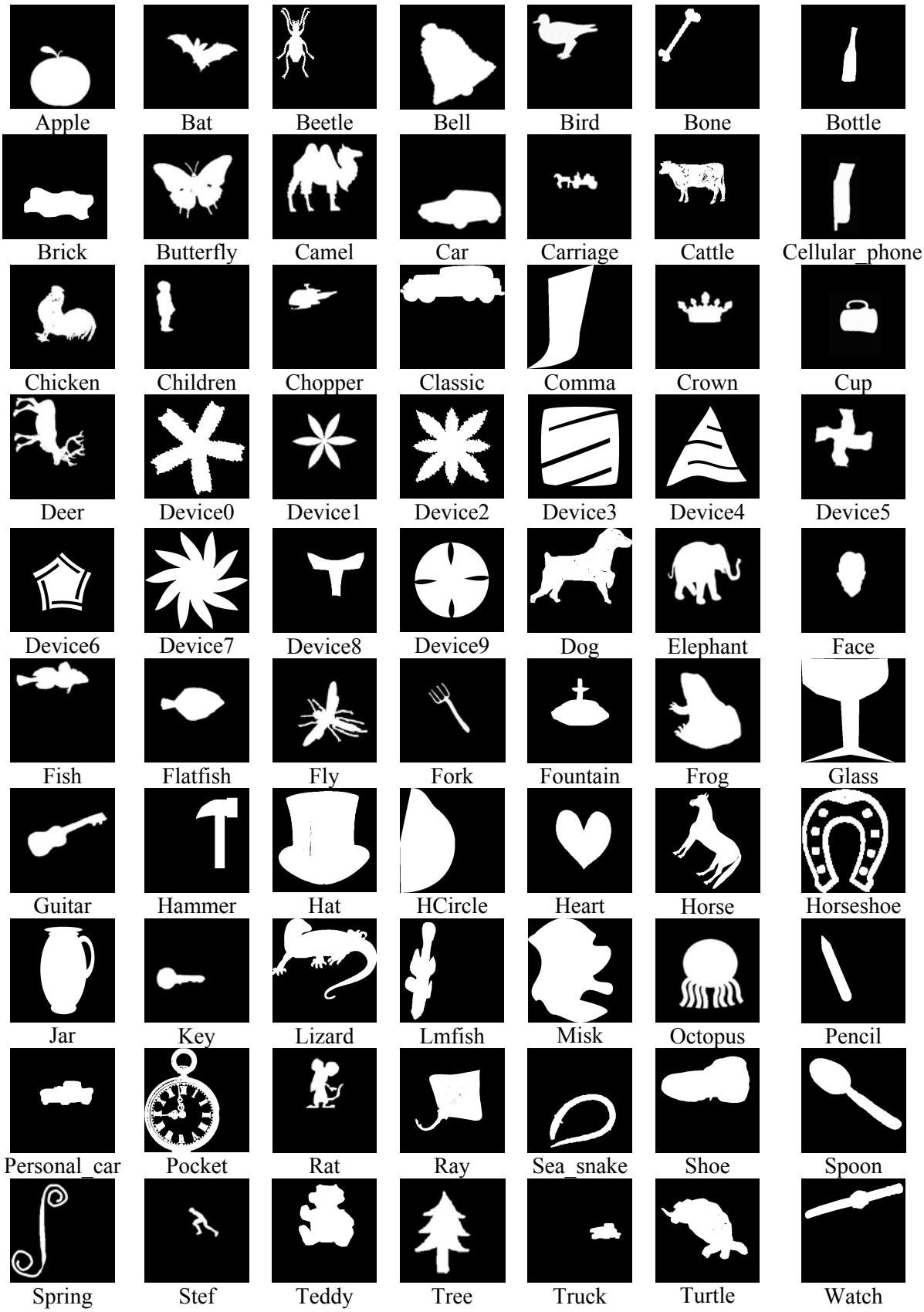
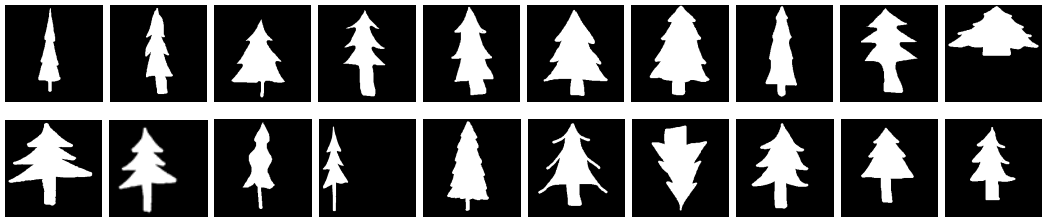
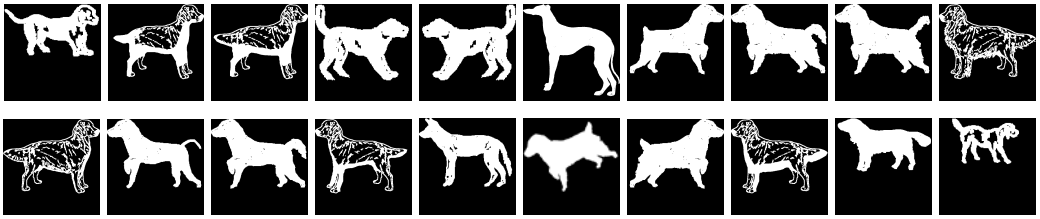


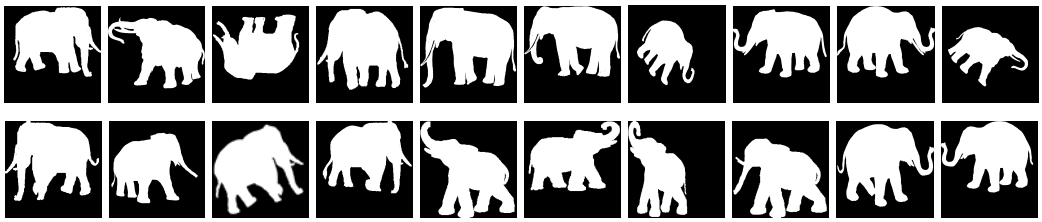
Figure 5.7: A sample from each object class in the database



(a) Tree



(b) Dog



(b) Elephant

Figure 5.8: Some objects variations

5.4.1 Original, single scales and average distance results without any noise in the database

We analyse the original, single scales and average distance performances of LPGF and HPGF based multiscale description without adding any noise to the database. We also remove existing noise in the database by filling object region (using morphological flood-fill operation), since there is noise only inside shapes.

In a multiscale description using LPGF, the object size is normalized to be 2500, which is the total number of intensity value, in a 151×151 size image. 5 different scales are selected for multiscale representation. The selected scales are: $\sigma_1 = 20$, $\sigma_2 = 15$, $\sigma_3 = 11$, $\sigma_4 = 8$ and $\sigma_5 = 5$. The size of the object descriptor matrix is 90×90 at each scale. These 5 scale values are determined experimentally to achieve the best performance of the proposed algorithm with LPGF. Note that as the number of scales increases, the computational complexity increases.

In a multiscale description using HPGF, the object size is similarly normalized to be 2500, which is the total number of intensity value, in a 151×151 size image. 5 different scales are selected for multiscale representation. The selected scales are: $\sigma_1 = 11$, $\sigma_2 = 8$, $\sigma_3 = 5$, $\sigma_4 = 3$ and $\sigma_5 = 1$. The size of the object descriptor matrix is 90×90 at each scale. These 5 scale values are also determined experimentally to achieve the best performance of the proposed algorithm with HPGF. As the number of scales increases, the computational complexity increases.

Table 5.1 shows the CCR% of the original, selected single scales using LPGF and average distance of selected scales. It is observed that the highest CCR% is achieved with the original that is without applying any LPGF. The CCR% of the original is 92.6% and as we apply LPGF with decreasing scales, which means as the objects become smoother, CCR% decreases. Taking average distances from these selected scales, with the method given in section 5.3, results with 91.1%. This is not higher than the original result and some single scale results. Therefore using LPGF is not effective, when there is no noise in the database.

Table 5.1: CCR% of the original, single scales using LPGF and average distance using LPGF

LPGF	*Original*	$\sigma_1 = 20$	$\sigma_2 = 15$	$\sigma_3 = 11$	$\sigma_4 = 8$	$\sigma_5 = 5$	*Average Distance*
CCR%	92.6 % 	92.2 % 	91.7 % 	91.4 % 	90.2 % 	88.5 % 	91.1 %

Table 5.2 similarly shows the CCR% of the original, selected single scales using HPGF and average distance of selected scales. It is observed that applying HPGF with scales $\sigma_3 = 5$, $\sigma_4 = 3$ and $\sigma_5 = 1$ perform better than the original (92.6 %). The highest CCR% is 95 % among the single scale results and is achieved at scale $\sigma_4 = 3$. This is the scale which represents the exterior parts of the object more than the boundary and the central part. The scales $\sigma_5 = 1$ and $\sigma_3 = 5$ give exactly the same result (93.9 %). After $\sigma_3 = 5$, as scale increases, the CCR% decreases. This is because we start to focus more on the boundary alone, which is more sensitive to shape variations. Averaging the

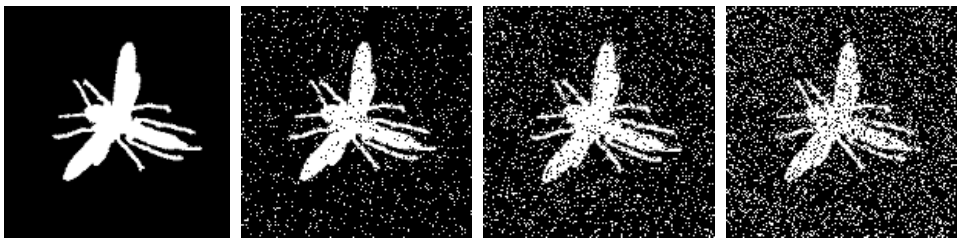
distances of these five scales, which represents the boundary and exterior parts of an object more than the central part, even increase CCR% more and makes it 95.7 %.

Table 5.2: CCR% of the original, single scales using HPGF and average distance using HPGF

HPGF	*Original*	$\sigma_1 = 11$	$\sigma_2 = 8$	$\sigma_3 = 5$	$\sigma_4 = 3$	$\sigma_5 = 1$	*Average Distance*
CCR%	92.6 % 	92.2 % 	92.4 % 	93.9 % 	95 % 	93.9 % 	95.7 %

5.4.2 Original, single scales and average distance results with added noise in the database

We experiment with the original, single scales and average distance performances with respect to increasing salt and pepper noise in the database. Figure 5.9 illustrates salt and pepper noise corrupted binary images with increasing density. In this evaluation, we do not remove the existing noise in the database as well (no region filling). Although some objects in the dataset contain noise inside the shape, adding salt and pepper type noise cause noise outside the shape as well. Salt and pepper noise is added to all objects in the database; therefore noisy test image is matched with the noisy images from database. It is also important to note that the noise is added after the object size normalization stage.



(a) D=0 (b) D=0.1 (c) D=0.2 (d) D=0.3

(e) D=0.4 (f) D=0.5 (c) D=0.6

Figure 5.9: Fly object with increasing density (D) of salt and pepper noise

Table 5.3 and Figure 5.10 show the CCR% of the original images, LPGF filtered images at different scales and the average distance of these scales. The results represent mean values obtained over four applications of each scale at each noise level. In Figure 5.10, the error bar represents minimum and maximum values at the data points. It is simpler to follow our explanations from the table, since the obtained results are very close to each other and cannot be seen well in the figure. It is observed that when there is no noise or small amounts of noise such as D=0.1 and D=0.2, applying LPGF at selected scales does not increase CCR% in comparison to the original. Even averaging the distances with selected scales does not effective. When there is noise more than D=0.2, applying LPGF at higher selected scales ($\sigma_1 = 20$, $\sigma_2 = 15$ and $\sigma_3 = 11$) increase CCR% slightly. Averaging the distances from these selected scales, at noise levels D=0.3 and D=0.4, increases the original as well as the single scales performances slightly. However, at noise levels D=0.5 and D=0.6, we do not observe any increased performance by average distance in comparison to the original and some single sales (higher scales).

Table 5.3: CCR% of the original, single scales using LPGF and average distance using LPGF with respect to the increasing density of salt and pepper noise.

Salt & Pepper noise density (D) / LPGF	0	0.1	0.2	0.3	0.4	0.5	0.6
Original	92.6 %	89.5 %	86.1 %	77.4 %	69.2 %	57.7 %	41.5 %
$\sigma_1 = 20$	91.7 %	88.1 %	84.0 %	78.7 %	69.2 %	59.4 %	44.2 %
$\sigma_2 = 15$	91.2 %	89.4 %	83.5 %	78.2 %	69.7 %	58.7 %	44.5 %
$\sigma_3 = 11$	90.7 %	89.0 %	82.5 %	79.0 %	69.6 %	57.8 %	43.7 %
$\sigma_4 = 8$	89.4 %	86.1 %	81.1 %	76.5 %	65.5 %	56.1 %	39.7 %
$\sigma_5 = 5$	88.1 %	84.3 %	77.2 %	69.2 %	58.7 %	48.1 %	34.4 %
Average Distance	90.4 %	89.5 %	85.2 %	80.5 %	70.3 %	55.0 %	41.5 %

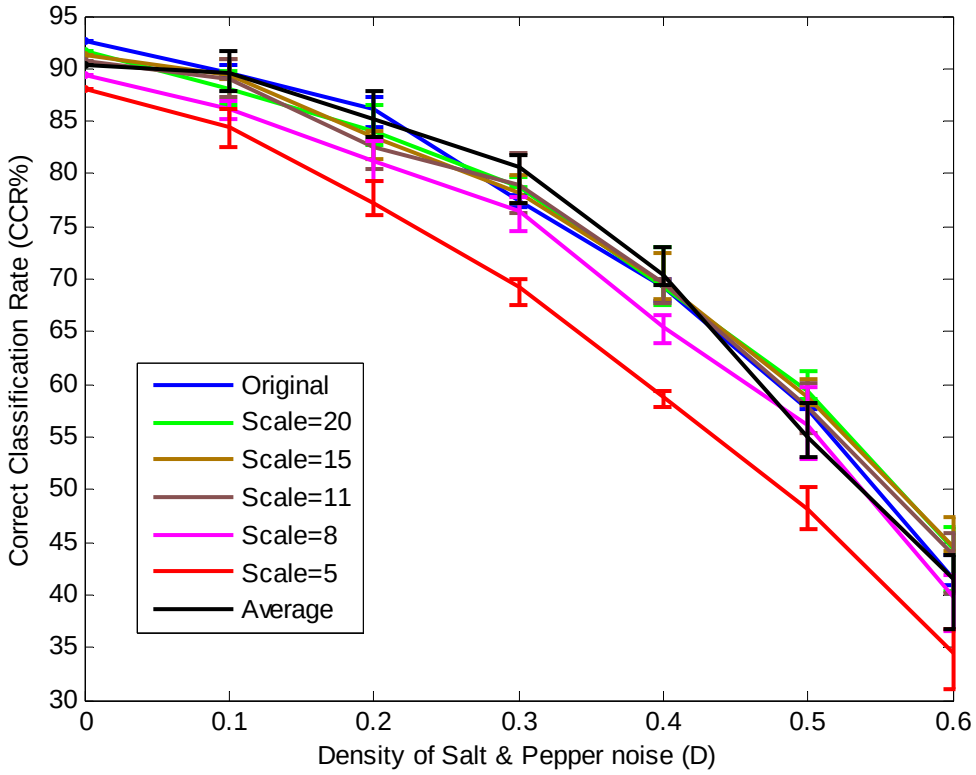


Figure 5.10: Classification performance of the original, the single scales and the average distance by LPGF based representation with respect to increasing salt and pepper noise in the database. CCR% is plotted with minimum and maximum values using error bars.

Table 5.4 and Figure 5.11 show the CCR% of the original image, the single scales using HPGF and the average distance using HPGF based representation, with respect to increasing density of salt and pepper noise. The results represent mean values obtained over four applications of each scale at each noise level. In Figure 5.11, the error bar represents minimum and maximum values at the data points. It is observed that when D=0, lower scales ($\sigma_3 = 5, \sigma_4 = 3$ and $\sigma_5 = 1$) perform better than the original (92.6 %). The best single scale result is achieved at $\sigma_4 = 3$, which is 94.7%. This scale represents the exterior regions of an object more than the boundary and the central part. Averaging the distances of the selected scales also improves the CCR% (95.5 %). When we add salt and pepper noise with increasing density, average distance always performs better than the original and the single scales. Only at D=0.6, which is very noisy and objects are not visible, the scale $\sigma_5 = 1$ performs slightly better than average distance. The scale $\sigma_5 = 1$ also performs better than the original at all noise levels. The scale $\sigma_4 = 3$, which achieves the best result without any added noise, performs better

than original until D=0.3. The performances of the higher selected scales goes down faster than lower selected scales, since the higher selected scales represent the boundary more than the exterior parts and the central part, and more sensitive to noise and shape variations.

Applying HPGF at selected scales and computing average distance improve CCR% in the dataset. This result occurs because of representing more the boundary and the exterior parts, which are more discriminative, than the central part.

Table 5.4: CCR% of the original, the single scales using HPGF and the average distance using HPGF with respect to the increasing density of salt and pepper noise.

Salt & Pepper noise density (D) / HPGF	0	0.1	0.2	0.3	0.4	0.5	0.6
Original	92.6 %	89.5 %	86.1 %	77.4 %	69.2 %	57.7 %	41.5 %
$\sigma_1 = 11$	91.7 %	59.1 %	25.4 %	10.2 %	3.7 %	2.6 %	2.4 %
$\sigma_2 = 8$	92.5 %	78.1 %	48.4 %	23.0 %	11.2 %	5.2 %	2.5 %
$\sigma_3 = 5$	93.8 %	89.3 %	77.4 %	56.7 %	34.7 %	15.2 %	4.7 %
$\sigma_4 = 3$	94.7 %	91.7 %	88.8 %	80.7 %	67.5 %	45.7 %	21.8 %
$\sigma_5 = 1$	93.7 %	92.1 %	90.9 %	86.6 %	80.3 %	69.0 %	52.3 %
Average Distance	95.5 %	93.6 %	92.2 %	88.5 %	82.0 %	71.3 %	52.0 %

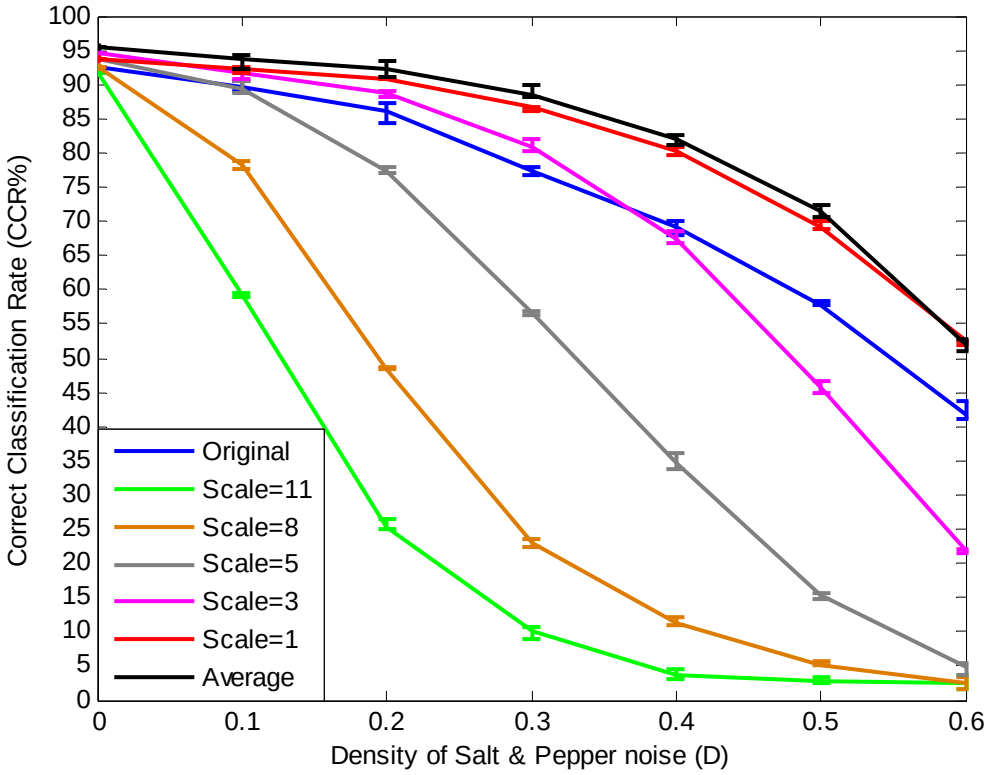


Figure 5.11: Classification performance of the original, the single scales and the average distance by HPGF based description with respect to increasing salt and pepper noise in the database. CCR% is plotted with minimum and maximum values using error bars.

5.4.3 Comparison with other techniques

Performance evaluation is also employed by comparing the multiscale description using LPGF (average distance) and multiscale description using HPGF (average distance) with each other as well as with elliptic Fourier descriptors (EFD) and Zernike moments (ZM). The evaluation is again achieved with respect to increasing salt and pepper noise in the database, and the noisy test image is matched with the noisy images from the database.

EFD are fast and robust boundary-based shape descriptors. The contour is represented with complex coordinates (position function) and then the Fourier expansion is performed to obtain the EFD, where the number of descriptors is 80 in this evaluation. To evaluate EFD, we use a Matlab implementation given in [95], and note that this is a non-optimal Matlab framework. We describe the boundary of the biggest region in the image, since there will be many regions after noise has been added.

Zernike moments (ZM) are region-based shape descriptors. They are an orthogonal moment set, which makes optimal utilization of shape information and allows accurate recognition. It is a potent moment technique for shape description [96]. To evaluate ZM, we use the algorithm given in [96], which uses 36 moments for description. We use a Matlab implementation given in [97] that is also a non-optimal framework.

Figure 5.12 and Table 5.5 show the correct classification rate (CCR%) of the multiscale description in 2-D using LPGF, of multiscale description in 2-D using HPGF, of EFD and of ZM, with respect to increasing salt and pepper noise. The results represent mean values obtained over four applications of each algorithm at each noise level. In Figure 5.12, the rectangle on graphs represents standard deviation from the mean value and error bar represents minimum and maximum values at data point. It is observed that HPGF based multiscale description performs better than LPGF based multiscale description, EFD and ZM. HPGF based multiscale description achieves 95.5% correct classification rate, while LPGF based multiscale description achieves 90.4%, ZM achieves 90% and EFD achieves 82% without adding noise to the database. As noise increases, the performance of all algorithms decreases and their performances degrade similarly. It is also observed that LPGF based multiscale description and ZM have very close performances. The success of HPGF based multiscale description in 2-D appears due to emphasizing the boundary and exterior parts of objects and also allowing the central part contribute slightly to classification. Although we use a multiscale representation (LPGF or HPGF), the proposed algorithm is easer to compute in comparison to Zernike moments. To obtain the Zernike moments, Zernike polynomials are computed, which are difficult and complex. On the other hand, in our algorithm, we rely on a polar transform and two Fourier transforms that are computed by the Fast Fourier Transform.

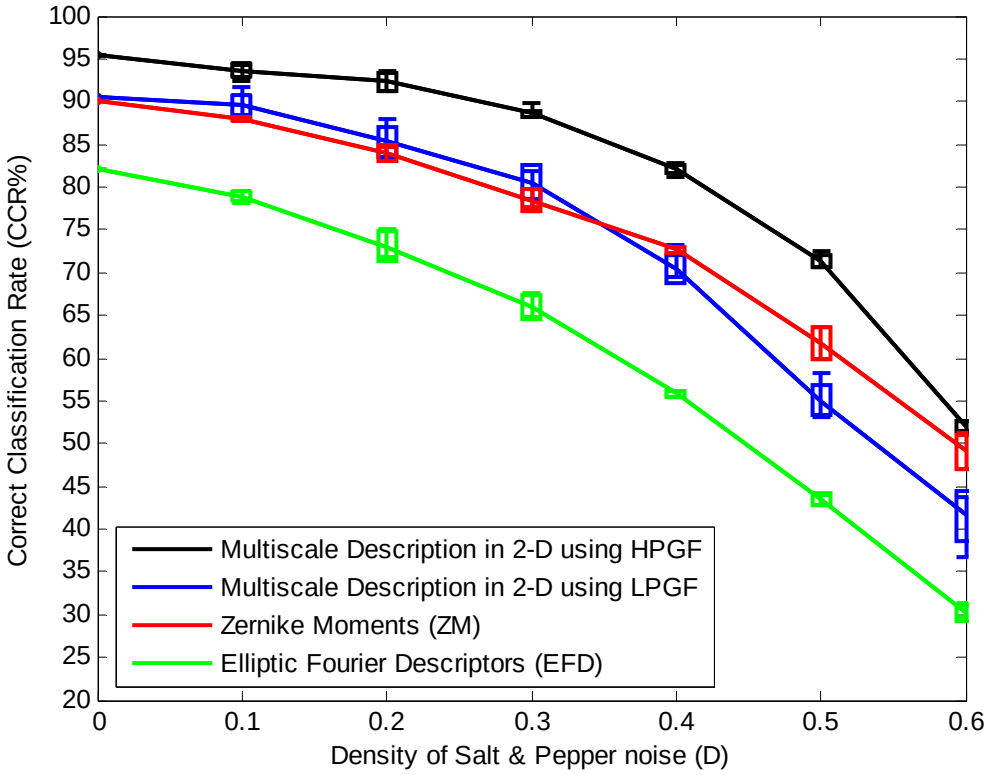


Figure 5.12: Classification performance of HPGF and LPGF based multiscale description in 2-D, ZM and EFD, with respect to increasing salt and pepper noise in the database. In graphs, the rectangle represents standard deviation from the mean value and error bar represents minimum and maximum values of the CCR%.

Table 5.5: CCR% of HPGF and LPGF based multiscale description in 2-D, ZM and EFD, with respect to increasing salt and pepper noise in the database.

Descriptions \ Salt & Pepper noise density (D)	0	0.1	0.2	0.3	0.4	0.5	0.6
Multiscale Description using HPGF	95.5 %	93.6 %	92.2 %	88.5 %	82.0 %	71.3 %	52.0 %
Multiscale Description using LPGF	90.4 %	89.5 %	85.2 %	80.5 %	70.3 %	55.0 %	41.5 %
Zernike Moments (ZM)	90.0 %	87.9 %	83.9 %	78.4 %	72.6 %	61.7 %	49.0 %
Elliptic Fourier Descriptors (EFD)	82.0 %	78.9 %	73.0 %	65.9 %	55.8 %	43.5 %	30.2 %

There are also other techniques, which used same database (MPEG-7 CE-Shape-1 Part B) for classification purpose. A subset of this shape database was used by Kunttu et al. [77]. Their descriptors are wavelet transform based multiscale contour Fourier descriptors, which is obtained by applying the Fourier transform to the coefficients of the multiscale complex wavelet transform. They applied classification for 30 classes without any noise in the dataset. The selected classes are: bone, bottle, brick, car, cellular phone, children, chopper, comma, deer, device0, device1, device2, device7, device8, face, fish, fountain, frog, glass, heart, key, lmfish, misk, octopus, pencil, personal car, pocket, shoe, teddy and truck. Using the leave-one-out classification with a nearest neighbour classifier, they achieve 94.2-96.3 % with respect to the length of descriptors.

Same subset was also recently used by McNeil and Vijayakumar [82] for classification without any noise in the dataset. In their work, shape boundary is represented with a large number of equally spaced points either defined by perimeter distance (perimeter descriptors) or radial angle (radial descriptors). Then, a probabilistic correspondence-based algorithm, which also incorporates with scale, translation and rotation invariance, is applied for shape matching. They note that suitability of the perimeter distance or radial angle for description depends on the classes in the dataset and these two descriptions can also be combined to improve classification in some datasets. They used the same testing procedure, leave-one-out classification with a nearest neighbour classifier, to compare with the wavelet-based multiscale contour Fourier descriptors described above. They only show the results of the perimeter descriptors, which performs 95.6-98.0 % with respect to the number of points selected on the boundary. They also evaluated their descriptors on the full dataset, which includes 70 classes, without any noise in the dataset. They achieved 95.7 % and 91.0 % with perimeter descriptors and radial descriptors respectively. They also combined perimeter and radial descriptors and achieved 96.2 % classification accuracy on the full dataset.

On the other hand, HPGF based multiscale description in 2-D achieves 99.2 % on the same subset using the leave-one-out classification with a nearest neighbour classifier. This result show that our algorithm, with HPGF based mustiscale description, outperforms both perimeter descriptors and multiscale contour Fourier descriptors on the subset. We also evaluated other algorithms on the same subset and observe that

LPGF based multiscale description in 2-D achieves 95.8 %, ZM achieves 92.6 %, and EFD achieves 87.8 %. Table 5.6 shows CCR% of the algorithms on the subset without any noise.

Table 5.6: CCR% of multiscale contour Fourier descriptors [77], perimeter descriptors [82], HPGF and LPGF based multiscale description in 2-D, ZM and EFD on the subset (30 classes) without any noise.

Descriptors	CCR %
Multiscale Description using HPGF in 2-D	99.2 %
Perimeter Descriptors [82]	95.6-98.0 %
Wavelet-based multiscale Contour Fourier Descriptors [77]	94.2-96.3 %
Multiscale Description using LPGF in 2-D	95.8 %
Zernike Moments (ZM)	92.6 %
Elliptic Fourier Descriptors (EFD)	87.8 %

On the full dataset, HPGF based multiscale description in 2-D achieves 95.7 %, which is better than radial descriptors, same as perimeter descriptors, and slightly less then the combined perimeter and radial descriptors. LPGF based multiscale description in 2-D achieves 91.1 %, ZM achieves 90.2 %, and EFD achieves 82 %. Table 5.7 shows CCR% of the algorithms on the full dataset without any noise.

Table 5.7: CCR% of perimeter descriptors, radial descriptors, combined perimeter and radial descriptors [82]. HPGF and LPGF based multiscale description in 2-D, ZM and EFD on the full dataset (70 classes) without any noise.

Descriptors	CCR %
Combined Perimeter-Radial Descriptors [82]	96.2 %
Multiscale Description using HPGF in 2-D	95.7 %
Perimeter Descriptors [82]	95.7 %
Multiscale Description using LPGF in 2-D	91.1 %
Radial Descriptors [82]	91.0 %
Zernike Moments (ZM)	90.2 %
Elliptic Fourier Descriptors (EFD)	82.0 %

5.5 Conclusions and Future Work

We have presented novel multiscale Fourier-based object descriptors in 2-D space for the purpose of recognition. We have used a low-pass Gaussian filter (LPGF) and a high-pass Gaussian filter (HPGF) for multiscale generation. Using the LPGF at different scales represents the inner and central part of an object more than the boundary. On the other hand using the HPGF at different scales represents the boundary and exterior parts of an object more than the central part. Our algorithm starts with object size normalization and we then compute a Fourier magnitude image that is translation invariant. At this stage, a LPGF or a HPGF with a selection of scale parameters is used to obtain multiscale Fourier magnitude images. To give rotation invariance, each image of different scale is polar mapped and then another Fourier magnitude image is computed to obtain the proposed object descriptors. For classification, the Euclidean distance is measured separately at each scale, and then the average distance is computed for each object. Multiscale description using HPGF, which represents the boundary and exterior parts of an object more than the central

part, outperforms multiscale description using LPGF, elliptic Fourier descriptors (EFD) and Zernike moments (ZM) with respect to increasing salt and pepper noise in the database. Multiscale description using HPGF in 2-D also performs better than Wavelet transform based multiscale contour Fourier descriptors and performs similar to the perimeter descriptors without any noise in the dataset. Classifying objects with this new multiscale Fourier-based object description using the HPGF in 2-D space increases immunity to noise and discrimination power.

In the future work, we can find persistent features over scales using feature set selection, which may increase discrimination and also reduce the number of features. A new classifier can be used instead of nearest neighbour classifier to increase correct classification rate. The proposed object description can be applied for the image retrieval purposes, as well as more evaluation can be done on different databases both for classification and retrieval purposes. In addition, we can also investigate phase information of the Fourier transforms, which is currently discarded in our algorithm. The phase has significant information about the image and it could be beneficial to include it in object description.

Chapter 6 Conclusions and Future Work

6.1 Conclusions

In this thesis, we have presented three different contributions to feature extraction based on using the heat conduction analogy. The solution of the heat conduction equation depends on properties of material, heat source as well as specified initial and boundary conditions. In our contributions, we represent particular heat conduction problems in the image and video domains for feature extraction. The first contribution is moving-edge detection that is a low-level feature extraction. The second contribution is shape extraction from images, which is a high-level feature extraction. The final contribution is silhouette object feature extraction for recognition purpose and this can be considered as a combination of low-level and high-level feature extraction.

In the first area of our research, we introduce a novel moving object edge detection technique. In this work, we first solve an anisotropic heat conduction problem in the spatial domain to remove noise and sharpen region boundaries for the purpose of obtaining high quality Sobel edge maps. Then, linear isotropic heat flow is applied in the temporal domain, with the proposed initial and boundary conditions, to calculate the total amount of heat flow. The moving-edge map is represented as the total amount of heat flow out from the reference frame. The overall process is completed by non-maxima suppression and hysteresis thresholding to obtain binary moving-edges. Evaluation indicates that this technique performs better than 2-D Sobel at all levels of Gaussian noise, without anisotropic heat diffusion in space. This result appears to be due to the averaging inherent in the new operator. Results also show that this technique

can detect moving-edges in image sequences, without using background subtraction. As such, the efficient combination of anisotropic and isotropic heat equations, with the proposed initial and boundary conditions, is applied in a new way to detect the edges of moving object.

In the second area of our research, a novel evolution-based segmentation algorithm is introduced using the heat flow analogy. We extract objects shapes by representing particular heat conduction problems in the image domain. First, we represent and solve a linear and non-homogeneous heat equation in the image domain to roughly segment objects of interest. Once the heat source is located at the region of interest, the segmentation is achieved by a temperature front (TF), which moves with the heat diffusing from the source. The representation in image domain is achieved by using a control function (CF) in the heat conduction equation. This formulation also provides advantage when the image is bimodal, since the CF attempts to segment whole image in this case, while the TF segments the region of source location. In the second part, an anisotropic diffusion, geometric heat flow (GHF), is applied to smooth the extracted boundaries and remove possible noise arising from the first part of the segmentation. Our algorithm is evaluated by comparing with popular active contour models, which are gradient vector flow snake (GVFS) and active contour without edges (ACWE). These models can also be defined as gradient-based parametric active contour (PAC) and region-based geometric active contour (GAC) respectively. Evaluation results indicate that TF+GHF can provide better performance than GVFS and ACWE with respect to increasing Gaussian noise. For the bimodal images, CF+GHF also has better performance than ACWE. These improvements are achieved by effective organization of our algorithm, TF+GHF segments better than GVFS by its basis on region-based information instead of gradient-based, which is sensitive to noise. TF+GHF and CF+GHF segment better than ACWE because of the smoothing operation of shape. TF+GHF and CF+GHF apply smoothing after rough segmentation without any relation to the regional statistic constraints, while ACWE uses smoothness constraint with regional statistic constraints during the segmentation. In addition, TF+GHF and CF+GHF are computationally more efficient and effective than both GVFS and ACWE based on the simulation results. Especially, there are big differences between our model and ACWE in computational efficiency. The main reason is the complexity of the level set method in ACWE. As a result, our approach works well in noisy conditions and it is

also computationally more efficient and easer to control and implement in comparison with active contour models. As such, the proposed heat conduction problems can be applied in the image domain with success for shape extraction.

In the third area of this research, we introduce new and efficient silhouette object descriptors by using the Fourier series and Gaussian filtering for recognition purposes. We investigate Fourier series and Gaussian filtering for feature extraction, since they are related to the solution of the linear and homogenous heat conduction problems. Analytic solution of the linear and homogeneous heat conduction problem can be achieved using Fourier series theory. In addition, solving the linear and homogeneous heat equation in infinite medium can be viewed as smoothing a signal with a low-pass Gaussian filter (LPGF). We develop new multiscale Fourier-based object description in 2-D space using a low-pass Gaussian filter (LPGF) and a high-pass Gaussian filter (HPGF), separately. A multiscale description provides more information about the object, increases discrimination power and immunity to noise. Using the LPGF at different scales (standard deviation) represents the inner and central part of an object more than the boundary. On the other hand using the HPGF at different scales represents the boundary and exterior parts of an object more than the central part. Our algorithm starts with object size normalization and we then compute a Fourier magnitude image that is translation invariant. At this stage, a LPGF or a HPGF with a selection of scale parameters is used to obtain multiscale Fourier magnitude images. To give rotation invariance, each image at a different scale is polar mapped and then another Fourier magnitude image is computed to obtain the proposed object descriptors. For classification, the Euclidean distance is measured separately at each scale, and then the average distance is computed for each object. Multiscale description using HPGF, which represents the boundary and exterior parts of an object more than the central part, outperforms multiscale description using LPGF, elliptic Fourier descriptors (EFD) and Zernike moments (ZM) with respect to increasing salt and pepper noise in the database. Multiscale description using HPGF in 2-D also performs better than Wavelet transform based multiscale contour Fourier descriptors and performs similar to the perimeter descriptors without any noise in the dataset. Classifying objects with this new multiscale Fourier-based object description using the HPGF in 2-D space increases immunity to noise and discrimination power.

6.2 Future Work

We have future work in the third area of this research, which is multiscale Fourier-based object description in 2-D space using a low-pass Gaussian filter (LPGF) and a high-pass Gaussian filter (HPGF). In the future work, we can find persistent features over the selected scales to increase discrimination and also reduce the number of features. A new classifier can be used instead of nearest neighbour classifier to increase correct classification rate. The proposed object description can be applied for the image retrieval purposes, as well as more evaluation can be done on different databases both for classification and retrieval purposes. In addition, we can also investigate phase information of the Fourier transforms, which is currently discarded in our algorithm. The phase has significant information about the image and it could be beneficial to include it in object description.

In overall conclusion, the heat analogy has been deployed both for low-level and high-level feature extraction purposes in image processing and computer vision successfully. It has given new insight to the feature extraction process and has led to techniques with attractive performance in terms of segmentation and computational requirement.

References

[1] M. S. Nixon and A. Aguado. Feature Extraction and Image Processing, 2nd Edition, Elsevier, 2007.

[2] D. J. Hurley, M. S. Nixon and J. N. Carter. A New Force Field Transform for Ear and Face Recognition, *IEEE International Conference on Image Processing*, 1: 25-28, 2000.

[3] D. J. Hurley, M. S. Nixon and J. N. Carter. Force Field Feature Extraction for Ear Biometrics, *Computer Vision and Image Understanding*, 98(3):491-512, 2005.

[4] X. U. Liu and M. S. Nixon. Water Flow based Complex Feature Extraction, *International conference on Advanced Concepts for Intelligent Vision Systems,* LNCS 4179:833-845, 2006.

[5] X. U. Liu and M. S. Nixon. Image and Volume Segmentation by Water Flow, *International Symposium on Visual Computing,* LNCS 4842:62-74, 2007.

[6] M. S. Nixon, X. U. Liu, C. Direkoğlu and D. J. Hurley. On Using Physical Analogies for Feature and Shape Extraction in Computer Vision, *1st British Computer Society Visions of Computer Science Conference,* 2008.

[7] Mark S. Nixon, Xin U. Liu, Cem Direkoğlu and David J. Hurley. On Using Physical Analogies for Feature and Shape Extraction in Computer Vision, *British Computer Society, The Computer Journal,* 2008, submitted.

[8] P. Perona and J. Malik. Scale-Space and Edge Detection using Anisotropic Diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8): 629-639, 1990.

[9] S. Manay and A. Yezzi. Anti-Geometric Diffusion for Adaptive Thresholding and Fast Segmentation. *IEEE Transaction on Image Processing*. 12(11):1310-1323, 2003.

[10] X. Ji and J. Feng. A New Approach to Thinning Based on Time-Reversed Heat Conduction Model, *IEEE International Conference on Image Processing*, 1:653-656, 2004.

[11] V. Caselles, F. Catte, T. Coll and F. Dibos. A Geometric Model for Active Contours. *Numerische Mathematic*, 66:1-31, 1993.

[12] R. Malladi, J.A. Sethian and B.C. Vemuri. Shape Modeling with Front Propagation: A Level Set Approach. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(2):158-175, 1995.

[13] S. Osher and J. A. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computation Physics*, 79(1):12-49, 1988.

[14] B. B. Kimia and K. Siddiqi. Geometric Heat Equation and Nonlinear Diffusion of Shapes and Images. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages: 113-120, 1994.

[15] S. K. Makrogiannis and N. G. Bourbakis. Motion Analysis with Application to Assistive Vision Technology, *IEEE International Conference on Tools with Artificial Intelligence*, pages: 344-352, 2004.

[16] J. Fourier, translated by A. Freeman. *The Analytical Theory of Heat*. Dover Publications, published in 1822, translated in 1878 and re-released 2003.

[17] C. Direkoğlu and M. S. Nixon. Low Level Moving-Feature Extraction via Heat Flow Analogy, *International Symposium on Visual Computing,* LNCS 4291:243-252, 2006.

[18] C. Direkoğlu and M. S. Nixon. Shape Extraction via Heat Flow Analogy, *International conference on Advanced Concepts for Intelligent Vision,* LNCS 4678:553-564, 2007.

[19] R. V. Churchill. Fourier Series and Boundary Value Problems, McGraw-Hill, 2nd edition, 1963.

[20] M. N. Ozisik. Heat Conduction, John Wiley & Sons, 1st edition, 1980.

[21] C. Direkoğlu and M. S. Nixon. Shape Classification using Multiscale Fourier-based Description in 2-D Space, *IEEE International Conference on Signal Processing*, 1:820-823, 2008.

[22] C. Direkoğlu and M. S. Nixon. Image-based Multiscale Shape Description using Gaussian Filter, *IEEE Indian Conference on Computer Vision, Graphics and Image Processing,* pages:673-678*,* 2008.

[23] J. P. Holman. Heat Transfer, McGraw-Hill, 9th edition, 2002.

[24] M. N. Ozisik. Basic Heat Transfer, McGraw-Hill, 1977.

[25] A. Witkin. Scale-Space Filtering, *International Joint Conference on Artificial Intelligence*, pages:1019-1021, 1983.

[26] J. Koenderink. The Structure of Images, *Biological Cybernetics*, 50:363-370, 1984.

[27] R. Hummel, Representations based on zero-crossings in scale-space, *IEEE International Conference on Computer vision and Pattern Recognition*, pages: 204-209, 1986.

[28] M. J. Black, G. Sapiro and D. H. Marimont. Robust Anisotropic Diffusion, *IEEE Transaction on Image Processing*, 10(2):421-432, 1998.

[29] F. Zhang and E. R. Hancock. Image Scale-Space from Heat Kernel, *Iberoamerican Congress on Pattern Recognition CIARP*, LNCS 3773:181-192, 2005.

[30] F. Zhang and E. R. Hancock. Heat kernel smoothing of scalar and vector image data. *IEEE International Conference on Image Processing*, pages:1549-1552, 2006.

[31] H. Krim and Y.Bao. Nonlinear Diffusion: A Probabilistic View, *International Conference on Image Processing*, 2:21-25, 1999.

[32] O. Scherzer and J. Weickert. Relations between Regularization and Diffusion Filtering, *Journal of Mathematical Imaging and Vision*, 12:43-63, 2000.

[33] C.J. Sze and H.Y.M. Liao. A New Image Flux Conduction Model and Its Application to Selective Image Smoothing, *IEEE Transaction on Image Processing*, 10(2):296-306, 2001.

[34] M. F. Auclair-Fortier and D. Ziou. A Global Approach for Solving Evolutive Heat Transfer for Image Denoising and Inpainting, *IEEE Transaction on Image Processing*, 15(9):2558-2574, 2006.

[35] V. P. Namboodiri and S. Chaudhuri. Image Restoration using Geometrically Stabilized Reverse Heat Equation, *IEEE International Conference on Image Processing*, 4:413-416, 2007.

[36] A. Robles-Kelly and E. R. Hancock. Vector field smoothing via Heat Flow, *IEEE International Conference on Pattern Recognition*, 2:94-97, 2004.

[37] A. Robles-Kelly and E. R. Hancock. Surface Height Recovery using Heat Flow and Manifold Embedding, *International Symposium on 3D Data Processing, Visualization, and Transmission*, pages: 860-867, 2004.

[38] A. Robles-Kelly and E. R. Hancock. Shape-From-Shading using the Heat Equation, *IEEE Transactions on Image Processing*, 16(1):7-21, 2007.

[39] S.T. Acton, A.C. Bovik and M.M. Crawford, Anisotropic Diffusion Pyramids for Image Segmentation, *IEEE International Conference on Image Processing*, 3:478-482, 1994.

[40] M. Kass, A. Witkin and D. Terzopoulos. Snakes: Active Contour models, *International Journal of Computer Vision*, pages:321-331, 1987.

[41] C. Xu and J. L. Prince. Snakes, Shapes and Gradient Vector Flow, *IEEE Transaction on Image Processing*, 7(3):359-369, 1998.

[42] B. Horn and B. Schunck, Determining Optical Flow, *Artificial Intelligence*, 17:185-204, 1981.

[43] J. Weickert and C. Schnorr, Variational Optic Flow Computation with a Spatio-Temporal Smoothness Constraint, *Journal of Mathematical Imaging and Vision*, 14:245-255, 2001.

[44] L. Alvarez, J. Weickert and J. Sanchez. A Scale-Space Approach to Nonlocal Optical Flow Calculations, *International Conference on Scale-Space theories in Computer Vision*, 1682:235-246, 1999.

[45] B. Xiao and E. R. Hancock. Heat Kernels, Manifolds and Graph Embedding. *Structural, Syntactic, and Statistical Pattern Recognition, Proceedings*, LNCS 3138:198-206, 2004.

[46] B. Xiao and E. R. Hancock. Graph clustering using heat content invariants. *Iberian Conference on Pattern Recognition and Image Analysis*, LNCS 3523:123-130, 2005.

[47] B. Xiao and E. R. Hancock. Clustering shapes using heat content invariants. *IEEE International Conference on Image Processing,* 1:1169-1172, 2005.

[48] F. Zhang, H. J. Qiu, and E. R. Hancock. Evolving Spanning Trees using The Heat Equation. *Computer Analysis of Images and Patterns*, LNCS 3691:272-279, 2005.

[49] H. F. Wang and E. R. Hancock. Probabilistic relaxation using the heat equation. *International Conference on Pattern Recognition*, 2:666-669, 2006.

[50] C. Kim and J. N. Hwang. Fast and Automatic Video Object Segmentation and Tracking for Content-based Applications, *IEEE Transactions on Circuits and Systems for Video Technology,* 12(2):122-129, 2002.

[51] Y. Ahn, K. Ahn and O. Chae. Detection of Moving Objects Edges to Implement Home security System in a Wireless Environment. *International Conference on Computer Science and Its Applications,* LNCS 3043:1044-1051, 2004.

[52] P. J. Myerscough ans M, S, Nixon. Temporal Phase Congruency. *IEEE Southwest Symposium on Image Analysis and Interpretation,* pages:76-79, 2004.

[53] S. H. Kim. A Novel Approach to Moving Edge Detection Using Cross Entropy. *International Conference on Graphics, Vision and Image Processing*, pages:21-24, 2005.

[54] X. Y. Zhang and R. C. Zhao. Automatic Video Object Segmentation Using Wavelet Transform and Moving Edge Detection, *IEEE International Conference on Machine Learning and Cybernetics*, pages:1174-1177, 2006.

[55] M. A. A. Dewan, M. J. Hossain, O. Chae. Reference Independent Moving Object Detection: An Edge Segment Based Approach, *International Conference on Knowladge-based and Intelligent Information & Engineering System,* LNAI 4692:501-509, 2007.

[56] J. F. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679-698, 1986.

[57] L. D. Cohen. On Active Contours Models and Balloons. *CVGIP: Image Understanding*, 53(2):211–218, 1991.

[58] V. Caselles, R. Kimmel and G. Sapiro. Geodesic Active Contours, *International Journal of Computer Vision*, 22(1):61-79, 1997.

[59] K. Siddiqi, Y. Lauziere, A. Tannenbaum and S. Zucker. Area and Length Minimizing Flows for Shape Segmentation, *IEEE Transaction on Image Processing*, 7(3):433-443, 1998.

[60] X. Xie and M. Mirmehdi. RAGS: Region-Aided Geometric Snake, *IEEE Transaction on Image Processing*, 13(5):640-652, 2004.

[61] D. Adalsteinsson and J. Sethian. A Fast Level Set Method for Propagating Interfaces, *Journal of Computational Physics*, 118(2):269-277, 1995.

[62] J. Sethian. Level Set Methods and Fast Marching Methods. New York: Cambridge Univ. press, 1999.

[63] J. Weickert, B. M. T. H. Romeny and M. A. Viergever. Efficient and Reliable Schemes for Nonlinear Diffusion Filtering, *IEEE Transactions on Image Processing*, 7(3):398-410, 1998.

[64] T. Chan and L. Vese. Active Contours without Edges, *IEEE Transaction on Image Processing*, 10(2):266-277, 2001.

[65] D. Mumford and J. Shah. Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems. *Communications on Pure and Applied Mathematics*, 42(4):577-685, 1989.

[66] L. A. Vese and T. F. Chan. A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model. *International Journal of Computer Vision*, 50(3):271-293, 2002.

[67] C. Zimmer and J. C. Olivo-Marin. Coupled Parametric Active Contours, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(11):1838-1842, 2005.

[68] N. Paragios and R. Deriche. Coupled Geodesic Active Regions for Image Segmentation : A Level Set Approach, *European Conference on Computer vision*, pages:224-240, 2000.

[69] X. Xie and M. Mirmehdi. MAC: Magnetostatic Active Contour Model, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):632-646, 2008.

[70] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641-647, 1994.

[71] P. W. Fung, G. Grebbin and Y. Attikiouzel. Model-based Region Growing Segmentation of Textured Images. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4:2313-2316, 1990.

[72] Y. Xiaohan, J. Yla-Jailski, O. Huttunen, T. Vehkomiiki, O. Sipila and T. Katila. Image Segmentation Combining Region Growing and Edge Detection, *International Conference on Pattern Recognition,* 3:481-484, 1992.

[73] M. Wasilewski. Active Contours without Edges Matlab Codes, 2005. http://www.postulate.org/segmentation.php.

[74] C. Xu and J. L. Prince. Gradient Vector Flow Snake Matlab Codes, 1999. http://iacl.ece.jhu.edu/projects/gvf/.

[75] H. Freeman and L.S Davis, A Corner Finding Algorithm for Chain Coded Curves, *IEEE Transactions on Computers*, 26:297-303, 1977.

[76] G. C. H. Chang and C. C. J. Kuo. Wavelet Descriptor of Planer Curves: Theory and Applications, *IEEE Transactions on Image Processing*, 5:56-70, 1996.

[77] I. Kunttu, L. Lepisto, J. Rauhamaa and A. Visa. Multiscale Fourier Descriptor for Shape Classification, *IEEE International Conference on Image Analysis and Processing*, papes:536-541, 2003.

[78] I. Kunttu, L. Lepisto, J. Rauhamaa and A. Visa. Multiscale Fourier Descriptor for Shape-Based Image Retrieval, *IEEE International Conference on Pattern Recognition*, 2:765-768, 2004.

[79] F. Mokhtarian and A.K. Mackworth. A Theory of Multiscale, Curvature-based Shape Representation for Planer Curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789-805, 1992.

[80] D. S. Zhang and G. Lu. A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures. *International Conference on Intelligent Multimedia and Distance Education*, pages:1-9, 2001.

[81] D. S. Zhang and G. Lu. A Comparative Study of Fourier Descriptors for Shape Representation and Retrieval, *Asian Conference on Computer Vision*, pages:646-651, 2002.

[82] G. McNeil and S. Vijayakumar. 2D Shape Classification and Retrieval, *International Joint Conference on Artificial Intelligence,* pages:1483-1488, 2005.

[83] G. McNeil and S. Vijayakumar. Hierarchical Procrustes Matching for Shape Retrieval, *IEEE International Conference on Computer Vision and Pattern Recognition,* 1:885-894, 2006.

[84] T. Adamek and N. E. O'Connor. A Multiscale Representation Method for Nonrigid Shapes with a Single Closed Contour, *IEEE Transactions on Circuit and Systems for Video Technology*, 14(5):742-753, 2004.

[85] R. S. Torres, A.X. Falcao and L.F. Costa. A Graph-based Approach for Multiscale Shape Analysis, *Pattern Recognition*, 37(6):1163-1174, 2004.

[86] I. E. Rube, N. Alajlan, M. Kamel, M. Ahmed and G. Freeman. Robust Multiscale Triangle-Area Representation for 2D Shapes*, IEEE International Conference on Image Processing*, 1:545-548, 2005.

[87] R.J. Prokop and A.P. Reeves. A Survey of Moment-based Techniques for Unoccluded Object Representation and Recognition, *CVGIP: Graphical Models and Image Processing*, 54(5):438–460, 1992.

[88] D. S. Zhang and G. Lu. Generic Fourier Descriptor for Shape-based Image Retrieval, *IEEE International Conference on Multimedia and Expo*, 1:425-428, 2002.

[89] M. K. Hu. Visual Pattern Recognition by Moment Invariants, *IRE Transactions on Information Theory*, IT-8:179–187, 1962.

[90] M.R. Teague. Image Analysis via the General Theory of Moments, *Journal of the Optical Society of America*, 70(8):920–930, 1979.

[91] C-H. Teh and R.T. Chin. On Image Analysis by the Method of Moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.

[92] A. V. Oppenheim and J. S. Lim. The Importance of Phase in Signals, *Proceedings of the IEEE*, 69(5):529-541, 1981.

[93] J. Wood. Invariant Pattern Recognition: A Review, *Pattern Recognition*, 29(1):1-17. 1996.

[94] G. Wolberg, S. Zokai. Robust Image Registration using Log-polar Transform, *IEEE International Conference on Image Processing*, 1:493-496, 2000.

[95] M. Nixon and A. Aquado. Elliptic Fourier Descriptors Matlab Code, Feature Extraction and Image Processing (Book), 2nd Edition, pages:307-308, 2008.

[96] W. Y. Kim and Y. S. Kim. A Region-based Shape Descriptor using Zernike moments, *Signal Processing: Image Communication*, 16(1):95-102, 2000.

[97] K. Chang. Zernike Moments Matlab Code, LANS Pattern Recognition Toolbox, 2005. http://www.mathworks.com/matlabcentral/fileexchange/7972

[98] M. Braun. Differential Equations and Their Applications, 4th Edition, Springer-Verlag, 1993.

[99] G. E. Myers. Analytical Methods in Conduction Heat Transfer, Genium Publishing Corporation, 1987.

Appendix A

Analytical Solution of Proposed Heat Conduction Problem and Comparison with the Numerical Solution in the Image Domain

We present a direct comparison of the analytical solution of the proposed heat conduction problem with the solution derived in the image domain for segmentation. The analytical result is derived by using Green's function. The image result is obtained by approximating finite difference operations in the image domain.

A.1 Analytical Solution of the Proposed Heat Conduction Problem on a Square Object

Consider the following boundary value problem for the temperature in a two-dimensional conductive finite solid, which has uniform conductivity and isotropic medium. The solid is also shown in Figure A.1 with the coordinates of vertices superimposed.

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{1}{k}q(x,y,t)=\frac{1}{\alpha}\frac{\partial T}{\partial t};\quad 0<x<c;\quad 0<y<d;\quad t>0 \qquad \text{(A.1a)}$$

$$T(x,y,t=0)=F(x,y),\quad \textit{Initial Condition} \qquad \text{(A.1b)}$$

$$\left.\begin{aligned} T(x=0,y,t)&=T_1 \qquad \text{(A.1c)}\\ T(x=c,y,t)&=T_2 \qquad \text{(A.1d)}\\ T(x,y=0,t)&=T_3 \qquad \text{(A.1e)}\\ T(x,y=d,t)&=T_4 \qquad \text{(A.1f)}\end{aligned}\right\}\ \textit{, Boundary Conditions}$$

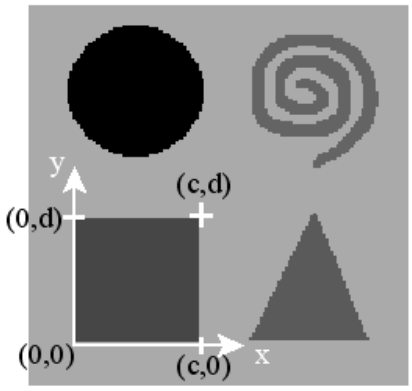


Figure A.1: Coordinates of the two-dimensional conductive finite solid.

Here we seek the temperature, $T(x,y,t)$, in a two-dimensional solid at position, (x,y), at time, t, caused by a specified energy generation function, $q(x,y,t)$, inside the body. Initially the temperature is a known function $F(x,y)$. The thermal properties are conductivity, k and diffusivity, α. In the proposed model, the boundary condition is Dirichlet, which means the temperature is specified at each boundary point. The temperature that satisfies the above equations will be found in two steps. First, Green's function will be defined, and then this function will be used to construct the temperature. Green's function is a basic solution to a linear differential equation, a building block that can be used to construct many useful solutions. For heat conduction, Green's function is proportional to the temperature caused by a concentrated energy source. The exact form of the Green's function depends on the differential equation, the body shape, and the type of boundary conditions present. Green's functions are named in honor of English mathematician and physicist George Green (1793-1841). The Green's function, $G(x,y,t\,|\,x',y',\tau)$, associated with the above example obeys the following equations [20]:

$$\alpha\left(\frac{\partial^2 G}{\partial x^2}+\frac{\partial^2 G}{\partial y^2}\right)+\delta(x-x')(y-y')(t-\tau)=\frac{\partial G}{\partial t};\ \ 0<x<c;\ \ 0<y<d;\ \ t>\tau \quad \text{(A.2a)}$$

$$G=0 \ \text{ for } \ t<\tau\ , \qquad \textit{Initial Condition} \quad \text{(A.2b)}$$

$$\left.\begin{aligned} G(x=0,y,t\,|\,x',y',\tau)&=0 \quad \text{(A.2c)}\\ G(x=c,y,t\,|\,x',y',\tau)&=0 \quad \text{(A.2d)}\\ G(x,y=0,t\,|\,x',y',\tau)&=0 \quad \text{(A.2e)}\\ G(x,y=d,t\,|\,x',y',\tau)&=0 \quad \text{(A.2f)} \end{aligned}\right\}\ \textit{, Boundary Conditions}$$

Note that the boundary conditions are Dirichlet that is the same type as the temperature problem in Equation A.1, but *homogeneous,* and that the energy generation term has been replaced by a product of three Dirac delta functions, $\delta(x-x')(y-y')(t-\tau)$, two for space and one for time. The Green's function, $G(x,y,t\,|\,x',y',\tau)$, represents the temperature response observed at point (x,y) and time t caused by an instantaneous concentrated heat source released at point (x',y') and time τ. Green's functions are *causal,* since there is no response before the heat source is released: $G=0$ for $t<\tau$. Finally, the thermal properties are constant, so the differential equation is linear; this is important since Green's functions may only be found for linear differential equations.

The temperature solution is constructed from a suitable distribution of the Green's function. The temperature for the problem in Equation A.1 and in Figure A.1 is given by [20]:

$$\begin{aligned} T(x,y,t)=&\int_{x'=0}^{c}\int_{y'=0}^{d} G(x,y,t\,|\,x',y',\tau=0)F(x',y')dx'dy'\\ &+\frac{\alpha}{k}\int_{\tau=0}^{t}\int_{x'=0}^{c}\int_{y'=0}^{d} G(x,y,t\,|\,x',y',\tau)q(x',y',\tau)d\tau dx'dy'\\ &+\alpha\int_{\tau=0}^{t} T_1\frac{\partial G}{\partial x'}\bigg|_{x'=0} d\tau-\alpha\int_{\tau=0}^{t} T_2\frac{\partial G}{\partial x'}\bigg|_{x'=c} d\tau\\ &+\alpha\int_{\tau=0}^{t} T_3\frac{\partial G}{\partial y'}\bigg|_{y'=0} d\tau-\alpha\int_{\tau=0}^{t} T_4\frac{\partial G}{\partial y'}\bigg|_{y'=d} d\tau \end{aligned} \quad \text{(A.3)}$$

In this expression, the first additive term is constructed from the initial condition, the second is from the energy generation (heat source) and the others for each boundary condition. If we consider the proposed heat conduction problem for object

segmentation, which has initial condition $T(x,y,t=0)=F(x,y)=0$ and the boundary conditions $T_1=T_2=T_3=T_4=0$, Equation A.3 can be simplified to,

$$T(x,y,t)=\frac{\alpha}{k}\int_{\tau=0}^{t}\int_{x'=0}^{c}\int_{y'=0}^{d}G(x,y,t\mid x',y',\tau)q(x',y',\tau)d\tau dx'dy' \tag{A.4}$$

which only includes the effect of the heat source. The next problem is determination of the Green's function due to the proposed problem. To determine Green's function, we need to consider only the homogeneous version of the problem in Equation A.1, which does not include the source function and has boundary temperatures $T_1=T_2=T_3=T_4=0$ as given below,

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}=\frac{1}{\alpha}\frac{\partial T}{\partial t};\quad 0<x<c;\quad 0<y<d;\quad t>0 \tag{A.5a}$$

$$T(x,y,t=0)=F(x,y) \tag{A.5b}$$

$$T(x=0,y,t)=0;\ \ T(x=c,y,t)=0 \tag{A.5c}$$

$$T(x,y=0,t)=0;\ T(x,y=d,t)=0 \tag{A.5d}$$

This homogeneous problem is solved by the method of separation of variables [20, 98, 99] and the solution is expressed formally in the form

$$T(x,y,t)=\int_{x'=0}^{c}\int_{y'=0}^{d}\psi(x,y,x',y',t)F(x',y')dx'dy' \tag{A.6}$$

For the problem in Equation A.5 [20] [98],

$$\psi(x,y,x',y',t)=\frac{4}{cd}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}e^{-\alpha\left(\beta_m^2+\lambda_n^2\right)t}\sin(\beta_m x)\sin(\lambda_n y)\sin(\beta_m x')\sin(\lambda_n y') \tag{A.7}$$

where

$$\beta_m=\frac{m\pi}{c},\quad \lambda_n=\frac{n\pi}{d}\quad \text{with}\quad (m,n)=1,2,3,4...$$

Then Equation A.6 is compared with the first additive term of the Equation A.3 and $G(x,y,t\,|\,x',y',\tau=0)$ is determined as

$$G(x,y,t\,|\,x',y',\tau=0)=\psi(x,y,x',y',t) \tag{A.8}$$

However, for the proposed problem in Equation A.4, we need to know $G(x,y,t\,|\,x',y',\tau)$. The desired Green's function, $G(x,y,t\,|\,x',y',\tau)$, is determined from $G(x,y,t\,|\,x',y',\tau=0)$ by replacing t by $(t-\tau)$ [20], as given below

$$G(x,y,t\,|\,x',y',\tau)=\frac{4}{cd}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}e^{-\alpha\left(\beta_m^2+\lambda_n^2\right)(t-\tau)}\sin(\beta_m x)\sin(\lambda_n y)\sin(\beta_m x')\sin(\lambda_n y') \tag{A.9}$$

Where

$$\beta_m=\frac{m\pi}{c},\quad \lambda_n=\frac{n\pi}{d}\quad \text{with}\quad (m,n)=1,2,3,4...$$

Then the analytical solution of the proposed problem is obtained in terms of the Green's function according to the Equation A.4, as

$$T(x,y,t)=\frac{\alpha}{k}\int_{\tau=0}^{t}\int_{x'=0}^{c}\int_{y'=0}^{d}\left[\frac{4}{cd}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}e^{-\alpha\left(\beta_m^2+\lambda_n^2\right)(t-\tau)}\sin(\beta_m x)\sin(\lambda_n y)\sin(\beta_m x')\sin(\lambda_n y')\right]$$
$$\cdot\, q(x',y',\tau)\,d\tau dx'dy' \tag{A.10}$$

after re-ordering, we obtain

$$T(x,y,t)=\frac{\alpha}{k}\frac{4}{cd}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}e^{-\alpha\left(\beta_m^2+\lambda_n^2\right)t}\sin(\beta_m x)\sin(\lambda_n y)$$
$$\cdot\int_{\tau=0}^{t}\int_{x'=0}^{c}\int_{y'=0}^{d}e^{\alpha\left(\beta_m^2+\lambda_n^2\right)\tau}\sin(\beta_m x')\sin(\lambda_n y')q(x',y',\tau)\,d\tau dx'dy' \tag{A.11}$$

In our case, the heat source is located at a point (a,b) and independent of time, which can be represented as $q(x,y,t)=q_c\delta(x-a)\delta(y-b)$, where q_c is a constant. After substituting this heat source form into the Equation A.11, we find

$$T(x,y,t)=\frac{4\alpha q_c}{kcd}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}e^{-\alpha\left(\beta_m^2+\lambda_n^2\right)t}\sin(\beta_m x)\sin(\beta_m a)\sin(\lambda_n y)\sin(\lambda_n b)\int_{\tau=0}^{t}e^{\alpha\left(\beta_m^2+\lambda_n^2\right)\tau}d\tau \quad \text{(A.12)}$$

Then we simplify Equation A.12 and obtain the following equation

$$T(x,y,t)=\frac{4q_c}{kcd}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\frac{\sin(\beta_m x)\sin(\beta_m a)\sin(\lambda_n y)\sin(\lambda_n b)\left(1-e^{-\alpha\left(\beta_m^2+\lambda_n^2\right)t}\right)}{\beta_m^2+\lambda_n^2}$$

Where

(A.13)

$$\beta_m=\frac{m\pi}{c},\quad \lambda_n=\frac{n\pi}{d}\quad \text{with}\quad (m,n)=1,2,3,4...$$

For the numerical solution in the image domain, we use the square object of size $(c,d)=(50,50)$ and the source position is $(a,b)=(35,16)$, as shown in Figure A.2. In addition, we equate the source term to a constant as $Q=\frac{q_c}{\rho c}=5$ and the thermal diffusivity $\alpha=\frac{k}{\rho c}=0.25$. As a result of these assumptions for calculation in the image domain, $\frac{q_c}{k}=\frac{5}{0.25}=20$. After locating these constants into Equation A.13, we can calculate the temperature value at any point (x,y) and time t by means of the analytic solution.

$$T(x,y,t)=0.032\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\frac{\sin(\beta_m x)\sin(35\beta_m)\sin(\lambda_n y)\sin(16\lambda_n)\left(1-e^{-0.25\left(\beta_m^2+\lambda_n^2\right)t}\right)}{\beta_m^2+\lambda_n^2}$$

Where

(A.14)

$$\beta_m=\frac{m\pi}{50},\quad \lambda_n=\frac{n\pi}{50}\quad \text{with}\quad (m,n)=1,2,3,4...$$

The solution is achieved by iterating the series summation above until we reach the stable result.

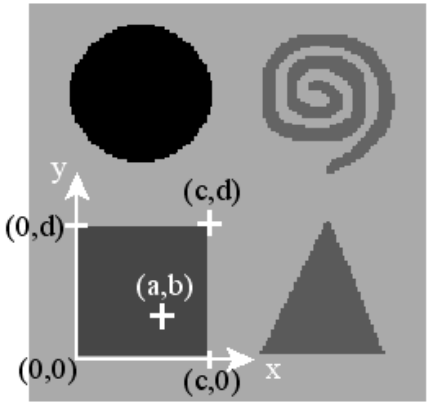


Figure A.2: Source Position

Before the comparison of analytical and numerical results in the image domain, we explain the finite difference representation of the heat conduction equation in the image domain.

A.2 Numerical Solution of the Proposed Heat Conduction Problem in the Image Domain by Using Finite Difference Representation

Consider the heat conduction equation in the image domain as

$$\frac{\partial \mathbf{I}(x,y,t)}{\partial t} = \mathbf{CF}(x,y,t)\left(\alpha\left(\frac{\partial^2 \mathbf{I}(x,y,t)}{\partial x^2} + \frac{\partial^2 \mathbf{I}(x,y,t)}{\partial y^2}\right)\right) + \mathbf{Q}(x,y,t) \qquad \text{(A.15)}$$

where $\mathbf{I}(x,y,t)$ represents an image pixel value in terms of temperature at each point and time, α is the thermal diffusivity, $\mathbf{Q}(x,y,t)$ is the source term and $\mathbf{CF}(x,y,t)$ is the control function. The control function is obtained from the region statistics of the source location on a given grey-level image. The region of interest in the space domain is bounded as, $0 \le x \le c$, $0 \le y \le d$, and in the time domain extends from $t = 0$ to infinity. We construct a finite difference net from the pixels of object in the x and y domain with constant mesh sizes Δx and Δy. In the time domain, the step size is Δt. Then, the space and time coordinates are denoted by

$$x = i\Delta x \qquad i = 0,1,2,...,L \quad \text{with} \quad c = L\Delta x \tag{A.16a}$$

$$y = j\Delta y \qquad j = 0,1,2,...,K \quad \text{with} \quad d = K\Delta y \tag{A.16b}$$

$$t = n\Delta t \qquad n = 0,1,2,3... \tag{A.16c}$$

and the temperature $\mathbf{I}(x, y, t)$ is represented by

$$\mathbf{I}(x, y, t) = \mathbf{I}(i\Delta x, j\Delta y, n\Delta t) \equiv \mathbf{I}_{i,j}^{n} \tag{A.17}$$

By using this notation, the finite difference representation of $\partial^2\mathbf{I}/\partial x^2$ at the pixel (i, j) $(x = i\Delta x, y = j\Delta y)$ for the n^{th} time step $(t = n\Delta t)$ is written with the *central-difference formula* as [20]

$$\left.\frac{\partial^2\mathbf{I}}{\partial x^2}\right|_{i,j,n} \cong \frac{\mathbf{I}_{i-1,j}^{n} - 2\mathbf{I}_{i,j}^{n} + \mathbf{I}_{i+1,j}^{n}}{(\Delta x)^2} \tag{A.18}$$

and for $\partial^2\mathbf{I}/\partial y^2$,

$$\left.\frac{\partial^2\mathbf{I}}{\partial y^2}\right|_{i,j,n} \cong \frac{\mathbf{I}_{i,j-1}^{n} - 2\mathbf{I}_{i,j}^{n} + \mathbf{I}_{i,j+1}^{n}}{(\Delta y)^2} \tag{A.19}$$

The first derivative with respect to the time variable, $\partial\mathbf{I}/\partial t$, is represented in the finite difference form at the n^{th} time step using the *forward-difference formula* as

$$\left.\frac{\partial\mathbf{I}}{\partial t}\right|_{i,j,n} \cong \frac{\mathbf{I}_{i,j}^{n+1} - \mathbf{I}_{i,j}^{n}}{\Delta t} \tag{A.20}$$

Introducing Equations A.18, A.19 and A.20 into the Equation A.15, the finite difference representation of the heat conduction equation in the image domain becomes

$$\frac{\mathbf{I}_{i,j}^{n+1} - \mathbf{I}_{i,j}^{n}}{\Delta t} = \mathbf{CF}(i, j, n)\left(\alpha\left(\frac{\mathbf{I}_{i-1,j}^{n} - 2\mathbf{I}_{i,j}^{n} + \mathbf{I}_{i+1,j}^{n}}{(\Delta x)^2} + \frac{\mathbf{I}_{i,j-1}^{n} - 2\mathbf{I}_{i,j}^{n} + \mathbf{I}_{i,j+1}^{n}}{(\Delta y)^2}\right)\right) + \mathbf{Q}(i, j, n) \tag{A.21}$$

In the image domain, we use equal spacing $\Delta x = \Delta y$ in both $x-$ and $y-$ directions and we obtain the solution of $\mathbf{I}_{i,j}^{n+1}$ as

$$\mathbf{I}_{i,j}^{n+1} = (1 - 4r\mathbf{CF}(i,j,n))\mathbf{I}_{i,j}^{n} + \mathbf{CF}(i,j,n)\left(r\left(\mathbf{I}_{i-1,j}^{n} + \mathbf{I}_{i+1,j}^{n} + \mathbf{I}_{i,j-1}^{n} + \mathbf{I}_{i,j+1}^{n}\right)\right) + \Delta t\mathbf{Q}(i,j,n) \quad \text{(A.22)}$$

where $r = \dfrac{\alpha\Delta t}{(\Delta x)^2}$ and $0 \le \dfrac{\alpha\Delta t}{(\Delta x)^2} \le 0.25$ to achieve stability criterion for the two dimensional case [23, 20]. The finite difference representation given by the Equation A.22 is called the explicit form. Because the unknown temperature value $\mathbf{I}_{i,j}^{n+1}$ at the time step $(n+1)$ can be directly determined from the knowledge of temperatures $\mathbf{I}_{i-1,j}^{n}$, $\mathbf{I}_{i+1,j}^{n}$, $\mathbf{I}_{i,j-1}^{n}$, $\mathbf{I}_{i,j+1}^{n}$ and $\mathbf{I}_{i,j}^{n}$ at the previous time step (n). In our proposed model, in the image domain $\Delta x = \Delta y = \Delta t = 1$, which yields $i = x$, $j = y$ and $n = t$ from Equation A.16. In addition, $\alpha = 0.25$ and

$$\mathbf{Q}(x,y,t) = \begin{cases} 5, & (x,y) = (e,f) \\ 0 & otherwise \end{cases} \quad \text{(A.23)}$$

where, $(e,f) = (a + offset1, b + offset2)$ is the source position in the image domain. In analytic domain, the origin, $(0,0)$, was at the corner of the square object and the source position was at the (a,b) as shown in Figure A.2. However, in the image domain the origin is at the image corner, so that $(offset1, offset2)$ is the difference between the origin in analytic domain and the origin in the image domain. After locating the numerical values into the Equation A.22, we obtain a numerical solution in the image domain for the proposed problem as given below,

$$\mathbf{I}_{x,y}^{t+1} = \mathbf{I}_{x,y}^{t} + \mathbf{CF}(x,y,t)\left(0.25\left(\mathbf{I}_{x-1,y}^{t} + \mathbf{I}_{x+1,y}^{t} + \mathbf{I}_{x,y-1}^{t} + \mathbf{I}_{x,y+1}^{t} - 4\mathbf{I}_{x,y}^{t}\right)\right) + \mathbf{Q}(x,y,t) \quad \text{(A.24)}$$

Errors involved in numerical solutions

The two-dimensional heat conduction equation is a partial differential equation, which includes first- and second-order continuous derivatives. In numerical solution, these

derivatives are approximated by finite differences using a Taylor series expansion. The error, in the approximation, is called the truncation error or the discretization error.

A.3 Comparison of Analytical and Numerical Results in the Image Domain

To compare analytical and numerical solutions, we choose one vertical and one horizontal line on the square object; these lines intersect at the centre of the square object, $(25,25)$, as well as they start from one boundary and end at the other boundary of the square object as shown in Figure A.3.

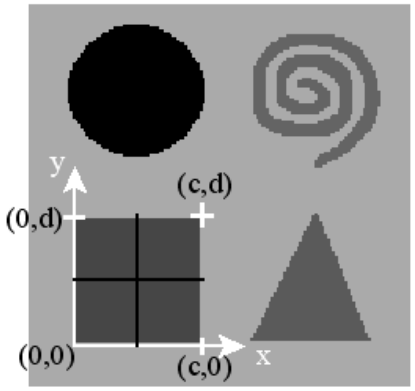


Figure A.3: Selected horizontal and vertical line positions to compare analytical temperature values with the numerically obtained in the image domain.

Figure A.4 shows analytical $(\mathbf{A})$ and image domain $(\mathbf{I})$ temperature values as well as absolute temperature error $|\mathbf{A}-\mathbf{I}|$ between them at the positions of selected *vertical* line. These results are calculated at $t=69$, when all the temperature values inside the square object starts to exceed zero in image domain calculations, in other words when the segmentation of square object is achieved. It is observed that all the temperature values are higher than zero except the boundary points. The error occurs, between the analytical and numerical results, are because of the discretization process in the numerical solution. The important results to observe are at the boundary positions, which are $(25,0)$ and $(25,50)$ in this line. In the position, $(25,0)$, we observe that both analytical and numerical results are zero. In the other boundary position, $(25,50)$, we observe that numerical result is zero, and the analytical result is zero within numerical

error. These results confirm that boundary temperature is kept at zero in the proposed problem.

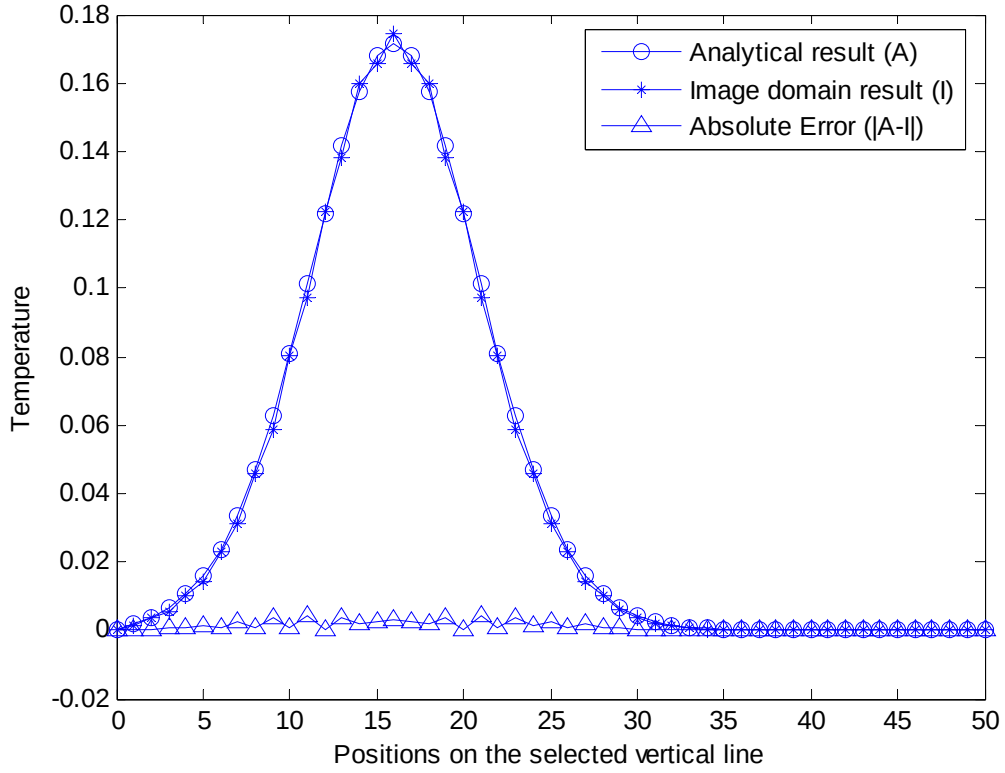


Figure A.4: Analytical and numerical (in the image domain) temperature values at the positions of the selected *vertical* line (from the one boundary to the other) and at t=69, when all of the temperature values inside the square object starts to be higher than zero, in other words when the segmentation of square object is achieved.

Figure A.5 shows analytical and numerical (in the image domain) temperature values at the positions of selected *horizontal* line at $t = 69$. It is observed that all of the temperature values inside the square object exceed zero except at the boundary positions. The error occurs, between the analytical and numerical results, because of the discretization process. The boundary positions, on the *horizontal* line, are $(0,25)$ and $(50,25)$. In the position, $(0,25)$, we observe that both analytical and numerical results are zero. In the other one, $(50,25)$, we observe that numerical result is zero, and the analytical result is zero within numerical error. These results also confirm that boundary temperature is kept at zero during the segmentation process.

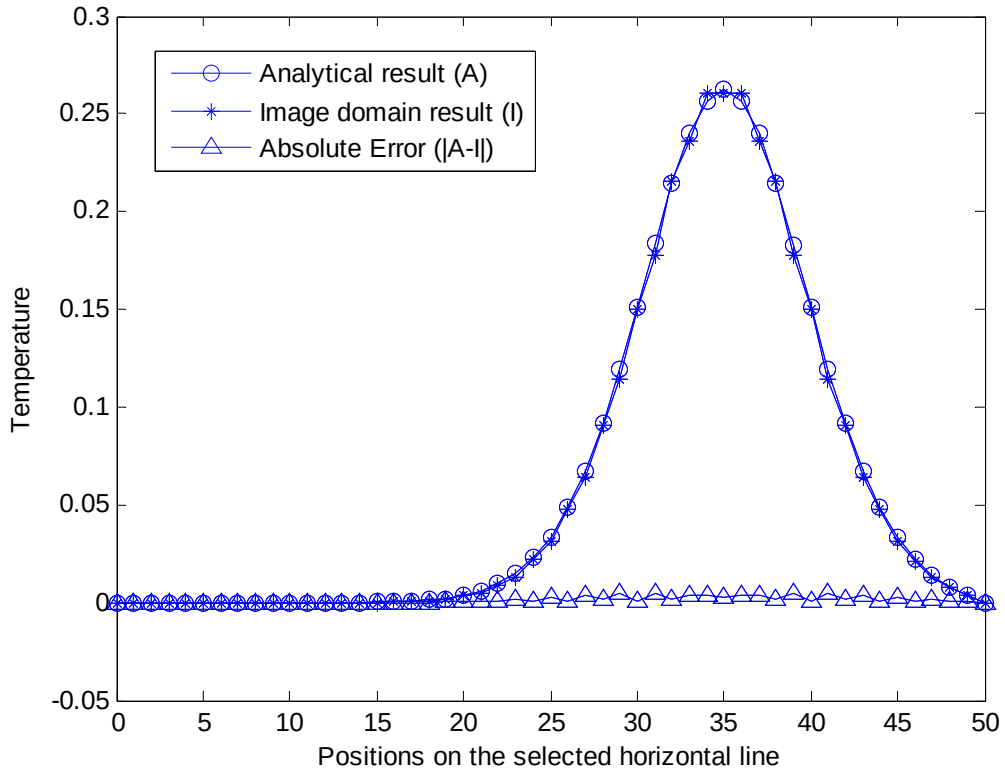


Figure A.5: Analytical and numerical (in the image domain) temperature values at the positions of selected *horizontal* line (from the one boundary to the other) and at t=69, when the all of the temperature values inside the square object starts to exceed zero, in other words when the segmentation of square object is achieved.

The temperature values at Figure A.4 and A.5 are observed at $t = 69$, when the segmentation of the square object is achieved in the image. At this time, our algorithm automatically stops, however if we continue diffusion until the steady-state solution, we can observe that temperature values increase, except at the boundary positions. Boundary temperature is always kept at zero. Figure A.6 shows the steady-state analytical and numerical temperature values at the positions of selected *vertical* line. For the numerical calculation in the image domain, we observed that steady-state is achieved after $t = 16442$. At this time, one of the boundary position, $(25,0)$, has analytical and numerical results zero. The other boundary position, $(25,50)$, has numerical result zero and the analytical result zero within numerical error. These results show that boundary temperature is always kept at zero until the steady-state and after the steady-state.

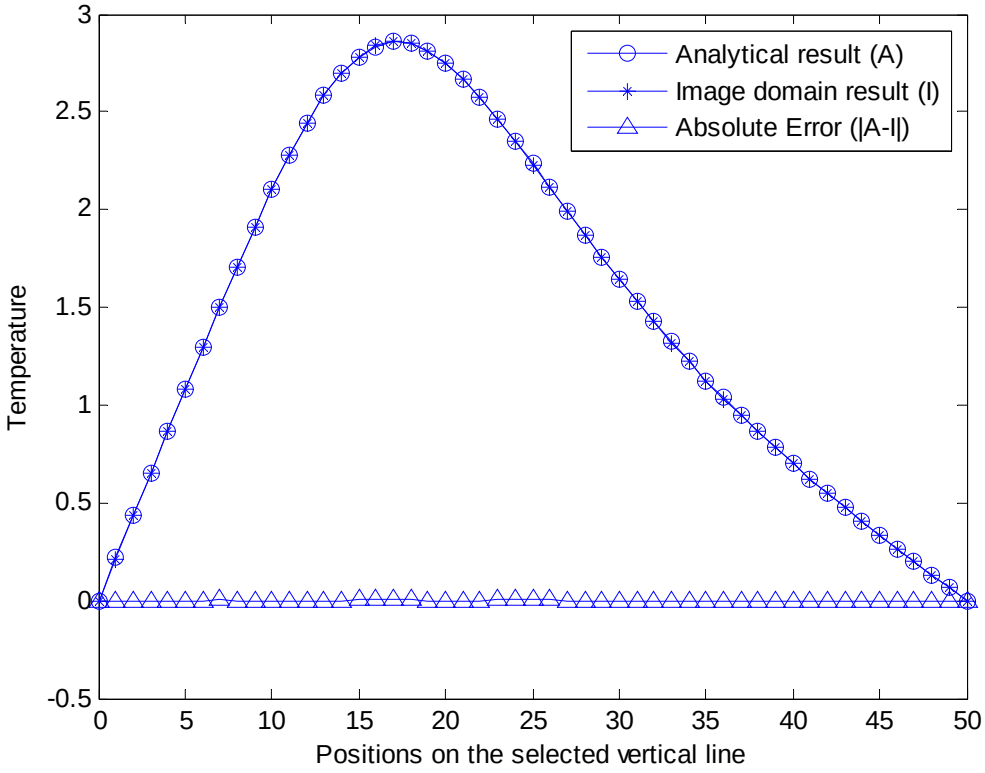


Figure A.6: Steady state analytical and numerical (in the image domain) temperature values at the positions of selected *vertical* line (from the one boundary to the other).

Figure A.7 similarly shows the steady-state analytical and numerical temperature values at the positions of selected horizontal line. In this line, one of the boundary point, $(0,25)$, has analytical and numerical temperature values of zero. The other one, $(50,25)$, has numerical result zero and the analytical result zero within numerical error. These results again confirm the proposed heat conduction problem in the image domain, which achieves segmentation.

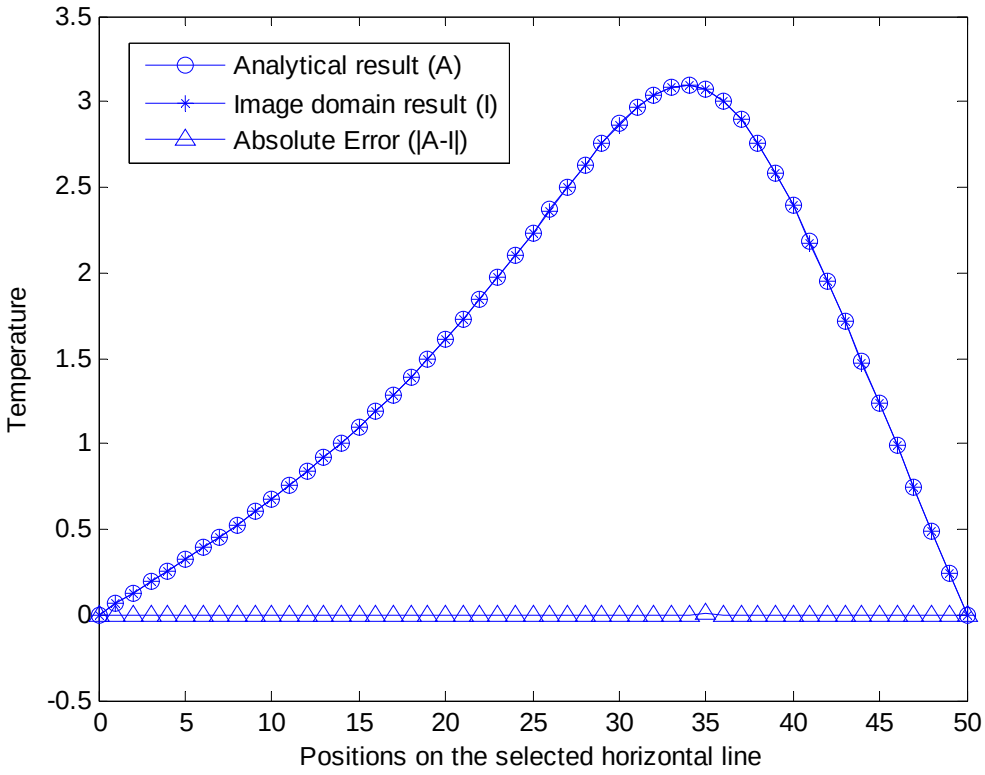


Figure A.7: Steady state analytical and numerical (in the image domain) temperature values at the positions of selected *horizontal* line (from the one boundary to the other).

Appendix B

Active Contours

Active contours (snakes) are classified either Parametric Active Contours (PAC) or Geometric Active Contours (GAC), according to their representation. PAC and GAC differ in their ability to handle multiple object detection and computational complexity and efficiency. Here, we explain the commonly used active contour models including Gradient Vector Flow (GVF) based PAC [41] and region-based GAC (active contours without edges) [64], which are compared with our shape extraction model using heat flow analogy in Chapter 4.

B.1 Parametric Active Contour (PAC)

The PAC is a parametric curve $C(s)=(\mathbf{x}(s),\mathbf{y}(s))$, $s\in[0,1]$, which is represented explicitly and moves through the spatial domain of an image. The first PAC model was introduced by Kass et al. [40]. It was formulated by minimizing the following energy functional,

$$\xi=\int_0^1 \alpha_G|C'(s)|^2+\beta_G|C''(s)|^2+\gamma_G\xi_{ext}(C(s))ds \tag{B.1}$$

Where, $C'(s)$ and $C''(s)$ are the first and second derivatives of the curve $C(s)$ and they impose smoothness and rigidity constraints, $\xi_{ext}(C(s))$ is external energy function, which is obtained from the image and takes small values on the object boundaries, α_G, β_G and γ_G are weighting parameters that control the smoothness, rigidity and external

energy of the curve respectively. A typical external energy function for a grey level image, $\mathbf{I}(x,y)$, is given below,

$$\xi_{ext} = -\left|\nabla\left[G_\sigma(x,y) * \mathbf{I}(x,y)\right]\right| \tag{B.2}$$

where $G_\sigma(x,y)$ is Gaussian filter with standard deviation σ and ∇ is the gradient operator. The solution to the Equation B.1 is obtained by first finding the Euler equation that minimizes the energy functional, ξ, and then making it dynamic with a partial derivative of $C(s,t)$ with respect to time t:

$$C_t(s,t) = \alpha_G C''(s,t) + \beta_G C''''(s,t) - \gamma_G \nabla \xi_{ext} \tag{B.3}$$

Where, $\alpha_G C''(s,t) + \beta_G C''''(s,t)$ is an internal force and $-\gamma_G \nabla \xi_{ext}$ is external force acting on the snake. The internal force discourages stretching and bending, while the external force is image dependent and pulls the snake towards the desired object edges. When the contour attains steady state, $C_t(s,t) = 0$, it achieves segmentation.

Problems, associated with this classic model, are initialization and poor convergence to concave regions. A constant force, which is called the balloon force [57], was added with direction normal to curve, to accelerate the motion so that the initial curve can be placed far away from the desired object boundary. However, choice of the balloon force is a problem. If the balloon force is large, the contour can pass through weak object boundaries. If the balloon force is small, the contour may not move on narrow regions of the object. Initialization and convergence to concave region problems were largely solved with the development of new external force model, which is called Gradient Vector Flow (GVF) [41]. GVF consists of a two-dimensional vector field $\mathbf{V}(x,y) = [\mathbf{u}(x,y), \mathbf{v}(x,y)]$ that minimizes the following energy functional,

$$\xi(\mathbf{u},\mathbf{v}) = \int\int \mu_G\left(\mathbf{u}_x^2 + \mathbf{u}_y^2 + \mathbf{v}_x^2 + \mathbf{v}_y^2\right) + \left|\nabla \mathbf{F}\right|^2 \left|\mathbf{V} - \nabla \mathbf{F}\right|^2 dxdy \tag{B.4}$$

Where, $\nabla\mathbf{F}$ is the gradient of the edge map $\mathbf{F}(x,y)$ that is derived from the image $\mathbf{I}(x,y)$, $(\mathbf{u}_x, \mathbf{u}_y)$ and $(\mathbf{v}_x, \mathbf{v}_y)$ represent $(\partial\mathbf{u}/\partial x, \partial\mathbf{u}/\partial y)$ and $(\partial\mathbf{v}/\partial x, \partial\mathbf{v}/\partial y)$ respectively

and μ_G is a regularization parameter between first and second part in the integrand. It can be seen that when $|\nabla \mathbf{F}|$ is small, energy is minimized by sum of the partial derivatives of the vector field, which applies diffusion to the gradient vectors of the edge map $\nabla \mathbf{F}$. On the other hand, when $|\nabla \mathbf{F}|$ is large, energy is minimized by the second part, which keeps $\mathbf{V}$ nearly equal to the gradient vectors of edge map $\mathbf{V} = \nabla \mathbf{F} = (\mathbf{F}_x, \mathbf{F}_y) = (\partial \mathbf{F}/\partial x, \partial \mathbf{F}/\partial y)$. The solution to the Equation B.4 is obtained with the following equations,

$$\frac{\partial \mathbf{u}}{\partial t} = \mu_G \Delta \mathbf{u} - (\mathbf{u} - \mathbf{F}_x)|\nabla \mathbf{F}|^2 \ , \quad \frac{\partial \mathbf{v}}{\partial t} = \mu_G \Delta \mathbf{v} - (\mathbf{v} - \mathbf{F}_y)|\nabla \mathbf{F}|^2 \tag{B.5}$$

Where Δ is the Laplacian operator. From these equations, it is also observed that when the image region, $\mathbf{I}(x, y)$, is homogeneous (constant), the second term in each equation is zero because the gradient of $\mathbf{F}(x, y)$ is zero. Therefore in those regions, **u** and **v** are determined by Laplace's equation, and the resulting GVF field is diffused from the object boundary vectors. In this way, GVF yields vectors that point into boundary concavities.

To obtain the corresponding dynamic snake equation with a GVF based external force, we replace $-\nabla \xi_{ext}$ in Equation B.3 with $\mathbf{V}$, yielding

$$C_t(s,t) = \alpha_G C''(s,t) + \beta_G C''''(s,t) + \gamma_G \mathbf{V} \tag{B.6}$$

The parametric curve, which solves the above dynamic equation, is called the GVF snake (GVFS). Although, initialization and convergence to concave region problems were largely solved with the development of GVF, PAC models can have difficulty with simultaneous detection of multiple objects. This is because they cannot handle the topological changes such as merging or splitting of the moving curve, since the curve is represented explicitly. To solve this problem, GAC models have been introduced, where the curve is represented implicitly in a level set function (the zero set).

B.2 Geometric Active Contours (GAC)

GAC uses the level set method [13] for curve evolution. In level set methods, a curve is represented implicitly by the zero level set $C = \{(x, y, t) \mid \phi(x, y, t) = 0\}$ of the scalar function $\phi(x, y, t)$. The level set function is the signed distance to the curve. This distance is arranged to be positive inside the curve and negative outside it. The curve itself is where the distance is zero, at the interface between two regions. The curve evolution using level sets can be briefly explained as follows: Assume we follow the curve as it propagates in a direction normal to itself with speed U, as shown below,

$$\frac{\partial C(s,t)}{\partial t} = U\vec{n} = U\frac{\nabla\phi}{|\nabla\phi|} \tag{B.7}$$

Where the normal vector $\vec{n}$ is given by $\vec{n} = \frac{\nabla\phi}{|\nabla\phi|}$, which is pointing outward in the direction normal to the contour surface. The contour interface C is the zero level set at all time,

$$\phi(C(t), t) = 0 \tag{B.8}$$

Taking the derivative with respect to time by using the chain rules, yields

$$\frac{\partial\phi}{\partial C} \cdot \frac{\partial C}{\partial t} + \frac{\partial\phi}{\partial t} = 0 \tag{B.9}$$

Then, by rearrangement and substitution from Equation B.7, ϕ is the solution to the following differential equation,

$$\frac{\partial\phi}{\partial t} = -U\nabla\phi \cdot \frac{\nabla\phi}{|\nabla\phi|} = -U|\nabla\phi| \tag{B.10}$$

The contour evolves using the speed term U, which is particularly a function of curvature (a geometric measure). The curvature is obtained from the curve evolution itself that is independent of the curve's parameterization. The level set method can

handle topological changes of the curve such as splitting and merging because of implicit representation. However, it is computationally more expensive than the explicit representation (PAC), since it evolves a level set surface instead of contour positions, as well as re-initialization of the curve is required at each step of the evolution for rescaling and regularization, which also prevents level set function to become too flat. Curve evolution with the level set method is achieved as follows: First, the initial implicit function (distance transform) is determined depending on the initial curve position. Then the curve is evolved, one iteration, according to the level set equation given in B.10. After that, the zero level set (curve position) is recovered from the resultant evolution. Finally, the curve is re-initialized again using the distance transform for the next evolution (iteration). Numerical approximation of the level set method is also important and needs optimization, since it affects segmentation results and the computational efficiency.

The first GAC model was proposed by Caselles et al. [11] and Malladi et al. [12]. In this model, the curve evolves with its curvature and balloon force (expanding) dependent speed in the direction normal, and stops depending on a gradient information obtained from the grey-scale image $\mathbf{I}$. The evolution of the gradient-based GAC is given by the following level set equation,

$$\frac{\partial \phi}{\partial t} = g\left(\left|\nabla \mathbf{I}\right|\right)\left(\kappa + b_f\right)\left|\nabla \phi\right|, \qquad \phi(x, y, t = 0) = \phi_0(x, y) \tag{B.11}$$

where, $C = \{(x, y, 0) \,|\, \phi_0(x, y) = 0\}$ defines the initial contour, $\kappa = div\left(\frac{\nabla \phi}{|\nabla \phi|}\right)$ is the curvature term and has the effect of smoothing the curve, div means divergence, b_f is a constant that causes expanding (balloon force) and accelerates the motion, $g\left(\left|\nabla \mathbf{I}\right|\right)$ is the stopping function of curve and derived from the gradient of the image, $\left|\nabla \mathbf{I}\right|$, with the following formulation,

$$g\left(\left|\nabla \mathbf{I}\right|\right) = \frac{1}{1 + \left|\nabla G_\sigma * \mathbf{I}\right|^p}, \qquad p \geq 1 \tag{B.12}$$

The assumption is, the curve evolves on smooth regions of the image since $g \cong 1$ and it stops on the edges since $g \cong 0$. However, in practice g has a range $0 < g \leq 1$ and it can never become equal to zero. Therefore, if there is any weak edge because of noise, the curve can pass through the boundary of the object. This problem occurs mainly because of the balloon force term b_f. Isotropic Gaussian smoothing may partially remove noise, but at the same time it smooth object edges and make them weaker.

The gradient-based GAC can detect multiple objects simultaneously but it has other important problems, which are boundary leakage, noise sensitivity, computational inefficiency and complexity of implementation. Some formulations, [58, 59], have been introduced to solve problems with boundary leakage and noise sensitivity by improving the gradient-based information. However, they can just increase the tolerance, since gradient-based information is always limited by noise. Xie and Mirmehdi [60] apply region segmentation to the original image and then find the Gradient Vector field (GVF) of the segmented image. This force is added to the GAC, but it can only provide more tolerance to toward weak edges. Several numerical schemes have also been proposed to improve the computational efficiency of the level set method, including narrow band [61], fast marching [62] and additive operator splitting [63]. Despite substantial improvements in efficiency, they can be very difficult to implement.

In [64], Chen and Vese introduced new GAC model without using edge information. They proposed to make the contour robust to initialization, noise and boundary leaking. Their model is based on Mumford-Shah functional for segmentation [65]. This model assumes that the image $\mathbf{I}$ is divided into two regions of approximately piecewise-constant intensities, of distinct values I^{in} and I^{out}. The object to be detected is represented by the region with the value I^{in}, and the background is represented by the region with value I^{out}.

Let the curve C evolves in a bounded open subset Ω of $\Re^2$. $inside(C)$ denotes the region enclosed by C and $outside(C)$ denotes the region outside of C. To extract the object boundary, B, they consider the following "fitting" term,

$$F^{in}(C)+F^{out}(C)=\int_{inside(C)}|\mathbf{I}(x,y)-\mu_{in}|^2+\int_{outside(C)}|\mathbf{I}(x,y)-\mu_{out}|^2 \tag{B.13}$$

Where, the constant μ_{in} is the average brightness inside the curve C, depending on the curve position, and μ_{out} is the average outside C. The boundary of the object B is the curve C that minimizes the fitting term,

$$B=\min_{C}\left(F^{in}(C)+F^{out}(C)\right) \tag{B.14}$$

If the curve C is outside the object, then $F^{in}(C)>0$ and $F^{out}(C)\approx 0$. If the curve C is inside the object, then $F^{in}(C)\approx 0$ and $F^{out}(C)>0$. If the curve is both inside and outside the object, then $F^{in}(C)>0$ and $F^{out}(C)>0$. When the curve C is at the boundary of the object, $F^{in}(C=B)=0$ and $F^{out}(C=B)=0$, and the minimum is achieved as shown below,

$$F^{in}(B)+F^{out}(B)=0 \tag{B.15}$$

In the their active contour model, they minimize the given fitting term as well as some regularization terms, such as length of C or area of the region inside C. The proposed energy functional is defined below,

$$\begin{aligned}\xi(\mu_{in},\mu_{out},C)&=\mu_A\cdot Length(C)+v_A\cdot Area(inside(C))\\&+\lambda_{in}\int_{inside(C)}|\mathbf{I}(x,y)-\mu_{in}|^2dxdy+\lambda_{out}\int_{outside(C)}|\mathbf{I}(x,y)-\mu_{out}|^2dxdy\end{aligned} \tag{B.16}$$

Where, $\mu_A\geq 0, v_A\geq 0, \lambda_{in},\lambda_{out}>0$ are chosen parameters and in all their numerical calculations $\lambda_{in}=\lambda_{out}=1$ and $v_A=0$. They consider the following minimization problem,

$$\inf_{\mu_{in},\mu_{out},C}\xi(\mu_{in},\mu_{out},C) \tag{B.17}$$

In the level set formulation of the model, $C \subset \Omega$ is represented by the zero level set of the function $\phi : \Omega \to \Re$, such as

$$\begin{cases} C = \{(x,y) \in \Omega : \phi(x,y) = 0\} \\ inside(C) = \{(x,y) \in \Omega : \phi(x,y) > 0\} \\ outside(C) = \{(x,y) \in \Omega : \phi(x,y) < 0\} \end{cases} \tag{B.18}$$

They use the Heaviside function H, and the one-dimensional Dirac delta function δ as defined below respectively, to express the terms in the ξ.

$$H(z) = \begin{cases} 1, & if \quad z \geq 0 \\ 0, & if \quad z < 0 \end{cases}, \quad \delta(z) = \frac{\partial}{\partial z} H(z) \tag{B.19}$$

After the level set formulation, the energy terms in ξ can be written as follows

$$\begin{aligned} \xi(\mu_{in}, \mu_{out}, \phi) &= \mu_A \int_\Omega \delta(\phi(x,y)) |\nabla \phi(x,y)| dxdy + v_A \int_\Omega H(\phi(x,y)) dxdy \\ &+ \lambda_{in} \int_\Omega |\mathbf{I}(x,y) - \mu_{in}|^2 H(\phi(x,y)) dxdy + \lambda_{out} \int_\Omega |\mathbf{I}(x,y) - \mu_{out}|^2 (1 - H(\phi(x,y))) dxdy \end{aligned} \tag{B.20}$$

In order to compute the associated Euler-Lagrange equation for the unknown function ϕ, they consider slightly regularized versions of functions H and δ, denoted by H_ε and δ_ε respectively. Keeping μ_{in} and μ_{out} fixed, and minimizing ξ with respect to ϕ gives the following equation

$$\frac{\partial \phi}{\partial t} = \delta_\varepsilon(\phi) \left[\mu_A div\left(\frac{\nabla \phi}{|\nabla \phi|} \right) - v_A - \lambda_{in}(\mathbf{I} - \mu_{in})^2 + \lambda_{out}(\mathbf{I} - \mu_{out})^2 \right] = 0 \tag{B.21}$$

Their approach especially works well for bimodal images and by its formulation, does not depend on boundary data. It is good at handling initialization, noise and boundary leakage but still suffers from computational complexity and difficulty in implementation, because of the level set method. Later, Chen and Vese [66] extended this approach to the multiphase level set framework to segment more than two-phase

(binary segmentation). However, they need more level set functions for this process, which means more complexity and difficulty.

Different types of image forces can be combined with PAC or GAC (level set) models to overcome limitations and improve segmentation. In [67], the Mumford-Shah functional [65] based region force was adapted by PAC to handle noise and initialization problems. Paragios and Deriche [68] unified boundary- and region-based forces and implemented using level sets to improve segmentation and solve initialization and noise problems. Recently, Xie and Mirmehdi [69] proposed a magnetic force based on magnetostatic theory using level sets to handle problems with initialization and convergence to concave regions.