Computer Vision: Image Shape Geometry and Classification

by

Hoang Dat Pham

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg

September 2018

Abstract

This research introduces the study of shape analysis with shape descriptors based on the geometry of image triangulation. The Delaunay approach is used to superimposes on an image with a mesh filled with various sized triangles that give rise to various simplexes. In its simplest form, a simplex is a collection of path-connected vertices. The motivation for this approach in image analysis is that Delaunay triangulation covers image shapes having unknown geometries with simplexes that have known geometries that we can measure and compare. This approach provides the foundation for the study of image shape geometry and the extraction of features for many applications in computer vision such as image processing, image segmentation, object recognition and classification. Shape descriptors are constructed with features extracted from the geometry of the simplexes covering planar images. In this research, shape descriptors provide a framework for tracking the persistence tolerance of a shape over sequences of image captured my camera. An application that can be benefit from this approach is video content retrieval and classification.

Keywords: Computer vision, content-based image retrieval(CBIR), description, feature vector, shape descriptor, persistence, classification, Delaunay Triangulation, similarity measure, persistence measure, shape analysis, video content retrieval.

Acknowledgements

First of all, I would like to express my sincere gratitude to my advisor, Prof. James Peters for giving me the opportunity to be part of his research group. I'm thankful for his thorough advice, steady inspiration, insightful feedback, solid encouragement and helpful suggestion. Without him, the completion of this thesis would not have been possible. I also would like to acknowledge the financial support from the University of Manitoba during my M.sc. program here.

I wish to thank all my colleagues and friends in the Computational Intelligence Laboratory for their generous help during my research work. I would like to thank all professors in our department for their interesting and helpful courses. I also wish to acknowledge the kind help and support from the staff, especially Amy Dario, at the Department of Electrical and Computer Engineering.

Finally, I would like to thank my parent, siblings and friends for their constant love, care and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computer vision from an engineering perspective is a collection of automated tasks, performed for reconstructing and interpreting of natural scenes captured by digital cameras for the purposes of retrieving images content[12][115]. Basic information retrieved from image content can contain elemental structures (e.g., pixels, edges, angles) and geometrical structures (e.g., Voronoï regions[99], Delaunay triangulations[98], colour, shape and textures)[101].Generally, information from a 3-D real world object is mapped to digital signals which are represented in 2-D images. These signals are then processed, segmented and extracted to yield useful features for some computer vision tasks such as classification and object recognition. Of all the geometrical structures, shape is important part of the description of the image content[138] . Due to the nature of computer vision which is a many-to-one mapping process, an object's dimensional information will be lost. Therefore, shape representation and description is a challenging task[138].

Description of shape can have either spatial or descriptive representation[97], with ex-

tracted features based on information contained in either the boundary region and descriptive interior content of the object[110]. Many descriptive measures of shape have been implemented, namely, shape signature, signature histogram, shape invariants, moments, curvature, shape context, shape matrix, spectral features etc[138][11][45].Similar shapes should have close descriptive proximity on the feature space and are perceptually near each other[55]. Thus near set theory [96] can be applied to identify the nearness of shape descriptions in computing the perceptual similarity of shapes.

Delaunay triangulations[31] is a partitioning representation that will approximate continuous space into discrete triangular-shaped regions[97]. As various triangles and vertices (simplexes) are generated, a collection of simplexes is called simplicial complex or nerve[97]. For simplicial complexes or nerves that have the maximum number of simplexes, they can be referred as maximal nucleus clusters (MNCs). Every MNCs are likely to have convex structure, if they are not residing on edge of triangulated region, with a simplex in the center called the nucleus and are adjoined by neighboring simplexes. Thus they provide the basis for geometrical features extraction in building shape description.

Perceptually similar shapes are shapes that have been rotated, translated, scaled and affinely transformed shapes[138]. In order to accommodate perceptual similarity between shapes for classification task, spacial and descriptive representation of shapes must be unique, scale invariant and also be robust to image transformation. Such feature representation can be obtained with Scale Invariant Feature Transform (SIFT)[81] or Speeded Up Robust Features (SURF)[56] for faster computation time.

1.1 Motivation

1.1.1 Thesis Objective

There were three major objectives for this thesis. First, to detect the shape of object on the foreground of the digital image. Second, to analyze the geometrical structures of the shape and construct the geometrical description of the shape. And third, intuition to classify the geometrical description of the shape.

1.1.2 Novelty of the Thesis

The novelty of the thesis is that we use image triangulation to detect and approximate the object shape on digital images. Shape detection, shape's geometrical description and shape approximation are the applications of this triangulation approach.

1.2 Major Contributions of the Thesis

In this thesis, I propose a novel approach to construct geometry-based, feature-based representation of shape descriptor with MNCs as a form of spacial and descriptive representation, such that the MNCs can cover the entire image object shape, which are useful for capturing region of interest for classification task and image object retrieval. Thus, I explore through experiments for application of building and tracking persistence of image object shapes over a sequences of image frames. Additionally, the proposed method can be useful in image analysis, object identification, object recognition, video content retrieval and content-based image retrieval(CBIR).

1.3 Organization of the Thesis

The thesis is arranged as follows:

- Chapter 1 contains the introduction of thesis, major contribution and thesis organization.
- Chapter 2 introduces the relevant background and established preliminaries for experiments in this thesis.
- Chapter 3 elaborates more details for shape descriptors and the proposed method for geometry-based image shape descriptor. This chapter also introduces application of geometry-based image shape descriptor in object shape detection and classification.
- Chapter 4 presents the experiments for application of proposed method for nerve persistence in image sequence.
- Chapter 5 concludes the thesis and suggests future directions of the research work presented in this thesis.

Chapter 2

Preliminaries and Backgrounds

This chapter reviews the proximal background for shape description, foundational techniques for image tessellation, relevant theories for topological computations in digital image, which accommodates the basis for conducting research for this thesis.

2.1 Computer Vision

Computer vision is multidisciplinary field that embraces various ways to perform tasks from acquiring, processing to analyzing digital images in order to interpret and yield numerical or symbolic meaningful information that is useful for the decision making process. This is a transformation process of visual images into sets of descriptions, which relies on sensor measurement and computational models designed for each specific tasks. Sensor measurement may depend on the reflected brightness, angle and position of the object, and camera parameters due to calibration and light sources; while computational models may be constructed based on the geometry, physics, statistics and learning theory[42].

Much similar to human's natural ability of perceiving and understanding natural scenes, computer vision can be described as a computational form of interpretation of vision representation captured electronically. It is also aimed to perform automate tasks that human visual system can do for electronically perceiving and understanding image[12, 92, 111, 9, 106, 53]. As such, the major tasks of computer vision in understanding image are object detection and object recognition. An approach that mimics human vision system is to construct a set of features obtained from digital images[73]. Such collections of features can be useful for classification, object detection and object recognition tasks. Some computer vision applications are described in[14]:

- Application of Morphological Operators in geosciences, material sciences, medical imaging
- Industrial Object Recognition: intelligent cameras
- Character Recognition in Industrial Production
- Motion Tracking
- 3-D Image Metrology for Industrial Production
- Reverse Engineering Using Optical Range Sensors
- 3-D Modeling of Objects from Image Sequences
- Three-Dimensional Fast Full-Body Scanning
- 3-D Model-Driven Person Detection

- Knowledge-Based Image Retrieval: is content-based image retrieval (CBIR)[93, 24, 65] with application on video retrieval systems. Two approaches have been focused for CBIR: the first approach is based on generating and storing object description[72, 18, 26, 116], and the other is used for matching similar objects based on color, texture, contour[63, 114, 28, 129, 34, 3, 104]. Also, video retrieval systems have been discussed with two differences using fixed pattern to classify image information[14]: Constructing video description every tenth frame and using color histograms[86, 35, 32] to monitor change of scenes based on key frame and dynamical mosaicing technique[84].

2.2 Topology of Digital Images

A digital image is a raster, two-dimensional, discrete space represented by a function g that is characterized by intensity and spacial coordinates (x,y): $g(x, y)$ [102]. Visual field of digital image is constituted by picture element known as picture point or pixel, of which intensity can be color or greyscale, is the corresponding intensity of light impacted on optical sensor. As such, two-dimensional image function can be represented as a matrix of rows and columns:

$$A(m,n) = \begin{bmatrix} g(1,1) & g(1,2) & g(1,3) & \dots & g(1,n) \\ g(2,1) & g(2,2) & g(2,3) & \dots & g(2,n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g(m,1) & g(m,2) & g(m,3) & \dots & g(m,n) \end{bmatrix}$$

In the context of topology, digital images can be described in terms of structures, where structures are group of features or pixels defined by probe functions. Topo-

logical properties of structures in digital images can be characterized by spatial and descriptive nearness. In detail, sets of structures(pixels or features) are deemed to be spatially near when they have pixels in common, and sets of structures are said to be descriptively near when they descriptively resemble each other with one or more feature values(color, histogram etc.).We can define a topological structure on a set X of picture elements as a structures with a set of subsets χ of X with properties as followings[102]:

(A.1) Every union of sets in χ is a set in χ.

(A.2) Every finite intersection of sets in χ is a set in χ.

Shape structures can be described as a collection of descriptive and spatial structures which are useful for constructing shape descriptors. Object shapes in digital image can be approximated with proximal $\breve{C}$ech complexes[37, 4].

2.3 Fundamental of Image Tessellations

Using Divide and Conquer approach[101], image space can be discretized into disjoint regions of triangles or polygons by using Voronoï approach(image tessellation for polygon) or Delaunay approach(image triangulation for triangles). Image tiling with Voronoï Tessellation or Delaunay Triangulation is constructed based on seeds, generating point or sites. Site points can be created with Scale Invariant Feature Transform (SIFT)[81], Speed-Up Robust Features (SURF)[56], image corners, image centroids[101]. Each seed generating method has its own advantages. Recall the mathematical representation for Voronoï partition for a given set of sites:

$$S = \{s_1, s_2, ..., s_k : i \in \mathbb{N}\}$$

Then the Voronoï $V(s_i)$ region is constructed as:

$$V(s_i) = \{x \in \mathbb{R}^n : \|x - s_i\| \leq \|x - s_j\|, \forall s_{i,j} \in S, i \neq j\} \tag{2.1}$$

With Euclidean distance is represented as $\|.,.\|$. Then the Voronoï diagram (or Dirichlet tessellation)[126, 128, 127] is a set of convex polygons for which the expression can be described as:

$$\mathbb{V} = \bigcup_{s_i \in S} V(s_i) \tag{2.2}$$

Delaunay Triangulation is planar segmentation introduced by B.N. Delaunay[31], which generates sets of triangles, including vertices and edges of the triangles, by joining two nearest generating points with straight line. And collections of triangles on the image space is referred as Delaunay mesh.[101]. In this thesis, experiments are performed based on Delaunay approach.

2.4 Image Geometry and Shape Description

Delaunay triangulation is used to segment image space into Delaunay meshes as a collection of filled triangles, formed by intersection of half spaces. Edge of half plane contains a line which is connected between two nearest generating points. Thus, they accommodate with a source for image geometry information with geometrical structures such as Nerve Structures[101, p. 69] and Complexes(Simplicial Complexes, Convex Set Systems, Delaunay Complexes and Alpha Complexes)[37, p. 61]. For this thesis, we stay focused on Nerve Structures and Simplicial Complexes.

2.4.1 Simplicial Complexes

According to Edelsbrunner[37, p. 74], simplicial complexes are used solely to represent topological spaces. As such, shape in topology of digital image can be represented in terms simplicial complexes.To reinforce the mathematical background, we now present the following definitions and properties regarding simplices and simpicial complexes

2.4.1.1 Simplices:

Definition. Let $u_0, u_1, ..., u_k$ be points in R^d. Point $x = \sum_{i=0}^{k} \alpha_i u_i$ is said to be affine combination[37, p. 62] of u_i if the $\sum \alpha_i = 1$.

Definition. A set of affine combinations is called affine hull[37, p. 62].

Definition. If and only if any two affine combinations $x = \sum \alpha_i u_i$ and $y = \sum \beta_i u_i$ are the same for $\alpha_i = \beta_i, \forall i$, then k+1 points are affinely independent.

Definition. If and only if the k vectors $u_i - u_0$ for $1 \leq i \leq k$, are linearly independent, then k+1 points are affinely independent.

Definition. If all α_i are non-negative, an affine combination, $x = \sum \alpha_i u_i$ is a convex combination[37, p. 62].

Definition. A convex hull is the set of convex combination.

Definition. A simplicial complex is a finite collection of simplices. A k-simplex is the set of convex combination $\delta = conv\{u_0, u_1, ..., u_k\}$ of k+1 affinely independent points with dimension $\dim\delta = k$. For lower dimension, k-simplices are named as

vertex for 0-simplex, edge for 1-simplex, triangle for 2-simplex and tetrahedron for 3-simplex.Edge, triangle and tetrahedron has, respectively, 2 vertices, 3 edges and 4 triangles as faces.

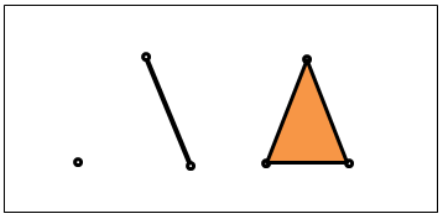


Figure 2.1: 0, 1, 2-simplex

Definition. Any subset of affinely independent points is affinely independent and also defines a simplex[37, p. 62].

Definition. A *face* of δ is the convex hull of a non-empty subset of u_i and it's *proper* if the subset is not the entire set. Then, $\theta \leq \delta$ if θ is a face and $\theta < \delta$ if it is a proper face of δ. δ is called a proper coface of θ if θ is a proper face of δ. δ has $2^{k+1} - 1$ faces because a set with size k+1 has 2^{k+1} subsets, all of the faces are proper except δ itself.[37, p. 62].

Definition. Let ϵ be all the proper faces. The boundary of δ is: $bd\delta = \bigcup \epsilon$. The interior of δ is: $int\delta = \delta - bd\delta$. A point $x \in \delta$, $\in int\delta$ if and only if $\forall \alpha_i > 0$.

2.4.1.2 Simplicial Complexes

According to Edelsbrunner[37, p. 63]:

Definition. A simplicial complex is a finite collections of simplices S : $\delta \in S$ and $\epsilon \leq \delta$

means $\epsilon \in S$, and $\delta, \delta_0 \in S$ means $\delta \cap \delta_0$ is either empty or a face of both.

Definition. K has the maximum dimension in all of its simplices.

Definition. The underlying space $|K|$ is the union of its simplices and the topology obtained from the ambient Euclidean space of simplices.

Definition. A polyhedron is the underlying space of a simplicial complex.

Definition. A triangulation of a topological space $\mathbb{X}$ is a simplicial complex S, including the homeomorphism between $\mathbb{X}$ and $|K|$. And if the topological space has a triangulation,then it's triangulable.

2.4.1.3 Abstract Simplicial Complexes

According to Edelsbrunner[37, p. 64]:

Definition. Abstract Simplicial Complexes is a finite collection of sets A that satisfies $\alpha \in A$ and $\beta \subseteq \alpha$ which yields $\beta \in A$. Set A contains it simplices. The simplex dimension is dim α = card $\alpha - 1$ and the complex dimension is the maximum dimension of each simplices. An abstract simplicial complex can be constructed by removing all simplices and keep only sets of vertices. We can have some of the following definitions for Abstract Simplicial Complexes.

Definition. A face of α is proper if $\beta \neq \alpha$, and is a non-empty subset $\beta \subseteq \alpha$.

Definition. The vertex set is defined as vertA $= \bigcup A$.

Definition. If set B: $B \subseteq A$ is an abstract simplicial complex, then B is a *subcomplex*.

Definition. Two abstract simplicial complexes are said to be isomorphic if there exists a bijection b : VertA → Vert B such that $\alpha \in A$ if and only if $b(\alpha) \in B$.

2.4.2 Nerve Structures

Nerve is a collection of simplicial complexes. The idea of nerve was introduced by Paul Alexandroff[5]. and later was studied intensively by Edelsbrunner and Harer[37], Eric, Gregory and Xavier[38], Leray[66], Michal and Henry [87] and Borsuk [19].

According to K. Borsuk, every polytope can be decomposed into brics, another name used for elementary simplexes. A polytope is the intersection of a number of finite closed half spaces[139]. The decomposition of a simplicial complex K covered by simplices $\Delta_1, ..., \Delta_n$ has the same nerve as K[19]. Since we will be using Delaunay triangulation to superimpose on the planar digital plane with a finite set of simplicial complexes, our entitled nerve is restricted to a collection of vertices, line segments and filled triangles in the Euclidean plane.

According to J. Peters,[97, p. 206], a **Nerve** is a planar simplicial complex K that contains simplices which share common intersection **nucleus** of the nerve. Providing a triangulation of a planar image space, a nerve of a simplicial complex K, NrvK, can be defined as:

$$NrvK = \{\Delta \subseteq K : \bigcap \Delta \neq \varnothing\} \tag{2.3}$$

On the planar image space, the simplices share proximal relationship to each other in the same nerve. Every nerve has, at least, a **nucleus** at the center, which is a collection of points that are intersected by the simplices belongs to that nerve. Nerves

can be constructed spatially or descriptively using contiguous simplexes or description parts of simplices that matches nucleus description[97, p. 206]. A **Delaunay mesh nerve** is a collection of simplices (vertices, edges and triangles) with vertices as generating points on digital image and are all adjacent to the center nucleus. A **maximal nucleus cluster (MNC)** is a Delaunay mesh nerve that has the maximal number of filled triangles surrounding the nucleus in the nerve[101, p. 71]. An example of an MNC in an image covered with Delaunay mesh nerves is shown in Figure 2.2.

Figure 2.2: Delaunay triangulation on an digital image with MNC

In Computer Vision, application in image object shape detection and object class recognition can be derived by constructing basic shape feature boundaries[7], shape

contour around the simplicial complex[97, p. 206]. A basic algorithm for 2-simplex based shape detection can be found in [97, p. 206].

2.4.3 Generating Points

Geometric structures overlaid on image constructed with Delaunay Triangulation is dependent on the types of generating points (or sites, seeds) that are used. In the context of image object shape detection, the most ideal generating points should locate on the edge boundary of image object shapes. A finite number of nerves will cover object shapes and shape detection or retrieval is then plausible. Our main interest is on generating points that are invariant to scaling, transformation, rotation and overlaying on edge of the image object shape. And there are a number of ways to generate sites on digital image, depending on the types of application:

2.4.3.1 Keypoint

Image keypoints have been used widely for extracting points on image with unique properties, which can be used as a collection of feature values for object recognition tasks[81, 56, 90, 135, 134]. Obviously, there are numerous keypoint locations on planar image space, so we have to locate keypoints that are most important. To elaborate more, the criteria for keypoint selection is based on being invariant to image translation, scaling, rotation, noise and small distortion[81, p. 2].The most common keypoint generators are Scale-Invariant Features Transform and Speed-Up Robust Features[56]

- **Scale-Invariant Features Transform(SIFT)**: Lowe's method accommodates a way to locate such unique keypoint by transform image into feature vectors that satisfies the above criteria. A Gaussian function $G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-x^2}{2\sigma^2}}$ is applied in scale space by convolving $G(x)$ with input image for every increment of σ. Keypoints are chosen at maxima and minima of difference of Gaussians function applied previously after a series of smoothing and resampling image[81]. SIFT keypoint stability and robustness is enhanced by discarding low-contrast candidate points and edge response points, and by assigning localized keypoints for dominant orientations.

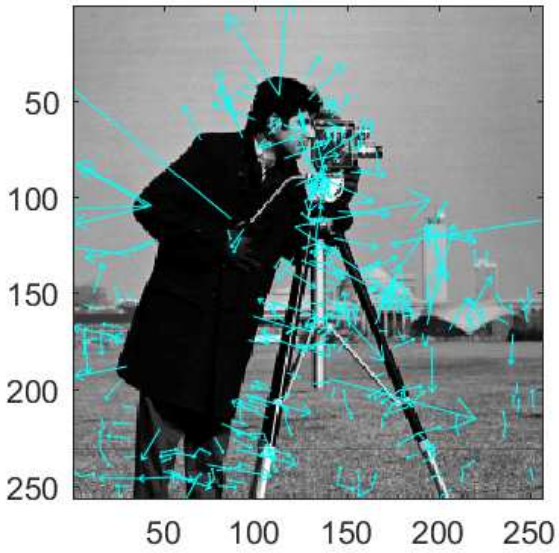


Figure 2.3: Keypoints detected with SIFT

- **Speed-Up Robust Features Transform(SURF)**: Another keypoint generator inspired by SIFT descriptor but it was faster in computation time and was claimed to be more robust against different image transformation than SIFT.

SURF follows the same principles as SIFT but the details in each phase are different. The algorithm has these major steps: interest point detection, interest point localization and interest point description and matching. SURF uses integral image to achieve fast computation for box type convolution filter[56, p. 3]. The sum of all pixel in the original image I can be evaluated quickly with integral image S:

$$S(x,y) = \sum_{i=0}^{x}\sum_{j=0}^{y} I(i,j) \tag{2.4}$$

After computing integral image, the sum of intensities is calculated at rectangle's corners. Keypoints detector was blob-like and was based on Hessian matrix. The determinant of the Hessian matrix is used to detect at locations and only the point with maximum determinant is selected. Unlike Hessian-Laplace detector by Mikolajzyk and Schmid[88], SURF also uses the determinant of the Hessian for choosing the scale[56], as done by Lindeberg[77]. The Hessian matrix $H(x,\sigma)$ with scale σ at a point x=(x,y) is defined as:

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{2.5}$$

Where $L_{ij}(x,\sigma)$ is the convolution of the second order derivative of the Gaussian $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image I(x,y) at the point x. The 9 x 9 box filters are approximation of a Gaussian with σ = 12 is used as lowest level(highest spatial resolution) for blob response maps[56]. An example of keypoint detection with SURF superimposed on image is Figure 2.4. Since SURF is faster and already implemented in Mathematica, we can consider using SURF for robust image keypoint detection.

Figure 2.4: Keypoints detected with SURF

2.4.3.2 Image Corner

Another remarkable approach for detecting interest point is using corner points as features. Corner points are defined as where the image structures such as edges intersect. Therefore, they are considered as domain for feature extraction for applications in motion detection, video tracking and object recognition.

Corner and interest point are interchangeably terms in literature.[133]. An interest point can be a corner but can also be a isolated point which is a local maximum or minimum intensity, i.e., line endings, point on a curve such as a ridge(concavity down) or valley(concavity up)[101, p. 184]. As a result, if only corners are detected, interest point will be included in corner points. And it's necessary to do local analysis to determine which corner points are real. As a consequence, detected corners are not very robust against image manipulation and distortion and it's often to introduce a redundancy filter to prevent the individual errors.The criteria for determining the quality of corner detection is based on whether the same corner is detected under

image transformation, rotation and translation.

The earliest algorithms for corner detection are **Moravec corner detection algorithms**[91]. This algorithms detect corner points that have low self-similarity. A patch at the center of the pixel is compared, based on similarity, to largely overlapping patches in the neighborhood to determine whether a corner is detected in each pixel. More similarity will result in lower number. Point of interest is indicated where the computed corner strength is locally maximal for any location in all computed locations, and the corner strength is defined as the smallest SSD(Sum of Squared Differences) between patches and its neighbors(horizontal, vertical and on two diagonals). As elaborated by the author, if an edge exists that is not in the direction of the neighbors, then the smallest SSD is large and the edge is incorrectly chosen as an interest point.

Subsequently, various corner-based detection algorithms have been developed:

- The Harris & Stephens Shi-Tomasi corner detection algorithms based on differential of the corner score subjects to direction[52, 121]
- Computing the corner location with subpixel accuracy by solving the point closet to all the tangent lines of the corner in a window with least-square solution[41]
- Multi-scale second moment matrix[76, 79, 75]
- Detecting point where the curvature of level curves and the gradient magnitude are both high at the same time[67, 68].

- Difference of Laplacian, Gaussian and Hessian[77, 76, 89, 82]
- Algorithm based on edge direction that changes rapidly[130].
- Smallest univalue segment assimilating nucleus(SUSAN)[112] and similar algorithm[122].
- Accelerated segment test[107].
- Automated algorithm for detecting interest points[123].
- Extension of Harris algorithm in space-time[70], spatio-temporal interest point generator[78, 60].

An example of image corner detected on an image in Figure 2.5:

Figure 2.5: Corner points detected

2.4.3.3 Edge Point

Edge points generators are quite popular in image processing for feature detection and object recognition. Edge points are found on regions of an image where the brightness changes sharply[23]. These points are typically connected into a set of curved line segments which we call edges. As a result, an edge tends to localize on the object of interest and are subject to feature detection and extraction[2].

2.4.3.4 Centroidal Point

Centroidal point or geometric centroid is the center of mass of a two-dimensional planar closed surface or a three-dimensional bounded solid. A two-dimensional planar closed surface (termed as Lamina) L has the surface density function $\sigma(x, y)$ and the mass M [8, 132]:

$$M = \int_L \sigma(x, y) dx dy. \tag{2.6}$$

The coordinates of the centroid are[131]:

$$\bar{x} = \frac{\int_L x\sigma(x, y) dx dy}{M} \tag{2.7}$$

$$\bar{y} = \frac{\int_L y\sigma(x, y) dx dy}{M} \tag{2.8}$$

An image region is a discretized region of Lamina, that is a bounded set of pixel point. The coordinates x_c, y_c of a bounded n x m rectangular 2D image region containing points with coordinates (x_i, y_i) for i=1,...,n [101, p. 204]:

$$x_c = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.9}$$

$$y_c = \frac{1}{m}\sum_{i=1}^{m} y_i \tag{2.10}$$

Basic steps to use image region centroids as generating points for Delaunay triangulation can be found in [101, p. 204].

2.4.4 Shape Description

Description of two-dimensional shapes can be categorized into two different schemes[119]:

External Representation, Spatial Representation or Contour-based shape representation[48, 97, 138]: Using object boundary and its features to represent shape. This method is subjects to boundary point, edge and line detection and computing features lies on the shape contour as a form of representation. Some boundary features are useful in studying image shape: [102]

- Bounding Box: smallest box containing image shape.
- Boundary Length: perimeter length surrounding the object shape.
- Boundary Set: set of all point along the boundary of the point cluster.
- Edge Length: number of pixels on an edge
- Edge Count: a count of the number of pixels in an edge in a digital image with respect to the gradient direction of each pixel in that edge.
- Edge Strength: ϵ is the average intensity of edge pixels, defined as $\epsilon = \frac{1}{n}\sum_{i=1}^{n} g(x_i)$ where n is the number of pixel and $g(x_i)$ is the i^{th} pixel's intensity.

- Edge Frequency: ϵ_f is the count of the number of edges in a region or entire image where each edge pixel has the same gradient orientation. $\epsilon_f = \sum_{i=1}^{n} \epsilon_i$. ϵ_i is the i^{th} edge.

Internal Representation, Descriptive Representation or Region-based shape representation[97, 138]: Using description of the region occupied by the object on the image plane. It maps features on the object regions with probe functions into a feature vector containing numbers in $\mathbb{R}^n$ [100]. A collection of one or more feature vector are called shape descriptors. Some types of shape features have been designed including:

- Shape signature using Fourier descriptor[58]
- Signature histogram[40]
- Bayesian methods for shape invariants[17]
- Zernike moments[64] and Hu spatial moments[57]
- Invariant curvature-based Fourier descriptor[1]
- Shape context for shape matching[13]
- Spectral features with hidden Markov models[22]

Some representation properties for judging the quality and effectiveness of shape descriptors are[119]: uniqueness, completeness, invariance to geometrical transformation, sensitivity and noise robustness.

Chapter 3 will elaborate for more detail and provide more insightful techniques for

Shape Description. Also, my proposed method for geometry-based shape description, as well as its application in shape classification, will be introduced in the next chapter.

2.5 Persistence

2.5.1 Persistent Homology

The main idea behind the theory of **Persistent Homology** is to measure the topological properties of the features, such that a sequence of subcomplexes(faces) of simplicial complexes is built and studied at different spatial resolution[37]. The purpose is to recognize the homological characteristics of features with relation to noise and perturbation. In order to gain deeper understanding of such properties of the homology group of simplicial complexes, a concept of *filtration* has been introduced[50].

Definition. The *filtration* of a monotonic function $f : K \to \mathbb{R}$ with K as a simplicial complex, is defined as a sequence of subcomplexes of K for the sublevel set $K(a) = f^{-1}(-\infty, a]$ denoted as $K_0, K_1, ..., K_n$ which satisfies:

$$\varnothing = K_0 \subseteq K_1 \subseteq ... \subseteq K_n = K$$

Definition. For $0 \leq i \leq j \leq n$, the inclusion of the underlying space $K_i \to K_j$ induces a homomorphism $f_p^{i,j} : H_p(K_i) \to H_p(K_j)$ for each dimension p. Thus, a sequence of homology group is present as a result of the filtration.

$$0 = H_p(K_0) \to H_p(K_1) \to ... \to H_p(K_n) = H_p(K)$$

Definition. The p^{th} *persistent homology groups* are the images of the above homomorphisms and the p^{th} persistent **Betti numbers** $\beta_p^{i,j}$ are the ranks of those groups. During the filtration, homology groups are destroyed and created. We don't have to know the exact manipulated structure of the homology groups during the process. We just need to know how the Betti numbers change during the filtration as it is sole metric for examining the existence and number of homology classes. A diagram called **Persistence Diagrams** is introduced to track the birth and death time of the homology classes[37, p. 181].

2.5.2 Persistence Diagrams

Persistent Betti numbers can be visualized in a two dimensional diagrams to monitor the relation of birth and death of classes. According to Herbert Edelsbrunner, let $\mu_p^{i,j}$ be the number of classes born at K_i and dying entering K_j for p dimension:

$$\mu_p^{i,j} = (\beta_p^{i,j-1} - \beta_p^{i,j}) - (\beta_p^{i-1,j-1} - \beta_p^{i-1,j}), \forall i < j$$

By drawing every pair of tuple (a_i, a_j) for every multiplicity $\mu_p^{i,j}$, we have a p^{th} *persistent diagram* of the filtration. On a persistence diagram, a class is represented as a point with its vertical distance to the diagonal line is the persistence of the class. All points will reside above the diagonal. An important relation between Betti number $\beta_p^{k,l}$ and $\mu_p^{i,j}$ can be yielded[37, p. 181]:

$$\beta_p^{k,l} = \sum_{i \le k} \sum_{j > l} \mu_p^{i,j} \forall 0 \le k \le l \le n$$

An example of persistence diagram with two points can be seen below. Each point on the diagram represents a tuple of birth and death times of a simplicial complex. By comparing the birth and death time of each different point, we can deduce a strong different topological property between them, and a similar topological property will be among those overlapping points. The x-axis is the birth time and y-axis is the death time[6].

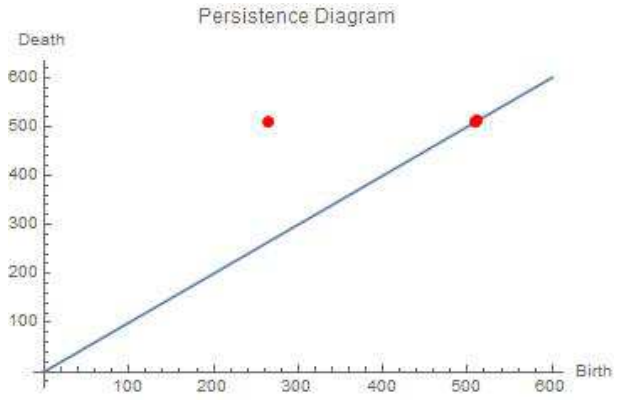


Figure 2.6: An Example of Persistence Diagram

In order to obtain persistent diagram, a filtration method is performed on the data points. Most popular filtration schemes are: A ball model filtration[71], $\check{C}$ech Complex Filtration[37, p. 72] and Vietoris-Rips Complex Filtration[37, p. 74].

2.5.3 R-Ball Model

Let $X = \{x_0, x_1, ..., x_n\}$ be the finite set of point cloud in $\mathbb{R}^n$. Balls with uniform radius r is cast on each point x_i (i=0,1,...n). The filtration of the point cloud is performed by increasing the balls' radius r. When a hole appears and disappears, respective

radius is recorded as birth and death time for the persistent homology. Hence, the complex with respect to radius r is $X_r := \cup_{x_i \in X} B(x_i, r)$where $B(x, r)$ is the ball with center x and radius r. For $r < 0$, $X_r := \varnothing$. An illustration for r-ball model filtration is shown below[94]:

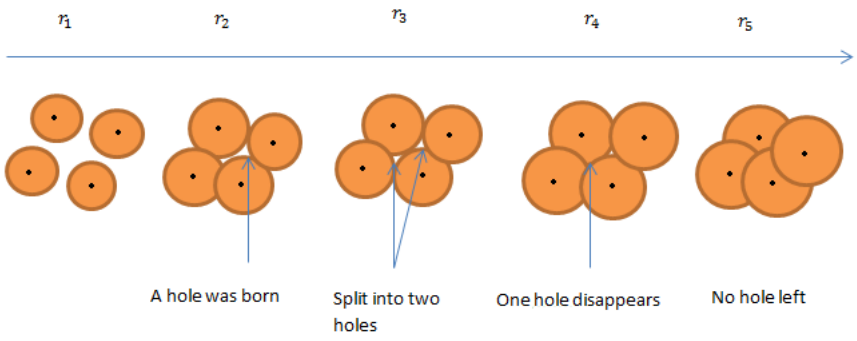


Figure 2.7: R-ball Model Filtration

2.5.4 Čech Complex Filtration

Given a set of points $X = \{x_0, x_1, ..., x_n\}$ in $\mathbb{R}^n$. A Čech Complex C_r is a collection of t-simplex in a subset X_{t+1} of X that contains any point $x' \in X_{t+1}$ such that x' lies inside another ball with the same radius r. Mathematically, let Čech Complex C_r is expressed as:[37, p. 72]

$$C_r = \{X_{t+1} \subseteq X | \bigcap_{x' \in X_{t+1}} B(x'_i, r) \neq \varnothing\} \tag{2.11}$$

The persistence diagram obtained from Čech complex filtration is the same as the persistence diagram from r-bal model [71].An example of Čech complex is shown below.

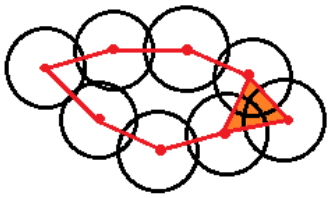


Figure 2.8: $\check{C}$ech Complex Filtration

2.5.5 Vietoris-Rips Complex Filtration

A Vietoris-Rips Complex is defined as a collection of 2- and higher dimensional simplices of any given point $x' \in X_{t+1}$ such that $d(x', x) \leq 2r$. That means the $\check{C}$ech Complex C_r is a subset or equal to Vietoris-Rips Complex R_r since R_r contains all the simplices satisfying the condition by the given edges [37, p. 74]. Mathematically,

$$R_r = \{X_{t+1} \subseteq X | d(x' \in X_{t+1}, x \in X) \leq 2r\} \tag{2.12}$$

And,

$$C_r \subseteq R_r \tag{2.13}$$

An illustration of Vietoris-Rips Complex can be seen below.

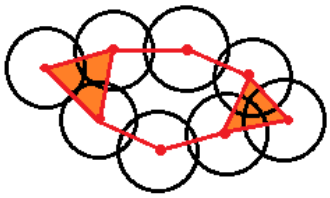


Figure 2.9: Vietoris-Rips Complex Filtration

2.6 Distance Measure

Distance measures are used to compare the level of similarity or stability of our set of features and persistence. For application in image classification, Euclidean Distance is used as a basic method to compare how sets resemble each other in the Euclidean space. In the topological space, two additional distance measures are considered for similarity measurement of the feature vectors: $\check{C}$ech Distance [25] and Hausdorff Distance[54, 108]. In order to measure stability of shape persistence, we use the Bottleneck Distance and Wasserstein Distance, which were introduced by Herbert Edelsbrunner in [37, p. 216-p. 221]

2.6.1 Similarity Measure for Classification

2.6.1.1 Euclidean Distance

Euclidean distance (or L^2) is quite popular in Engineering Computation, Signal Processing and Machine Learning. In Euclidean space $\mathbb{R}^n$, the norm of a vector (or

alternately, points) $x = (x_1, x_2, ..., x_n)$ (denoted as $\|x\|$) is:[100, p. 99]

$$\|x\| = \sqrt{x_1^2 + x_2^2 + ... + x_n^2} \tag{2.14}$$

The Euclidean distance metric betweer two vectors (or points) $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ in $\mathbb{R}^n$, is defined as:

$$\|x - y\| = \sqrt{\sum_{i=1}^{n}(x_i^2 - y_i^2)}, \forall i = 1, 2, ...n \tag{2.15}$$

2.6.1.2 Čech Distance

Descriptive feature vectors can be compared using Čech Distance. Given feature sets X and Y, the descriptive Čech Distance between sets X and Y is defined as[25, 102] :

$$D(Q(X), Q(Y)) = \inf \{d(\Phi(x), \Phi(y)) : \Phi(x) \in Q(X), \Phi(y) \in Q(Y)\} \tag{2.16}$$

where Φ is a set of probe functions that map feature points X,Y to real numbers representing feature values and:

$$d(\Phi(x), \Phi(y)) = \sum_{i=1}^{n} |\phi_i(x) - \phi_i(y)| : \phi_i \in \Phi \tag{2.17}$$

2.6.1.3 Hausdorff Distance

The Hausdorff distance is used to measure how far two sets of a metric space are apart. Let X and Y be two non-empty sets of a metric space (M, d), then the Hausdorff distance between them is defined as[54, 108] :

$$d_H(X, Y) = max\{\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)\} \tag{2.18}$$

2.6.2 Stability Measure for Persistence

Two distance metrics are used to measure the similarity between persistence diagrams: bottleneck distance and Wasserstein distance[37, p. 216-p. 221]. Stability of persistence homology is referred to as being stable during perturbation (filtration), or in other words, robustness against noise. And, similarity between persistence diagrams reflects upon stability of persistence homology during filtration.

2.6.2.1 Bottleneck distance

The bottleneck distance is a crude way to compute general results of the persistence diagrams [37, p. 216]. Given X and Y as the two persistence diagrams, then their bottleneck distance is expressed as:

$$W_\infty(X,Y) = \inf_{\eta:X\to Y} \sup_{x\in X} \|x-\eta(x)\|_\infty \tag{2.19}$$

Where, η is the bijections that maps $X \to Y$, and the infimum over all distance between two points $x = (x_1, x_2)$ and $y = (y_1, y_2)$ is

$$\|x-y\|_\infty = max\{|x_1-y_1|, |x_2-y_2|\} \tag{2.20}$$

2.6.2.2 Wasserstein Distance

The Wasserstein distance has more attention to detail of the persistence diagrams but additional criteria should be met for stability[37, p. 216]. The degree q of Wasserstein distance is introduced between diagram X and Y. The Wasserstein Distance is

computed by finding the infimum for the sum of q-th powers of L_∞ distances between points in the persistence diagram[37, p. 219]:

$$W_q(X,Y) = [\inf_{\eta: X \to Y} \sum_{x \in X} \|x - \eta(x)\|_\infty^q]^{1/q} \tag{2.21}$$

If q goes to infinity, the bottleneck distance becomes the limit of the Wasserstein distane[37, p. 220].

Chapter 3

Geometry-Based Shape Descriptors for Object Shape Detection

3.1 Introduction

In this chapter, I introduce a novel and fast object shape detection technique based on geometry of Delaunay triangulation, which can automatically detect, capture the boundary point of the object shape and yield the one MNC as the only region that cover image object shape. Object shape detection plays an important role in many computer vision and pattern recognition applications such as image retrieval , object recognition and object classification. According to [115], object detection is the process of applying a recognition algorithm to every possible sub-window in an image to find likely regions that overlays on particular objects. Such algorithm should be error-prone and effective. A number of specific-purpose detectors have been developed including face detector, pedestrian detectors, eigenfaces[124], instance (known object)

recognition [81]. A typical approach for object detection involves the basic following steps: image preprocessing, feature extraction and classifying algorithms. Image preprocessing step includes a variety of techniques whose sole purpose is to transform image into feature space for feature collection. Depending on the object detection tasks, image prepocessing may involve removing noise, image segmentation, edge detection, filtering, morphology transforming, color conversion, correction, shareholding, enhancement, artificially mirroring, rotating, scaling, translating, etc. Feature extraction is the task of collecting meaningful properties of image on feature space that are useful for recognition and classification task. A feature vector is constructed based on feature space that is useful for subsequent tasks. For object detection, feature vectors are used to represent shape, known as shape descriptors. Classifying algorithms consist basic distance-based classifier for simple object-similarity comparison, or more advance, machine learning algorithm such as support vector machine (SVM)[36] or neural networks(NN)[51] in respect to the complexity of the recognition tasks.

3.2 Overview of Shape Descriptor

As discussed, shape description techniques are classified into two methods: contour-based methods and region-based methods. For each method, there are two different approaches within: structural approaches and global approaches[138]. This section provides a summary of related work for various kinds of shape descriptors that have been designed.

3.2.1 Contour-based shape descriptor

Contour-based shape descriptor relies on features that resides on the boundary proximity. Two approaches are used to construct the descriptor: Continuous approach (global) and discrete approach (structural).

3.2.1.1 Continuous approach

With continuous methods or global methods, shape boundary is not divided into subsets. Shape similarity is determined by comparing the distance metric (Euclidean or Hausdorff) of the multi-dimensional feature vector created from the shape boundary. For this approach, numerous types of shape descriptors have been suggested[138]:

- Simple shape descriptors: area, circularity, eccentricity, major axis orientation, bending energy[62], convexity, ratio of principle axis, circular variance and elliptic variance[103].

- Correspondence-based shape matching with Hausdorff Distance to measure shape similarity by point-to-point matching[29].

- Shape signature: using boundary points to derive a one dimensional function to describe a shape. There are many type of shape signatures: centroidal profile, complex coordinates, centroid distance, tangent angle, cumulative angle, curvature, area and chord-length[30, 125, 136]. Shift matching is involved to find similarity between two shapes. Shape signatures are susceptible to noise and small boundary variant.

- Boundary moments: using moments of shape signature to reduce the dimensions

of the boundary descriptor. The normalized boundary moments are invariant to shape transformation (translation, rotation and scaling)[85].

- Elastic matching: proposed by Bimbo and Pala [15]. For this technique, a deformed template is generated and the similarity between the shape of the template and the shape of the image object is calculated by minimizing the compound function $F = S + B + M$, where S and B is the strain energy and bend energy, and M is the measurement of the overlapping between the deformed template and the image object[138].

- Autoregressive model: using a stochastic modeling of 1-D autoregressive function to calculate the shape descriptors[61].

- Scale space method: shape is described with scale space representation by tracking the inflections positions in the boundary of shape after passing through low-pass Gaussian filters with variable widths. Trivial inflections on the boundary can be eliminated by increasing the width of Gaussian filter, thus the shape becomes smoother. This process is refereed as the interval tree[10].

- Spectral transform: using Fourier descriptor[27] and wavelet descriptor[120] as spectral descriptor to overcome noise sensitivity and boundary variations. Such descriptors are obtained with spectral transform of 1-D shape signature above. Comparing to Fourier descriptor, wavelet descriptor accommodates more resolutions for both spatial space and spectral space, but at a cost of sacrificing frequency resolution for higher spatial resolution and complicated matching scheme[138].

3.2.1.2 Discrete approach

Another class of shape representation results from the discrete approach, where shapes are discretized into boundary segments named as primitives[138]. The discrete approach constructs an organized set of selected primitives to represent shape. A string made up of chained features such as sides of polygons, spline, curvature, curve fitting[95] is formed as a basis for direct matching description or as an input for classifier using more advance machine learning algorithm.

- Chain code representation: Introduced by Freeman[43], chain code uses oriented, unit-sized sequence of line segments to represent a shape object. A sequence of vector with unit length and limited orientation, namely a unit-vector, is used to describe an arbitrary curve on the object. In computation, boundary points are related to the nearest point on the grid that was superimposed on the image boundary. At first, a point is initiated, then the subsequent chain code is created with 4-direction or 8 direction for matching process. The chain code technique is usually high-dimensional and susceptible to noise.[138].

- Polygon decomposition: implemented by Groskey and Mehrotra[49], this technique disintegrates shape boundary into polygon primitives (line segment, vertex). Feature vector consists a string of four parts: internal angle, vertex distance and vertex's x and y coordinates. These elements varies as object shape scales up or down, so the feature vector is not invariant to transformation. Editing distance is used to compare the similarity between two feature vectors. To ensure robustness, only five sharpest vertices are selected for each shape. The feature vector is restructured into binary tree or m-nary tree. And the match-

ing process have two parts: feature-by-feature matching and model-by-model matching. In feature-by-feature matching part, provided features of a query shape is searched through the index tree. If there is any similar data feature found, then the list of entitled shape is retrieved. In the second part of model-by-model matching, the retrieved model and the query shape is matched with editing distance of their feature vectors.

- Smooth curve decomposition: Implemented by Berretti et al.[109], tokens (another term for primitives) are retrieved by using the curvature zero-crossing points from a Gaussian smoothed boundary. Maximum curvature and orientation are set for tokens' features and the weighted Euclidean distance is used to compare the similarity. Token is indexed in the feature database by M-tree. Similar shapes are retrieved from the database by : first, obtaining the similar tokens of the query shape, and second, matching the query shape and similar shapes with model-by-model matching method. Obviously, this approach is not rotation invariant because curve orientation is selected as feature.

- Scale space method : A model-by-model matching technique from the perspective of shape in scale space, implemented by Dudek and Tsotsos in[33]. This method considers obtaining shape primitives first from a curvature-tuned smoothing technique. A segment descriptor is formed by gathering segment's length, ordinal position and curvature tuning from each primitive. To describe the shape, a string is created from segment descriptors. In order to compute the similarity score of the two string descriptors, a technique using dynamic programing with model-by-model matching is used. Shape features are integrated

into a curvature scale space to match shape in different scale. This can improve robustness and computation time. But the disadvantage of this approach is that the descriptors are not scale invariant because of the inclusion length of the segment descriptor[138].

- Syntactic analysis: a technique uses a set of *predefined primitives* to describe shape[44]. In this method, the set of predefined primitives is named the codebook and the codewords are primitives. A shape is represented as a grammatical string of codewords. Shape matching is done by finding the minimal number of edit operation to modify one string to match another string.

- Shape invariants: Construction of shape descriptor based on invariant theory. Generally, invariant theory studies the collection of transformations that have the properties of being composed and inverted. Name of the invariant is accustomed to the number of features used, i.e. unary invariant for singe feature, binary invariant for two feature, ternary invariant, quaternary invariant and so on. A number of popular geometric invariants have been designed: cross-ratio, length ratio, distance ratio, angle, area[74] and triangle[59]. For algebraic invariants: determinant, eigenvalues[113]. Geometric invariants and algebraic invariants are most favored in circumstances where straight line or algebraic curves can be used to represent boundary[138].

3.2.2 Region-based shape descriptor

Region-based shape descriptor considers using all the pixel in the shape region to represent shape as contradicting to using only the boundary pixel in contour base

methods. Most typical technique in this approach is exploiting moment descriptors. Region-based shape descriptors are categorized into two approaches: global and structural approach[138].

3.2.2.1 Global approach

Global approach considers describing shape as a single numeric feature vector. And the metric distance's still used in comparing shape similarity.

- Geometric moment invariants: The most popular technique using moment invariants is Hu's moment[57]. The Hu method is based on the algebraic theory of Boole, Cayley and Sylvester:

$$m_{ab} = \sum_{x} \sum_{y} x^a y^b f(x, y) \; a, b = 0, 1, 2, ... \tag{3.1}$$

  Geometric moment is a set of moment invariants derived from a nonlinear combinations of the lower order moments, which share the properties of transformation-invariant. A normalization process called zscore normalization[105] is used to normalize the obtained moment invariants since they have small values.

- Algebraic moment invariants: introduced by Taubin and Cooper[117], it is calculated from the first m-central moments to derive predefined matrices $M_{i,j}$ with the algebraic moment invariants as the eigenvalues. Matrix $M_{i,j}$ has scaled factors of the central moments as the elements. This technique differs from Hu's geometric moment invariants that it's invariant to affine transformation and can be created to arbitrary order[138].

- Orthogonal moments: by substituting the transform kernel $x^a y^b$ in 3.1 with general kernel $P_a(x)P_b(y)$, we can have a generalized version of the moment. Introduced by Teague[118], orthogonal moments are formed by replacing $x^a y^b$ in 3.1 with Zernike polynomial and Legendre polynomial.

  For Zernike moments:

$$Z_{ij} = \frac{n+1}{\pi} \sum_x \sum_y V_{ij}^*(x,y) f(x,y),\ x^2 + y^2 \leq 1 \tag{3.2}$$

  With $V_{ij}(x,y) = V_{ij}(\rho cos\theta, \rho sin\theta) = R_{i,j}(\rho) exp(kj\theta)$

  And, $R_{i,j}(\rho) = \sum_{t=0}^{(i-|j|)/2} (-1)^t \times \frac{(i-t)!}{t!((i+|j|)/2-t)!((i-|j|)/2-t)!} \rho^{i-2t}$, ρ and θ are the radius and angle of pixel at the center of gravity of shape

  For Legendre moments:

$$L_{ij} = \frac{(2i+1)(2j+1)}{4} \sum_x \sum_y P_i(x) P_j(y) f(x,y) \tag{3.3}$$

  With $P_i(x) = \frac{1}{2^i i!} \frac{d^i}{dx^i} (x^2 - 1)^i$

  The main advantage of orthogonal moments is precise reconstruction of described shape so that shape information is optimally exploited.

- Generic Fourier descriptor: introduced by Zhang and Lu[137] to overcome the limitations of Zernkie moment descriptors, which are complex computation, inconsistent features and feature loss during extraction[138]. The generic Fourier

descriptor is obtained by transforming polar-raster sampled shape image with 2-D Fourier:

$$PF_2(\rho,\phi) = \sum_r \sum_i f(r,\theta_i) exp[j2\pi(\frac{r}{R}\rho + \frac{2\pi i}{T}\phi)] \tag{3.4}$$

Where R and T are the radial frequency resolution and angular frequency resolution and $0 \leq r < R$ and $\theta_i = i(2\pi/T)(0 \leq i < T); 0 \leq \rho < R, 0 \leq \phi < T$[137].

The Manhattan distance is used to compare the similarity between generic Fourier descriptor of the two shapes.

- Grid based method: proposed by Lu and Sajjanhar[83], this technique uses a grid of cells to superimpose on a shape, and then scans the grid from left to right, top to bottom and results in a bitmap. The shape is described as a binary feature vector by assigning 1 to the cells covered by the shape and 0 for those not covered. The similarity between shapes is computed with binary Hamming distance. The main advantages of this technique are simple computation and MPEG-4 compatibility. The main disadvantage is due to the major axis being unreliable and susceptible to noise[138].

- Shape matrix: introduced by Goshtasby[47], is a similar technique to normal raster sampling. But, a polar raster of concentric circles and radial lines is superimposed on the center of the mass of the image shape instead of superimposing with a normal square grid. Shape is scale normalized with maximum shape radius and is then sampled for binary value at the intersections of the

circles and radial lines. To construct the shape matrix, matrix columns are the circles and the rows are radial lines. Shape matrix as shape descriptor is not subject to translation, rotation and scaling but matching shape matrix requires expensive computation[138].

3.2.2.2 Structural approach

In this approach, a region-based shape descriptor divides shape regions into subsets useful for shape description.

- Convex hull: a region R is convex if and only if a straight line segment between any two points of the region, remains inside the region. A convex hull is the smallest convex region that R belongs to. The difference between the convex hull and the region R is called the convex deficiency [138]. Convex hull can be constructed with boundary tracing method[85] and morphological methods[46]. Convex deficiency usually has insignificant or purposeless elements on the boundary under the effect of noise or digitization of natural scenes. The first step should be smoothing the boundary before converting to convex hull. One of the effective technique is using polygon approximation[138] since it's cheaper to compute and extract convex hull. The construction of shape descriptor as a string of concavities can be straightforward with only important convex deficiencies on the boundary. A recursive process can be implemented to have a concavity tree until all convex deficiencies are convex and this results in a full representation of the shape. Useful description of concavity can be used as features: area, bridge, length, etc. Shape is represented as a string or graph

,hence shape similarity matching requires string or graph matching[138].

- Medial axis: using region skeleton to represent shape. A skeleton is defined as a set of connected medial line along the figure limbs[30]. The idea behind skeleton is that the irrelevant information is eliminated and only the useful topological structure information is retained for recognition. Introduced by Blum, skeleton representation can be obtained with *medical axis transform*[16]. The skeleton descriptor can be broken down into segments for graph representation depends on particular tasks. And, graph matching is used to compare similarity. Medial axis poses a challenging computation in implementation, and having high noise sensitivity[138].

3.3 Geometry-Based Shape Descriptors

This section introduces geometry approach to construct shape descriptor based on nerve complexes. Also, a brief description of proximity and Edelsbrunner-Hare nerve complexes is introduced. In addition, the proposed technique for constructing geometrical shape descriptor with MNCs, as well as its application in shape classification are presented at the end of the section.

3.3.1 Geometry of Image Triangulation

Delaunay triangulation superimposed on digital image accommodates a basis for analysis of meshes covering shape region on images. As mentioned in chapter 1. Delaunay triangulation divides an image into sets of vertices, segments and triangle. which are referred as 0,1,2-simplices. Let δ denotes the proximal relationship between two

nonempty set A and B in a space X. And, let clA be the **closure** of A, then clA is defined as [97]:

$$clA = \{x \in X \,:\, x\,\delta\,A\} \tag{3.5}$$

A filled 2-simplex consist the boundary and its interior. Then the interior of A is: intA = clA - bdyA. And the filled 2-simplex (**filled triangle**) is defined as: filA = intA U bdyA .

A basic algorithm for triangualting a digital image with Delaunay approach is as below:

Algorithm 3.1: Image Triangulation algorithmn

Input : Read digital image img.

Output: Delaunay Mesh M covering image.

1 $Vertices \longleftarrow SiteGenerator$;

2 $img \longmapsto VerticesCoordinates$;

3 $D \longleftarrow VerticesCoordinates$;

4 /* D contains SiteGenerator coordinates used for triangulating image. */ ;

5 $D \longmapsto DelaunayMesh\ M$;

6 $M \longmapsto img$;

3.3.2 Computational Proximity

According to James Peters[100], computational proximity provides an algorithmic approach to analyze nonempty sets to determine proximal relation, namely, *spatial proximity* and *descriptive proximity*. **Proximities** are defined as the mathematical

relations of nearness between nonempty sets and a **proximity space** encompasses dimensions that proximities exist between nonempty sets. Spatial proximity is relation of nonempty sets that are spatially close in Euclidean space, whether they are asymptotically close or overlapping with common points. Descriptive proximity is another relation of nonempty sets for having one or more matching descriptive feature in descriptive space. And, **strong proximity** encompasses special nearness relation that nonempty sets have common elements in Euclidean space.

To revisit the computational proximity[100], $\check{C}$ech proximity axiom [25] (denoted as δ_C) and Lodato proximity axiom [80] (denoted as δ_L)are introduced:

Let X be a nonempty set and $A, B, C \subset X$.

$\check{C}$ech Proximity Axioms[25, p. 439]:

(**P1**) $\varnothing \notin A, \forall\, A \subset X$.

(**P2**) $A\,\delta\,B \Leftrightarrow B\,\delta\,A$.

(**P3**) $A \cap B \neq \varnothing \Rightarrow A\,\delta\,B$.

(**P4**) $A\,\delta\,(B \cup C) \Leftrightarrow A\,\delta\,B \,or\, A\,\delta\,C$.

Lodato Proximity Axiom(δ_L) [80]:

δ_L satisfies the $\check{C}$ech proximity axioms and

(**P5**$_L$) $A\,\delta_L\,B$ and $|b|\,\delta_L\,C$ for each $b \in B \Rightarrow A\,\delta_L\,C$.

3.3.2.1 Descriptive Proximity

Basic *object features* such as *object surface area* and *object location* are considered for **Descriptive Proximities**[97]. Object characteristic and object location are sets of

real value feature vectors mapped by probe functions $\phi : X \longrightarrow \mathbb{R}$. A simple feature vector for computational descriptive proximities consists object color and centroid coordinates: $\Phi(A) = (\phi_L(A), \phi_1(A), ...\phi_n(A))$.

Nonempty sets that share the same real value of feature vectors as a result of the mapping probe function ϕ, are said to have Descriptive Intersection.

Mathematical expression for **Descriptive Intersection**[97]:

(Φ) $\Phi(A) = \Phi(x) \in \mathbb{R}^n : x \in A$, is a set of feature vectors.

$(\underset{\Phi}{\cap})$ A $\underset{\Phi}{\cap}$ B $= x \in A \cup B : \Phi(x) \in \Phi(A)$ & $\Phi(x) \in \Phi(B)$.

Given two nonempty sets A and B:

$A \underset{\mathbb{R}}{\longrightarrow} \Phi(A) = (\phi_1(A), \phi_2(A), ..., \phi_n(A) \in \mathbb{R}^n$.

$B \underset{\mathbb{R}}{\longrightarrow} \Phi(B) = (\phi_1(B), \phi_2(B), ..., \phi_n(B) \in \mathbb{R}^n$.

Then:

$A \, \delta_\Phi \, B \Leftrightarrow \exists \, x \in A, y \in B : \phi(x) = \phi(y)$

The extension of Lodato Axioms for Descriptive proximity can be deduced:

Descriptive Lodato Axioms[102]:

(**DP0**) $\varnothing \not\delta_\Phi A, \forall A \subset X$.

(**DP1**) $A \, \delta_\Phi \, B \Leftrightarrow B \, \delta_\Phi \, A$.

(**DP2**) $A \underset{\Phi}{\cap} B \neq \varnothing \Rightarrow A \, \delta_\Phi \, B$.

(**DP3**) $A \, \delta_\Phi \, (B \cup C) \Leftrightarrow A \, \delta_\Phi \, B \, or \, A \, \delta_\Phi \, C$.

(**DP4**) $A \, \delta_\Phi \, B$ and $|b| \, \delta_\Phi \, C$ for each $b \in B \Rightarrow A \, \delta_\Phi \, C$.

According ton James Peters[97], strong proximity $\overset{\ll}{\delta}$ of intersecting nonempty sets

can be extended to the descriptive strong proximity $\overset{\twoheadrightarrow}{\delta}_{\Phi}$.

Let A, B, C be nonempty sets in a topological space X: A, B, $C \subset X$ and $x \in X$. Then the d*escriptive strong Lodato proximity* $\overset{\twoheadrightarrow}{\delta}_{\Phi}$ is a relation defined between two sets if they satisfy with following axioms:

Descriptive Strong Lodato proximity[97, p. 214]:

(**dsnN0**) $\varnothing \not\overset{\twoheadrightarrow}{\delta}_{\Phi} A$, $\forall\ A \subset X$, and $X \overset{\twoheadrightarrow}{\delta}_{\Phi} A, \forall\ A \subset X$

(**dsnN1**) $A \overset{\twoheadrightarrow}{\delta}_{\Phi} B \Leftrightarrow B \overset{\twoheadrightarrow}{\delta}_{\Phi} A$.

(**dsnN2**) $A\ B \Rightarrow A \underset{\Phi}{\cap} B \neq \varnothing$.

(**dsnN3**) If $\{B_i\}_{i \in I}$ is an arbitrary family of subsets of X and $A \overset{\twoheadrightarrow}{\delta}_{\Phi} B_i$ for some $i^* \in I$ such that $\text{int}(B_{i*}) \neq \varnothing$, then A $\overset{\twoheadrightarrow}{\delta}_{\Phi} \bigcup_{i \in I} B_i$.

(**dsnN4**) intA $\underset{\Phi}{\cap}$ intB $\neq \varnothing \Rightarrow$ A $\overset{\twoheadrightarrow}{\delta}_{\Phi}$ B.

The following conditions are assumed for each descriptive strong proximity:

(**dsnN5**) $\Phi(x) \in \Phi(int(A)) \Rightarrow x \overset{\twoheadrightarrow}{\delta}_{\Phi} A$.

(**dsnN6**) $\{x\} \overset{\twoheadrightarrow}{\delta}_{\Phi} \{y\} \Leftrightarrow \Phi(x) = \Phi(y)$.

Shape interiors mapped by the probe functions are basic requirement for shape matching according to axioms (**dsnN5**).

3.3.3 Edelsbrunner-Harer Nerve

Let ζ be a nonempty set. According to Edelsbrunne[37], an Edelsbrunner-Harer nerve ζ ($Nrv\zeta$) consists of all non-empty subcollections whose sets have a non-empty common intersection.

$$Nrv\zeta = \{X \subseteq \zeta : \bigcap X \neq \varnothing\} \tag{3.6}$$

Recalling that, a nonempty set consists a set of its interior and another set of its boundary. A nonempty set is open iff its boundary is an empty set. And, a nonempty set is closed iff its boundary is not an empty set. Depend on the distance proximity value that we choose, two set can be relatively proximal. For instance, Let A and B be nonempty set in a topological space X. For a chosen $\epsilon \geq d(A, B)$, then, $A \, \delta \, B$ is true[97, p. 215].

James Peters further extends the concept of Edelsbrunner-Harer nerve in the context of proximity[97, p. 216] with the *closure nerve*.

A **closure nerve** (NrvCL ζ) is defined as:

$$NrvCL\zeta = \{X \subseteq \zeta : \bigcap clX \neq \varnothing\} \tag{3.7}$$

If X is restricted to a filled 2-simplex and intersection of the closure of X is a vertex, we then have a new expression for **closure nerve** in triangulated planar surface cite[p. 216]jamesheela:

$$NrvCLt\zeta = \{\Delta \in NrvF : \bigcap cl\Delta \neq \varnothing\} \tag{3.8}$$

For descriptive proximity: $NrvA \; \delta_\Phi \; NrvB$ if $\Delta_A \cap \Delta_B \neq \varnothing$ with $\Delta_A \in NrvA$, $\Delta_B \in NrvB$.

And, for descriptive strong proximity: $NrvA \; \overset{\lll}{\delta}_\Phi \; NrvB$ if $\Delta_A \underset{\Phi}{\cap} \Delta_B \neq \varnothing$ with

$\Delta_A \in NrvA$, $\Delta_B \in NrvB$.

We have the following theories based on the facts of our definition of nerves.

Lemma[97, p. 217]: Each **closure nerve** $NrvCLt\zeta$ have a nucleus.

Proof[97, p. 217] All filled triangles in the cluster are adjacent to each other and share a common vertex, which is the nucleus.

Definition[97, p. 217]: A pair of filled triangle(2-simplex) ΔA, ΔB in a triangulated regions are **separated triangles**, provided $\Delta_A \cap \Delta_B \neq \varnothing$ (the triangles have no points in common) or ΔA, ΔB have an edge in common and do not have a common nucleus vertex.

Theorem[97, p. 217]: A nonempty set of vertices V in a space X covered with filled triangles. If v, v' $\in$ V are vertices of separated filled triangles on X, then the space has more than one nerve.

Proof[97, p. 217] Since v and v' belongs to separated filled triangles, we can always find at least two separated nerves that each nerve contain each separated triangle.

Theorem[97, p. 217]: Filled triangles in nerves with a common edge are strongly near nerves.

Proof[97, p. 217] From the definition of $\overset{\ll}{\delta}$.

Theorem[97, p. 217]: Strongly near nerves are descriptively near nerves.

Proof[97, p. 217] Strongly near nerves has intersections that descriptively belong to each nerves. Thus, they are descriptively near.

Theorem[97, p. 217]: Nerves containing interior regions with matching descriptions are strongly near nerves.

Proof[97, p. 217] From the definition of $\overset{\approx}{\delta}$.

3.3.3.1 Nerve Complexes

Nerve complexes based on keypoint are structures of 0,1,2- simplex, obtained when an image planar sure is triangulated. Every nerve complexes have a nucleus which is a keypoint, surrounded and shared in common by adjacent 2-simplices. Hence, a nerve complex is characterized by number of interior 2-simplices or the number of 1-simplice connecting the cuncleus.

As an image is triangulated, image geometry is covered by sparse nerve complexes that depend on the location of the keypoints, therefore, the type of keypoint generator is used

An algorithm for grouping a nerve complex at a chosen keypoint:

Algorithm 3.2: Finding Nerve at given keypoints

Input : Delaunay Mesh cover an image img.

Output: Nerve complexes.

1 /* Break down Delaunay Mesh into set of vertices */ ;

2 $SetofVertices\, V \longleftarrow DelaunayMesh\, M$;

3 /* Select a number of vertices */ ;

4 $V \longmapsto SelectedVertices\, V_s$;

5 /* Initiate an empty set for MNCs */ ;

6 $\Delta \leftarrow \varnothing$;

7 **foreach** $v \in V_s$ **do**

8 Find nerve complex triangle Nrv K adjacent to v $\longrightarrow \Delta_v$;

9 $\Delta \leftarrow \Delta_v$;

10 Return Δ;

3.3.3.2 Maximal Nucleus Cluster

Recall that, **maximal nucleus cluster** or **maximal nerve clusters** (MNC) is the nerve with the maximum number of triangles (2-simplices) with a common vertex. MNCs may cover a fraction or the entire region of image object shape. Thus they are useful for capturing an image object shape.

An algorithm for finding the MNCs on a triangulated digital image is as below:

Algorithm 3.3: Finding MNCs

Input : Delaunay Mesh cover an image img.

Output: Maximal Nerve complexes.

/* Break down Delaunay Mesh into set of vertices */ ;

$SetofVertices\ V \longleftarrow DelaunayMesh\ M$;

/* Initiate an empty set for MNCs */ ;

$\Delta \leftarrow \varnothing$;

$N \leftarrow 0$;

foreach $v \in V$ **do**

    Find triangles adjacent to v $\longrightarrow NumberofTriangles\ N_v$;

    N $\leftarrow N_v$;

    $N_{max} \leftarrow$ *maximum of* N;

    Find location of vertex with N_{max} in N $\longrightarrow Loc_{vmax}$;

    Find triangles adjacent to $vmax$ at $Loc_{vmax} \longrightarrow \Delta_{MNC}$;

    $\Delta \leftarrow \Delta_{MNC}$;

Return Δ;

An example of detected MNCs:

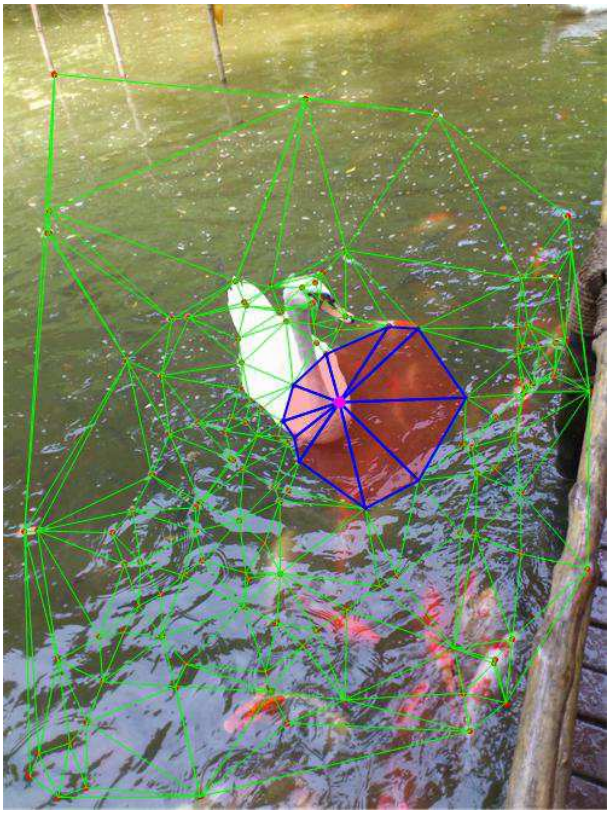


Figure 3.1: Detected MNCs coverting a portion of object shape

3.3.3.3 Strongly near MNCs

Near MNCs have close proximity relation. On a triangulated digital image, near MNCs overlap with 0,1 or 2-simplices. Strongly near MNCs overlap each other with 2-simplices. Usually, there are at least two 2-simplices in overlap region between two strongly near MNCs if these MNCs are all convex set. It's because the nucleus of each MNCs is the boundary vertex of the other MNCs. Thus, a larger part of image object shape is covered by overlap MNCs

To find the overlap of strongly near MNCs, we use algorithm below:

Algorithm 3.4: Finding MNCs overlap

Input : Delaunay Mesh cover an image *img*.

Output: Maximal Nerve complexes overlap.

```
/* Break down Delaunay Mesh into set of vertices */ ;
SetofVertices V ⟵ DelaunayMesh M;
/* Initiate an empty set for MNCs */ ;
Δ ← ∅;
N ← 0;
foreach v ∈ V do
    Find triangles adjacent to v ⟶ NumberofTriangles N_v;
    N ← N_v;
    N_max ← maximum of N;
    Find location of vertex with N_max in N ⟶ Loc_vmax;
    Find triangles adjacent to vmax at Loc_vmax ⟶ Δ_MNC;
    Δ ← Δ_MNC;
Return Δ;
foreach pair of MNCs in Δ do
    Find IntΔ = ⋂Δ;
Return IntΔ;
```

An example for detecting overlap MNCs is below, which is an example of the use of an image hole centroid as keypoint generator.

Figure 3.2: MNCs overlap on triangulated digital image

3.3.3.4 Order of Nerve Complex

The nerve complexes are characterized by the number of triangles (2-simplexes) attacted to a single vertex. Thus, it's reasonable to introduce the order of nerves as the number of adjacent 2-simplices to the nucleus, since that gives us more choice for selecting the nerves and the overlaid region on the image. For example, let k be the order of the MNCs on a triangulated digital image, then the subsequent nerves' orders are: k-1, k-2, k-3... . We can refer the MNC as the first order nerve complex, k-1 nerve complex as the second order nerve complex and k-2 nerve complex as the third order nerve complex.

An example of various order nerve complexes and their overlap region can be seen below.

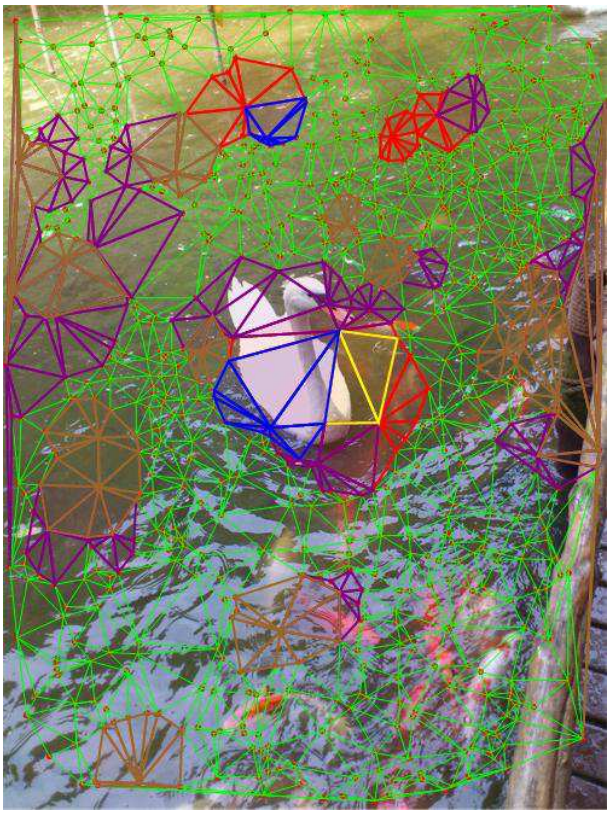


Figure 3.3: First, second and third order nerve complex and overlap regions

Note that red color is endowed for the MNCs, brown is for the 2^{nd} order nerve complexes and purple is for the 3^{rd} order nerve complexes. Their overlap 2-simplices are shown as blue and yellow color. The image object shape is completely covered and retrieved by the cluster of overlap 1^{st}, 2^{nd} and 3^{rd} nerve complexes.

In order to find the cluster of 1^{st}, 2^{nd} and 3^{rd} nerve complexes, we use the following

algorithm:

Algorithm 3.5: Finding MNCs overlap

Input : Digital image img.

Output: 1^{st}, 2^{nd} and 3^{rd} nerve complexes.

1 /* Detect edges on img*/ ;

2 $img \longmapsto EdgeDetected\, edg$;

3 /* Measure edge component and find hole centroids*/ ;

4 $edg \longmapsto HolesCentroid\, C$;

5 /* Use hole centroid as keypoint to triangulate image*/ ;

6 $C \longrightarrow DelaunayMesh\, M$;

7 Find MNCs $\longrightarrow$ Δ_{MNC};

8 Find MNC order k = number of triangle in any MNC;

9 Set k1=k-1;

10 Set k2=k-2;

11 Find all nerves have k1 order in $M \longrightarrow \Delta_{k1}$;

12 Find all nerves have k2 order in $M \longrightarrow \Delta_{k2}$;

13 Return Δ_{MNC}, Δ_{k1}, Δ_{k2};

Let Δ_1, Δ_2, Δ_3 be 1^{st} ,2^{nd} and 3^{rd} order nerve respectively. The object shape region contains a cluster of nerves that are strongly near together: $\Delta_1 \overset{\curlywedge}{\delta} \Delta_2 \overset{\curlywedge}{\delta} \Delta_3$.

Stages of the algorithm for finding cluster of overlapping nerves and detecting object shape are illustrated as below Figure 3.4:

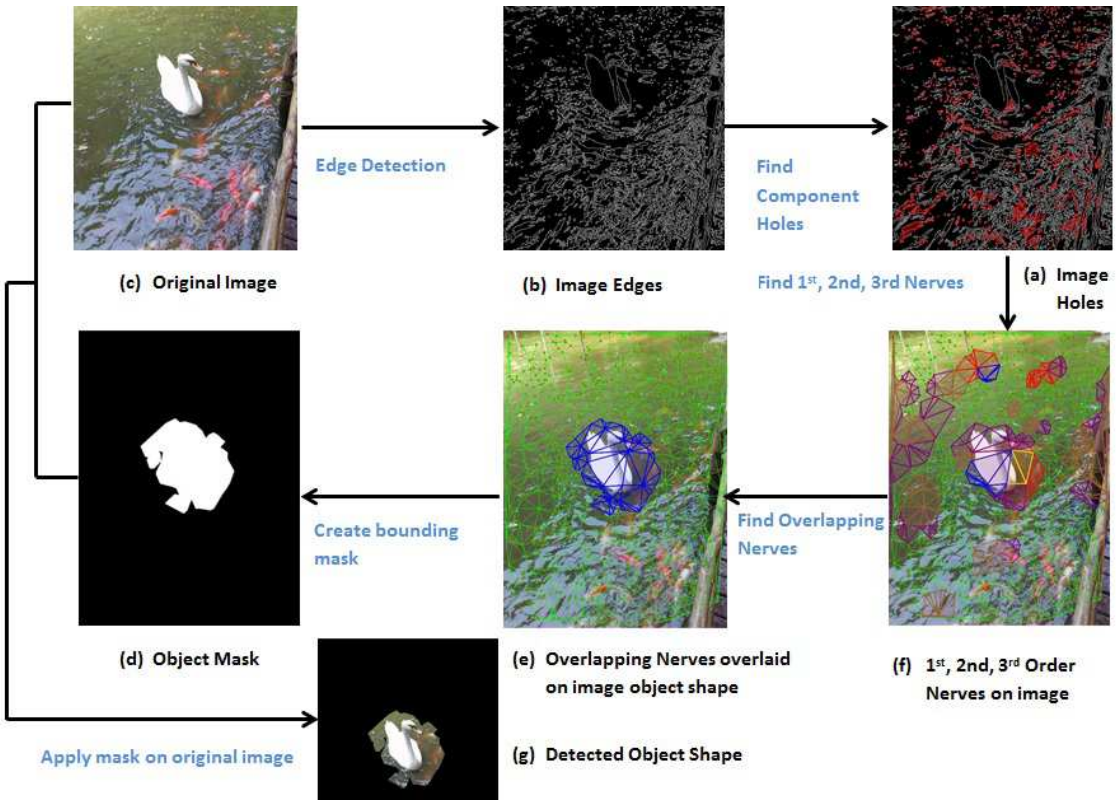


Figure 3.4: Detected Nerves covering the object shape

In the above example, we use edge detection[23] for image pre-processing and connected component centroid as a keypoint generator. We have limited the area of connected components (holes) measured by 10 and removing holes that reside on the boundary of image.

3.4 Maximal Nucleus Cluster for Object Shape Detection

Experimentally, MNCs tends to cover a large part of image object shape [101] as the object shape encompasses regions that have uniform descriptive interiors. In this section, a novel technique is proposed for using boundary and interior of MNC as a shape descriptor.

3.4.1 Proposed Method

First, we start with a 2D color image that is pre-processed for **image resizing**, **background removal** and **edge detection**. **Image resizing** helps enhance the computation speed of the algorithm. **Background removal** involves morphological transformation of the image and removal of small components and dominant color on the foreground. Unless the image background is smooth and uniform, background removal should completely remove the foreground pixels component and retain only the image object shape. This step is important as the subsequent tasks will rely heavily on the boundary of the object shape, and if any feature point that scatters away from boundary point of the object shape would reduce the accuracy of MNCs shape detection. Background removal helps remove unimportant feature points that are outside of the boundary shape.

Edge detection is a digital image processing technique for finding the boundaries of an object by detecting sharp change in image brightness, often with discontinuities. Common edge detection methods include: Sobel method, Canny method, Prewitt

method, Roberts method, and fuzzy logic method etc. According to Canny[23]: "The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information about object boundaries ". For Canny method, the following steps are introduced in[23].

1. Applying Gaussian filter to smooth the image and remove noise.

2. Finding the intensity gradient of the image by computing the first derivative in the horizontal and vertical direction, and then compute the edge gradient and direction.

3. Non-maximum suppression is applied with magnitude and direction of the gradient. It suppresses the gradient values and make edges look thinner by setting them to 0 for intensity that's not locally maximal.

4. Applying double threshold to filter out the edge pixels caused by noise and color variation. By setting the upper and lower threshold at a determinable value, edge pixel gradient value is selected according to the upper and lower threshold. Edge pixel value that's smaller than the lower threshold is suppressed.

5. Tracking edge with "hysteresis" and suppressing other edge that is weak or not connected to strong edge.

Canny's edge detection is used in this thesis because it filters out weak edge on the foreground and leave only strong edge on the boundary of the image object shape.

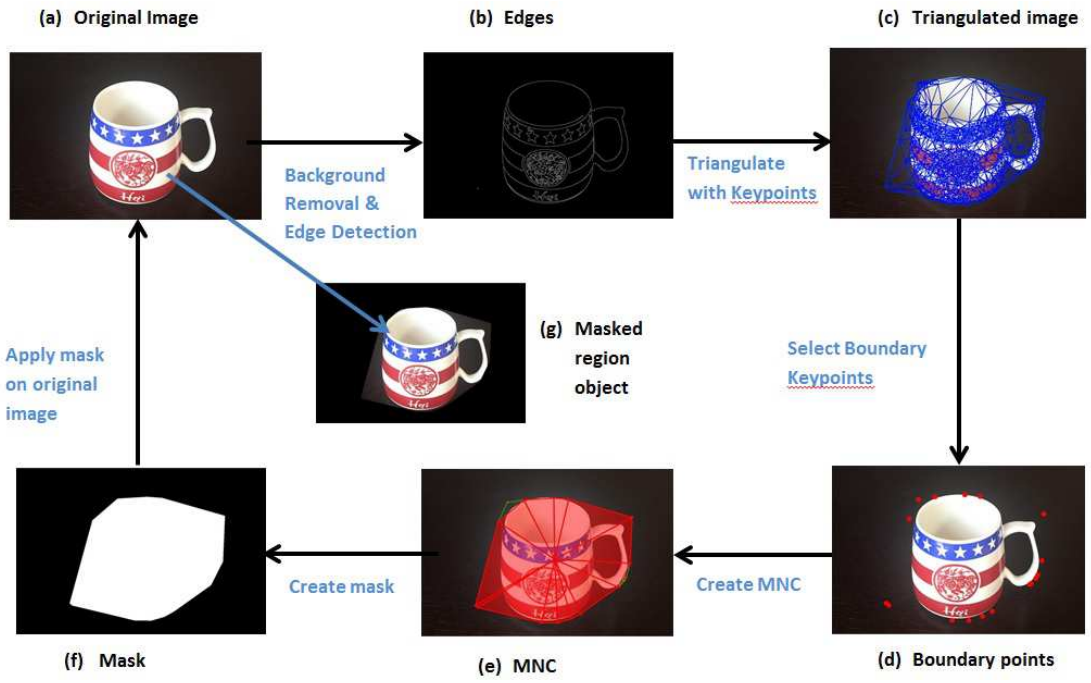


Figure 3.5: Stages of the proposed method

The next step in the proposed method is selecting a type of keypoint generator. As elaborated in section 2.4.3, there are four types of generating points.We will combine SURF[56] and centroidal points as generating points for this research. We would prefer using SURF[56] over SIFT because SURF has been integrated into *Mathematica* and it is as robust as SIFT but faster in computation time.

After detecting the keypoints with SURF, we only select the keypoint that stays close to or on the boundary edge. An algorithm similar to solving *traveling salesman problem* which was already implemented in *Mathematica* as a function, is used to connected the shortest contour of the image object shape. We then compute the centroidal point. The centroidal point is determined with equations 2.9 and 2.10.

The final step is to triangulate the image with the selected keypoints. As the pro-

posed algorithm only return a triangle mesh that cover the entire image object shape, then it's not necessary to run Algorithm 3.3 to find the MNC. And the final step is to construct a feature vector that is useful for classification task or object shape detection in video. The pseudo code for constructing the MNC shape descriptor is given in Algorithm 3.6 and Figure 3.5.

Algorithm 3.6: MNC descriptor

Input : Digital image img.

Output: MNC descriptor.

/*Morphological transformation of img, remove small component if necessary */

1 $img \underset{BackgroundRemoved}{\longmapsto} imgR$;

/*Apply edges detection on $imgR$ */

2 $imgR \underset{EdgeDetected}{\longmapsto} edg$;

/*Use SURF to generate keypoint */

3 $edg \underset{SURF}{\longmapsto} Keypoint\ K = \{k_1, k_2, ..., k_n\}$;

/*Use keypoint to triangulate the image img */

4 $img \underset{K=\{k_1,k_2,...,k_n\}}{\longmapsto} DelaunayMesh\ M = \bigcup \Delta_{1,2,..}$;

/*Transform $DelaunayMesh\ M$ into a convex hull */

5 $M \longmapsto ConvexHull\ CH$;

/*Find region centroid of$ConvexHull\ CH$ */

6 $ConvexHull\ CH \longrightarrow Centroid\ C = v_c$;

/*Find region boundary of $DelaunayMesh\ M$ */

7 $DelaunayMesh\ M \longrightarrow RegionBoundary\ RB$;

/*Transform $RegionBoundary\ RB$ to vertices */

8 $RB \longmapsto Vertices\ V = (v_1, v_2, ..., v_n)$;

9 $NewVertices\ newV = V \bigcup C = (v_c, v_1, v_2, ..., v_n)$;

/*Triangulate img with new set of keypoints $newV$ */

10 $img \underset{newV}{\longmapsto} MNC$;

3.4.2 MNC Classification

This subsection gives a brief introduction on the application of the proposed object detection method in descriptive distance-based classification.

A general approach to descriptive distance-based classification encompasses a construction of a set of descriptive features mapped by probe functions and similarity measurement based on distance metric.

Let C_A and C_B be the list of points coordinates of a given test object shape A and a query object shape B on a proximity space X: $C_A = (C_{A1}, C_{A2}, ..., C_{An})$ and $C_B = (C_{B1}, C_{B2}, ..., C_{Bn})$. Let $\Phi(A) = (\Phi(A)_1, \Phi(A)_2, ...\Phi(A)_n)$ and $\Phi(B) = (\Phi(B)_1, \Phi(B)_2, ...\Phi(B)_n)$ be the set of interior features of A and B mapped by probe functions $\Phi = (\Phi_1, \Phi_2, ..., \Phi_n)$. And, D and D_Φ are respectively the spatial and descriptive distance measure between these sets. We said that these two sets match the description and are near in space if they have the following two conditions for a given proximal tolerance ϵ and descriptive proximal toleranceϵ':

(**A.1**) B δ A $\Leftrightarrow$ $D(A, B) < \epsilon$.

(**A.2**) B δ_Φ A $\Leftrightarrow$ $D_\Phi(A, B) < \epsilon'$.

As introduced in section 2.6.1, we can use Euclidean and Hausdorff distance metric for measuring the spatial relation and $\breve{C}$ech distance as a descriptive distance measure.

After the object shape is detected with the MNC, a feature vector can be con-

structed for the interior region of the MNC. A list of features can be considered for contributing to form a feature vector:

- Number of interior triangles.
- Total area covered by MNC.
- Location of the nucleus.
- Average pixel intensity.
- Diameter of the MNC.
- Image histogram.
- Gradient Orientation.
- Moments.

Another straightforward method that can be used to match description of two shape regions is template matching[21].The pseudo code for the classification of MNC is given in Algorithm 3.7 using Hausdorff distance as spatial proximity measure and $\check{C}$ech distance as descriptive proximity measure.

Algorithm 3.7: MNC Classification

Input : Test MNC A, query MNC B, ϵ and ϵ'.

Output: Classifying B.

`/*Computing Hausdorff distance between A and B */`

$D_h = \varnothing$;

foreach $c_{Ai} \in C_A$ **do**

  $d_{AB} = \varnothing$;

  foreach $c_{Bi} \in C_B$ **do**

    $d_{AB} := \min(d_{AB} \bigcup d(c_{Ai}, c_{Bi})$;

    `/*`$d(c_{Ai}, c_{Bi})$ `is the Manhattan Distance */`

  $D_h := \max(D_h \bigcup d_{AB})$;

`/*Computing` $\breve{C}$`ech distance between A and B */`

$D_c = \varnothing$;

foreach $\Phi(A)_i \in \Phi(A)$ **do**

  foreach j *in* Φ_i **do**

    $d_{\Phi j} = \text{sum}(|\Phi_{ij}(A) - \Phi_{ij}(B)|)$;

  $D_c = \min(D_c \bigcup d_{\Phi j})$;

`/*Checking conditions A.1 and A.2 */`

if $D_h < \epsilon$ *and* $D_c < \epsilon'$ **then**

  B is the same class as A ;

else

  B is not the same class as A;

3.5 Results, Analysis and Conclusion

3.5.1 Caltech 101 Dataset

A set of experiments are carried out on the Caltech 101, a picture dataset of objects with 101 classes. Each class has 40 to 800 images. Most classes have around 50 images[69]. And the size of a typical image in dataset is around 300 x 200. Sample images in the dataset are shown in Fig. 3.6.

Figure 3.6: Caltech 101 dataset sample images

3.5.2 Experiment results

The proposed algorithm was implemented in Mathematica and was run on a 2.4GHz Intel Core i5 processor with 64-bit Window OS version 8. It takes average 0.094 second to compute the algorithm. Thus the proposed method is suitable for real-time image object shape detection. Some of the images in the dataset was tested with the proposed algorithm and the results are below:

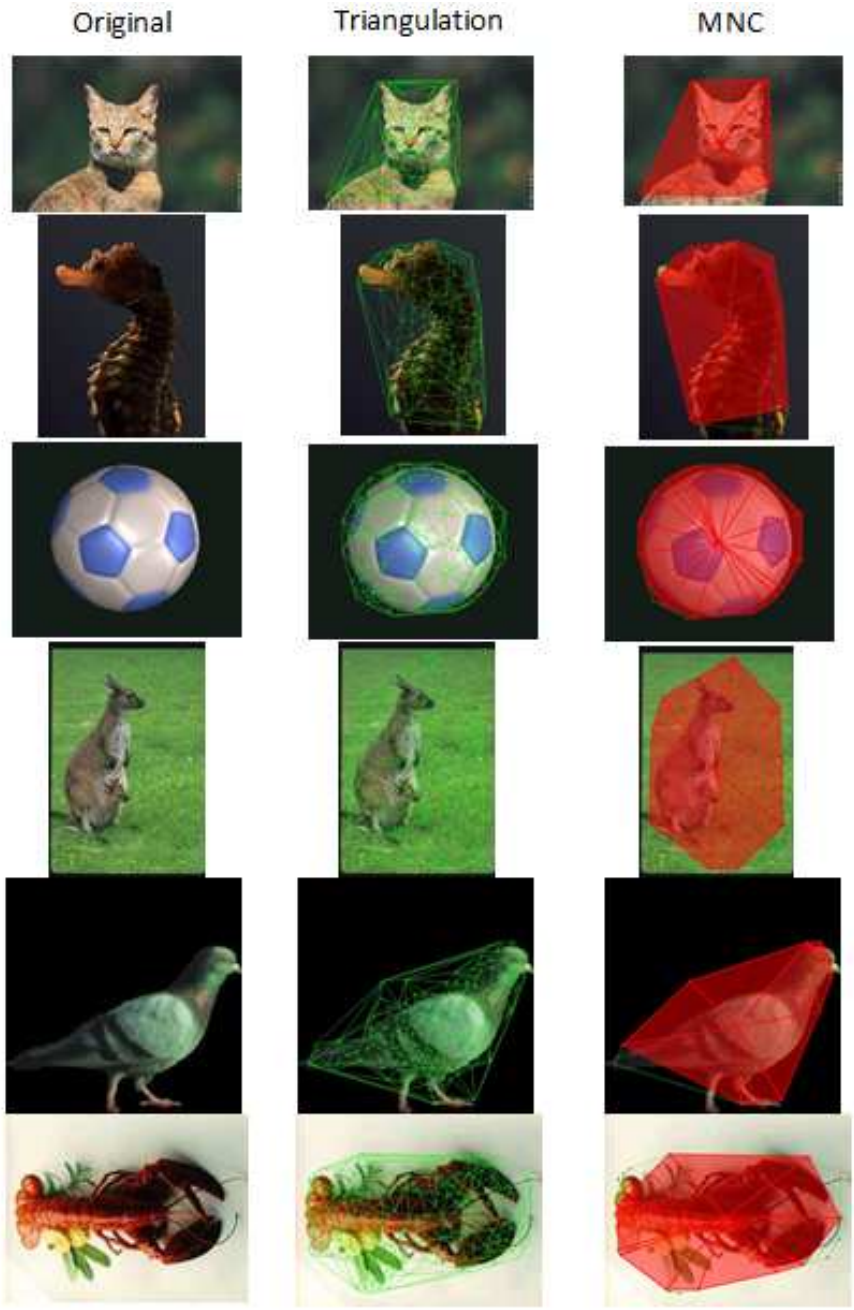


Figure 3.7: Some results obtained by running proposed algorithm

By observing Figure 3.7, the proposed method completely captures the entire object shape in the image foreground. For most of the images in the dataset, MNC

algorithm always captures the image object shape. However, in some image where there are multiple objects in the background, the MNC region tends to cover a large area of the foreground as in Figure 3.8. As there are still some connected components on the foreground after the image-preprocessing steps, keypoints are generally generated on these connected components.

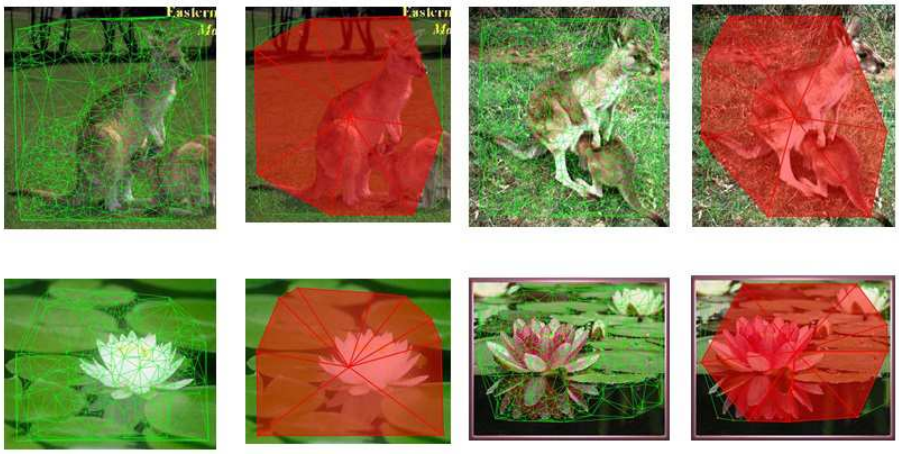
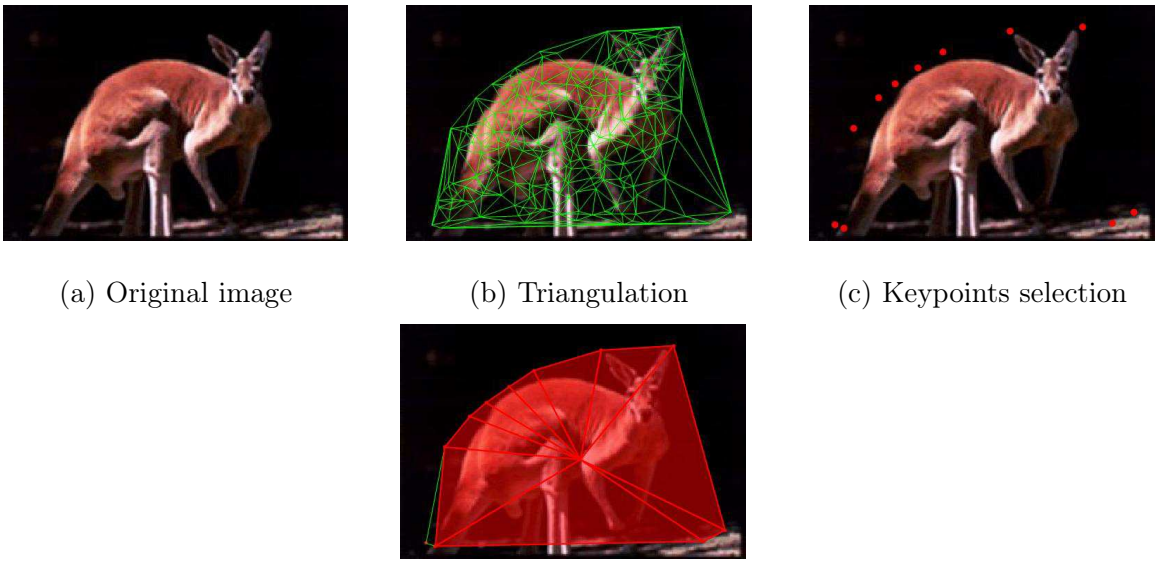


Figure 3.8: Some unsatisfactory results

Each image region with an MNC overlay is characterized by the features of the covering MNC. Thus, geometrical and descriptive properties MNC can be exploited to represent shape as a geometrical MNC shape descriptor. For an example of a MNC superimposing on a kangaroo image below, an MNC descriptor can be created for the tabulated features and descriptive features.



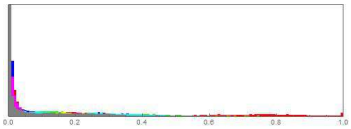
(a) Original image (b) Triangulation (c) Keypoints selection

(d) MNC on image

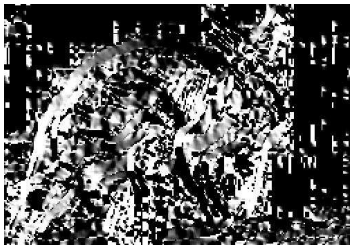
Figure 3.9: MNC detection on a digital image

Features	MNC
Number of triangles	10
Area	33731.9
Average pixel intensity	0.289237, 0.168142, 0.157479
Centroid location	155.957, 86.6923
Maximal radius	152.14

Table 3.1: MNC descriptor



(a) Histogram of MNC



(b) Gradient Orientation of MNC

Figure 3.10: Descriptive Features of MNC

3.5.3 Conclusion

In this chapter, the mathematical background on descriptive proximity is presented to provide a framework for object shape detection based on strongly descriptive nerve. A novel method also has been proposed for fast object shape detection, which is suitable for implementation in real-time video processing. This results in an MNC covering object shape pose prominent features to represent shape as descriptor. Many applications rely on detecting and capturing the region of interest for higher level of computation such as object recognition. Thus this proposed algorithm is applicable to such a context.

In order to perform higher level of object recognition, we can apply basic distance based classification to compare object similarity or more advance machine learning models, which has been mentioned previously: neural network, SVM or manifold

learning etc. The mathematical background and the proposed algorithm in this chapter will contribute toward the application in video processing in the next chapter.

Chapter 4

Nerve Persistence

4.0.1 Introduction

In this chapter, experiments of nerve persistence are introduced in video frames to explore the persistence properties of the detected nerves for the proposed method. A video is a sequence of image frames and a single object in a video from different angles may have persistent features in each image frame of the sequence.

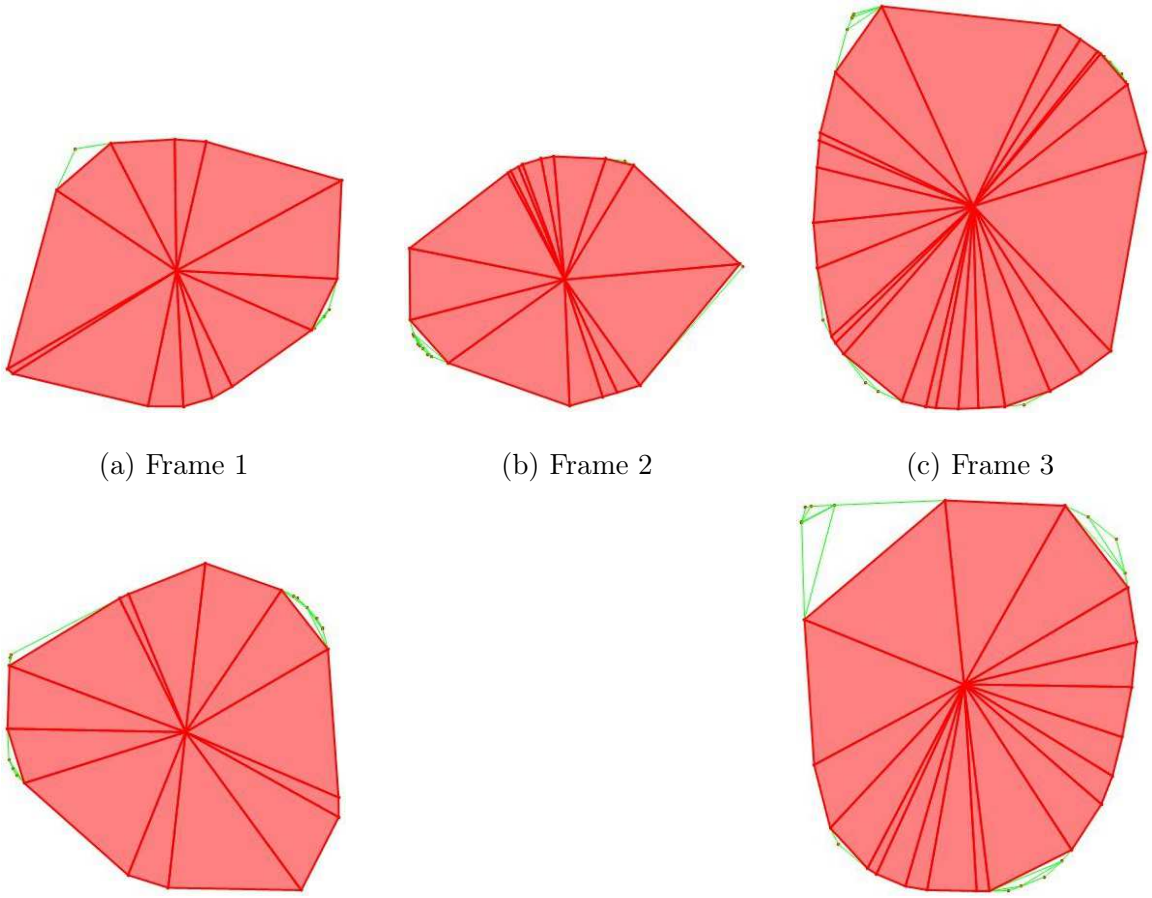
4.0.2 Persistence Homology of the Proposed Nerve Detection

In the proposed algorithm in Chapter 3, it is interesting to explore the persistence homology of the nerve shape throughout the image sequences (video) of a single object. For each frame, stability measurements are calculated to determine the persistence of the nerve shape detected by the proposed algorithm. These measurements include the calculation of the persistence diagrams of the feature points homology and the stability measurements with bottleneck distance [37, p. 216] and Wasserstein distance[37, p. 216], and additionally, other proposed persistent measurements. The persistence

diagram (PD) is constructed with the r-ball model filtration and the Vietoris-Rips Complex filtration [37, p. 74]. And, other persistent measurements used are Earth's mover distance, Hausdorff distance and nerve area difference. We consider to experiment with a sequence of image frames as in Figure 4.1, which contains a single object from different angles and scales.

Figure 4.1: Sample sequence of image frames

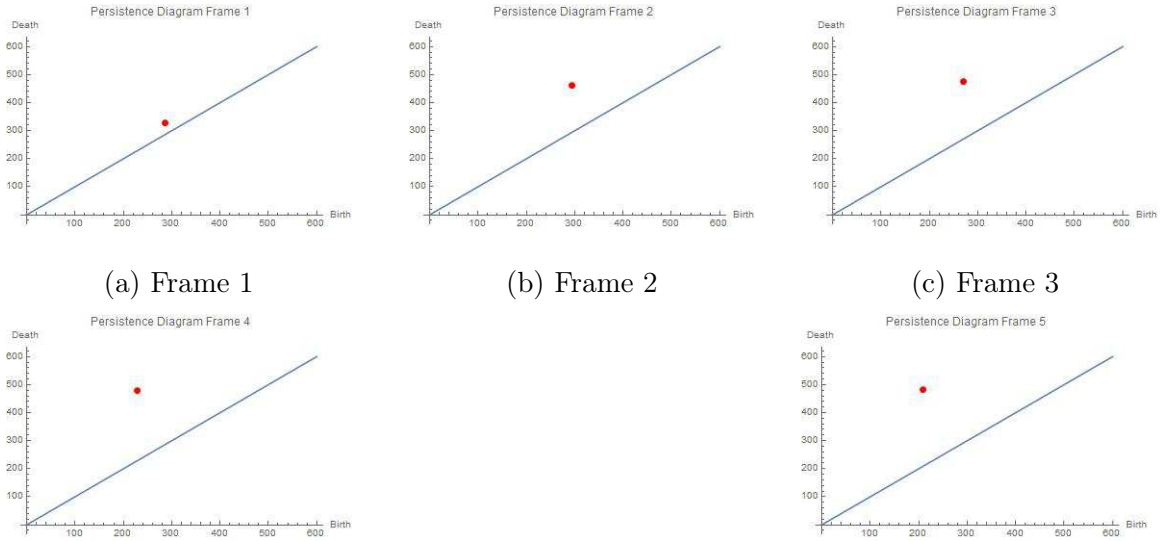
On each image frame, algorithm 3.6 is applied to detect the nerve. Then for each boundary vertice of the nerve, the filtration is performed in order to record the birth and death time of the persistent diagrams of the nerve boundary vertices. The persistence homology of the nerve detection algorithm is characterized by the persistence diagrams. These graphs below show the obtained persistence diagrams by during the r-ball model filtration. The r-ball filtration algorithm was implemented manually in Mathematica.



(a) Frame 1 (b) Frame 2 (c) Frame 3

(d) Frame 4 (e) Frame 5

Figure 4.2: Detected Nerves on Image Frames

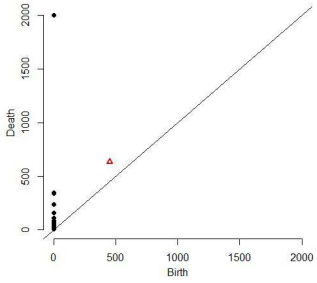


(a) Frame 1 (b) Frame 2 (c) Frame 3

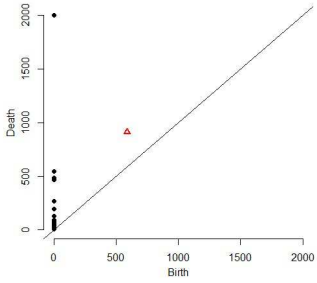
(d) Frame 4 (e) Frame 5

Figure 4.3: Persistence diagrams with r-ball model filtration

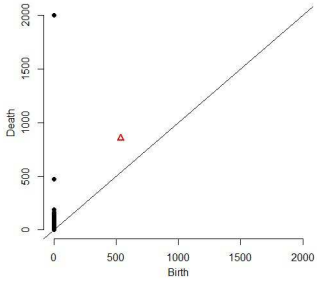
The Vietoris-Rips Complex filtration was performed with R using TDA package[39]. And, the persistence diagrams retrieved by Vietoris-Rips Complex filtration are shown below:



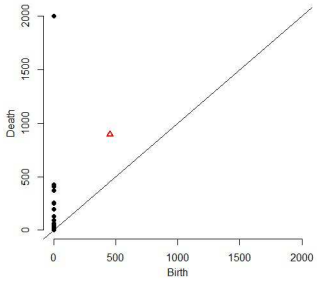
(a) Frame 1



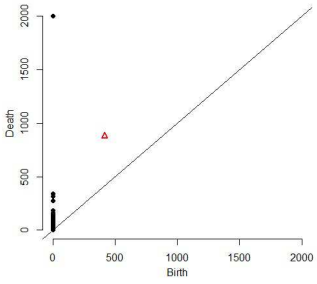
(b) Frame 2



(c) Frame 3



(d) Frame 4



(e) Frame 5

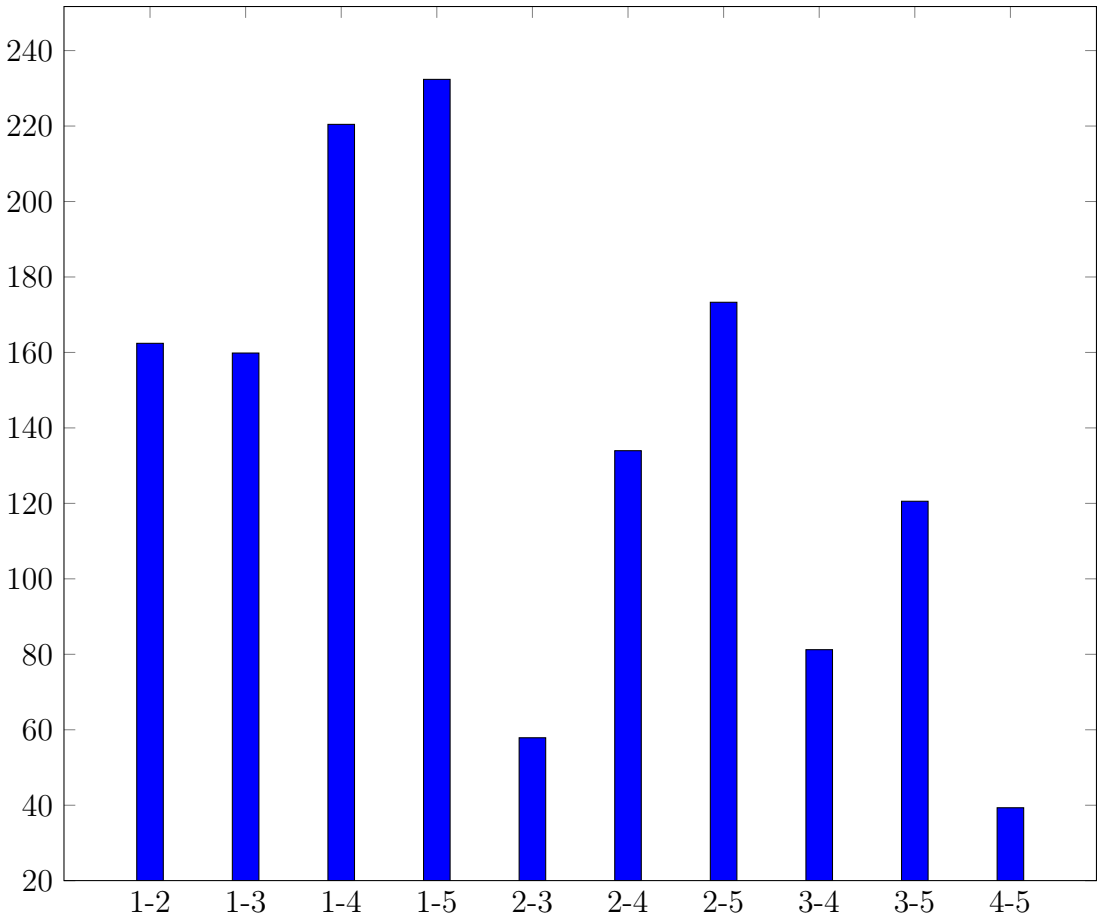
Figure 4.4: Persistence diagrams with Vietoris-Rips complex filtration

Next, the stability measurements which include Bottleneck Distance and Wasserstein Distance are carried out between two image frames. The results are shown below:

Bottleneck Distance						
	PD1	PD2	PD3	PD4	PD5	Sum
PD1		162.4175	159.8432	220.4462	232.3694	775.0763
PD2	162.4175		57.88336	133.9621	173.2954	527.5584
PD3	159.8432	57.88336		81.22741	120.5607	419.969
PD4	220.4462	133.9621	81.22741		39.33332	474.969
PD5	232.3694	173.2954	120.5607	39.33332		565.5588

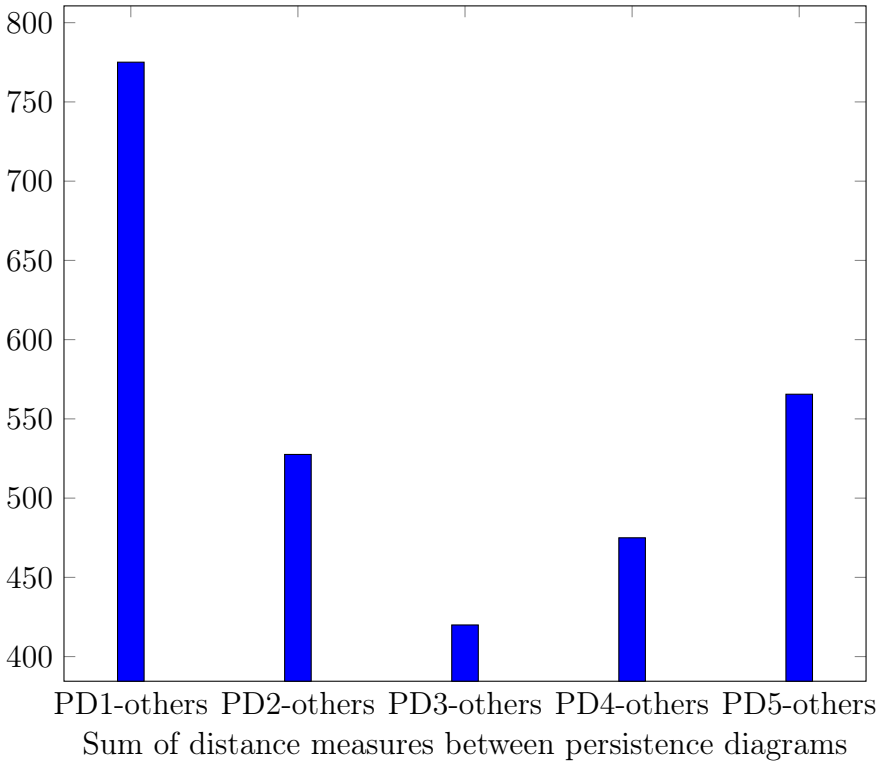
Table 4.1: Bottleneck distance measures between persistence diagrams

In the table, each numerical value is the measurement between two persistence diagrams using bottleneck distance. And each persistence diagram is obtained from each image frame. For instance, measurement between persistence diagram 1 (PD1) and persistence diagram 2(PD2) is 162.4175. The last column on the right contains the sum value from one persistence diagram to the rest of the other persistence diagrams. By calculating this way, we can know which persistence diagram has the most and least persistence measurements. In this bottleneck distance measure, persistence diagram 1 (PD1) has the biggest persistence measurements and persistence diagram 3 (PD3) has the smallest persistence measurements. And, distance measure from PD1 to PD2 is equal to distance measure from PD2 to PD1.



Bottleneck distance measures between persistence diagrams on each frame

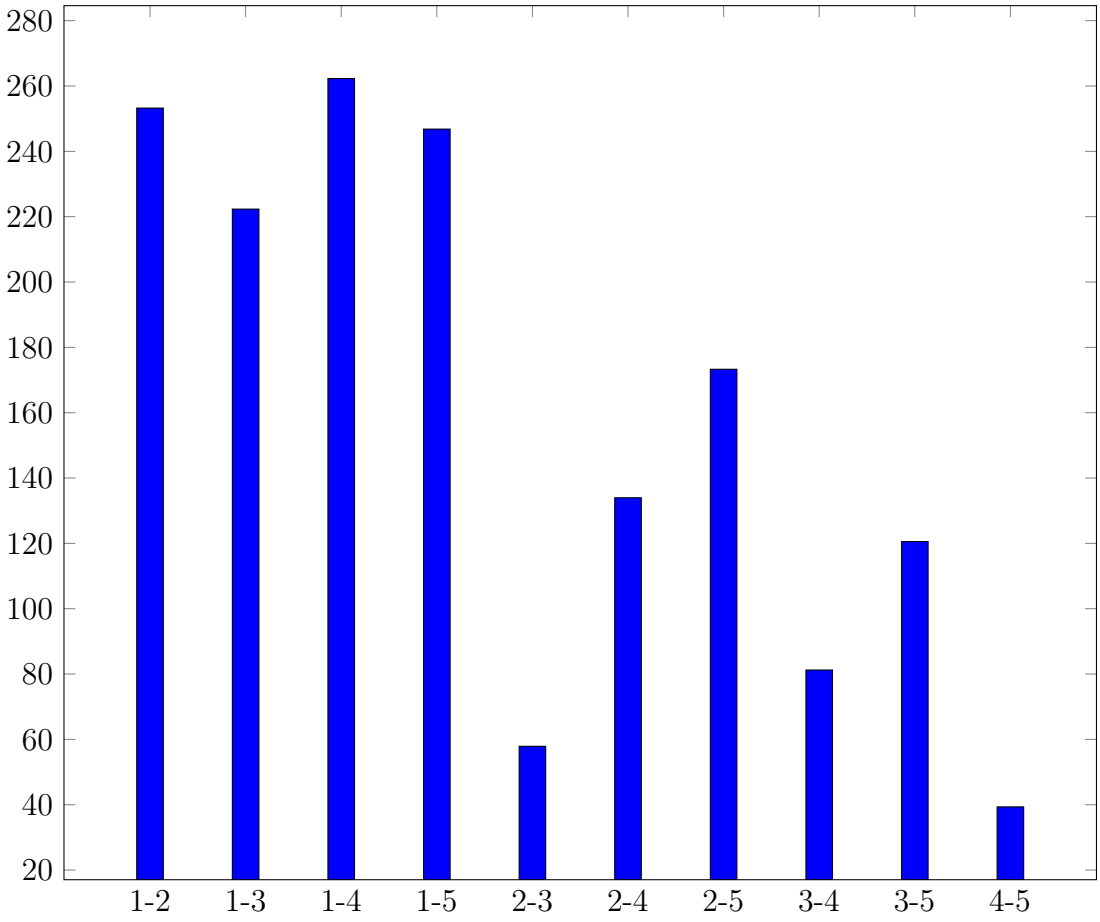
In the bar graph, 1-2 denotes the measurement between persistence diagram 1 and persistence diagram 2. Bottleneck distance measure between persistence diagram 1 and persistence diagram 5 (or 1-5) has the largest value while 4-5 has smallest value. Then persistence diagram 4 and 5 have similar persistent measure.



The sum of distance measure bar graph shows the sum value for distance measure from one persistence diagram to others.

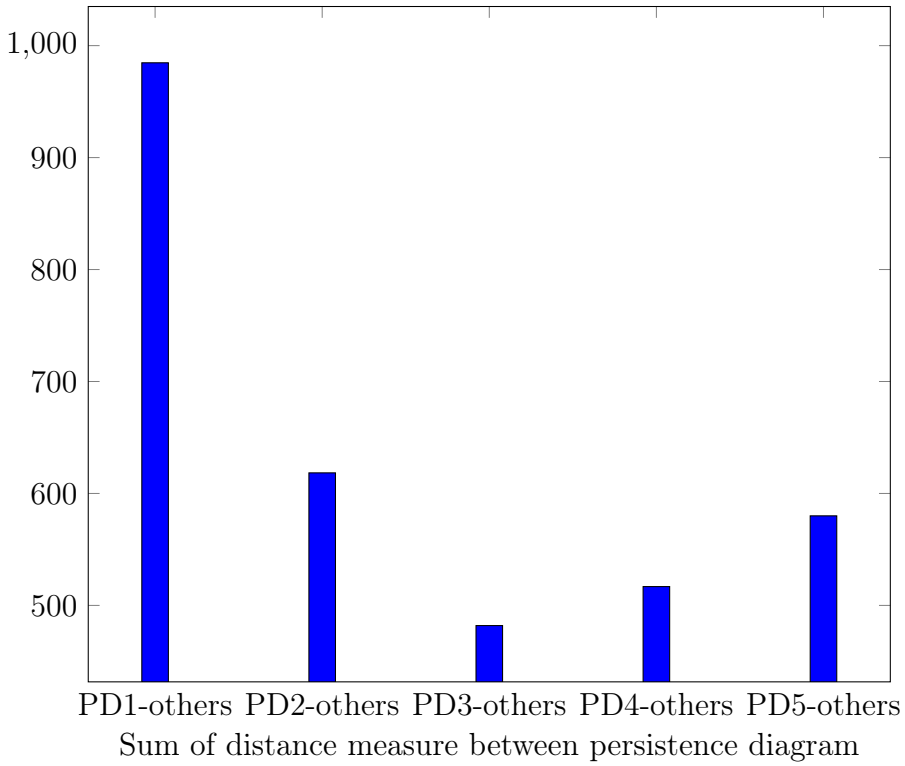
Wasserstein Distance						
	PD1	PD2	PD3	PD4	PD5	Sum
PD1		253.2464	222.3174	262.2961	246.8092	984.6691
PD2	253.2464		57.88336	133.9621	173.2954	618.3873
PD3	222.3174	57.88336		81.22741	120.5607	481.9889
PD4	262.2961	133.9621	81.22741		39.33332	516.8189
PD5	246.8092	173.2954	120.5607	39.33332		579.9986

Table 4.2: Wasserstein distance measures between persistence diagrams



Wasserstein distance measures between persistence diagram of each frame

Wasserstein distance measures obtain much similar measurements as bottleneck distance. Persistence diagram 1 and 3 still have the highest and lowest distance measures.



For both bottleneck and Wasserstein distance measure results, stability distance measure of persistence diagram 1 and the rest achieves highest sum value. And stability distance measure between persistence diagram 3 and the others stays at lowest value. This means the detected nerve in image frame 1 and 3 have biggest and smallest variation respectively in term of persistence homology and stability measurement.

In addition, we consider proposing other measurements for nerve persistence using: Hausdorff distance, earth mover's distance and area differences. Hausdorff distance is calculated between nerve vertices from one image frame to another using Hausdorff distance equation 2.18. The histograms of the interior region of two nerves are compared with earth mover's distance, which is equivalent to Wasserstein distance

between the two distributions. And, the change in area covered by the nerve in each frame provides a simple measure for nerve persistence. The algorithm for proposed method is shown below.

Algorithm 4.1: Proposed Persistence Measure

Input : Detected Nerves $N1$ and $N2$.

Output: Persistence distance measure.

`/*Get the boundary vertices of the nerves */`

1 $N1 \longmapsto V1_i$;

2 $N2 \longmapsto V2_i$;

`/*Computing earth's mover distance between histogram of each nerve */`

3 $D_E = d_e(H(N1), H(N2))$;

`/*Computing Hausdorff distance */`

4 **foreach** $v_i \in V1$ *and* $v_j \in V2$ **do**

5 Compute minimal Euclidean distance $D_E = min(d_E(v_i, v_j) \; \forall \; v_j \in V2$;

6 Return $D_H = max(D_E) \; \forall \; v_i \in$ V1;

`/*Computing absolute area difference */`

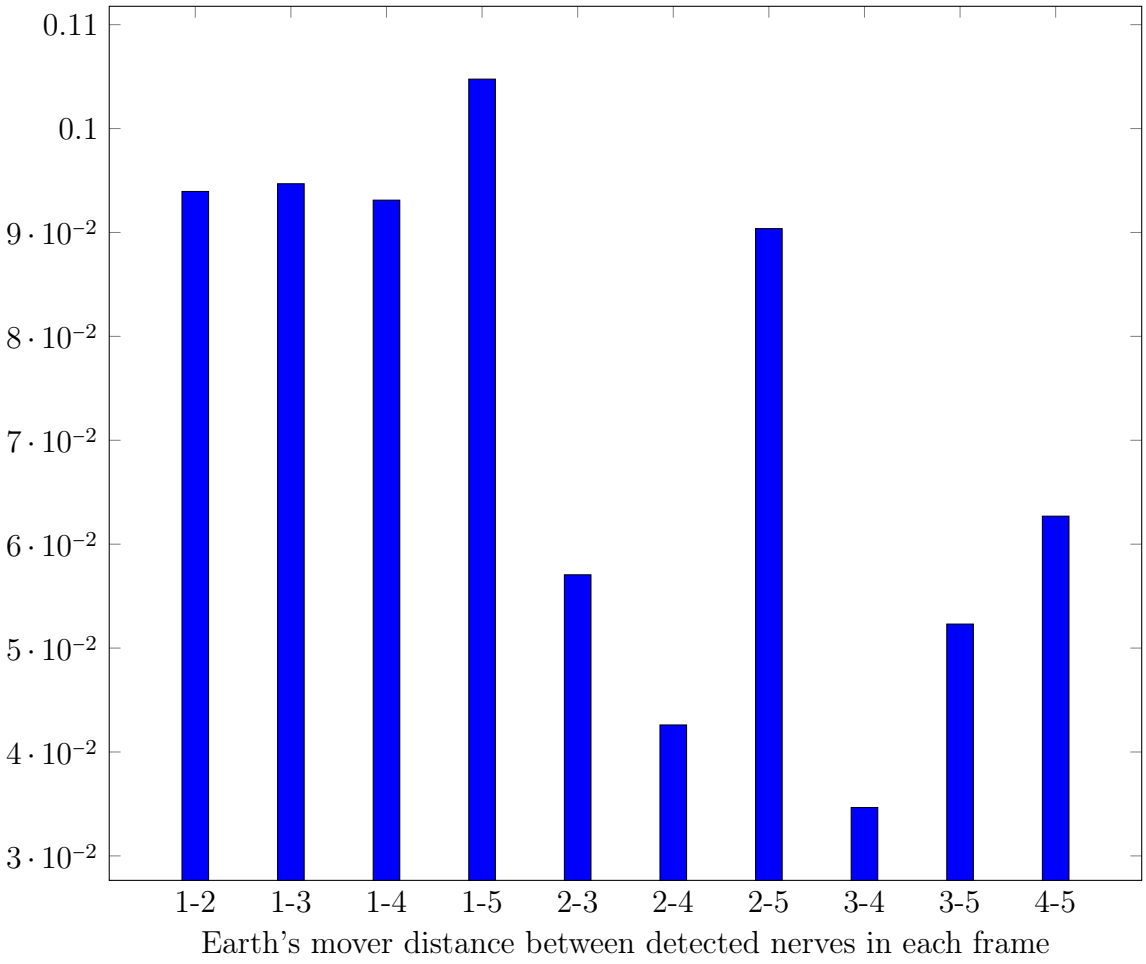
7 $D_A = |A(N1) - A(N2)|$;

The algorithm 4.1 was implemented in Mathematica and the results are presented as followings.

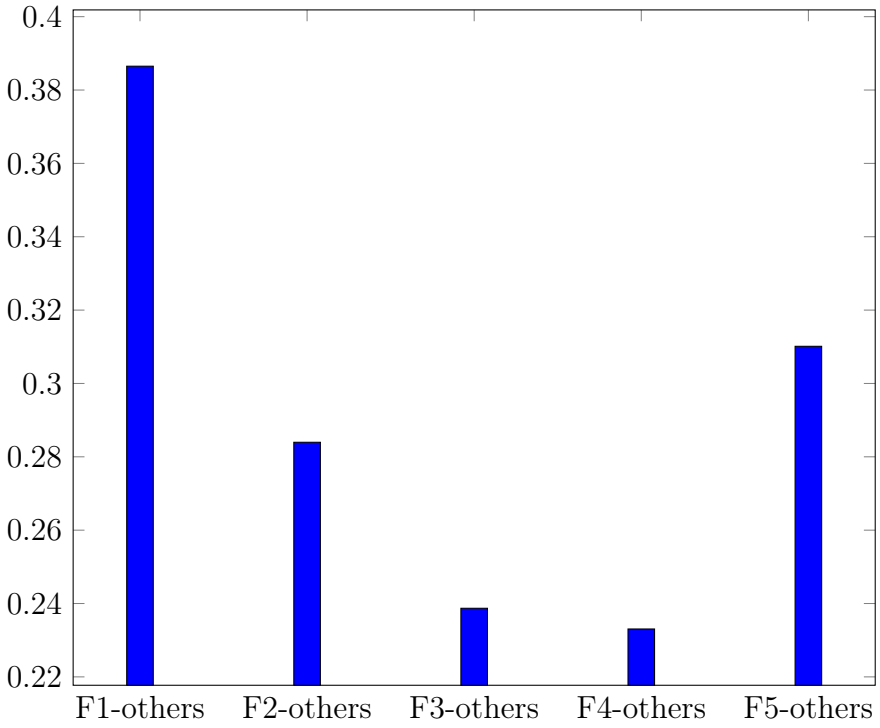
Earth's Mover Distance						
	F1	F2	F3	F4	F5	Sum
F1		0.0939444	0.0946875	0.0931088	0.104757	0.386498
F2	0.0939444		0.0570528	0.0426	0.0903689	0.283966
F3	0.0946875	0.0570528		0.0346552	0.0523104	0.238706
F4	0.0931088	0.0426	0.0346553		0.0626944	0.233059
F5	0.104757	0.0903688	0.0523104	0.0626944		0.310131

Table 4.3: Earth's mover distance between detected nerves in each frame

The tabulated results show the distance measure between nerve histogram of each image frame using Earth's mover method. The rightmost column contains the overall sum value of one frame to all others. Frame 1 contains nerve that has largest distance measure comparing to others and Frame 4 has the least measure. This implies frame 1 is most different in terms of descriptive measure and frame 4 is the most similar frame to the rest.



These bar graphs show how value of distance measure of one frame relates to others. Earth's mover distance measure between frame 1 and 5 is at highest and measure between frame 3 and 4 is lowest. Frame 3 and 4 have similar nerve description.

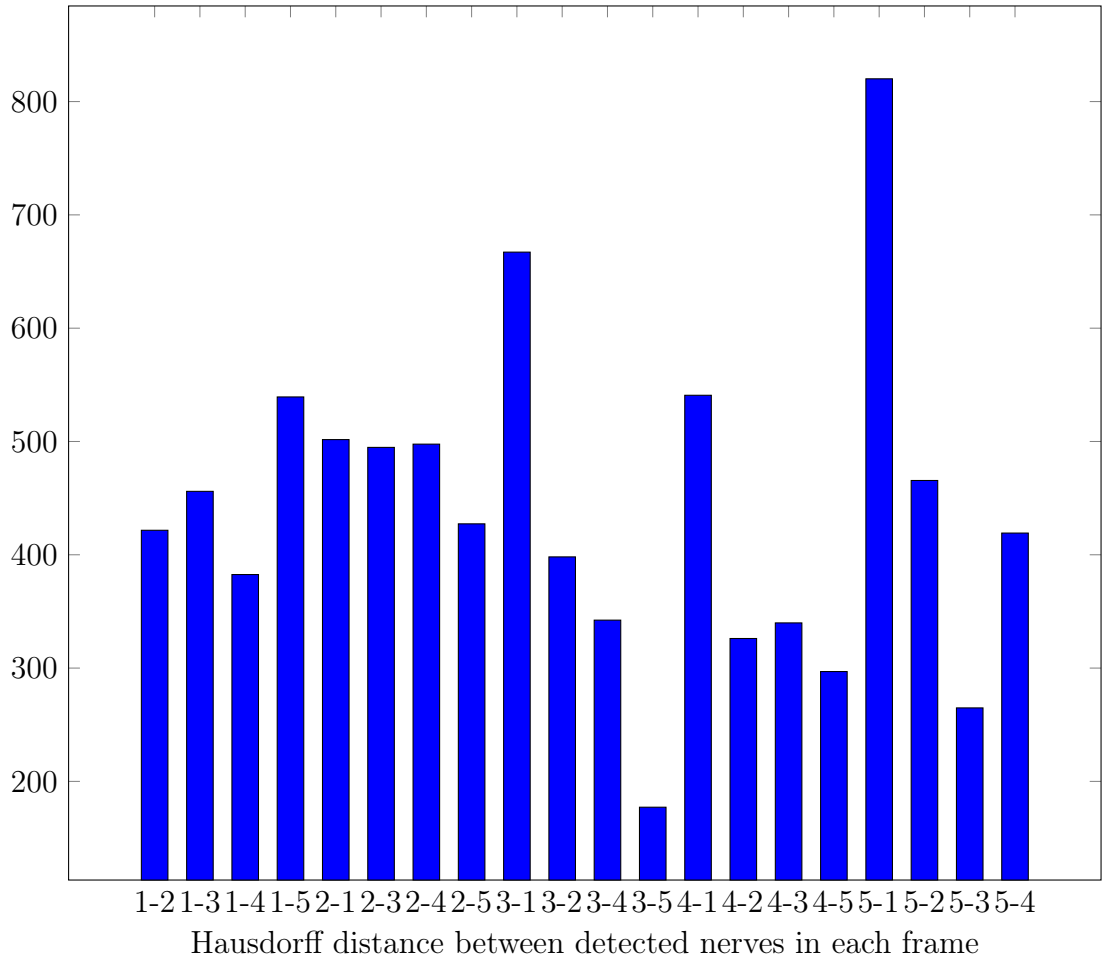


Hausdorff Distance						
	F1	F2	F3	F4	F5	Sum
F1		421.634	455.994	382.555	539.314	1799.497
F2	501.673		494.782	497.661	427.308	1921.424
F3	667.132	398.141		342.316	177.217	1584.806
F4	540.791	326.107	339.88		296.949	1503.727
F5	820.095	465.577	264.856	419.174		1969.702

Table 4.4: Hausdorff distance between detected nerves in each frame

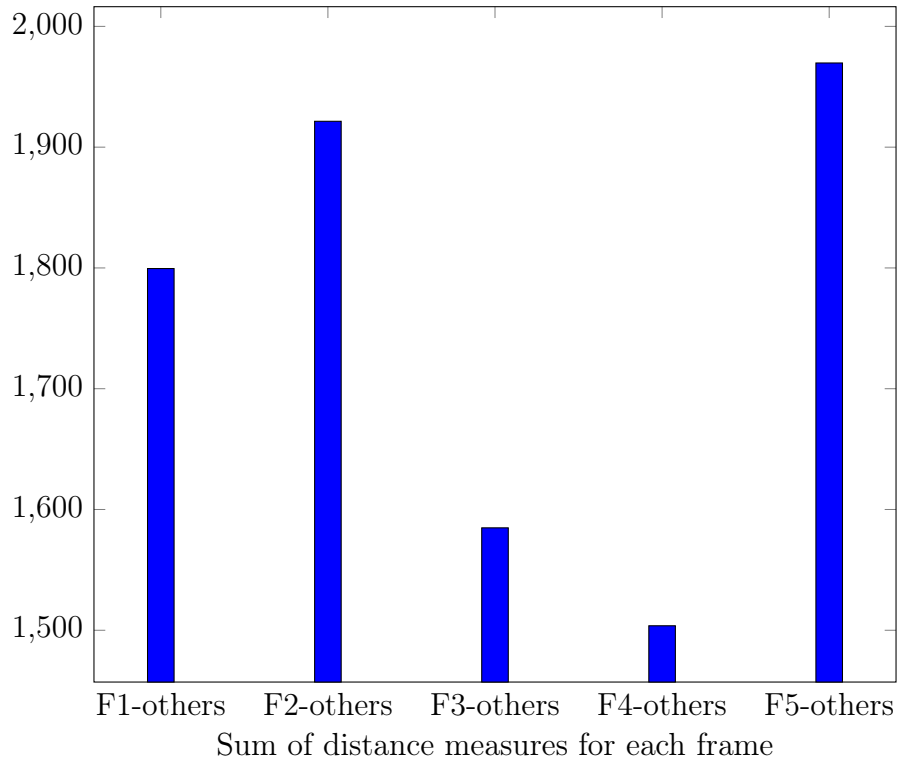
The tabulated distance measures for Hausdorff distance presents the computed

values between nerves in each frame. They show how nerves are relatively close to each other in image space. Distance measure between nerve in frame 5 and the rest is at biggest and for frame 4 is at lowest. This implies nerve in frame 4 is nearest to the other nerves in the image space. Due to the nature of the Hausdorff equation, the Hausdorff distance from nerve A to nerve B is not equal to the Hausdorff distance from nerve B to A. Then for each distance measure from one nerve to another, we have a unique value.



The bar graphs present Hausdoff distance between nerves in image frames. Clearly,

each distance measurement is different to others. Measurement from nerve 3 to 5 has the lowest value. Then, nerve 3 and 5 are close in image space.

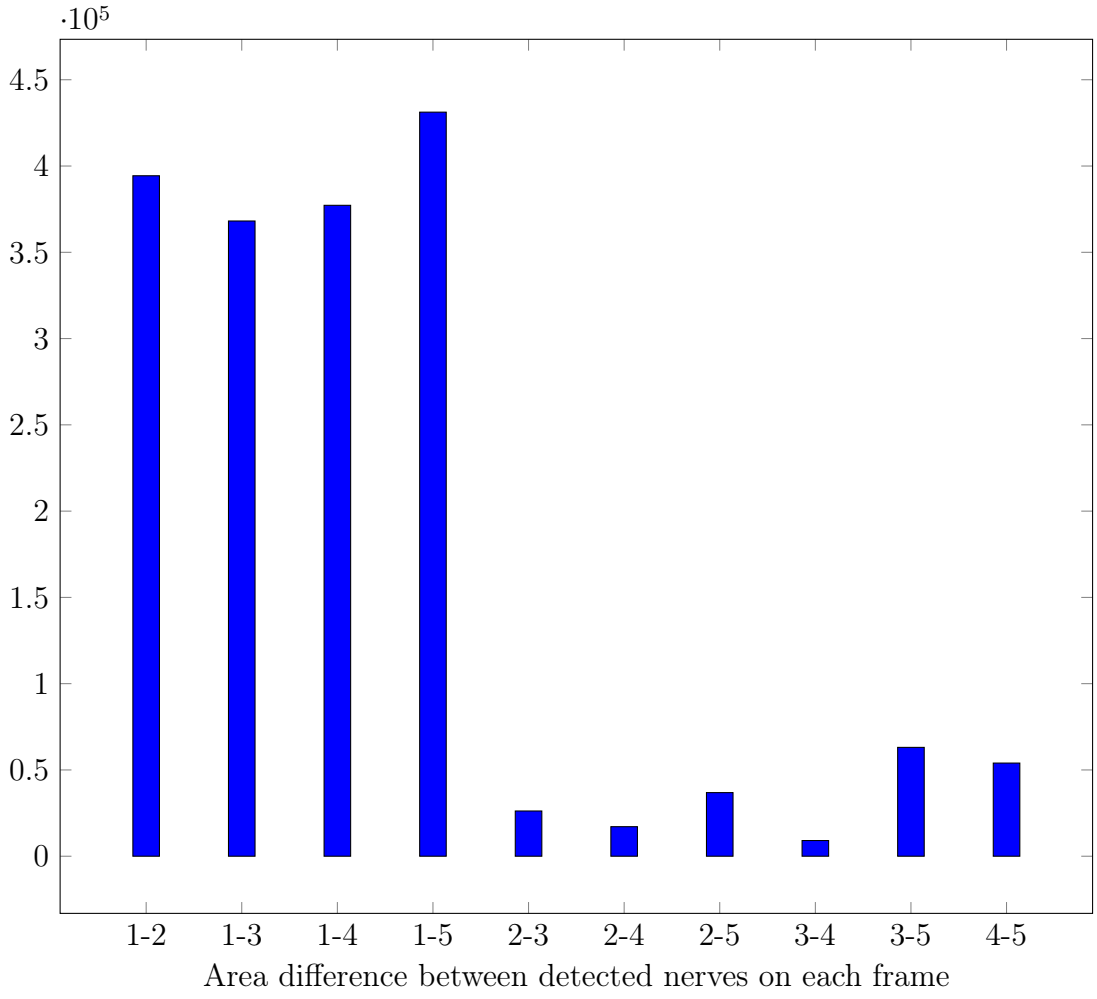


Area difference between nerves provide a simple comparison for the size of nerve. The area difference is computed by taking absolute subtraction of areas occupied by nerves in each of two frames.

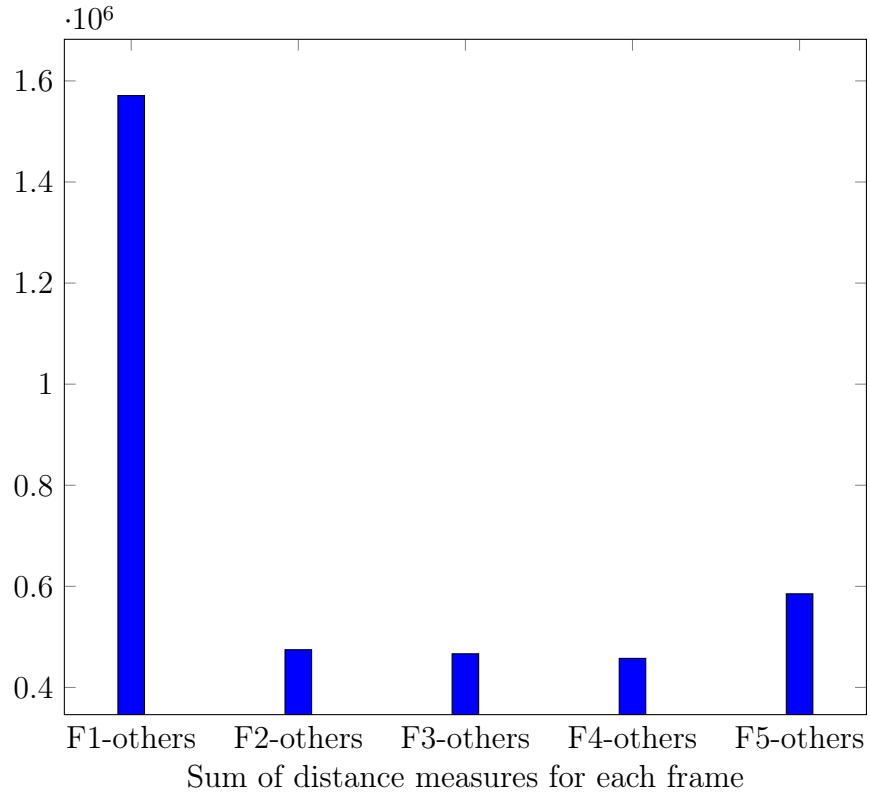
Area Difference						
	F1	F2	F3	F4	F5	Sum
F1		394361.45	368145	377237	431227	1570970
F2	394361.45		26216.3	17124.3	36865.6	474567.7
F3	368145	26216.3		9092.08	63082	466535.4
F4	377237	17124.3	9092.08		53989.9	457443.3
F5	431227	36865.6	63082	53989.9		585164.5

Table 4.5: Area difference between detected nerves on each frame

The tabulated results show the symmetric values that differentiate from one nerve area to another. The sum of all area differences from one nerve to the others is presented on the rightmost column. The nerve detected in frame 1 has largest overall area difference and the nerve in frame 5 has the smallest.



On bar graph, frame 3 and 4 have minimal difference in term of area change while frame 1 and 5 have biggest change in area. Comparing detected nerve by area change is subject to scale and rotation of the object shape in image frame. Frame that has minimal transformation will surely give minimal area change.



By summing the area change from 1 nerve to the rest of the other nerves, we can have a clear view on how detected nerve in a single frame is affected by image transformation and rotation and which image frame has the least and greatest variation of area change throughout the sequence of image frames.

4.0.3 Key Frame and Maximum Scene Change

Based on the previous experiments, we can use these results to detect key frame and maximum scene change in video. We define the key frame the frame that has the minimum average distance of persistence measure and the maximum scene change is the image frame that has the maximum average value of the persistence distance measure. For that purpose, the sum of the persistence distance measurements needs

to be normalized. We propose using mix max scaling to bring each distance measure of persistence to having equal weight in order to determine the average. According to S. Aksoy and R. Haralick, min-max scaling is defined as

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{4.1}$$

As our experiment results have different scales. It's necessary to apply min-max scaling to normalize the distance values to scale range of [0,1]. Then we can extract the mean values for determining the key frame and scene change.

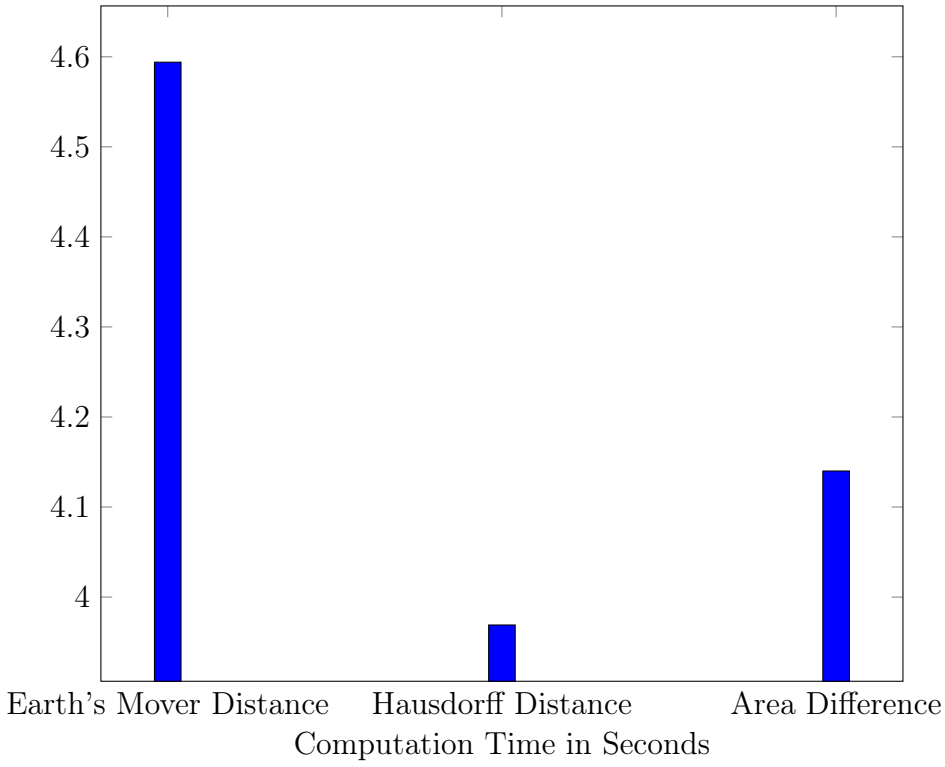
	Bottleneck	Wasserstein	Earth's	Hausdorff	Area	Mean
F1-	1	1	1	1	0.634734	0.926947
F2-	0.303868	0.271342	0.331777	0.015378	0.896394	0.363752
F3-	0	0	0.036805	0.008165	0.173999	0.043794
F4-	0.155963	0.069289	0	0	0	0.04505
F5-	0.410742	0.194974	0.502297	0.1147	1	0.444543

Table 4.6: Mean value of persistence measures

Where F1-, F2-, F3-, F4-, F5- denote the normalized sum of persistence distance measure from frame 1, 2, 3, 4, 5 to the other frames. We can suggest that the key frame is the image frame that has the minimal mean value, which is 0.043794 for persistence measure of image frame 3. The biggest scene change based on the persistence measure is image frame 1 with mean value of 0.926947.

4.0.4 Application in Real Time Video Processing

The running time and complexity of the computation increases with the number of video frame we use. For instance, if the running time for either earth's mover distance, area difference and Hausdorff distance algorithm is τ, then the running time for applying either earth's mover distance, area difference algorithm in a video with n frames is $\frac{2}{n(n-1)}\tau$. And for Hausdorff distance algorithm, the running time is $n(n-1)\tau$. In order to reduce the running time in real time video, we can compare nerve persistence in two consecutive frames and return the distance measure results. The comparison of the running time for each of the proposed algorithm is below.



In real time video processing, the persistence of the nerves in two consecutive image frames are measured and recorded. Then scene change and key frame can be determined by the recorded measurements in a similar fashion as previously stated.

4.0.5 Conclusion

This chapter serves to explore the persistence homology of previously proposed algorithm for detecting nerve on image frame. Therefore, a novel approach for detecting scene change and key frame has been proposed based on the persistence measure of the detected nerve. Both persistence measure based on descriptive measure (earth's mover distance of histogram distribution) and spatial measure (Hausdorff distance and area difference) has been experimented and contribute relatively as factors for determining key frame and scene change. The least persistence change of nerve in key frame happens when the nerve has the least homology, descriptive and spatial change on the image space. The ultimate purpose behind is to explore which detected feature points of the detected nerve are robust in image object shape rotation, transformation and are best representations of shape over a sequence of image frames.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis reviews image shape descriptor and introduced a novel approach on constructing image object shape detection based on computational proximity and geometrical nerve simplexes. This geometrical approach leads to the application in capturing the region of interest and detecting object shape in digital image and video frames.

In Chapter 3, computational proximity was further elaborated, and an algorithm for its application in object shape detection using Edelsbrunner-Harer nerve simplexes was proposed. The algorithm achieves precise result in capturing the region of interest overlaid by a nerve that's useful for image object shape representation. The proposed algorithm relies heavily on the robust feature points that ideally reside on the boundary of the object shape. Thus the degree of precision is affected by such

feature point generation methods. This chapter also presents an intuition and algorithm for classification of the proposed nerve detection. In an image where there's clutter on the foreground around the object shape, cluster of nerves that share proximal relation, can capture the image object shape effectively.

In Chapter 4, persistence homology, stability measurement and other measurements has been explored and applied over a sequence of image frames. Each image frame have been measured and compared with the others in term of nerve persistence and stability measurement. Distance measure between each persistence measure shows how each nerve features persist in each of the video frame. This leads to an application in determining maximum scene change and a key frame in a sequence of image frame. It's important to note that the persistence measure is based on the location and detection of the feature point, and thus the persistence measures are also subject to such feature point generation method.

5.2 Recommendation and Future Work

The application of the proposed algorithm for computational proximity and Edelsbrunner-Harer nerve simplexes can be considered as an fast region of interest proposal method, which can be applied in object shape detection and classification tasks. Machine learning model can be experimented on the detected region for object recognition: random forest [20], support vector machine (SVM)[36] or neural networks(NN)[51].

Another application for the proposed technique in Chapter 4 is in content-based image retrieval and scene change detection. The computational complexity increases with the size of the video and the number of images frames. In order to improve the proposal technique, other background removal and feature point generation technique can be further explored.

Bibliography

[1] O. B. A. El-ghazala and S. Belkasimb. Invariant curvature-based fourier shape descriptors. *Journal of Visual Communication and Image Representation*, 23:622–633, 2012. doi: https://doi.org/10.1016/j.jvcir.2012.01.011.

[2] E. A-iyeh. Voronoi tessellation quality: Applications in digital image analysis. *University of Manitoba*, pages 27–28, 2017.

[3] R. P. A. Pentland and S. Sclaroff. Photobook: Tools for content-based manipulation of image database. *In Proc. of SPIE on Storage and Retrieval for Image and Video Databases, San Jose, CA*, pages 34–47, 1994.

[4] M. Z. Ahmad and J. F. Peters. Proximal cech complexes in approximating digital image object shapes. theory and application. *Theory and Applications of Mathematics and Computer Science*, 2017.

[5] P. Alexandroff. Uber den allgemeinen dimensionsbegriff und seine beziehungen zur elementaren geometrischen anschauung. In *Mathematische Annalen*, volume 98, pages 617–635, 1928. DOI: https://doi.org/10.1007/BF01451612.

[6] K. P. Amanmeet Garg, Donghuan Lu and M. F. Beg. Topology of surface displacement shape feature in subcortical structures. *International Workshop on Graphs in Biomedical Image Analysis*, 2017.

[7] A. P. Andreas Opelt and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision*, pages 16–44, 2008.

[8] H. Anton. *Calculus: A New Horizon, 6th ed.* Newyork: Wiley, 1999.

[9] B. Anup and L. Xiaobo. *Computer Vision: Systems, Theory And Applications.* World Scientific, 1993.

[10] H. Asada and M. Brandy. The curvature primal sketch. *IEEE Trans. Pattern Anal. Mach. Intell*, 8(1):2–14, 1986.

[11] M. S. K. Babu Mehtre and W. F. Lee. Shape measures for content based image retrieval: A comparison. *Information Processing and Management*, pages 319–337, 1997.

[12] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice Hall, first edition, 1982.

[13] S. Belongie and J. Malik. Matching with shape contexts. *IEEE Workshop on Contentbased Access of Image and Video Libraries*, 2000.

[14] H. H. Bernd Jahne and P. Geibler. *Handbook of Computer Vision and Applications.* Academic Press, 1999.

[15] A. D. Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Trans. Pattern Anal. Mach. Intell*, 19(2):121–132, 1997.

[16] H. Blum. *A transformation for extracting new descriptors of shape*. 1967.

[17] D. Bontemps and S. Gadat. Bayesian methods for the shape invariant model. *Electronic Journal of Statistics*, 8:1522–1568, 2014. ISSN: 1935-7524.

[18] G. Bordogna. Pictorial indexing for an integrated pictorial and textual information retrieval environment. *Information Services and Use*, 16:165–173, 1990.

[19] K. Borsuk. On the imbedding of systems of compacta in simplicial complexes. *Fundamenta Mathematicae*, 35(1):217–234, 1948.

[20] L. Breiman. Random forests. In *Machine Learning*, volume 45(1), pages 5–32. Kluwer Academic Publishers, 2001.

[21] R. Brunelli. *Template Matching Techniques in Computer Vision*. 2009.

[22] J. Cai and Z.-Q. Liu. Hidden markov models with spectral features for 2d shape recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1454–1458, 2001. doi: 10.1109/34.977569.

[23] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[24] A. Cawkell. Imaging systems and picture collection management: a review. *Information Services and Use*, 12:301–325, 1992.

[25] E. Cech. *Topological Spaces*. John Wiley & Sons, 1966.

[26] A. Chakravarthy. Toward semantic retrieval of pictures and video. *In Proc. Riao'94, Intelligent Multimedia Information Retrieval Systems and Management*, pages 676–686, 1994.

[27] R. Chellappa and R. Bagdazian. Fourier coding of image boundaries. *IEEE Trans. Pattern Anal. Mach. Intell*, 6(1):102–105, 1984.

[28] C. Chen and C. Chang. An object-oriented similarity retrieval algorithm for iconic image databases. *Pattern Recognition Letters*, 14:465–470, 1993.

[29] D. Chetverikov and Y. Khenokh. Matching for shape defect detection. *Lecture Notes in Computer Science*, 1689:367–374, 1999.

[30] E. R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*. 1997.

[31] B. N. Delaunay. *Sur la sphère vide*. Izvestia Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk (7), 1934.

[32] U. Desai. Coding of segmented image sequences. *M.I.T. Media Laboratory Perceptual Computing Section Technical Report*, No. 286, 1994.

[33] G. Dudek and J. Tsotsos. Shape representation and recognition from multiscale curvature. *Comput. Vision Image Understanding*, 68(2):170–189, 1997.

[34] I. G. E. Binaghi and T. Schettini. Image retrieval using fuzzy evaluation of color similarity. *International Jour. Pattern Recognition and Artificial Intelligence*, 8:945–968, 1994.

[35] T. L. E. Deardorff and J. M. et al. Video scene decomposition with the motion picture parser. *In IS&T/SPIE Symposium on Electronical Imaging Science &*

Technology (Digital Video Compression and Processing on Personal Computers: Algorithms and Technologies), San Jose, CA, SPIE 2187:44–55, 1994.

[36] R. F. E. Osuna and F. Girosi. Training support vector machines: An application to face detection. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, page 130–136, 1997.

[37] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[38] X. G. Eric Verdiere, Gregory Ginot. Multinerves and helly numbers of acyclic families. *Symposium on Computational Geometry - SoCG '12, Jun 2012, Chapel Hill, United States*, pages 209–218, 2012.

[39] B. T. F. et al. Statistical tools for topological data analysis, 2018. https://www.rdocumentation.org/packages/TDA/versions/1.6.4l.

[40] S. S. Federico Tombari and L. D. Stefano. Unique signatures of histograms for local surface description. *European Conference on Computer Vision*, pages 356–369, 2010.

[41] W. Forstner and E. Gulch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. *ISPRS*, 1987.

[42] D. A. Forsyth and J. Ponce. *Computer Vision, A Modern Approach*. Prentice Hall, 2003.

[43] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. Electron. Comput*, pages 260–268, 1961.

[44] K. Fu. Syntactic methods in pattern recognition. *Academic Press, New York*, 1974.

[45] S. Giannarou and T. Stathaki. *Shape Signature Matching for Object Identification Invariant to Image Transformations and Occlusion*. Springer, first edition, 2007.

[46] R. Gonzalez and R. Woods. *Digital Image Processing*. 1992.

[47] A. Goshtasby. Description anddiscrimination of planar shapes using shape matrices. *IEEE Trans. Pattern Anal. Mach. Intell.*, 7:738–743, 1985.

[48] J. Gratus and T. Porter. Spatial representation: discrete vs. continuous computational models: a spatial view of information. *Theor. Comput. Sci.*, 365(3):206–215, 2016.

[49] W. Groskey and R. Mehrotra. Index-based object recognition in pictorial data management. *Comput. Vision Graphics Image Process*, 52:416–436, 1990.

[50] D. L. H. Edelsbrunner and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.

[51] S. B. H. Rowley and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[52] C. Harris and M. Stephens. A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[53] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[54] F. Hausdorff. Set theory. *American Mathematical Soc.*, 119, 1957.

[55] C. J. Henry. Near sets: Theory and applications. *Ph.D thesis*, 2010.

[56] T. T. Herbert Bay, Andreas Ess and L. V. Gool. Speeded-up robust features. In *Computer Vision and Image Understanding*, volume 110, pages 346–359, 2008.

[57] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8, 1962.

[58] Y. Hu and Z. Li. An improved shape signature for shape representation and image retrieval. *Journal of Software*, 8(11):2925–2929, 2013.

[59] C. L. Huang and D. H. Huang. A content-basedimage retrieval system. *Image Vision Comput*, 16:149–163, 1998.

[60] J. v. G. I. Everts and T. Gevers. Evaluation of color spatio-temporal interest points for human action recognition. *IEEE Transactions on Image Processing*, 23(4):1569–1589, 2014.

[61] T. K. I. Sekita and N. Otsu. Complex autoregressive model for shape recognition. *IEEE Trans. Pattern Anal. Mach. Intell*, 14:489–496, 1992.

[62] J. W. I. Yong and J. Bowie. An analysis technique for biological shape. *Comput. Graphics Image Process*, 25:357–370, 1974.

[63] J. B. J. Brolio, B. Draper and A. Hanson. Sr: a database for symbolic processing in computer vision. *Computer*, 22:22–30, 1989.

[64] C. T. J. Gu, H. Shu and L. Luo. A novel algorithm for fast computation of zernike moments. *Pattern Recogn*, 35(12):2905–2911, 2002. doi:10.1016/S0031-3203(01)00194-7.

[65] R. Jain. Nsf workshop on visual information management systems. *Computer Science and Engineering Division, The University of Michigan: Workshop Report*, 1992.

[66] L. Jean. *L'anneau d'homologie d'une representation*, volume 222. Les Comptes rendus de l'Academie des sciences, 1946.

[67] L. Kitchen and A. Rosenfeld. Gray-level corner detection. *Pattern Recognition Letters*, 1:95–102, 1982.

[68] J. J. Koenderink and W. Richards. Two-dimensional curvature operators. *Journal of the Optical Society of America*, 5:1136–1141, 1988.

[69] R. F. L. Fei-Fei and P. Peronai. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.

[70] I. Laptev and T. Lindeberg. Space-time interest points. *International Conference on Computer Vision. IEEE*, page 432–439, 2003.

[71] T. Le and M. Yamada. Persistence fisher kernel: A riemannian manifold kernel for persistence diagrams. *RIKEN Center for Advanced Intelligence Project, Japan*, 2018.

[72] C. Leung. Architecture on an image database system. *Information Services and Use*, 10:391–397, 1990.

[73] F.-F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2(524):vol. 59 no.4 pp 536–542, 2005.

[74] S. Z. Li. Shape matching basedon invariants. *Shape Analysis, Progress in Neural Networks*, 6:203–228, 1999.

[75] T. Lindeberg. Scale-space. In *Wiley Encyclopedia of Computer Science and Engineering*, volume IV, page 2495–2504, 1994.

[76] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Springer, 1994.

[77] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, pages 79–116, 1998.

[78] T. Lindeberg. Spatio-temporal scale selection in video data. *Journal of Mathematical Imaging and Vision*, 60(4):525–562, 2018.

[79] T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d structure. *Image and Vision Computing*, 15, 1997.

[80] M. Lodato. On topologically induced generalized proximity relations. *Ph.D thesis. Rutgers University*, 1962. Supervisor: S. Leader.

[81] D. Lowe. Object recognition from local scale-invariant features. In *Proc. 7th IEEE Int. Conf. on Computer Vision*, volume 2, pages 1150–1157, 1999. DOI: 10.1109/ICCV.1999.790410.

[82] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[83] G. Lu and A. Sajjanhar. Region-basedshape representation and similarity measure suitable for content-basedimage retrieval. *Multimedia Syst*, 7:165–174, 1999.

[84] W. N. e. a. M. FLickner, H. Sawhney. Query by image and video content: the qbic system. *IEEE Computer: Special issue on Content Based Picture Retrieval Systems*, 1995.

[85] V. H. M. Sonka and R. Boyle. Image processing, analysis and machine vision. *Chapman & Hall, London, UK, NJ), Melbourne, Australia*, pages 193–242, 1993.

[86] W. W. M. Yeung, B. Yeo and B. Liu. Video browsing using clustering and scene transitions on compressed sequences. *In IS&T/SPIE Symposium on Electronical Imaging Science & Technology (Multimedia Computing and Networking), San Jose, CA*, 1995.

[87] H. A. e. a. Michał Adamaszek. Nerve complexes of circular arcs. In *Discrete Computational Geometry*, volume 56, page 56:251, 2016. DOI: https://doi.org/10.1007/s00454-016-9803-5.

[88] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. *ICCV*, pages 525–531, 2001.

[89] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[90] H. B. Mitchell. Image key points. In *Image Fusion*, pages 163–166. Springer, Berlin, Heidelberg, 2010.

[91] H. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. *Tech Report CMU-RI-TR-3 Carnegie-Mellon University, Robotics Institute*, 1980.

[92] T. Morris. *Computer Vision and Image Processing*. CMacmillan Education UK, 2003.

[93] G. Nagy. Image databases. *Image and Vision Computing*, 3(3):111–117, 1985.

[94] I. Obayashi and Y. Hiraoka. Persistence diagrams with linear machine learning models. *Journal of Applied and Computational Topology*, 1:421–449, 2018.

[95] T. Pavlidis. Algorithms for graphics and image processing. *Computer Science Press, Rockville, MD*, page 143, 1982.

[96] J. Peters and S. Naimpally. Applications of near sets. *Notices Amer. Math. Soc.*, pages vol. 59 no.4 pp 536–542, 2012.

[97] J. Peters and S. Ramanna. Shape descriptions and classes of shapes. a proximal physical geometry approach. In *Advances in Feature Selection for Data*

and Pattern Recognition, volume 138, pages 203–225. Springer International Publishing., 2018.

[98] J. F. Peters. Proximal delaunay triangulation regions. *arXiv:1411.6260 [math-MG]*, pages 1–4, 2014.

[99] J. F. Peters. Proximal voronoi regions. *arXiv:1411.3570 [math-MG]*, pages 1–4, 2014.

[100] J. F. Peters. *Computational Proximity. Excursions in the Topology of Digital Images*. Springer International Publishing Switzerland, first edition, 2016.

[101] J. F. Peters. *Foundation of Computer Vision*. Springer International Publishing Switzerland, first edition, 2017.

[102] J. F. Peters. *Topology of Digital Images: Visual Pattern Discovery in Proximity Spaces*. Springer Verlag Berlin Heidelberg, 2017.

[103] M. Peura and J. Iivarinen. Efficiency of simple shape descriptors. *in Proceedings of the Third International Workshop on Visual Form, Capri, Italy*, page 443–451, 1997.

[104] R. Picard and T. Minka. Vision texture for annotation. *Multimedia Systems*, 3:3–14, 1995.

[105] Y. L. e. a. R.L. Kennedy. Solving data mining problems through pattern recognition. *Prentice-Hall, PTR, Upper Saddle River, NJ*, pages 11–16, 1998.

[106] S. B. Roberto Cipolla and G. M. Farinella. *Computer Vision: Systems, Theory And Applications*. Springer, 2010.

[107] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. *European Conference on Computer Vision*, 2006.

[108] W. Rucklidge. Efficient visual recognition using the hausdorff distance. *Springer Verlag New York Inc.*, 1996.

[109] A. B. S. Berretti and P. Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Trans. Multimedia*, 2(4):225–239, 2000.

[110] M. Safar and C. Shahabi. Mbc-based shape retrieval: basics, optimizations, and open problems. *Springer Science*, pages 191–208, 2006.

[111] L. Shapiro. *Computer Vision.* Prentice Hall, 2011.

[112] S. M. Smith and J. M. Brady. Susan – a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[113] D. M. Squire and T. M. Caelli. Invariance signature: characterizing contours by their departures from invariance. *Comput. Vision Image Understanding*, 77:284–316, 2000.

[114] M. Swain and D. Ballard. Indexing via color histogram. *In DARPA Image Understanding Workshop*, 11-13:623–630, 1990.

[115] R. Szeliski. *Computer Vision. Algorithms and Applications.* Springer, 2011.

[116] J. K. T. Hermes, C. Klauck and J. Zhang. Image retrieval for information systems. *In Proc. SPIE—The Inter. Soc. for Optical Engineering, Storage and Retrieval for Image and Video Databases*, pages 394–403, 1995.

[117] G. Taubin and D. Cooper. Object recognition basedon moment. *Geometric Invariance in Computer Vision, MIT Press, Cambridge, MA*, pages 375–397, 1992.

[118] M. Teague. Image analysis via the general theory of moments. *J. Opt. Soc. Am.*, 70(8):920–930, 1980.

[119] Thessaloniki. Shape description, 1998.

[120] Q. Tieng and W. Boles. Recognition of 2d object contours using the wavelet transform zero-crossing representation. *IEEE Trans. Pattern Anal. Mach. Intell*, 19(8):910–916, 1997.

[121] C. Tomasi and J. Shi. Detection and tracking of point features. *Pattern Recognition*, 37:165–168, 2004.

[122] M. Trajkovic and M. Hedle. Fast corner detection. *Image and Vision Computing*, 16(2):75–87, 1998.

[123] L. Trujillo and G. Olague. Automated design of image operators that detect interest points. *Evolutionary Computation*, 16(4):483–507, 2008.

[124] M. Turk and A. Pentland. Eigenfaces for recognition. 3(1):71–86, 1991.

[125] P. van Otterloo. A contour-orientedapproach to shape analysis. *Prentice-Hall International Ltd*, pages 90–108, 1991.

[126] G. Voronoi. Sur un problème du calcul des fonctions asymptotiques. *Journal für die reine und angewandte Mathematik*, 126:241–282, 1903.

[127] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les paralléllοèdres primitifs. *Journal für die reine und angewandte Mathematik*, 134:198–287, 1908.

[128] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. premier mémoire. sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die reine und angewandte Mathematik*, 133:97–178, 1908.

[129] R. B. W. Niblack and W. E. et al. The qbic project: Querying image by content using color, texture, and shape. *In Proc. SPIE Storage and Retrieval for Image and Video Database, San Jose, CA*, pages 173–181, 1993.

[130] H. Wang and M. Brady. Real-time corner detection algorithm for motion estimation. *Image and Vision Computing*, 13(9):695–703, 1995.

[131] Weisstein and W. Eric. Geometric centroid, 2018. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/GeometricCentroid.html.

[132] Weisstein and W. Eric. Lamina, 2018. A Wolfram Web Resource. http://mathworld.wolfram.com/Lamina.html.

[133] A. Willis and Y. Sui. An algebraic model for fast corner detection. *IEEE 12th International Conference on Computer Vision*, pages 2296–2302, 2009.

[134] M. Wozniak and Z. Marszalek. An idea to apply firefly algorithm in 2d image key-points search. In *Information and Software Technologies*, volume 465, pages 312–323. Springer, Cham, 2014.

[135] Y. L. Xiaoyi Feng and et. al. Extracting local binary patterns from image key points: Application to automatic facial expression recognition. In *Image Analysis*, volume 7944, pages 339–348. Springer, Berlin, Heidelberg, 2013.

[136] D. Zhang and G. Lu. A comparative study of fourier descriptors for shape representation and retrieval. *in: Proceedings of the Fifth Asian Conference on Computer Vision (ACCV02), Melbourne, Australia*, pages 646–651, 2002.

[137] D. Zhang and G. Lu. Generic fourier descriptor for shape-basedimage retrieval. *Proceedings of IEEE International Conference on Multimedia and Expo (ICME2002)*, 1:425–428, 2002.

[138] D. Zhang and G. Lu. Review of shape representation and description techniques. *The journal of the Pattern Recognition Society*, pages 1–19, 2004.

[139] G. M. Ziegler. *Lectures on Polytopes*, volume 152. Springer-Verlag, 1995.