# Thesis Name

Javier Fernández

June 20, 2010

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

## 1.2 Thesis motivation and purpose

## 1.3 The problem

## 1.4 Objectives

### 1.4.1 General Objectives

### 1.4.2 Specific Objectives

# Chapter 2

# Theoretical Framework

## 2.1 Endrov

## 2.2 Thresholding

Thresholding is a process of image segmentation that can be used to create binary images from grayscale images.A binary image is a type of discrete image in which the value of a pixel is either 1 or 0 depending on whether the pixel belongs to the foreground or to the background.

As stated on [?] during the thresholding process, individual pixels in an image are marked as "object" pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as "background" pixels otherwise. This convention is known as *threshold above.* Variants include *threshold below*, which is opposite of threshold above; *threshold inside*, where a pixel is labeled "object" if its value is between two thresholds; and *threshold outside*, which is the opposite of *threshold inside* [?]. Typically, an object pixel is given a value of 0 while a background pixel is given a value of 1 Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's label.

In image processing applications where the study is focused on particular objects contained in an image, thresholding becomes an effective and simple tool to separate these objects from the background. Commongly, the gray levels belonging to the object are substantially different from the gray levels of the background pixels. In [?, p.146] many thresholding applications on image processing are mentioned such as: document image analysis, where the goal is to extract printed characters, logos, graphical content, or musical scores; map processing where lines, legends and characters are to be found; scene processing, where a target is to be detected; and quality inspection of materials, where defective parts must be delineated, among many others.

The key parameter in the thresholding process is the thresholding value (or values for *threshold inside approach*). The value can be automatically computed, what is called *automatic thresholding*, as well as set or tuned through user input.
According to the information they are exploiting, the different thresholding methods can be categorized. In [**?**, p.147], Sezgin and Sankur categorize the thresholding methods in six groups:

- Histogram shape-based methods: the peaks, valleys and curvatures of the smoothed histogram are analyzed
- Clustering-based methods: gray-level samples are clustered in two parts as background and foreground, or modeled as a mixture of two Gaussians.
- Entropy-based methods: algorithms that use the entropy of the foreground and background regions, the cross-entropy between the original and binarized image, etc.
- Object attribute-based methods:Ssearch a measure of similarity between the gray-level and the binarized images, such as fuzzy shape similarity, edge coincidence, etc.
- Spatial methods: use higher-order probability distribution and/or correlation between pixels
- Local methods: adapt the threshold value on each pixel to the local image characteristics.

Below, in Fig.**??**, two images are shown that correspond to a grayscale image and binary image obtained by thresholding.

## 2.3 Distance Transform

A distance transform or distance map is a representation of a digital image that is obtained by converting a digital binary image, consisting in object and non-object pixels, to another

(a) Grayscale Image

(b) After Threshold effect image

Figure 2.1: Grayscale Image before and after a thresholding effect is applied. Images taken from [?]

image in which each pixel has a value corresponding to the distance to the nearest non-object pixel. The object pixels can be considered as foreground and the non-object pixels as background. The obtained image is then a sort of grayscale representation of the foreground pixels in the binary image.
The pixel mapping depends mainly on the distance metric, which is the measurement method of distance between image pixels. Different metrics have been studied to find distance maps such as *City Block or Manhattan*, *Chessboard*, *Euclidean*, *Chamfer 3-4*, *Octogonal*, among others.[?, p.363]. There exist a great amount of distance metric of other kinds, that are useful for different purposes. These are commongly derived from the previous. Below are shown the images obtained by applying different distance metrics.

As stated on [?] Distance transforms play a central role in the comparison of binary images, particularly for images resulting from local feature detection techniques such as edge or corner detection. For example, both the Chamfer and Hausdorff matching approaches make use of distance transforms in comparing binary images. Distance maps can also be interpreted as landscapes of islands where the label of every pixel indicates the height of the region. This allows the detection of ridges and peaks which is a straightforward way to find the skeleton of an object.[?, 237]. The nature of distance transforms in which the objects are represented as contour layers of different depth makes them also a useful tool to edge analysis and to improve efficiency of morphology algorithms such as *Thinning* and

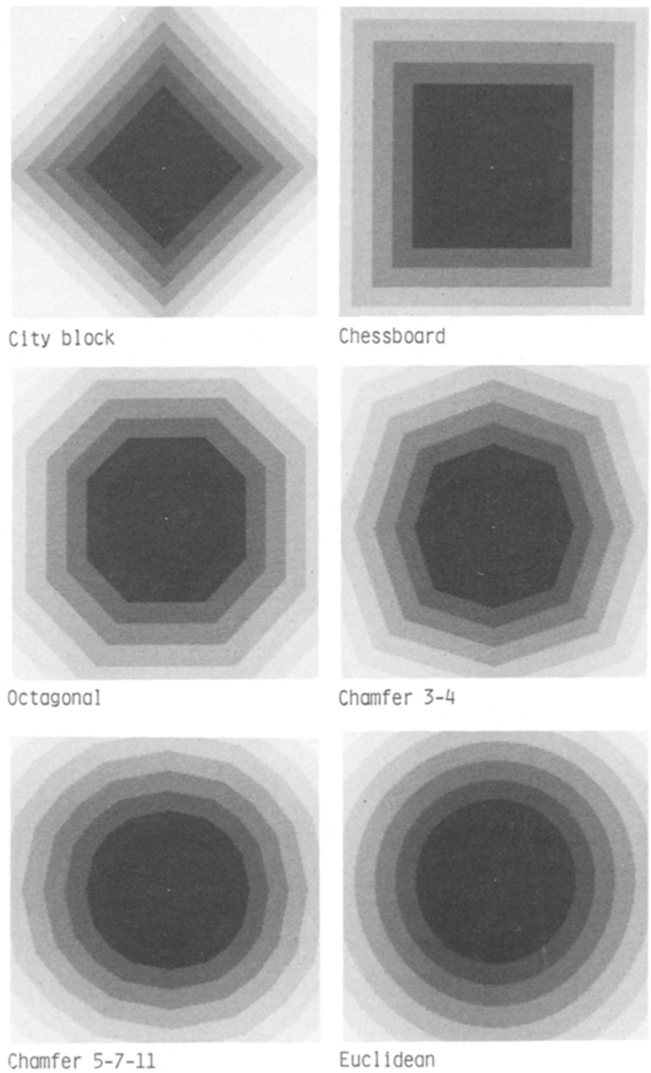



Figure 2.2: The distances from a point for the six DTs. The lighter the color the larger the distance [**?**, p.365]

*Thickening.*

## 2.4 Skeletonization

## 2.5 Image Segmentation

### 2.5.1 Clustering

## 2.6 Shape Matching

Shape matching is a central problem in visual information systems, computer vision, pattern recognition, and robotics [?].It consist in identifying the area or contour of an specifical shape or class of shapes in an image, and plays a fundamental role in content extraction from images and content-based image retrieval. In [?] Veltkamp explains that shape matching deals with transforming a shape and measuring the resemblance with another one, using some similarity measure, that normally correspond to the notion of distance between shapes. The concept of shape is abstract, but most approaches in shape matching represent a shape as a geometrical object. This can be both a set of points, curves, surfaces, solids,etc. and a geometrical pattern modulo some transformation group, in particular similarity transformations (translations, rotations and scalings), as it is stated on [?].

There are different studied ways to approach the shape matching problem. Since the approach followed in this thesis work is related to those that deal with computational geometry, the emphasis on this section will be on the last ones. Computational Geometry studies algorithms that can be declared in terms of geometry.

In [?] Veltkamp and Hagedoorn mention different approaches of shape matching such as: tree prunning, the generalized Hough transform or pose clustering, geometric hashing, the alignment method, statistics, deformable templates, relaxation labeling, Fourier descriptors, wavelet transform, curvature scale space and neural networks. They also categorize the matching techniques in two main groups: *global image transforms* and *global objects methods*. The *global image transform* group refers to the techniques that "transform the image from color information in the spatial domain to color variation in the frequency domain". These approaches do not represent the shape explicitly for matching, instead they represent color or intensity transitions in the image. This makes impossible to measure the difference of two images in terms of shape as well as match a shape with a specifical part of an image. On the other hand the *global object methods* work with a complete object area or contour and can analyze specifical areas in the image instead of requiring to process the whole image as in the global image transforms. In order to perform a proper matching, the objects in the image have to be completely and clearly segmented. Some of these methods are: *moments*, where an object is described as a set of moments, *modal matching*, where the boundary is used insted of the area and is described with Fourier descriptors and *curvature scale space*, where a a scale space and parameterized representation of the contour of the objects is used.

Veltkamp describes in [**?**] four different forms in which shape matching is studied, given two shape patterns and a dissimilarity measure. These are:

- **Computation Problem:** Compute the dissimilarity between the two patterns
- **Decision Problem:** For a given threshold, decide whether the dissimilarity is smaller than the threshold
- **Decision Problem:** For a given threshold, decide whether there exists a transformation such that the dissimilarity between the transformed pattern and the other pattern is smaller than the threshold
- **Optimization Problem:** Find the transformation that minimizes the dissimilarity between the transformed pattern and the other pattern.

A well studied optimization approach for shape matching is Active Contour Models (*Snakes*), in which is inspired much of the shape fitting approach of this work (see Sec **??**).In [**?**] a snake is defined as an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. The *snakes* are said to be active contour models because they lock onto nearby edges, localizing them accurately.

The *snakes* model is defined as a controlled continuity spline that is bound to internal and external image forces, called energies. The external energy models how well the deformed model matches the data. The internal energy models the objects resistance to be pushed by the external force into directions not coherent with the prior knowledge, as it is stated on [?].The internal energy imposes the a “piecewise smoothness constraint”,[?]. This means that a contour is pushed to an image feature by the external force while the contour itself exhibits resistance to be deformed into a non-smooth curve.

Given these definitions, let $M$ be the model and $D$ a data set the energy $E$ can be defined as:

$$E(M) = E_{ext}(M, D) + E_{int}(M)$$

where $E_{ext}$ is the external energy function and $E_{int}$ the internal energy function.As it is explained on [?] the image forces push the snake toward salient image features like line, edges and subjective contours, while the external constraint forces are responsible for putting the snake near the desired local minimun. The optimization algorithm will consist then in minimizing this objective function until the best solution is found.

## 2.7 Cardinal Splines

## 2.8 Triangle mesh and rasterization

# Chapter 3

# Methodology

## 3.1 Development Methodology

## 3.2 Solution Design Methodology

### 3.2.1 Description

similarity measures of Vetkamp has a nice explanation of how to approach shape matching.

### 3.2.2 Thresholding

Since the main purpose of this study is to fit the shape of *C.Elegans* worms on digital images, it is useful to differentiate these from the rest of the image in order to perform a more accurate analysis. The shape of the worms can be characterized as objects and the rest of the image as background. More precisely the image pixels can be separated into two groups: object pixels, that are all of those that belong to a worm shape and background pixels, that are all the remaining ones.

Given this theoretical characterization, a thresholding filter would come to be a useful tool to locate the objects of study in the digital representation and to discard unnecessary information, obtaining a binary image from the original one. A binary image would then provide an initial segmentation of the processed image, being as well a key element to obtain a distance transformation, as it is explained in Sec. **??**.

(a) Original image (b) Percentil Threshold. Value=0.074

Figure 3.1: Worms in liquid media original image and binary image obtained through Percentile Thresholding with a percentile value of 0.074

#### 3.2.2.1 Implementation

There are four thresholding filters for 2D images implemented on *Endrov*, these are: *Fukunaga*,*Max entropy*, *Otsu* and *Percentile*, that cover the histogram and entropy-based thresholding methods categories as defined in Sec.**??**. Considering that the implemented methods are sufficiently different and given the transparency of *C.Elegans* worms is hard to determine theoretically which would be the most appropriate thresholding method to obtain an accurate binary image, from the study data-set. In order to select a thresholding method a series of experiments where performed tweaking the parameters for the different mentioned methods, as it is explained on Sec.**??**. The selected method was *Percentile Threshold 2D* with a percentile value oscilating from 0.072 to 0.09 on the different test images.

Figure **??** shows a binary image obtained after applying the *Percentile Threshold 2D* method with a percentile value of 0.074

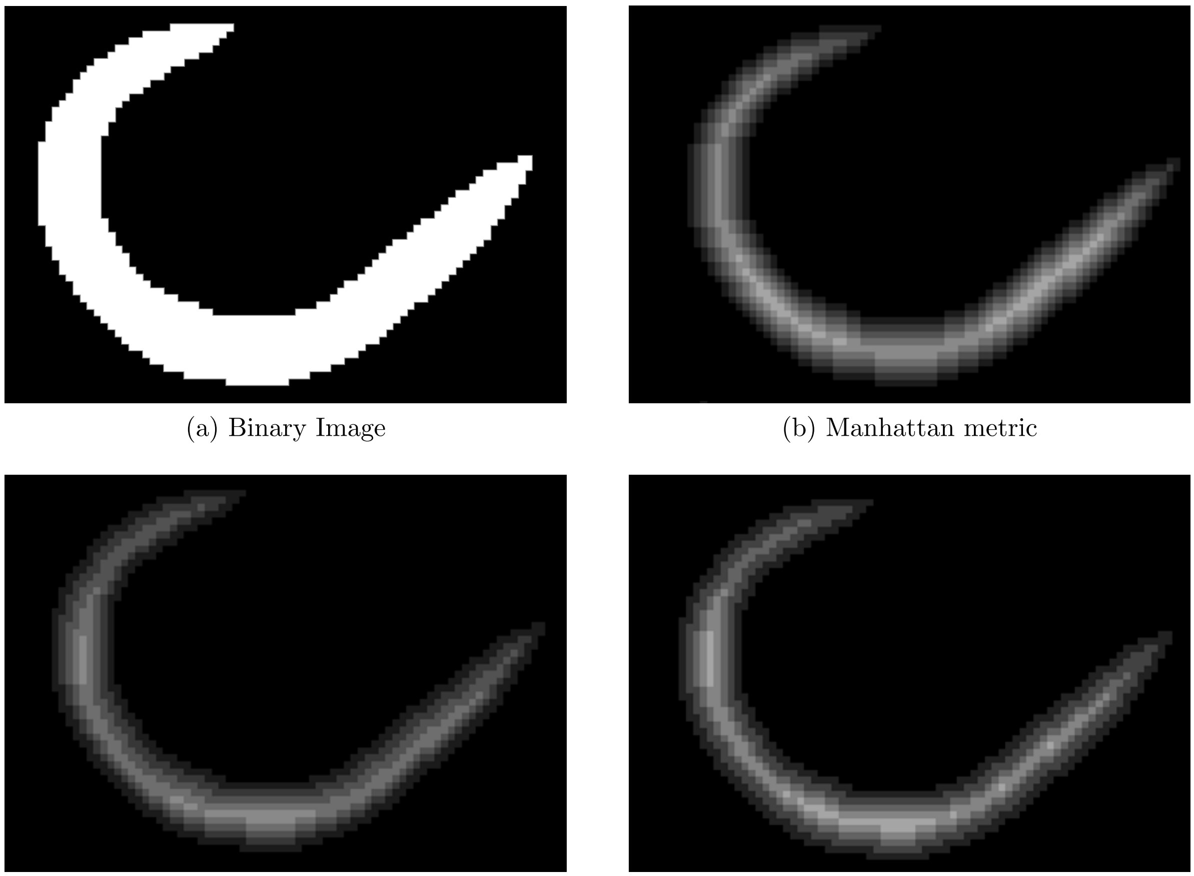
### 3.2.3 Distance transformation

In this shape fitting approach for *C.Elegans* worms the distance transformation of the given image is used thoroughly for contour detection and different kinds of image segmentation procedures. Specifically the distance map allows to detect and follow the exact contour of isolated worms (Sec. **??**), is useful in the shape profile generation (Sec. **??**), and essential in the heuristical guessing of the more likely worm-paths on *worm clusters* (Sec. **??**). It also improves the performance of the iterative thinning algorithm designed by *Zhang and Suen* [**?**] as it is described on Sec.**??**

#### 3.2.3.1 Implementation

As stated in [**?**, p.196] the algorithms of DT can be categorized into two classes: one is the iterative method which is efficient in a cellular array computer since all the pixels at each iteration can be processed in parallel, and the other is sequential (or recursive) method which is suited for a conventional computer by avoiding iterations with the efficiency to be independent of object size. Using the general machines that most people working in digital image processing have access to, sequential algorithms are often much more efficient than iterative ones. For this reason a sequential approach was chosen to calculate the distance transformation of the input images. Particularly the two-scans transformation using 3x3 neighborhoods [**?**] which is both efficient and easy to implement.

In the mentioned paper a distance map calculation algorithm is described which consist on only two scans of the image bitmap, one left to right - top to bottom, and another right to left - bottom to top, with one operation per pixel. This makes the complexity of the algorithm $\mathcal{O}(N)$ for $N$ the size of the image array. In [**?**, p.197] a pseudo-code for *Chessboard* and *Manhattan or city-block* distances is given, while in [**?**, p.198] the definition is extended to improve the efficiency of the calculations needed to generate a distance map using *Euclidean* distances. The two-scans algorithm was implemented using the three different distance metrics mentioned before. This allows a wider analysis on the behavior and accuracy of the shape fitting process from one metric to another."The city block or chessboard distance measures are sensitive to the rotations of an object, but the Eculidean distance measure is rotation invariant. However, its square root operation is costly..." [**?**, p.332]. Given the strait shape of worms and the different levels of accuracy of the distance metrics it is hard to



(a) Binary Image (b) Manhattan metric

(c) Chessboard metric (d) Euclidean metric

Figure 3.2: Binary Image and Three Distance Transformation metrics from a single worm image

tell at first sight which would be the most adequate to use. The figure **??** shows the binary image three and the distance maps obtained from a single worm image.

### 3.2.4 Worm Skeletonization

### 3.2.5 Worm Segmentation

#### 3.2.5.1 Clustering

#### 3.2.5.2 Path guessing

### 3.2.6 Worm Shape Descriptor

#### 3.2.6.1 Worm Profile Generation

### 3.2.7 Triangle mesh and rasterization

### 3.2.8 Profile-driven shape fitting

Check b-splines snakes.. Talks about avoding internal energy

#### 3.2.8.1 Worm Cluster shape fitting

#### 3.2.8.2 Isolated Worms shape fitting

# Bibliography

[1] Carlo Arcelli and Gabriella Sanniti di Baja. Ridge points in euclidean distance maps. *Pattern Recognition Letters*, 13:237–243, 1992.

[2] Gunilla Borgefors. Distance transformation in digital images. *Computer Vision, Graphics and Image Processing*, 34:344–371, 1986.

[3] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Cornell Computing and Information Science Technical Report*, 2004.

[4] Andrew Witkin Michael Kass and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–333, 1988.

[5] Mehmet Sezgin and Bulent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13:146–165, 2004.

[6] Linda G. Shapiro and George C. Stockman. *Computer Vision*. Prentice Hall, 2002.

[7] Frank Y. Shih and Christopher C.PU. A skeletonization algorithm by maxima tracking on euclidean distance transform. *Pattern Recognition*, 28:331, 1994.

[8] Thomas Vetter Thomas Albrecht, Marcel Luthi. Deformable models. gravis.cs.unibas.ch/publications/CH$_{D}eformable_{M}odels09.pdf$.

[9] Remco C. Veltkamp. Shape matching: Similarity measures and algorithms. Dept. Computing Science, Utrecht University.

[10] Remco C. Veltkamp and Michiel Hagedoorn. State-of-the-art in shape matching. *Principles of visual information retrieval.-(Advances in pattern recognition)*, 2:87–112, 2001.

[11] Wikipedia. Thresholding(image processing). http://en.wikipedia.org/wiki/Thresholding_(image_processing), June 2010.

[12] Frank Y.Shih* and Yi-Ta Wu. Fast euclidean distance transformation in two scans using a 3 x 3 neihborhood. *Computer Vision and Image Understanding*, 93:195–205, 2002.

[13] T.Y. Zhang and C.Y. Suen. A fast parallel algorithm for thinning digital patterns. *Image Processing and Computer Vision*, 27:235–239, 1984.