Feature extraction for image and point set analysis

Demetrios P. Gerogiannis

Ph.D. Dissertation

— ♦ —

Ioannina, December 2014

Τμημα Μηχανικων Η/Υ & Πληροφορικης
Πανεπιστημιο Ιωαννινων

Department of Computer Science & Engineering
University of Ioannina

Εξαγωγή Χαρακτηριστικών για Ανάλυση Εικόνων και Σημείων

Η ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

υποβάλλεται στην
ορισθείσα από την Γενική Συνέλευση Ειδικής Σύνθεσης
του Τμήματος Μηχανικών Η/Υ και Πληροφορικής
Εξεταστική Επιτροπή

από τον

Δημήτριο Π. Γερογιάννη

ως μέρος των Υποχρεώσεων για τη λήψη του

ΔΙΔΑΚΤΟΡΙΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ

Δεκέμβριος 2014

"Όποιος ξέρει από πριν πού θέλει να πάει, δεν πηγαίνει πολύ μακριά."
Ναπολέων Βοναπάρτης

Bolero, Maurice Ravel

Committees

Advisory Committee:

Christophoros Nikou, Associate Professor, Department of Computer Science and Engineering, University of Ioannina, Greece *(supervisor)*

Aristidis Lykas, Professor, Department of Computer Science and Engineering, University of Ioannina, Greece

Ioannis Fudos, Associate Professor, Department of Computer Science and Engineering, University of Ioannina, Greece

Examination Committee:

Christophoros Nikou, Associate Professor, Department of Computer Science and Engineering, University of Ioannina, Greece *(supervisor)*

Aristidis Lykas, Professor, Department of Computer Science and Engineering, University of Ioannina, Greece

Ioannis Fudos, Associate Professor, Department of Computer Science and Engineering, University of Ioannina, Greece

Antonios Argyros, Professor, Department of Computer Science, University of Crete, Greece

Ergina Kavallieratou, Assistant Professor, Department of Information and Communication Systems Engineering, University of the Aegean, Greece

Nikolaos Nikolaidis, Assistant Professor, Department of Computer Science, Aristotle University of Thessaloniki, Greece

Andreas Savakis, Professor, Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, USA

Dedication

I wish to dedicate this work to my parents. They inspired me to walk the path of knowledge. An Ithaca is reached, let us sail to a new one!

Acknowledgements

I wish to thank my parents Panagiotis and Alexandra, my brother Thomas and my best friends Kyriakos, George, Alexander for their material and moral support all those years of study. Moreover, I wish to thank my supervisor, Prof. C. Nikou and Prof. A. Lykas for their guidance during my Academic journey. We had a very constructive collaboration.

Table of Contents

0.1 Overview

0.2 Structure of the thesis

I Features and Applications

1 Modeling sets of unordered points using line segments 1

1.1 Introduction 1

1.2 A Direct Split and Merge (DSaM) Framework for Line Segment Detection 3

1.2.1 Split Process 3

1.2.2 Merge Process 5

1.3 Evaluation of the Line Segment Detection Algorithm 6

1.3.1 Numerical evaluation 7

1.3.2 Comparison with the Hough Transform 11

2 Applications 14

2.1 Vanishing Point Detection 15

2.1.1 Introduction 15

2.1.2 The algorithm 15

2.1.3 Numerical Evaluation 17

2.2 Point cloud sampling and reconstruction 21

2.2.1 Introduction 21

2.2.2 The algorithm 22

2.2.3 Numerical Evaluation 24

2.3 Shape encoding for edge map image compression 30

2.3.1 Introduction 30

2.3.2 The algorithm 30

2.3.3 Numerical evaluation 31

2.4 Retinal Fundus Image Feature Characterization 36

2.4.1 Introduction 36

2.4.2 The algorithm 37

2.4.3 Numerical evaluation 38

2.5 Elimination of outliers from 2D point sets using the Helmholtz principle 41

2.5.1 Introduction 41

2.5.2 The Helmholtz principle 42
2.5.3 The algorithm 42
2.5.4 Numerical evaluation 45

II Image and Point set Registration

3 Registering sets of points using Bayesian regression 42
3.1 Introduction 42
3.2 Registration of sets of points via regression 45
3.3 Experimental Results and Discussion 47

4 Registering images and sets of points using Mixture Models 59
4.1 Introduction 59
4.2 Image registration with mixtures of Gaussian and Student's *t*-distributions 62
4.3 Robust registration of point sets with mixtures of Student's *t*-distributions 67
4.4 Experimental results 68

5 Epilogue 77
5.1 Conclusions 77
5.2 Future Work 79
I The Hungarian algorithm 93
II Relevance Vector Machines 94

List of Figures

1 Example of various images where line segments could be used to describe the depicted information: (a) road cracks, (b) maps, (c) object edges, (d) building edges, (e) road lanes. . . . .

2 Example of images depicting structured worlds. (a) Indoor scene (b),(c) Outdoor scenes. . . . .

3 Example of shape reconstruction. (a) The initial set of points describing a shape, (b) The characteristic points (green stars) are extracted from the initial shape (red points), (c) The reconstruction result (blue points) superpositioned over the initial shape (red points). Notice the small deviation between real and computed data. . . . .

4 Explanation of the registration problem. The goal is to determine that geometric transformation that will be applied on the left image (yellow background) and will align the pixels such that the pixels of the blue circle in the left will be matched with those of the green circle in the right and the pixels of the blue ellipse in the left will be corresponded with those of the green ellipse in the right image. . . . .

1.1 Split process. (a) At iteration $t+1$, the ellipse with center μ^t is split into two new ellipses e_1 and e_2, with centers μ_1^{t+1} and μ_2^{t+1} given by (1.4). (b) The new centers are marked with a star (*). The reassignment of the points to the new centers is shown. Points of one category, assigned to e_1, are marked with a square, while points assigned to e_2, are marked with a circle. 5

1.2 Steps of the split and merge process. The process is initialized with the mean and the covariance of the full set of points. (a) Split into 2 ellipses. (b) Split into 4 ellipses. (c) End of split (35 ellipses). (d) The final merge result (23 ellipses). The figure is better seen in color. 5

1.3 Some representative images of the databases we used in our experiments. Please note that in some cases inner structures exist. This does not permit to extract an ordering of the points (a)MPEG7 [1], (b) Gatorbait [2], (c) Brown [3], (d) ETHZ [4]. 8

1.4 A representative result of the modeling of a shape from the MPEG7 dataset [1]with the proposed (left) and Kovesi [5] (right) methods. Green boxes highlight the differences regarding the modeling error of the two methods. Although in general both methods modeled the shape globally, locally the proposed method modeled more accurately the shape contour. 10

1.5 (a) - (c) The primitive images used to create the artificial dataset for experiments with Gaussian additive noise. (d) Contour degraded by additive Gaussian noise of 18dB. A representative test image produced by randomly repeating the patterns of images in (a)-(c). 11

1.6 Experimental results using the datasets of figure 1.5 (e) that demonstrate the performance of our method in presence of Gaussian additive noise in terms of model complexity error. The vertical axis represents the absolute error between the real number of segments and the one computed by our method. 12

1.7 (a)-(c) Results of the PPHT algorithm to a set of points representing the shape of a bone (MPEG7 dataset) y varying the minimum number of points in a bin (namely, 5, 15 and 25). Only a small fraction of the lines is drawn for visualization purposes. Note the overlapping lines. (d) The result of our method. The figure is better seen in color. 13

2.1 $\Pi_s(\mathbf{x}; \mathbf{a}, \mathbf{b}, \lambda)$ for $\lambda = 50$ and $\lambda = 1$. 17

2.2 Representative results of the VP detection with the proposed method (the figure is better viewed in color). 19

2.3 Representative results (the figure is better viewed in color). 20

2.4 An example of the sampling process. The black points represent the original set of points, while the red line is the is their *summary* computed by DSaM [6]. The vertical blue lines depict the limits of the histogram bins. The green points are those selected to represent the sampled set because they are closer to the mean value of the bin. The figure is better seen in color. . 23

2.5 The value $\delta = 1.6$, which minimizes $\Phi(\delta)$ was used in our experiments. . . . 25

2.6 Representative results of sampling of the Gatorbait dataset [2]. Details of the upper left part of a fish contour. Sampling with (a) the proposed method, (b) the method of Malik [7], (c) Monte Carlo sampling and (d) Random sampling. 27

2.7 Shape reconstruction of the Gatorbait dataset [2] using (a)-(b) DSaM, (c)-(d)NN-S, (e)-(f) Kovesi [5], (g)-(h) vensor voting [8, 9]. 29

2.8 Representative results of the reconstruction method on the Gatorbait [2] dataset. (a) The original image. Results extracted with (b) DSaM [6], (c) polygon approximation [10] with automatic tuning (d) polygon approximation [10] with threshold value set to 5 pixels. 34

2.9 Representative results of the reconstruction method on the MPEG7 [1] dataset. (a) The original image. Results extracted with (b) DSaM [6], (c) polygon approximation [10] with automatic tuning (d) polygon approximation [10] with threshold value set to 5 pixels. 34

2.10 Rate-distortion curves for (a) the Gatorbait dataset [2] and (b) the MPEG7 dataset [1]. The blue line corresponds to the compression results based on DSaM [6] and the red line refers to polygon approximation [11]. 35

2.11 An example of a video object plane. (a) The initial image, (b) the video object plane mask and (c) the reconstruction result with the DSaM method. The red points depict the initial shape and and the green points show the contour reconstructed with our method. 35

2.12 The different features that the proposed algorithm can detect. The yellow point is an *end-point*, the orange point is an *interior-point* and the green point is a *crossover*. All the other points are *junctions* (a *T-junction* is shown in red, and a *bifurcation* is shown in blue). The image is better viewed in colour. 36

2.13 An instance of the point characterization algorithm. Points $\mathbf{x}$ (in red and black) correspond to the thinned lines of the extracted vessels. Green and blue points are the extreme points $\mathbf{y}$ computed by our DSaM algorithm. The yellow circle demonstrates the neighborhood of that point ($\mathcal{N}(\mathbf{y})$). Red and black points lying in that circle are considered as neighbors of that extreme point. In that case, those points belong either to line cluster with index 1 or to line cluster with index 2. Thus, $\mathcal{CN}(\mathbf{y}) = 2$. The orange point corresponds to the nearest neighbor of $\mathbf{y}$ among the points of neighbor line cluster (black points). d is the minimum distance between the aforementioned nearest neighbor and the extreme points of its line cluster. 38

2.14 (a) The original retinal image. (b) The manual segmentation of the vessels in (a). (c) Result of thinning the image in (b). (d) The confidence regions depicted as circles with a radius equal to 1% of the diagonal of the bounding box of the original set. The figure is better seen in color. 39

2.15 The F measure, (2.10), for various values of parameter w in (2.9). The black square indicates the point that corresponds to the maximum value of F measure. This value ($F = 0.95$) occurs for $w = 2.9$ and provides a precision rate of 91.59% and a recall rate of 98.58%. More details are given in section 2.4. 40

2.16 (a) A a set of points (in red color) degraded by equal in number outliers (in blue color). (b) The distribution of the sorted lengths of the line segments approximating the point set of (a) using a line segment detection algorithm. The horizontal axis represents the indices of the segments and the vertical axis represents the lengths. (c) The Pareto distribution for various values of the parameter a with $b = 1$. 43

2.17 An example of the definition of the neighborhood of a segment. Points A and B (cyan squares) are the starting/ending points of segment 1. The yellow circle with radius T determines the neighborhood. Line segments 2 and 3 are part of the neighborhood while segment 4 is not. The same configuration applies to point B. 44
2.18 Outlier elimination from the data set of Fig. 2.16(a) by (a) the first and (b) the last iterations of the proposed method, (c) Xianchao et al. [12], (d) DBScan [13]. The red boxes highlight representative false points provided by the methods. 46
2.19 Boxplots of the line fitting errors for the compared methods. Notice the different scales at the abscissas. 47
2.20 (a) A test image and (b) its degraded version at SNR $= -1$dB. 48
2.21 Line segment modeling of image in figure 2.20(b) computed (a) by DSaM[6], PA [10]. Notice that the PA is trapped by the outliers and produces a large number of short line segments, while DSaM manages to provide a valid model. 48

3.1 A false matching simulation example, with a point set used in [14]. (a) Correct correspondences between the source and the target sets are represented by line segments. (b) Two points were falsely matched on purpose simulating a wrong correspondence solution. The yellow box depicts the false matched points. (c) The result of TPS [15]. Notice that large registration errors are present. (d) The result of the proposed registration scheme based on RVM regression. In this case the registration is correct. 45
3.2 The initial set of points used in our experiments, [14]. (a) *Sine*, (b) *Blob*, (c) *Fish* and (d) *Ideogram*. 50
3.3 (a) 2D range data used in our experiments [16]. (b) 3D set of points representing a face used in our experiments (3D face) [17]. 50
3.4 Rigid transformation experiment with 2D points of a range scan [16]. (a) Reference set of points (red) and deformed set of points (black) of a 3D face. (b) Registration result for the proposed method. 50
3.5 Non rigid transformation experiment. (a) Reference set of points (red) and deformed set of points (black). Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color. 51
3.6 Non rigid transformation experiment. (a) Reference set of points (red) and deformed set of points (black). Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color. 51
3.7 Non rigid transformation experiment with 3D points [17]. (a) Reference set of points (red) and deformed set of points (black) of a 3D face. (b) Registration result for the proposed method. 52

3.8 Rigid transformation experiment in presence of noise. (a) Reference 2D range set of points (red) and deformed set of points (black), [16] corrupted with zero mean additive Gaussian noise. (b) Registration result for the proposed method. The difference is better highlighted in color. 53

3.9 Non rigid transformation experiment in presence of noise. (a) Reference set of points (red) and deformed set of points (black) corrupted with zero mean additive Gaussian noise. Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color. 53

3.10 Non rigid transformation experiment in presence of noise. (a) Reference set of points (red) and deformed set of points (black) corrupted with zero mean additive Gaussian noise. Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color. 54

3.11 Non rigid transformation experiment in presence of noise. (a) Reference 3D set of a face points (red) and deformed set of points (black), [17] corrupted with zero mean additive Gaussian noise. (b) Registration result for the proposed method. The difference is better highlighted in color. 54

3.12 Curves representing the number of points correctly transformed with respect to a threshold determining the correct transformation using RVM (top row) and TPS (bottom row) when a number of initial false matches is established in source and target sets. A point in the source set is correctly transformed if, after transformation, its distance with respect to its correct counterpart is below the threshold. The left column shows results with false assignments that preserve the one-to-one matching. In that case the RVM provides a consistent behavior and its curves are all at 100% correct transformation. The right column shows results with false assignments that do not preserve the one-to-one matching. 57

3.13 Smoothness (3.4) of the RVM (top row) and TPS (bottom row) under various number of false matches. The left column shows results with false assignments that preserve the one-to-one matching. Notice that the scale of vertical axis at the top-left plot is 10^{-5} indicating a very smooth transformation. The right column show results with false assignments that do not preserve the one-to-one matching. 58

4.1 A univariate Student's t-distribution ($\mu = 0$, $\sigma = 1$) for various degrees of freedom. As $\nu \rightarrow \infty$ the distribution tends to a Gaussian. For small values of ν the distribution has heavier tails than a Gaussian. 65

4.2 A 2D point set and the obtained models (a) GMM and (b) SMM. 68

4.3 The point set of figure 4.2 with 5% outliers and the obtained models by (a) GMM and (b) SMM. Notice that the GMM solution is affected by the outliers while the SMM is more robust. 69

4.4 (a) A three-class piecewise constant image with intensity values 30, 125 and 220, and (b) its negative image (corresponding values, 225, 130 and 35). (c) The image in (a) degraded by uniform noise at $14\,dB$. This image was then registered to the image in (b). The bottom line shows the registration errors for the compared methods. The ground truth solution is 0 deg for the rotation and zero translation (the original image). (d) MI, (e) GMM, (f) SMM. The errors present the difference between the noise free registered image and the reference image. the values are scaled for better visualization. 70

4.5 Mean registration error versus signal to noise ratio (SNR) for the 3-class registration experiment of figure 4.4. 71

4.6 The objective function in eq. (4.2) for the registration of the image of figure 4.4(a) with its counterpart rotated by 20 degrees and translated by 10 pixels. 71

4.7 A pair of NIH 3T3 electron microscope images (400x magnification) of rat cells under (a) normal and (b) fluorescent light. 72

4.8 (a) Image of Europe on 8 January 2007 at 01h00, provided by MeteoSat. (b) Image of Europe on 9 January 2007 at 01h00, provided by MeteoSat (by courtesy of Meteo-France). Notice the large amount of outliers (cloudy regions in different locations in the image pair) introducing important difficulties in the registration process. 73

4.9 Example of a set of points used in the experiments. (a) A point set (presented by dots) was generated by 3 Gaussians with means $(-16, 9)$, $(0, 5)$, $(18, 9)$ and spherical covariance matrices of standard deviation 2. The points were corrupted with 9% outliers. The resulting modeling of the noisy set by (b) a 3-component GMM, (c) a 4-component GMM with the fourth component modeling the distribution of outliers and (d) a 3-component SMM. 74

4.10 Registration error as a function of outliers for the experiment presented in figure 4.9. 75

4.11 Modeling of a *shaped* point set from the GatorBait100 [2] data base by (a) GMM with $K = 30$ components and (b) SMM with $K = 30$ components. Notice that the two models provided similar solutions. The bottom row shows the modeling of the point set with 20% missing points and 10% outliers by (c) GMM and (d) SMM. Notice that the solution of the SMM was less affected. In all cases the mixtures were similarly initialized using the K-means algorithm. The axes in (c) and (d) are normalized to the range of the outliers. 76

List of Tables

1.1 Short description of the databases used in our experiments. 7
1.2 Modeling Error Δ (1.1) 8
1.3 Model Complexity MC (1.5) 9

2.1 Algorithm Performance 18
2.2 Number of two successive frames where the distance of the detected VP in the two frames is less or equal to a threshold 21
2.3 Hausdorff distance between the original and the sampled sets using different sampling methods. 26
2.4 Bull's Eye Rates for the retrieval of sampled sets using different sampling methods. 27
2.5 Experimental results for the Gatorbait dataset [2] (38 shapes). 33
2.6 Experimental results for the MPEG7 dataset [1] (1400 shapes). 34
2.7 Experimental results for the VOP of figure 2.11. 36
2.8 Statistics on the Hausdorff distance (2.16) on the 38 shapes of the GatorBait100 data set [2] 46
2.9 Statistics on the Hausdorff distance (2.16) on the experiments based on the test image of figure 2.20(a). 48

3.1 Registration error statistics for rigid transformations using different kernels on the shapes of figure 3.2. The kernel width varies between 5% and 30% of the mean variance of the reference set. 48
3.2 Registration error statistics for non rigid transformations using different kernels on the shapes of figure 3.2. The kernel width varies between 5% and 30% of the mean variance of the reference set. 48
3.3 Mean registration error for rigid transformations. 52
3.4 Mean registration error for non-rigid transformations. 52
3.5 Mean execution time (sec) of the compared methods for the whole set of experiments presented in section 3.3. The Hungarian-RVM is partially implemented in Matlab (RVM training) and C (Hungarian algorithm). RPM is totally implemented in Matlab while both CPD and GMMReg are totally implemented in C. 52
3.6 Average number of iterations of the compared methods for the whole set of experiments presented in section 3.3. 53

3.7 Mean registration error for rigid transformations in presence of noise. . . . 55
3.8 Mean registration error for non-rigid transformations in presence of noise. . 55
3.9 Registration error statistics for non-rigid transformations. 55
3.10 Registration error statistics for non-rigid transformations. 56

4.1 *Statistics on the registration errors for the images in fig. 4.7 with varying number of mixture components. The errors are expressed in pixels.* 72
4.2 *Statistics on the registration errors for the images in fig. 4.8 with varying number of mixture components. The errors are expressed in pixels.* 73
4.3 Registration errors for the *shaped* point set of figure 4.11 when it is corrupted by 15% outliers. 76

Algorithm Index

1 Direct Split-and-Merge Algorithm 6

2 Vanishing point detection algorithm 18

3 Shape reconstruction from a 2D point cloud 24

4 Image compression 31

5 Image decompression 32

6 Rules for vessel features characterization 39

7 Outlier elimination based on the Helmholtz principle. 44

8 The RVM-Hungarian method for registration of sets of points 47

9 The Hungarian algorithm for square cost matrices 94

10 The Hungarian algorithm for rectangular cost matrices (unbalanced problems) 95

Glossary

Brown A dataset used in experimental evaluation. It contains 137 object silhouettes in total, belonging to 13 different categories.

Disconnectivity The disconnectivity of two sets of points X, Y is the smallest distance between a point in X and a point in Y. It is used in the DSaM algorithm.

DSaM Direct Split and Merge method for line segment detection.

EM Expectation-Maximization algorithm. A framework that by optimizing the likelihood extracts the parameters of a model. In our work we used the EM algorithm to train a GMM/SMM.

ETHZ A dataset used in experimental evaluation. It contains 257 real images depicting scenes of 5 categories (Giraffe, Cup, Swan, Apple Logo and Bottle).

Gatorbait100 A dataset used in experimental evaluation. It contains 38 fish silhouettes in total, belonging to 8 different categories.

Linearity The linearity is a measure that describes how close the points are to a straight line. It is used in the DSaM algorithm.

MPEG7 A dataset used in experimental evaluation. It contains 1400 object silhouettes in total, belonging to 70 different categories.

VP The Vanishing Point is the point at which the parallel lines of a 3D real world image are intersected after projecting them onto the 2D plane of an image.

Abstract

Gerogiannis, Demetrios, P. PhD, Department of Computer Science and Engineering, University of Ioannina, Greece. December, 2014. Feature Extraction for Image and Point Set Analysis. Thesis Supervisor: Christophoros Nikou.

This thesis is divided into two parts. The first part focuses on an algorithm that fits line segments to a set of unordered points and its application to computer vision problems. The method is based on the observation that a set of collinear points are characterized by a covariance matrix whose minimum eigenvalue is low and therefore defines an eccentric (elongated) ellipse. At first, a single ellipse is fitted to the whole set of points which is then iteratively split to a large number of highly eccentric ellipses. Then, a merge process follows in order to combine neighboring ellipses with almost collinear major axes to reduce the complexity of the model. Experimental results on various databases show that the proposed scheme is an efficient technique for modeling unordered sets of points and shapes by line segments. A number of computer vision application of the method are also presented: the localization of the vanishing point in an image sequence, the detection of retinal fundus image features, such as end-points, junctions, and crossovers, an algorithm for sampling image edges and a framework for modeling and removing outliers from a set of unordered points. All of the above methods were successfully compared to various alternative methods of the related literature and provided in general better results.

The second part of the thesis focuses on the problem of image and point set registration. Registration is the process of determining the parameters of a geometric transformation that brings into alignment two images or point sets. In this work, the images/point sets to be registered are modeled by a mixture model and a method relying on the minimization of the distance between distributions is proposed. We address the problems of single and multimodal registration by employing both Gaussian mixture models and mixtures of Student's -t distributions, which are robust to outliers. Moreover, we express the task of registration as a Bayesian regression problem with by modeling the non rigid transformation by relevance vector machines which provide a closed form solution for the estimation of the transformation. An iterative algorithm is presented which first determines the correspondence between pixels/points in the two data images/points sets and then the non rigid transformation is estimated based on that data association.

ΕΚΤΕΤΑΜΕΝΗ ΠΕΡΙΛΗΨΗ ΣΤΑ ΕΛΛΗΝΙΚΑ

Δημήτριος Γερογιάννης του Παναγιώτη και της Αλεξάνδρας. PhD, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Δεκέμβριος, 2014. Εξαγωγή Χαρακτηριστικών για Ανάλυση Εικόνων και Σημείων. Επιβλέποντας: Χριστόφορος Νίκου.

Η παρούσα διατριβή αποτελείται από δύο θεματικές ενότητες. Στην πρώτη ενότητα παρουσιάζεται μία μέθοδος μοντελοποίησης ενός συνόλου μη διατεταγμένων σημείων από ένα σύνολο ευθυγράμμων τμημάτων και η εφαρμογή της σε διάφορα προβλήματα υπολογιστικής όρασης. Η μέθοδος βασίζεται στην παρατήρηση ότι ένα σύνολο συνευθειακών σημείων χαρακτηρίζεται από έναν πίνακα συμμεταβλητότητας του οποίου η ελάχιστη ιδιοτιμή έχει πολύ μικρή τιμή και ορίζει μία έλλειψη με μεγάλη εκκεντρότητα. Αρχικά, το σύνολο των σημείων προσεγγίζεται από μία έλλειψη η οποία στη συνέχεια διαχωρίζεται επαναληπτικά σε περισσότερες ελλείψεις ώστε το σύνολο των σημείων να προσεγγιστεί από έναν αριθμό έκκεντρων ελλείψεων. Στη συνέχεια, λαμβάνει χώρα μία διαδικασία συγχώνευσης των ελλείψεων που έχουν συγγραμικούς μέγιστους άξονες για να μειωθεί η πολυπλοκότητα του μοντέλου. Πειραματικά αποτελέσματα δείχνουν την αποτελεσματικότητα της μεθόδου να συμπιέζει την πληροφορία μη δομημένων συνόλων σημείων αλλά και σχημάτων. Επίσης, παρουσιάζεται η εφαρμογή της μεθόδου στον εντοπισμό του σημείου διαφυγής σε εικονοσειρές, στον εντοπισμό και χαρακτηρισμό εικόνων του βυθού του αμφιβληστροειδούς χιτώνα του οφθαλμού, στη δειγματοληψία χαρτών ακμών από 2Δ εικόνες καθώς και στην εξάλειψη του θορύβου και ακραίων μετρήσεων σε 2Δ σύνολα σημείων. Όλες αυτές οι μέθοδοι συγκρίνονται επιτυχώς με μεθόδους της βιβλιογραφίας.

Το δεύτερο μέρος της διατριβής εστιάζει στο πρόβλημα της υπέρθεσης εικόνων και συνόλων σημείων. Υπέρθεση είναι η διαδικασία της εκτίμησης του γεωμετρικού μετασχηματισμού που φέρνει σε αντιστοιχία δύο σύνολα σημείων ή εικόνες. Στην εργασία αυτή, οι εικόνες/σύνολα σημείων μοντελοποιούνται από μικτές κατανομές και η υπέρθεση επιτυγχάνεται με την ελαχιστοποίηση της απόστασης μεταξύ των κατανομών. Προτείνεται η μοντελοποίηση των δεδομένων με μικτές κανονικές κατανομές όσο και από μικτές κατανομές Student's *t* οι οποίες είναι εύρωστες σε δεδομένα που δεν ακολουθούν το κυρίαρχο μοντέλο.

Επίσης, η διαδικασία της υπέρθεσης περιγράφεται ως ένα πρόβλημα Μπεϋζιανής παλινδρόμησης με τη μοντελοποίηση του μετασχηματισμού από μηχανές διανυσμάτων συνάφειας (RVM) τα οποία παρέχουν μία κλειστής μορφής λύση για το γεωμετρικό μετασχηματισμό. Στο πλαίσιο αυτό παρουσιάζεται ένας επαναληπτικός αλγόριθμος που εκτελεί ένα βήμα αντιστοίχισης μεταξύ των εικονοστοιχείων/σημείων και στη συνέχεια με βάση αυτή την αντιστοίχιση εκτιμάει τον ελαστικό γεωμετρικό μετασχηματισμό που συνδέει τα δύο σύνολα.

Prologue

0.1 Overview

0.2 Structure of the thesis

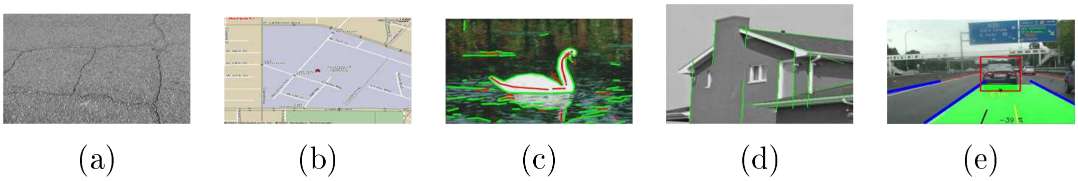
0.1 Overview

The field of computer vision has been advancing during the last years, benefited from the development of the technology and the available computational resources. Many methods have been proposed in a high level to deal with the difficult problem of simulating human perception. A common characteristic of all these methods is that they are based on preliminary feature extraction techniques to derive meaningful information from images for further postprocessing.

Features are very basic entities that carry information related to a specific problem. Computing features is performed via various algorithms and the process is called feature extraction and their representation may vary. A principal characteristic is that feature models tend to be as simple as possible. Lines and line segments are widely used in the computer vision literature as feature representation models. They present low complexity and their aggregation can produce more complex models enabling the accurate representation of more complex structures in an image. Since the early stages of the development of the computer science fields the interest was focused on the extraction of lines and line segments on a set of points, that in many cases, is derived from the edges of an image. The pioneering Hough Transform became the basis upon which many variants were based and a numerous of applications used them as a preprocessing step.

The fact that a majority of structures depicted in images (e.g. buildings, furniture, cars, human bodies, trees, etc.) can be decomposed into a set of lines, and more specifically line segments, makes the latter an important feature to recognize in images. Figure 1 depicts some representative examples of images were line segments could be used to describe the image content. The typical Hough Transform is only capable of computing lines, while its variants that produce line segments demand a lot of effort, as they are based on trial and error, to adjust the corresponding thresholds. In addition, recently proposed methods may solve the 2D problem, but their generalization to more dimensions, i.e.

extraction of planes, is not trivial nor can they deal with point coordinates. Alternatively, they operate directly on images and thus, they cannot handle data collected via other methods, e.g. range data. All these factors motivated us to study the problem of line segment detection. Moreover, taking into account that in most of the methods, threshold values tuning has not been thoroughly studied, we take care for setting the values of the various thresholds used by our line segment detection method. In brief, the main problem we wish to tackle in the first part of this dissertation, may be concluded to the following: given a set of unordered points $X = \{\mathbf{x}_i \in \mathbb{R}^2 | i = 1, \ldots, N\}$ find the set of line segments $E = \{\epsilon_j | j = 1, \ldots, K\}$ be modeling accurately the points, where ϵ_i is the line describing the i-th segment, while the number K of the line segments is unknown. In addition, provide a method for automatic tuning of any parameters of the line segment method.



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

Figure 1: Example of various images where line segments could be used to describe the depicted information: (a) road cracks, (b) maps, (c) object edges, (d) building edges, (e) road lanes.

As soon as the line segments are extracted, various applications can benefit from the established model. The following list describes in brief the applications that were studied in this dissertation. The reader is referred to the related section for more details.

- The detection of a vanishing point in an image depicting structured environments, i.e. places were the edges between the various regions of the image are clearly established (e.g. the edge line between a wall and the floor). Some representative examples of the aforementioned structured environments are shown in figure 2, both for indoor and outdoor scenes. The vanishing point is the point of the image space where parallel lines of the real world intersect after projecting them to the image space. This point is an important feature that can be used for posterior analysis of the image (e.g. extraction of the road plane, the walls, etc.) or autonomous navigation.

- The sampling of point clouds in order to provide a new set with fewer points, preserving the initial information. Sampling is an important preprocessing step in many computer vision algorithms, because the latter present high complexity and thus, their efficient execution is related to the number of the observations they are parsing. A similar problem is the reconstruction of a shape, based on some characteristic points. In brief, given a shape (i.e. a set of 2D points) it is asked to detect those characteristic points from the initial set that summarize the shape and enables the reconstruction of the initial set of points (i.e. shape) with as low distortion as possible. Figure 3 demonstrates a shape reconstruction example. In

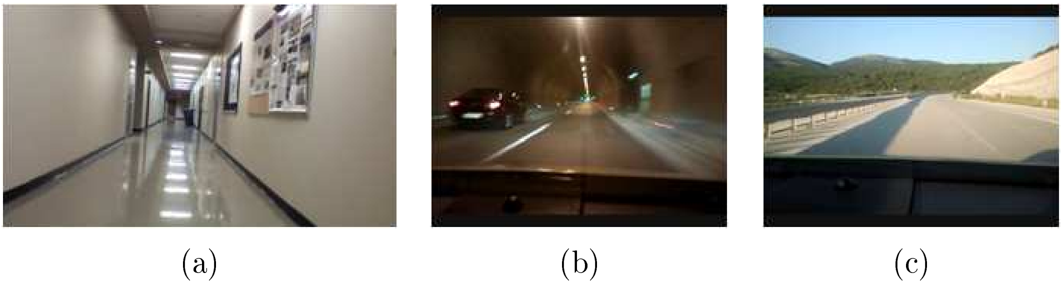


Figure 2: Example of images depicting structured worlds. (a) Indoor scene (b),(c) Outdoor scenes.

figure 3(a) the initial set is demonstrated, while figure 3(b) depicts the extracted characteristic points (green stars). In figure 3(c) the reconstruction result is shown (blue points) superpositioned over the initial shape (red points). Notice the small deviation between real and computed data.

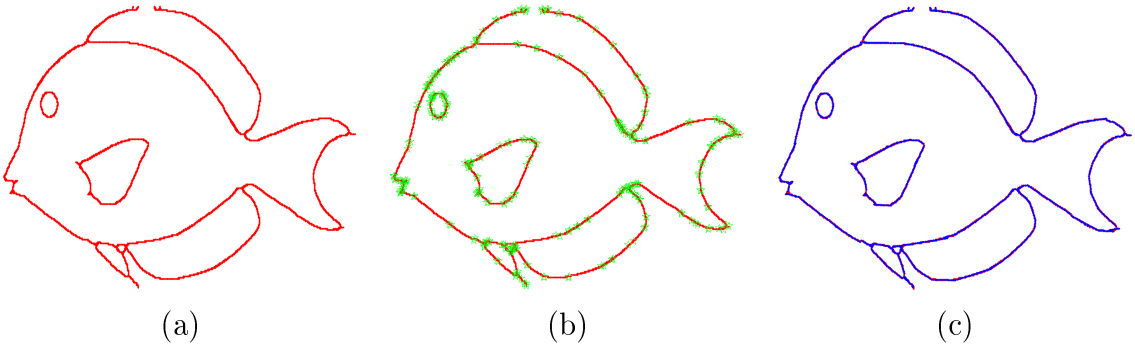


Figure 3: Example of shape reconstruction. (a) The initial set of points describing a shape, (b) The characteristic points (green stars) are extracted from the initial shape (red points), (c) The reconstruction result (blue points) superpositioned over the initial shape (red points). Notice the small deviation between real and computed data.

- The compression of bilevel images that depict the edge map of a real image. More precisely, we dealt with the problem of encoding binary images that depict the contour of various shapes. This type of images is mainly used to describe objects in the MPEG4 standard, in terms of video encoding. Thus, it is plausible to encode individual objects in a video frame, a fact that provides freedom to the end user, regarding the presentation options. The efficiency of the compression method is dictated by the achieved compression rate with respect to distortion.

- The characterization of a retinal fundus image. The tree structure of the veins in a retinal fundus image can be encoded with line segments. Then it is easy to detect the intersection points of the various veins and proceed to a post processing algorithm that analyzes this special points.

- A method for extracting meaningful structures in presence of outliers. In general, an outlier is considered every point that does not obey the general model of the real data. In other words, as outliers can be considered all those points that are structureless, provided that a valid model that describes the structured data is established. That model is a set of line segments in our work.

On the other hand, we dealt also with the problem of image and point set registration. Registration is a very common problem and in many cases it is a preprocessing step for other methods, e.g. the automatic evaluation of the development of a patient's condition based on the observation of some time varying medical images. In general, registration relies on the determination of that particular geometric transformation parameter values, that upon being applied to one image/set of points it will bring it into alignment with a reference image/set of points. Figure 4 explains the registration problem. The goal is to determine that geometric transformation that will be applied on the left image of figure 4 (yellow background) and will align the pixels such that the pixels of the blue circle in the left will be matched with those of the green circle in the right and the pixels of the blue ellipse in the left will be corresponded with those of the green ellipse in the right image. A challenging problem, which motivated us to deal with registration, occurs when the two images to be registered are of different modalities. A basic observation is that similar structures in the two images have similar probabilistic representation of their intensity. Thus, upon perfect alignment the distance between those distributions will be small. To that end, a mixture of Gaussian distributions was employed and in order to handle the presence of outliers we extended the model with Student's-t distributions.

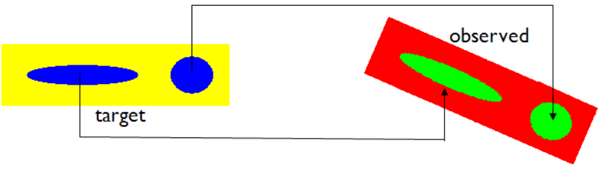


Figure 4: Explanation of the registration problem. The goal is to determine that geometric transformation that will be applied on the left image (yellow background) and will align the pixels such that the pixels of the blue circle in the left will be matched with those of the green circle in the right and the pixels of the blue ellipse in the left will be corresponded with those of the green ellipse in the right image.

A basic factor that affects the final registration result is the model that is selected to describe the registration transformation. In case of rigid transformations (i.e. rotations and translations) the model is trivial. However, this is not true when a non-rigid transformation is taken into consideration, where more complicated models need to be considered. In that framework, we employed a Bayesian framework, the Relevance Vector Machines, to provide a more robust model that can handle false matches and prevent the transformation from global failing, by reducing the impact of a false match in a local region. Moreover, this approach provides a closed formula for modeling the registration transformation.

The contribution of this thesis can be summarized into the following:

- An iterative framework for line segment detection to summarize unordered point sets.
- A voting scheme for the detection of vanishing points in structured images.
- A method for efficiently annotating retinal fundus images.
- A method for efficiently sampling unordered 2D points.
- A comparative study between line segment extraction methods for bilevel image compression.
- A method for extracting structures (e.g. shapes) in presence of outliers.
- A Bayesian approach for modeling a non-rigid registration transformation which is robust to false matches.
- An algorithm for registering multimodal images and cloud of points.

0.2 Structure of the thesis

The first part of this thesis deals with line segment extraction from a set of unordered points and applications. The second part presents our work in the field of image and point set registration.

In **Chapter 1**, we introduce an iterative method for the extraction of line segments. A short introduction of the related literature is provided and the proposed algorithm is described in detail. Finally, an extensive experimental evaluation is provided comparing our method with other commonly used approaches.

In **Chapter 2**, some applications based on line segment detection are introduced. A short introduction is presented for each application, to describe the problem and the various solutions provided in the related literature. Then, the proposed method is presented along with an experimental evaluation and comparison with the state-of-the-art. Thus, **section 2.1** deals with the detection of the vanishing point in structured images, **section 2.2** presents an efficient algorithm for sampling unordered points, in **section 2.3** a method for shape encoding and bilevel image compression is presented, in **section 2.4** a method for characterizing a retinal fundus image is demonstrated, and finally, in **section 2.5** an algorithm for extracting structured information (e.g. shapes) in presence of outliers is introduced.

In **Chapter 3**, the modeling of a non rigid transformation for point set registration is presented. The algorithm is described in detail and various experimental results are demonstrated.

In **Chapter 4**, we describe a solution of the rigid registration problem based on mixture models.

Part I

Features and Applications

Chapter 1

Modeling sets of unordered points using line segments

1.1 Introduction

1.2 A Direct Split and Merge (DSaM) Framework for Line Segment Detection

1.2.1 Split process

1.2.2 Merge process

1.3 Evaluation of the Line Segment Detection Algorithm

1.3.1 Numerical Evaluation

1.3.2 Comparison with the Hough Transform

1.1 Introduction

Lines are one of the most basic models to describe features in an image due to their simplicity, regarding the modeling parameters. Moreover, lines are suitable models for describing real world structures as most of the human made scenes are being represented by flat surfaces. Lines can be used to summarize features in a higher level, e.g. contours.

Examples regarding the importance of line extraction include the detection of vanishing points [18], the vectorization of raster images [19] and the detection of road structures and parts [20] are among applications necessitating line segment description of image structures. In many of the aforementioned problems, the involved algorithms assume that they are provided with an ordered point set and standard polygonal approximation [10, 21] is then applied. However, determining the ordering of point sets is not a trivial task and in the method described herein we relax this assumption by making no prior hypothesis about the ordering of the points.

In the above context, the Hough transform (HT) is a widely used method for line fitting and many variants have been proposed to improve its efficiency [22, 23]. One of these variants is the randomized Hough transform (RHT) [24, 25] which randomly selects a number of pixels from the input image and maps them into one point in the parameter space which was shown to be less complex, compared to the original algorithm, as far as time and storage issues are concerned. In [26], the probabilistic HT was proposed whose basic idea is to apply a random sampling of edge points to reduce computational complexity and execution time. Further improvements were introduced in [27]. A similar concept was proposed in [28], where an orientation-based strategy was adopted to filter out inappropriate edge pixels, before performing the standard HT line detection which improves the randomized detection process. Also, the idea of fuzziness is integrated in the main algorithm in [29] to model the uncertainty imposed to the contour due to noise. Thus, a point can contribute to more than one bin in the standard HT process. A general comparison between probabilistic and non-probabilistic HT variants can be found in [30].

The robust HT is introduced in [31] where both the length and the end points of the lines may be computed. Moreover, the algorithm in [32] provides a method for adopting a shape dependent voting scheme for the calculation of the histogram bins. Finally, a novel HT based on the eliminating particle swarm optimization (EPSO) is proposed in [33], to improve the execution time of the algorithm. The problem parameters are considered to be the particle positions and the EPSO algorithm searches the optimum solution by eliminating the "weakest" particles, to speed up the process.

Line segment fitting may also be used in a shape description process. The commonly used algorithm of Moore [34] was a first solution to shape following and utilizes the neighborhood of points. However, this algorithm is appropriate only for traversing curves without intersections and produces models with high complexity, although improvements of the main algorithm have also been considered up to date [35]. Another common model fitting method is the RANSAC algorithm [36], which despite the fact that it provides robust estimations, it is appropriate for fitting only one model at a time. Other approaches are the incremental line fitting [37] which is sensitive to noise and, most importantly, needs sequential ordering of the points and probabilistic methods [38] based on the Expectation-Maximization algorithm, generally necessitating the prior determination of the number of model components.

More recently a new method was introduced that relies on the Helmholtz principle: 'no structure is perceived in white noise', based on the work of [39] for adaptive thresholding. Its main characteristic, according to the authors, is that this method is parameterless and can accurately control the false positive and false negative detections. In brief, initially the image gradient is computed at each pixel and then through a region growing algorithm they try to align points whose gradient direction is within a predefined threshold. Although that there is a threshold parameter, the authors claim that their method is nearly parameterless because the decision threshold on the number of control points in a given segment is in a $\sqrt(\log)$ dependency of the expected number of false alarms. The reader is refereed to [40, 41] for more details and to [42] for the implementation details of the method.

1.2 A Direct Split and Merge (DSaM) Framework for Line Segment Detection

Let $X = \{\mathbf{x}_i | i = 1, \dots, N\}$ be a set of points and $E = \{\epsilon_j | j = 1, \dots, K\}$ be the set of line segments modeling the points, where ϵ_i is the line describing the i-th segment.

We define the modeling error Δ induced by the representation of line segments:

$$\Delta(X, E) = \sum_{i=1}^{N} \sum_{j=1}^{K} \delta_{ij} d(\mathbf{x}_i, \epsilon_j), \tag{1.1}$$

where K is the number of line segments the model uses to model the points, $\mathbf{x}_i \in \mathbb{R}^2$, $i = 1, \dots, N$ are the points, $d(\mathbf{x}_i, \epsilon_j)$ is the perpendicular distance of point $\mathbf{x}_i$ to line ϵ_j, δ_{ij} is an indicator function whose value is one if point $\mathbf{x}_i$ belongs to line segment ϵ_j and is zero otherwise.

In order to prevent overfitting, models having a large number of line segments should be penalized. Therefore, an optimal model would have both low value of Δ and low complexity.

The computation of the ellipses, modeling the line segments, is performed in two steps: an iterative *split process*, where points are modeled by a number of line segments represented by the major axes of the corresponding ellipses and an iterative *merge process*, where small line segments are merged to reduce the model complexity. The split process tries to minimize the modeling error while the merge process decreases the model complexity, i.e. the number of line segments compared to the total number of points in the set.

In what follows the two steps are presented in detail.

1.2.1 Split Process

The ultimate goal of this step is to cover the point space with line segments representing the long axes of elongated ellipses and therefore, each point of the shape should be assigned to an eccentric ellipse. A split criterion is defined, based on Gestalt theory [43], which models the linearity and the connectivity the human brain uses when modeling contours.

In order to split a set X, it should be either non linear or disconnected, or both. Linearity describes how close the points are to a straight line, while disconnectivity measures how concentrated these points are. In the ideal case, the covariance matrix of collinear $2D$ points should have a very large eigenvalue and a zero eigenvalue. The eigenvector corresponding to the larger eigenvalue indicates the direction of the line segment. If the linearity property is relaxed, the less collinear the points become (i.e. they diverge from the linear assumption) the larger the value of the minimum eigenvalue is. Based on that observation, in our method, linearity is described by the minimum eigenvalue of the covariance matrix of the points in X. Also, the disconnectivity W of two sets of points X, Y is the smallest distance between a point in X and a point in Y:

$$W(X, Y) = \min_{\substack{\mathbf{x} \in X \\ \mathbf{y} \in Y}} |\mathbf{x} - \mathbf{y}|. \tag{1.2}$$

In the case of a single set, disconnectivity is the largest distance between two successive points in that set. It may be computed by projecting the points onto both axes defined by the eigenvectors of the covariance matrix of the set. Then, successive points are defined by scanning along the axes and their distances are computed. Let X_i be the projection of a set X onto the the eigenvector e_i. The disconnectivity of X is defined as

$$W(X) = \max_{\substack{j=1,\ldots,N-1 \\ i=1,\ldots,d}} |\mathbf{x}_i^j - \mathbf{x}_i^{j+1}|, \tag{1.3}$$

where N is the number of points in X, d is the dimension of X (here $d = 2$) and $\mathbf{x}_i^j$ is the j-th point of the sorted set X_i. A large value of disconnectivity indicates a better separation of the point sets. The projections onto all of the eigenvectors should be examined as we do not know *a priori* which direction to follow while splitting. Although intuitively one would suggest to split along the direction of the principal axis, we observed that in many cases that approach was not the best. Also, let us note that as the ordering of the points is not known *a priori*, their projection onto the eigenvectors of their covariance matrix, provides a natural way of ordering.

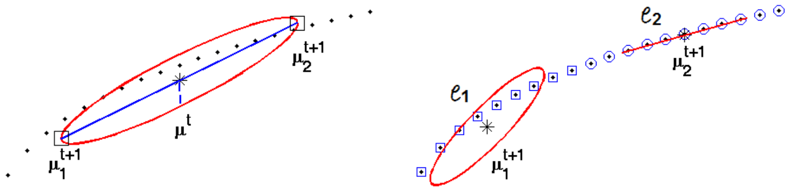
The disconnectivity of a single set of points is also important to be estimated in the split step, as there may exist subsets that although they are linear, they are disconnected.

The split of an ellipse should be performed along the direction defined by an eigenvector of its covariance. In order to select the split direction, the axis corresponding to an eigenvector is considered as the discrimination border between the split line clusters and points belonging to the same subplane are grouped together. Then the disconnectivity of each line cluster is computed. Finally, the direction with the largest disconnectivity is selected for splitting (figure 1.1).

Eventually, the adopted strategy that minimizes Δ and prefers elongated ellipses can be expressed as follows: split every ellipse whose minimum eigenvalue is greater than a threshold T_1 (linearity) and the maximum gap, within the current segment is greater than a threshold T_2 (disconnectivity). The process is initialized with one ellipse, corresponding to the covariance of the initial points set centered at the mean value of the point locations. Thresholds T_1 and T_2 may be computed with a heuristic algorithm, as explained in subsection 1.3.1.

At iteration $t + 1$, a given ellipse, characterized by the eigenvalues λ_1^t and λ_2^t of its covariance matrix Σ^t (with $\lambda_1^t \geq \lambda_2^t$), with center μ^t, is split to two new ellipses with centers the two antipodal points on the major axis:

$$\begin{aligned} \mu_1^{t+1} &= \mu^t + \sqrt{\lambda^t} e^t, \\ \mu_2^{t+1} &= \mu^t - \sqrt{\lambda^t} e^t, \end{aligned} \tag{1.4}$$



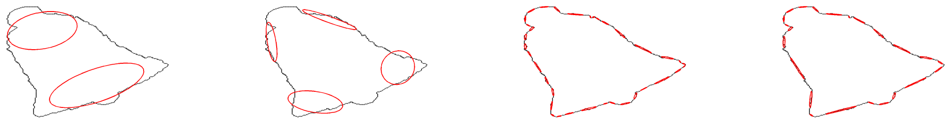
(b) (b)

Figure 1.1: Split process. (a) At iteration $t+1$, the ellipse with center μ^t is split into two new ellipses e_1 and e_2, with centers μ_1^{t+1} and μ_2^{t+1} given by (1.4). (b) The new centers are marked with a star (*). The reassignment of the points to the new centers is shown. Points of one category, assigned to e_1, are marked with a square, while points assigned to e_2, are marked with a circle.

where e^t, λ^t are the eigenvector and the eigenvalue corresponding to the split direction along which split is performed (figure 1.1).

The points of the split ellipse are then reassigned to the two new ellipses according to the nearest neighbor rule. In this way, new ellipses occur, which are more elongated as they have greater eccentricity and their minor axes are closer to the contour (figure 1.2). Moreover, this detailed representation of the point set provides accurate modeling of the joints, corners and parts of the contour exhibiting high curvature.

A variant of the method would be to compute the covariance matrix of the points on the convex hull of the point set, which provide more robustness to outliers.



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

Figure 1.2: Steps of the split and merge process. The process is initialized with the mean and the covariance of the full set of points. (a) Split into 2 ellipses. (b) Split into 4 ellipses. (c) End of split (35 ellipses). (d) The final merge result (23 ellipses). The figure is better seen in color.

1.2.2 Merge Process

The role of the merge process is to reduce the complexity of the model. In case there exist adjacent ellipses whose major axes have similar orientations, it would be beneficial to merge and replace them with a more elongated ellipse. Therefore, in this step, ellipses are merged using the following rule: merge two consecutive ellipses, if the resulting ellipse has minimum eigenvalue smaller than a threshold T_1 (linearity) and the marginal width between the two line clusters is smaller than a threshold T_2 (disconnectivity).

Note that the threshold T_1 could be set equal to the threshold used in the split process, where the value of parameter T_1 specifies whether an ellipse has low eccentricity and needs to be split. In the merge process, it indicates whether two candidates for merging ellipses would result in an ellipse with high eccentricity. One could use the same threshold in both processes, assuming the same significance. On the other hand, a relaxation of the merge threshold could lead to a rougher model of the points, smoothing out details like joints. In our experiments, the merge threshold was selected to be the same with the split threshold. The same applies for threshold T_2 that indicates whether two segments are close enough to be considered as one line segment.

The overall description of the method is presented in Algorithm 1.

```
SPLIT PROCESS
input: The set of points $X = \{\mathbf{x}_i | i = 1, \dots, N\}$.
output: A set of ellipses $\{\mu_j, \Sigma_j\}$.
Initialize the algorithm by estimating the mean and covariance of the point locations.
while there are ellipses to split do
  Split every ellipse whose minor eigenvalue is greater than $T_1$ and its disconnectivity
  is greater than $T_2$.
    • Select the direction that provides the greatest disconnectivity.
    • Set the centers of the new ellipses according to (1.4).

MERGE PROCESS
input: The ellipses from the split process $\epsilon_j = \{\mu_j, \Sigma_j\}, j = 1, \dots, M$.
output: A reduced set of ellipses.
while there are ellipses to merge do
  for all ellipses $\epsilon_i, i = 1, \dots, M$ do
    if merging $\epsilon_i$ with $\epsilon_j$ provides an ellipse whose minor eigenvalue is less than $T_1$
    and its disconnectivity is less than $T_2$ then
      Accept merging.
      Set $\epsilon_i$ to the ellipse that result from merging
```

Algorithm 1: Direct Split-and-Merge Algorithm

1.3 Evaluation of the Line Segment Detection Algorithm

In this section we evaluate the efficiency of the introduced algorithm. To that end, two categories of experiments were conducted. The purpose was to investigate the performance of the method both in shape data, but also in real images. Thus, various well-known databases were employed, that contain either object silhouettes or scenes of real images.

The GatorBait100 database [2] consists of 38 shapes of different fishes grouped in 8 categories. The shapes of this database are not closed and contain many junctions. The MPEG7 shape database [1] consists of 1400 silhouettes of various objects clustered in 70 categories.The shape silhouette database used in [3], that contains 137 silhouettes of various objects, clustered in 13 categories, was also used in our experiments. Finally, to investigate the behavior of the proposed algorithm in real scene images, the images (257) from the ETHZ image set [4] were also used. Table 1.1 gives a brief description of each database. In all cases, the edges were extracted and the coordinates of the edge pixels were used to describe the contour. The Canny edge detector [44] was used in all cases.

Table 1.1: Short description of the databases used in our experiments.

Database	**# categories**	**# shapes/scenes**	**Description**
GatorBait100 [2]	8	38	Fish silhouettes
MPEG7 [1]	70	1400	Object silhouettes
Brown [3]	13	137	Object silhouettes
ETHZ [4]	5	257	Real Scene Images

1.3.1 Numerical evaluation

In this section, we present the results [1] of comparing the DSaM method with the widely used implementation of Kovesi [5]. This is an implementation of the polygon approximation [10] method. The algorithm assumes the traversal of the points is known. Initially, it selects an arbitrary point and starts traversing the shape. A line segment is computed by all points that have been visited so far, and the processes iterates for all points in the traversal order. Then, the modeling error is computed, in terms of deviation of points from the current line segment. If the deviation after a point is used top compute the line segment is larger than a threshold, this point is considered as the starting point of a new line segment. The process terminates when all points have been visited.

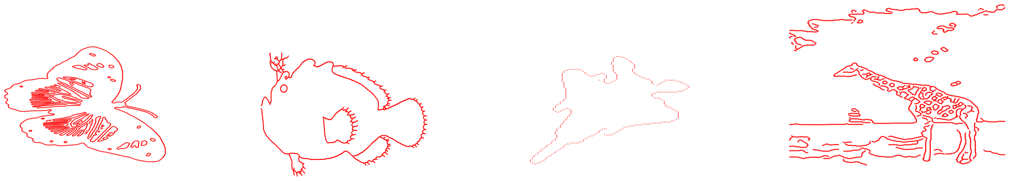
Tables 1.2 and 1.3 summarize the numerical results. Some representative images from those databases are given in figure 1.3. As it can be observed, in some cases, there exist inner structures and thus, the ordering of the points is not obvious. Note that to share common parameters, in the Kovesi [5] implementation, we used the disconnectivity threshold of our method. The execution time for computing that value, is not included in the execution time of the Kovesi implementation. The model complexity is computed by the index:

$$MC = \frac{\#\text{ellipses}}{\#\text{points}}. \tag{1.5}$$

Lower values of MC imply lower complexity and therefore a more compact representation.

The distortion, is the measure of the quality of the fitting, and is computed as the average distance between a point and its corresponding line segment, as computed by

[1] Matlab code available at http://www.cs.uoi.gr/∼dgerogia



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

Figure 1.3: Some representative images of the databases we used in our experiments. Please note that in some cases inner structures exist. This does not permit to extract an ordering of the points (a)MPEG7 [1], (b) Gatorbait [2], (c) Brown [3], (d) ETHZ [4].

Table 1.2: Modeling Error Δ (1.1)

MPEG7 [1] (70 shapes)					
method	**mean**	**std**	**median**	**min**	**max**
DSaM	0.489	0.093	0.509	0.080	0.773
Kovesi	2.796	3.977	1.736	0.533	46.984
GatorBait100 [2] (38 shapes)					
method	**mean**	**std**	**median**	**min**	**max**
DSaM	0.454	0.033	0.452	0.383	0.509
Kovesi	2.215	0.862	1.981	1.477	6.473
Brown [3] (137 shapes)					
method	**mean**	**std**	**median**	**min**	**max**
DSaM	0.492	0.119	0.514	0.105	0.894
Kovesi	2.871	6.192	1.095	0.617	33.632
ETHZ [4] (255 scenes)					
method	**mean**	**std**	**median**	**min**	**max**
DSaM	0.494	0.061	0.503	0.257	0.635
Kovesi	2.299	1.340	1.914	1.056	12.655

Table 1.3: Model Complexity MC (1.5)

MPEG7 [1] (70 shapes)

method	**mean**	**std**	**median**	**min**	**max**
DSaM	3.954%	0.013%	3.788%	0.269%	11.429%
Kovesi	3.624%	0.017%	3.406%	0.269%	12.442%

GatorBait100 [2] (38 shapes)

method	**mean**	**std**	**median**	**min**	**max**
DSaM	3.280%	0.004%	3.172%	2.732%	4.541%
Kovesi	2.524%	0.005%	2.378%	1.961%	3.904%

Brown [3] (7 scenes)

method	**mean**	**std**	**median**	**min**	**max**
DSaM	5.792%	0.016%	6.186%	0.921%	10.145%
Kovesi	6.586%	0.022%	6.911%	0.335%	10.821%

ETHZ [4] (16 scenes)

method	**mean**	**std**	**median**	**min**	**max**
DSaM	5.342%	0.014%	5.195%	2.436%	8.427%
Kovesi	5.402%	0.018%	5.205%	1.601%	11.040%

each method. Please note that the average length of the diagonal of the bounding box of the various datasets, is about 500 units (ranging from 300 units in Brown [3] to 700 units in ETHZ [4]).

In the proposed algorithm, there are two parameters to be *a priori* specified, a threshold that determines the elongation of an ellipse (T_1) and a threshold characterizing the disconnectivity of a set (T_2). Both parameters are used to decide whether to split (in the split process) or merge (in the merge process). A small value preserves the details, while a larger one provides more coarse results. For our experiments, we computed the values of the parameters as:

$$T_1 = \frac{1}{N}\sum_{i=1}^{N} \lambda_i \tag{1.6}$$

$$T_2 = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{1.7}$$

where N is the number of the points of the set, λ_i is the smallest eigenvalue of the covariance matrix of points $\{\mathbf{x}\,|\,\mathbf{x} \in N^{\alpha}_{\mathbf{x}_i},\, i = 1, \ldots, N\}$, with $N^{\alpha}_{\mathbf{x}}$ being the α- neighborhood of $\mathbf{x}$, and

$$d_i = \min_{\mathbf{y}\in N^{\alpha}_{\mathbf{x}_i}} ||\mathbf{x}_i - \mathbf{y}||,\, i = 1, \ldots, N. \tag{1.8}$$

Large values for α decrease the model complexity providing larger modeling error and details are not preserved. In our experiments, we set $\alpha = \lceil 0.01 \times N \rceil$ for computing the values of T_1 and T_2. To make our implementation more efficient, instead of taking all points into consideration, we computed a random permutation of the indices of points and used only the first 10% of them. Thus, in high density datasets, like in the ETHZ database [4], the values of the thresholds could be estimated quickly.

In general, the DSaM method and the Kovesi implementation produce models with similar complexity, a fact that is obvious, since they employ the same thresholds. However, the DSaM method provides much more accurate results w.r.t distortion (Table 1.2 and Table 1.3).

Figure 1.4 explains the meaning of the modeling error. Larger modeling error is associated with greater deviations of the computed model from the shape contour. The reader may observe that the proposed method provides a more accurate result, compared to the computation with the Kovesi [5] method.

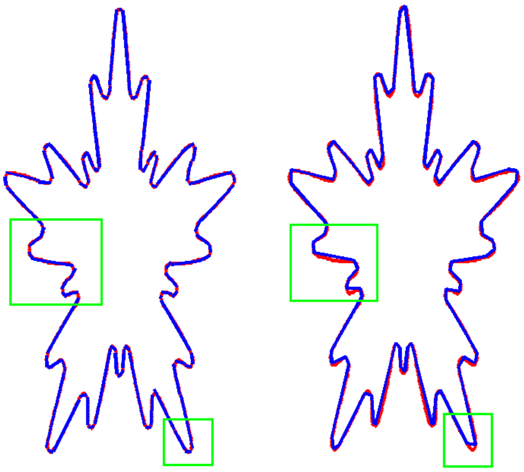
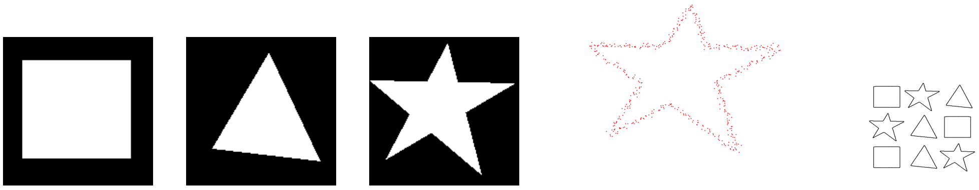


Figure 1.4: A representative result of the modeling of a shape from the MPEG7 dataset [1]with the proposed (left) and Kovesi [5] (right) methods. Green boxes highlight the differences regarding the modeling error of the two methods. Although in general both methods modeled the shape globally, locally the proposed method modeled more accurately the shape contour.

As our method models the line segments with ellipses, we tried to fit line segments by exploiting various modifications of a typical Gaussian Mixture Model (GMM) [45], for example using an incremental GMM, or imposing constraints in the update step of the covariance matrices (decomposing the covariance matrices with SVD, replacing the corresponding minimum eigenvalue with a very small value, threshold T_1, and then recomputing the covariance matrix). All these variants failed to produce an efficient result. Moreover, the execution time was quite high (10 times compared to those of DSaM). Thus, we opted for excluding the results from this presentation.

Finally, we conducted experiments to verify the robustness of the method against the presence of noise that degrades the contour of an object. To that end, we used three patterns (see figure 1.5 (a)-(c)) which were randomly repeated, to create new images. A representative image is given in figure 1.5 (e). As a ground truth for the number of line segments, we used 4 for the square, 3 for triangle and 10 for the star.

The set of unordered points was produced by simple edge detection. Then, zero



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

Figure 1.5: (a) - (c) The primitive images used to create the artificial dataset for experiments with Gaussian additive noise. (d) Contour degraded by additive Gaussian noise of 18dB. A representative test image produced by randomly repeating the patterns of images in (a)-(c).

mean Gaussian noise with varying standard deviation was added in order to get several configurations of signal-to-noise ratio (SNR). A representative result of a degraded contour is given in figure 1.5 (d). Note that no ordering of points may be established in that case and thus polygon approximation may not be performed. To make the experiment independent from the noise configuration each experiment was repeated 20 times. The algorithm assumes that a form of binary data (e.g. an edge map) is provided. Degradation by noise is performed after the edge extraction in order to examine the behavior of the algorithm to the detection of line segments. If the noise was added to the original image the edges would be erroneous and we would not have a standard baseline for evaluating the algorithm.

In figure 1.6, we present the results of the experiments. The error is expressed as the absolute difference between the real number of segments and the one computed by our method. It can be observed that while the magnitude of the noise decreases, the error is also decreased. The difference between true and estimated number of segments is generally small, 3 on average with low variance (±2 segments), compared to the total number of line segments, 90 line segments on average, corresponding to 3% deviation between true and estimated measurement. Thus, it could be claimed that the proposed method exhibits a consistent and efficient performance even if the contour is corrupted.

1.3.2 Comparison with the Hough Transform

Since the proposed algorithm fits line segments to a set of points, we also tested it against the commonly used Hough Transform (HT). However, since the standard HT is appropriate for fitting lines and not line segments, we applied the Progressive Probabilistic Hough Transform (PPHT), as proposed in [46] and implemented in the OpenCV library [47]. The implementation of PPHT imposes three parameters: (i) a threshold, indicating the minimum number of points in a bin, in the line parameter space, in order to consider that the line is represented by a sufficient number of points, (ii) the minimum length of a line segment and (iii) the maximum gap between line segments lying on the same axis. In our experiments, we fixed the last two parameters (after a trial and error procedure

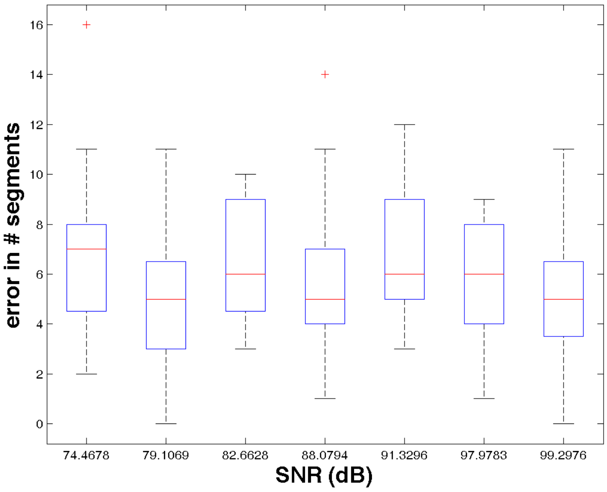


Figure 1.6: Experimental results using the datasets of figure 1.5 (e) that demonstrate the performance of our method in presence of Gaussian additive noise in terms of model complexity error. The vertical axis represents the absolute error between the real number of segments and the one computed by our method.

keeping those parameters that best fit the examined points) and varied the threshold. The obtained results for the PPHT exhibited significant irregularities such as a large number of overlapping lines for the same segment. Also, the corners of the shapes were not correctly captured. Representative experiments on the MPEG7 dataset [1] are shown in figure 1.7(a)-(c) while the solution of our DSaM algorithm is illustrated in 1.7(d).

The PPHT is based on a histogram which correlates the accuracy of the result with the number of bins used. Also, a threshold must be established to eliminate lines with small participation in the final result. A small number of bins may lead to an underestimation of the number of segments, while a large number of bins increases the complexity of the model. As far as the threshold is concerned, its value may have similar effects in the final model. A large value may drop some segments, while a small value may be responsible for a large number of lines fitted, analogous to a GMM with one component per point. A more important drawback of the PPHT is that many overlapping lines may model the same line segment. Figure 1.7 presents solutions of PPHT for a given set of points and various parameters values.

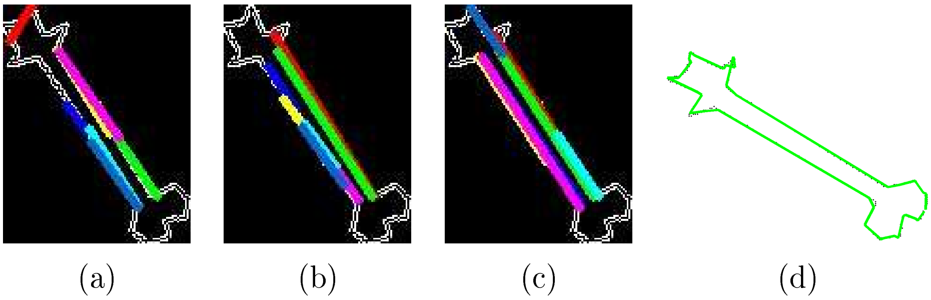

(a) (b) (c) (d)

Figure 1.7: (a)-(c) Results of the PPHT algorithm to a set of points representing the shape of a bone (MPEG7 dataset) y varying the minimum number of points in a bin (namely, 5, 15 and 25). Only a small fraction of the lines is drawn for visualization purposes. Note the overlapping lines. (d) The result of our method. The figure is better seen in color.

Chapter 2

Applications

2.1 Vanishing Point Detection

- 2.1.1 Introduction
- 2.1.2 The algorithm
- 2.1.3 Numerical evaluation

2.2 Point cloud sampling and reconstruction

- 2.2.1 Introduction
- 2.2.2 The algorithm
- 2.2.3 Numerical evaluation

2.3 Shape encoding for edge map image compression

- 2.3.1 Introduction
- 2.3.2 The algorithm
- 2.3.3 Numerical evaluation

2.4 Retinal Fundus Image Feature Characterization

- 2.4.1 Introduction
- 2.4.2 The algorithm
- 2.4.3 Numerical evaluation

2.5 Elimination of outliers from 2D point sets using the Helmholtz principle

- 2.5.1 Introduction
- 2.5.2 The Helmholtz principle
- 2.5.3 The algorithm
- 2.5.4 Numerical evaluation

2.1 Vanishing Point Detection

2.1.1 Introduction

Human-made scenes, such as roads, buildings and their facades or indoor corridor boundaries have a large number of parallel lines in the 3D space. In the framework of a pinhole camera model, two parallel lines are projected onto a pair of converging lines in the 2D image space provided that their 3D plane is not fronto-parallel to the image plane. The common point of intersection of all 3D parallel lines (generally belonging to different planes) in the 2D image is called the vanishing point. The detection of a vanishing point in an image is a crucial step in many computer vision applications, like robot navigation, camera calibration, single view 3D scene reconstruction and pose estimation.

In the related literature, there are two main categories of methods for vanishing point detection. There are techniques requiring knowledge of the intrinsic parameters of the camera, which exploit the notion of 3D parallelism and prominent structures of the scene orthogonal to each other, also called *Manhattan directions* [48, 49]. There are also techniques assuming no knowledge of the internal camera parameters, such as the method in [50] using the Helmholtz principle for image partitioning, the Expectation-Maximization (EM) framework adopted in [51] or the non-iterative algorithm based on consensus sets [52].

In contrast to the above methods, which may base their estimation in the existence of three orthogonal vanishing points, images acquired in structured environments such as roads or corridors are a specific category where the detection of a single vanishing point may be sufficient for the underlined application (e.g. vision-based robot motion along a corridor). The general strategy consists in partitioning the image into accumulator cells collecting votes from the line segments having their intersection in the specific cell. The detection of peaks in the accumulator space provides the vanishing points [53, 54].

An alternative approach is presented in that chapter, based on the DSaM algorithm of chapter 1. Secondly, a voting step is applied through a kernel, where candidate vanishing points are assigned weights proportional to the lengths of the line segments they belong to. Therefore, longer line segments which are more probable in indoor environments (e.g. the intersection of wall and ground) are more probable to contribute to the determination of the vanishing point.

2.1.2 The algorithm

Given an indoor scene (e.g. a corridor), the first step of the method consists in detecting the edges of the image. Therefore, the probabilistic boundaries are first computed [55] though in simpler, non textured environments, the output of the standard but established Canny edge detector detector [44] is generally acceptable. Then the DSaM algorithm is performed to model the scenes line segments. Please note that other methods may also be applied.

The next step consists in fitting line segments to the extracted edges. Various algorithms may be employed, like the one described in 1.

After the determination of the line segments in terms of the long axes of highly eccentric ellipses, the set $\mathcal{C}_{vp}$ of candidate vanishing points is constructed by computing all the pairwise intersection points between all the lines. To further improve the efficiency of the method, intersection points that lie outside the image plane could be ignored but this issue is optional and depends on the specific application. For example, in a corridor, the vanishing point lies within the image plain and intuitively the vanishing line usually appears somewhere in the viewer's horizon. On the other hand, if the algorithm is to be used by a robot navigation system, the detection of the vanishing point outside the image plane may indicate an abrupt turn.

Thence, a weight $w(\mathbf{p}) = |l_1^{\mathbf{p}}||l_2^{\mathbf{p}}|$ is assigned to each point $\mathbf{p} \in \mathcal{C}_{vp}$ which is equal to the product of the lengths $|l_1^{\mathbf{p}}|$ and $|l_2^{\mathbf{p}}|$ of the two line segments whose intersection is the candidate vanishing point $\mathbf{p}$. Thus, a candidate vanishing point produced by short segments, or one long and one short segment, is attributed with a small weight.

In the final step of our workflow, the vanishing point is computed by selecting one of the candidate points, which is achieved through a voting scheme. This step is similar in spirit to the approach presented in [53]. However, the main difference with respect to that method is that, in our algorithm, each line segment votes only for the intersection points belonging to it while in [53] a line segment votes for every candidate vanishing point (even if it does not belong to the segment).

In a voting scheme, an important factor is the size of the accumulator array bins, which in our case is a grid covering the image support $G \equiv [0, G_w] \times [0, G_h] \subset \mathbb{R}^2$. The grid is uniformly divided into equally sized cells B_i, $i = 1, ..., N$, using a scaling factor $\sigma \in (0, 1]$ imposing a bin size of $[\sigma G_w, \sigma G_h]$.

In order to assign weights to each candidate vanishing point present in a given bin, a kernel function centered at each array bin is employed. To this end, a 2D Π-Sigmoid kernel is applied to each bin [56], imposing thus a fuzzy concept to the borders of the cell. The support of a 1D Π-Sigmoid kernel:

$$k(x) = \frac{1}{b-a}\left[\frac{1}{1+e^{-\lambda(x-a)}} - \frac{1}{1+e^{-\lambda(x-b)}}\right], \tag{2.1}$$

with $\lambda > 0$, which is depicted in figure 2.1, approximates a uniform kernel whose borders are fuzzified in order to avoid abrupt changes. Thus, points under the plateau contribute equally with their votes while the contribution of points lying at the extremities falls off quickly but it does not become zero, depending on the value of the parameter λ. The larger the value of λ the less fuzzy the kernel borders become and consequently the edges of the kernel are very steep (fig. 2.1). By these means, the capture region of the kernel allows points from the neighboring bin to contribute with a relatively low non-zero weight.

A 2D Π-Sigmoid kernel $\Pi_s(\mathbf{x}; \mathbf{a}, \mathbf{b}, \lambda)$ with parameters $\mathbf{a} = (\alpha_1, \alpha_2)$, $\mathbf{b} = (b_1, b_2)$, with $\alpha_i \leq b_i$, and λ is a separable function that may be generated from the product of two $1D$

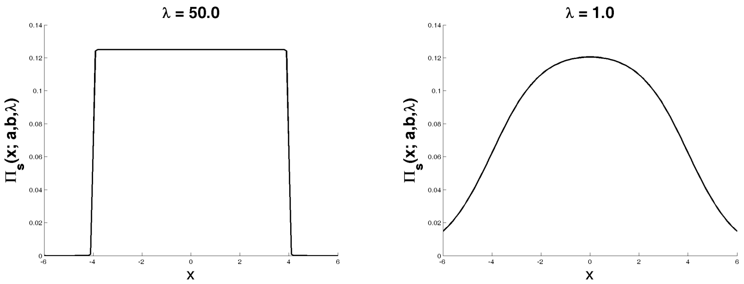


Figure 2.1: $\Pi_s(\mathbf{x}; \mathbf{a}, \mathbf{b}, \lambda)$ for $\lambda = 50$ and $\lambda = 1$.

kernels:

$$\Pi_s(\mathbf{x}; \mathbf{a}, \mathbf{b}, \lambda) = \prod_{d=1}^{2} \frac{\frac{1}{1+e^{-\lambda(x_d - a_d)}} - \frac{1}{1+e^{-\lambda(x_d - b_d)}}}{b_d - a_d}.$$

Parameters **a** and **b** control the width of the kernel, while the slope λ controls the fuzziness of the kernel.

Thus, the total votes casted to cell B_i are computed by:

$$V(B_i) = \sum_{\mathbf{p} \in \mathcal{C}_{vp}} w(\mathbf{p}) \Pi_s(\mathbf{p}; \mathbf{a}_i, \mathbf{b}_i, \lambda). \tag{2.2}$$

The voting process is concluded by detecting the dominant cell B^* according to:

$$B^* = \arg\max_{B_i} \{V(B_i)\}. \tag{2.3}$$

Finally, the coordinates of the vanishing point are computed as the weighted average of all the candidate vanishing points with respect to the Π-Sigmoid kernel centered at the cell B^*:

$$\mathbf{p}^* = \frac{\sum\limits_{\mathbf{p} \in \mathcal{C}_{vp}} w(\mathbf{p}) \mathbf{p} \Pi_s(\mathbf{p}; \mathbf{a}^*, \mathbf{b}^*, \lambda^*)}{\sum\limits_{\mathbf{p} \in \mathcal{C}_{vp}} w(\mathbf{p}) \Pi_s(\mathbf{p}; \mathbf{a}^*, \mathbf{b}^*, \lambda^*)}, \tag{2.4}$$

where $\mathbf{a}^*, \mathbf{b}^*, \lambda^*$ are the parameters of the kernel corresponding to B^* in (2.3). Note that all the candidate points contribute to the solution. However, the importance of the points in the dominant cell is overwhelming. The steps of the method are summarized in Algorithm 1. In order to increase the robustness of the algorithm and to speed it up, line segments that are shorter than a threshold T and their orientation is close to horizontal or vertical within θ degrees are pruned. Although this rule could be omitted, it was deduced that setting the value of parameter T to 5% of the size of the diagonal of the image and $\theta = \pm 15^o$ improves significantly the performance of the method.

2.1.3 Numerical Evaluation

To evaluate our method, we created two sequences, with 35 and 18 frames respectively, with a frame size of 320 × 240 pixels each. The images in each sequence were captured

```
input: A color image.
output: The coordinates of the vanishing point.
Detect the edges of the image.
Detect line segments (e.g. use the algorithm introduced in [6]).
Prune segments whose length is below $T$ and their orientation is vertical or horizontal within $\pm\theta^o$.
Compute the coordinates of pairwise intersection points between all segments.
Voting
Calculate the votes for each cell $B_j$, $j = 1, \ldots, N$ using (2.2).
Find the dominant cell using (2.3).
Compute the vanishing point using (2.4).
```

Algorithm 2: Vanishing point detection algorithm

periodically by a robot moving on a specific course. The sequences represent an indoor corridor under various illumination conditions. To make the task more challenging, in the second sequence, a person walking towards the robot appears in all of the frames. Then, 5 individuals were asked to detect manually the vanishing point in each image. The ground truth vanishing point for each image was considered to be the mean point indicated by the volunteers. The standard deviation of the various vanishing points provided by the humans is 11 pixels which is approximately 3.5% of the shorter image dimension.

In order to investigate the dependence of the final result on the values of the parameters λ of the Π-Sigmoid kernel and the grid resolution tuned by σ, experiments were performed, examining the mean detection error and the execution (in Matlab) time with respect to those parameters. Note that as the grid resolution decreases the algorithm demands more execution time because it integrates a larger number of kernels. The results are summarized in Table 2.1, where we have tested the behavior of the algorithm for two configurations for the parameter λ, namely $\lambda = 10$ and $\lambda = 50$. As it may be observed, the proposed method exhibits a consistent behavior since the variation of the detection error is rather small concerning the different configurations of the parameters. The pair of parameters $\lambda = 50$ and $\sigma = 0.05$ is a good compromise between detection error and execution time. The method provides, in general, accurate results considering that its detection error is in average 2% of the image diagonal. Moreover, as the algorithm was developed in Matlab it may be further accelerated and easily integrated in embedded systems.

Table 2.1: Algorithm Performance

	$\lambda = 10$				$\lambda = 50$			
σ	0.05	0.08	0.10	0.20	0.05	0.08	0.10	0.20
Time per image (sec)	0.2	0.1	0.1	0.1	0.2	0.1	0.1	0.1
Error (pixels)	5.4	6.7	7.6	15.4	5.4	6.8	7.5	15.4

We also compared our direct split-and-merge framework (DSaM) to the Hough Transform (HT), which is widely used for line detection. We kept the proposed voting scheme in both algorithms. At first, the HT needs is relatively difficult to be tuned due to the tedious task of determining the bin sizes. Moreover, the HT provided large errors (of the order of 15 pixels) and thus failed to correctly detect the vanishing point in indoor images because it was affected by spurious points.

Representative results of our method are given in figure 2.2. The images depict frames of an indoor corridor, with and without obstacles. The corresponding error between real and computed vanishing point is 1.54 pixels. The green lines correspond to the line segments computed by the DSaM algorithm and represent the image edges. The blue lines represent the edges contributing to the detection of the vanishing point. The red star sign depicts the vanishing point as it was computed by our method, while the yellow circle is the ground truth.

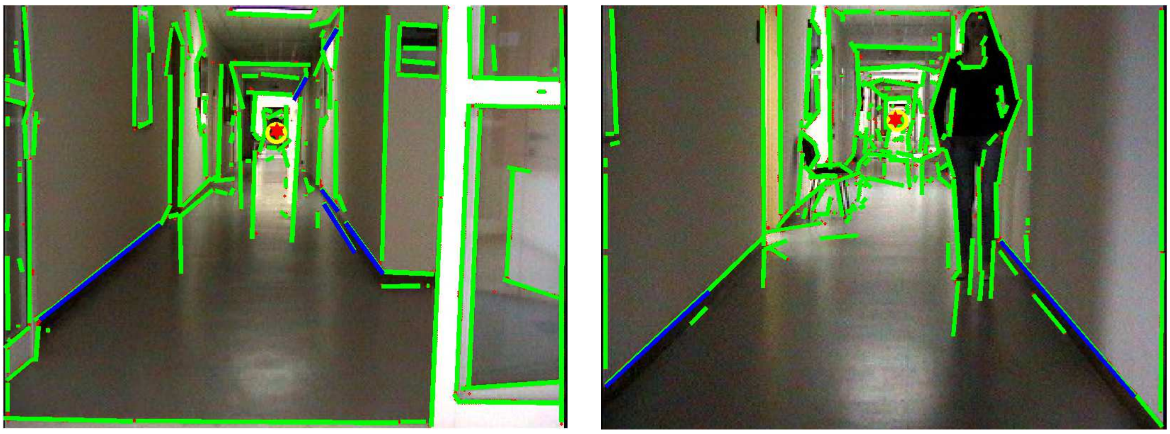


Figure 2.2: Representative results of the VP detection with the proposed method (the figure is better viewed in color).

Finally, we conducted some experiments to compare the proposed voting scheme against the trivial case of the computation of the VP using least squares. In this approach, all vanishing lines are computed and the VP is considered to be the point that has the shortest distance to all lines. In a preprocessing step, a line pruning procedure eliminates the lines that are either vertical or horizontal, within a threshold range. We compared various line segment detection methods (HT, Kovesi [5], LSD [40, 41, 42]) and configurations. For this experimental setup, we employed a video stream from a vehicle moving straight on a highway. 19 successive frames were extracted from the video, that correspond to a distance of about 130 meters. Since the ground truth is not known, we tested the stability of each method, by computing the number of successive frames where the distance between the detected VP in the two frames is less or equal a threshold. The distance between two VP detected in successive frames is a rejection criterion for the validity of the computation. Thus, the stability of the result is critical. Figure 2.3 shows a representative frame of the video stream, while Table 2.2 summarizes the experimental results.

Ideally, large numbers should appear in the first columns of Table 2.2, indicating that the majority of the VP are located close to each other. On the other hand, large numbers in the last columns of Table 2.2 indicate that the detected VP are moving within the image space between successive frames, and thus the associated method is not considered stable.

Multiple appearance of some methods, indicate a different parameter tuning. The name of the method indicates the line detection algorithm and the algorithm used for point detection. In case of the HT executions, the size of the bins of the histogram in the voting space was altered, leading to a different number of lines contributing to the detection of the VP. As far as the number of peaks that should be detected in the HT voting process, it was set to the same number of line segments detected by our method. Since our method provided accurate results, this tuning minimized the impact of an erroneous selection of the number of lines in the HT voting process. Regarding the method of Kovesi, different values for the merging threshold were considered. To minimize the impact of an arbitrary selection of the threshold value, we also employed the merge threshold provided by our method in one of the variations for the VP detection based on the method of Kovesi.

One may observe that the proposed voting scheme is superior to the conventional least squares approach, as it provides more stable results. Moreover, the combination of the DSaM method, presented in chapter 1 and the LSD line detector along with the proposed voting scheme manage to provide a quite stable and efficient detection, with the latter method presenting slightly better results. Note also that the DSaM result is highly related to the Canny edge detection, a case that is handled intrinsically in the LSD algorithm.

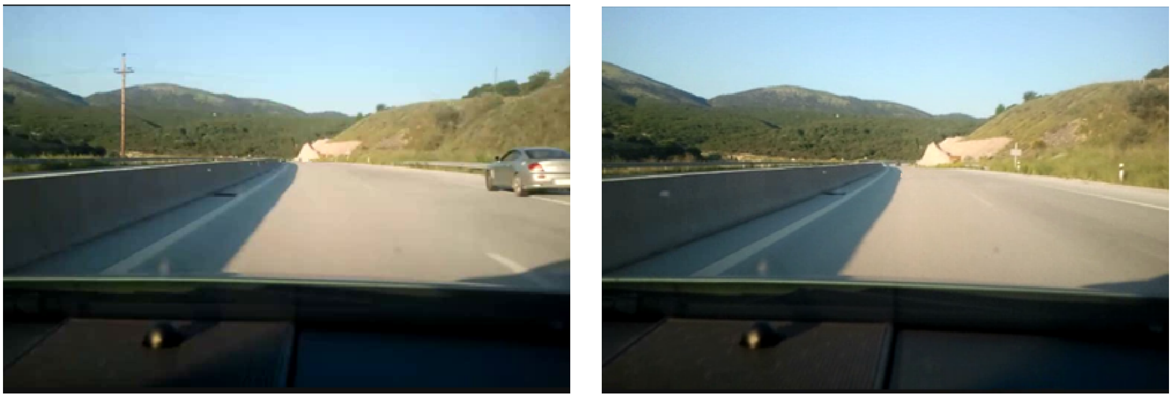


Figure 2.3: Representative results (the figure is better viewed in color).

Table 2.2: Number of two successive frames where the distance of the detected VP in the two frames is less or equal to a threshold

Method	Distance (in pixels)										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	≥10
DSaM+Π-Sigmoid	0	0	1	5	6	7	8	9	10	11	8
HT+LSE (1)	0	0	0	0	0	0	0	0	0	0	19
HT+LSE (2)	0	1	1	1	2	2	2	2	3	3	16
HT+LSE (3)	0	0	1	1	1	2	5	5	5	5	14
HT+LSE (4)	0	0	0	0	0	0	0	0	0	0	19
HT+LSE (5)	0	1	1	1	2	2	2	2	3	3	16
Kovesi+Π-Sigmoid (1)	0	0	1	1	1	2	2	2	4	4	15
Kovesi+Π-Sigmoid (2)	0	0	0	0	1	1	1	1	2	3	16
Kovesi+LSE (1)	0	0	0	0	0	0	0	0	0	0	19
Kovesi+LSE (2)	0	0	0	1	1	2	3	4	5	5	14
LSD+Π-Sigmoid	0	0	2	5	7	10	10	10	15	17	2
LSD+LSE	0	0	2	2	2	2	2	4	7	7	12

2.2 Point cloud sampling and reconstruction

2.2.1 Introduction

As modern image analysis and computer vision algorithms have become more complex requiring a large number of operations and the the data to be processed are big, a preprocessing step is a necessary task that may assist toward efficient and fast processing. In many cases, that step involves sampling an original image or it edge map (e.g. computation of the vanishing point) in order to keep a fraction of points that describe with fidelity the initial information. More specifically, in image processing, this leads to edge pixel sampling so as to extract the eventually hidden patterns (e.g. object contours) inside an initial observation so that the result is as close as possible to the observation.

The most straightforward approach is to apply random sampling, which assumes that the edge points are observations of a random variable that follows a specific distribution. As soon as we model that distribution, point sampling is augmented to sampling observations from a known distribution. In the simplistic random sampling, it is assumed that the original set follows a uniform distribution. A more advanced, but notoriously time consuming probabilistic model is Monte Carlo sampling [45]. J. Malik independently proposed contour sampling in [57] to apply it to an object retrieval algorithm [7]. Initially, a permutation of the points is computed and a large number of the samples is drawn from that permutation. Then iteratively, the pair of points with the minimum pairwise distance is detected and one of them is kept as a valid sample. This process is iterated until the desired number of samples is reached and it ensures that points from image regions with large density will be part of the final data set.

In [58], the fast marching farthest point sampling method is introduced for the progressive sampling of planar domains and curved manifolds in triangulated point clouds or implicit forms. The basic idea of the algorithm is that each sample is iteratively selected as the middle of the least visited area of the sampling domain. For a comprehensive review of the method, the reader is also referred to [59].

The Fourier transform and other 2D/3D transforms have been applied for describing shape contours for compression purposes. For example, in [60], the idea is to warp a 3D spherical coordinate system onto a 3D surface so as to model each 3D point with a parametric arc equation. However, this method demands an ordering of points and cannot model clouds of points, where more complex structures, like junctions and holes, are present.

A framework for shape retrieval is presented in [61]. It is based on the idea of representing the signature of each object as a shape distribution sampled from a shape function. An example of such a shape function would be the distance between two random points on a surface. The drawback of the algorithm is that the number of initial points has to be relatively small for the method to be fast and efficient. This may lead to a compromise between the number of points of the sample and the information loss. Moreover, the efficiency of the method highly depends on the presence of noise.

This type of edge sampling is a preponderant step before other algorithms are applied. This is the case in [7], where sampling reduces the amount of data for object recognition and in [62], where a shape classification algorithm necessitates a small number of samples to reduce its complexity. Hence, the quality of sampling may affect the final result if the resulting point set does not preserve the coherence of the initial information. Considering also that most of these algorithms demand large complexity in terms of resources (i.e. memory allocation) in order to extract complex features that discriminate better the various data, one may come to the conclusion that sampling may be a very crucial step.

In this work, we propose an algorithm for fast, accurate and coherent sampling of image edge maps. The procedure consists of two steps. At first, the image edges are summarized by a set of line segments, which reduces the initial quantity of points but accurately preserves the underlining information contained in the edge map. Then, based on the ellipse-based representation, a decimation of the ellipses is performed and samples are drawn according to their location on the long ellipse axis.

2.2.2 The algorithm

The first step of the algorithm relies in extracting the line segments that model the point clouds. This task can be fulfilled by the DSaM algorithm explained in chapter 1, or any other line segment algorithm introduced in the related literature. However, the accuracy of the result is highly related to the efficiency of the line segment modeling.

Assume now that the goal is to sample the set of points that are presented in Fig. 2.4 and keep only $Q\%$ of them. The black dots represent the original points. This may be considered as the output of the DSaM algorithm [6]. More specifically, these points lie on the long axis of a highly eccentric horizontal ellipse. The axis is shown in red. In order to approximate the local point distribution, a histogram is computed with a number of bins equal to $Q \times L$, where L is the number of initial points in the set to be sampled. Then, we represent each bin by its mean value, which under e.g. Gaussian assumption it is the geometric center of the points in the bin and we select in each bin the point that is closer (in terms of Euclidean distance) to this geometric mean. By repeating the procedure for each line segment produced by the application of the DSaM algorithm we are able to sample the original point cloud.

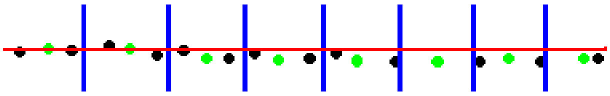


Figure 2.4: An example of the sampling process. The black points represent the original set of points, while the red line is the is their *summary* computed by DSaM [6]. The vertical blue lines depict the limits of the histogram bins. The green points are those selected to represent the sampled set because they are closer to the mean value of the bin. The figure is better seen in color.

It may be easily understood that the efficiency of the approach is highly related to the correct determination of a model approximating the local manifold of the point set. The larger the deviation of the model from the local manifold becomes, the less accurate is the sampling method. This is true as the model fails to compute the histograms correctly and therefore to establish accurately the bin centers. Consequently, the selected samples will be less representative of the distribution of the initial edges. For that reason, we relied on the DSaM algorithm which may accurately describe the edge map.

An important issue of the sampling algorithm is the value of the sampling frequency Q, that is how densely should we sample? Moreover, the number of samples should vary locally with respect to the number of image edges present in an image region. To this end, based on the clustering of points to highly eccentric ellipses, we propose to select the number of samples to be equal to δ times the number of points that are present in the mostly populated ellipse, where $\delta \geq 1.0$. This guarantees that, highly concentrated image regions will be more densely sampled but also that sparse regions should always have some representatives as they have already been assigned to an ellipse. This is in contrast to random or even Monte Carlo based sampling where sparse areas may have no representatives in the sampled data set.

In other words, the sampling rate is computed by

$$Q = \delta \frac{R}{L}, \tag{2.5}$$

where R is the maximum number of members in the clusters, L is the total number of points in the original set and $\delta \geq 1.0$ is a real positive number. The larger the value of δ is, the more samples we get, and thus the closer to the initial set our sampling result is. In other words, the estimation of the p-value of parameter δ is a compromise between the quality of the result and its complexity.

So far we have presented an algorithm for reducing the number of points in a set with a minimal impact regarding the information loss. This rationale, can be extended, to extract shapes from point clouds. In other words, we will present a method for extracting shapes with a desired resolution, in terms of number of points. If we assume that the manifold locally can be approximated linearly, then the DSaM method introduced in section 1 can be employeed. A line segment ϵ can be described by its starting and ending points, $\mathbf{x_s}$ and $\mathbf{x_e}$ respectively, i.e. $\epsilon = \{\mathbf{x_s}, \mathbf{x_e}\}$ where $\mathbf{x_s}, \mathbf{x_e} \in \mathbb{R}^2$. In principal, a line is modeled by the following equation $Ax + By + \Gamma = 0$. where $x,\ y,\ A,\ B, \Gamma \in \mathbb{R}$ and (x, y) is a point laying onto the line. Since $\mathbf{x_s}$ and $\mathbf{x_e}$ are given, determining $A,\ B,\ \Gamma$ is trivial. Then following a similar process with the sampling algorithm described earlier, we can reproduce the corresponding shape part, at any desired resolution. Thus, starting from point $\mathbf{x_s}$ and following the direction of the line segment with a predefined step $\lambda \in \mathbb{R}^+$ each time, we may reconstruct (approximate) the initial points. The value of the step λ controls the density of the result: the higher its value is, the more points we extract. Algorithm 3 demonstrates the steps of the proposed shape reconstruction method.

The contribution of the DSaM method is that in manages to model accurately enough the hidden manifold of the point cloud and thus provide efficient features (line segments) for shape reconstruction. A direct application of this method would be the efficient detection of the projection of random point onto the contour of a shape, like the method proposed in [63].

```
input: The set of unordered points $X = \{\mathbf{x}_i | i = 1, \dots, N\}$, $M \in \mathbb{N}$.
output: The computed set of points $Y = \{\mathbf{x}_i | i = 1, \dots, M\}$ that describe a shape.
Run the DSaM algorithm (refer to chapter 1) to model the manifold.
Let $\epsilon_i = \{\mathbf{x_s^i}, \mathbf{x_e^i}\}$, $i = 1, \dots, K$ be the line segments extracted, with $\mathbf{x_s^i}$, $\mathbf{x_e^i}$ being the starting and ending points of the $i$-th segment respectively.
for i=1:K do
  Let $\mathcal{R}_i = \{r_0^i, r_2^i, \dots, r_{M_i}^i\}$ be the reconstructed points based on the segment $\epsilon_i$, where $r_0^i = \mathbf{x_s^i}$ and $r_{M_i}^i = \mathbf{x_e^i}$.
  $\lambda_i \in \mathbb{R} = \frac{|\mathbf{x_e^i} - \mathbf{x_s^i}|}{M_i}$.
  for j=1:M-2 do
    $r_j^i = r_{j-1}^i + \lambda \vec{e}$, where $\vec{e}$ is the unit vector with direction similar to the direction of $\epsilon_i$.
$Y = \bigcup_{i=1}^{K} \{\mathcal{R}_i\}$.
```

Algorithm 3: Shape reconstruction from a 2D point cloud

2.2.3 Numerical Evaluation

For our experiments we used two datasets with contours of various objects: The MPEG7 dataset [1] and the Gatorbait dataset [2]. For more details about those datasets, please reffer to section 1.3. The edges were extracted with the Canny edge detector [44] and the coordinates of the edge pixels were used as input to our experiments.

A minimum description length (MDL) approach [45] is adopted to compute the value of δ. We define:

$$\Phi(\delta) = \mathcal{D}(\downarrow (X_{or}, \delta), X_{or}) + \lambda | \downarrow (X_{or}, \delta)| \tag{2.6}$$

where X_{or} is the original set of points, $\downarrow (X_{or}, \delta)$ is the output of the sampling process applied to set X_{or} with the sampling rate computed by (2.5), $| \cdot |$ denotes the cardinality of the corresponding set and $\mathcal{D}(P, Q)$ is the Hausdorff distance between the set of points P and Q.

In order to learn parameter δ, we randomly selected 119 images from our dataset. The DSaM sampling method was executed for various values of the parameter δ in the interval $[1.0, 3.0]$ and $\delta = 1.6$ minimized $\Phi(\delta)$, which was used in our experiments (Figure 2.5).

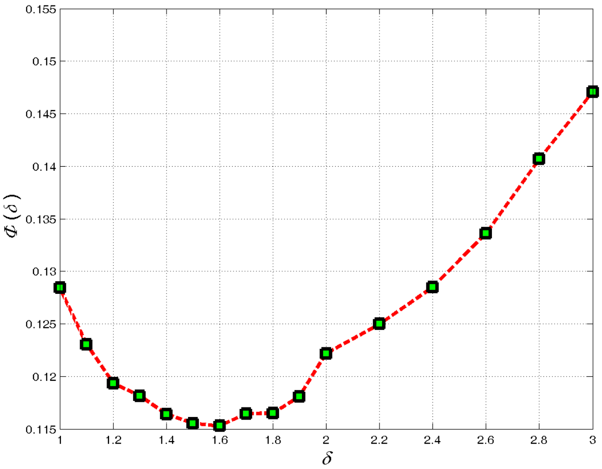


Figure 2.5: The value $\delta = 1.6$, which minimizes $\Phi(\delta)$ was used in our experiments.

The efficiency of our method was evaluated by comparing it to widely used methods such as the sampling scheme proposed by Malik [57], Monte Carlo sampling and simple random sampling. In order to quantitatively measure the fidelity of the sampled point set to the original one we used the Hausdorff distance between the aforementioned point sets.

The rational is that the smaller the $D(X, Y)$ becomes, the closer the sample is to the initial data. This concept may be considered as a try to minimize the distortion-compression rate. In other words, we wish to sample a set of points (compress) by keeping the information loss small (distortion). Moreover, to establish a common baseline, we used the same sampling rate for all of the compared methods, which is the one described in the previous section. In order to avoid any possible bias, we also tested smaller sampling rates for the other methods. the idea was to explore whether they produce better results, in terms of similarity with the original shape using these smaller sampling rates. However, the results proved that by decreasing the sampling rate the results become poorer for the other methods.

The overall results are summarized in Table 2.3. As it can be observed, our method provides better results in all cases with regard to all of the compared methods. Representative results on Gatorbait [2] dataset are demonstrated in Fig. 2.6. The reader may observe that our method manages to preserve better the details of the original set, as it produces more uniform results and thus the distribution of the points in the sampled set is closer to the original.

Table 2.3: Hausdorff distance between the original and the sampled sets using different sampling methods.

MPEG7 [1] (70 shapes)

Algorithm	**mean**	**std**	**min**	**max**
Proposed method	0.00	0.02	0.00	0.29
Malik [57]	0.00	0.05	0.00	1.10
Monte Carlo	0.03	0.32	0.00	6.21
Random Sampling	0.01	0.19	0.00	3.31

GatorBait100 [2] (38 shapes)

Algorithm	**mean**	**std**	**min**	**max**
Proposed method	0.21	0.04	0.05	0.35
Malik [57]	0.30	0.11	0.11	0.51
Monte Carlo	3.08	3.27	0.61	17.02
Random Sampling	1.09	0.38	0.51	1.88

In a second set of experiments we examined the improvement of the result of a shape retrieval algorithm that includes a sampling preprocessing step. This is a very common problem in computer vision and image analysis and much research has been performed in this field. We focused on the pioneering algorithm introduced in [7] and explored the improvement of the detection rate (*Bull's Eye Rate*) by applying various sampling methods including ours.

The overall evaluation is presented in Table 2.4. It may be seen that the proposed method improves the retrieval percentage . In case of the Gatorbait dataset [2] the improvement of the Bull's Eye Rate is around 2.5% with respect to the second method.

In order to measure the similarity between two shapes we adopted the χ^2 distance between shape contexts, as explained in [7]. However, in this problem, a more informative index should be applied to take into consideration the deformation (e.g. registration) energy that is demanded so as to transform one edge map onto the other. Yet, as we wish to investigate the improvement that our method provides in terms of similarity between samples and original signals, we opted not to compute the related parts of the similarity metric in [7]. Moreover, to speed up our experimental computations, we opted not to use a dynamic programming approach to guarantee a one-to-one matching. Instead, we assigned each point from one set to each closest in the other and computed the cost of

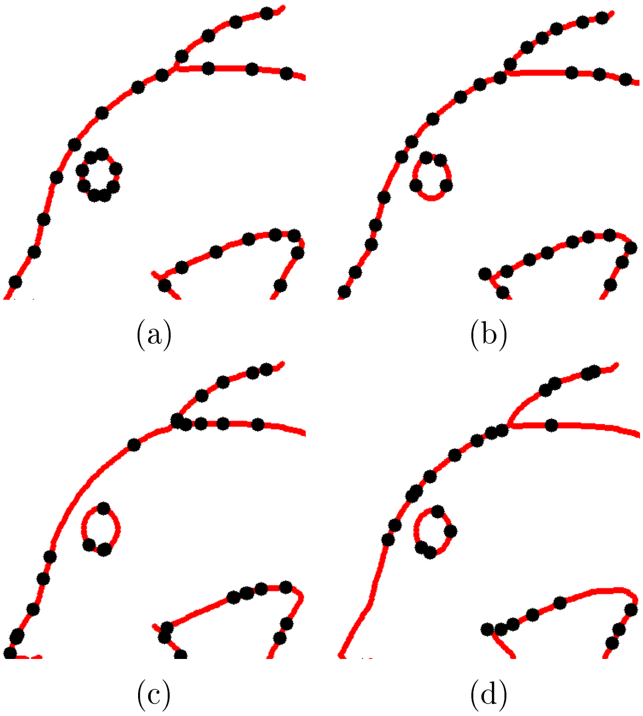


Figure 2.6: Representative results of sampling of the Gatorbait dataset [2]. Details of the upper left part of a fish contour. Sampling with (a) the proposed method, (b) the method of Malik [7], (c) Monte Carlo sampling and (d) Random sampling.

this assignment in terms of corresponding histogram distances. By repeating that process for all points and summing the related distances, we computed the total distance between two shapes. These remarks, explain the differences in Bull's Eye Rate index computed for the MPEG7 dataset, compared to the one provided in [7].

Since a crucial step of the proposed method is the accurate manifold detection, we compared our sampling method with a widely used method for manifold detection, namely Locally Linear Embedding (LLE) [64]. The reader may observe that LLE does not provide accurate results, since it fails to model the various inner structures and junctions that are present in the experimental data (Table 2.4).

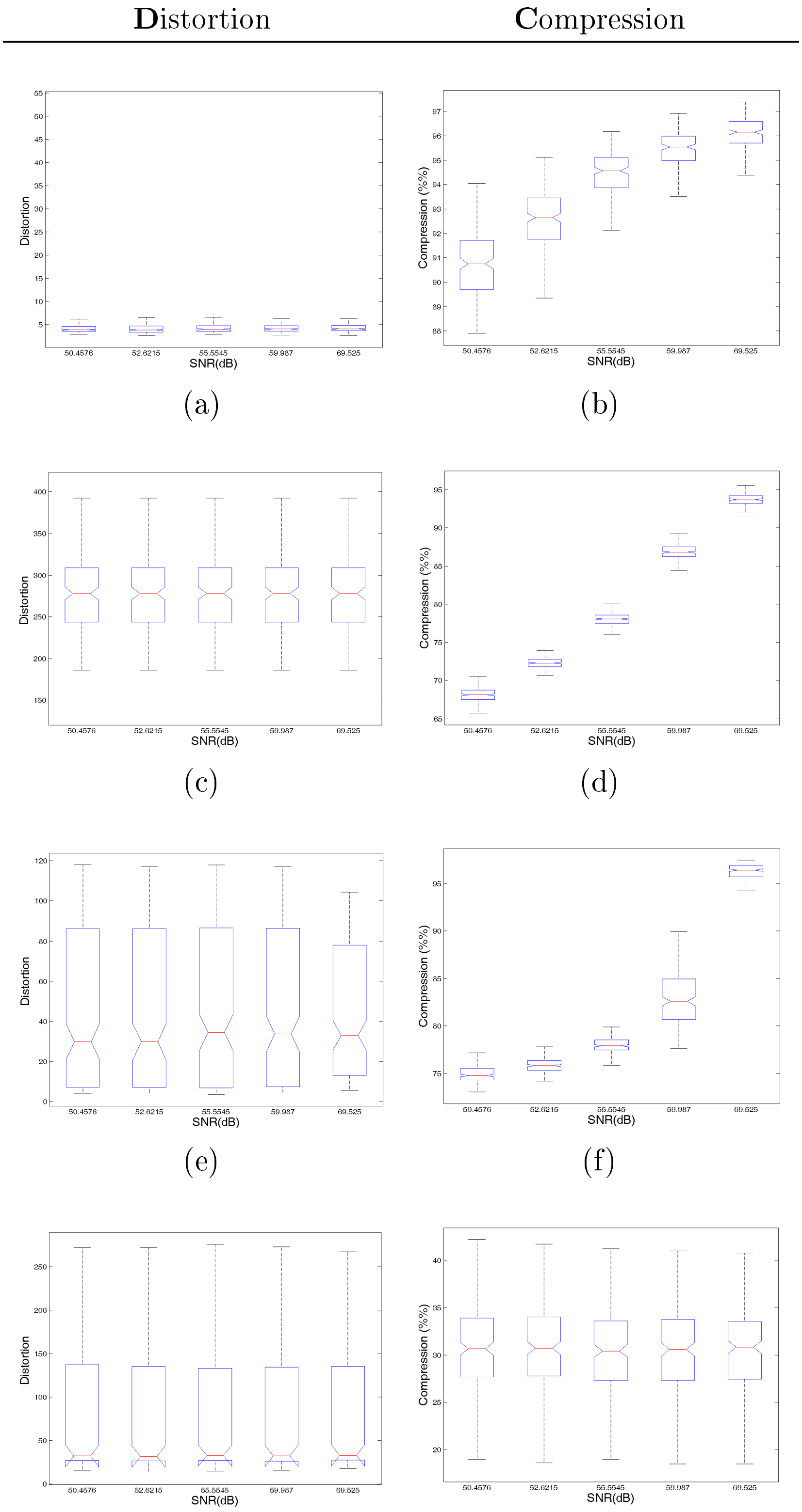
Table 2.4: Bull's Eye Rates for the retrieval of sampled sets using different sampling methods.

Algorithm	**Bull's Eye Rate**	
	MPEG7 [1]	**GatorBait100** [2]
Proposed method	65.40%	96.57%
Malik [57]	64.96%	93.89%
Monte Carlo	50.71%	77.69%
Random Sampling	57.86%	91.11%
LLE [64]	53.81%	93.98%

One may argue that instead of using the DSaM algorithm, we could traverse the point set by visiting the nearest neighbor of each point successively following a polygon approximation variation. For demonstration purposes, we call this approach Nearest-Neighbor Split (NN-S). Finally, as our algorithm is based on line segment fitting, we also tested it against one widely used similar algorithm [5], that utilizes nearest neighbors.

Moreover, since we are dealing with manifold learning from point clouds, we also tested the tensor voting framework, [8, 9]. Tensor voting is a robust and efficient method, however its threshold tuning is not trivial and the result is highly related to the selected threshold values. In general, tensor voting, manages to extract a large part of the contour. However, the large distortion is due to the fact that some parts of the shape silhouette are missing, i.e. holes are present, or new points are added, that are not present in the pure data. For our experiments, we selected those threshold values that provided the best result. Figure 2.7 demonstrates the results of our experiments. As it can be seen in the figures, the proposed method provides more accurate results compared to the other methods in terms of shape reconstruction as it is explained below.

For this experimental configuration, we disturbed the shape silhouettes with additive Gaussian noise with progressively increasing variance. The evaluation of the method is based on a distortion-compression model for varying signal-to-noise (SNR) ratio, where the distortion is measured in terms of shape similarity between the reconstructed shape and the initial one, and compression is the number of segments computed via DSaM over the total number of points. Please note that although this approach is similar to our analysis in 1, in this section the distortion is computed in a different way, which is more meaningful in the context of the application.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 2.7: Shape reconstruction of the Gatorbait dataset [2] using (a)-(b) DSaM, (c)-(d)NN-S, (e)-(f) Kovesi [5], (g)-(h) vensor voting [8, 9].

2.3 Shape encoding for edge map image compression

2.3.1 Introduction

Shape representation is a significant task in image storage and transmission, as it can be used to represent objects at a lower computational cost, compared to non-encoded representations. For example, the widely used MPEG4 Part 2 object-based video standard uses shape coding for describing regions, called video object planes, that represent an object [65]. In that case, accurate shape encoding leads to better preservation of contour details.

The pioneering work in [66], where sequences of line segments of specified length and direction are represented by chain codes was proposed for the description of digitized curves, contours and drawings and it was followed by numerous techniques. Shape coding is a field that has been studied extensively in the past but it is still very active. Various methods have been studied in [67], including the context-based arithmetic encoding (CAE), which has been adopted by the MPEG4 Part 2 standard.

The digital straight line segments coder (DSLSC) was introduced in [68] for coding bilevel images with locally straight edges, that is, single binary shape images and bilevel layers of digital maps. DSLS models the edges by digital straight line segments (DSLS) [69]. Compared to standard algorithms like JBIG [70], JBIG-2 [71] and MPEG4 CAE [65],[67] DSLSC provides better results, as it fully exploits the information given by the local straightness of the boundary, which is not hte case for the other methods.

DSLSC is further improved in [72], where the segmentation of the alpha plane in three layers (binary shape layer, opaque layer, and intermediate layer) is employed. Experimental results demonstrated substantial bit rate savings for coding shape and transparency when compared to the tools adopted in MPEG4 Part 2.

Discrete straight lines were also employed in [73] for shape encoding and improvement of the compression rate is reached by carrying out a pattern substrings analysis to find high redundancy in binary shapes.

A lossless compression of map contours by context tree modeling of chain codes is described in [74]. An optimal n-ary incomplete context tree is proposed to be used for improving the compression rate.

A JBIG-based approach for encoding contour shapes is introduced in [75], where a method is presented that manages to efficiently code maps of transition points, outperforming, in most cases, differential chain-coding.

2.3.2 The algorithm

Line segments are important features in computer vision, as they can encode rich information with low complexity. We take advantage of this feature for encoding a 2D set of points describing a shape as a collection of line segments that approximate the manifold of the shape, by assuming that the manifold is locally linear. The initial and ending points of each line segment may be considered as the characteristic points carrying the compressed information that can reproduce the initial shape. The larger the number of characteristic points is, the better shape information is preserved.

A line segment ϵ may be described by its starting and ending points $\mathbf{x_s}^\epsilon$ and $\mathbf{x_e}^\epsilon$ respectively. The collection of the starting and ending points of all the segments modeling the shape manifold are the characteristic points of the shape. Note that the characteristic points are ordered. Moreover, since the traversal of the line segments is known, the line segment can be described by its starting point and the transition vector towards the ending point. The ending point of one segment is the starting point of its successor in the traversal order. Eventually, the shape can be encoded by selecting an arbitrary initial point from the characteristic points and by the corresponding transition vectors after visiting each segment based on the traversal order, in a similar manner described in [11].

To reconstruct the image we need to reconstruct all the points contributing to the computation of each line segment ϵ based on the characteristic points. In principle, a line is modeled by the parametric equation $Ax + By + C = 0$. where x, y, A, B, $C \in \mathbb{R}$ and (x, y) is a point laying onto the line. If the starting ($\mathbf{x_s}^\epsilon$) and ending ($\mathbf{x_e}^\epsilon$) points of a line segment ϵ are given, determining A, B, C is trivial. Thus, starting from point $\mathbf{x_s}^\epsilon$ and following the direction of the line segment with a predefined step $\lambda \in \mathbb{R}^+$ each time, we may reconstruct (approximate) the initial points. The value of the step λ controls the density of the result: the higher its value is, the larger is the number of extracted points. In case of points laying on an image grid, integer arithmetics need to be considered and selecting $\lambda = 1$ yields the algorithm of Bresenham [76] which may reconstruct the line segment pixels efficiently and handle the aliasing effect.

Algorithms 4-5 describe the proposed framework for compression/decompression of bi-level images of edge maps.

```
input: An edge map image I, representing shapes.
output: A set of features S that encodes image I.
Detect the line segments that describe I. Let K be the number of line segments detected.
Detect the traversal order of the line segments.
Refine shape, i.e. close gaps between line segments. Extract the characteristic point $P = \mathbf{p_i},\ i = 1 \ldots K$, based on the shape traversal.
$S = \{\mathbf{p_1}\}$.
for i=2:K do
    $S = S \cup \{\mathbf{dx},\ dx = \mathbf{p_{i-1}} - \mathbf{p_i}\}$.
```

Algorithm 4: Image compression

2.3.3 Numerical evaluation

In this section, the experimental investigation of the proposed method is presented regarding its robustness and efficiency. To that end, a compression-distortion study was carried out.

```
input: A set of features S that encodes an image I.
output: The reconstructed image I.
Recover the characteristic points $P = \{\mathbf{p_i},\ i = 1 \dots K\}$, based on initial point and transitions encoded in S.
for i=2:K do
    Produce the set of points $R$, e.g. [76], containing the points of the line segment from $\mathbf{p_{i-1}}$ to $\mathbf{p_i}$.
    Set the pixels of I corresponding to coordinates of points in R on.
```

Algorithm 5: Image decompression

Compression was computed as the ratio of the file size between the compressed and the original files. Various lossless methods were considered and the corresponding size of the output files they produced was used as the reference original file size. The methods against which we compared the proposed framework are the CCITT G4 standard [77] (denoted as FAX4 herein), adopted amongst others by the TIFF image file format for binary images, and the widely used standards JBIG [70] and JBIG2 [71].

As far as the distortion is concerned, a twofold computation was performed in terms of measuring the loss of information and the similarity between the initial and the final edge map images. Therefore, the distortion index adopted by MPEG4 [78], given by

$$D_R = \frac{\text{Number of pixels in error}}{\text{Number of interior pixels}}, \tag{2.7}$$

was also used in this work. The Hausdorff distance between the original edge map X and the reconstructed edge map Ys, given by

$$D_H(X, Y) = \max_{\mathbf{x} \in X} \min_{\mathbf{y} \in Y} \{|\mathbf{x} - \mathbf{y}|_1\}, \tag{2.8}$$

was used to measure the similarity between X and Y.

For the experimental study, we used two datasets. The Gatorbait dataset [2] contains the silhouettes of 38 fishes, belonging to 8 categories. The MPEG7 dataset [1] contains 1400 object contours belonging to 70 categories, with 20 members per each category. All datasets consist of binary images. In the case of the Gatorbait100 dataset, the images were initially thinned so as to extract the contour line. Let us note that in this case, there are some inner structures that were also considered in our experiments.

The overall results of our experimental analysis are demonstrated in Tables 2.5 and 2.6, with results for various configurations of the line segment detection algorithms considered. The values next to the method prefix in the first column of the Tables indicates the corresponding configuration of the method. We used the line segment method presented in chapter 1 and the widely used polygon approximation [10], as proposed in [11]. In our study the values considered for the DSaM thresholds were $\{[0.3, 2.0], [0.4, 2.0], [0.5, 2.0], [0.8, 2.0], [1.3, 2.0], [2.3, 2.0]\}$. The second threshold was measured in pixels. As far as the polygon approximation (PA) is concerned, this algorithm applies one threshold controlling the deviation of a set of points from linearity. In that case, the thresholds used were $\{1, 2, 5, 7, 10\}$ pixels. The percentages regarding the compression values in Tables 2.5 and 2.6 refer to the file size produced by the proposed compression scheme compared to the corresponding file size produced by the related lossless method as mentioned on the second row of the tables. More specifically, in Table 2.5, in the first row, we may conclude that the proposed scheme, using DSaM for line modeling, provides a compressed shape, which on average (over the whole data set) employs 16% of the bits employed when compressed by CCTTI G4 (FAX4) [77], 30% of the bits used when compressed by JBIG [70], 30% of the bits employed by a JBIG2 compression [71], 30% of the bits used when compressed with the PWC method [79] and 0% when used the bilevel encoder implemented in the open source program DjVu [80]. Moreover, the average distortion in terms of information loss is $D_R = 9\%$ and the average Hausdorff distance between the original shape and the compressed shape is $D_H = 8$ pixels. Recall that FAX4, JBIG and JBIG2 are lossless compression algorithms.

Table 2.5: Experimental results for the Gatorbait dataset [2] (38 shapes).

Method	**Compression**					**Distortion**	
	FAX4 [77]	JBIG [70]	JBIG2 [71]	PWC [79]	DjVu [80]	D_R (2.7)	D_H (2.16)
DSaM#1	10%	24%	32%	39%	32%	10%	8
DSaM#2	10%	23%	30%	36%	30%	7%	7
DSaM#3	9%	22%	28%	35%	29%	5%	7
DSaM#4	8%	20%	26%	31%	26%	5%	8
DSaM#5	8%	19%	25%	30%	25%	3%	10
DSaM#6	7%	17%	22%	27%	22%	3%	11
PA#1	17%	39%	51%	62%	52%	11%	4
PA#2	11%	27%	35%	42%	35%	1%	4
PA#3	7%	18%	23%	28%	23%	2%	8
PA#4	6%	15%	20%	25%	20%	2%	12
PA#5	6%	14%	18%	22%	18%	6%	17

Figures 2.8-2.9 demonstrate some representative results of the proposed method with various line segment detection algorithms. One may observe that the compression based on the DSaM line segment detection preserves more details of the initial set, compared to the polygon approximation algorithm, whose result is more coarse.

Finally, the rate-distortion curves for the above experiments are presented in Figure 2.10. The blue line corresponds to the compression results based on DSaM [6], while the red line refers to the results based on a compression using polygon approximation [11]. As it can be observed, the DSaM method provides a clearly better performance. Note that the bits needed to encode the information for each method cannot be fixed directly, as they are affected by the tuning of the associated thresholds and parameters. Thus, equal bit rates cannot be established for DSaM and polygon approximation.

Table 2.6: Experimental results for the MPEG7 dataset [1] (1400 shapes).

Method	**Compression**					**Distortion**	
	FAX4 [77]	JBIG [70]	JBIG2 [71]	PWC [79]	DjVu [80]	D_R (2.7)	D_H (2.16)
DSaM#1	16%	30%	29%	44%	37%	9%	5
DSaM#2	15%	28%	28%	41%	35%	9%	5
DSaM#3	14%	27%	26%	39%	33%	9%	6
DSaM#4	13%	24%	24%	35%	30%	10%	7
DSaM#5	12%	23%	22%	34%	28%	10%	7
DSaM#6	10%	20%	19%	29%	25%	12%	9
PA#1	25%	47%	46%	68%	58%	7%	2
PA#2	17%	32%	32%	47%	40%	4%	3
PA#3	11%	21%	21%	31%	26%	7%	6
PA#4	9%	18%	18%	27%	23%	9%	9
PA#5	8%	16%	16%	24%	20%	12%	12

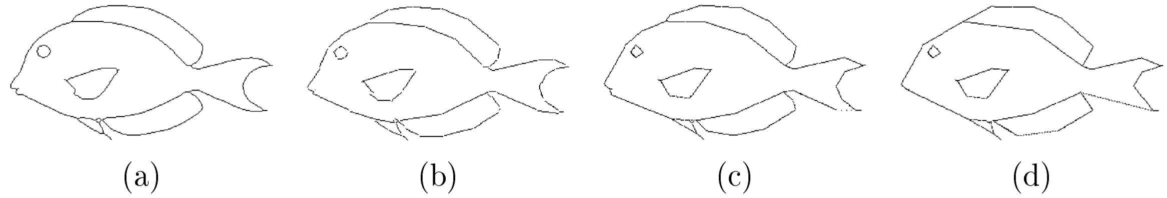


Figure 2.8: Representative results of the reconstruction method on the Gatorbait [2] dataset. (a) The original image. Results extracted with (b) DSaM [6], (c) polygon approximation [10] with automatic tuning (d) polygon approximation [10] with threshold value set to 5 pixels.

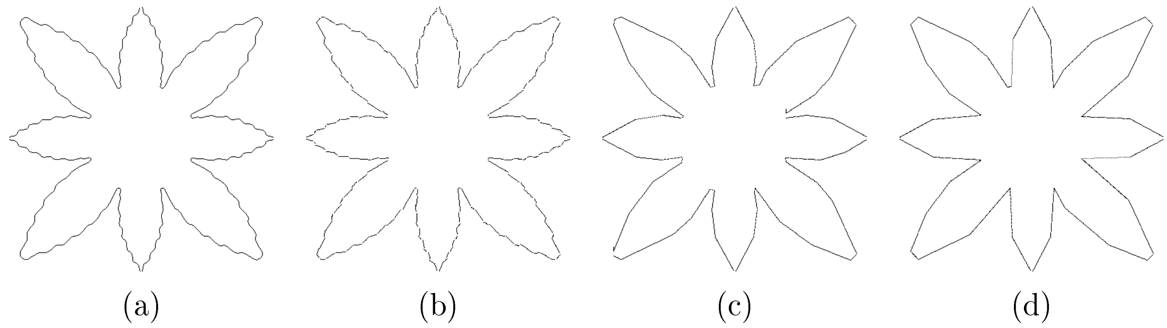
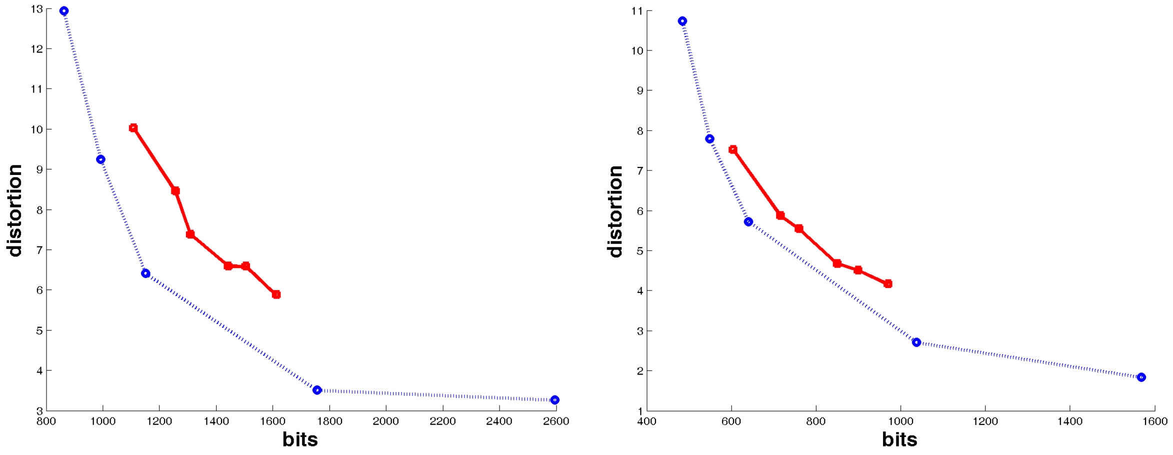


Figure 2.9: Representative results of the reconstruction method on the MPEG7 [1] dataset. (a) The original image. Results extracted with (b) DSaM [6], (c) polygon approximation [10] with automatic tuning (d) polygon approximation [10] with threshold value set to 5 pixels.

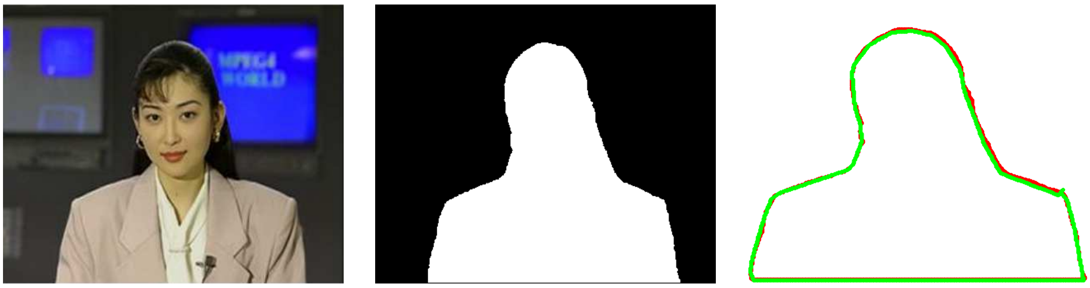
Another application of shape coding is the description of Video Object Planes (VOP) in MPEg4 [65] standard. In brief, a VOP is a region of image that describes an object. Through VOP, MPEG4 standard manages to encode independent objects. Figures 2.11



(a) (b)

Figure 2.10: Rate-distortion curves for (a) the Gatorbait dataset [2] and (b) the MPEG7 dataset [1]. The blue line corresponds to the compression results based on DSaM [6] and the red line refers to polygon approximation [11].

a, b demonstrate an example of a video frame and its corresponding VOP. Again, we performed the same experimental investigation regarding the efficiency of the proposed encoding scheme. Figure 2.11 c demonstrates the reconstruction result of the contour, based on the DSaM method. The red points depict the initial shape and the and the green points show the contour reconstructed with our method. Table 2.7 presents the results of the comparison regarding the VOP encoding of figure 2.11 b.



(a) (b) (c)

Figure 2.11: An example of a video object plane. (a) The initial image, (b) the video object plane mask and (c) the reconstruction result with the DSaM method. The red points depict the initial shape and and the green points show the contour reconstructed with our method.

One may observe that the proposed encoding framework provides satisfactory results in terms of compression, while offering low distortion. The DSaM method provides similar or better results compared to the polygon approximation that is used in the MPEG standard, in terms of compression, but with far lower distortion. Also, DSaM manages to significantly improve the compression rate providing an image quality (in terms of distortion) similar to the lossless algorithms.

Table 2.7: Experimental results for the VOP of figure 2.11.

Method	**Compression**					**Distortion**	
	FAX4 [77]	JBIG [70]	JBIG2 [71]	PWC [79]	DjVu [80]	D_R (2.7)	D_H (2.16)
DSaM#1	16%	44%	42%	66%	56%	169%	4
DSaM#2	12%	32%	30%	47%	40%	151%	4
DSaM#3	10%	27%	26%	40%	34%	178%	5
DSaM#4	9%	24%	23%	36%	30%	193%	4
DSaM#5	9%	23%	22%	35%	29%	261%	5
DSaM#6	7%	20%	19%	30%	25%	267%	4
PA#1	17%	45%	43%	68%	58%	105%	4
PA#2	10%	27%	26%	40%	34%	117%	3
PA#3	6%	16%	15%	24%	20%	213%	5
PA#4	5%	15%	14%	22%	19%	302%	7
PA#5	5%	14%	13%	21%	18%	345%	9

2.4 Retinal Fundus Image Feature Characterization

2.4.1 Introduction

The detection and characterization of various topological features of the retinal vessels is an important step in retinal image processing algorithms within an autonomous diagnosis system. A deviation from common topological feature patterns may be an indicator of anomaly. A comprehensive study may be found in [81]. In a typical retinal vessel structure, more than 100 *junctions* may be present [82], a fact that makes the manual characterization a tedious and time consuming task. Typical retinal features are presented in figure 2.12.

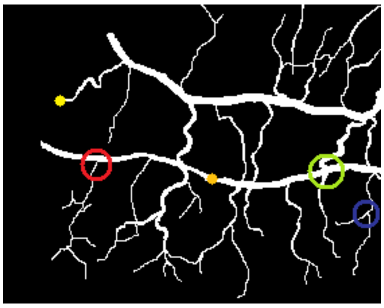


Figure 2.12: The different features that the proposed algorithm can detect. The yellow point is an *end-point*, the orange point is an *interior-point* and the green point is a *crossover*. All the other points are *junctions* (a *T-junction* is shown in red, and a *bifurcation* is shown in blue). The image is better viewed in colour.

In the current investigation, three types of features are detected: *end-points* (points at the extremities of the vessels), *interior-points* (points along a vessel), *junctions* (a new vessel is a branch of a longer one - *T-junctions* or a vessel is split into two or more new vessels - *bifurcation*) and *crossovers* (one vessel passes over another). Please note that further processing is needed to distinguish between a *crossover* and a *bifurcation*.

A methodology that extracts features from the retinal fundus image and characterizes them will be presented. The goal is to detect the intersection points between the vessels, as they could provide useful information to an automatic diagnosis system.

2.4.2 The algorithm

The first step of our method is to extract the line segments that model the center line of the vessels. Thus, it is crucial that an accurate preprocessing step towards that direction is applied. For each line segment, its extreme points are the points that have the largest distance from the corresponding extreme points of the same line cluster.

In figure 2.13, the points $\mathbf{x}$ (summarizing the vessel structure) are depicted with red and black color, while the corresponding extreme points $\mathbf{y}$ are presented with green and blue dots. A rule is defined to characterize a point as *end point* or *junction* or *crossover*. Let $\mathcal{C}(\mathbf{x})$ be the index of the line cluster point where $\mathbf{x}$ belongs to. In figure 2.13, two line clusters are shown ($\mathcal{C}(\mathbf{x}) = 1$ and $\mathcal{C}(\mathbf{x}) = 2$). To define the neighborhood of extreme points, a neighborhood radius threshold is defined as

$$T_n = w * \bar{d} \tag{2.9}$$

where $\bar{d}$ is the mean distance between all the nearest neighbors and w is a constant that is learned, as explained later in this section. Thus, for an extreme point $\mathbf{y}$ the neighborhood $\mathcal{N}(\mathbf{y})$ is the set of the points $\mathbf{x}$ such that $||\mathbf{x} - \mathbf{y}|| \leq T_n$. Note that $\mathbf{y} \in \mathcal{N}(\mathbf{y})$. In order to characterize point $\mathbf{y}$, we define $\mathcal{CN}(\mathbf{y})$ as the number of distinct line clusters that points $\mathbf{x} \in \mathcal{N}(\mathbf{y})$ belong to.

In the example shown in figure 2.13, the studied extreme point $\mathbf{y}$ is the green one, while its neighborhood $\mathcal{N}(\mathbf{y})$ is defined by a circle centered at $\mathbf{y}$ with radius T_n. Red and black points lying within that circle are the neighbors of $\mathbf{y}$. Those points belong either to line cluster 1 or to line cluster 2 and thus $\mathcal{CN}(\mathbf{y}) = 2$. If all neighbors of $\mathbf{y}$ belong to the same line cluster with that of $\mathbf{y}$, then $\mathbf{y}$ would be an *end-point* ($\mathcal{CN}(\mathbf{y}) = 1$). In case where $\mathcal{CN}(\mathbf{y}) > 2$, $\mathbf{y}$ would be a *junction* or a *crossover*. The algorithm in its current form does not discriminate between them.

A special case occurs when $\mathcal{CN}(\mathbf{y}) = 2$, where the studied point $\mathbf{y}$ is either an *interior-point* or a *junction* (*T-junction*). In that case, further elaboration is needed to characterize the extreme point by examining whether $\mathcal{N}(\mathbf{y})$ contains an extreme point or not. Thus, if $\mathbf{y}$ belongs also to the neighborhood of $\mathbf{y}'$, with $\mathbf{y}'$ denoting the neighbor of $\mathbf{y}$, then $\mathbf{y}$ is an *interior-point*, otherwise it is a *junction* (*T-junction*). Please note that the neighboring relationship we are describing in that section is **not** reflective. For example in figure 2.13, the green point, which is one of the extreme points of cluster 1, is neighbor to cluster 2 (as there are some points of cluster 2 within the yellow circle that defines the neighborhood of the green point). However, none of the extreme points of cluster 2 contain a point from cluster 1 in their neighborhood, and thus cluster 1 is not neighboring to cluster 2.

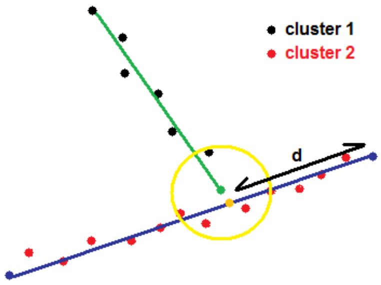


Figure 2.13: An instance of the point characterization algorithm. Points $\mathbf{x}$ (in red and black) correspond to the thinned lines of the extracted vessels. Green and blue points are the extreme points $\mathbf{y}$ computed by our DSaM algorithm. The yellow circle demonstrates the neighborhood of that point ($\mathcal{N}(\mathbf{y})$). Red and black points lying in that circle are considered as neighbors of that extreme point. In that case, those points belong either to line cluster with index 1 or to line cluster with index 2. Thus, $\mathcal{CN}(\mathbf{y}) = 2$. The orange point corresponds to the nearest neighbor of $\mathbf{y}$ among the points of neighbor line cluster (black points). d is the minimum distance between the aforementioned nearest neighbor and the extreme points of its line cluster.

In our example in figure 2.13, this means that one of the two blue points would be inside the yellow circle. In case that $\mathcal{N}(\mathbf{y})$ contains no extreme point, as shown in figure 2.13, then we compute the minimum distance (denoted by d in figure 2.13) between the nearest neighbor (orange point in figure 2.13) of $\mathbf{y}$ among the points of the other neighboring line cluster (black points in figure 2.13) and the corresponding extreme points (blue points in figure 2.13). If $d \leq \bar{d}$, then $\mathbf{y}$ is a (*junction*) (*T-junction*), otherwise it is an *interior-point.*

A detailed description of the rules used to characterize an extreme point $\mathbf{y}$ is presented in Algorithm 6.

2.4.3 Numerical evaluation

To investigate the accuracy of the proposed algorithm, experiments were conducted on the DRIVE database [83], which includes 40 retinal images along with their manual extraction of the vessels. The ground truth used in [82], [84] was also employed. In our experiments, the manual segmentations were employed, as the scope of our algorithm is to detect *junctions*, *crossovers* and *end points*. The reader should refer to [85] or [86] for a detailed vessel extraction algorithm, which is a preprocessing step of the whole chain. At first a Canny edge detector [44] is applied to extract the borders of the vessels and then a thinning algorithm [87] is used, to extract the center line of the vessels. In figure 2.14(a), the original image is shown. Figure 2.14(b) presents the manual segmentation, while the data used in our retinal parsing algorithm are shown in figure 2.14(c).

Note that since the ground truth refers to the original vessels and not to their center lines, which is the input of our method, we determined a value T_{conf} that defines a confidence region around a ground truth point. A computed point is considered to match a ground truth point if it lies in its confidence region. In our experiments, T_{conf} is defined

```
input: An extreme point $\mathbf{y}$ computed by the DSaM algorithm and the corresponding set of vessel skeleton points $\mathbf{x} \in \mathcal{N}(\mathbf{y})$.
output: The label of $\mathbf{y}$.
if $\mathcal{CN}(\mathbf{y}) = 1$ then
  $\mathbf{y}$ is an end-point.
else if $\mathcal{CN}(\mathbf{y}) = 2$ then
  $\mathbf{y}$ is either a junction or an interior-point.
  $\mathcal{Q} = \{\mathbf{x} \in \mathcal{N}(\mathbf{y}) | \mathcal{C}(\mathbf{x}) \neq \mathcal{C}(\mathbf{y})\}$
  $\mathbf{z} = \arg\min_{\mathbf{x} \in \mathcal{Q}} \{|\mathbf{x} - \mathbf{y}|\}$
  $d = |\mathbf{y} - \mathbf{z}|$.
  if $d \leq \bar{d}$ then
    if the line cluster of points of $\mathcal{Q}$ is equal to $\mathcal{C}(\mathbf{y})$ then
      $\mathbf{y}$ is an interior-point
    else
      $\mathbf{y}$ is a junction (T-junction).
  else
    $\mathbf{y}$ is a junction (T-junction).
else if $\mathcal{CN}(\mathbf{y}) > 2$ then
  $\mathbf{y}$ is a junction (bifurcation) or a crossover.
```

Algorithm 6: Rules for vessel features characterization

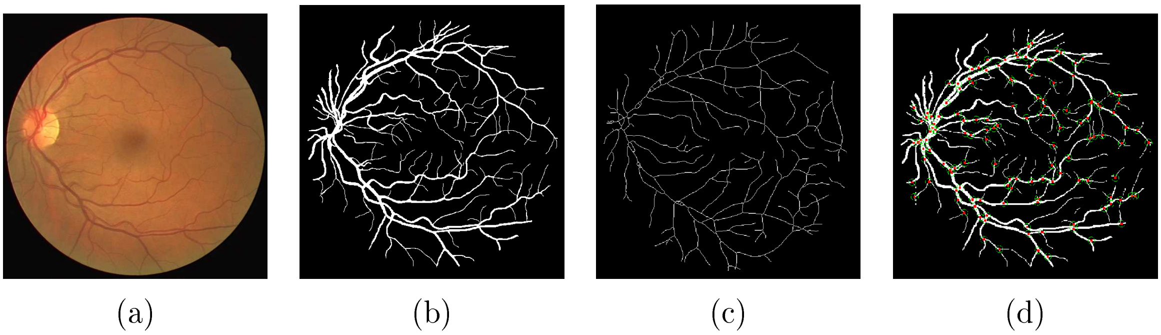


Figure 2.14: (a) The original retinal image. (b) The manual segmentation of the vessels in (a). (c) Result of thinning the image in (b). (d) The confidence regions depicted as circles with a radius equal to 1% of the diagonal of the bounding box of the original set. The figure is better seen in color.

as a percentage (1%) of the length of the diagonal of the bounding box of points $\mathbf{x}$. Figure 2.14(d) shows the confidence regions depicted as circles with a radius equal to T_{conf}. To establish a robust value for constant w (eq. (2.9)), the precision and recall rates were computed for values of w between 1.8 and 4.0 with a step of 0.8. Then the F measure (harmonic mean) was calculated as

$$F = 2\frac{PR}{R+P}, \tag{2.10}$$

where P is the precision and R is the recall:

$$P = \frac{TP}{TP + FP}, \tag{2.11}$$

$$R = \frac{TP}{TP + FN}, \tag{2.12}$$

where TP is the number of true positives, that is, the number of relevant items retrieved, FP is the number of false positives, that is, the number of irrelevant items retrieved and FN is the number of false negatives, that is, the number of relevant items not retrieved.

Figure 2.15 shows the plot of F measure for various values of parameter w in eq. (2.9). In our experiments, the F measure takes its maximum value for $w = 2.9$. The corresponding point is depicted with a black square in figure 2.15. In that case, the corresponding precision is equal to 91.59% while the recall is 98.58%. The value of $\bar{d}$ computed from our experimental data is approximately equal to $\sqrt{2}$, which leads to $T_n = 4.10$, eq. (2.9), corresponding to a neighborhood radius equal to 4 pixels.

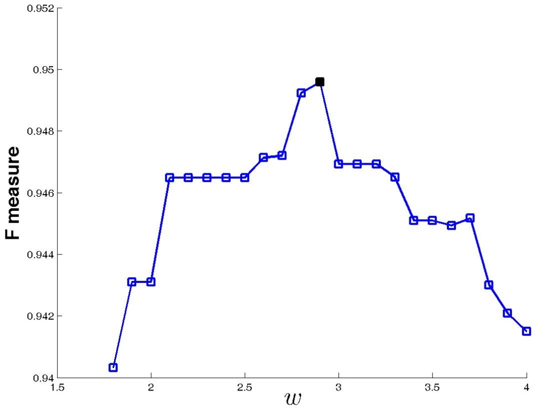


Figure 2.15: The F measure, (2.10), for various values of parameter w in (2.9). The black square indicates the point that corresponds to the maximum value of F measure. This value ($F = 0.95$) occurs for $w = 2.9$ and provides a precision rate of 91.59% and a recall rate of 98.58%. More details are given in section 2.4.

The mean execution time was 109 sec for the extraction of the line segments and 12 sec for the extraction and characterization of features using Matlab on a typical Dual Core 2x2.50 GHz machine with 2.0 GB RAM.

2.5 Elimination of outliers from 2D point sets using the Helmholtz principle

2.5.1 Introduction

The modeling and removal of outliers from a set of points has been an active research topic for many decades in image processing and computer vision and a variety of algorithms have been proposed [88]. They may be as simple as the median filter to more elaborate which are based on random sampling, like RANSAC [36] or probabilistic models [89].

The Gaussian assumption for data generation has been widely adopted but it is appropriate only for sparse outlier distributions. In general, it involves the comparison of Euclidean distances between points with the mean of the distribution expanded by a number of standard deviation [90]. Kernel density estimators-based methods provide a probabilistic approach to determine if a point belongs to the uncorrupted set and are inherently related to clustering or classification techniques that separate pure data from outliers [91, 92].

The number of neighbors of a point is a key issue in characterizing it as outlier [93]. The main hypothesis is that pure data are more densely populated than outlying points and many algorithms have been designed based on this idea. The adopted strategy consists in defining a neighborhood for each point, determining a feature that characterizes the neighborhood and rejecting points with features having a value smaller than a threshold. In [12], the number of common neighbors is defined as a similarity index between points and points with neighborhood size smaller than a threshold are rejected as outliers. An octree is used in [94] to cluster points and an implicit quadric is fit to them to smooth out outliers.

Inspired by the geometric Gestalt theory, which addresses the answer to the fundamental problem in computer vision: ”How to arrive at global percepts from the local, atomic information contained in an image?” [95], Desolneux et al. proposed methods for detecting geometric structures [39] and edges [96] in images by a parameter free method based on the Helmholtz principle [97]. The principle states that an observed geometric structure is perceptually meaningful if its number of occurrences is very small in a random situation. In this context, geometric structures are characterized as large deviations from randomness. The principle is accompanied by an *a contrario* assumption against which structures are detected.

In this section, we propose an algorithm for outlier elimination and structure extraction from 2D point clouds based on the Helmholtz principle. The main difference with the methods in [39, 96] is that the input to the algorithm is not an image whose pixels lay on a regular grid but a set of scattered points irregularly distributed in space. To overcome this limitation, at first, the point set is approximated by a locally linear manifold consisting of a set of line segments. We show that the lengths of the line segments follow a Pareto distribution which is our *a contrario* model.

2.5.2 The Helmholtz principle

The Helmholtz principle is a general hypothesis of the Gestalt theory [95] interpreting the way human perception works. Intuitively, it states that if we take into consideration randomness as the normal case for our observations then meaningful features and interesting events should not be expected. Consequently, if they are observed they should appear with a small probability. Moreover, the small probability of observing an event is not a factor to consider it as meaningful (or true observation not generated by noise). Take as an example the toss of an unbiased coin. The probability of getting a head (H) or a tail (T) is 1/2 respectively. If we toss the coin successively N times then the probability of observing any of the possible sequences of H and T is $(1/2)^N$, which is a decreasing function of N and approaches zero as $N \to \infty$. More specifically, the following sequences:

$$S_1 = \underbrace{HTHHTTTHTHHTHT...H}_{N\,\text{times}}$$

$$S_2 = \underbrace{HHHHHHHHHHHHH...H}_{N\,\text{times}}$$

have equal probabilities of appearance. However, S_2 is not expected to appear for an unbiased coin. Therefore, the low probability of an event may not characterize it as a deviation from randomness, as its probability may not truly model the randomness of an event.

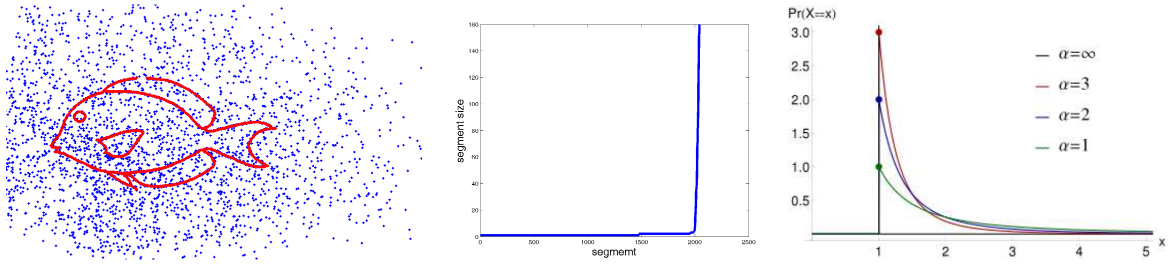
Using the same sequences S_1 and S_2, we may define another pair of random variables n_H and n_T modeling the number of H and T present in a sequence. Since the coin is unbiased, the expectations of both variables is $N/2$. Although this is confirmed in S_1, in sequence S_2 the observed values for n_H and n_T is N and 0 largely deviating from the expected values.

The above observations lead to the conclusion that the small probability of an event may not be an accurate indication that this event is meaningful and we need to take into consideration that the model we use to validate an event describes the randomness of all possible observations. Turning back to the last example of the coin toss, randomness was modeled only by counting the number of H and T in a sequence and not by the probability of a sequence to appear. Taking both issues into account yields the complete model used to describe randomness which is called *a contrario* model.

2.5.3 The algorithm

Let $X = \{\mathbf{x}_i\}_{i=1,\ldots,N}$ be a set of observed 2D points including both data points and outliers (Fig. 2.16(a)).

In order to eliminate the outliers, we compute at first an approximation of the point set by line segments. To this end, the direct split-and-merge (DSaM) algorithm presented in chapter 1 can be employed, which summarizes any point set by a set of the major axes of highly eccentric ellipses. The number of ellipses is automatically determined by the



(a) (b) (c)

Figure 2.16: (a) A a set of points (in red color) degraded by equal in number outliers (in blue color). (b) The distribution of the sorted lengths of the line segments approximating the point set of (a) using a line segment detection algorithm. The horizontal axis represents the indices of the segments and the vertical axis represents the lengths. (c) The Pareto distribution for various values of the parameter a with $b = 1$.

algorithm and it depends on the number and the spatial distribution of the point sets. In the example of Fig. 2.16(a), the large number of outliers will provide a large number of line segments with relatively small lengths (due to noise) and a smaller number of line segments with larger lengths (due to both the uncorrupted data and the noise around them). This distribution of the lengths of the line segments, shown in Fig. 2.16(b) after sorting the lengths in increasing order, leads to consider an *a contrario* probabilistic model of the lengths by a Pareto distribution with density [98]:

$$\text{Pareto}(x; a, b) = \begin{cases} \frac{ab^a}{x^{a+1}}, & x \geq b \\ 0, & x < b \end{cases} \tag{2.13}$$

where $b > 0$ and a is a parameter controlling the slope of the curve (Fig. 2.16(c)). Herein, the length of the segment is considered in terms of the number of the points contributed to its computation. The line segment detection algorithm provides line segments with uniformly distributed points. Therefore, the length of a segment is equivalent to the number of points belonging to it.

The purpose of the *a contrario* model is to describe the randomness of the data. However, it might be possible that outliers are organized in such a way that they generate short line segments that are not part of the desired structure. The Pareto distribution computes the probability that a segment of a given length appears in the observations. In an analogy to the coin toss example, this event may be expressed by the probability of getting H or T (with more possible outcomes, which are the lengths of line segments). By expanding our initial intuition regarding the rareness of the observation, it is possible that segments due to outliers would be isolated, as the intrinsic feature of noise is to be structureless. Therefore, in order to set up the *a contrario* model, the neighborhood of a line segment should be defined to account for isolated structures.

Each line segment has a starting and an ending point. The neighborhood $\mathcal{N}(\beta)$ of a segment β is defined as the set of all those segments β_j whose starting/ending points are located at a distance less than a threshold to the starting/ending points of β:

$$\mathcal{N}(\beta) = \{\beta_j : |\beta^k - \beta_j^l| \leq T,\ k, l \in \{s, t\}\}, \tag{2.14}$$

where the superscripts $\{s, t\}$ indicate the starting or the ending point of a segment. Figure 2.17 demonstrates the definition of the segment neighborhood for a given line segment. The neighborhood can be iteratively expanded to take into account the neighbors of neighbors up to a fixed depth.

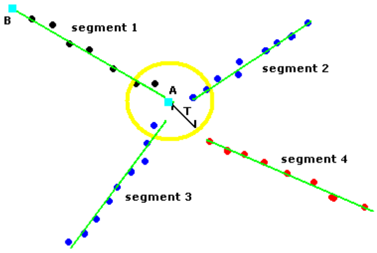


Figure 2.17: An example of the definition of the neighborhood of a segment. Points A and B (cyan squares) are the starting/ending points of segment 1. The yellow circle with radius T determines the neighborhood. Line segments 2 and 3 are part of the neighborhood while segment 4 is not. The same configuration applies to point B.

Therefore, the *a contrario* model is based on the assumption that a line segment is more probable to be a true observation if its neighboring segments have large lengths. This may be expressed by the likelihood:

$$\mathcal{L}(\beta) = \prod_{\beta_j \in \mathcal{N}(\beta)} \text{Pareto}(\beta_j; a, b). \tag{2.15}$$

Consequently, if $\mathcal{L}(\beta) < \epsilon$ we consider the line segments to be a true observation. The threshold is automatically determined as $\epsilon = 10^{-a}/D$, where D is the maximum depth of the neighborhood expansion. It may be observed that the value of ϵ is independent from the data. Thus, following the rationale in [40, 41, 42], it may be asserted that the proposed method is parameter free. The procedure is presented in Algorithm 7.

```
input: A set of points $X$, the depth of expansion $D$.
output: A set of points $Y$.
while convergence is not reached do
    Summarize $X$ by line segments (e.g. [6]). Let $B_i$ be the points contributing to
    segment $\beta_i$, for $i = 1 \dots N$.
    $Y = \emptyset$.
    for $i = 1 \dots N$ do
        if $\mathcal{L}(\beta_i) \leq 10^{-a}/D$ then
            $Y = Y \cup B_i$.
```

Algorithm 7: Outlier elimination based on the Helmholtz principle.

2.5.4 Numerical evaluation

To investigate the efficiency of the proposed method for outlier elimination, we used the Gatorbait database [2]. Degradation of the data set was artificially performed in the following way. For each point of the original data set, an outlier was generated by multiplying the coordinates of that point with a uniformly distributed random number in the interval $(0, 1]$. The number of outliers added was set equal to the number of pure data points. Moreover, the pure data were degraded by zero-mean additive Gaussian noise with an appropriate standard deviation in order to obtain a signal to noise ratio (SNR) of 55 dB (e.g. Fig. 2.16(a)). The algorithm was applied to 50 different realizations of outliers.

We conducted comparisons with a density-based method (DBScan [13]) and the algorithm of Xianchao et al. [12]. Let us also note that other established methods, such as the algorithm in [89], were also considered but they failed to provide an acceptable result in our framework of highly corrupted point sets. Finally, we also show the results of the simple, but in many cases powerful, median filter for image denoising to highlight the order of magnitude of the obtained accuracy with respect to a well known baseline.

To evaluate the results provided by the different algorithms we employed the Hausdorff distance between two sets of points X and Y:

$$d_{\mathcal{H}}(X, Y) = \max_{\mathbf{x} \in X} \min_{\mathbf{y} \in Y} \{|\mathbf{x} - \mathbf{y}|\}, \tag{2.16}$$

where X is the original set of points (the ground truth) and Y is the computed set of points after outliers removal.

Table 2.8 summarizes the performance of the compared methods. As it may be seen, our method may successfully recover the initial shape. Its maximum distance (10.3), although smaller than the other algorithms, is due to the fact that, in a few cases, parts of the pure data were pruned because the outliers were close to them. Moreover, we examined the sensitivity of our method to parameter a of the Pareto distribution by applying the algorithm using a variety of values for this parameter, namely $a = \{2, 3, 4, 5\}$. As it may be observed, the method is consistent and its performance does not depend on this parameter. Larger values of a may not be employed as the numerator in the Pareto distribution (2.13) increases beyond computer accuracy. As b is the mode of the distribution, we have set $b = 2$ in all of the experiments relying on the fact that we search line segments and any two points define a line segment. This relatively low value for b is not in favor of our algorithm, as the model accounts for less populated line segments which generally are due to noise. However, the results showed the robustness of the proposed approach.

Furthermore, it is worth noting that DBScan [13] needs tedious parameter tuning (performed here by trial and error) and the method in [12] did not detect many outliers laying near the shape contour. Representative results are shown in Fig. 2.18.

A second set of experiments addresses the problem of outlier elimination for line fitting. Following the same principles as in the previous experiments, a set of 500 collinear points

Table 2.8: Statistics on the Hausdorff distance (2.16) on the 38 shapes of the GatorBait100 data set [2]

Method	**mean**	**std**	**median**	**min**	**max**
Proposed method (a = 2)	6.12	1.3	5.8	4.2	10.3
Proposed method (a = 3)	6.07	1.3	5.8	4.1	10.3
Proposed method (a = 4)	6.05	1.4	5.6	4.0	10.3
Proposed method (a = 5)	5.99	1.3	5.7	4.1	10.3
DBScan [13]	12.62	3.1	11.2	10.5	23.1
Xianchao et al. [12]	84.59	19.6	83.0	51.4	129.7
Median Filter	208.17	17.0	208.4	174.9	243.8

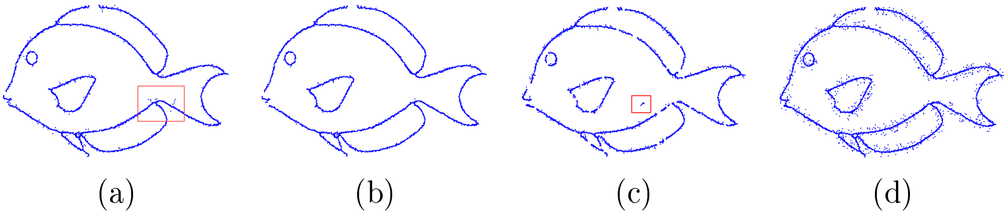
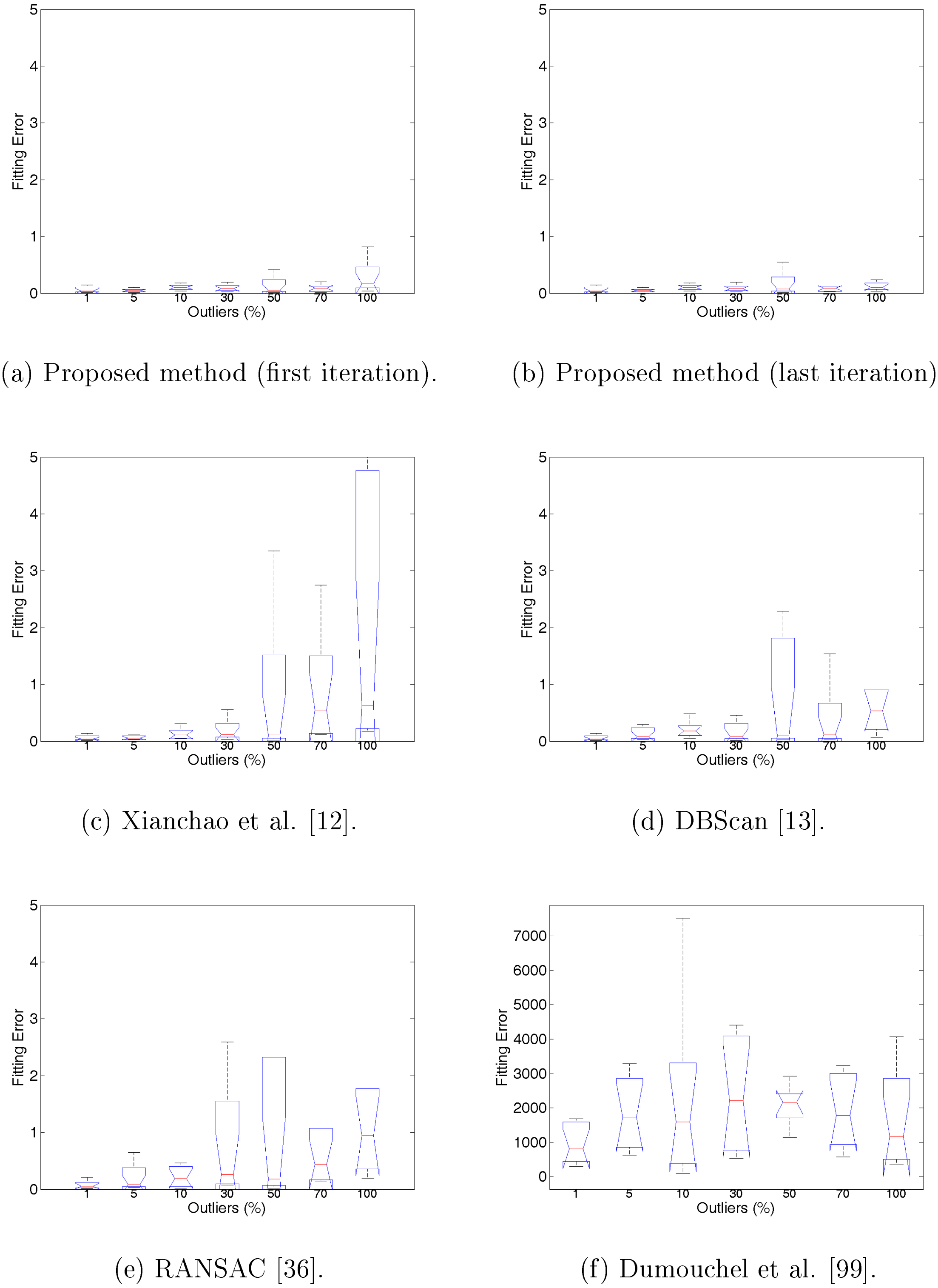


Figure 2.18: Outlier elimination from the data set of Fig. 2.16(a) by (a) the first and (b) the last iterations of the proposed method, (c) Xianchao et al. [12], (d) DBScan [13]. The red boxes highlight representative false points provided by the methods.

were corrupted by outliers and Gaussian noise. Various experiments were conducted with an increasing number of outliers at each configuration. In the more challenging setup, the number of outliers was equal to the number of points. Each experiment was repeated 50 times and statistics on the fitting error, in terms of Euclidean distance between the estimated and the true parameters of the lines were computed. The performances of the compared methods are shown in Fig. 2.19. For a more meaningful evaluation, we have also compared our method with two robust algorithms, namely RANSAC [36] and the robust regression method proposed in [99]. As it may be seen, our algorithm outperforms both of these methods which are established in the computer vision literature. Please notice the different scales in the abscissas in the graphs in Fig. 2.19 which clearly show the accuracy of the proposed algorithm as its maximum error, even in the more challenging scenario is less than one coordinate unit. On the other hand, only RANSAC is relatively competitive but its fitting errors are much more important.

A final set of experiments investigated the dependence of the proposed framework on the involved line segment detection algorithm. To this end, the Direct Split-and-Merge (DSaM) framework [6] and the widely used polygon approximation (PA) [10] [5] were employed in the corresponding step of Algorithm 7. In both cases, the parameters of the algorithm were set as $a = 2$, $b = 2$, $D = 3$. The test image of Figure 2.20(a) was degraded by zero-mean additive white Gaussian with varying standard deviation and then the various outlier elimination methods were compared. The algorithm was applied to 50 different realizations of outliers. Figure 2.20(b) demonstrates a degraded instance of



(a) Proposed method (first iteration).

(b) Proposed method (last iteration).

(c) Xianchao et al. [12].

(d) DBScan [13].

(e) RANSAC [36].

(f) Dumouchel et al. [99].

Figure 2.19: Boxplots of the line fitting errors for the compared methods. Notice the different scales at the abscissas.

the test image (SNR $= -1$dB). The overall results are shown in Table 2.9, where it may be observed that DSaM provides better results compared to PA, due to the fact that PA cannot compute valid line segments. Representative results of the line segment modeling are shown in figure 2.21. Notice that the PA is trapped by the outliers and produces a large number of short line segments, while DSaM manages to provide a valid model. This confirms that the proposed framework can be accurate independently of the line segment detection algorithm selected, provided that the latter establishes a valid model.

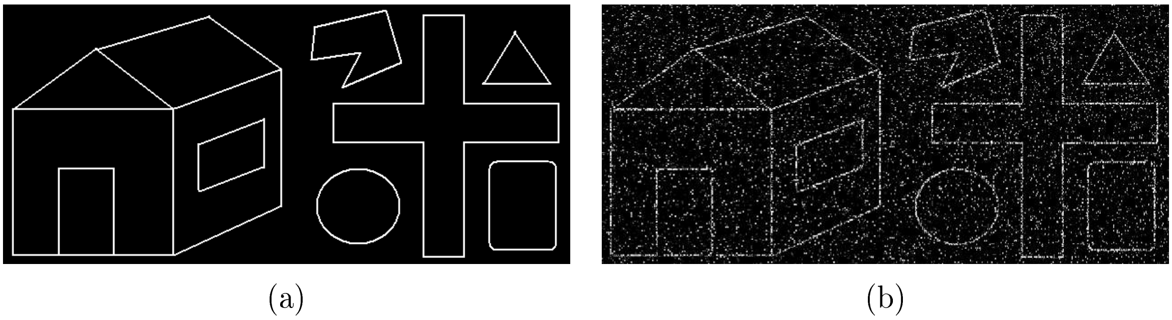


Figure 2.20: (a) A test image and (b) its degraded version at SNR = −1dB.

Table 2.9: Statistics on the Hausdorff distance (2.16) on the experiments based on the test image of figure 2.20(a).

Method	**mean**	**std**	**median**	**min**	**max**
Helmholtz + DSaM [6]	16.86	2.45	16.03	15.00	21.00
Helmholtz + PA [10]	81.02	19.50	88.41	46.84	94.87
DBScan [13]	64.97	43.22	91.83	2.24	98.81
Xianchao et al. [12]	29.29	19.22	28.00	10.00	49.24

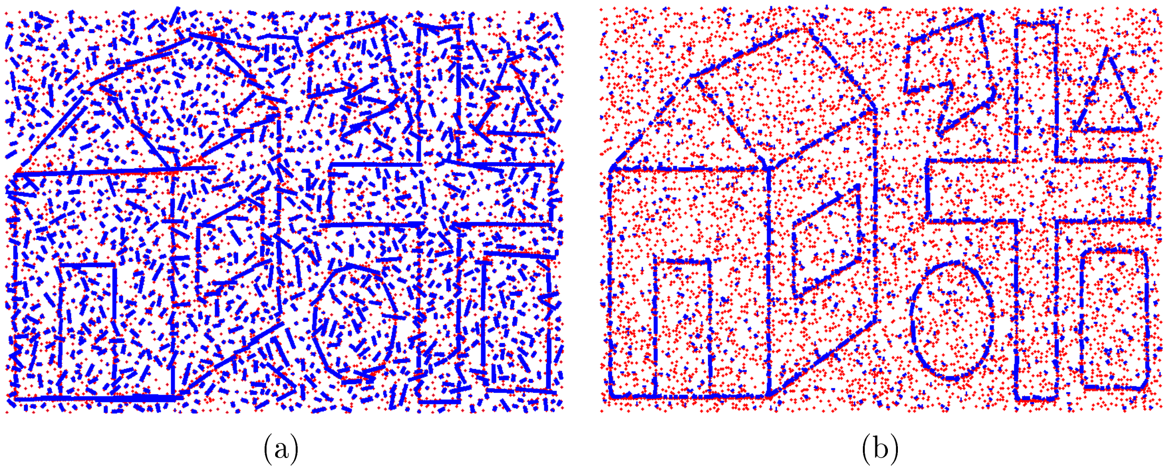


Figure 2.21: Line segment modeling of image in figure 2.20(b) computed (a) by DSaM[6], PA [10]. Notice that the PA is trapped by the outliers and produces a large number of short line segments, while DSaM manages to provide a valid model.

Part II

Image and Point set Registration

Chapter 3

Registering sets of points using Bayesian regression

3.1 Introduction

3.2 Registration of sets of points via regression

3.3 Experimental Results and Discussion

3.1 Introduction

Registration of two sets of points is a common step in many applications in computer vision, pattern recognition, image processing and medical image analysis. The problem consists in determining a geometrical transformation that brings two sets of points into alignment. This could be achieved, for instance, through the establishment of correspondences. However, the problem is not always well-posed and becomes more complicated by the existence of noise or outliers, making the determination of correspondences harder. Another drawback rises from the geometric transformation itself, as there may be an infinite number of allowed high dimensional mappings. Also, the definition of the similarity measure is an open issue, since one can choose from a variety of metrics.

Many methods have been proposed to solve the correspondence problem. A straightforward approach is based on the nearest neighbor criterion to establish correspondences, as in the Iterated Closest Point (ICP) algorithm [100]. However, despite its simplicity, this method results in many local minima, providing a suboptimal solution, and does not guarantee that the correspondence is one-to-one. Many variants of this algorithm have been proposed improving the behavior of the method in the presence of noise. A detailed review can be found in [101]. Nevertheless, in all cases, ICP methods necessitate a good initialization near to the optimal solution in order to prevent the energy function from getting trapped in local minima.

The Robust Point Matching (RPM) algorithm [14] relies on a deterministic annealing scheme. The algorithm applies the softassign principle [102] for matching and the thin-plate spline interpolation [15] for non-rigid mapping. The rationale is to transfer the assignment problem from a hard approach to a soft one, that is to define a probability for each assignment.

The Coherent Point Drift (CPD) algorithm was also proposed in [17], where the registration is treated as a Maximum Likelihood (ML) estimation problem with motion coherence constraints over the velocity field such that one point set moves coherently in order to be aligned with the other. In that case, transformation parameter estimation and determination of correspondences are simultaneously handled.

Mixture models were proposed as a framework to solve the registration problem (GMMReg) [103]. Each set of points is represented by a mixture of Gaussians and registration is defined as a problem of aligning the two mixtures. The L_2 metric is used as a measure of mixture alignment. An extension of the method using robust Student's-t modeling for the data was also proposed in [104].

Shape context was considered in [105], where an iterative algorithm is designed to account for the shape matching, registration and detection. The problem is formulated in terms of probabilistic inference using a generative model and the EM algorithm. Shape features are used in a data-driven technique to address the problem of initialization.

A technique for establishing correspondences is proposed in [106], where features of a 2-D point set which are invariant with respect to a projective transformation are extracted. The proposed algorithm is based on the comparison of two projective and permutation invariants of five-tuples of the points. The best-matched five-tuples are then used for the computation of the projective transformation and the one having the maximum number of corresponding points is used.

Moreover, in [107], a novel technique is introduced to solve the rigid point pattern matching problem in Euclidean spaces of any dimension. The point pattern matching is modeled as a weighted graph, where nodes represent points and the weights of the edges are equal to the Euclidean distances between nodes. The graph matching is formulated as a problem of finding a maximum probability configuration in a graphical model.

In [108], the notion of a neighborhood structure for the general point matching problem is introduced. Then, the point matching problem is formulated as an optimization problem to preserve local neighborhood structures during matching. The method has a simple graph matching interpretation, where each point is a node in the graph, and two nodes are connected by an edge if they are neighbors. The optimal match between two graphs is the one that maximizes the number of matched edges.

The majority of the registration methods mentioned so far, model the non-rigid mapping through a spline interpolation method, and in particular with the thin-plate spline (TPS) [15]. In this work, we consider the transformation parameter estimation issue as a regression problem and a Bayesian model, namely Relevance Vector Machine (RVM) [109] is used to solve this problem. We consider here the standard RVM although the method may employ other variants such as the twinned RVM [110] which applies double training or the multivariate RVM [111].

Our work is motivated by the pioneering research presented in [112] where correspondences are estimated using a softassign approach. Softassign is a technique for solving an assignment problem. As opposed to hard assignment, softassign weights each matching to indicate the quality of the correspondence. Hard assignment is the limit version of softassign. In the work herein, instead of solving the assignment problem based on the smallest distance, we utilize this distance to create a cost matrix that describes the cost of an assignment. Then, the correspondences are extracted with a combinatorial optimization algorithm, the Hungarian algorithm [113]. The rationale behind this algorithm is to assign a single task to a single worker, based on an assignment cost matrix, such that the total cost is minimum. The temporal complexity of the algorithm is polynomial, and in particular $\mathcal{O}(n^3)$. After the correspondence between points has been established, a Bayesian regression model (RVM) is used to infer the transformation parameters.

The Hungarian algorithm has also been used in [7], where a feature-based registration method is demonstrated. Points are assumed to describe a shape and a histogram (*shape context*) is calculated, describing the space distribution of points. This histogram is used to define the cost matrix of the Hungarian algorithm. Our work differentiates from [7] in the way the geometric transformation is treated. We estimate the transformation's parameters (both rigid and non rigid) through regression (RVM) while the latter method uses thin plate splines interpolation. Another substantial difference is that in our approach points are not considered as parts of a shape representation, since our method is more general.

The main contribution of this work is that using a regression framework based on RVM addresses the problem of eventual false correspondences with respect to methods relying on interpolation schemes, like TPS. More precisely, a single false correspondence may lead to a totally erroneous registration if TPS is used. For example, this is a case of the RPM [14], or the GMMReg [103] methods. In figure 3.1(a), the correct correspondences between the source and the target sets are represented by line segments. In figure 3.1(b), two points were falsely matched on purpose simulating a wrong correspondence solution. The result of TPS [15] is shown in figure 3.1(c), where large registration errors are present. The result of the proposed registration scheme based on RVM regression is shown in figure 3.1(d), where the registration is correct. TPS by its definition tries to minimize the total bending energy to provide a smooth model, which is an approach that restricts the capability of providing good results in areas where the correspondence is correctly established. On the other hand, RVM considers only the local neighborhood to extract the regression model, due to the priors it implies on each point. The closed form solution for the transformation model provided by RVM is continuous and locally smooth depending on the assignment solution and more importantly, it is robust to false matches. The correspondence estimation step used in this work is the Hungarian algorithm. Alternative methods could also be used to solve the correspondence problem, like the standard softassign approach [102].

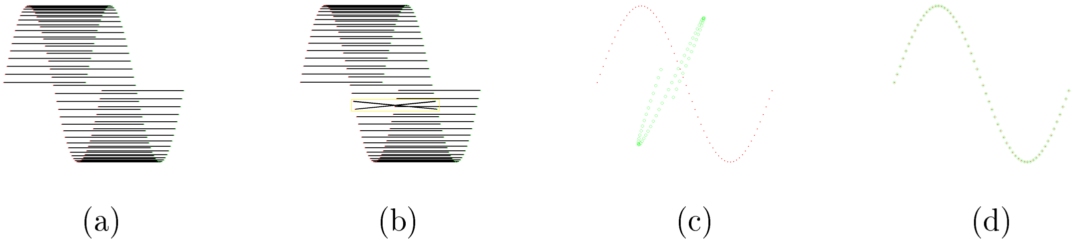

(a) (b) (c) (d)

Figure 3.1: A false matching simulation example, with a point set used in [14]. (a) Correct correspondences between the source and the target sets are represented by line segments. (b) Two points were falsely matched on purpose simulating a wrong correspondence solution. The yellow box depicts the false matched points. (c) The result of TPS [15]. Notice that large registration errors are present. (d) The result of the proposed registration scheme based on RVM regression. In this case the registration is correct.

The proposed method is similar in spirit with RPM [14] and CPD [17] in the sense that it employs a framework of iteratively updating the correspondence and the estimation of the transformation parameters. Both RPM and CPD are based on an expectation-maximization (EM) [114] framework. In RPM and CPD the E-step estimates soft correspondences through a posterior distribution. In our method the E-step involves any correspondence estimation algorithm, which in our case is the Hungarian algorithm. A major difference in our approach with respect to RPM and CPD is that in the M-step the transformation is estimated using Bayesian regression (RVM). On the other hand RPM uses TPS interpolation. Moreover, RVM provides a closed form transformation both for the rigid and non-rigid cases compared to CPD, where the two cases have to be modeled with different set of parameters [17]. A modeling of a rigid registration case with a non-rigid model may provide inaccuracies in the CPD result.

We evaluated our method by comparing it with the CPD [17], RPM [14] and GMMReg [103] algorithms for both rigid and non-rigid transformations. The results indicate that our method is more accurate than the state of the art methods compared concerning the robustness against false matching during the correspondence estimation step and the parametrization of the method. The innovation of our method is that by utilizing a robust correspondence estimation algorithm initially, we could relax the constraints imposed in the transformation modeling step so as to handle any erroneously matched points.

3.2 Registration of sets of points via regression

In a point set registration problem two sets of points are involved. The source point set $X = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^{N_x}$ and the target point set $T = \{\mathbf{t}_i \in \mathbb{R}^d\}_{i=1}^{N_t}$. In our experiments, we assume that $N_x = N_t = N$. In case the two sets have different cardinalities, we add extra points (as described in appendix I). In our method the registration transformation is modeled by a RVM. However, one would observe that the RVM described in appendix II is defined for univariate output vectors. In other words, the target variable has to be a scalar. However, in our case $\mathbf{t}_i \in \mathbb{R}^d$, and thus, in order to overcome this difficulty, we used d distinct RVMs, one for each dimension k. Alternatively, one could use a multivariate RVM, as described in [111] or the twinned RVM [110].

Eventually, the proposed model is a *vector valued* function $\mathcal{T}$, having parameters W, representing the geometric transformation bringing set X into alignment with set T. Thus, ideally we would have for every point $\mathbf{t}_i \in T$, $i = 1, ..., N$:

$$\mathbf{t}_i = \mathcal{T}(\mathbf{x}_{C_i}; W) = [\mathcal{T}^1(\mathbf{x}_{C_i}; \mathbf{w}^1), ..., \mathcal{T}^k(\mathbf{x}_{C_i}; \mathbf{w}^k), ..., \mathcal{T}^d(\mathbf{x}_{C_i}; \mathbf{w}^d)]^T, \tag{3.1}$$

where $W = \{\mathbf{w}^k \in \mathbb{R}^N\}_{k=1}^d$, with $\mathbf{w}^k$ being the weight vector of dimension N for the k-th RVM, $\mathcal{T}^k$ is a RVM as described by (5.1) in appendix II and C_i is the index of the point in X corresponding to the i-th point of T. In other words, the ideal transformation is

$$t_i^k = \mathcal{T}^k(\mathbf{x}_{C_i}; \mathbf{w}^k) \; i = 1, \ldots, N, \; k = 1, \ldots, d \tag{3.2}$$

with t_i^k representing the k-th component of point $\mathbf{t}_i \in T$.

The proposed method consists of an iterative scheme, that, at each iteration alternates between the method for establishing correspondences (e.g. Hungarian algorithm) and the method for estimating the registration transformation (RVM training). The corresponding objective function that is minimized has the following form:

$$J(\delta, W) = \sum_{i=1}^{N}\sum_{j=1}^{N} \delta_{ij} \mathcal{C}_{\mathbf{x}_i, \mathcal{T}(\mathbf{x}_j; W)} + \sum_{i=1}^{N}\sum_{j=1}^{N} \delta_{ij} ||\mathbf{t}_i - \mathcal{T}(\mathbf{x}_j; W)||^2. \tag{3.3}$$

Its optimization involves two steps at each iteration. In the first step, we assume a known registration transformation $\mathcal{T}$ (RVM) and try to estimate the optimal correspondences δ_{ij} with the Hungarian algorithm. Thus, the objective function is minimized with respect to δ_{ij}, $\forall \; i, j$. In the second step, the correspondences δ_{ij} are fixed to the values computed in the first step and a RVM training process takes place to update the registration transformation $\mathcal{T}$ in order to match the estimated correspondences. Thus, in this second step, the objective function is minimized with respect to the set of weights $W = \{\mathbf{w}^k \in \mathbb{R}^N\}_{k=1}^d$. Since both computational steps at each iteration minimize the objective function J, the whole iterative process converges to a minimum of J.

The overall procedure is presented in Algorithm 8. Each iteration of the registration algorithm involves two steps. At first, correspondences between points of the source set X and the target set T are estimated by the Hungarian algorithm and then based on these correspondences, d RVMs are trained, one per dimension, to solve the regression problem of transforming the source set to the target set. We initialize the registration transformation as the identity mapping.

```
1: Initialize the registration transformation as the identity mapping and select the type of basis function $\phi_i(x)$, $i = 1, ..., N$ for the RVM.
2: Determine the correspondences between sets of points $X$, $T$.
3: Calculate distance matrix $\mathcal{C}_{ij} = \|\mathcal{T}(\mathbf{x}_i; W) - \mathbf{t}_j\| \; \forall \, i = 1, \dots N, \; j = 1, \dots, N$.
4: Solve the assignment problem with the Hungarian algorithm, where $\mathcal{C}$ is an assignment cost matrix.
5: **Transformation parameters estimation - train one RVM per dimension of point set** X.
6: **for all** RVM$_k$, $k = 1, \dots, d$ **do**
7:    Calculate $\mathbf{m}^k$ and $\mathbf{\Sigma}^k$ by (5.7) and (5.8).
8:    Calculate $a_i^k, \beta^k, \gamma_i^k$ by (5.9), (5.10) and (5.11).
9:    Iterate steps (3.1) and (3.2) until convergence to obtain the new RVM$_k$, with $\mathbf{w}^k = \mathbf{m}^k$.
10: Iterate steps 2, 3 until convergence of the objective function $J(\delta, W)$ (3.3).
```

Algorithm 8: The RVM-Hungarian method for registration of sets of points

3.3 Experimental Results and Discussion

In order to evaluate our method, several experiments were conducted in a collection of sets of points, firstly used in [112], and widely used in the related literature (figure 3.2). The algorithm was tested both for its accuracy and its robustness to noise. Experiments with real data were also conducted. For that purpose we used the 2D range data from [16] (figure 3.3(a)) and the 3D face of [17] was also used in our experiments (figure 3.3(b)).

Experiments are divided into two types, according to the transformation type (either rigid or non rigid). In case of non rigid transformation, the non rigid deformation was followed by a rigid one, to make the problem more challenging. In that case the whole transformation remains non-rigid. In all cases the registration transformation was initialized to the identity mapping. The angle of the rigid transformation varied between $[0°, 10°]$, while the translation, varied between $[-0.2, 0.2] \times [-0.2, 0.2]$. The registration error is defined as the Euclidean distance between the reference point and its corresponding registered. Points of figure 3.2 range in $[0, 1] \times [0, 1]$, while those of figure 3.3(a) range in $[-40, 10] \times [-10, 30]$ and of figure's 3.3(b) in $[-2, 2] \times [-2, 2] \times [-2, 2]$.

In our implementation, different kernels were examined (Gaussian, Student's t-kernel and Laplacian) as described in [109] and implemented in [115]. The Laplacian kernel, $K(x, y) = e^{-\frac{|x-y|}{\sigma}}$, proved to be the most efficient model for the tested input data shown in figure 3.2. However, the differences are not considerable as the registration accuracies of the compared methods differ at the third decimal digit. Table 3.1 presents the registration error statistics of the rigid case, while table 3.2 demonstrates the results of the non rigid case. In all cases, variable kernel widths were used in the range between 5% and 30% of the mean variance of the reference set. The Laplacian kernel proved to be less sensitive to changes in the variations of the kernel width compared to the Gaussian and Student's

t-kernels.

Table 3.1: Registration error statistics for rigid transformations using different kernels on the shapes of figure 3.2. The kernel width varies between 5% and 30% of the mean variance of the reference set.

Kernel	**mean**	**std**	**median**	**min**	**max**
Gaussian	0.0021	0.0019	0.0013	0.0009	0.0049
Student's t	0.0007	0.0012	0.0001	0.0000	0.0026
Laplacian	0.0000	0.0000	0.0000	0.0000	0.0000

Table 3.2: Registration error statistics for non rigid transformations using different kernels on the shapes of figure 3.2. The kernel width varies between 5% and 30% of the mean variance of the reference set.

Kernel	**mean**	**std**	**median**	**min**	**max**
Gaussian	0.0029	0.0022	0.0020	0.0013	0.0061
Student's t	0.0009	0.0015	0.0002	0.0001	0.0031
Laplacian	0.0001	0.0001	0.0000	0.0000	0.0002

In order to compare our method with the state-of-the-art, our results were compared with the CPD [17], the RPM [14] and the GMMReg [103] algorithms. In this experimental configuration the kernel width σ was set to 20% of the variance of source point set X for all cases of our experiments.

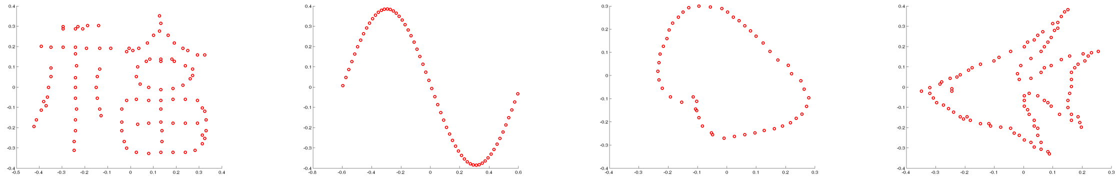
The code for implementing these algorithms was found in the web pages of the corresponding authors. RPM was implemented in Matlab environment, while CPD and GMMReg were programmed in C/C++ (Mex files). Therefore, this has an impact on the different execution times of the algorithms. Our method was partially implemented in Matlab (RVM training [109], by the official web page of Mike Tipping [115]) and in C (Hungarian algorithm for rectangular and square cost matrices, an implementation found in the Mathwork File Exchange web page). Several experiments were conducted (rigid and non rigid transformations) and apart from the registration error (root mean squared error) the execution time and convergence rate (i.e. how many iterations were necessary for the algorithm to converge) were also measured. A general conclusion is that the proposed Bayesian regression framework provides better results compared to RPM, where this algorithm either demands an extra post processing refinement step (e.g. registration of the centroids, figures 3.6(c)) or completely fails (e.g. tables 3.3, 3.4, 3.7, 3.8).

Each experiment was run 20 times and error statistics were calculated. In each configuration, a different registration transformation parameter set was used. The execution times are presented in table 3.5, along with the convergence rate in table 3.6 for experiments with noise free data and points in presence of noise. The initial sets of points, with representative results are demonstrated in figure 3.4 for the rigid case and in figures 3.5, 3.6 for the non rigid case. Also, to investigate the robustness of the algorithm to noise, the points were corrupted by Gaussian noise (with zero mean and small variance so as the shape does not change significantly). The initial sets of points, with the estimated results are demonstrated in figures 3.8 - 3.11, while the statistics are presented in tables 3.3 (rigid case) and 3.4 (non rigid case) for noise free points and in tables 3.7 (rigid case) and 3.8 (non rigid case) for points corrupted by Gaussian noise with zero mean value. To further support the statistical presentation of the registration error results, the p-value was computed, so as to verify the statistical significance of the analysis. Notice that in case of uncorrupted data, the deviations between real and computed values are too small for all the studied methods, and thus the p-value was not computed. In the case of data corrupted by Gaussian noise, there are differences between the results provided by each method. As it may be observed in the last row of Table 3.7 and Table 3.8, in all cases, the computed p-value is less than a threshold of significance level of 5%, which is usually employed.

One can observe that the proposed method, provides better results in general compared to CPD, RPM and GMMReg. Observe for example the concentration of estimated target points in an erroneous space (no underlying corresponding source points) in figure 3.6(b) and figure 3.6(d), even in the case of noise free data. The same also applies in case of points corrupted by noise, where CPD and GMMReg provided results that describe the shape of the target set quite well but there are points with no underlying correspondences, e.g. figures 3.9 (c) and 3.9 (e) or figures 3.10 (c) and 3.10 (e). On the other hand, RPM computed a good matching between the registered shapes but a refinement step is needed to achieve perfect registration, e.g. figure 3.6(c). In general, RPM proved too difficult to be tuned, and therefore provided a high rate of failures.

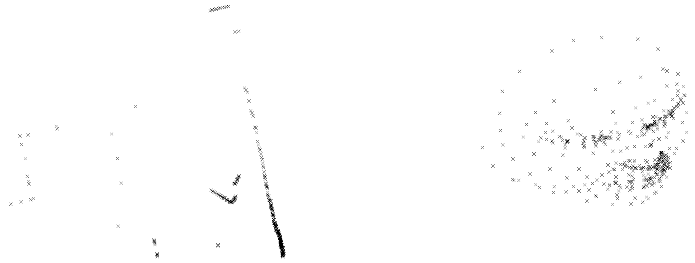
Concerning the CPD and the GMMReg methods, the time complexity of these algorithms are lower than ours which is partially implemented in Matlab (table 3.5). The fact that under similar comparison conditions our algorithm may provide similar results is justified by the convergence comparison (table 3.6), where one may observe that our technique converges quite faster than CPD and GMMReg. A general conclusion regarding the comparison of our method and CPD/GMMReg is, that, taking into account the registration error, the implementation and parameter tuning (e.g. selecting the type of transformation rigid or non rigid) along with the time complexity our method may provide better registration results, under the condition that a good assignment solution is provided.

Another issue studied in our experiments is the integration of an annealing scheme, either in the correspondence establishment (step 1 of algorithm 8) or in the RVM training (step 2 of algorithm 8). One approach was to embed softassign [102], as solution to the correspondence establishment instead of the Hungarian algorithm. The annealing temperature was initialized to 10% of the maximum pairwise distance between the points. After each iteration, the annealing temperature was reduced to 0.9 of its previous value.



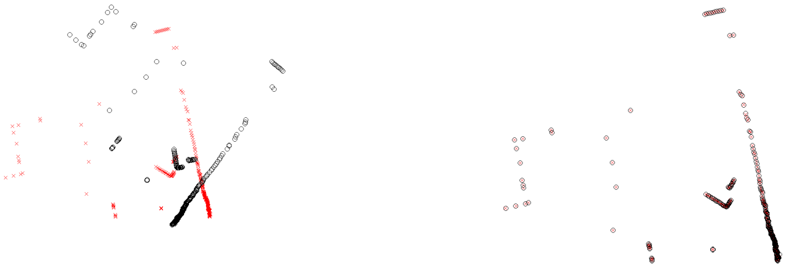
(a) (b) (c) (d)

Figure 3.2: The initial set of points used in our experiments, [14]. (a) *Sine*, (b) *Blob*, (c) *Fish* and (d) *Ideogram*.



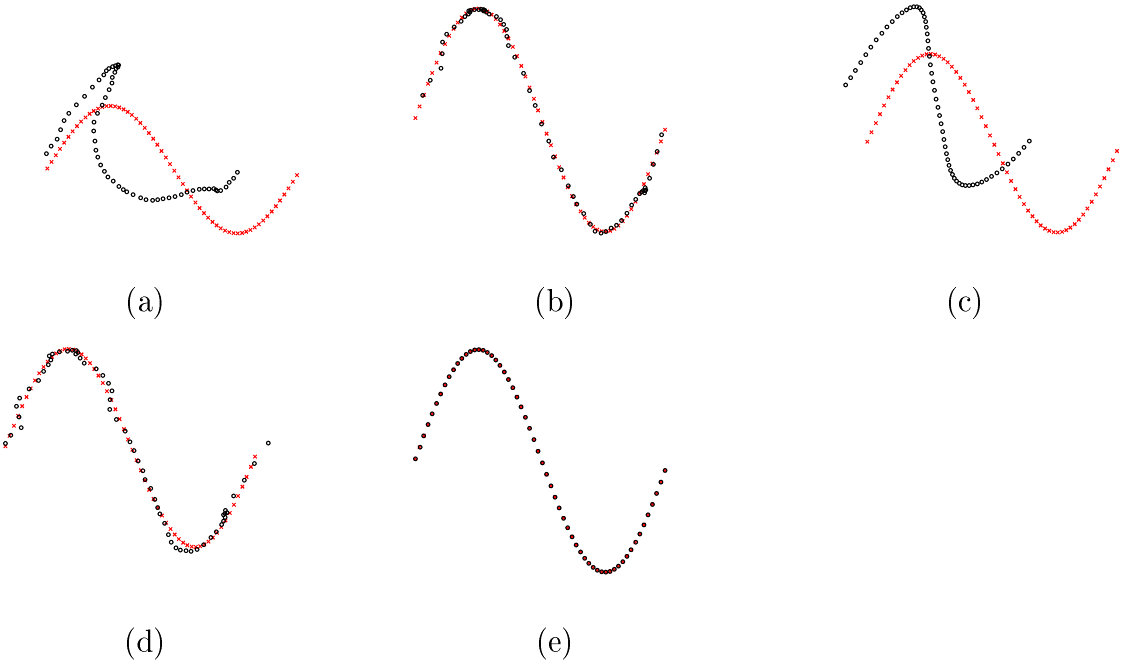
(a) (b)

Figure 3.3: (a) 2D range data used in our experiments [16]. (b) 3D set of points representing a face used in our experiments (3D face) [17].



(a) (b)

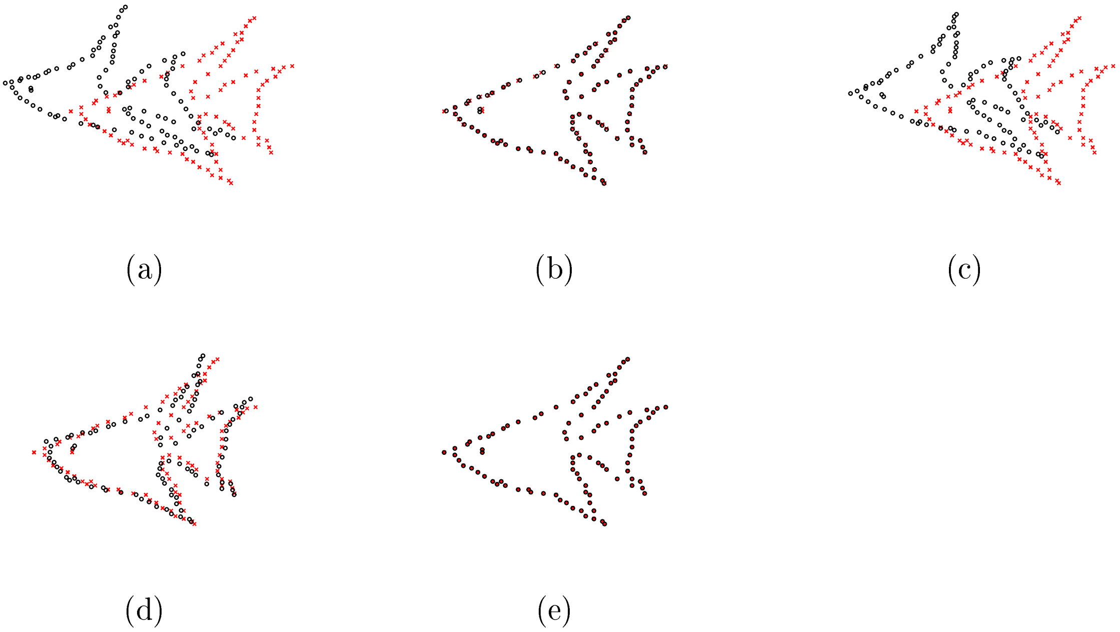
Figure 3.4: Rigid transformation experiment with 2D points of a range scan [16]. (a) Reference set of points (red) and deformed set of points (black) of a 3D face. (b) Registration result for the proposed method.



(a) (b) (c)

(d) (e)

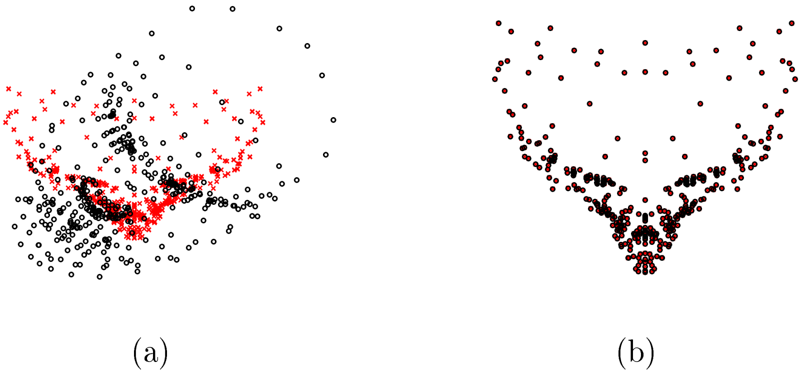
Figure 3.5: Non rigid transformation experiment. (a) Reference set of points (red) and deformed set of points (black). Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color.



(a) (b) (c)

(d) (e)

Figure 3.6: Non rigid transformation experiment. (a) Reference set of points (red) and deformed set of points (black). Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color.



(a) (b)

Figure 3.7: Non rigid transformation experiment with 3D points [17]. (a) Reference set of points (red) and deformed set of points (black) of a 3D face. (b) Registration result for the proposed method.

Table 3.3: Mean registration error for rigid transformations.

point set	**Hungarian-RVM**	**CPD** [17]	**RPM** [14]	**GMMReg** [103]
Sine	0.00	0.00	14.62	0.00
Blob	0.00	0.00	11.53	0.00
Fish	0.00	0.00	22.60	0.00
Ideogram	0.00	0.00	23.62	0.00
2D range	0.01	0.04	fail	0.00

Table 3.4: Mean registration error for non-rigid transformations.

point set	**Hungarian-RVM**	**CPD** [17]	**RPM** [14]	**GMMReg** [103]
Sine	0.00	0.00	13.45	0.00
Blob	0.00	0.00	11.19	0.00
Fish	0.00	0.00	21.87	0.00
Ideogram	0.00	0.00	23.50	0.00
3D face	0.00	0.08	fail	0.00

Table 3.5: Mean execution time (sec) of the compared methods for the whole set of experiments presented in section 3.3. The Hungarian-RVM is partially implemented in Matlab (RVM training) and C (Hungarian algorithm). RPM is totally implemented in Matlab while both CPD and GMMReg are totally implemented in C.

	Hungarian-RVM	**CPD** [17]	**RPM** [14]	**GMMReg** [103]
Pure Data	0.43	0.08	1.97	0.19
Gaussian Noise	0.30	0.08	2.53	0.49

Table 3.6: Average number of iterations of the compared methods for the whole set of experiments presented in section 3.3.

point set	**Hungarian-RVM**	**CPD** [17]	**RPM** [14]	**GMMReg** [103]
Pure Data	2	21	97	55
Gaussian Noise	2	20	87	56

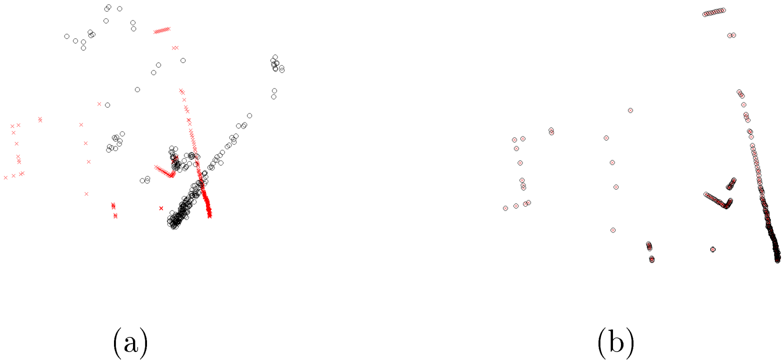


Figure 3.8: Rigid transformation experiment in presence of noise. (a) Reference 2D range set of points (red) and deformed set of points (black), [16] corrupted with zero mean additive Gaussian noise. (b) Registration result for the proposed method. The difference is better highlighted in color.

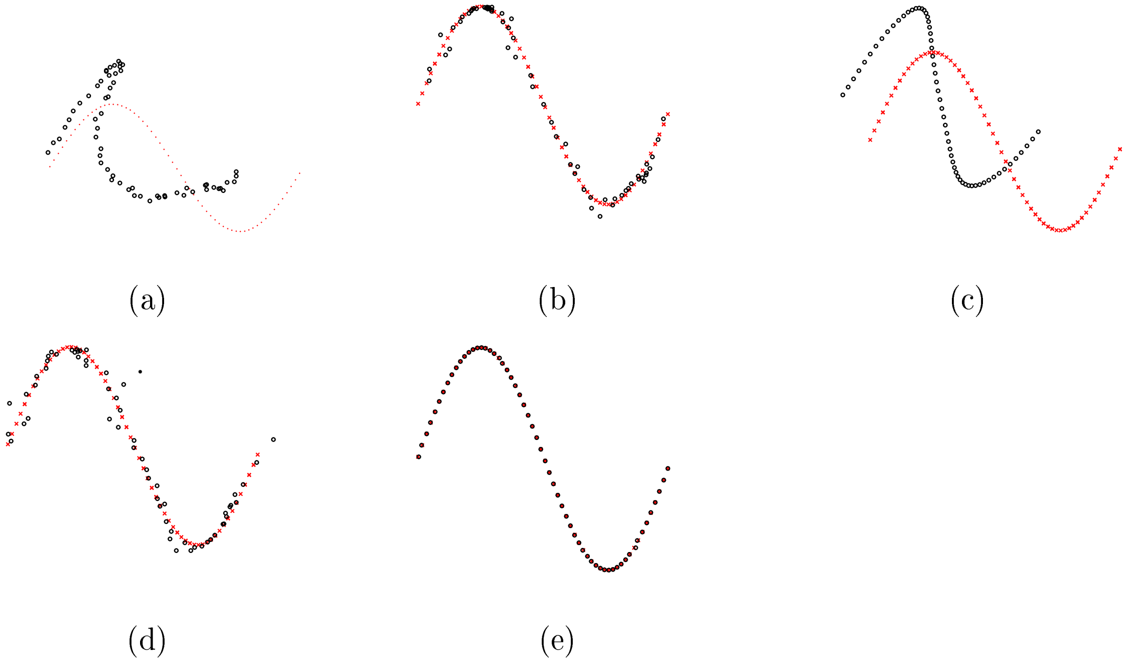


Figure 3.9: Non rigid transformation experiment in presence of noise. (a) Reference set of points (red) and deformed set of points (black) corrupted with zero mean additive Gaussian noise. Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color.

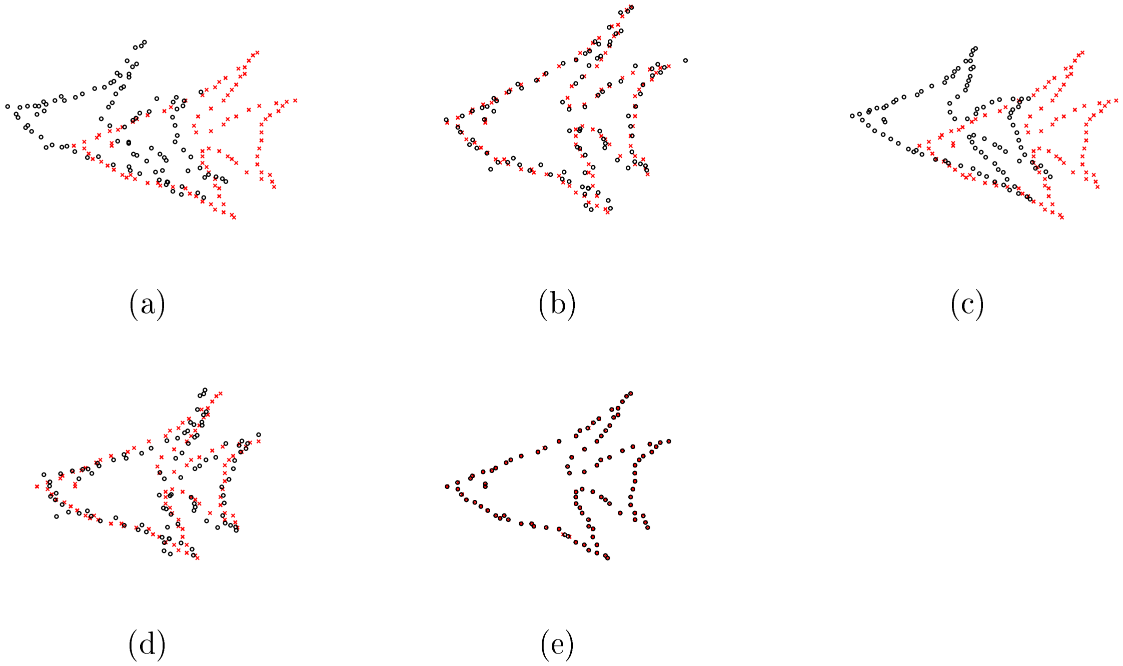


Figure 3.10: Non rigid transformation experiment in presence of noise. (a) Reference set of points (red) and deformed set of points (black) corrupted with zero mean additive Gaussian noise. Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color.

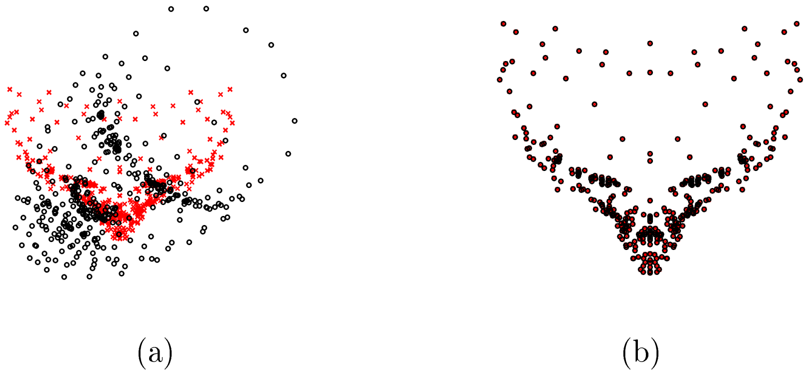


Figure 3.11: Non rigid transformation experiment in presence of noise. (a) Reference 3D set of a face points (red) and deformed set of points (black), [17] corrupted with zero mean additive Gaussian noise. (b) Registration result for the proposed method. The difference is better highlighted in color.

The results for the various combinations are presented in tables 3.9 and 3.10. As it can be observed, all the matching variants provide similar accuracy, regarding the mean squared error. However, considering the complexity of the model, direct application of the Hungarian algorithm appeared to be the most efficient approach. A few parameters have to be estimated while the execution time is considerably smaller. A straightforward implementation of the Hungarian algorithm demands less than one third of the softassign execution time. Based on the aforementioned remarks, we prefer the combination of the Hungarian algorithm (for solving the correspondence problem) with RVMs (for estimating the transformation) without any annealing scheme.

Table 3.7: Mean registration error for rigid transformations in presence of noise.

point set	**Hungarian-RVM**	**CPD** [17]	**RPM** [14]	**GMMReg** [103]
Sine	0.00	0.01	14.62	0.01
Blob	0.00	0.01	11.53	0.01
Fish	0.00	0.01	22.61	0.01
Ideogram	0.00	0.01	23.86	0.01
2D range	0.01	0.51	fail	0.48
p-value	-	0.00	10^{-15}	0.00

Table 3.8: Mean registration error for non-rigid transformations in presence of noise.

point set	**Hungarian-RVM**	**CPD** [17]	**RPM** [14]	**GMMReg** [103]
Sine	0.00	0.01	14.68	0.01
Blob	0.00	0.01	11.47	0.01
Fish	0.00	0.01	22.48	0.01
Ideogram	0.00	0.01	23.72	0.01
3D face	0.00	0.01	fail	0.00
p-value	-	10^{-4}	0.05	0.03

Table 3.9: Registration error statistics for non-rigid transformations.

point set	**Hungarian-RVM**				
	mean	**std**	**median**	**max**	**min**
sine	0.00	0.00	0.00	0.00	0.00
blob	0.00	0.00	0.00	0.00	0.00
fish	0.00	0.00	0.00	0.00	0.00
ideogram	0.00	0.00	0.00	0.00	0.00

Table 3.10: Registration error statistics for non-rigid transformations.

point set	**Softassign-RVM**				
	mean	**std**	**median**	**max**	**min**
Sine	0.02	0.02	0.02	0.01	0.04
Blob	0.02	0.02	0.02	0.00	0.04
Fish	0.01	0.01	0.01	0.01	0.02
Ideogram	0.01	0.00	0.01	0.00	0.01

More experiments were conducted to investigate the robustness of RVM regression with respect to TPS interpolation in the registration of point sets. For that purpose, we fixed the correspondences between the reference and the target sets in order to contain a number of false matches. Two types of experiments were performed. In the first type, the false matches preserved the one-to-one correspondence, that is, one point of the source set corresponds to exactly one point in the target set (one to one). In the other type of experiments, one point of the source set may correspond to one or more points in the target set (one to many). Then, we applied the transformation (TPS or RVM) and we counted the number of correct alignments. An alignment of two points was considered to be correct if the Euclidean distance between a transformed point and its corresponding was less than a predefined threshold. By varying the threshold we may plot a curve demonstrating the performance of the compared methods. These curves are shown in figure 3.12 for various cases of false matches on a set of 60 2D points. The curves correspond to a rigid transformation on the set of figure 3.2(a). The translation parameters were fixed to $[0.2, 0.3]^T$ and the rotation angle varied in the interval $[0°, 80°]$ with a step of 10° degrees. The curves in figure 3.12 show the average values between all angles examined per threshold. Notice that the RVM regression always provides an accurate result and justifies our claim that it can model better a registration transformation. In case of one to one correspondence, the target and the transformed sets almost coincide, while in the case of one to many correspondences, the registration result is close to the target set, and the shape is generally preserved. On the other hand, TPS completely fails to model the registration transformation even with few false matches. This behavior is justified by the fact that RVM does not consider the whole set for extracting the regression model. A representative example is also shown in figure 3.1.

In the same spirit, we examined the smoothness of the resulting transformation of RVM with respect to TPS. Following the same procedure, the number of false correspondences was gradually increased and the smoothness of the transformation was computed. We define the smoothness of a transformation $\mathcal{T}$ that registers set X to Y as

$$\mathcal{S}(\mathcal{T}) = \sum_{\substack{\mathbf{x}\in X \\ \mathbf{x}'\in\mathcal{N}(\mathbf{x})}} \left(d(\mathbf{x},\mathbf{x}') - d(\mathcal{T}(\mathbf{x}) - \mathcal{T}(\mathbf{x}'))\right)^2 \tag{3.4}$$

where $\mathcal{N}(\mathbf{x})$ is the set of nearest neighbors of $\mathbf{x}$ in X, $d(\mathbf{p},\mathbf{t}) = ||\mathbf{p} - \mathbf{t}||$ is the Euclidean

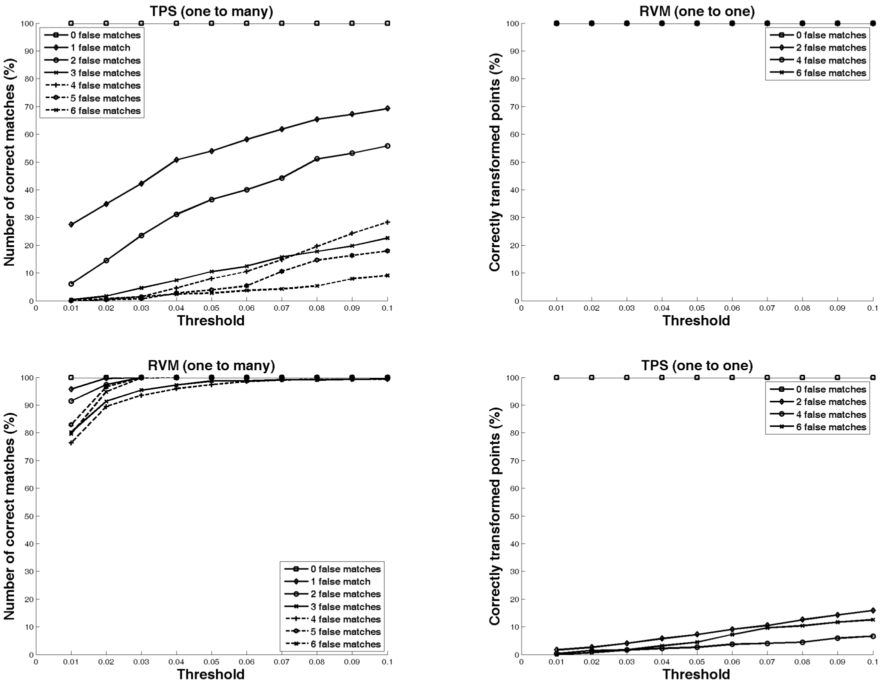


Figure 3.12: Curves representing the number of points correctly transformed with respect to a threshold determining the correct transformation using RVM (top row) and TPS (bottom row) when a number of initial false matches is established in source and target sets. A point in the source set is correctly transformed if, after transformation, its distance with respect to its correct counterpart is below the threshold. The left column shows results with false assignments that preserve the one-to-one matching. In that case the RVM provides a consistent behavior and its curves are all at 100% correct transformation. The right column shows results with false assignments that do not preserve the one-to-one matching.

distance between points $\mathbf{p}$ and $\mathbf{t}$, while $\mathcal{T}$ is either the RVM or the TPS transformation. The quantity given by (3.4) has a high value (indicating non smoothness) when a point and its neighbors in the source set have counterparts located at distant points in the target set. In other words, if the distance of the points to its neighbors in the source set is relatively different with respect to the distance of their counterparts in the target set a high penalty is added in the smoothness quantity. The curve in figure 3.13 presents the smoothness of the transformation by varying the number of false matches, and the number of neighbors in $\mathcal{N}(\mathbf{x})$. Notice that RVM provides a quite smoother transformation although it may result to foldings if the number of false matches is increased.

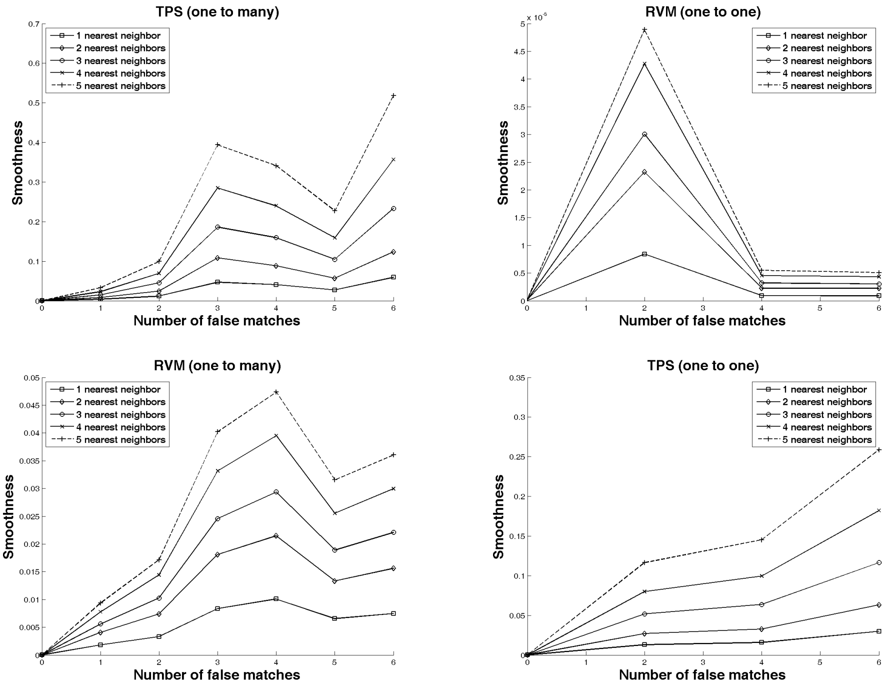


Figure 3.13: Smoothness (3.4) of the RVM (top row) and TPS (bottom row) under various number of false matches. The left column shows results with false assignments that preserve the one-to-one matching. Notice that the scale of vertical axis at the top-left plot is 10^{-5} indicating a very smooth transformation. The right column show results with false assignments that do not preserve the one-to-one matching.

Chapter 4

Registering images and sets of points using Mixture Models

4.1 Introduction

4.2 Image registration with mixtures of Gaussian and Student's t-distributions

4.3 Robust registration of point sets with mixtures of Student's t-distributions

4.4 Experimental results

4.1 Introduction

The goal of image registration is to geometrically align two or more images in order to superimpose pixels representing the same underlying structure. Image registration is an important preliminary step in many application fields involving, for instance, the detection of changes in temporal image sequences or the fusion of multimodal images. For the state of the art of registration methods we refer the reader to [116]. Medical imaging, with its wide variety of sensors (MRI, nuclear, ultrasonic, X-Ray) is probably one of the first application fields [117, 118, 119]. Other research areas related to image registration are remote sensing, multisensor robot vision and multisource imaging used in the preservation of artistic patrimony. Respective applications include the following of the evolution of pathologies in medical image sequences [120], the detection of changes in urban development from aerial photographs [121] and the recovery of underpaintings from visible/X-ray pairs of images in fine arts painting analysis [122].

The overwhelming majority of change detection or data fusion algorithms assume that the images to be compared are perfectly registered. Even slightly erroneous registrations may become an important source of interpretation errors when inter-image changes have to be detected. Accurate (i.e. subpixel or subvoxel) registration of single modal images remains an intricate problem when gross dissimilarities are observed. The problem is even more difficult for multimodal images, showing both localized changes that have to be detected and an overall difference due to the variety of responses by multiple sensors.

Since the seminal works of Viola and Wells [123] and Maes *et al.* [124], the maximization of the mutual information measure between a pair of images has gained an increasing popularity as a criterion for image registration [125]. The estimation of both marginal and joint probability density functions of the involved images is a key element in mutual information based image alignment. However, this method is limited by the histogram binning problem. Approaches to overcome this limitation include Parzen windowing [123, 126], where we have the problem of kernel width specification, and spline approximation [127, 128]. A recently proposed method relies on the continuous representation of the image function and develops a relation between image intensities and image gradients along the level sets of the respective intensity [129].

Gaussian mixture modeling (GMM) [45, 130] constitutes a powerful and flexible method for probabilistic data clustering that is based on the assumption that the data of each cluster has been generated by the same Gaussian component. In [131], GMMs were trained off-line to provide prior information on the expected joint histogram when the images are correctly registered. GMMs have also been successfully used as models for the joint [132] as well as the marginal image densities [133], in order to perform intensity correction. They have also been applied in the registration of point sets [134] without establishing explicit correspondence between points in the two images. The parameters of GMMs can be estimated very efficiently through maximum likelihood (ML) estimation using the EM algorithm [8]. Furthermore, it is well-known that GMMs are capable of modeling a large variety of pdfs [130].

An important issue in image registration is the existence of outlying data due to temporal changes (e.g. urban development in satellite images, lesion evolution in medical images) or even the complimentary but non redundant information in pairs of multimodal images (e.g. visible and infrared data, functional and anatomical medical images). Although a large variety of image registration methods have been proposed in the literature only a few techniques address these cases [135, 120, 136].

The method proposed in this study is based on mixture model training. More specifically, we train a mixture model once for the reference image and obtain the corresponding partitioning of image pixels into clusters. Each cluster is represented by the parameters of the corresponding density component. The main idea is that a component in the reference image corresponds to a component in the image to be registered. If the images are correctly registered the sum of distances between the corresponding components is minimum.

A straightforward implementation of the above idea would consider models with Gaussian components. However, it is well known that GMMs are sensitive to outliers and may lead to excessive sensitivity when the number of data points is small. This is easily understood by recalling that maximization of the likelihood function under an assumed Gaussian distribution is equivalent to finding the least-squares solution which lacks robustness. Consequently, a GMM tends to over-estimate the number of clusters since it uses additional components to capture the tails of the distributions [137]. The problem of attaining robustness against outliers in multivariate data is difficult and increases with the dimensionality. In this work, we consider mixture models (SMM) with Student's-t components for image registration. This pdf has heavier tails compared to a Gaussian [138]. More specifically, each component in the SMM mixture originates from a wider class of elliptically symmetric distributions with an additional parameter called the number of degrees of freedom. In this way, a more robust mixture model is employed than the typical GMM.

The main contributions of the proposed registration method are the following: (i) the histogram binning problem is overcome through image modeling with mixtures of distributions which provide a continuous representation of image density. (ii) Robustness to outlying pixel values is achieved by using mixtures of Student's t-distributions. The widely used method of maximization of the mutual information is outperformed. (iii) The method may be directly applied to vector valued images (e.g. diffusion tensor MRI) where standard histogram-based methods fail due to *the curse of dimensionality.* (iv) The proposed method is faster than histogram based methods where the joint histogram needs to be computed for every change in the transformation parameters.

Moreover, the registration problem is extended to the case of point sets where the nature of the problem is different since there is no spatial ordering contrary to image grids (e.g. pixelized images). Therefore, the difficulty consists in simultaneously estimating the transformation parameters and establishing correspondences between points.

In the related literature of point set registration, the standard approach is the well known Iterative Closest Points (ICP) algorithm [100] and its variants [101, 139, 140, 141]. In [14, 112] a robust point matching algorithm is proposed relying on *soft-assign* [142] and an iterative optimization procedure. The *soft-assign* is based on a matrix whose entries describe the probability that a point of one set matches upon transformation to one of the other set. Mutual information was also used as a constraint [143] for point set matching under the above framework. Features extracted from the point sets are employed in [7, 105], a kernel-based method is used in [144] and a method modeling the point sets by a GMM with constraints on the component centers is presented in [145]. Also, an approach to the construction of an atlas from multiple point sets is proposed in [146]. Finally, a work related to the herein proposed approach is presented in [134]. The authors propose to model the probability density function (pdf) of the points of the two sets by GMMs and estimate the transformation parameters through the minimization of an energy function describing the distance of the two GMMs. Our model completes this study by proposing a more robust framework for modeling the point sets.

4.2 Image registration with mixtures of Gaussian and Student's t-distributions

Let I_{ref} be an image of $N \times N$ pixels with intensities denoted as $I_{ref}(x^i)$, where x^i, $i = 1, ..., N^2$, is the i^{th} pixel index. The purpose of rigid image registration is to estimate a set of parameters $\mathcal{S}$ of the rigid transformation $T_{\mathcal{S}}$ minimizing a cost function $E(I_{ref}(\cdot), I_{reg}(T_{\mathcal{S}}(\cdot)))$ that, in a similarity metric-based context, expresses the similarity between the image pair. In the 2D case the rigid transformation parameters are the rotation angle and the translation parameters along the two axes. In the 3D case, there are three rotation and three translation parameters. Eventually, scale factors may also be included, depending on the definition of the transformation.

Consider, now, a partitioning of the reference image I_{ref} into K clusters (groups) by training a mixture model with K components with arbitrary pdf $p(I(x); \Theta)$:

$$\phi(I_{ref}(x)) = \sum_{k=1}^{K} \pi_k p(I_{ref}(x); \Theta_k^{ref})$$

Therefore, the reference image is represented by the parameters Θ_k^{ref}, $k = 1, \ldots, K$ of the mixture components. The partitioning of the image is described using the function $f(x) : [1, 2, ..., N] \times [1, 2, ..., N] \rightarrow \{1, 2, ..., K\}$, where $f(x) = k$ means that pixel x of the reference image I_{ref} belongs to the cluster defined by the k^{th} component. Let us also define the sets of all pixels of image I_{ref} belonging to the k^{th} cluster:

$$P_k = \{x^i \in I_{ref}, i = 1, 2, ..., N^2 | \delta(f(x^i) - k) = 1\}$$

for $k = 1, ..., K$, where $\delta(x)$ is the Dirac function:

$$\delta(f(x^i) - k) = \begin{cases} 1, & \text{if } f(x^i) = k \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

The above mixture-based segmentation of the reference image is performed once, at the beginning of the registration procedure. The reference image I_{ref} is, thus, partitioned into K groups, generally, not corresponding to connected components in the image. This spatial partition is projected on the image to be registered I_{reg}, yielding a corresponding partition of this second image (i.e., the partitioning of the reference image acts as a mask on the image to be registered). Then, we assume that the pixel values of each cluster k in I_{reg} are modeled using a mixture component with parameters Θ_k^{reg} obtained from the statistics of the intensities of pixels in group k of I_{reg}.

In order to apply our method it should be possible to define a distance measure $D(\Theta_k^{ref}, \Theta_k^{reg})$ between the corresponding mixture components with pdf $p(I)$. Then the energy function we propose, is expressed by the weighted sum of distances between the corresponding components in I_{reg} and I_{ref}:

$$E(I_{ref}(\cdot), I_{reg}(T_{\mathcal{S}}(\cdot))) = \sum_{k=1}^{K} \pi_k D(\Theta_k^{ref}, \Theta_k^{reg}) \tag{4.2}$$

where π_k is the mixing proportion of the k^{th} component:

$$\pi_k = \frac{|P_k|}{\sum\limits_{l=1}^{K} |P_l|}$$

where $|P_k|$ denotes the cardinality of set P_k. If the two images are correctly registered the criterion in (4.2) assumes that the total distance between the whole set of components would be minimum.

For a given set of transformation parameters $\mathcal{S}$, the total energy between the image pair is computed through the following steps:

- segment the reference image $I_{ref}(\cdot)$ into K clusters by a mixture model.
- for each cluster $k = 1, 2, ..., K$ of the reference image:
  - project the pixels of the cluster onto the transformed image to be registered $I_{reg}(T_{\mathcal{S}}(\cdot))$.
  - determine the parameters Θ_k^{reg} of the projected partition of I_{reg}.
- evaluate the energy in eq. (4.2) by computing the distances between the corresponding densities.

In the case of GMMs, the above registration procedure can be applied as follows:

Consider the multivariate normal distributions $N_1(\mu_1, \Sigma_1)$ and $N_2(\mu_2, \Sigma_2)$ and denote $\Theta_i = \{\mu_i, \Sigma_i\}$, with $i = \{1, 2\}$, their respective parameters (mean vector and covariance matrix). The Chernoff distance between these distributions is defined as [147]:

$$\begin{aligned} C(\Theta_1, \Theta_2, s) &= \frac{s(1-s)}{2}(\mu_2 - \mu_1)^T [s\Sigma_1 + (1-s)\Sigma_2]^{-1}(\mu_2 - \mu_1) \\ &+ \frac{1}{2}\ln\left(\frac{|s\Sigma_1 + (1-s)\Sigma_2|}{|\Sigma_1|^s |\Sigma_2|^{1-s}}\right). \end{aligned}$$

The Bhattacharyya distance is a special case of the Chernoff distance with $s = 0.5$:

$$B(\Theta_1, \Theta_2) = \frac{1}{8}(\mu_2 - \mu_1)^T \left[\frac{\Sigma_1 + \Sigma_2}{2}\right]^{-1} (\mu_2 - \mu_1) + \frac{1}{2}\ln\left(\frac{|\frac{\Sigma_1+\Sigma_2}{2}|}{\sqrt{|\Sigma_1||\Sigma_2|}}\right) \tag{4.3}$$

A representative GMM for the reference image can be obtained via the EM algorithm [45]. Therefore, the reference image is represented by the parameters $\Theta_k^{ref} = \{\mu_k^{ref}, \Sigma_k^{ref}\}$, $k = 1, \ldots, K$ of the GMM components. After projecting the pixel groups of the reference image to obtain the corresponding groups in the registered image, the parameters Θ_k^{reg} can be estimated by taking the sample mean μ_k^{reg} and the sample covariance matrix Σ_k^{reg}:

$$\mu_k^{reg} = \frac{1}{|P_k|}\sum_{i=1}^{N^2} I_{reg}(T_{\mathcal{S}}(x^i))\delta(f(x^i) - k) \tag{4.4}$$

and

$$\Sigma_k^{reg} = \frac{1}{|P_k|} \sum_{i=1}^{N^2} (\Delta I_k^i)(\Delta I_k^i)^T \delta(f(x^i) - k), \tag{4.5}$$

where $\Delta I_k^i = I_{reg}(T_{\mathcal{S}}(x^i)) - \mu_k^{reg}$. The role of $\delta(f(x^i) - k)$ in eq. (4.4) and (4.5) is to determine the support (the pixel coordinates) for the calculation of the mean and covariance. These parameters are computed *on the image to be registered* for the pixel coordinates belonging to the k^{th} group *on the reference image*. This also implies a Gaussian mixture model for the components of I_{reg}. The total distance between the two images is computed using eq. (4.2), where the Bhattacharyya distance between the corresponding Gaussian components is considered as distance measure D.

However, GMMs are very sensitive to outlying data and their outcome is largely influenced by pixels not belonging to the dominating model. In order to overcome this drawback of GMMs, we have employed in our registration method mixtures of Student's t-distributions. These mixtures are more robust to outliers as it is described in the next section.

A d-dimensional random variable X that follows a multivariate t-distribution with mean μ, positive definite, symmetric and real $d \times d$ covariance matrix Σ and has $\nu \in [0, \infty)$ degrees of freedom has a density expressed by:

$$p(x; \mu, \Sigma, \nu) = \frac{\Gamma\left(\frac{\nu+d}{2}\right) |\Sigma|^{-\frac{1}{2}}}{(\pi\nu)^{\frac{d}{2}} \Gamma\left(\frac{\nu}{2}\right) \left[1 + \nu^{-1}\delta(x, \mu; \Sigma)\right]^{\frac{\nu+d}{2}}} \tag{4.6}$$

where $\delta(x, \mu; \Sigma) = (x - \mu)^T \Sigma^{-1} (x - \mu)$ is the Mahalanobis squared distance and Γ is the Gamma function.

It can be shown that the Student's t distribution is equivalent to a Gaussian distribution with a stochastic covariance matrix. In other words, given a weight u following a Gamma distribution parameterized by ν:

$$u \sim Gamma(\nu/2, \nu/2). \tag{4.7}$$

the variable X has the multivariate normal distribution with mean μ and covariance Σ/u:

$$X|\mu, \Sigma, \nu, u \sim N(\mu, \Sigma/u), \tag{4.8}$$

It can be shown that for $\nu \rightarrow \infty$ the Student's t-distribution tends to a Gaussian distribution with covariance Σ. Also, if $\nu > 1$, μ is the mean of X and if $\nu > 2$, $\nu(\nu-2)^{-1}\Sigma$ is the covariance matrix of X. Therefore, the family of t-distributions provides a heavy-tailed alternative to the normal family with mean μ and covariance matrix that is equal to a scalar multiple of Σ, if $\nu > 2$ (fig. 4.1). A K-component mixture of t-distributions is given by

$$\phi(x, \Psi) = \sum_{i=1}^{K} \pi_i p(x; \mu_i, \Sigma_i, \nu_i) \tag{4.9}$$

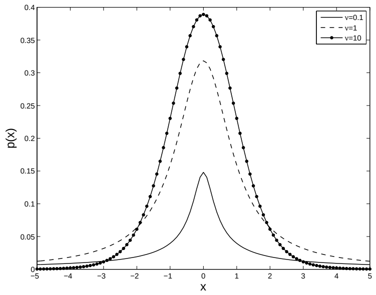


Figure 4.1: A univariate Student's t-distribution ($\mu = 0$, $\sigma = 1$) for various degrees of freedom. As $\nu \to \infty$ the distribution tends to a Gaussian. For small values of ν the distribution has heavier tails than a Gaussian.

where $x = (x_1, ..., x_N)^T$ denotes the observed-data vector and

$$\Psi = (\pi_1, ..., \pi_K, \mu_1, ..., \mu_K, \Sigma_1, ..., \Sigma_K, \nu_1, ..., \nu_K)^T. \tag{4.10}$$

are the parameters of the components of the mixture.

A Student's t-distribution mixture model (SMM) may also be trained using the EM algorithm [138]. Consider now the complete data vector

$$x_c = (x_1, ...x_N, z_1, ..., z_N, u_1, ..., u_N)^T \tag{4.11}$$

where $z_1, ..., z_N$ are the component-label vectors and $z_{ij} = (z_j)_i$ is either one or zero, according to whether the observation x_j is generated or not by the i^{th} component. In the light of the definition of the t-distribution, it is convenient to view that the observed data augmented by the z_j, $j = 1, ..., N$ are still incomplete because the component covariance matrices depend on the degrees of freedom. This is the reason that the complete-data vector also includes the additional missing data $u_1, ..., u_N$. Thus, the E-step on the $(t+1)^{th}$ iteration of the EM algorithm requires the calculation of the posterior probability that the datum x_j belongs to the i^{th} component of the mixture:

$$z_{ij}^{t+1} = \frac{\pi_i^t p(x_j; \mu_i^t, \Sigma_i^t, \nu_i^t)}{\sum_{m=1}^{K} \pi_m^t p(x_j; \mu_m^t, \Sigma_m^t, \nu_m^t)} \tag{4.12}$$

as well as the expectation of the weights for each observation:

$$u_{ij}^{t+1} = \frac{\nu_i^t + d}{\nu_i^t + \delta(x_j, \mu_i^t; \Sigma_i^t)} \tag{4.13}$$

Maximizing the log-likelihood of the complete data provides the update equations of the respective mixture model parameters:

$$\pi_i^{t+1} = \frac{1}{N} \sum_{j=1}^{N} z_{ij}^t, \tag{4.14}$$

$$\mu_i^{t+1} = \frac{\sum_{j=1}^{N} z_{ij}^t u_{ij}^t x_j}{\sum_{j=1}^{N} z_{ij}^t u_{ij}^t}, \tag{4.15}$$

$$\Sigma_i^{t+1} = \frac{\sum_{j=1}^{N} z_{ij}^t u_{ij}^t (x_j - \mu_i^{t+1})(x_j - \mu_i^{t+1})^T}{\sum_{j=1}^{N} z_{ij}^{t+1}}. \tag{4.16}$$

The degrees of freedom ν_i^{t+1} for the i^{th} component, at time step $t+1$, are computed as the solution to the equation:

$$\log\left(\frac{\nu_i^{t+1}}{2}\right) - \psi\left(\frac{\nu_i^{t+1}}{2}\right) + 1 - \log\left(\frac{\nu_i^t + d}{2}\right) + \frac{\sum_{j=1}^{N} z_{ij}^t (\log u_{ij}^t - u_{ij}^t)}{\sum_{j=1}^{N} z_{ij}^t} + \psi\left(\frac{\nu_i^t + d}{2}\right) = 0 \tag{4.17}$$

where $\psi(x) = \frac{\partial(ln\Gamma(x))}{\partial x}$ is the digamma function.

At the end of the algorithm, the data are assigned to the component with maximum responsibility using a maximum *a posteriori* (MAP) principle.

The Student's t-distribution is a heavy tailed approximation to the Gaussian. It is therefore, natural to consider the mean and covariance of the SMM components to approximate the parameters of a GMM on the same data as it was described in the previous section. If the statistics of the images follow a Gaussian model, the degrees of freedom ν_i are relatively large and the SMM tends to be a GMM with the same parameters. If the images contain outliers, parameters ν_i are weak and the mean and covariance of the data are appropriately weighted in order not to take into account the outliers. Thus, the parameters of the SMM, computed on the reference image I_{ref}, are used as component parameters Θ_k^{ref} in a straightforward way as they generalize the Gaussian case by correctly addressing the outliers problem. After projection of the pixel groups of the reference image to their corresponding groups in the registered image, the parameters Θ_k^{reg} are computed using the sample mean (4.4) and the sample covariance matrix (4.5).

Once model inference is accomplished, the Bhattacharyya distance between the components of the Student's t-mixtures is minimized. The difference with respect to the GMM is that the covariance matrices are properly scaled by the Gamma distributed parameters u as it is defined in equations (4.7)-(4.8).

Finally, let us notice that the energy in (4.2) may be applied to both single and multimodal image registration. In the latter case, the difference in the mean values of the distributions in (4.3) should be ignored, as we do not search to match the corresponding Student's t-distributions in position but only in shape. In that case, the distance in (4.3) becomes:

$$B(\Theta_1, \Theta_2) = \ln \left(\frac{|\frac{\Sigma_1+\Sigma_2}{2}|}{\sqrt{|\Sigma_1||\Sigma_2|}} \right) \tag{4.18}$$

which is equivalent to a correlation coefficient between the two distributions.

4.3 Robust registration of point sets with mixtures of Student's t-distributions

An extension of the registration algorithm to handle point sets is described in this section. Given two sets of points X and Y such that Y is derived from X after applying a rigid transformation $T_{\mathcal{S}}$ with parameters $\mathcal{S}$, that is $Y = T_{\mathcal{S}}(X)$, the problem consists in estimating the transformation parameters from the two data sets without prior knowledge on any correspondence. In fact, in our formulation, there could be no exact correspondence at all due to noise or outlying points.

Let us denote $p(x)$ the density at a point $x \in X$ and assume that it is expressed by a GMM of M components:

$$p(x) = \sum_{j=1}^{M} \pi_j^x \mathcal{N}(x|\mu_j^x, \Sigma_j^x). \tag{4.19}$$

By the same assumption, the density at a point $y \in Y$ is given by another GMM:

$$q(y) = \sum_{j=1}^{N} \pi_j^y \mathcal{N}(y|\mu_j^y, \Sigma_j^y). \tag{4.20}$$

Considering the transformed point set distribution as $p_{R,t}(x)$, where R is the rotation matrix ant t is the translation vector, that is

$$p_{R,t}(x) = \sum_{i=1}^{M} \pi_i^x \mathcal{N}(x|R\mu_i^x + t, R\Sigma_i^x R^T), \tag{4.21}$$

we seek to minimize the energy function:

$$D(p_{R,t}, q) = \int [p_{R,t}(z) - q(z)]^2 dz \tag{4.22}$$

with respect to R and t. More specifically, we seek to match the continuous shapes of the mixtures $p_{R,t}$ and q over their region of support. Equation (4.22) may be simplified:

$$D(p_{R,t}, q) = \int \left[p_{R,t}^2(z) + q^2(z) - 2p_{R,t}(z)q(z)\right] dz \tag{4.23}$$

The first two terms are invariant under rigid transformation and therefore, the above expression yields the maximum of the product of the two distributions over the whole sets of points. This is equivalent to maximizing the correlation between the pdfs. The cross term may be also expressed as[134]:

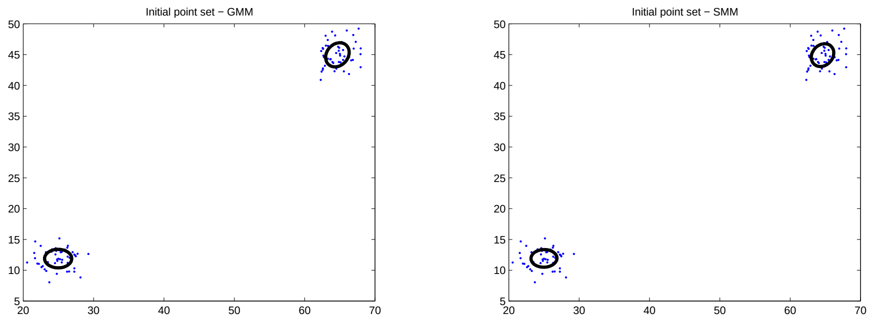
$$\int\int p_{R,t}(x)q(y)dxdy = \sum_{i=1}^{M}\sum_{j=1}^{N}\pi_i^x\pi_j^y\mathcal{N}(0|R\mu_i^x + t - \mu_j^y, R\Sigma_i^x R^T + \Sigma_j^y) \tag{4.24}$$

meaning that given the i^{th} component from the frst mixture and the j^{th} component from the second mixture, each term of the sum is evaluated as a Gaussian pdf with mean vector $R\mu_i^x + t - \mu_j^y$ and covariance matrix $R\Sigma_i^x R^T + \Sigma_j^y$ at $x = 0$.

Replacing the GMMs by the more robust SMMs in the above equations (4.19) and (4.20) leads to a better modeling of the point sets. Figures 4.2 and 4.3 illustrate the performance of a mixture of Student's t-mixture with respect to a standard GMM to model a $2D$ point set. In the original set, both methods correctly captured the shape of the data (fig. 4.2). On the other hand, when a small amount of outliers (5%) was present in the set the GMM failed to provide a satisfactory solution while the heavier tailed SMM correctly modeled the point sets (fig. 4.3). Thus, SMM seems to be a preferable model for density-based point set registration.

An alternative approach would be to provide a model for the outliers using a GMM with a background component or, generally, a probabilistic a model for false observations [138, 148]. However, as it will be shown in the experimental results, if the background outliers are not uniformly or normally distributed this approach has its limitations.

Let us note that the above formulas also apply for the registration of point sets using the mixtures of Student's t-distributions by properly computing the components mean vectors and covariance matrices following the definition of the distributions (4.7)-(4.8) and the respective EM algorithm described in section 4.2.



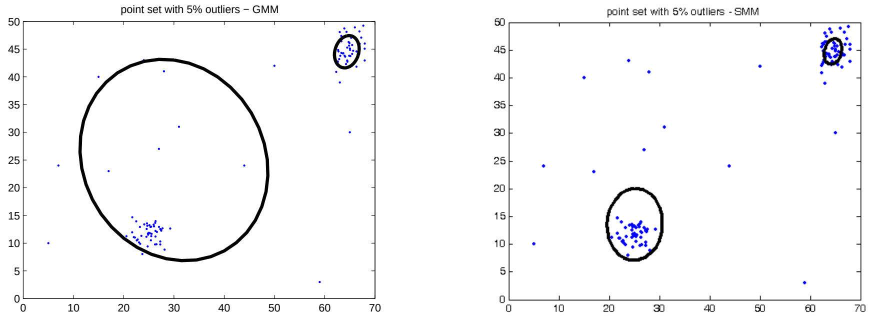
(a)

(b)

Figure 4.2: A 2D point set and the obtained models (a) GMM and (b) SMM.

4.4 Experimental results

A large number of interpolations are involved in the registration process. The accuracy of the rotation and translation parameter estimates is directly related to the accuracy



(a) (b)

Figure 4.3: The point set of figure 4.2 with 5% outliers and the obtained models by (a) GMM and (b) SMM. Notice that the GMM solution is affected by the outliers while the SMM is more robust.

of the underlying interpolation model. Simple approaches such as the nearest neighbor interpolation are commonly used because they are fast and simple to implement, though they produce images with noticeable artifacts. More satisfactory results can be obtained by small-kernel cubic convolution techniques. In our experiments, we have applied a cubic interpolation scheme, thus preserving the quality of the image to be registered.

The Matlab optimization toolbox was used to perform optimization. In particular we tested the algorithm with a derivative free optimization algorithm (simplex) and a Quasi-Newton algorithm (BFGS) with a numerical calculation of the derivatives. Notice that the methods mentioned perform only local optimization, thus depending the final result highly with the initial starting point. Global optimization methods may also be considered but they are highly time consuming.

In order to evaluate the proposed method, we have performed a number of experiments in some relatively difficult registration problems. Registration errors were computed in terms of pixels and not in terms of transformation parameters. Registration accuracies in terms of rotation angles and translation vectors are not easily evaluated due to parameter coupling. Therefore, the registration errors are defined as deviations of the corners of the registered image with respect to the ground truth position. Let us notice that these registration errors are less forgiving at the corners of the image (where their values are larger) with regard to the center of the image frame.

At first, we have simulated a multimodal image registration example. The image in 4.4(a) is an artificial piecewise constant image. The image in 4.4(b) is its negative image. The image in 4.4(a) was degraded by uniformly distributed noise in order to achieve various SNR values (between 14 dB and -1 dB). The degraded images underwent several rigid transformations by rotation angles varying between $[0, 20]$ degrees and translation parameters between $[-15, 10]$ pixels. To investigate the robustness of the proposed method to outliers we have applied the algorithm with $K = 3$ components considering both GMMs and SMMs, and 256 histogram bins in the case of the normalized MI. Figure 4.5 illustrates the average registration errors for the different SNR values. For each SNR, four different transformations were applied to the image and the average value of the registration error is presented. For comparison purposes, the performance of the MI method is also shown. As it can be observed, both the GMM and the SMM-based registration methods outperform the MI which fails when the SNR is low. Moreover, the heavier tailed SMM demonstrates better performance for considerable amounts of noise.

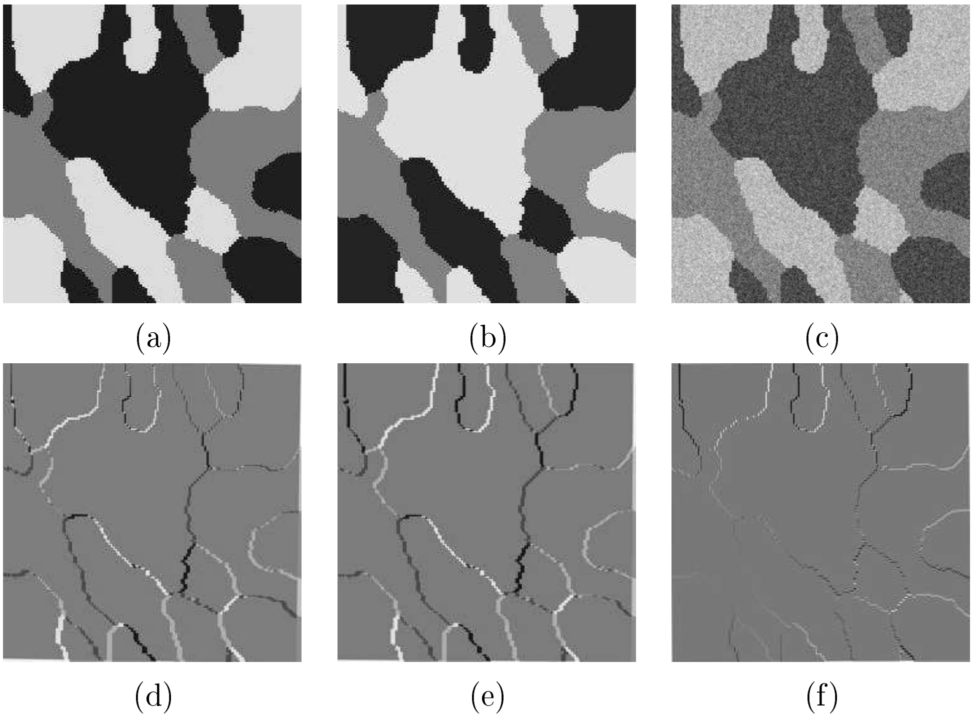


Figure 4.4: (a) A three-class piecewise constant image with intensity values 30, 125 and 220, and (b) its negative image (corresponding values, 225, 130 and 35). (c) The image in (a) degraded by uniform noise at $14\,dB$. This image was then registered to the image in (b). The bottom line shows the registration errors for the compared methods. The ground truth solution is 0 deg for the rotation and zero translation (the original image). (d) MI, (e) GMM, (f) SMM. The errors present the difference between the noise free registered image and the reference image. the values are scaled for better visualization.

Furthermore, let us notice that the proposed energy function involving the Battacharyya distances is convex around the true minimum (fig. 4.6) as it is also the case for the MI [149].

An open issue in mixture modeling is the determination of the number of components. In our experiments, in the case of non artificial images, the number of components is unknown. If the number of components of the mixtures is neither too high (overfitting) nor too low (underfitting) with respect to the ground truth the registration accuracy is not affected by that parameter. In order to demonstrate it, we have performed the experiments involving non artificial images by varying the number of components in the experiments.

In that framework, the proposed registration method was tested on a multimodal

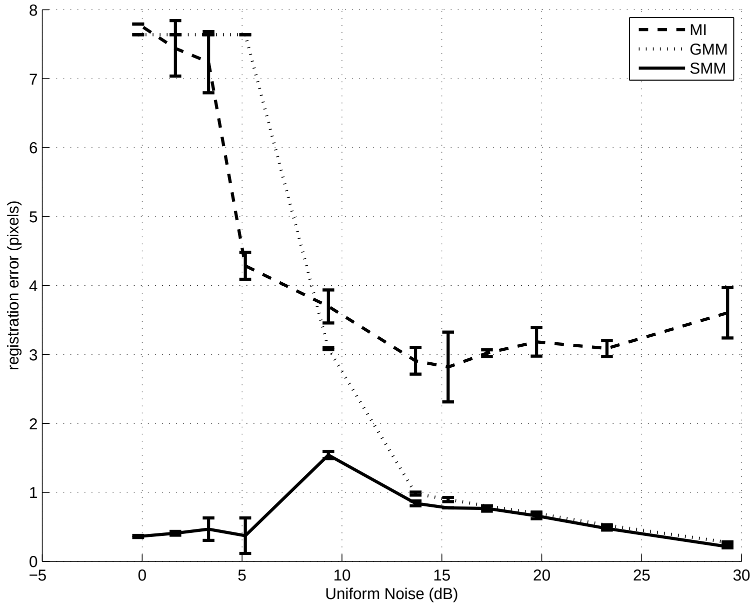


Figure 4.5: Mean registration error versus signal to noise ratio (SNR) for the 3-class registration experiment of figure 4.4.

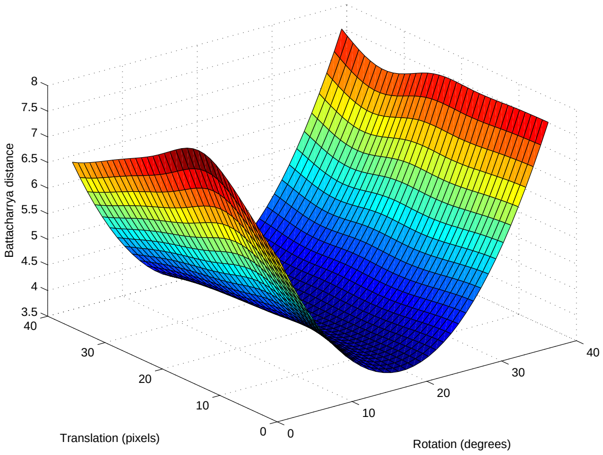


Figure 4.6: The objective function in eq. (4.2) for the registration of the image of figure 4.4(a) with its counterpart rotated by 20 degrees and translated by 10 pixels.

Table 4.1: *Statistics on the registration errors for the images in fig. 4.7 with varying number of mixture components. The errors are expressed in pixels.*

Registration errors - Cell images

	K	mean	std.	median	max	min
MI	256 bins	3.663	0.957	4.019	4.25	1.461
SMM	2	3.157	0.009	3.153	3.178	3.150
SMM	3	2.955	0.636	3.148	3.178	1.146
SMM	4	2.956	0.604	3.159	3.101	1.146
SMM	5	2.953	0.640	3.152	3.177	1.132

image pair such as the cell images in fig. 4.7. The complimentary but not redundant information carried by the multimodal images increases the difficulty of the registration process. In both experiments we have applied 20 rigid transformations to one of the images, for each configuration of the transformation parameters, with rotation angles varying between $[0, 20]$ degrees and translation parameters between $[-15, 10]$ pixels.

The experiments in the case of the images in figure 4.7 were realized with the number of components varying from $K = 2$ to $K = 5$. For the MI we used 256 histogram bins. Table 4.1 summarizes the statistics on the registration errors. As it can be observed, the SMM method achieves highly better registration accuracy. Also, the number of components did not significantly affect the registration accuracy.

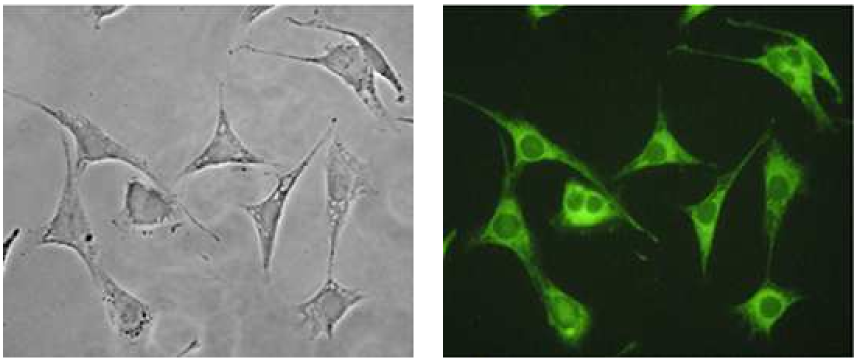

(a) (b)

Figure 4.7: A pair of NIH 3T3 electron microscope images (400x magnification) of rat cells under (a) normal and (b) fluorescent light.

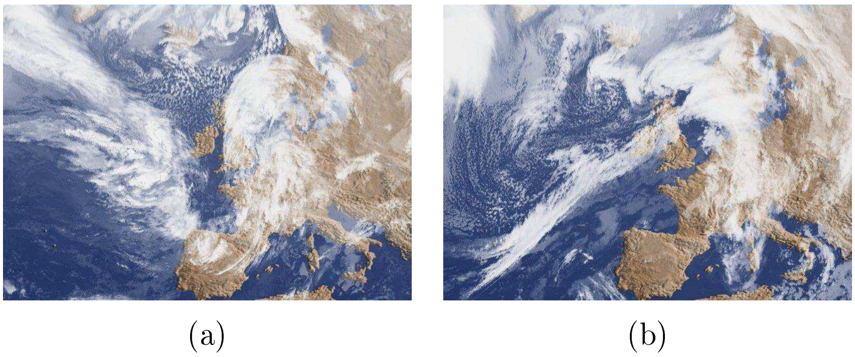
A last experiment demonstrating the ability of the proposed SMM method to deal with outliers is the registration of a remotely sensed image pair. The meteorological images of Europe in fig. 4.8 were acquired at different dates. The image in fig. 4.8(b) underwent 20 rigid transformations for each parameter instance, with values of rotation angle uniformly sampled in the interval $[0, 20]$ degrees and translations between $[-15, 10]$ pixels. The experiments were realized with the number of components varying between $K = 2$ and $K = 6$ for GMM and SMM and 256 bins for the MI.

Table 4.2: *Statistics on the registration errors for the images in fig. 4.8 with varying number of mixture components. The errors are expressed in pixels.*

Registration errors - Satellite images

	K	mean	std	median	max	min
MI	256 bins	6.742	1.493	7.463	7.733	3.565
SMM	2	2.975	0.013	2.979	2.991	2.951
SMM	3	1.857	1.202	1.251	3.653	1.283
SMM	4	2.129	2.289	2.960	3.651	1.359
SMM	5	1.208	0.237	1.142	1.999	1.141
SMM	6	1.210	0.238	1.145	2.001	1.142

The large amount of clouds at different locations in the image pair introduce difficulties in the registration procedure. It is worth commenting that the MI method failed to register the images and systematically provided registration errors of the order of 6 pixels. The SMM method produced very small registration errors which are summarized in table 4.2.



(a)

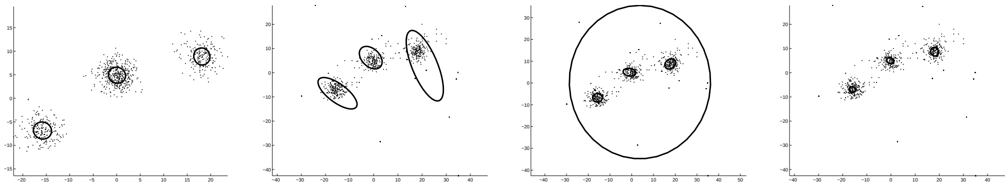
(b)

Figure 4.8: (a) Image of Europe on 8 January 2007 at 01h00, provided by MeteoSat. (b) Image of Europe on 9 January 2007 at 01h00, provided by MeteoSat (by courtesy of Meteo-France). Notice the large amount of outliers (cloudy regions in different locations in the image pair) introducing important difficulties in the registration process.

In order to evaluate the proposed point set registration method we have performed three types of experiments. At first, a 2D set of 600 points was generated from three different Gaussian distributions with means $(-16, 9)$, $(0, 5)$ and $(18, 9)$ and spherical covariance matrices with the standard deviation being 2 in each dimension. The point set underwent rotations varying between $[-90°, 90°]$ and translations varying between $[-100, 100]$ in both dimensions. In all of the cases the proposed algorithm provided solutions close to the true transformation parameters. The registration error was measured as the average distance between the points transformed by the true parameters and the points obtained by the estimated transformation. In all cases, the order of the registration error was approximatively 10^{-6}. This experiment was repeated for increased number of non overlapping components and the previous results were confirmed.

A second experiment consisted in comparing the SMM not only to a typical GMM but also to a GMM having an extra background component (called GMMb) in order to model the outliers. This is a standard technique to capture the distribution of outliers and it is also proposed in [138, 148]. We have observed that when the outliers are normally or uniformly distributed the performance of the two approaches (GMMb and SMM) is similar because the fourth component is a good model for outliers. However, if the outliers are *signal-dependent* the fourth component does not provide the optimal solution.

In our experiments, the previous point set was corrupted by outlying data from 1% up to 15%. Each of the three set of points was corrupted by a uniform noise having range the double of the initial range of the points generated by the respective component. By these means, the outliers are sparsely distributed around each component. Also, 1% extra outliers were globally added to make the problem more challenging. For each configuration of the percentage of the outliers, 5 registration experiments were performed with random translation and rotation parameters. A representative example for 9% of points being contaminated is shown in figure 4.9. In figure 4.10, the results for the registration errors are summarized. As it can be observed, although the GMMb performs better than the standard GMM due to its background component, the SMM provides smaller registration errors consistently. This behavior is easily explained by the shapes of the ellipses in figures 4.9(b) and 4.9(c). Both the GMMb and the SMM estimated small covariances but in GMMb the orientations of the ellipses diverge more from the noise-free case. Finally, it is worth noticing that the standard ICP registration algorithm fails in all cases to provide an acceptable registration.



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

Figure 4.9: Example of a set of points used in the experiments. (a) A point set (presented by dots) was generated by 3 Gaussians with means $(-16, 9)$, $(0, 5)$, $(18, 9)$ and spherical covariance matrices of standard deviation 2. The points were corrupted with 9% outliers. The resulting modeling of the noisy set by (b) a 3-component GMM, (c) a 4-component GMM with the fourth component modeling the distribution of outliers and (d) a 3-component SMM.

Finally, we have tested the efficiency of the proposed method to the registration of *shaped* or *structured* point sets, contrary to the scattered points of the previous example. This type of problems may come up from many computer vision applications such as comparison of trajectories in object tracking or shape discrimination and the presence of outliers makes registration difficult even if a good initialization is provided. To this end, we have applied the registration algorithm to data from the *Gaitor Bait 100* data base

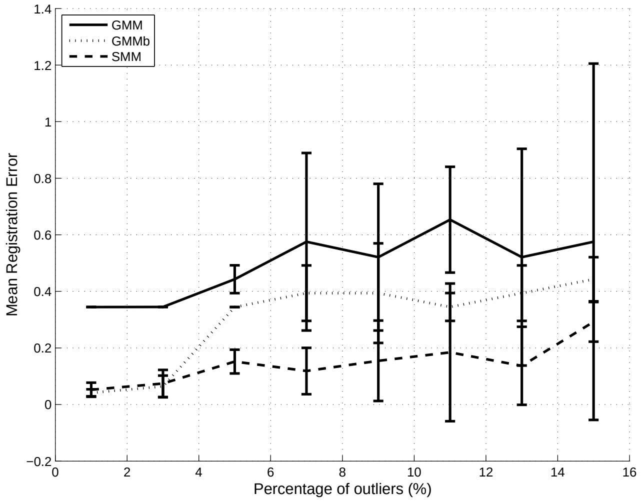


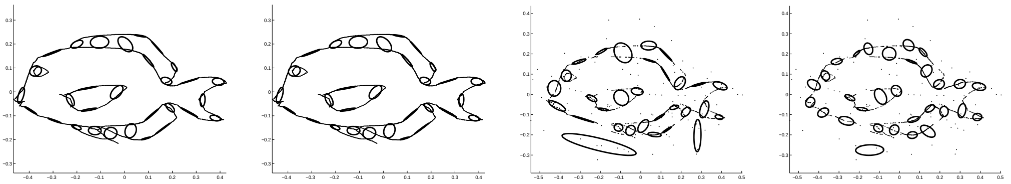
Figure 4.10: Registration error as a function of outliers for the experiment presented in figure 4.9.

(as provided by the Department of Computer and Information Science and Engineering, University of Florida, USA, http://www.cise.ufl.edu/).

In this experimental setting, we begin by illustrating the differences of the compared methods (GMM and SMM) in capturing the data. At first, the same shape, was modeled by a GMM (fig. 4.11(a)) and an SMM (fig. 4.11(b)) both with $K = 30$ components. The methods employed the same initialization by the K-means clustering algorithm. As it can be observed, both methods provided similar approximations. Consequently, the registration algorithm is not affected and the compared methods (GMM and SMM) provide equivalently good performances.

We then eliminated a certain amount of points by to simulate missing data and added outliers to the remaining points. In that case, we also used the same K-means initialization which naturally provided a certain number of centers that captured the structure of the outliers. However, in any case, the SMM modeled the degraded data better than the GMM by eliminating the majority of erroneous centers, due to its heavier tails. A representative example is presented in figures 4.11(c) and 4.11(d) where the missing data percentage is 20% and the percentage of outliers is 10%. In these figures, one can observe that the GMM finally provided two noisy components of relatively large covariance. On the other hand, due to the heavier tails of the SMM components, not only more outlier points were absorbed by the components located on the fish shape, but also the erroneous component has smaller support. This is important in a registration procedure because the L_2 distance in eq. (4.24) will be less influenced in the case of the SMM, as indicated by the experiments that follow.

The original point set was artificially rotated, translated and corrupted by outliers at 15%. The transformed point set was then registered to its original, noise free counterpart. We have compared the proposed GMM and SMM algorithms with the ICP by initializing them from the ground truth. The results are summarized in table 4.3, where it is clear



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

Figure 4.11: Modeling of a *shaped* point set from the GatorBait100 [2] data base by (a) GMM with $K = 30$ components and (b) SMM with $K = 30$ components. Notice that the two models provided similar solutions. The bottom row shows the modeling of the point set with 20% missing points and 10% outliers by (c) GMM and (d) SMM. Notice that the solution of the SMM was less affected. In all cases the mixtures were similarly initialized using the K-means algorithm. The axes in (c) and (d) are normalized to the range of the outliers.

that both of the proposed methods (GMM and SMM) perform better than the ICP. Also, SMM is more accurate than the less robust GMM. It is worth noticing that the ICP algorithm, as it is sensible to initialization, is always trapped around the same minimum.

Table 4.3: Registration errors for the *shaped* point set of figure 4.11 when it is corrupted by 15% outliers.

Method	mean	std	median	max	min
ICP	40.3784	15.8546	43.6067	58.0508	10.3555
GMM ($K = 15$)	2.6950	1.5169	2.8450	5.1540	0.5894
SMM ($K = 15$)	2.1136	0.8052	1.8880	3.5104	1.2366
GMM ($K = 20$)	2.4334	1.1380	2.4886	4.5563	0.9656
SMM ($K = 20$)	1.9506	0.9084	2.0361	3.4830	0.5927

Chapter 5

Epilogue

5.1 Conclusions

5.2 Future work

5.1 Conclusions

The objective of this thesis was twofold: to introduce a method for extracting features from images and sets of points for further analysis and to present a framework for solving the image and point set registration problem.

As far as the feature extraction is concerned, we focused on modeling data with line segments. We were motivated by the fact that line segments show simplicity and at the same time they can be combined into groups to model more complicated structures. The pioneering work of Hough Transform has been the basis for the development of many variants on the literature. A major problem of this class of methods is that they assume the number of lines as a prerequisite. Thus, the result is highly related to the tuning of the algorithm. Moreover, they are prone to erroneous detections. This observation, motivated us to propose a framework that tackles that problem by estimating the number of underlying line segments. Our method relies on two observations: i) the covariance matrix of the points belonging to a line segment produces an ellipse that is highly eccentric, in fact the linearity of the points is modeled by the minimum eigenvalue of the corresponding covariance matrix and ii) the points that belong to a line segment should follow a uniform distribution, which is explained by the fact that the distance between successive points is small. Eventually, those observations are quantifying some remarks of the Gestalt theory for human perception regarding linear structures. The proposed algorithm was described in chapter 1.

Considering line segments as informative features of an image or a set of points, some applications are then demonstrated based on the detected line segments. Chapter 2 is dedicated to explain those applications.

In the beginning we dealt with a basic problem of autonomous navigation, naming the detection of the vanishing point of a real scene. A lot of methods have been proposed for solving that problem. The most common workflow is to extract the edges of an image with an edge detector - most often the Canny edge detector is used - fit lines and then compute the common intersection point. This approach it is widely used. However, we noted that it demands a precise tuning of the line detection algorithm. A basic objective in this work was to preserve the simplicity of the workflow. Since an efficient line segment detection was introduced in a previous chapter, we focused on the common intersection point estimation process. Thus, an efficient voting scheme was established based on distributions that model a grid laying onto the image plane and collects votes. An important advantage of this approach is that it enables the establishment of a closed form solution. The complete development of this approach is a matter of ongoing research.

A line segment is defined by its direction along with its starting/ending points, as it establishes a framework for reproducing points in any desired density. This observation led us to the development of an algorithm for efficient sampling of shapes that preserves the initial distribution of points. Sampling is a common preprocessing step for many methods. In our study, we found that by adopting an efficient sampling method we managed to improve already proposed algorithms, related to shape retrieval. The sampling scheme was further developed and embedded in a shape reconstruction algorithm that enables the efficient compression of information with minimal loss.

Line segments are ideal for modeling tree structures like the retinal fundus image, since they permit to locate intersections that may be explained as bifurcations and junctions. A related algorithm was presented by defining a neighboring criterion based on the Euclidean distance.

Finally, a method for eliminating outliers was described. The algorithm is based on the Helmholtz principle regarding human perception and states that in a random generated image the expectation of observing a structure should be low, ideally zero. In this case, line segments serve as models of underlying structures that may appear and assist the computation of the corresponding probability of a line segment. The principle idea is that long line segments should be rare. A Pareto distribution was used to model the probability function of the random variable that describes the length of a line segment. The method was tested both in terms of shape extraction from heavily degraded point clouds and line fitting in a set were a large amount of outliers were present. Our method proved to be robust and efficient compared to other state-of-the-art and widely used methods.

The second part of this dissertation focused on image and point set registration. A framework that models the registration transformation was introduced. A Bayesian regression framework, namely the Relevance Vector Machines, was used to describe a non-rigid transformation. The basic characteristic of this approach is that it manages to handle false-matches without compromising the efficiency of the model, compared to the commonly used thin plate splines. Moreover, our method provides a closed form solution for the transformation that may be used for post computations.

The thesis concludes with the description of a method for solving the rigid image and point set registration problem, employing mixture models. By modeling the intensity distribution of the observed and the reference image/point set and measuring their distance under a rigid transformation, we compute a quantity that is minimized with respect to the transformation's parameters. The advantage of the method is that it can handle multi modal images providing simultaneously efficient results, compared to the state of the art methods.

5.2 Future Work

The output of this thesis may be the basis for further research, especially in the field of the analysis of sets of scattered points. The following topics are of interest for more detailed investigation:

- The Helmholtz principle may be adopted to eliminate the need of split/merge thresholds. The Helmholtz principle states that no perception should be produced on an image of noise. In other words, if we consider the input set of points as a random distribution of noise, then line segments should be less possible to be detected. Thus, by defining a model to compute the likelihood of an observed linear structure, we may declare this observation as valid if the corresponding probability is two small. This approach differs from the method introduced in [40, 41, 42] as it is more general and does not assumes the existence of a grid, as is the case for images.

- The development of a local area descriptor based on the line segments that can be used either for image/point set registration or matching. Line segments can be described by their start/end points, their length and direction. Relying on those features, we can produce a descriptor of the local neighborhood of the line segment and then employ it to compute the similarity between two line segments. If the line segments are associated with image edges, then their similarity metric is equal to the similarity of the corresponding image area that provided the edges. In [150], a line descriptor, that follows this rationale is introduced.

- The use of Kalman filters [151] to reduce the distortion of the proposed binary image compression framework based on the DSaM algorithm.

- The use of DSaM to detect more complex structures. For instance, line segments could be grouped based on the local curvature so as to extract more meaningful structures, such as traffic signs.

- The use of DSaM and the retinal fundus image annotation criteria to produce a graph that describes cracks on pavements and then extract graph based features to classify them.

- The use of DSaM as a preprocessing step for extracting line segments for vectorizing raster images, like the line drawings mentioned in [152].

Bibliography

[1] Dataset: MPEG-7. college of science and technology, temple university. (2002)

[2] Dataset: GatorBait 100, University of Florida (2008)

[3] Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004) 550–571

[4] Ferrari, V., Tuytelaars, T., Gool, L.V.: Object detection by contour segment networks. In: European Conference on Computer Vision. (2006) 14–28

[5] Kovesi, P.D.: Matlab and Octave functions for vomputer vision and image processing (2009)

[6] Gerogiannis, D., Nikou, C., Likas, A.: Modeling sets of unordered points using highly eccentric ellipses. EURASIP Journal on Advances in Signal Processing (2014)

[7] Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using Shape Contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 1–19

[8] Medioni, G., Lee, M.S., Tang, C.K.: A Computational Framework for Segmentation and Grouping. Elsevier (2000)

[9] P. Mordohai and G. G. Medioni: Tensor Voting. A Perceptual Organization Approach to Computer Vision and Machine Learning. Morgan and Claypool (2007)

[10] Ramer, U.: An iterative procedure for the polygonal approximation of plane curves. Computer Graphics and Image Processing (1972) 244–256

[11] O'Connell, K.J.: Object-adaptive vertex-based shape coding method. IEEE Transactions on Circuits and Systems for Video Technology **7** (1997) 251–255

[12] Xiaochao, W., Xiuping, L., Hong, Q.: Robust surface consolidation of scanned thick point clouds. In: International Conference on Computer-Aided Design and Computer Graphics. (2013) 38–43

[13] Ester, M., Kriegel, H.P., Jórg, S., Xiaowei, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: International Conference on Knowledge Discovery and Data Mining, AAAI Press (1996) 226–231

[14] Chui, H., Rangarajan, A.: A new algorithm for non-rigid point matching. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2000) 40–51

[15] Bookstein, F.L.: Principal Warps: Thin-Plate Spilnes and the decomposition of deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence **11** (1989) 567–585

[16] Tsin, Y., Kanade, T.: A correlation-based approach to robust point set registration. In: European Conference on Computer Vision. (2004) 558–569

[17] Myronenko, A., Song, X.: Point set registration: Coherent Point Drift. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 2262–2275

[18] Cantoni, V., Lombardi, L., Porta, M., Sicard, N.: Vanishing point detection: representation analysis and new approaches. In: 11th International Conference on Image Analysis and Processing. (2001) 90–94

[19] Dori, D., Wenyin, L.: Sparse pixel vectorization: An algorithm and its performance evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 202–215

[20] Se, S., Brady, M.: Road feature detection and estimation. Machine Vision and Applications **14** (2003) 157–165

[21] Kolesnikov, A.: ISE-bounded polygonal approximation of digital curves. Pattern Recognition Letters **33** (2012) 1329–1337

[22] Illingworth, J., Kittler, J.: A survey of the Hough Transform. Computer Vision, Graphics, and Image Processing **44** (1988) 87–116

[23] Leavers, V.F.: Which Hough Transform? Computer Vision, Graphics and Image Processing: Image Understanding **58** (1993) 250–264

[24] Xu, L., Oja, E., Kultanen, P.: A new curve detection method: Randomized Hough Transform (RHT). Pattern Recognition Letters (1990) 331–338

[25] McLaughlin, R.A.: Randomized Hough Transform: Improved ellipse detection with comparison. Pattern Recognition Letters **19** (1998) 299–305

[26] Kiryati, N., Eldar, Y., Bruckstein, A.M.: A probabilistic Hough Transform. Pattern Recognition **24** (1991) 303–316

[27] Matas, J., Galambos, C., Kittler, J.: Progressive probabilistic Hough Transform. In: British Machine Vision Conference. Volume I., London, UK (1998) 256–265

[28] Chung, K., Lin, Z., Huang, S., Huang, Y., Liao, H.M.: New orientation-based elimination approach for accurate line-detection. Pattern Recognition Letters **31** (2010) 11–19

[29] Chatzis, V., Pitas, I.: Fuzzy cell Hough Transform for curve detection. Pattern Recognition **30** (1997) 2031–2042

[30] Kälviäinen, H., Hirvonen, P., Xu, L., Oja, E.: Probabilistic and non-probabilistic Hough transforms: overview and comparisons. Image and Vision Computing **13** (1995) 239–252

[31] Atiquzzaman, M., Akhtar, M.W.: A robust Hough Transform technique for complete line segment description. Real Time Imaging (1995) 419–426

[32] Chau, C., Siu, W.: Adaptive dual-point Hough Transform for object recognition. Computer Vision and Image Understanding **96** (2004) 1–16

[33] Cheng, H.D., Guo, Y., Zhang, Y.: A novel Hough Transform based on eliminating particle swarm optimization and its applications. Pattern Recognition **42** (2009) 1959–1969

[34] Moore, G.A.: Automatic scanning and computer process for the quantitative analysis of micrographs and equivalent subjects. Pictorial Pattern Recognition (1968) 275–362

[35] Li, C., Wang, Z., Li, L.: An improved Hough Transform algorithm on straight line detection based on Freeman chain code. In: 2nd International Congress on Image and Signal Processing. (2009) 1–4

[36] Martin, F., Robert, B.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–385

[37] Bhowmick, P., Bhattacharya, B.: Fast polygonal approximation of digital curves using relaxed straightness properties. IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007) 1590–1602

[38] Leite, J., Hancock, E.: Iterative curve organisation with the EM algorithm. Pattern Recognition Letters **18** (1997) 143–155

[39] Desolneux, A., Moisan, L., Morel, J.M.: Meaningful allignments. International Journal of Computer Vision **40** (2000) 7–23

[40] von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: On straight line segment detection. Journal of Mathematical Imaging and Vision **32** (2008) 313–347

[41] von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: A fast line segment detector with a false detection control. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 722–732

[42] von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: A line segment detector. Image Processing On Line **2** (2012) 35–55

[43] Marr, D.: Vision. Freeman Publishers, New York (1982)

[44] Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986) 679–698

[45] Bishop, C.M.: Pattern recognition and machine learning. Springer, Dordrecht, The Netherlands (2006)

[46] Matas, J., Galambos, C., Kittler, J.: Robust detection of lines using the progressive probabilistic Hough Transform. Computer Vision and Image Understanding **78** (April 2001) 119–137

[47] Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV. O'Reilly Media, Sebastoupol, California (2008)

[48] Coughlin, J.M., Yuille, A.: Manhattan world: compass direction from a single image by Bayesian inference. In: IEEE International Conference on Computer Vision. Volume 2. (1999) 941–947

[49] Denis, P., Elder, .H., Estrada, F.J.: Efficient edge-based methods for estimating Manhattan frames in urban imagery. In: European Conference on Computer Vision ECCV. Volume 2. (2008) 107–210

[50] Almansa, A., Desolneux, A., Vamech, S.: Vanishing point detection without any a priori information. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 502 – 507

[51] Kosecka, J., Zhang, W.: Video compass. In: European Conference on Computer Vision. (2002) 476–491

[52] Tardif, J.P.: Non-iterative approach for fast and accurate vanishing point detection. In: IEEE International Conference on Computer Vision, Kyoto, Japan (September 27 - October 4, 2009) 1250–1257

[53] Rother, C.: A new approach to vanishing point detection in architectural environments. Image and Vision Computing **20** (2002) 647–655

[54] Li, B., Peng, K., Ying, X., Zha, H.: Vanishing point detection using cascaded 1d Hough Transform from single images. Patter Recognition Letters **33** (2012) 1–8

[55] Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: IEEE International Conference one Computer Vision. Volume 2. (2001) 416–423

[56] Alivanoglou, A., Likas, A.: Probabilistic models based on the Pi-Sigmoid distribution. In: 3rd IAPR Workshop on Artificial Neural Networks in Pattern Recognition. (2008) 36–43

[57] University, B.: Matching with Shape Contexts (2002)

[58] Moenning, C., Dodgson, N.A.: Fast marching farthest point sampling. In: EUROGRAPHICS. (2003)

[59] Bronstein, A., Bronstein, M., Bronstein, M., Kimmel, R.: Numerical geometry of non-rigid shapes. Springer-Verlag New York Inc (2008)

[60] Chen, J.J., Chiang, C.C., Lin, D.W.: A generalized 3d shape sampling method and file format for storage or indexing. In: International Conference on Image Processing. Volume 2. (2000) 780–783

[61] Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Transactions on Graphics **21** (2002) 807–832

[62] Wang, J., Bai, X., You, X., Liu, W., Latecki, J.: Shape matching and classification using height functions. Pattern Recognition Letters **33** (2012) 134–143

[63] Azariadis, P.N., Sapidis, N.S.: Drawing curves onto a cloud of points for point-based modelling. Computer-Aided Design **37** (2005) 109–122

[64] Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290** (2000) 2323–2326

[65] Std., I.I.: Information technology – coding of audio-visual objects – part 2: Visual coding. http://www.digitalpreservation.gov/formats/fdd/fdd000080.shtmln (1999) Accessed on January, 2015.

[66] Freeman, H.: Computer processing of line-drawing images. ACM Comput. Surv. **6** (1974) 57–97

[67] Katsaggelos, A.K., Kondi, L.P., Meier, F.W., Ostermann, J., Schuster, G.M.: MPEG-4 and rate-distortion-based shape-coding techniques. Proceedings of the IEEE **86** (1998) 1126–1154

[68] Aghito, S.M., Forchhammer, S.: Context-based coding of bilevel images enhanced by digital straight line analysis. IEEE Transactions on Image Processing **15** (2006) 2120–2130

[69] Klette, R., Rosenfeld, A.: Digital straightness - a review. Discrete Applied Mathematics **139** (2004) 197–230

[70] 11544, I.I.S.: Coded representension of picture and audio information - progressive bi-level image compression (1993)

[71] 14492, I.I.S.: Coded representension of picture and audio information - lossy/lossless coding of bi-level images (JBIG2) (2000)

[72] Aghito, S.M., Forchhammer, S.: Efficient coding of shape and transparency for video objects. IEEE Transactions on Image Processing **16** (2007) 2234–2244

[73] Sanchez-Cruz, H.: Proposing a new code by considering pieces of discrete straight lines in contour shapes. Journal of Visual Communication and Image Representation **21** (2010) 311–324

[74] Akimov, A., Kolesnikov, A., Fränti, P.: Lossless compression of map contours by context tree modeling of chain codes. Pattern Recognition **40** (2007) 944–952

[75] Pinho, A.J.: A JBIG-based approach to the encoding of contour maps. IEEE Transactions on Image Processing **9** (2000) 936–941

[76] Bresenham, J.E.: Algorithm for computer control of a digital plotter. IBM Systems Journal **4** (1965) 25–30

[77] ITU-T: T.6: Facsimile coding schemes and coding control functions for group 4 facsimile apparatus. http://www.itu.int/rec/T-REC-T.6/en (1984) Accessed on January, 2015.

[78] Wang, H., Schuster, G.M., Katsaggelos, A.K., Pappas, T.N.: An efficient rate-distortion optimal shape coding approach utilizing a skeleton-based decomposition. IEEE Transactions on Image Processing **12** (2003) 1181–1193

[79] Ausbeck, J.P.J.: The piecewise-constant image model. IEEE Proceedings (2000) 1779–1789

[80] : Djvu libre (2015)

[81] Patton, N., Aslam, T.M., MacGillivray, T., Dearye, I.J., Dhillonb, B., Eikelboomf, R.H., Yogesana, K., Constablea, I.J.: Retinal image analysis: Concepts, applications and potential. Progress in Retinal and Eye Research **25** (2006) 99–127

[82] Azzopardi, G., Petkov, N.: Detection of retinal vascular bifurcations by trainable V4-like filters. In: 14th International Conference on Computer Analysis of Images and Patterns, Spain (2011) 451–459

[83] Staal, J., Abramoff, M., Niemeijer, M., Viergever, M., van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging **23** (2004) 501–509

[84] Azzopardi, G., Petkov, N.: Retinal fundus images - Ground truth of vascular bifurcations and crossovers (2011)

[85] Lemaitre, C., Perdoch, M., Rahmoune, A., Matas, J., Miteran, J.: Detection and matching of curvilinear structures. Pattern Recognition **44** (2011) 1514–1527

[86] Sofka, M., Stewart, C.V.: Retinal vessel extraction using multiscale matched filters, confidence and edge measures. IEEE Transactions on Medical Imaging **25** (2006) 1531–1546

[87] Gonzalez, R.C., Woods, R.E.: Digital image processing, 3rd edition. Prentice Hall, Upper Saddle River, New Jersey (2008)

[88] C. C. Aggarwal: Outlier Analysis. Springer (2013)

[89] Thompson, R.: A note on restricted maximum likelihood estimation with an alternative outlier model. Journal of the Royal Statistical Society **47** (1985) 53–55

[90] Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (PCL). In: IEEE International Conference on Robotics and Automation, Shanghai, China (2011)

[91] Schall, O., Belyaev, A., Seidel, H.P.: Robust filtering of noisy scattered point data. In: Eurographics Symposium on Point-Based Graphics. (2005) 71–77

[92] Weyrich, T., Pauly, M., Keiser, R., Heinzle1, S., Scandella, S., Gross1, M.: Post-processing of scanned 3d surface data. In: Eurographics Symposium on Point-Based Graphics. (2004) 85–94

[93] Ertöz, L., Steinbach, M., Kumar, V.: Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In: 2^{nd} SIAM International Conference on Data Mining. (2003)

[94] Xie, H., McDonnell, K.T., Hong, Q.: Surface reconstruction of noisy and defective data sets. In: IEEE Visualization. (2004) 259–266

[95] Desolneux, A., Moisan, L., Morel, J.M.: Gestalt theory and computer vision. In: Seeing, Thinking and Knowing. Volume 38. Kluwer Academic Publishers (2004) 71–101

[96] Desolneux, A., Moisan, L., Morel, J.M.: Edge detection by Helmholtz principle. Journal of Mathematical Imaging and Vision **14** (2001) 271–284

[97] Desolneux, A., Moisan, L., Morel, J.M.: From Gestalt Theory to Image Analysis: a Probabilistic Approach. Springer (2008)

[98] Arnold, B.C.: Pareto distributions. Statistical distributions in scientific work series. International Co-operative Publishing House (1983)

[99] Dumouchel, W., O'Brien, F.: Integrating a robust option into a multiple regression computing environment. In: Computing and Graphics in Statistics. Springer-Verlag New York, Inc. (1991) 41–48

[100] Besl, P.J., McKay, N.D.: A method for registration of 3d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992) 239–256

[101] Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: The 3^{rd} International Conference on 3D Digital Imaging and Modeling. (2001)

[102] Gold, S., Lu, C.P., Rangarajan, A., Pappu, S., Mjolsness, E.: New algorithms for 2d and 3d point matching: Pose estimation and correspondence. In Tesauro, G., Touretzky, D., Leen, T., eds.: Advances in Neural Information Processing Systems. Volume 7., The MIT Press (1995) 957–964

[103] Jian, B., Vemuri, B.C.: Robust point set registration using Gaussian mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011) 1633–1645

[104] Gerogiannis, D., Nikou, C., Likas, A.: The mixtures of Student's *t*-distributions as a robust framework for rigid registration. Image and Vision Computing **27** (2009) 1285–1294

[105] Tu, Z., Zheng, S., Yuille, A.: Shape matching and registration by data-driven EM. Computer Vision and Image Understanding **109** (2008) 290–304

[106] Suk, T., Flusser, J.: Point-based projective invariants. Pattern Recognition **33** (2000) 251–261

[107] Caetano, T.S., Caelli, T., Schuurmans, D., Barone, D.A.C.: Graphical models and point pattern matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006) 1646–1663

[108] Zheng, Y., Doermann, D.S.: Robust point matching for non-rigid shapes by preserving local neighborhood structures. IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006) 643–649

[109] Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research **1** (2001) 211–244

[110] Tan, L., Fyfe, C.: Canonical correlation analysis - a relevance vector approach. In: The 9^{th} European Symposium on Artificial Neural Networks. (2001)

[111] Iwai, Y., Cipolla, R.: Multivariate sparse Bayesian regression and its application for facial feature detection. In: IAPR Conference on Machine Vision Applications. (2005)

[112] Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. Computer Vision and Image Understanding **89** (2003) 114–141

[113] Papadimitriou, C.H., Steiglitz, K.: Combinatorial optimization. Algorithms and complexity. Dover (1998)

[114] Dempster, P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society **39** (1977) 1–38

[115] Tipping, M.: Matlab code for univariate RVM implementation (2006)

[116] Zitova, B., Flusser, J.: Image registration methods: a survey. Image and Vision Computing **21** (2003) 1–36

[117] Maintz, J.B.A., Viergever, M.A.: A survey of medical image registration techniques. Medical Image Analysis **2** (1998) 1–36

[118] Bankman, I.: Handbook of medical image processing and analysis. Academic Press (2000)

[119] Hajnal, J.V., Hill, D.L.G., Hawkes, D.J.: Medical image registration. CRC Press (2001)

[120] Nikou, C., Heitz, F., Armspach, J.P.: Robust voxel similarity metrics for the registration of dissimilar single and multimodal images. Pattern Recognition **32** (1999) 1351–1368

[121] Jensen, J.R.: Introductory digital image processing: a remote sensing perspective. 3rd edn. Prentice Hall (2004)

[122] Heitz, F., Maître, H., de Couessin, C.: Event detection in multisource imaging: application to fine arts painting analysis. IEEE Transactions on Acoustic, Speech and Signal Processing **38** (1990) 695–704

[123] Viola, P., W. Wells III: Alignment by maximization of mutual information. International Journal of Computer Vision **24** (1997) 137–154

[124] Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., Suetens, P.: Multimodality image registration by maximization of mutual information. IEEE Transactions on Medical Imaging **16** (1997) 187–198

[125] Pluim, J., Maintz, J.B.A., Viergever, M.: Mutual information based registration of medical images: a survey. IEEE Transactions on Medical Imaging **22** (2004) 986–1004

[126] Hermosillo, G., Faugeras, O.: Dense image matching with global and local statistical criteria. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 1., Los Alamitos, USA (2001) 73–78

[127] Thévenaz, P., Unser, M.: Optimization of mutual information for multiresolution image registration. IEEE Transactions on Image Processing **9** (2000) 2083–2099

[128] Mattes, D., Haynor, D.R., Vesselle, H., Lewellen, T.K., Eubank, W.: Nonrigid multimodality image registration. In Hanson, K., ed.: SPIE Medical Imaging Conference. Volume 4322., San Diego, USA (2003) 1609–1620

[129] Rajwade, A., Banerjee, A., Rangarajan, A.: A new method of probability density estimation with application to mutual information based image registration. In: IEEE Conference on Computer Vision and Pattern Recognition, New York, USA (2006) 1769–1776

[130] McLachlan, G.: Finite mixture models. Wiley-Interscience (2000)

[131] Leventon, M., Grimson, E.: Multi-modal volume registration using joint intensity distributions. In: Medical Image Computing and Computer Assisted Intervention Conference. (1998) 1057–1066

[132] Guimond, A., Roche, A., Ayache, N., Meunier, J.: Three-dimensional multimodal brain warping using the demons algorithm and adaptive intensity corrections. IEEE Transactions on Medical Imaging **20** (2001) 58–69

[133] Hellier, P.: Consistent intensity correction of MR images. In: IEEE International Conference on Image Processing. Volume 1., Barcelona, Spain (2003) 1109–1112

[134] Jian, B., Vemuri, B.C.: A robust algorithm for point set registration using mixture of gaussians. In: IEEE International Conference on Computer Vision, Beijing, China (2005) 1246–1251

[135] Herbin, M., Venot, A., Devaux, J.Y., Walter, E., Lebruchec, F., Dubertet, L., Roucayrol, J.C.: Automated registration of dissimilar images: application to medical imagery. Computer Vision, Graphics and Image Processing **47** (1989) 77–88

[136] Roche, A., Pennec, X., Malandain, G., Ayache, N.: Rigid registration of 3d ultrasound with MR images: a new approach combining intensity and gradient information. IEEE Transactions on Medical Imaging **20** (2001) 1038–1049

[137] Bishop, C., Svensen, M.: Robust Bayesian mixture modelling. Neurocomputing (2005) 69–74

[138] Peel, D., McLachlan, G.J.: Robust mixture modeling using the *t*-distribution. Statistics and Computing **10** (2000) 339–348

[139] Granger, S., Pennec, X.: Multi-scale EM-ICP: A fast and robust approach for surface registration. In: European Conference on Computer Vision. (2002) 418–432

[140] Chetverikov, D., Stepanov, D., Krsek, P.: Robust Euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. Image and Vision Computing **23** (2004) 299–309

[141] Phillips, J., Liu, R., Tomasi, C.: Outlier robust ICP for minimizing RMSD. In: The 6^{th} International Conference on 3D Digital Imaging and Modeling. (2007)

[142] Rangarajan, A., Chui, H., Bookstein, F.L.: The softassign procrustes matching algorithm. In: The 15^{th} International Conference on Information Processing in Medical Imaging. Lecture Notes in Computer Science. Volume 1230. (1997) 29–42

[143] Rangarajan, A., Chui, H., Duncan, S.: Rigid point feature registration using mutual information. Medical Image Analysis **4** (1999) 1–17

[144] Taron, M., Paragios, N., Jolly, M.: Registration with uncertainties and statistical modeling of shapes with variable metric kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009) 99–113

[145] Myronenko, A., Song, X., M. Á. Carreira-Perpiñán: Non-rigid point set registration: Coherent Point Drift. In Schölkopf, B., Platt, J., Hoffman, T., eds.: The 20^{th} Conference on Advances in Neural Information Processing Systems. MIT Press, Vancouver, Canada (2006) 1009–1016

[146] Wang, F., Vemuri, C.B., Rangarajan, A., Schmalfuss, M.I., Eisenschenk, J.S.: Simultaneous non-rigid registration of multiple point sets and atlas construction. In: European Conference on Computer Vision. (2006) 551–563

[147] Fukunaga, K.: Introduction to statistical pattern recognition. Academic Press (1990)

[148] Brandt, S.S.: Maximum likelihood robust regression by mixture models. Journal of Mathematical Imaging and Vision **25** (2006) 25–48

[149] Pluim, J., Maintz, J.B.A., Viergever, M.: Interpolation artefacts in mutual information-based image registration. Computer Vision and Image Understanding **77** (2000) 211–232

[150] Wang, L., U. Neumann, U., You, S.: Wide-baseline image matching using line signatures. In: International Conference on Computer Vision. (2009) 1311–1318

[151] Prince, S.J.: Computer vision. Models, learning and inference. Cambridge University Press (2012)

[152] Hilaire, X., Tombre, K.: Robust and accurate vectorization of line drawings. IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006) 8901–904

[153] Bourgeois, F., Lassalle, J.C.: An extension of the Munkres algorithm for the assignment problem to rectangular matrices. Communications of the ACM **12** (Dec. 1971) 802–804

Appendix

I The Hungarian algorithm

The Hungarian algorithm is a combinatorial optimization method which solves the assignment problem. Assume that there are m tasks that have to be assigned to n workers. Each assignment is weighted with a cost (or profit), thus a complete bipartite weighted graph is produced, having as vertices the workers and the tasks. The goal is to calculate that particular assignment such that the total cost is minimum. The assignment has to be one to one. Sometimes the algorithm is used to maximize the total profit. In that case, we subtract the maximum entry of the cost matrix from all its cells. In case $m \neq n$, the problem is called unbalanced and the standard Hungarian algorithm may provide a false solution. A modification of the algorithm to handle rectangular cost matrices is introduced in [153]. Algorithm 10 presents the steps of this modification. In algorithms 9, 10 the terms starred, primed, covered and uncovered are characterizations assigned to a zero element (stared, primed) or to rows and columns (covered, uncovered), that guides the execution of the algorithm and distinguish the examined elements (zeros and rows or columns). The algorithm along with a detailed description may also be found in [153].

Hereafter we consider that $m = n$, and thus we will exploit only variable n to indicate the dimension of the problem. The output of the Hungarian algorithm is the optimal assignment, that minimizes the total cost. The complexity of the algorithm is $\mathcal{O}(n^3)$ in case of a balanced problem, while it may be increased in case of unbalanced problems, as a lot of trials are made to extract the solution of the problem.

More specifically, suppose we have a weighted undirected bipartite graph with n nodes, with c_{ij} indicating the weight of edge from node i to node j. The variable δ_{ij}, where $i, j \in \{1, \ldots, n\}$ indicates whether edge (i, j) is included in the matching. More specifically, $\delta_{ij} = 1$ means that the corresponding edge is included in the matching, whereas $\delta_{ij} = 0$ signifies that the edge (i, j) is not part of the matching process. The following restrictions apply:

- $\sum_{i=1}^{n} \delta_{ij} = 1$,
- $\sum_{j=1}^{n} \delta_{ij} = 1$,
- $\delta_{ij} > 0, \forall i, j \in \{1, \ldots, n\}$.

The goal of the Hungarian algorithm is the following:

Given a n × n matrix $\mathcal{C}$, where $\mathcal{C}_{ij}$ is the weight of assigning worker i with task j, minimize $\sum_{i=1}^{n}\sum_{j=1}^{n}\delta_{ij}\mathcal{C}_{ij}$.

The steps of the Hungarian algorithm, or Hungarian method as it is met regularly in the literature are described in algorithm 9. Details may be found in [113].

```
 1: From each row subtract off the row min.
 2: From each column subtract off the row-column min.
 3: Use as few lines (vertical, horizontal) as possible to cover all rows and columns con-
    taining zeros in the matrix (trial and error). Suppose $k$ lines are used for covering.
 4: if $k < n$ then
 5:     Let $m$ be the minimum uncovered number.
 6:     Subtract $m$ from every uncovered number.
 7:     Add $m$ to every number covered with two lines.
 8:     goto 3.
 9: else if $k = n$ then
10:     goto 11.
11: Starting with the top row, go downwards making assignments. An assignment can be
    (uniquely) made only whenthere is exactly one zero in the row.
```

Algorithm 9: The Hungarian algorithm for square cost matrices

In original version, the Hungarian algorithm assumes a square cost matrix, i.e. equal number of tasks and workers. A modification of the algorithm to handle rectangular cost matrices is introduced in [153], solving thus problems with different number of workers and tasks (unbalanced problems). Algorithm 10 presents the related procedure. The reader should notice that our goal is to propose a method that can model a registration transformation upon an assignment between two point sets has been determined. The Hungarian algorithm is a solution to that problem. In order to provide a complete framework, the revised Hungarian algorithm, that handles unbalanced sets is also included in our work.

II Relevance Vector Machines

The RVM model can be used to solve either the problem of classification or regression. In general, in order to use a RVM, we have to assume that we have a set of examples of input vectors $X = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^{N}$ along with corresponding scalar targets $\mathbf{t} = \{t_i\}_{i=1}^{N}$. Our goal is to train a model so as to learn the functional mapping between input vectors $\mathbf{x}_i$ and targets t_i. Since the points in a registration problem lay in a continuous space, it is implied that the target variable $\mathbf{t}$ is continuous, leading to a regression problem. A detailed description of RVM theory may be found in [109] and [45].

More specifically, we seek that particular model f with parameters $\mathbf{w} = \{w_1, w_2, \ldots, w_N\}$ such that $f(\mathbf{x}_i; \mathbf{w}) \simeq t_i, i = 1, \ldots, N$, assuming that $\mathbf{x}_i$ corresponds to t_i. The model f

```
1: Let $k = min(n, n)$ and $l = max(n, m)$ for a cost matrix $A$, $m \times n$.
2: if number of rows is larger than number of columns then
3:    goto 3.
4: if number of rows is less than number of columns then
5:    goto 12.
6: Update A:
7: for all row of A do
8:    Subtract the minimum element from each element in the row.
9: for all column of A do
10:    Subtract the minimum element from each element in the column.
11: for all zeros of matrix A do
12:    Find a zero at location Z of the matrix A.
13:    if there is no starred zero in its row nor its column then
14:       Star Z.
15: Cover every column containing a $0^*$.
16: if k columns are covered then
17:    {The starred zeros form the desired independent set (assignment solution).}
18:    STOP.
19: for all all zeros are covered do
20:    Choose a non covered zero and prime it; then consider the row containing it.
21:    if there is no starred zero Z in this row then
22:       goto 26.
23:    if there is a starred zero Z in this row then
24:       Cover this row and uncover the column of Z.
25: goto 35.
26: There is a sequence of alternating starred and primed zeros constructed as follows:
27: repeat
28:    Let $Z_0$ denote the uncovered $0'$.
29:    Let $Z_1$ denote the $0^*$ in $Z_0$'s column (if any).
30:    Let $Z_2$ denote the $0\prime$ in $Z_l$'s row.
31: until The sequence stops at a $0\prime$, $Z_{2k}$, which has no $0^*$ in its column.
32: Unstar each starred zero of the sequence, and star each primed zero of the sequence.
33: Erase all primes and uncover every line.
34: goto 15.
35: Let h denote the smallest non covered element of the matrix; it will be positive.
36: Add h to each covered row; then subtract h from each uncovered column.
37: goto 18 without altering any asterisks, primes, or covered lines.
```

Algorithm 10: The Hungarian algorithm for rectangular cost matrices (unbalanced problems)

may be analyzed into a finite linear sum of N non-linear functions ϕ_j, called kernels. Thus,

$$f(\mathbf{x}_i;\mathbf{w}) = \sum_{j=1}^{N} w_j \phi_j(\mathbf{x}_i) = \mathbf{w}^T \Phi(\mathbf{x}_i), \tag{5.1}$$

where $\Phi(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \phi_2(\mathbf{x}_i), \ldots, \phi_N(\mathbf{x}_i))^T$.

Assume now that the targets $\{t_i\}_{i=1}^{N}$ are samples drawn from the model with additive noise ϵ_i:

$$t_i = f(\mathbf{x}_i;\mathbf{w}) + \epsilon_i \tag{5.2}$$

where ϵ_i are independent samples from some noise process. Hereafter we will assume a Gaussian distribution with zero mean and variance σ^2 for ϵ_i. Thus, a probability density model occurs:

$$p(t_i|\mathbf{x}_i) = \mathcal{N}(t_i|f(\mathbf{x}_i;\mathbf{w}), \sigma^2), \tag{5.3}$$

where $\mathcal{N}$ is a Gaussian distribution over t_i with mean $f(\mathbf{x}_i;\mathbf{w})$ and variance σ^2.

A second assumption concerns the statistical independence of target variables t_i. The likelihood of the target vector $\mathbf{t}$ is

$$p(\mathbf{t}|\mathbf{w},\sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{t} - \mathbf{\Phi}\mathbf{w}\|^2\right\}, \tag{5.4}$$

where $\mathbf{t} = (t_1 \ldots t_N)^T$, $\mathbf{w} = (w_1 \ldots w_N)^T$ and $\mathbf{\Phi} = (\Phi(x_1) \ldots \Phi(x_N))$.

In Bayesian methodology, a common practice to prevent over-fitting, caused by the large number of parameters, is to impose some additional constraints, penalizing the complexity of the model. These hyperparameters are imposed over parameters $\mathbf{w}$ of the linear model in (5.1). The goal is to reduce the number of discrete functions of the sum, thus occurring a less complex model. This is achieved by adopting a zero-mean Gaussian prior over $\mathbf{w}$, or

$$p(\mathbf{w}|\mathbf{a}) = \prod_{i=1}^{N} \mathcal{N}(w_i|0, a_i^{-1}), \tag{5.5}$$

where $\mathbf{a} = (a_1 \ldots a_N)^T$ with a_i representing the precision of the corresponding parameter w_i. One can explain these hyperparameters as selectors over each parameter w_i which is the weight of function ϕ_i participating in the total sum. If the variance of the corresponding prior is large then the resulting probability is low, eliminating the term in the sum. This means that the corresponding basis function $\phi_i(\mathbf{x}_j)$ plays no role in the prediction made by the model.

The posterior distribution of weights is Gaussian and takes the form

$$p(\mathbf{w}|\mathbf{t}, X, \mathbf{a}, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m}, \mathbf{\Sigma}), \tag{5.6}$$

where β is the inverse of σ in (5.4) and

$$\mathbf{m} = \beta\mathbf{\Sigma}\mathbf{\Phi}^T\mathbf{t}, \tag{5.7}$$

$$\mathbf{\Sigma} = (\mathbf{A} + \beta\mathbf{\Phi}^T\mathbf{\Phi})^{-1}, \tag{5.8}$$

with $\mathbf{A} = diag\{a_i\}$.

Eventually, an iterative learning process occurs. Initially, we choose some values for $\mathbf{a}$, β, thus evaluating the mean and covariance of the posterior using (5.7) and (5.8). Then we iterate, until a convergence criterion is satisfied, by re-estimating the hyperparameters:

$$a_i = \frac{\gamma_i}{m_i^2}, \tag{5.9}$$

$$\beta^{-1} = \frac{\|\mathbf{t} - \mathbf{\Phi m}\|^2}{N - \sum_{i=1}^{N} \gamma_i}, \tag{5.10}$$

where m_i is the i^{th} component of the posterior mean defined by (5.7). and the quantity γ_i is computed as:

$$\gamma_i = 1 - a_i \Sigma_{ii}, \tag{5.11}$$

where Σ_{ii} is the i_{th} diagonal component of the covariance matrix $\mathbf{\Sigma}$ given by (5.8).

The result of the training process described above is learning parameters $\mathbf{w}$ of equation (5.1).

Author's Publications

Journal Publications

J1 D. Gerogiannis, C. Nikou and A. Likas. The mixtures of Student's *t*-distributions as a robust framework for rigid registration. **Image and Vision Computing**, Vol. 27, No 9, pp. 1285-1294, 2009.

J2 D. Gerogiannis, C. Nikou and A. Likas. Registering sets of points using Bayesian regression. **Neurocomputing**, Vol. 89, pp.122-133, 2012.

J3 D. Gerogiannis, C. Nikou and A. Likas. Modeling sets of unordered points using highly eccentric ellipses. **EURASIP Journal on Advances in Signal Processing**, 2014:11, 2014.

J4 D. Gerogiannis, C. Nikou and A. Likas. Elimination of outliers from 2D point sets using the Helmholtz principle. **Submitted**.

Conference Publications

C1 D. Gerogiannis, C. Nikou and A. Likas. Rigid image registration based on pixel grouping. In Proceedings of the 14th International Conference on Image Analysis and Processing (**ICIAP'07**), 10-14 September 2007, Modena, Italy.

C2 D. Gerogiannis, C. Nikou and A. Likas. Robust image registration using mixtures of t-distributions. Proceedings of the 8th IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis (**MMBIA'07**), in conjunction with ICCV'07, 14-20 October 2007, Rio de Janeiro, Brazil.

C3 D. Gerogiannis, C. Nikou and A. Likas. A split-and-merge framework for 2D shape summarization. 7th International Symposium on Image and Signal Processing and Analysis (**ISPA'11**), pp. 206-2011, 4-6 September 2011, Dubrovnik, Croatia.

The reader may find a digital copy of all the above publications at my personal academic webpage (www.cs.uoi.gr/∼dgerogia).

C4 D. Gerogiannis, C. Nikou and A. Likas. Fast and efficient vanishing point detection in indoor images. International Conference on Pattern Recognition (**ICPR'12**), pp. 3244-3247, 11-15 November 2012, Tsukuba, Japan.

C5 D. Gerogianis, C. Nikou and A. Likas. Global sampling of image edges. IEEE International Conference on Image Processing (**ICIP'14**), 27-30 October 2014, Paris, France.

C6 D. Gerogiannis, C. Nikou and L. P. Kondi. Shape encoding for edge map image compression. **Submitted**.

Other publications not related to this thesis

P1 D. Gerogiannis, C. Nikou. Tex-Lex: Automated generation of texture lexicons using images from the World Wide Web. International Conference on Digital Signal Processing (**DSP'13**), 1-3 July 2013, Santorini, Greece.

P2 A. Giotis, D. Gerogiannis and C. Nikou. Word spotting in handwritten text using contour-based models. 14th International Conference on Frontiers in Handwriting Recognition (**ICFHR'14**), 1-4 September 2014, Hersonisos, Crete, Greece, pp. 399-404, 2014.

P3 D. Gerogiannis, S. A. Sofianos, I. E. Lagaris, G. A. Evangelakis. One-Dimensional Inverse Scattering Problem in Acoustics, **Brazilian Journal of Physics**, Vol. 41, pp. 248-257, 2011.

Short Vita

I was born in Ioannina, Greece in February 1983. I received my BSc and MSc both from the Department of Computer Science, University of Ioannina in 2004 and 2007 respectively and since 2008 I am a Phd candidate at the same Department. I am an active researcher in computer vision and my research interests include image segmentation and registration, point set registration, feature extraction, pattern recognition and autonomous navigation. I am an IEEE Student Member and from October 2012 till June 2013 I was the interim Chair of the IEEE student branch of the University of Ioannina. Since my early years in the University, I was interested in entrepreneurship. I have setup two startups till now, one involved in video-on-demand advertisements and one with digital marketing solutions for mobile devices. During those years I have participated in various international and local business contests. In the past, I have worked as an instructor at the Department of Informatics and Telecommunication of the Technological Educational Institute of Epirus, Greece.