

# Automatic Landmarking for Non-cooperative 3D Face Recognition

Clement Creusot

PhD

The University of York  
Computer Science

December 2011

# Abstract

This thesis describes a new framework for 3D surface landmarking and evaluates its performance for feature localisation on human faces. This framework has two main parts that can be designed and optimised independently. The first one is a keypoint detection system that returns positions of interest for a given mesh surface by using a learnt dictionary of local shapes. The second one is a labelling system, using model fitting approaches that establish a one-to-one correspondence between the set of unlabelled input points and a learnt representation of the class of object to detect.

Our keypoint detection system returns local maxima over score maps that are generated from an arbitrarily large set of local shape descriptors. The distributions of these descriptors (scalars or histograms) are learnt for known landmark positions on a training dataset in order to generate a model. The similarity between the input descriptor value for a given vertex and a model shape is used as a descriptor-related score.

Our labelling system can make use of both hypergraph matching techniques and rigid registration techniques to reduce the ambiguity attached to unlabelled input keypoints for which a list of model landmark candidates have been seeded. The soft matching techniques use multi-attributed hyperedges to reduce ambiguity, while the registration techniques use scale-adapted rigid transformation computed from 3 or more points in order to obtain one-to-one correspondences.

Our final system achieves better or comparable (depending on the metric) results than the state-of-the-art while being more generic. It does not require pre-processing such as cropping, spike removal and hole filling and is more robust to occlusion of salient local regions, such as those near the nose tip and inner eye corners. It is also fully pose invariant and can be used with kinds of objects other than faces, provided that labelled training data is available.



# Contents

<b>1</b>	<b>Introduction</b>	<b>21</b>
1.1	Motivation . . . . .	21
1.2	The Shape Matching Board Analogy . . . . .	23
1.3	Applications . . . . .	25
1.4	Context . . . . .	26
1.5	Thesis Rationale . . . . .	27
1.6	Contributions . . . . .	30
1.7	Vocabulary Disambiguation . . . . .	33
1.7.1	Points Related Vocabulary . . . . .	33
1.7.2	Biometric Related Vocabulary . . . . .	34
1.7.3	Matching Related Vocabulary . . . . .	35
1.8	Thesis Structure . . . . .	37
<b>2</b>	<b>Literature Review/Background</b>	<b>41</b>
2.1	Face Analysis . . . . .	43
2.1.1	Brief History of Face Analysis . . . . .	43
2.1.2	Face Recognition . . . . .	45
2.1.3	Insights from Human Psychology Research . . . . .	71
2.2	Feature Localisation . . . . .	73
2.2.1	Feature Localisation on 3D Faces . . . . .	75
2.2.2	Feature Localisation on 3D Meshes . . . . .	82
2.3	Conclusion . . . . .	82
2.3.1	Input Data Problems . . . . .	83

2.3.2	General Problems . . . . .	84
2.3.3	Gap in the Research Literature . . . . .	86
<b>3</b>	<b>Detailed Strategy</b>	<b>89</b>
3.1	Preliminary Choices and Justifications . . . . .	89
3.2	Problem Statement . . . . .	93
3.3	Decomposition to Sub-problems of Varying Difficulty . . . . .	96
3.4	Datasets . . . . .	98
3.4.1	FRGC . . . . .	99
3.4.2	Bosphorus . . . . .	100
3.4.3	Data Preparation . . . . .	100
3.5	Metrics and Performance Evaluations . . . . .	103
3.5.1	Target Model . . . . .	103
3.5.2	Landmarks' Ground Truth . . . . .	104
3.5.3	Cost Functions . . . . .	107
3.6	Expected Limitations . . . . .	108
3.7	Conclusion . . . . .	108
<b>4</b>	<b>Automatic 3D Keypoints Detection</b>	<b>111</b>
4.1	Introduction . . . . .	112
4.2	Descriptor Maps . . . . .	115
4.2.1	Normals . . . . .	115
4.2.2	Neighbourhoods . . . . .	116
4.2.3	Curvature . . . . .	118
4.2.4	Simple Descriptors . . . . .	123
4.2.5	Histogram Descriptors . . . . .	125
4.3	Descriptor Matching . . . . .	128
4.3.1	Distributions of Local Shapes . . . . .	128
4.3.2	Descriptor-Landmark Score Maps . . . . .	130
4.3.3	Dealing with Histograms . . . . .	131
4.4	Local Maxima . . . . .	133
4.5	Objective Functions . . . . .	134

4.6	Method 1: Linear Combination of Score Maps . . . . .	135
4.6.1	Workflow . . . . .	136
4.6.2	Experiments . . . . .	138
4.6.3	Results . . . . .	143
4.6.4	Conclusion . . . . .	146
4.7	Method 2: Non-linear Combination of Score Maps . . . . .	147
4.7.1	Boosting Technique . . . . .	147
4.7.2	Number of Classifiers . . . . .	148
4.7.3	Computing Distributions Separation . . . . .	149
4.7.4	Matching Scores vs Raw Descriptor Values . . . . .	150
4.7.5	Comparisons with LDA Scoring . . . . .	152
4.8	Conclusion . . . . .	155
<b>5</b>	<b>Point Labelling using Structural Matching</b>	<b>159</b>
5.1	Introduction . . . . .	159
5.2	Graph and Hypergraph Representation . . . . .	162
5.2.1	Node Attributes . . . . .	164
5.2.2	Hyperedge Attributes . . . . .	165
5.3	Graph and Hypergraph Matching . . . . .	168
5.3.1	Inexact Hypergraph Matching . . . . .	170
5.3.2	Two Different Kind of Matching . . . . .	172
5.3.3	Seeding . . . . .	173
5.3.4	Are Multi-Attributes Worthwhile? . . . . .	175
5.3.5	Workflow . . . . .	181
5.3.6	Correspondence Stochasticity . . . . .	181
5.3.7	Limitation . . . . .	181
5.4	Postprocessing/ Correspondence Cleaning . . . . .	182
5.4.1	Hypergraph Matching as a Clustering Problem . . . . .	182
5.4.2	Rigid Registration by Unit-Quaternion Clustering . . . . .	183
5.4.3	RANSAC - Random Sample Consensus . . . . .	186
5.5	Hypergraph Matching by Relaxation . . . . .	187

5.5.1	Offline Training Process . . . . .	188
5.5.2	Graph Matching . . . . .	189
5.5.3	Results . . . . .	192
5.6	Global Matching Based on a Spectral Method . . . . .	196
5.6.1	<i>Duchenne</i> Tensor Matching . . . . .	196
5.6.2	Data Generation . . . . .	197
5.6.3	Effect of Missing Points . . . . .	200
5.6.4	Effect of Stepping . . . . .	200
5.7	Conclusion . . . . .	200
<b>6</b>	<b>Automatic 3D Face Landmarking</b>	<b>203</b>
6.1	Introduction . . . . .	204
6.2	Experimental Framework . . . . .	205
6.2.1	Workflow . . . . .	205
6.2.2	Model-fitting . . . . .	206
6.3	Results . . . . .	208
6.3.1	Computation Time Performance . . . . .	217
6.4	Conclusion . . . . .	218
<b>7</b>	<b>Conclusions</b>	<b>219</b>
7.1	A Global Picture in Chronological Order . . . . .	220
7.2	Summary of Contributions . . . . .	222
7.2.1	Keypoint Detection Using a Dictionary of Local Shapes . . . . .	222
7.2.2	Keypoint Labelling Using Learnt Structural Models . . . . .	223
7.2.3	Final Landmarking System . . . . .	224
7.3	Limitations and Future Research . . . . .	225
7.3.1	General Points . . . . .	225
7.3.2	Keypoint Detector . . . . .	228
7.3.3	Labelling . . . . .	229
7.3.4	Limitations at a Fundamental Level . . . . .	231
7.4	Final Conclusion . . . . .	231

<b>A</b>	<b>Local Shape Descriptor Distributions</b>	<b>233</b>
<b>B</b>	<b>Descriptor-Landmark Score Maps</b>	<b>239</b>
<b>C</b>	<b>Unsupervised Learning of a Landmark Model</b>	<b>245</b>
C.1	Automatic Saliency Discovery . . . . .	246
C.1.1	Local Shape Descriptors . . . . .	247
C.1.2	Likelihood of Being a Landmark Given a Local Descriptor . . . . .	247
C.1.3	The Saliency Metric . . . . .	248
C.1.4	The Ubiquity Metric . . . . .	249
C.1.5	Classification . . . . .	250
C.1.6	Selecting a Set of Landmarks . . . . .	250
C.1.7	Input Parameters . . . . .	251
C.1.8	Summary of Workflow . . . . .	252
C.2	Datasets . . . . .	252
C.2.1	BFM . . . . .	252
C.2.2	FRGC . . . . .	253
C.3	Results . . . . .	255
C.4	Conclusion and Future Work . . . . .	258
<b>D</b>	<b>Abandoned Ideas</b>	<b>261</b>



# List of Tables

2.1	Landmark-based face recognition results . . . . .	62
2.2	Result comparison . . . . .	69
2.3	Details of the tests E1, E2, E3 and E4. . . . .	71
2.4	Result Comparison . . . . .	71
3.1	Statistics per model of sex and coarse ethnic groups within the FRGC. . . . .	99
3.2	Statistics per identity of sex and coarse ethnic groups within the FRGC. . . . .	99
3.3	Statistics per model of coarse expression categories within the FRGC. . . . .	99
3.4	Categories within the Bosphorus database. . . . .	100
4.1	LDA weights for Configuration 1 . . . . .	142
4.2	LDA weights for Configuration 1 - Size only . . . . .	143
4.3	LDA weights for Configuration 1 - Descriptors only . . . . .	143
5.1	Global and local transformation statistics . . . . .	185
5.2	Variables definition . . . . .	193
5.3	Results per landmark . . . . .	194
5.4	Results per landmark (0-13) on subsets of the Bosphorus database. The abbreviations used in the first two columns have been defined in Section 3.4.2. . . . .	194
6.1	Results comparisons on the FRGC dataset . . . . .	211
6.2	Results of landmark localisation on the Bosphorus dataset. . . . .	214
7.1	Comparisons between expert systems and our machine learning approach. . . . .	225
B.1	Landmark score maps. . . . .	244





# List of Figures

1.1	The matching shape board puzzle analogy . . . . .	24
1.2	Extracting face symbolic representation . . . . .	24
1.3	Problem Breakdown . . . . .	29
1.4	Example of keypoint detection. . . . .	29
1.5	Keypoint vs. Landmarks . . . . .	34
1.6	Block diagram of the thesis structure. . . . .	37
2.1	Proportion of the face and eyes. Sketches by L. Da Vinci. . . . .	43
2.2	Juror questionnaire . . . . .	45
2.3	Configural measures on the face . . . . .	46
2.4	Workflow of face recognition systems on-line part. . . . .	47
2.5	Face recognition taxonomy . . . . .	52
2.6	PCA vs. FLD . . . . .	53
2.7	PCA vs. NMF . . . . .	54
2.8	Kernel Discriminant Analysis (KDA) . . . . .	55
2.9	Comparing PCA, LDA, KPCA and KDA . . . . .	55
2.10	Pseudo-2D Hidden Markov Model . . . . .	56
2.11	Canonical transformation of the face . . . . .	57
2.12	Nearest vs. Normal search . . . . .	59
2.13	Elastic Bunch Graph Matching . . . . .	61
2.14	Point signature . . . . .	61
2.15	Iso-geodesic surfaces . . . . .	63
2.16	Keypoint repeatability . . . . .	64

2.17	Face automatic drawing . . . . .	65
2.18	Face recognition by region . . . . .	66
2.19	Face recognition by region results . . . . .	67
2.20	Curvature based features . . . . .	67
2.21	Example of internal and external features. From [Ellis et al., 1979]. . . . .	72
2.22	Featural vs. Configural Face recognition . . . . .	73
2.23	Caricature effect . . . . .	74
2.24	Hand-placed landmarks and semi-landmarks . . . . .	77
2.25	Profile-based facial feature detection . . . . .	77
2.26	Face histogram descriptor . . . . .	79
2.27	Ridge line detection . . . . .	80
2.28	Taxonomy of landmark candidate detection systems . . . . .	83
2.29	Neutralising expression through morphing . . . . .	86
3.1	Example of localisations . . . . .	90
3.2	Limitations of 2D face systems . . . . .	91
3.3	2D main limitations . . . . .	91
3.4	Visualisation of the problem . . . . .	94
3.5	Problem Breakdown . . . . .	94
3.6	Keypoint detection example . . . . .	95
3.7	Labelling Problems . . . . .	97
3.8	Sub-problem evaluations . . . . .	98
3.9	Sample of meshes from the FRGC dataset. . . . .	101
3.10	Sample of meshes from the Bosphorus dataset. . . . .	101
3.11	Mesh generation from depth-maps. . . . .	103
3.12	Plain mesh (Left). 2D texture mapping (Center). 2D contour mapping (Right). . . . .	103
3.13	Position of the 14 landmarks . . . . .	104
3.14	Different chin shapes . . . . .	104
3.15	Exocanthion vs. lateral orbit . . . . .	105
3.16	Landmark Positioning Refinement . . . . .	106
3.17	Summary of the landmarking framework. . . . .	109

4.1	Example of keypoint detection.	113
4.2	Related Work Diagram	114
4.3	Example of neighbourhood	116
4.4	2D Curvature	119
4.5	Example of descriptor maps	122
4.6	Example of local volume computation	124
4.7	Principal Curvature Descriptor Maps	125
4.8	Example of Descriptor maps	126
4.9	Example of spin image histograms.	127
4.10	Examples of distribution	130
4.11	Descriptor-Landmark score maps examples	132
4.12	Method 1 workflow	136
4.13	Method 1 workflow	137
4.14	Position of the 14 landmarks	137
4.15	Visualisation of the Score Hypercube	139
4.16	Example of class generation for the LDA	140
4.17	Example of normalised landmark score maps.	140
4.18	Examples of extracted keypoints on faces from the FRGC v2 data set.	144
4.19	Landmark retrieval results	145
4.20	Landmark retrieval numbers	145
4.21	Keypoint repeatability results	145
4.22	Number of classifiers	149
4.23	Adaboost classification - Score vs Descriptor values	151
4.24	Distributions intersection - Adaboost on scores vs Adaboost on raw values	152
4.25	LDA vs Adaboost Scoring	153
4.26	Distributions intersection - LDA vs Adaboost	154
4.27	Method 2 results	154
4.28	Landmark retrieval results	154
4.29	LDA vs. Adaboost: landmark retrieval error rates at fixed radius.	155
4.30	Multiface keypoint detection	157

5.1	Model of the labels to be retrieved. . . . .	160
5.2	Problem to solve . . . . .	160
5.3	Examples of hypergraph representation . . . . .	164
5.4	Example of direct neighbourhoods . . . . .	164
5.5	Example of graph with non-complete connectivity. . . . .	167
5.6	Random-angle triangle space representation . . . . .	168
5.7	Examples of Exact Graph Matching Problems . . . . .	169
5.8	Hypergraph matching representation . . . . .	170
5.9	Graph matching example . . . . .	171
5.10	Score computation . . . . .	175
5.11	Score combination . . . . .	176
5.12	Probability Density Functions for node properties. . . . .	177
5.13	Probability Density Functions for edge properties. . . . .	178
5.14	Example of matching probabilities for 2-edges using different properties . . . .	179
5.15	Example of matching probabilities for 3-hyperedges using different properties	180
5.16	Correspondence Workflow . . . . .	181
5.17	Hypergraph matching by correspondence clustering . . . . .	183
5.18	Unit Quaternion Clustering . . . . .	184
5.19	Steps for candidate elimination. . . . .	191
5.20	Limitation of the dual hyperedge relaxation system. . . . .	192
5.21	Limitations linked to permutations. . . . .	192
5.22	Examples of results on the Bosphorus dataset. . . . .	194
5.23	Example of synthetic set of points for hypergraph matching evaluations. . . .	199
5.24	Effect of missing data in the query on the matching results. . . . .	201
5.25	Effect of stepping with extra and missing nodes. . . . .	202
6.1	Workflow of the landmarking system. . . . .	206
6.2	Landmark retrieval rate for the 14 landmarks on the FRGC test set. . . . .	209
6.3	Examples of landmarking in cases with missing noses. . . . .	210
6.4	Examples of localisations on rotated meshes. . . . .	210
6.5	Examples of landmarks localisation on large input meshes. . . . .	212

6.6	Distance error for the 14 landmarks on the FRGC test set. . . . .	213
6.7	Examples of landmark localisation on the Bosphorus dataset. . . . .	215
6.8	Four worst cases in the FRGC test set. . . . .	216
6.9	Distribution of global registration errors on the FRGC test set. . . . .	216
6.10	Examples of discrete failures on the Bosphorus dataset. . . . .	217
6.11	Distribution of global registration errors on the Bosphorus test set. . . . .	217
7.1	Difficult case from the Bosphorus dataset. . . . .	227
7.2	Multiview 2D depth map. . . . .	230
C.1	Examples of landmark model in the literature . . . . .	246
C.2	FRGC and BFM models . . . . .	253
C.3	Saliency and Ubiquity map construction. . . . .	254
C.4	Saliency map at different locality. . . . .	255
C.5	Manual vs. Automatic: landmarks position . . . . .	256
C.6	Manual vs. Automatic: map differences . . . . .	257
C.7	Automatic detection on ubiquity map . . . . .	257
D.1	Examples of local extrema. . . . .	262
D.2	Keypoints as point-cluster centroids. . . . .	263
D.3	Ridgelines on faces. . . . .	264
D.4	Example of contour lines drawn on the $k_1$ curvature at level $C_0 = 0.0$ . . . . .	264
D.5	Graph construction workflow. . . . .	264
D.6	Example of hyperedges of undetermined degree . . . . .	265
D.7	Face recognition using Thin Plate Spline (TPS) . . . . .	266
D.8	Local Symmetry Descriptor. . . . .	267
D.9	Generation of depth-map contours. . . . .	268
D.10	FRGC dataset error. . . . .	268
D.11	Example of depth-map contours. . . . .	269
D.12	Multiview 2D depth map. . . . .	270



# List of Algorithms

1	Boosting score computation. . . . .	148
2	Pseudo-code for the elimination procedure called at each iteration of the graph matcher. . . . .	193
3	Tensor Power Iteration (Tensor Notation) . . . . .	197
4	Power Iteration for our Sparse structure of candidates in the case of degree 3 hyperedges. . . . .	197
5	Power Iteration for our Sparse structure of candidates for degree 3 (ordered) and 2 (symmetric). . . . .	198

# Acknowledgments

I would like to sincerely thank and to express my deepest gratitude to all the people that have made this PhD possible and have helped me during the three years and three months of this project.

My first thanks go to my two supervisors Prof. Jim Austin and Dr. Nick Pears. I would like to thank Jim for hiring me in the first place through his company Cybula Ltd. which allowed me to start my PhD. I first met Jim in the most ideal and relaxed setting possible for an interview: on a sunny day in a mill, next to a swimming pool surrounded by vineyards in the south of France. Since I began my PhD, Jim has always been very supportive of the new research ideas I proposed and he has always given me good advice about the management of my project. His expertise on pattern recognition and graph matching, as well as his support as the head of the Advance Computer Architecture group has really contributed to the success of my research throughout the PhD.

I would also like to thank Nick, who I first met as my PhD assessor at the university, and who soon become my closest supervisor. His expertise in computer vision and feature localisation has been extremely valuable as my PhD goals slowly drifted from face recognition to 3D shape landmarking. Nick has always been very involved in discussing new ideas, suggesting literature and helping with various aspect of the PhD. Nick encouraged me to write and publish and was a very active and constructive co-author and reviewer. His feedback has always been thorough and helpful, teaching me to better understand the audience of the paper, as well as the expectations and standards in the field.

I would like to thank all the people from the European Virtual Anthropology Network (EVAN) for their help. My research project has been partly supported by the European Union FP6 Marie Curie Actions MRTN-CT-2005-019564 "EVAN". I would like to thank them in particular for giving me the opportunity, as a fellow, to meet interesting people



in the field of anthropology, for showing me how 3D shape analysis tools can help make scientific discoveries, and for their excellent research training sessions around Europe.

Finally, I would like to thank my family and friends for their support during my PhD:

- My parents, for supporting me during my – rather long – studies.
- My friends in York, for keeping me entertained (and sane) during these three years, for showing me the best of British life, and for giving me too many reasons to miss York and the UK when I leave.
- My old and long-term friends, now spread all over Europe, for their long phone calls and for the much needed holiday breaks we shared.

# Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree other than Doctor of Philosophy of the University of York. This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by explicit references.

I hereby give consent for my thesis, if accepted, to be made available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .....(candidate)

Date .....

# Chapter 1

## Introduction



---

Where it all starts: Genesis of a human face between the second and third month of pregnancy.

BBC 1 Documentary *Inside the Human Body: Creation* - 5 May 2011.

Computer-Generated Imaging (CGI) based on live ultrasound scans.

### 1.1 Motivation

Sensing the world is one of the most useful abilities developed in the natural world. All the senses can be used to acquire information from the world and observations can in turn be used to act upon the world. Through our senses the outside world is acting upon us, modifying our state, and triggering reactions. When trying to reproduce these *interactions* in the relationships between machines and the rest of the world (including us) it appears that the most difficult part is not so much to make machines produce actions but to make them observe and understand the environment.

Some senses are more easy than others to interpret. The senses of taste and smell are relatively complex to reproduce in terms of chemistry but the interpretation of results is easy: the same proportion of basic chemical compounds corresponds to the same taste/smell. There is no computational burden here and the complexity is mainly engineering. Other senses are simple to reproduce because of their very low dimensional input, for example the sense of balance and movement (the vestibular system in the inner ear) provides information about orientation relative to the gravitational field and information about movement using inertia. These can easily be reproduced in machines using gyroscopes, and here again, interpreting the input is relatively straightforward.

Sight, hearing and touch are usually the most difficult senses from a computational point of view. Hearing stands a little aside from the other two as it is sequential in time. Sight and touch are both similar in the way that they allow description of the geometry of the world. The difference between the two is smaller than believed. Indeed, blind and non-blind people can “see” through touch (using the hands [Kilgour and Lederman, 2002] or even the tongue [Callaghan and Mahony, 2010]), and vision systems can “touch” objects through light (3D reconstruction using passive or active vision). Finding ways that machines can use these senses is a real challenge. Nature, however inspirational it might be, cannot always be copied as it is. We have no evidence that animal vision is reproducible with current computer architectures and available technologies. To allow a wide range of applications, solutions have to be found to enable machines to perform specific vision tasks with existing hardware.

In this thesis, we focus on one vision task in particular: face processing, for which humans have evolved a specific region of the brain (namely the fusiform gyrus [McKone et al., 2006]). This specialisation of the human brain to see faces is remarkable, allowing us humans to facialise almost anything, from shapes in clouds to Mr Potato Head or simple smileys. Our aim in this thesis is relatively modest as we simply want to be able to “see” realistic human faces, i.e. to make sense of a sensorial input as being a face, composed of facial features in a particular layout.

## 1.2 The Shape Matching Board Analogy

A very well-known pre-school toy is the shape matching board puzzle in which objects have to be placed in holes of corresponding shape (see Figure 1.1). In a thought experiment, we can easily imagine three different participants for this game:

- Level 1: a child too young to match the shapes visually: his strategy would be to take one object and try to force it into different holes until he succeeds.
- Level 2: a child old enough to master shapes but not old enough to speak: his strategy would be to match the shape of the hole and object mentally to determine the correspondences.
- Level 3: an older individual mastering language (e.g. an adult): his strategy would be to analyse the shape of the hole, associate it with a semantic category (e.g. “it’s a star shape”) and look for objects having the same semantic association.

In the last case, the user is no longer matching the shape but the semantic labels of those shapes. Of course, if several star-shaped objects are present, the user of level 3 will fall back to the mental geometric comparison of level 2 and, if the differences are not easily identifiable, he will fall back to the trial-and-error approach of level 1.

These three strategies correspond to different styles of algorithm.

- The level 1 approach corresponds to an exhaustive or random search.
- The level 2 approach corresponds to a matching problem in a geometric space.
- The level 3 approach corresponds to a matching problem in a semantic space.

In order to reproduce level 2 and 3, a machine has to be able to extract symbolic representation from the input data. This implies the ability to discard all unhelpful information while retaining enough information to solve the higher level problem of matching.

In this thesis we focus on problems at level 2. The main contribution of this thesis is to enable a symbolic geometric representation of a face to be extracted from an input mesh (see Figure 1.2). The representation takes the form of a set of landmarks associated with known facial features.



Figure 1.1: Matching shape board puzzle as an analogy for symbolic geometric representation extraction.

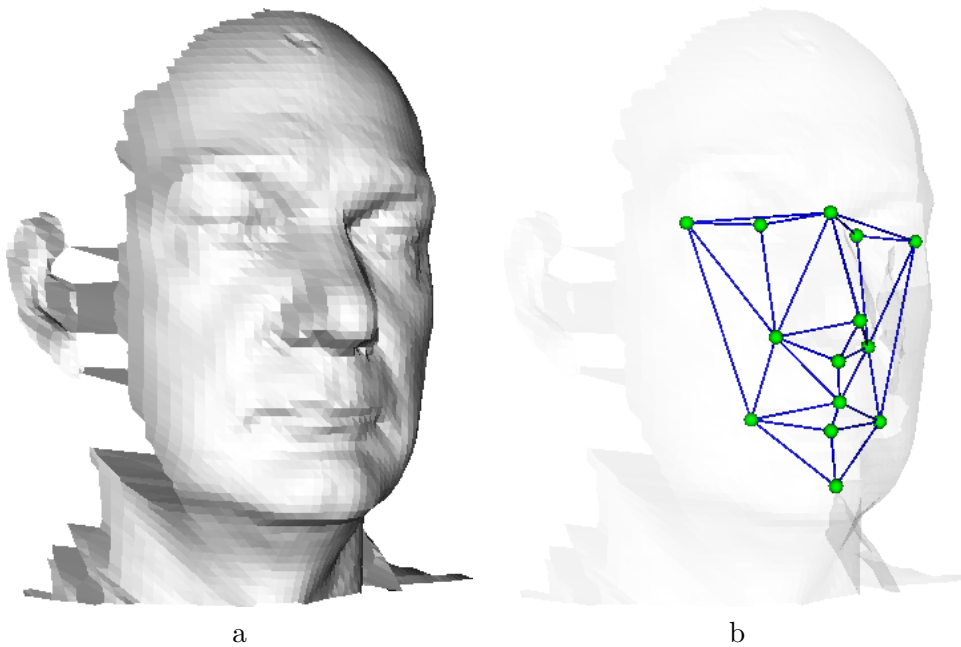


Figure 1.2: (a) Input from the real world. (b) Extracted symbolic representation of the face (as a set of facial landmarks).

## 1.3 Applications

Faces play a crucial role in our everyday life. They are our primary modality for recognising people and they also provide information about attention and emotional states. Enabling machines to see faces has a huge range of possible applications in industry.

**The face as a biometric data** The main application for automatic face processing remains surveillance and access control. The face is a very accessible biometric modality that can be used either for identity checking or identity recognition. Unlike fingerprints, it can be captured at a distance, and unlike ears and irises, the face is not usually occluded involuntarily. The voluntary hiding of one's face usually raises suspicions. Airport border control police agents can be asked to watch CCTV in order to search for people on a watch-list. This is a very repetitive task at which human are not really reliable.

The main applications of face processing, the final objective of which is to replace human agents, are:

- Face verification: This is usually used to check whether you are who you claim to be, or whether you have the credentials to access a given area.
- Face recognition: This is used to retrieve the identities of people present from databases (watch-list, passenger list, police records, and so on).

**Enabling new Human-machine interactions** An increasingly interesting set of applications for face processing is emerging because of recent advances in computer vision, robotics and mobile device technology. Machines are no longer confined to factory plant and are more and more present in our lives. Here is a shortlist of interactions with machines requiring face understanding:

- Face recognition for interactions: If you ask a robot to fetch you something, the robot should be able to come back to you. If you have moved or if several people are present in the room the robot should identify where you are and, for that, it needs to recognise you. Face recognition is therefore a natural part of robotic systems that interact with humans.

- Emotional adaptation: Robots that will interact with humans will need some social skills. To be able to adapt to the interlocutor’s emotional state is essential to make the exchanges as natural as possible. This can be done through a combination of facial expression recognition and voice recognition.
- Avatar control: Facial gesture is something one wants to reproduce in virtual worlds. Whether it is to animate a 3D character in a movie, a player’s avatar in a game, or a remote tele-presence impersonation of you, emotions and lip movements have to be captured and understood by the machine.
- Driver drowsiness monitoring: Anybody is able to monitor a car driver’s face for signs of drowsiness but realistically nobody would pay someone to do that. A computer vision-based understanding of the face is the only current cost-effective way of allowing such applications to occur.

## 1.4 Context

This thesis is concerned with different 3D face analysis techniques for facial feature localisation. At the beginning of this project we noticed that, unlike 2D vision techniques that take advantage of machine learning methods, 3D vision uses more heuristic approaches that strongly limit the number of cases in which they can be used. In a preliminary literature review we observed that, while 3D face recognition claims to be more robust (by design) to pose variation, it is in fact as bad as 2D in dealing with those changes. This was not due to data representation issues or recognition technique problems, but simply due to the pre-processing: the initial correspondences between faces could not be retrieved when large pose variations or large occlusions are present. For example, such systems will try to detect the tip of the nose as the most concave or the most protruding point in the input, which of course is not true in the general case. Both continuous (registration) and discrete (feature localisation) classes of current correspondence techniques cannot cope with these changes. Therefore, we decided to investigate feature localisation on 3D faces and came up with machine learning approaches to replace the existing expert-system-based “recipe-like” methods. As a machine learning approach is used, all dependencies linked to the application domain



vanished, leaving us with a generic framework for the landmarking of 3D surfaces. Evaluation on 3D face databases showed that our framework outperforms the state-of-the-art 3D face landmarking system in terms of robustness and, in some cases, in terms of precision, while making far fewer assumptions about the input data.

## 1.5 Thesis Rationale

The research area that we are interested in is 3D face processing. In particular, we focus on machine-learning-based 3D face keypoint detection and landmarking.

**Starting Point** Our project started with the idea that 3D face recognition can be greatly improved if the landmarking of facial features was more robust in dealing with the variations often observed with non-cooperative subjects: mainly changes in pose and occlusions. We argue that the existing techniques for landmarking, while becoming more and more complex over the years, will never succeed in reaching the given level of generality required when the system cannot predict what the input will contain. Our main criticism of these techniques is that the detection rules are always enforced by the researcher instead of being learnt by the system. This “expert system” characteristic of the existing approaches makes them vulnerable when new inputs are presented to the system. Indeed, the rules extracted by the researchers are usually naive and unrealistic as they relate to simple correlations observed between the targeted landmarks and extremal field values of a limited number of 3D descriptors.

To alleviate these limitations, we focused our efforts on finding practical ways to replace those expert-system approaches with machine learning techniques while keeping our framework as versatile as possible.

**Ideal Goal** Our objective was to improve the current state-of-the-art in terms of feature localisation and registration on non-cooperative 3D faces containing large variation in pose and occlusions. Ideally, a perfect system would be able to extract faces from any input, as long as some features are visible. This should be robust whatever action is performed by the non-cooperative subject: talking, talking into a cell phone, eating while walking,

coughing, blowing his nose, laughing, and so on. Moreover, in order to be usable in real world applications, the system should be fast so that near real-time response is achievable.

**Strategy overview** Our approach throughout the project has been to break down problems into smaller pieces. The landmarking problem, the problem of finding positions associated with known labels, is cut down into smaller sub-problems (as seen in Figure 1.3):

- Detect possible landmark candidate positions (keypoints) on the mesh.
- Select one candidate position for each targeted label resulting in a set of labels.
- (Optional) Refine the results by moving the obtained landmarks around to optimal local positions.

In turn, detecting the initial point of interest position can be split into two sub-problems (see Figure 1.4):

- Compute score maps over the input mesh representing the likelihood of a vertex being a shape of interest.
- Select maxima on these maps as keypoints.

Constructing a set of landmark from the keypoints can be split in three sub-problems:

- Associate each position with a list of candidate labels using local descriptors (seeding of the label correspondences).
- Eliminate candidates by running correspondence filtering methods such as graph matching and registration techniques.
- Select a doubly semi-stochastic (i.e. one-to-one) matching from the set of correspondences.

In our framework, machine learning techniques are used to detect points of interest on the mesh. To reduce the set of correspondences and to select one-to-one associations between the detected points and the learnt model, different matching techniques have been investigated, including hypergraph and tensor matching techniques as well as clustering and global registration approaches.

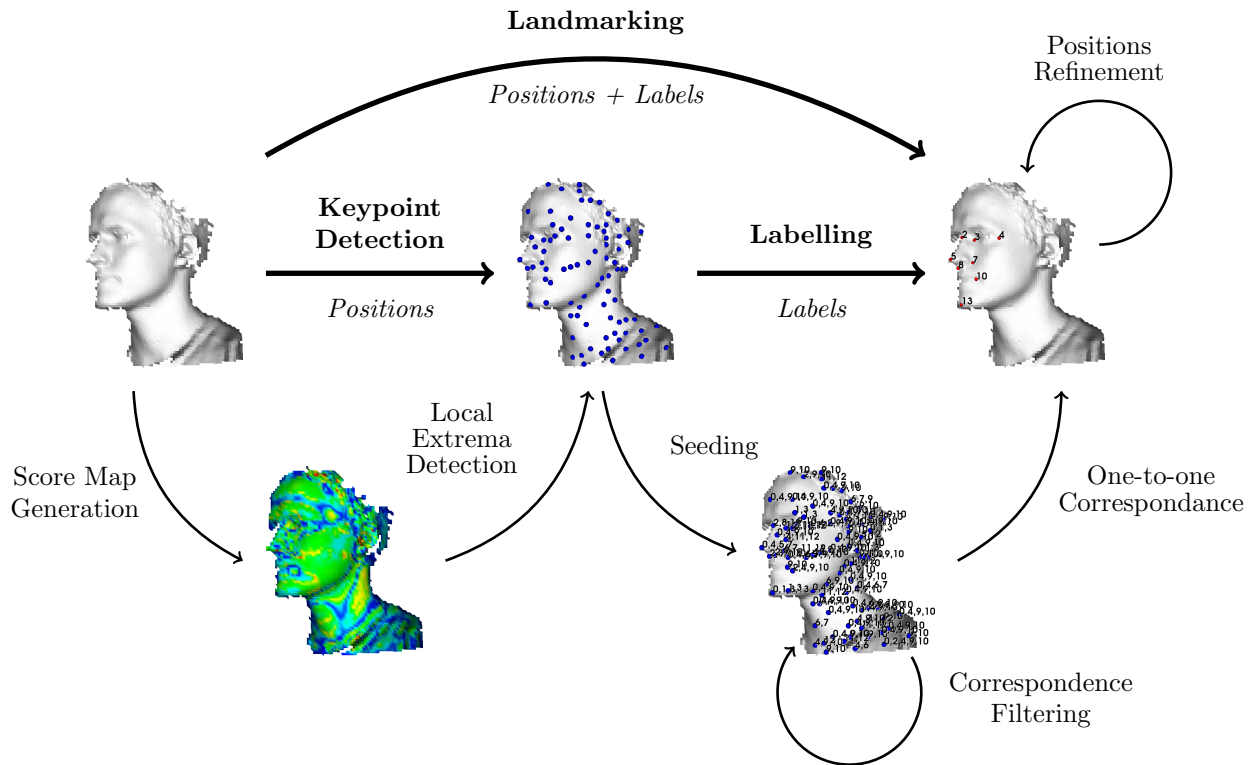


Figure 1.3: Problem Breakdown: the landmarking problem is split into two sub-problems that are solved independently: keypoint detection and point labelling.

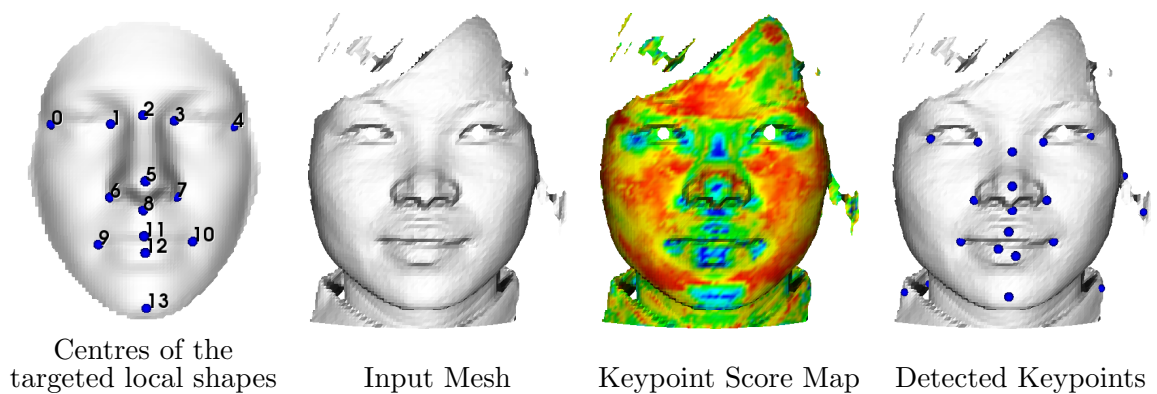


Figure 1.4: Example of keypoint detection using our method on a 3D scan from the FRGC database.

**Result overview** For landmarking applications, our system outperforms the state-of-the-art methods on the two databases tested (the FRGC and the Bosphorus). Our technique is, by design, more robust in dealing with imperfect input data and does not expect every single targeted landmark to be present in the input. It does not require pre-processing such as cropping, spike removal and hole filling and is also more robust in dealing with occlusion of salient regions, such as the nose and eyes.

Our keypoint detector can learn any shape of interest from a labelled training database and look at patterns using any number of scalar and histogram descriptors. The cues from the different descriptors are merged in final score maps in a generic way, easily transferable to other kinds of application, as long as training is available.

## 1.6 Contributions

The main contributions of our project, as discussed in this thesis, can be grouped in several categories.

1. Contributions in the approach and the way that we looked at the problem.
2. Practical contributions to the field in terms of novel methods, evaluations of hypotheses and results.
3. Contributions to the community in providing tools and data to quicken advances of other research in this particular domain.
4. Minor contributions to the community by disclosing our unsuccessful experimentations and abandoned paths, which can possibly save the time of new research students in this area.

### Contributions in the approach

1. To replace existing expert systems for 3D face feature localisation with machine learning techniques, such that the rules used for detection are learnt and not enforced by the designer of the system. This also allows the detection of less salient features for which humans struggle to create rules.

2. To relax some of the assumptions about the input data to enable non-cooperative face recognition. Most detectors use a global approach, trying to find extremal points in the input. These expert-designed patterns might be true for the face region but do not apply to all the unknown non-face parts of the input data.
3. To increase the number of landmarks to be searched in order to minimise the risk of task failure due to occlusions and spurious data in non-cooperative cases.
4. To increase the number of descriptors to be used conjointly to detect or label points.
5. To move away from the assumption that the best descriptors for detection are also the best descriptors for labelling.
6. To investigate the use of hypergraph matchers for the correspondence finding on 3D objects.

### **Practical contributions**

1. A new method for keypoint detection on meshes using a dictionary of learnt local shapes.
2. An evaluation of its performance in different conditions.
3. An extension of this method to a non-linear boosting classifier.
4. A study of the behaviour of 10 widely used 3D descriptors over a range of scales at 14 landmark positions.
5. A new method for labelling, using a hypergraph matching technique of relaxation by elimination.
6. A evaluation of its performance for landmark label retrieval with unit-quaternion clustering disambiguation.
7. A comparison of different correspondence techniques (RANSAC vs. tensor matching) in cases presenting missing data.
8. A new framework for landmarking by labelling automatically detected keypoints.

9. An evaluation of its performance on the FRGC and Bosphorus datasets on which our method shows better results than state-of-the-art methods.

**Contributions through data and tools** Building tools is a necessary part of any research project. To facilitate results reproducibility and save other researchers' time, tools and database related data have been published on the author's webpage<sup>1</sup> throughout this PhD.

Example of tools are:

- Converters between different 3D data structures and file formats: `.abs`(structured point cloud, FRGC), `.bnt` (structured point cloud, Bosphorus), `.obj` (mesh), `.ply` (mesh), `.off` (mesh), `.stl`(mesh), `.pcd` (structured point cloud), `.png` (2D depth map)
- Readers and writers for our landmark data files.
- Local 3D descriptors computation: principal curvature, functions of the principal curvature (H,K,SI, and so on), volume descriptor, histograms descriptor (Spin Images, spherical images), and so on.
- a 3D viewer for all data files used in the project (meshes, landmarks, curves, score maps, texture, hypergraphs and so on).

Examples of data provided are:

- Global registration transformation matrices (computed using ICP) for the FRGC database (with a video for visual checking).
- Landmarks obtained using our system on the FRGC and Bosphorus database.
- Ground-truth landmarks used for results computation.
- Mean models of the face computed using mean 2D depth map of registered faces (mean white male, mean white female, mean Asian male, mean Asian female, all).

---

<sup>1</sup><http://www.cs.york.ac.uk/~creusot>

**Minor contributions through failures and discarded ideas** If research is seen as a landscape, warning researchers about the dead-ends and the treacherous paths is as crucial as charting the walkable roads. As a rule, there are always more ideas that don't work than ideas that work. In Appendix D of the thesis, abandoned ideas are presented. At no point did we formally prove that those techniques are not worth studying longer than we did, but hints were found that they might be time-consuming, full of pitfalls and too uncertain to be considered in the allotted time for this project.

## 1.7 Vocabulary Disambiguation

An essential part of the thesis is to make sure the author and the reader share the same vocabulary and notions. In addition to the definitions provided in each chapters, notions used throughout the thesis are briefly described in this section.

### 1.7.1 Points Related Vocabulary

This thesis mainly focuses on the detection of point (zero-dimensional) features over 3D surface meshes, the position of which is represented as a tuple  $(x, y, z)$ . Depending on what is attached to it, the point will change in nature. Figure 1.5 shows example of keypoints, landmarks and landmark candidates.

**Keypoint** A keypoint is synonymous with *point of interest*. It is an *unlabelled* point that is “special/uncommon” with regards to criteria that vary depending on the context in which the keypoints are used. A recurring pattern of keypoints will be that they are relatively sparse compared to the input data and that they are repeatable between different instances (captures) of the same object. A keypoint doesn't have any label or semantic category attached to it.

**Landmark** A landmark is a point associated with a label and can be represented as a tuple (point, label). Different landmarks on different objects sharing the same label will be corresponding landmarks. Only one-to-one correspondences are allowed between objects. Therefore, a landmark will be unique within an instance of an object. Two different landmarks on the same capture cannot have the same label.

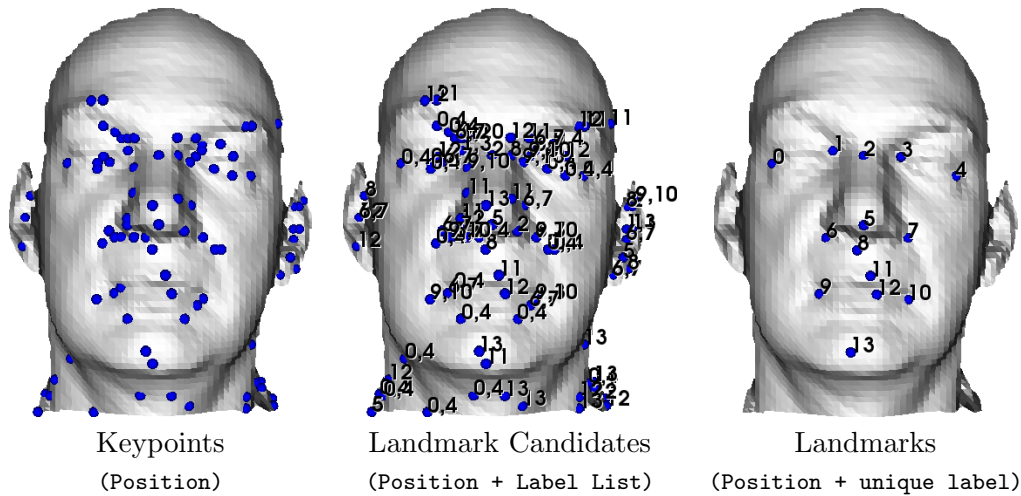


Figure 1.5: Example of visual representation of keypoints, landmark candidates and landmarks.

**Landmark candidate** A landmark candidate is a point for which no label has yet been determined. It can be represented as a tuple (point, list of possible labels). In our framework, a keypoint is seeded with possible landmark labels and becomes a landmark candidate. When one-to-one correspondences have been found between the query object and a model, the landmark candidates that have been selected are upgraded to landmarks.

### 1.7.2 Biometric Related Vocabulary

**Face verification** A face verification system is a process taking two faces in input and returning a true value if the two faces are considered to be from the same individual.

**Face recognition** A face recognition system is a process taking a unknown face and a database in input and returning (if available) the identity from the database that best corresponds to the query face.

**Equal Error Rate** When evaluating face verification systems, the percentage of false negatives and false positives are inversely related. The fewer false negatives you obtain, the more likely a false positive will be. The weights given to these two cost functions depend on the application. A border control face verification systems will prefer to limit false positives and allow a larger number of false negatives where a human operator can



intervene. Less sensitive applications where the risk associated with false positives is lower and where having a human operator involved is not cost-effective will prefer the opposite (e.g. biometric pocket money for a primary school restaurant). In order to evaluate verification systems independently from the business domain, a standard metric is used. It is the Equal Error Rate (EER) that corresponds to the rate at which the False Acceptance Rate (false positives) and the False Rejection Rate (false negatives) are equal.

**Rank 1 recognition rate** When evaluating face recognition systems, an obvious measure of the overall performance is to look at the percentage of cases in which the identity returned by the system is correct. This is called the rank-1 recognition rate. Systems that can output a ordered list of possible matches can also be evaluated using the rank-N recognition rate, representing the percentage of cases in which the correct identity is among the N first matches returned by the system.

### 1.7.3 Matching Related Vocabulary

**Graph** A graph is composed of a set of nodes and a set of edges connecting pairs of nodes. Only non-oriented graphs are discussed in this thesis, where the edge connections are not ordered.

**Hypergraph** An hypergraph is an extension of the concept of a graph in which the connectivity is no longer bounded by two. Hyperedges can connect any number of nodes. The hypergraph is said to be uniform if all the hyperedges have the same degree of connectivity. For example, a graph can be represented by a 2-uniform hypergraph and *vice versa*. Both oriented and non-oriented hypergraphs are used in this thesis.

**Feature** The word feature can mean a variety of different things depending on the context. In this thesis it usually refers to local artifacts that have been detected (points, curves or other). When dealing with faces, we will use the term “facial features” to describe the parts of the face that have a name (e.g. nose, eye, mouth). When dealing with on-manifold space reduction and classification techniques, the term features will refer to the basis axis of the

manifold in which the data points are represented. The number of features will refer to the number of dimensions used to describe a particular pattern in that space.

**Featural information** When matching two data structures, we will use the term featural information to describe any value attached to a local neighbourhood around the local feature (points, curves). The notion of locality can vary depending on the application and is usually a parameter of the system.

**Configural information** The configural information relates to any value describing the relationship between two or more ordered or non-ordered local features of the data structure to be matched. It can be, for example, the Euclidean or geodesic distance between two points (degree 2), the minimal distance between two curves (degree 2), the area of the triangle defined by three points (degree 3), the ratio between the length of two segments (degree 4) and so on.

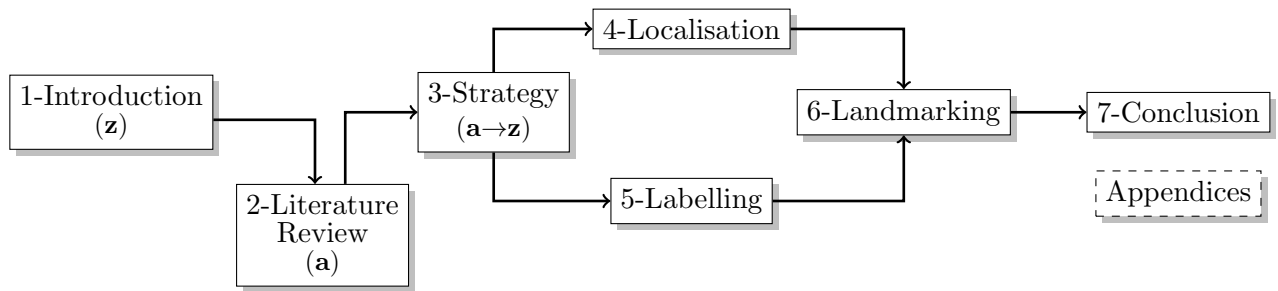


Figure 1.6: Block diagram of the thesis structure.

## 1.8 Thesis Structure

In this section, the global organisation of the thesis is presented. Each chapter is summarised in a few paragraphs to help the reader construct a clear image of the whole project. Figure 1.6 shows the structure of the thesis. Most of the work presented here is also featured in the following papers:

- **3D Face Landmark Labelling**

Clement Creusot, Nick Pears, and Jim Austin

In Proceedings of the ACM workshop on 3D object retrieval, 3DOR 10, pages 27 - 32, Firenze, Italy. doi:10.1145/1877808.1877815.

- **Automatic Keypoint Detection on 3D Faces Using a Dictionary of Local Shapes**

Clement Creusot, Nick Pears, and Jim Austin

International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011

pages 204 - 211, Hangzhou, China. doi:10.1109/3DIMPVT.2011.33.3DIMPVT

- **A Machine-learning Approach to Keypoint Detection and Landmarking on 3D Meshes**

Clement Creusot, Nick Pears, and Jim Austin

Under Review - Submitted 2011-10-15 - IJCV Special Issue on 3D Imaging, Processing and Modeling Techniques

To help the reader glance through this thesis in the future, small bullet-point summaries are dispatched in each chapter. They appear in small blue-highlighted framed boxes.

## Chapter 2 - Literature Review

In Chapter 2, we describe the state-of-the-art of research in different areas related to the analysis of human faces. This allows us to outline the gap in research that we aimed to fill. The first part is dedicated to the broad field of face analysis, with a description of the early history of the field, a deep review and taxonomy of face recognition methods and an overview of other face analysis techniques. As the brain is always a good source of inspiration, we review some of the work performed on human vision as studied in psychology and discuss the implication of this work for computer vision systems.

The third part of this chapter is dedicated to the problem of feature detection on faces. Examples of 3D techniques are reviewed and the main limitations of the current methods are outlined.

To conclude, we summarise the remaining unsolved problems in the field and describe different ideas to fill those gaps.

## Chapter 3 - Detailed Strategy

While the literature review has allowed us to outline the gap in research, it is still necessary to plan how to approach the problem, how to break it down into manageable parts and how to evaluate our success in solving the problem. This chapter is about our high level strategy and how we aim to get closer to our ideal objective. As a first step, we formally describe the main problems that we want to solve. Then, we explain the choices and reasonable assumptions that need to be made in order for the problem to be solvable. Priorities between competing objectives are given and explained. The third part explains how success will be measured for each of the different problems. The fourth part describes the databases that have been used to run our experiments. Details about the content of the database and the way we managed them are given. In the final part, the expected limitations of our work are detailed.

## Chapter 4 - Automatic Keypoint Detection

In our approach to the feature correspondence problem we assume that some points are repeatable over most captures of human faces. In order to find those keypoints (unlabelled candidates) classic techniques consist of selecting local extrema over the map of one descriptor

(e.g. the maximum or minimum of Gaussian curvature). Here, a new technique is described which is able to deal with several descriptor maps and several kind of distributions over those maps. The aim of this detector is to be able to manage points that are potentially repeatable but whose distributions over given descriptors are not extremal.

In the first part, a description of the framework is given. In the second part, the way descriptors are computed in our experiments is explained. We then describe how the statistical distributions of those scores are learnt over a input dictionary of shapes of interest. For each learnt shape, the different descriptors are merged using linear (LDA) or non-linear (Adaboost) techniques. The next part explains how the matching maps computed for each shape are merged together to form one unique final map. Details about the local extrema detector over this map is given, before the experiments are described. After giving the results obtained with this technique, we discuss some of the remaining issues and detail some other paths of research that we have tried concerning keypoint detection.

## **Chapter 5 - Point Labelling Using Structural Matching**

In this chapter, we consider the problem of labelling a set of points using a learnt graphical model. In the first part, we recall some basic concepts about graphs and hypergraphs and describe how we compute the multi-attributed hypergraphs used for the experiments. In the second part, we focus on matching the structure with a learnt model, using different steps of seeding and relaxation. In the third part, we discuss post-processing techniques that can be used after the graph matching process to guarantee the stochasticity of the correspondence. We then describe the experiments and the results obtained with our technique. In the last part, conclusions and limitations are drawn from the experiments.

## **Chapter 6 - Automatic Landmarking**

This chapter is really about applying the theoretical framework to the real problem of face landmarking in difficult cases (pose variation and occlusions). In the first part, we discuss the problem of optimising both the keypoint detection and the labelling for this particular problem. In the second part, the experiment framework is described. The next section deals

with the results obtained and the comparison with other techniques. In the last section, conclusions are made about the whole technique and its possible extension.

This chapter mainly concerns the results of feature detection using a combination of automatic point detection and labelling. The results are analysed under several variations and compared to results found in the literature.

## Chapter 7 - Conclusion

In the last chapter, we come back to the work achieved and the results obtained during the research project presented in the thesis. We summarise our contributions and evaluate how these discoveries might impact the research field and help automate particular 3D vision tasks. Paths for future research and improvement of the current methods are also proposed.

## Appendices

In the appendices, tables of full results are given when space was limited in the chapters of the thesis. One appendix explains in more details an idea that we proposed near the end of the project. Its aim is to discover the model set of landmarks that should be used by automatically detecting saliency on registered 3D surfaces. The last appendix presents the ideas that have been researched and not finished or later abandoned during this project. We think this can help save time to new research students in 3D landmarking.

## Chapter 2

# Literature Review/Background

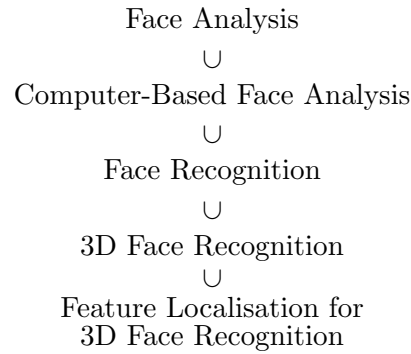
Faces are like snowflakes: every one is different. And like snowflakes, the face's structure is based upon a few simple rules. However, unlike snowflakes, faces are not made of ice, and therefore they don't melt, although what I am trying to do is make a computer what can spot faces that HAVE "melted" - so to speak - by spotting variations from the normal rules. To achieve this, I have spent the past three years scanning the faces of snowmen. I am beginning to suspect this was a mistake.

---

Rick Tongs impersonating Tim J. Hutton, In Tim J. Hutton's PhD thesis *Dense Surface Models of the Human Face* – University College London (2004)

The scope of this thesis is fairly large and encompasses a lot of different specific fields in different areas of research, from 3D differential geometry to graph theory, via machine learning techniques. The main objective of this chapter is not to detail every technical field used in this thesis but rather to focus on the fields of application, where limitations are more easily observable. Our approach has been data-centred rather than method-centred. Our first and foremost goal is the analysis of the human face, from the understanding of the shape of its parts, to the recognition of their attached identity and expression. Here, different theoretical areas are linked together to solve a particular problem: the processing of 3D surfaces for face recognition.

This chapter follows the journey made throughout the PhD, from the very general to the very specific:



Learning the history of these contributions and finding the state-of-the-art results for each particular problem have been essential steps to determine the gaps in research that we aimed to fill: 3D face landmarking towards automatic non-cooperative recognition.

Background information related to some of the tools or techniques used in this work will be presented separately throughout the thesis when appropriate.

First, a very coarse history of face analysis is given. Then, a few pioneering papers about face recognition are presented. These papers give enough examples to allow the presentation of a classification in the following section. A literature review on recognition systems is given afterwards. The last sections are used to evoke some related topics like feature detection, the psychological aspects of vision that can be useful and to evoke some of the remaining problems in face processing.

## Rationale

**What:** To review the literature in the field of face analysis and face recognition.

**Why:** To discover interesting problematics and gaps in research to be filled.

**How:** By understanding the history of field and analysing recent advances and their limitations.

**Priorities:** The background literature was reviewed with a bias toward 3D face techniques and non-cooperative face recognition.



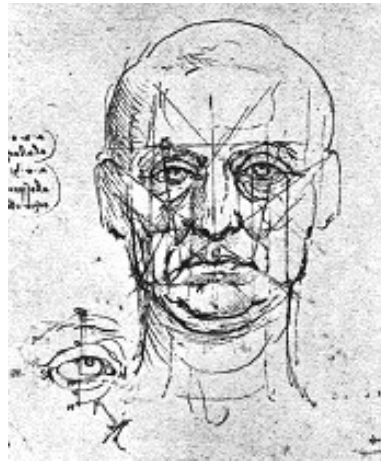


Figure 2.1: Proportion of the face and eyes. Sketches by L. Da Vinci.

## 2.1 Face Analysis

### 2.1.1 Brief History of Face Analysis

Face analysis is a very old topic that has almost always existed in art and science. It is hard to know who was the first person to put a measurement instrument onto someone's face, but chances are that it was an artist or philosopher looking for perfect proportions and the definition of beauty. Plato with his "beauty, proportion and truth" (Philebus 65a-66a) started to theorise about a knowledge that dates back to the ancient Egypt and possibly before: proportions in the body and face contain information that act upon our judgement of beauty. The analysis of face proportion in art continued for a very long period and still exists today.

The ancient Greeks gave birth to another study of faces: the physiognomy which claims that "It is possible to infer character from features" (Prior Analytics(2.27), Aristotle). This pseudo science is now discredited but had a lot of followers during the Renaissance and has introduced new ways of classifying faces according to their shape and related features.

The understanding of the face made a great leap forward with the emergence of modern anatomy in 15th century Italy. The knowledge of the pure mechanical interaction of the muscles and bones behind the skin is still fundamental for a lot of face related studies, especially those dealing with shapes and expressions.

Most of the other fields in face analysis emerged during the 20th century. The improvement of medical techniques allowed the first face skin graft in 1917 (by Harold Gillies) which in turn led to the development of facial plastics and cosmetic surgery. The importance of aesthetics in facial surgery (medical or not) is quite obvious. Therefore, there was a great need for knowledge concerning faces, psychology and ethics in order to know what a face should be.

The study of neurology, psychology and sociology that also flourished in the 20th century started to ask fundamental questions about how humans can see and identify faces. What are the limits of these capabilities and how important is the face in human social interactions? Many properties of human face-image processing have been discovered. Many biases in face recognition linked to a too specific specialisation have been documented, like the Thatcher effect, the own-race effect, the own-age effect, the hollow face illusion and many others [Thompson, 1980]. The ability to recognise faces from birth to adulthood has been studied [Campbell et al., 1995], [Slater and Quinn, 2001], in normal individuals as well as in people with disabilities like prosopagnosia (face blindness) and schizophrenia. Furthermore, psychological experiments have been done to determine our ability to judge age, sex and attractiveness [Schaefer et al., 2006], as well as emotions.

The 20th century has also seen the rise of face recognition as a science. In 1955-56 the first facial composite was designed (case Janet Marshall) using photo samples instead of simple drawings. At this time, the police artist had to be both an artist and a photographer in order to produce realistic portraits. But, with time, the system improved and sets of ready-made normalised features were produced to allow rapid facial composite creation. Nowadays software has taken over, but research still continues to improve the systems. The use of face and skulls in forensics also developed with applications in justice and occasionally in archaeology (e.g. mummy face reconstruction [Gill-Robinson et al., 2006]).

The democratisation of computers has allowed for new applications in face analysis. Some applications aim to detect expressions [Whitehill et al., 2008], [Zafeiriou and Pitas, 2008], [Zhan et al., 2008], spontaneity of expressions [Valstar et al., 2006], driver drowsiness, attractiveness [Whitehill and Movellan, 2008], and so on. Automatic face recognition started to appear in the 1970s, with the main purpose of allowing the development of identity checking, surveillance systems and improving human machine interaction, especially in robotics.

4. EYES	1	2	3	4	5	28
	a. Opening					
	Slit	-	Medium	-	Wide	
b. Spacing	1	2	3	4	5	29
	Close - Medium - Wide					
	Close	-	Medium	-	Wide	
c. Shade	1	2	3	4	5	30
	Light - Medium - Dark					
	Light	-	Medium	-	Dark	
5. NOSE	1	2	3	4	5	31
	a. Length					
b. Nostrils	1	2	3	4	5	32
	Narrow	-	Medium	-	Flared	
7. CHIN	1	2	3	4	5	28
	a. Front					
b. Profile	1	2	3	4	5	29
	Receding	-	Straight	-	Jutting	
8. EARS	1	2	3	4	5	30
	a. Length					
b. Protrusion	1	2	3	4	5	31
	Flat	-	Medium	-	Sticking Out	
c. Lobes	1	2	3	4	5	32
	Attached	-	Medium	-	Not Attached	
9. CHEEKS	1	2	3	4	5	33
	Sunken	-	Average	-	Full	

Figure 2.2: Extract of the questionnaire for feature characterisation filled by a juror. From [Goldstein et al., 1971].

## 2.1.2 Face Recognition

### 2.1.2.1 The Emergence of Computer-Based Face Recognition

The arrival of the computer age has allowed real experimentation on image processing of the face for recognition. The first major written trace of face recognition using a computer is a technical report published in 1966 by Bledsoe ([Bledsoe, 1966] cited in [Goldstein et al., 1971]).

Not long after, [Goldstein et al., 1971] started to look at face recognition using coarse characterisations of features made by a set of jurors. At this time, the computer's only purpose was to mimic the human decision-making process by finding a best correspondence, using a simple description of the face. The description gives 34 features including the shapes, sizes (small, medium, large), aspects of the nose, mouth, hair, chin, as well as some distance between features (short, medium, long), and other visual characteristics (see Figure 2.2). The measure was made by several people and correlated using simple statistics. The principal idea of computer vision was already there: the huge quantity of information contained by the face should be reduced to a discriminative subset in order to be matched by a computer.

At the same period in 1973, Kanade in his PhD thesis [Kanade, 1973], showed that it is possible to automate the whole process. His method was to find landmarks (distinguishable

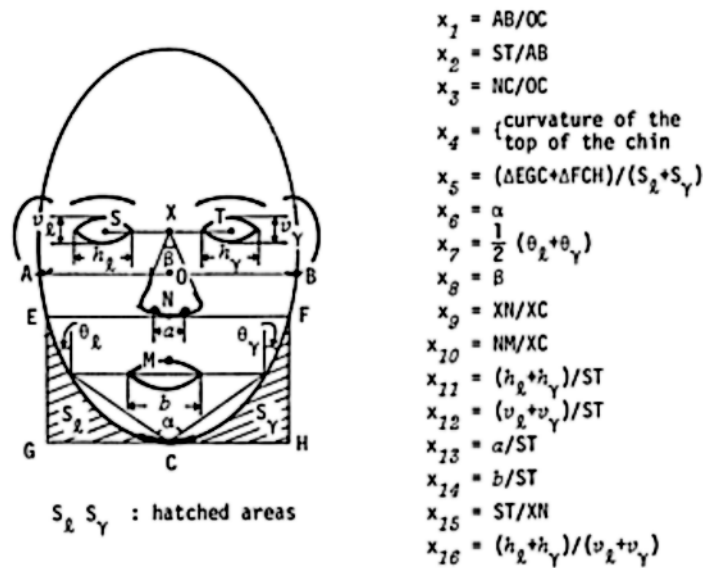


Figure 2.3: The 16 measures used for the matching process in [Kanade, 1973].

points) on a 2D picture using an edge map. The corresponding picture in the database was selected using a simple distance in the feature space.

One of the more interesting breakthroughs in face recognition occurred in 1991 when Turk and Pentland [Turk and Pentland, 1991a] designed a face recognition system based on a representation of the face introduced a few years before by [Sirovich and Kirby, 1987] called “Eigenpictures”. The idea is that the whole picture of the face is approximated by the combination of a relatively small number of “eigenpictures” (renamed “eigenfaces” in face recognition). These “eigenfaces” form the generative basis of a low-dimensional subspace of the space defined by all the pictures of a big dataset. They are computed from the whole high-dimensional space using a Principal Component Analysis (PCA). The dimension of this face space is the number of pixels in the image and each face image is represented by the sum of the vectors corresponding to the intensity value of its pixels. The PCA determines the directions in which the inter-model variations are greater and thus gives a smaller basis that describes most of the major variations in the database. These variations are closely linked to identities if pose, illumination and expression are controlled..

These first attempts at face recognition systems show two general approaches. The first one is the feature-based approach that is used by [Goldstein et al., 1971] and [Kanade, 1973].

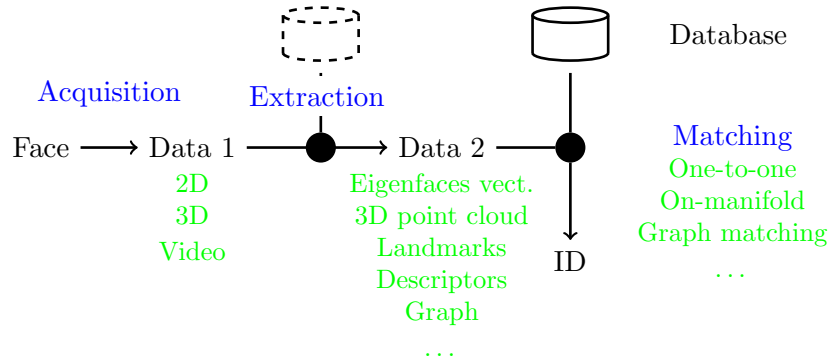


Figure 2.4: Workflow of face recognition systems on-line part.

The second one is the holistic approach as used by Turk and Pentland [Turk and Pentland, 1991a]. These two are the ancestors of almost all the face recognition techniques that are used today.

### 2.1.2.2 Taxonomy of Approaches

Before detailing the evolution that has occurred since the first face recognition systems were developed, some comments about how to classify those methods will be presented. Every paper on face recognition has used different methods on different kinds of data. It is quite difficult to classify them without giving more importance to one criterion of classification over another. Here is a short explanation of how the papers will be classified in the rest of the literature review.

**Common workflow** A common workflow exists for the on-line part of all face recognition systems. The different steps shown in figure 2.4 are:

- **Acquisition:** The capture of the person's face to identify. In this project, data acquisition is not discussed. In our systems the input is always a digital representation of faces (2D picture, 3D mesh, structured point cloud).
- **Extraction:** A critical part of all facial recognition systems is to reduce the amount of data used for matching. The extracted information should be discriminative. Some of the techniques can use previously learnt material to help this process (represented by the dashed database). For example, in [Turk and Pentland, 1991a], the extracted information is the vector combination of eigenfaces which best approximates the 2D image

(the learnt material is the eigenfaces). In [Kanade, 1973], the extracted information is a set of 16 landmark-related measures that can be seen in figure 2.3.

- **Matching:** In the matching process, the identity of the face should be found. The problem is to find the face that is the most similar. A basic technique is to compare the probe with the whole database and get the closest corresponding face given a certain metric (or more generally an evaluation function, as the subadditivity [or triangle inequality] and symmetry are not needed). For example, this could be the computation of a “distance” between the points cloud of the probe and each point cloud in the database. This kind of method can be fast when the object to match is small (for example a vector of low dimensionality) but can become time-expensive when the database gets bigger and when the size of the descriptor to match is big (for example point clouds, 3D meshes). Another technique is to search only a local area of the space, testing only the closest faces using a rejection system to eliminate the less probable candidates. Some other kinds of matching methods iteratively eliminate bad candidates until very few remain.

In practice, the separation between these processes can sometimes be less clear. Indeed the extraction of the discriminative information can be split between the pre-processing part and the matching part, meaning that the matching process is, by itself, an extractor and a matcher. It happens, for example, when the matching algorithms are based on neural networks. A first extraction takes place to prepare the data for the matching and the neural network will refine the selection with previously learnt patterns.

The extraction part is probably the more tricky and can contain a lot of different processes (face detection, hole filling, spike removal, noise reduction [smoothing], caricature, local feature detection, registration and projection).

The extracted information can be seen as hidden variables which, imaged through a specific evaluation function, is an approximation of the face or the structure of the face. The problem is to find the inverse process.

**Classification criteria**

**Data type** One obvious criterion is the input data used. In the case of face recognition, it can be a 2D image, a range image, a 3D model, a video, or even a 3D video [Sun and Yin, 2008]. Another, that can be very different, is the kind of data used just before the matching process which is generally closely linked to the matching process.

**Feature-based or holistic** The differentiation between feature-based and holistic methods is also very important. Almost all the techniques of face recognition use two kinds of information present on the face: featural and configural information. The featural information is contained in a “localised” part of the input data while the configural information is linked to variation between “remote” parts. In the case of holistic techniques, both featural and configural information are mixed. For example, eigenfaces composing a picture are by themselves pictures which contain both featural and configural information. The comparison of face cloud points or the ICP fitting on the whole 3D model of the face are holistic methods too. In the case of the feature-based methods, featural and configural are distinguishable. For example, the shape of the nose will be featural information while the distance between eyes will be configural information.

Notice that some holistic methods can detect local features as a side effect. This is the case with the Non-negative Matrix Factorisation (NMF) used on 2D faces (see [Lee and Seung, 1999]).

**View-based (anisotropic) vs pose-invariant (isotropic)** Another criterion is the isotropy of the data before matching. For example, in the case of a system using a holistic method on 2D pictures one or several implicit projection directions are necessarily chosen (usually front and/or profile view). Such systems will be bound to deal explicitly with either registration and/or orientation change tolerance. Isotropic methods use direction independent data and then do not have to deal with registration before the matching: registration is included in the matching process. This is the case for feature-based methods like graph matching or for holistic methods as in point cloud distance.

**Types of algorithm used** Most of the algorithms used for matching in the literature are very different, but they can be grouped in different coarse categories. The techniques cited here are explained in the following sections.

**Simple distances** The more common and simple methods define similarity between models as a distance in a simple space. It can be an Euclidean distance in the unmodified feature space, a Hausdorff distance between two point clouds or other kinds of simple comparison.

**Distance in a reduced space** Both holistic and feature-based methods may consider high dimensional data before matching. Therefore, some of them use space reduction techniques to work on an approximation of the input data. The Principal Component analysis (PCA), the Independent Component Analysis (ICA) or the Non-negative Matrix Factorisation (NMF) are good examples of such space reduction.

**Tessellation of a reduced space** Other techniques try to reduce the space dimensionality by considering the classes that compose the space. They do not try to find the best approximation of the whole space, but rather try to find the subspace that best scatters the class corresponding to identities, or alternatively, try to find sets of class-separating hyperplanes. For example, this is the case with Linear Discriminant Analysis (LDA), Kernel Discriminant Analysis (KDA), or multiclass Support Vector Machine (SVM) techniques.

**Black-box statistical learning** Some techniques rely on automatic learning techniques to reduce the quantity of information. These systems are usually opaque: the information can not be read or interpreted directly. Mainly, two kinds of methods are used for this purpose: the ones based on Neural Networks (NN), and the ones based on Hidden Markov Models (HMM).

**3D transformation methods** When dealing with 3D points sets, two methods appear quite often to determine the similarity between two models. The first is the Iterative Closest Point (ICP) which registers the two models. The distance between the two models after registration is a similarity measure. The second (less used) is the Thin Plates Spline (TPS)



technique which deforms one model to the other. The binding energy is then used as a similarity measure.

**Graph matching methods** A last class of techniques are those using graph matching. In this case, the global similarity is obtained progressively using local similarities and relational similarities.

**Retained classification** In this thesis, the first level of classification corresponds to the holistic, feature-based and hybrid techniques. In the second level, the dimensionality of the data before matching is taken into account. The last level can be used to group papers that use similar matching techniques. We do not make fundamental differences between 2D and 3D methods because the line between the two is very thin. Indeed, some 2D methods use multiple view and some 3D methods use 2D information after projection. Figure 2.5 shows the classification of most of the matching techniques presented in the literature review.

### 2.1.2.3 Holistic Methods

Holistic methods are probably the most used for face recognition especially when 2D images are used.

**2D image matching** Most of the 2D image matching techniques are descended from the PCA technique of Turk and Pentland (see section 2.1.2.1) and represent the image as a big vector. Such techniques try to best reduce and organise the face space generated by these vectors.

In [Belhumeur et al., 1997], it was noticed that the main drawback of the PCA methods was their sensitivity to light change in intensity and direction which often required them to discard the first two or three principal components. Their idea is that the faces of the same individual lie into the convex subspace of the whole face space and are then linearly separable. While a classic PCA only reduces the dimensionality of the face space, the Fisher Linear Discriminant method (FLD) takes into account the included classes (identities). This method tries to find directions that maximise the separation between classes, defined as the ratio of the variance between the classes to the variance within the classes. Figure 2.6 gives

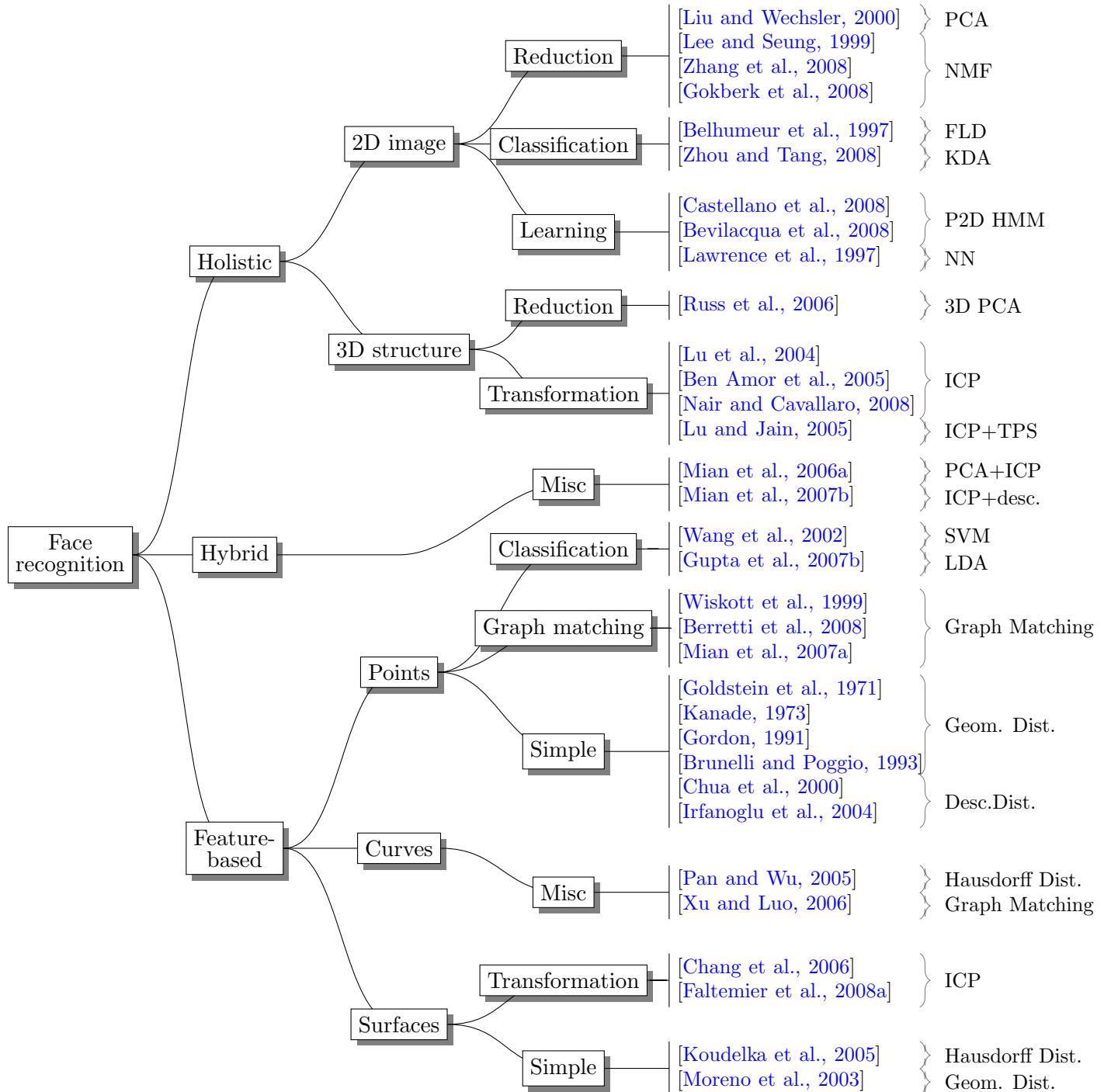


Figure 2.5: Taxonomy retained to present most of the matching techniques used in face recognition. The techniques will be detailed in the sections 2.1.2.3, 2.1.2.4 and 2.1.2.5.

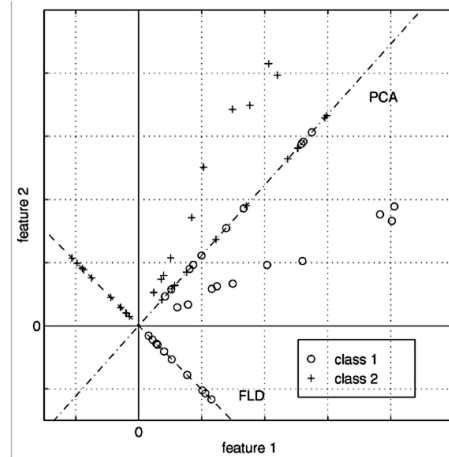


Figure 2.6: Difference of principles between a PCA and a FLD component. Extract from [Belhumeur et al., 1997].

a good idea of the principle with two classes in 2D: the aim is to find directions that best scatter classes, not directions that best approximate the whole space.

In both [Lee and Seung, 1999] and [Zhang et al., 2008], Non-negative Matrix Factorisation (NMF) is used to reduce the dimensionality of the 2D face space. This technique is very similar to a PCA but varies on one point. If we denote  $V$  as the matrix  $n \times m$  where  $n$  is the number of pixels of one image and  $m$  the number of images, then the aim of a classic PCA method is to find  $W$  and  $H$  such that  $V \approx WH$  where  $W$  is a basis and  $H$  the encoding of each image of the database in this basis. In the case of the PCA, the vector in the basis  $W$  are all orthogonal and can be positive or negative as well as the coefficient in  $H$ . The NMF method is different in the sense that all the coefficients computed in the factorisation  $V \approx WH$  are positive. It means that only additive linear combinations of the basis are allowed (see figure 2.7). Consequently the vector in the database looks more like features or localised parts of the face and the encoding of one image is more like the selection of this feature. This method gives quite good results, especially in [Gokberk et al., 2008], where it reaches between 71.55% (1 picture per person during the training) and 99.67% (4 pictures per person during the training).

Another approach to enhance space reduction is to use Genetic Algorithms to construct a better basis on the face space. It is used in [Liu and Wechsler, 2000], where Evolutionary Pursuit is implemented to improve the basis found by a basic PCA. The chromosomes

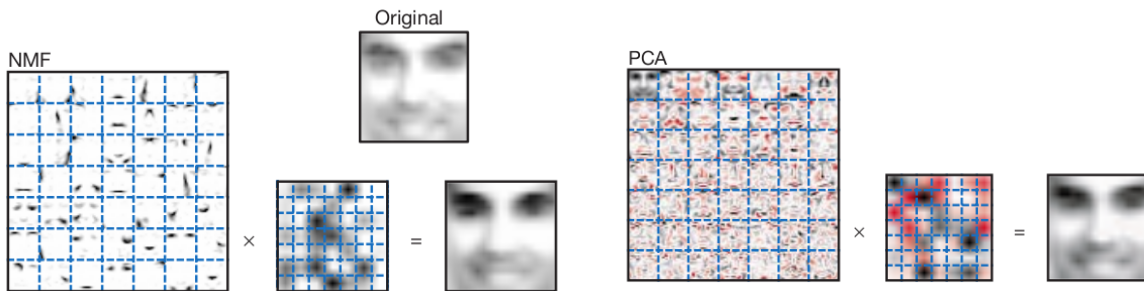


Figure 2.7: Difference between a NMF and PCA method. The red corresponds to negative values. Extract from [Lee and Seung, 1999]

correspond to the rotation of vectors of the basis and the evolution is driven by a fitness function link to the overall accuracy achieved so far. This method reaches 92.14% rank-one recognition with 26 vectors on the FERET database.

The limitation with all the previously discussed face space reduction systems is that all consider that identities are separable in a linear subspace. But what if this subspace cannot be approximated linearly? Methods have been designed to deal with the non-linearity class separation and some of them have been used in face recognition. In [Zhou and Tang, 2008], the Kernel Trick is used (see figure 2.8) with a classic Linear Discriminant (LDA) and an Improved Linear Discriminant Analysis (ILDA [Zhou et al., 2006]). The two corresponding methods, KDA and KIDA, have been tested on the Yale database using 5 training samples per person and give 95.1% and 97.5% accuracy respectively. However, no comparison is done between these non-linear methods and their linear counterpart in this paper. Some comparisons can be found in [Li et al., 2003b], where the PCA and LDA methods are compared to their kernel version KPCA and KDA (see figure 2.9).

In [Feng and Yuen, 2000], the 2D image is registered using a 3D model and symmetry information is used to complete the missing part. Then, the Spectroface technique is used to extract invariant information from the image. The Spectroface method consists of decomposing the image by a wavelet transform until a given compression is reached. Next, a Fourier transform is computed. This technique is known to be robust in dealing with in-plan rotations, translations and scale changes. However, in this paper the rank one recognition rate

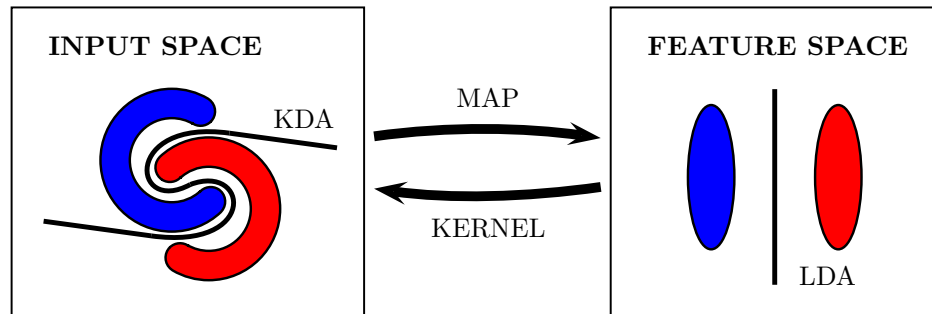


Figure 2.8: Example of a non-linear problem. The space is assumed to be transformable into a space in which Linear Discriminant Analysis (LDA) can be applied to the data. The kernel of the separator can be computed in the input space and the method is then called Kernel Discriminant Analysis (KDA). Graphic inspired by [Li et al., 2003a].

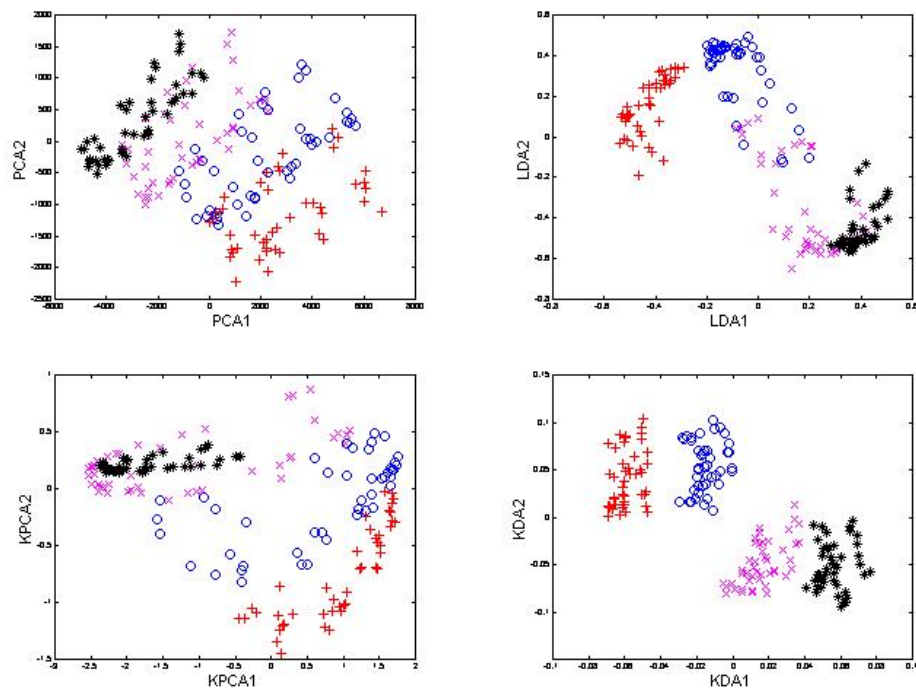


Figure 2.9: Different distributions obtained with the PCA, LDA, KPCA and KDA methods. The complete experiment included 540 pictures of 12 people. Here, the first four subjects are shown. The KDA seems to be more efficient at scattering the classes. Extracted from [Li et al., 2003b].

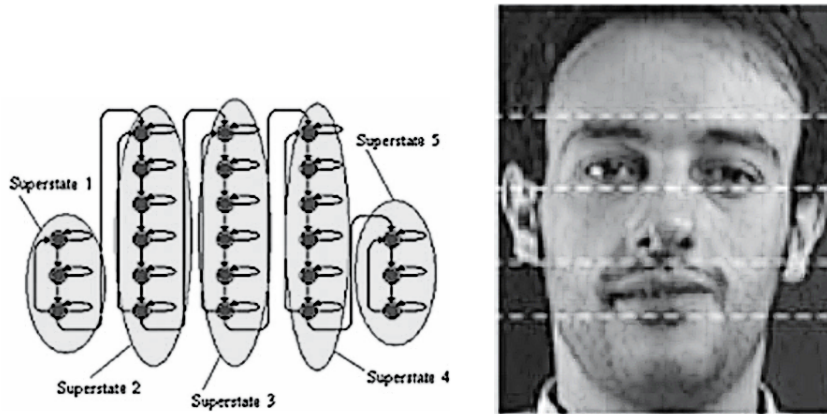


Figure 2.10: P2D-HMM 3-6-6-6-3 extract from [Bevilacqua et al., 2008].

Each superstate corresponds to a horizontal strip in the picture (5 strips) and each inner state corresponds to a vertical part of the strip: 3 parts for the first and last strips (forehead and chin). 6 parts for the other strips (eyes, nose and mouth strips).

is below 85% in the semi-automatic mode (landmark adjusted by hand for the registration) and decreases to 75% in the fully automatic mode.

In [Bevilacqua et al., 2008], a Pseudo 2D Hidden Markov Model is used (Pseudo 2D-HMM also called Embedded HMM) in which every state corresponds to a zone of the image (see figure 2.10). The Pseudo 2D-HMM is basically a classic 1D HMM which corresponds to a 2D journey on the picture (Right to Left and Top to Bottom). The value attributed to each state is a compression of the corresponding part of the picture. The compression is obtained using an Artificial Neural Network (ANN) with a bottleneck architecture (230 neurons for the input and output and only 50 for the hidden level). This technique seems to give quite good results but the number of probes used in the experiment is too small to give a real impression of the capability of this technique. Indeed, they reach 100% recognition rate on 51 people from the ORL database but, out of the ten pictures available per person, nine were used in the training part and only one was used as a probe. A similar technique is also used in [Castellano et al., 2008], on 3D data converted into the Canonical form.

**3D matching** To compare two 3D models without using 2D methods can be done but requires new techniques. Most of them are based on the difference remaining between two surfaces or point clouds after registration. Very often, the Iterative Closest Point technique



Figure 2.11: Different expressions of the same face with their corresponding Canonical form. Extract from [Bronstein et al., 2004b].

(ICP) introduced by Besl and McKay in 1992 [Besl and McKay, 1992] is used for the registration. In this iterative method each iteration consists of associating the closest points from the two models together ( $\sigma$  correspondence) and then trying to find a rigid transformation (translation  $\vec{t}$  and rotation  $R$ ) that minimises the mean square distance between the sets of points. The function to minimise is:

$$f(R, \vec{t}) = \sum_{i=1}^{N1} \sum_{j=1}^{N2} \delta_{i\sigma(j)} \|p_{1_i} - (R(p_{2_j}) + \vec{t})\|^2 \quad (2.1)$$

where  $N1$  and  $N2$  are the number of points in the two models and  $p_{x_k}$  the  $k$ -th point of the  $x$ -th model.

In [Bronstein et al., 2004a], [Bronstein et al., 2007] and [Bronstein et al., 2004b], the idea that expression can be considered as non elastic deformation (transformation where the surface is bent but not stretched) of the face is used, which means that the distances on the surface of the model (geodesics) should be invariant. To verify this hypothesis, they first transformed the face to make it invariant to expression and then used a classical 3D rigid matching method. Firstly, the geodesic distance between each point of the source model is computed to form a new surface called the Canonical form of the face (see figure 2.11). Then, to match the two surfaces, their moment signatures are computed (see [Elad et al., 2001] and compared with a simple Euclidean distance. Only a small database has been used to test the accuracy of this method. It appears to be a good way to deal with expressions but it is possible that some discriminative information is lost while using this transformation. In [Castellano et al., 2008], this data representation is used and is matched with a 3-6-6-6-3 Pseudo 2D HMM and gives 98% of Rank one recognition on a small database of 51 faces.

In [Lu et al., 2004], the model is coarsely registered using landmarks detected with the Shape Index of the curvature. Then a hybrid ICP technique was used to align finely the probe with the model. Both the Root-Mean-Square (RMS) distance given by the ICP and the cross-correlation between the Shape Indexes of a set of corresponding points were used to determine the matching scores. The matching error rate obtained was 3.5% on a database of 18 people with 113 test scans.

In [Ben Amor et al., 2005], the ICP method is used on both a complete face and smaller parts (detected with a watershed segmentation). The residual distance between the 3D surfaces after registration is used to identify the degree of similarity between a probe and a model from the database. The same kind of technique is used in [Nair and Cavallaro, 2008], where landmarks are detected to segment the face (e.g. forehead, eyes, nose). The best rank one recognition rate achieved is 93.7% on the GavabDB database using only the nose region. In [Lu and Jain, 2005], rigid matching using ICP is coupled with a non-rigid method to improve the results. The Thin Plates Spline deformation (TPS [Bookstein, 1989]) is performed between the two registered surfaces and the wrapping energy is used to determine the similarity between them. On a database of 196 people, the rank one recognition rate increases from 85% for the ICP alone to 89% with a TPS-based classifier. However, the ICP-based method has a serious drawback: one comparison is costly because the process is iterative and all the models on the database have to be tested to recognise one probe.

In [Lu and Jain, 2005], the displacement vector field created by the non-rigid TPS registration between two faces is used to discriminate (using a SVM) between faces corresponding to different identities. More classic techniques using rigid registration distances are also used in this paper. Tests show that combining the techniques improves the rank-one matching accuracy. The best rank-one matching result achieved in this paper is 91%, with 18 errors in 196 scans (98 neutral, 98 smiling).

In [Russ et al., 2006], PCA is used directly on the 3D model. In order to do this they must construct a dense correspondence between all the faces in the database and a model in order to be sure that each dimension in the face space is represented on the probe and corresponds to the same point of the face. First, five features of the face are used to scale and coarsely register the face. Then, the ICP method is used to align more precisely the face on the model. The dense correspondence is made between the two using the Normal



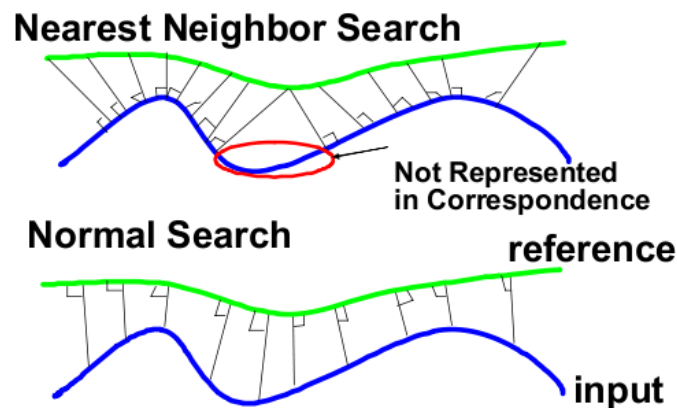


Figure 2.12: Difference between the Nearest Neighbour Search and the Normal Search. Extract from [Russ et al., 2006].

Search method (see figure 2.12). This 3D PCA technique gives a rank one identification rate of between 91% and 97% on the FRGC database (Spring and Fall 2003).

#### 2.1.2.4 Feature-Based Methods

After Goldstein [Goldstein et al., 1971] and Kanade [Kanade, 1973] (introduced in section 2.1.2.1) many people have looked at feature-based face recognition. Usually, the detected features are attached either to points, curves or surfaces.

The way the features are detected will be discussed both here and later in section 2.2.1. Here we focus on the matching techniques used and on the results obtained when they are applied to face recognition.

**Point-based features** In 1991, Gordon in her PhD thesis [Gordon, 1991] tested two 3D face recognition techniques. One is holistic and consists of computing the volume between two registered 3D representations of the face. The other is feature-based and consists of using the local curvature and the distance between landmarks. The features are localised using ridge and valley lines, umbilical points, symmetry matching and so on. A simple Euclidean distance in the scaled feature space has been used to compute the similarity between faces. Results of 90% and above have been reached on a 24 model database of 8 people using a subset of features. Similarly, in [Brunelli and Poggio, 1993], one of the two 2D intensity

techniques presented uses geometrical measures involving 35 detected features. A weighted Euclidean distance is used and reaches 90% on a 188 face database of 47 people.

A very different kind of landmark detection and matching is used in [Wiskott et al., 1999] where the Elastic Bunch Graph Matching technique (EBGM) is detailed. This method is based on two assumptions:

- First, that landmarks can be consistently represented by a set of local Gabor wavelet transforms of different frequencies and orientations called “jet” (See figure 2.13);
- Second, that a graph over those landmarks and the corresponding jets are discriminative enough to allow face recognition.

A model of the graph containing bunches of possible jets for each vertex is constructed from a hand-landmarked database and is used to help the registration of the graph onto new probe faces. Once the graph is in place, a similarity function is computed using the difference between the jets of corresponding vertices. On the Feret database 98% rank one recognition was achieved on frontal to frontal recognition and 84% for profile to profile recognition. This approach is quite robust in dealing with pose variation (compared to other techniques) but remains rather limited by the fact that the local measures attached to the landmarks are intrinsically two-dimensional. Consequently the rank one recognition rate for frontal to profile and profile to frontal matching are both under 20%. An eye seen from a frontal view or a profile view will not generate the same jets, so this technique is probably not the best candidate for profile to frontal recognition.

In [Chua et al., 2000] a new point descriptor called point signature [Chua and Jarvis, 1997] was used. The point signature of a point  $p$  consists of a set of  $n$  pairs  $(\alpha, d)$  computed from the curve defined by the intersection of the object surface with a sphere of fixed radius  $r$  and centre  $p$ . The plane that best fits this curve is determined and translated to the point  $p$ . Every  $\alpha = k * 360/n$  degree, the distance  $d$  between the point on the curve and the computed plane is measured. The point with maximal distance to the plane is used to construct the reference vector. The reference vector used to compute the angle point to the projection of the point which distance to the plan is maximum. The evaluation of this descriptor is not very detailed in that paper and concerns a database of only six people.

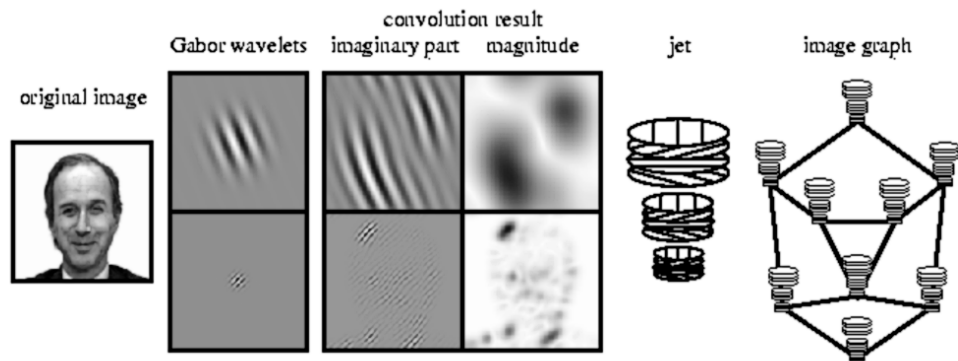


Figure 2.13: Different elements playing a part in the EBGm technique. Extracted from [Wiskott et al., 1999].

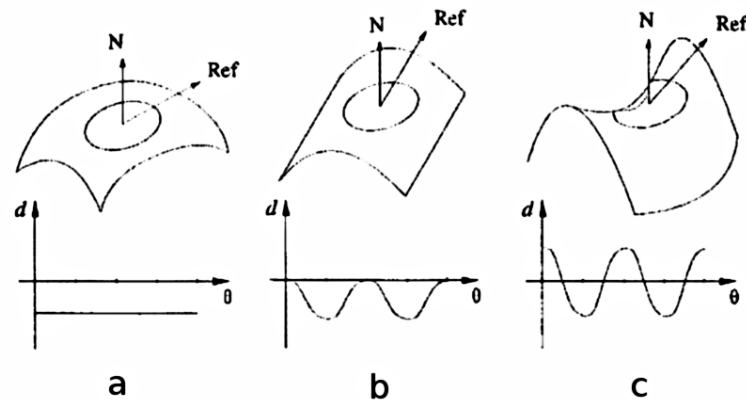


Figure 2.14: Example of point signature. Extracted from [Chua and Jarvis, 1997].

However, [Irfanoglu et al., 2004] have used it for comparison and achieved quite good results (over 90% rank-one recognition) on the 3D-RMA database.

In [Wang et al., 2002], both the Gabor wavelet transform (on the 2D grey level picture) and the point signature (on the 3D image) are used to extract information from the face. However, this time the feature space is not searched by classic distance but by looking at one-to-one class separation using a Support Vector Machine technique (SVM) and a decision Direct Acyclic Graph. They reach around 90% recognition rate on a small database of 50 people.

When dealing with feature-based recognition it is sometimes hard to know if the data extracted is discriminative enough. In 2007, [Gupta et al., 2007b] used a hand-landmarked

Table 2.1: Rank-one recognition rate obtained in [Gupta et al., 2007b].

Algorithm	Neutral/Neutral	Neutral/Non-Neutral	Neutral/All
2D range PCA	70.21	68.31	69.68
2D range LDA	91.25	95.10	92.31
EUC LDA	97.92	96.72	97.59
GEO CURV LDA	98.75	98.36	98.64

set of 25 landmarks and tried to see what recognition performances could be reached using only the geodesic distance (GEO) and the ratio between the geodesic and the Euclidean distance (CURV). First, the most discriminative GEO and CURV features are selected with a Stepwise LDA. The first 117 GEO and 131 CURV features are then mixed and a final selection is performed keeping only 146 features. The space generated by these features is then approximated by 11 dimensions. The similarity between two faces is simply based on the Euclidean distance in this 11-dimension space. The results obtained on a 1128 3D model database of 105 subjects are given in table 2.1. The EUC LDA method is similar to the GEO CURV LDA method that we have just explained but it uses only Euclidean distance.

This paper shows that very simple geometric information can be very discriminative. The main problem remaining is that the landmarks are very difficult to find automatically. The same team has tried to see the variation performance when the picked landmarks were not anthropological [Gupta et al., 2007a], but this study is not very meaningful. It just shows that the non-anthropological landmarks selected for the experiment were not good, not that the anthropological ones are the best.

In [Berretti et al., 2008], a segmentation of the 3D face into Iso-Geodesic surfaces is used to construct a graph (see figure 2.15). The vertices of this graph correspond to the computed areas and the edges correspond to the relationship between those surfaces in terms of 3D Weighted Walkthroughs(3DWW). The Walkthroughs between two sets of points,  $A$  and  $B$ , is a triple  $(i, j, k)$  where

$$i = \begin{cases} -1 & \text{if } x_b < x_a \\ 0 & \text{if } x_b = x_a \\ +1 & \text{if } x_b > x_a \end{cases}, j = \begin{cases} -1 & \text{if } y_b < y_a \\ 0 & \text{if } y_b = y_a \\ +1 & \text{if } y_b > y_a \end{cases}, k = \begin{cases} -1 & \text{if } z_b < z_a \\ 0 & \text{if } z_b = z_a \\ +1 & \text{if } z_b > z_a \end{cases}$$

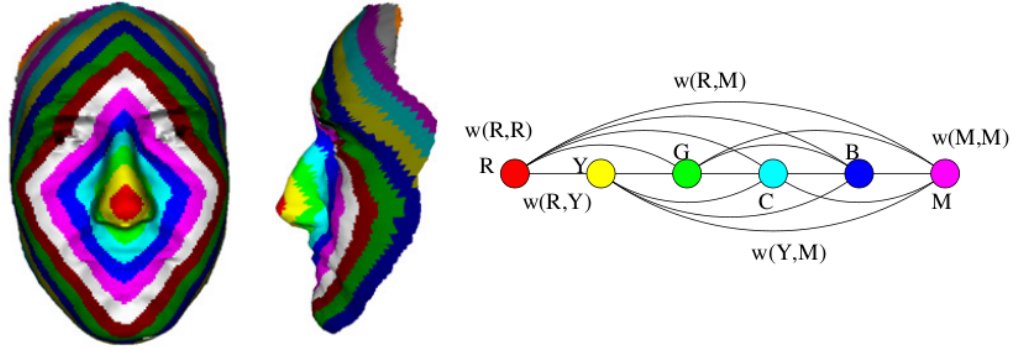


Figure 2.15: Example of Iso-Geodesic surfaces and the corresponding graph. Extracted from [Berretti et al., 2008].

and where  $a \in A$  and  $b \in B$ . The weights are linked to the number of couples  $(a, b)$  that correspond to each Walktrough between  $A$  and  $B$ . The created graph is matched to the database using a similarity function based on distances between 3DWW. This very small representation of the face gives a rank one recognition rate of above 90% on the Gavab 3D database (61 people).

In [Mian et al., 2007a] and [Mian et al., 2008] a local 3D descriptor of the surface called Keypoint was introduced. For any point  $p$  of the surface a large neighbourhood is constructed with all the  $n$  points in the sphere of radius  $r$  centred in  $p$  and their coordinates are stored as columns in a matrix  $L$ . The covariance matrix  $C$  of these point coordinates is then computed:

$$C = \frac{1}{n} \sum_{k=1}^n L_k L_k^T - m m^T \quad (2.2)$$

where  $m$  is the centroid of the neighbourhood. A PCA is performed on  $C$  to get a matrix of eigenvectors  $V$  which is used to compute a normalised version of  $L$  and a “gradient-like” variable  $\delta$ :

$$L' = V(L - m) \quad (2.3)$$

$$x\delta = \max(L'_x) - \min(L'_x) - (\max(L'_y) - \min(L'_y)) \quad (2.4)$$

If  $\delta$  is greater than a certain threshold the keypoint is selected. These keypoints are quite repeatable for the same individual but vary between people (see figure 2.16). In the experiment 200 features are selected and a vector of 11 values describing the local surface is attached

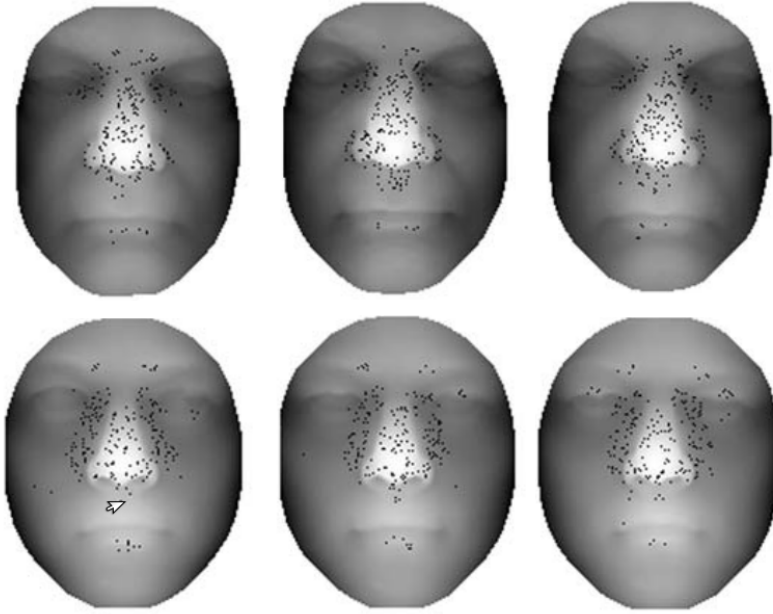


Figure 2.16: Repeatability of the Keypoint for the same individual (row) and variability between people. Extracted from [Mian et al., 2008].

to each one. The similarity between those vectors and the number of point matches with the gallery face are two of the similarity measures used. The graph between those points is constructed by a Delaunay triangulation and used to measure the two other metrics (the sum of the difference of in the length of corresponding edges, and the sum of the distance between corresponding vertices after registration). This method gives a rank one recognition rate of 93.5% on the FRGC v2 database and should probably be more robust than other methods on databases with more significant pose variation.

**Curve-based features** It is quite hard to separate curve-based methods from the others because most of them do use surfaces or landmarks in addition to curves. Here, we present some rare examples where only information along curves is used.

In [Pan and Wu, 2005], the symmetry plane of the face is computed to extract vertical and horizontal profiles of the face. One-to-one profile matching is then performed using the Hausdorff distance. This landmark-free technique gives fair results on the 3D\_RMA database (6.67% EER in semi-automatic mode) and seems to give even better results when merged with the Statistical Discriminative Model (SDM) method described in the same paper (reaching 4.44% EER).

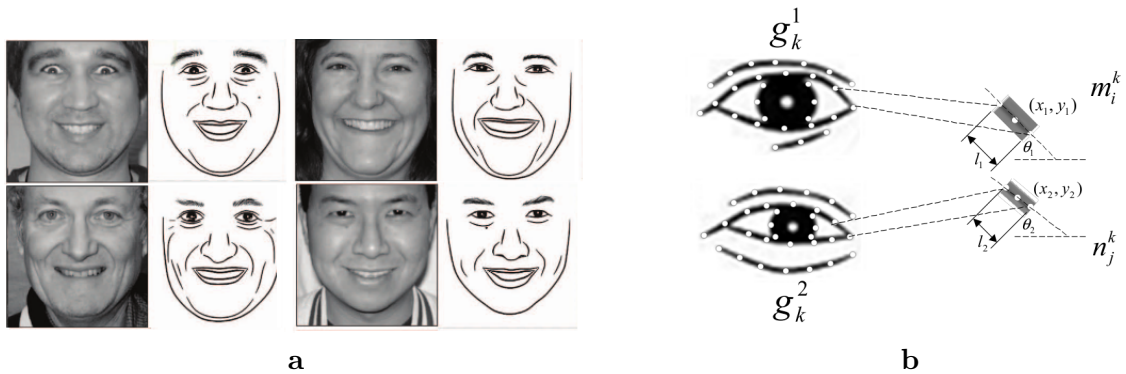


Figure 2.17: a - Input images and corresponding sketching result;  
 b - Similarity measurement (size, angle, distance) between corresponding segments.

Extracted from [Xu et al., 2008], and [Xu and Luo, 2006].

In 2006, [Xu and Luo, 2006] first experimented with sketch-driven face recognition from 2D images. The idea is to construct a facial composite of a face using an edge map (see figure 2.17a). The graph of the segment obtained is matched with the database using a similarity function, taking into account for each couple of corresponding vertices the difference in length of the segment, the difference in orientation, and the distance between their middle points (see figure 2.17b). They obtained up to 90% Rank one recognition on Neutral/Angry faces from the AR database. The results fall to 84% when the people are smiling.

**Surface-based features** The surface-based features methods are usually quite simple. Most of the time it consists of matching cropped parts of the image/mesh by using the same technique as most holistic systems.

A 2D example is given in [Brunelli and Poggio, 1993] where part of the face (eyes, nose and mouth region) are matched with a 2D grey-level correlation method. A 3D case is given in which an ICP technique is used on a small part of the 3D face. In [Chang et al., 2006] the ICP methods was tested on multiple crops around the nose (see figure 2.18) for recognition and compared to the use of an ICP on the whole face. They showed that matching a small region around the nose gives better results than matching the whole face with ICP. The average Rank-one recognition rate using nose matching is around 95% and 80% for the neutral and non-neutral face against 91% and 61% for the whole face (the database contains



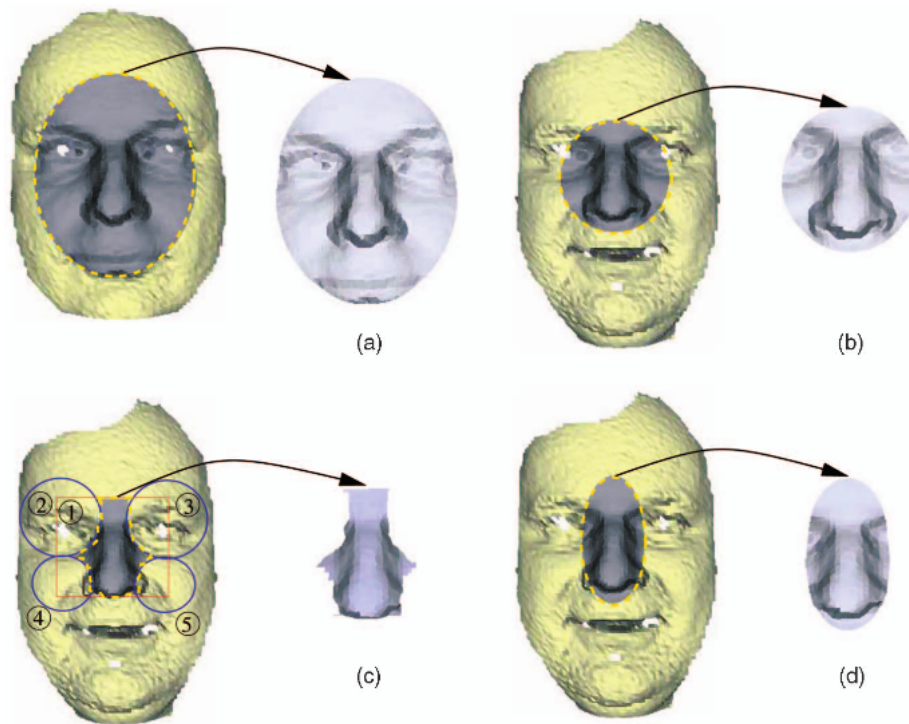


Figure 2.18: Different crops around the nose region. Extracted from [Chang et al., 2006].

a little more than 4000 scans of 449 people). When combining several nose matchings they reach 97.1% and 86.1%. However, no information about the computation time is given.

In [Koudelka et al., 2005] the 3D model is first registered using 5 landmarks detected via radial symmetry, directional derivative zero crossings maps and profile curves. The matching is performed with a fast Hausdorff distance between the probe and gallery model. The idea that close points on the range image correspond to close points in 3D space is used to constrain the search for the Hausdorff distance for each point and help reduce computation complexity. The rank one recognition rate obtained with this method is 94% on the FRGC v1 database.

In 2008, [Faltemier et al., 2008a] published a very interesting study about 3D face recognition using ICP region matching. On the probe face 38 regions are cropped and each one is matched using the ICP method with the corresponding region of the gallery faces (see figure 2.19). The distance after registration gives a score for each region and a count is used to combine the scores into a matching decision. The rank-one recognition rate achieved is 97.2% on the FRGC v2 database. If this technique gives quite good results in terms of



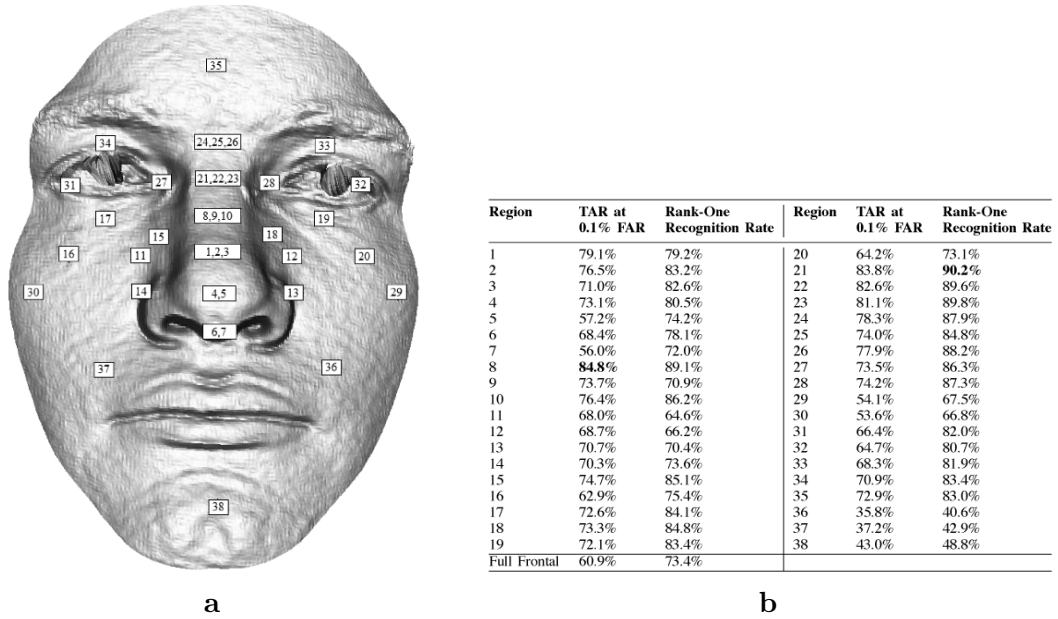


Figure 2.19: a - Centres of the 38 regions used for local ICP matching  
 b - Individual region matching performance  
 Extracted from [Faltemier et al., 2008a].

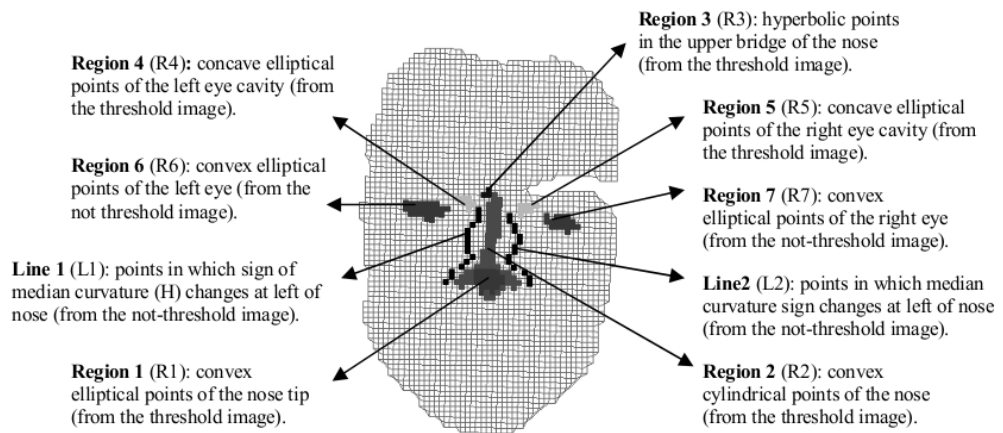


Figure 2.20: Region used to extract features. Extracted from [Moreno et al., 2003].

recognition, the computational cost is very significant: when used as a verification system (one probe against one model) it takes almost 10 seconds to get a decision (7.5 seconds for the data pre-processing and 2.3 seconds for the ICP matching).

Some other techniques are very similar to the point-based feature recognition methods with the exception that the extracted features are not points. For example, [Moreno et al., 2003] used 3D feature detection using curvature to extract 9 specific regions (surfaces on the mesh)

and curves (see figure 2.20). Then, 85 measures on these features were evaluated and ranked. In the experiment, the first 35 measures were used and the recognition was based on a simple Euclidean distance in the feature space. A rank one recognition rate of 78% was reached on a database of 420 3D models of 60 people.

### 2.1.2.5 Hybrid Methods

Some methods cannot really be classed as holistic nor feature-based because they combine techniques from both sides.

In [Mian et al., 2006a], weight is given to the conjecture that states that 3D shape is better than 2D texture for recognition. However, combining techniques still improves the recognition rates. In this paper they combine four recognition systems, using :

- a PCA on the textures;
- a PCA on the depth maps;
- an ICP registration of the upper part of the face (eyes and forehead);
- an ICP registration of the nose;

They reach 100% verification rate at 0.0006 FAR on the FRGC v1 database. Using only the two ICP systems gives 98%. They also show that the upper part of the face is shown to be more discriminative than the nose alone. The same team in [Mian et al., 2006b], reaches 98.03% and 89.25% identification rates on the FRGC v2 at 0.001 FAR. In this paper, they simply used an ICP matching with a mask in which only the nose, the eyes and the forehead are present.

In [Mian et al., 2007b], a multimodal 2D-3D hybrid technique is used for recognition. Four different matchings are combined to determine the identity of the subject. A 2D matching using Scale Invariant Feature Transform (SIFT) [Lowe, 2004], is combined with a 3D matching using Spherical Face Representation (SFR) (explained in section 2.2.1) to reject all improbable identities. The remaining models in the gallery are compared to the probe using two ICP matchings on the eye-forehead and nose parts of the 3D model. The recognition rate for this system achieved 99.7% for neutral expressions and 98.3% for non-neutral. The measures were performed at 0.001% FAR on the FRGC v2 database. The

Table 2.2: Comparison of methods and data representation performed by the same team on the 3D-RMA database. From [Dutagaci et al., 2006].

Representation	Method	Dimensionality	Performance
3D Point Cloud	2D DFT	799	95.86
	ICA	50	99.79
	NMF	50	99.79
2D Depth Image	Global DFT	127	98.24
	Global DCT	121	96.58
	ICA	50	96.79
	NMF	50	94.43
3D Voxel	3D DFT	127	98.34

main errors observed with these techniques were linked to bad detection of the nose (and then a bad segmentation of the 3D model). Orientation of the model, hair, and exaggerated expression were the principal cause of failure for the nose detection. Hair and exaggerated expression also led to false recognition during the ICP matching process.

#### 2.1.2.6 Comparisons

In [Dutagaci et al., 2006], a comparison between several data representations and several face recognition methods is performed. This paper is very interesting because the same methods are applied in the same conditions to the same database. So the differences in performance have a real sense here. While all methods are fairly good, the ICA and NMF seem to give better results with the 3D point cloud than with the 2D depth image (see table 2.2).

In 2008, the same kind of study was performed in [Gokberk et al., 2008]. The results show that the best methods in the tested set are the ICA and NMF on point clouds, and the Discrete Cosine Transform on the 2D depth map (see table 2.3 and 2.4). Once again, the ICA and NMF on the 2D depth map give worse results than on the point clouds, but we see that the difference gets smaller when the number of faces used in the training part increases. Two hypotheses can be made to explain this difference. The first is that this difference is linked to an imperfect registration before projection in the case of the 2D depth map. The second is that the fact of projecting the data along one direction introduces a bias which is prejudicial to the recognition. The 2D method may have more difficulties in using the x,y changes and the depth changes in the same way, even if the data contains the same amount of information.

Another interesting point to this study is that the best techniques at recognising the face when only one scan per person has been used during the training are all based on the curvature and normals. The Shape Index, Principal Direction and Surface Normal seem to be the more discriminative when very little training is allowed. But these techniques are very simple (a basic sum of L1 distances between corresponding points) and do not compete with the ICA and NMF when the training set gets bigger.

Table 2.3: Details of the tests E1, E2, E3 and E4.

	Training samples per subject	Number of subjects	Training scans	Test scans
E1	1	195	195	659
E2	2	164	328	464
E3	3	118	354	300
E4	4	85	340	182

Table 2.4: Comparison of methods and data representation performed by the same team on the FRGC v1.0 database. The three best results are highlighted for each of the tests. The test E1 to E4 are detailed in table 2.3. Extract from [Gokberk et al., 2008].

Representation	Method	Dimensionality	Performance			
			E1	E2	E3	E4
3D Point Cloud	Coordinates(x,y,z)	49.680	87.71	94.68	97.92	98.90
	ICA	90	85.66	<b>98.71</b>	<b>99.67</b>	<b>99.89</b>
	NMF	90	85.13	<b>97.77</b>	<b>99.25</b>	<b>100.00</b>
	Surface Normals	49.680	<b>89.07</b>	96.84	98.92	99.45
Depth Image	Pixel	90.201	55.99	70.19	79.75	87.69
	DCT	49	78.53	<b>97.63</b>	<b>99.58</b>	<b>99.78</b>
	DFT	49	75.95	97.13	99.08	99.56
	ICA	80	72.46	96.55	98.92	99.01
	NMF	70	71.55	95.83	98.67	99.67
Curvature	Shape Index	16.560	<b>90.06</b>	96.55	98.67	99.34
	Principal Direction	99.360	<b>91.88</b>	97.13	99.08	99.45
	Mean	16.560	87.41	95.69	98.50	98.90
	Gaussian	16.560	84.37	93.89	97.25	98.46
3D Voxel	DFT	53	64.26	91.16	97.92	99.34
Texture	Pixel	90.201	64.04	77.16	84.33	92.53
	Gabor	35.480	74.73	87.36	91.92	96.26

### 2.1.3 Insights from Human Psychology Research

Face recognition by humans has its advantages and drawbacks. But even if the human senses should not always be mimicked when designing their computer counterparts, some studies of them can give us interesting clues on what is possible and what is not.

One thing that is almost always true in nature is that a simple solution is selected unless a good reason prevents it.

**The use of the internal features** A good example can be seen in [Ellis et al., 1979] where they have shown that humans use a combination of internal (the inner part of the face) and external features (the shape of the face, the hair) to recognise unfamiliar faces, but



Figure 2.21: Example of internal and external features. From [Ellis et al., 1979].

that internal features are far more important to help recognise familiar faces (see figure 2.21). This difference shows that when more expertise on faces is needed, inner features are the best candidates. It justifies why the inner part of the face is used in our face processing systems. Another good reason (probably correlated to the first) is that outer features are often subject to variation (hair cut, hair colour, beard).

**Featural and configural clues in human vision** Another interesting finding is that featural and configural information are not necessarily both needed to recognise a face. [Schwaninger et al., 2002] have shown that humans are able to recognise people using either featural or configural information of the face. Their experiments consisted of a modified image recognition task (see Fig. 2.22). The first set of images contained scrambled features of 2D faces. Participants that tried to recognise these faces got good results while only using featural information. In order to eliminate the information that allowed recognition of these faces the experimenters applied a blur on the scrambled faces such that the recognition rate became almost null. They then unscrambled the faces while keeping the blur. Again, participants reached high recognition rates, now using mainly configural information. This experiment justifies the idea that both featural and configural information contains discriminative cues about people's identities.

**Haptic processing of the face** One may argue that looking at 3D face recognition without visual texture is problematic because 3D models may not contain enough information. However, studies of the haptic abilities of humans show that geometry of the face can be

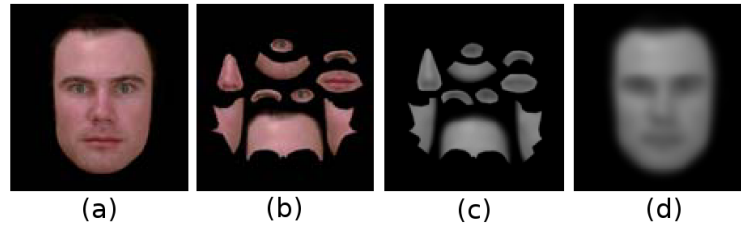


Figure 2.22: Example of stimuli used for the experiments.

- (a) Original picture
  - (b) Scrambled face - High recognition rate
  - (c) Scrambled and blurred face - Very low recognition rate
  - (d) Blurred face - High recognition rate
- Extracted from [Schwaninger et al., 2002].

sufficient. Both [Kilgour and Lederman, 2002], and [Lederman et al., 2007], show that untrained people can recognise unfamiliar faces using only casted masks of the faces. The recognition rate obtained in these experiments are a little lower than ones using live faces (where texture and temperature can help) but remains well above chance. We can expect trained people, like blind people for example, to get an even better score in that kind of experiment.

**Caricature** Studies have shown that humans do not necessarily remember faces as they are. Indeed people have a greater ability to recognise caricatures of famous people than to recognise original pictures of the same people [Lee et al., 2002], [Lewis, 1999]. Other studies have shown that if caricatures are used in the learning part instead of the veridical faces, then participants will better succeed in recognising new pictures of those faces. It is called the Reverse-Caricature effect [Rodríguez et al., 2008]. The same idea can be used with computers and can help detect what the features are on one face that differ most from the mean face (see figure 2.23).

## 2.2 Feature Localisation

The face can either be considered as a whole or as a set of features. Feature localisation consists of finding specific locations on the face which can be repeatably localised on all faces or on all captures of the same person. The information that can be extracted from the features can be used to register, to recognise faces, or to select areas of interest for further



Figure 2.23: Original face (left) and caricatures using landmark displacement from an average face with texture remapping. Extracted from [Craw et al., 1999].

processing. The vast majority of face recognition techniques use feature localisation at some point for one of these tasks. There exists plenty of 2D techniques for feature detection but we will focus here mainly on 3D methods not previously explained in 2.1.2.4. The reader can refer to [Cristinacce, 2004], for an in-depth presentation of feature detection methods on 2D textures.

The main uses of feature detection are:

- Registration: For example, when using a 2D PCA method for matching, the two first causes of variability between models, before light condition or identities, are the scale and orientation [Turk and Pentland, 1991b]. Therefore, great attention should be given to the 3D registration in the pre-processing, especially if a 2D matching technique is used afterwards in the process.
- Feature-based Recognition: The information attached to each feature or to the relations between features can contain discriminative information about the identity of the model and therefore can be used for recognition purposes.
- Region-of-interest segmentation: Sometimes landmarks or curves can be detected only to determine region on the input. For example, many methods crop the face using a sphere centred in the tip of the nose. Other techniques try to detect only local areas (nose, forehead, eyes) [Chang et al., 2006], [Mian et al., 2007b].

Feature localisation is a component of almost all automatic face processing system. Some people might argue that this is not true for holistic technique working through global registration (e.g. ICP) but in practice they never work without a pre-registration or the detection of a common point on the two objects (e.g. the nose tip). To our knowledge, the only case



where these methods work without pre-registration is when the input data is very clean. For 3D face data, obtaining a clean segmentation without detecting features can only be done using skin colour segmentation. In all other cases and for all other techniques, the localisation of common features on the query scans is obligatory.

In this section, techniques for feature localisation on 3D face data are reviewed. Examples of unlabelled features detection on other classes of 3D objects are presented in a second part.

### 2.2.1 Feature Localisation on 3D Faces

#### 2.2.1.1 Landmarks

Landmarks differ from simple points in the fact that they have a high-level definition. It can be a name or a specific property, for example “tip of the nose” or “left corner of the mouth”. This implies that if the same landmarks are detected on two similar objects an implicit correspondence between that pair of points can be made. There are many types of facial landmarks that have already been defined. The best known are the 47 anthropological cranio-facial landmarks used by Farkas [Farkas, 1994].

For face registration, at least three landmarks are needed for alignment. Some of the methods are designed to detect only one kind of landmark. For example, there are plenty of methods whose only task is to detect accurately the tip of the nose [Mian et al., 2006b], [Yang et al., 2009]. For example in [Berretti et al., 2008], the tip of the nose is the only point detected and strips located at iso-geodesic distances from it are extracted and serve as descriptors of the face.

In 3D, many techniques use curvature to detect or select points of interest. In chapter 8 of [Hallinan et al., 1999], curvature is used to detect candidate points. Then, a symmetry matching process is used to determine the landmarks to keep. Bounding regions are also determined using ridge lines.

In [D’Hose et al., 2007], Gabor wavelets are used on a projected curvature picture along two directions (vertical and horizontal). The combined responses give good coarse indicators of the location of several landmarks (tip of the nose, left/right upper/lower corners of the nose). Then, the ICP method is used on local parts of the face to get more precise landmark positioning. The entire algorithm takes an average of 16 seconds to detect 7 landmarks.

[Nair and Cavallaro, 2009] use a 3D Point Distribution Model (PDM) which is quite similar to what [Cootes and Taylor, 1999] used in 2D to detect the position of the landmarks. Three landmarks (tip of the nose and inner corners of the eyes) are used to start the registration. They are detected by finding a matching triangle among a set of preselected points found using curvature (shape index and curvedness index).

In [Segundo et al., 2007] profile and curvature are used to detect the nose (tip, corners and base) and the inner corner of the eyes. The feature detection time is about 0.4 seconds with a detection rate of more than 99.7% on the FRGC v2.0 database. The paper does not precisely indicate what a “correct detection” is. However, it explains that a few misdetections occur on faces where some depth information is missing (for parts of the nose) and/or on some faces rotated more than 15 degrees from a frontal position.

In [Moreno et al., 2003], a combination of median and Gaussian curvature is used to detect features such as the tip of the nose, the ridge and sides of the nose, the upper bridge of the nose, the eye regions and the inner corner of the eyes. On the 420 image database with various expressions and orientations, the nose elements are detected with rates above 98% while the eye regions are detected with rates going from 83 to 94%.

In [Irfanoglu et al., 2004], a base mesh is created using hand-landmarked faces which are warped onto the mean landmarks using the Thin Plate Spline (TPS) method. The new face is registered to this base using the ICP method, and distance to the projection plan, symmetry plans, curvature and normal are used to detect precisely the 10 fiducial landmarks. A TPS is then performed to densely register the new face to the model. What is surprising with this method is that there is no pre-registration before the ICP which gives no solution in cases in which the ICP converges toward a local minimum.

[Lu et al., 2004] select only three points (depending on the view) using local curvature. These points are used for a coarse first registration. Then, an Hybrid ICP algorithm is used to refine it. In a following paper [Lu and Jain, 2006], they manually define 20 landmarks and automatically compute 74 semi-landmarks along the geodesic paths linking the manual landmarks (see figure 2.24).

In [Mian et al., 2006a] slices of the 3D face are used. A cubic spline is used to interpolate each profile and detect its intersection with the ridge of the nose using the mid point between points of maximum slope (see figure 2.25). The ridge of the nose is constructed from those

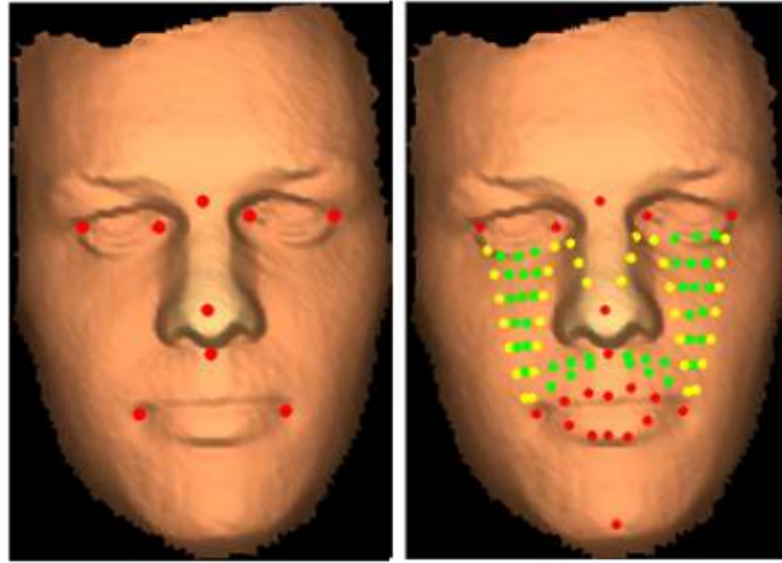


Figure 2.24: Hand placed landmarks (left). Same picture with the 74 semi-landmarks added (right). Extracted from [Lu and Jain, 2006].

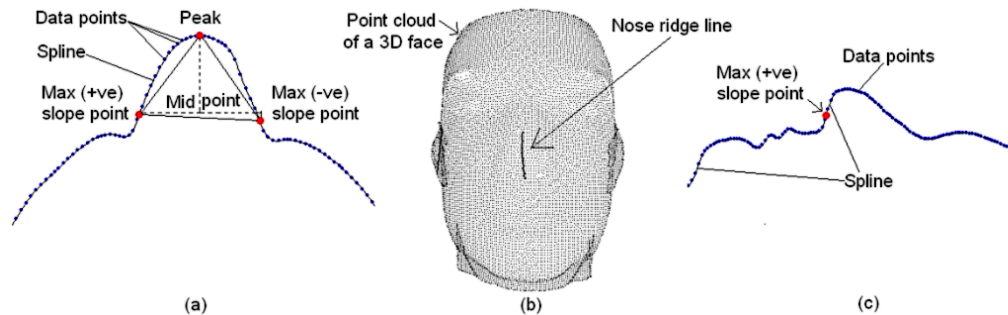


Figure 2.25: (a) Detection of the ridge of the nose point on each slice. (b) Detected ridge of the nose. (c) Profile computed with the ridge of the nose and selected landmark. Extracted from [Mian et al., 2006a].

points and is used to determine a vertical profile. In [Mian et al., 2006b], they use a similar technique to detect the tip of the nose via a coarse to fine technique consisting of taking the ridge of the nose point corresponding to the highest triangle. The separation of slices is first coarse and is then refined to detect the landmark precisely. The detection accuracy of the nose is 98.3% on the FRGC v2 database (only 85 failures out of 4950). The main causes for failure were attributed to hair and exaggerated expressions.

### 2.2.1.2 Histogram-Based Feature Detection

In order to detect points that have a similar local neighbourhood, histogram descriptors of these local areas can be used. An histogram descriptor should be robust in dealing with pose variation, i.e. the local basis in which it is produced should be determined using only local information (normal at the point, direction of greater gradient, and so on).

One of the most used 3D histogram descriptors is the Spin Image [Johnson and Hebert, 1999]. It encodes the local shape relative to a mesh vertex and its normal. In particular, it is a histogram of radius and height values, where the radius is the orthogonal distance to the normal, and the height is a signed distance, relative to the vertex, in the direction of the normal. The name “Spin Image” is used because we can visualise a gridded half-plane being rotated around the vertex normal and neighbouring vertices being accumulated in cells (bins) to form the shape histogram. In this sense, the cells of the histogram are analogous to the pixels of an image. The cells (“pixels”) are not required to be square and the cell size can vary from cell to cell, for example by following a log function. In this thesis, only fixed-sized cells are considered. The parameters for this descriptor are the number of radial cells, the number of vertical cells and the radial and vertical cell sizes.

In [Mian et al., 2007b], a simpler histogram descriptor is used. Instead of representing the vertices in cylindrical coordinates like the spin image does, they are represented in spherical coordinates and only the radius is used as a dimension for binning: the spaces between consecutive spheres centred on the point are the bins. This Spherical Face Representation (SFR) is a vector where the  $i$ -th value corresponds to the number of points intersecting the  $i$ -th bins defined from the tip of the nose. This histogram descriptor is only used to describe the nose tip. Figure 2.26 shows both SFR and Spin Image principles.

In [Pears, 2008] multiple-scale spheres are also used to construct a histogram descriptor but the values considered for each radius are the signed distance to the surface computed with a Radial Basis Function (RBF) over a discretisation of the inner ball. This Spherically-Sampled RBF (SSR) method succeeds in recognising the tip of the nose of 99.6% of the models in a database of 1736 3D images and gives better results than the Spin Image in the same test conditions.

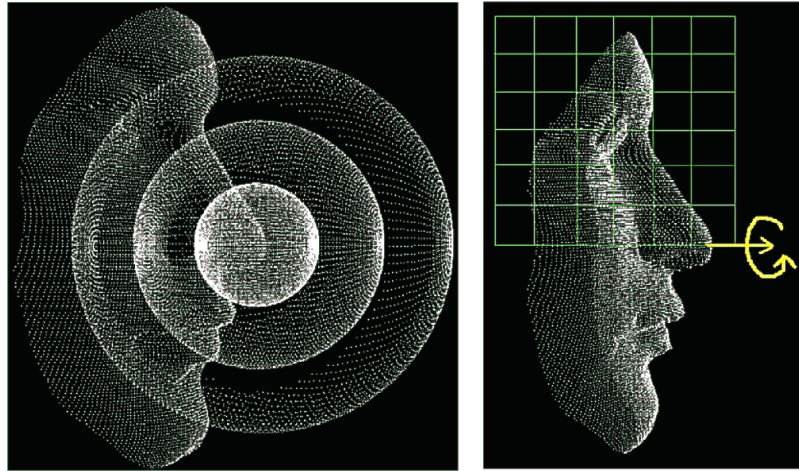


Figure 2.26: Principle of SFR (left) and Spin Image (right). Extracted from [Mian et al., 2007b].

The Point Signature by [Chua and Jarvis, 1997], and Curvature-based Keypoint by [Mian et al., 2007a], (both already described in section 2.1.2.4) are two other examples of how a local shape can be represented.

### 2.2.1.3 Line and Surface Detection

The information that is extracted from the face is either a point, a curve or a surface. Most of the techniques end with the determination of landmarks but this is not always required. For example, if you want to compare two noses, you may want to extract the entire convex part of the face which represents the nose. In this case, landmark extraction is not mandatory.

The techniques that do not use landmarks fall most of the time into two categories: segmentation and line detection techniques.

In 1991, [Gordon and Vincent, 1992] localised eye regions and the nose region by detecting areas surrounded by ridge lines using dilatation of the lines, merging of adjacent regions and other complex recipes (e.g. symmetry correspondence).

Some techniques try to localise features directly on range images. For example in [Suganthan et al., 2008], where the angle between surfaces (the dihedral angle) is computed to help determine the “edges” in the 2D depth map. In [Ben Amor et al., 2005], a watershed-based segmentation is used on the face to perform comparison by region.

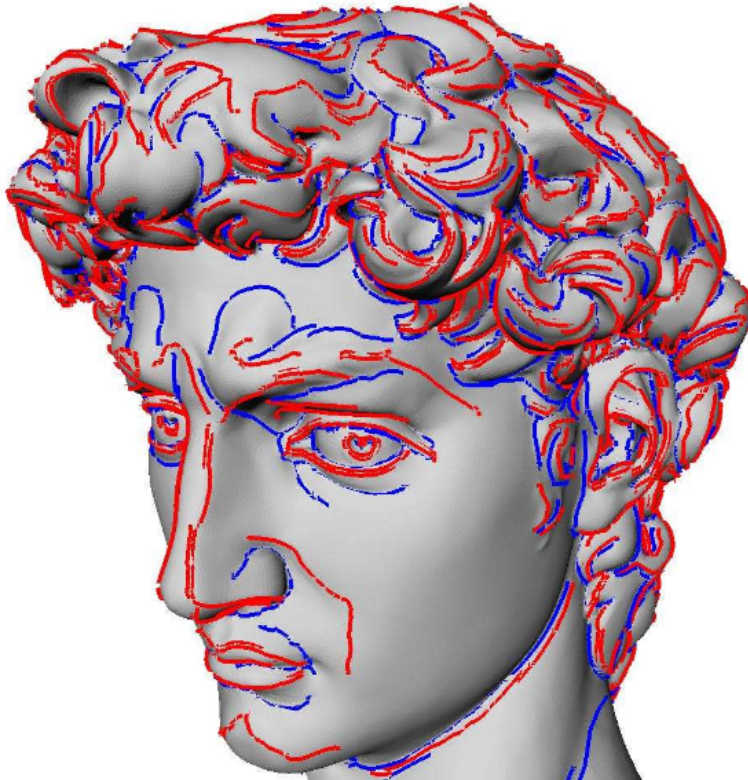


Figure 2.27: Valley (blue) and ridge (red) lines detected on Michelangelo's David. Extracted from [Pouget, 2005].

The best known curves that can be detected on 3D surfaces are the ridge lines (sometimes called crest lines). They correspond to extrema of curvature along principal directions (see figure 2.27). The valley and ridge lines detected correspond to the salient part of the object. The reader can refer to Pouget's thesis [Pouget, 2005] for more details.

#### 2.2.1.4 Summary

A typical multiple landmark detection approach is presented in [Colbry et al., 2005]. First, they pre-process the face to remove spikes before cropping the upper part of the scan as being the region of interest. Then the nose is localised as the closest vertex to the camera or the most extreme point in a particular direction (left or right) or the one with the largest shape index. The inner corners of the eyes are detected as the points with the smallest shape index. This kind of approach has a lot of variants and is widely used in both academic and commercial systems.



Other authors have noticed that the sagittal slice of the face remains identical over orientation changes and therefore can be used to detect the nose. In [Faltemier et al., 2008b], contours of the mesh are extracted at varying angles until it matches a previously learnt nose profile signature. The system achieved 98.52% accuracy for the nose tip with variations of angle of up to  $90^\circ$ . Some non-pose-invariant techniques have also used transverse slices to detect the nose tip and the nose corners [Segundo et al., 2007] [Mian et al., 2006b].

To summarise, most papers on 3D face landmarking have their keypoint detection system grouped in one of the following categories:

- Curvature/Volume Extrema: The candidates are defined as extrema over curvature and/or volume based descriptor maps [Chang et al., 2006] [Colbry et al., 2005] [D'Hose et al., 2007] [Pears et al., 2010] [Romero and Pears, 2009a] [Segundo et al., 2007] [Szeptycki et al., 2009].
- Directional Extrema: The candidates are defined as the extremal points in given directions [Chang et al., 2006] [D'Hose et al., 2007]. This is only used for nose tip detection.
- 2D Curve Extrema: By using profiles and slicing of the mesh, the detection of salient points is reduced to finding extremal points along a two-dimensional curve [Faltemier et al., 2008b] [Mian et al., 2006b] [Segundo et al., 2007].

Several previous studies have acknowledged the limitations imposed by heuristic approaches and have employed machine learning techniques instead [Berretti et al., 2010] [Zhao et al., 2011]. However, they usually employed 2D descriptors on depth-maps, making their systems unusable in scenarios presenting a large rotation from the frontal view. To the best of our knowledge, it appears that no 3D machine learning method exists for facial landmark candidate (keypoint) detection (see Fig. 2.28). This is a gap in the literature that we aim to fill, to enable better landmarking on face scans of non-cooperative subjects. Our proposed system is sufficiently generic to be applied to meshes of other general classes of objects in any application where landmarks of interest can be manually defined on a set of training scans.

### 2.2.2 Feature Localisation on 3D Meshes

Computing keypoints in order to determine correspondences is useful for all kinds of object matching applications. In [Mian et al., 2010], keypoints are computed using a coarse curvature descriptor to localise objects in scenes with occlusions. In [Zaharescu et al., 2009], an approach called Mesh DoG is presented. This is a multi-scale approach that makes use of Difference-of-Gaussians (DoG), and thus has similarities to the DoG approach applied to 2D images in the SIFT descriptor [Lowe, 2004]. In this approach, any surface descriptor map can be convolved with a set of Gaussian kernels of different scales (standard deviations). Subtracting convolutions across two adjacent scales gives the DoG operator response. Keypoints are then extracted as the local maxima across scale space, using non-maximal suppression in a one-ring neighbourhood in the current and adjacent scales. A similar DoG-based approach is presented in [Castellani et al., 2008]. However, here the DoG operator is applied to the actual mesh over a range of scales. The amount that a vertex moves between Gaussian filtering at one scale and the next is projected along the vertex normal. Keypoints are extracted as points of maximal normal movement over local neighbourhoods and local scales. In [Itskovich and Tal, 2011], two kinds of curvature-related descriptor (Shape Index and Willmore Energy) are combined to detect the keypoints on archaeological objects in order to detect regions matching a given pattern. This last paper is one of the rare cases in which more than one descriptor is used for the keypoint candidate selection. Another example is [Dibeklioglu et al., 2008], in which the Shape index, the Difference Map, and the Gradient of the image are combined for landmark localisation. Besides, when several descriptors are used, combining them is usually done using fixed coefficients. In this thesis, a framework is presented to determine automatically how descriptors should be combined for the particular problem of 3D face landmark candidate detection.

## 2.3 Conclusion

While very good face recognition results can be achieved with existing techniques, some drawbacks are still present and a lot of problems remain unsolved. The difficulty is that often the problems are not defined: we know that the result is not good but there is no easy way to find what the causes are.



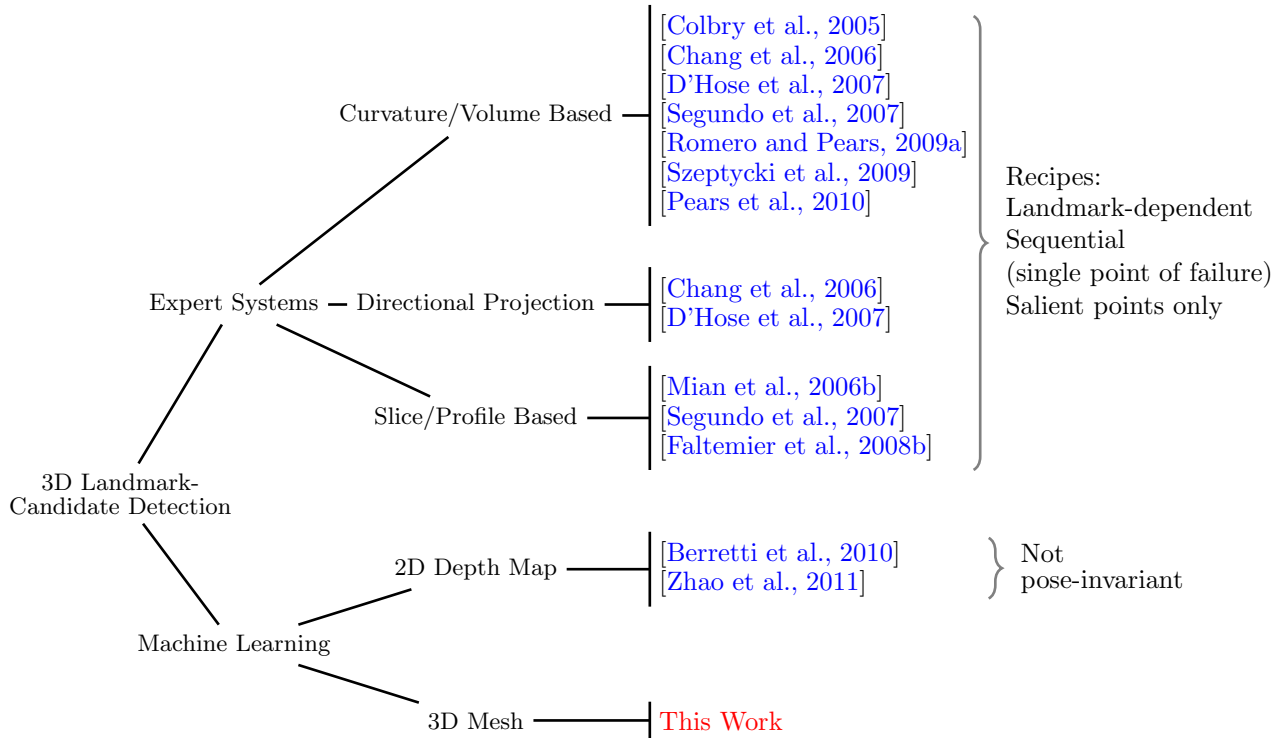


Figure 2.28: Related work: Facial landmark candidate detection usually appears as a single section in face landmarking papers. Most of them are sequential recipes. Some use machine learning techniques but, in these cases, they are always used with 2D representations (e.g. SIFT [Lowe, 2004] on depth maps).

Here we recapitulate the known (or assumed) problems that undermine state-of-the-art face recognition.

### 2.3.1 Input Data Problems

**Capture problems** Capturing data is not a transparent process and the quality of the 3D Model should not be assumed to be perfect. In addition to classic noise, some artefacts can appear like peaks due to reflection of the skin or accessories, or distortion linked to motion of the face during the capture.

**Extra data** Even if the capture system were perfect the face can contain elements that vary too much to be considered in the matching process (for example glasses, hair in front of the face, beards, a hand and mobile phone in front of the cheek, and so on). Some techniques try to deal with this problem (PCA methods can accept glasses and beards and still work

reasonably well). Others try to erase such information in pre-processing using skin detection methods for example.

**Missing data** The way the 3D model is captured can produce self occlusions that lead to holes in the surface (very often around the nose region). For example, when the face is captured in profile, half of the face is missing from the input. Therefore, the system should be able to deal with a large amount of missing data.

### 2.3.2 General Problems

**Automation** Most of the best techniques are not fully automatic and usually the part which is manual is the localisation of landmarks [Gupta et al., 2007b]. Making these systems automatic is often very difficult because they have been designed to use landmarks that are easy to locate for a human operator but not necessarily so for a machine.

**Problems attached to 2D techniques** Acquiring 2D images has the advantage of being cheap and easy. However, while 2D pictures contain enough information for recognition, dealing with this all-in-one data representation is quite difficult. The two kinds of information that are generally considered (the texture and shape of the face) are mixed with illumination, makeup, accessories, position and background information. Methods exist to deal with most of these problems but issues like the position of the face are still very difficult to handle. It requires either the use of a 3D model to register the 2D picture [Blanz and Vetter, 2003], [An and Chung, 2008], or to have a large set of images for each individual.

Even a featural method like the Elastic Bunch Graph Matching [Wiskott et al., 1999], suffers from these problems and gives a recognition rate of under 20% for frontal to profile experiments.

A recent paper [Amberg and Vetter, 2011] provides a technique similar in many respects to the work presented in this thesis but for 2D data. Their approach is to split the landmarking problem in a keypoint detection and a labelling problem. The detection is performed using a decision forest on small patches to select interesting points (1-3%). The points are then associated with label candidates using a 2-class LDA-like method (where the two set are different landmark classes). For the labelling disambiguation, they use a “branch and bound”

approach with a cost function based on a learnt shape model. They achieve good results with this machine learning approach but the choice of the “branch and bound” technique makes the success rate drop significantly in case of missing points. A RANSAC approach is likely to have been more robust to missing data often observed in non-cooperative cases.

**Dense registration problems** A dense registration is achieved when every point of a first model has a corresponding point in a second model. If all the database is densely registered on one defined model then every 3D point of any of the faces has a label (the index of the corresponding point in the model). This can be very useful but some problems exist:

- The registration is a global process, which means that some parts can be better registered than others, depending on some design choices.
- The dense registration requires clean data. If hair, beards, or accessories are present, they will affect the registration.
- A fine registration is most of the time very computationally expensive.

A technique that needs a registration will have to deal with these problems and accept the error that they can introduce.

**Same face variability** A main problem with faces is that they change. Some changes are quite subtle (e.g. ageing); some are more obvious (expressions). To deal with the problem of expression, several approaches have been proposed:

- Expression invariant models (e.g. [Bronstein et al., 2007])
- Morphing to neutral before matching (e.g. [Hsieh et al., 2009])
- Multiple expression gallery

It is of course a problem for real life face recognition but usually facial expressions are not kept for very long. If the system can capture the face over a reasonable period of time it should be able to identify the individual. Testing the recognition ability discarding expression variation can be done easily as almost all the available databases specify the expression in the meta-data.

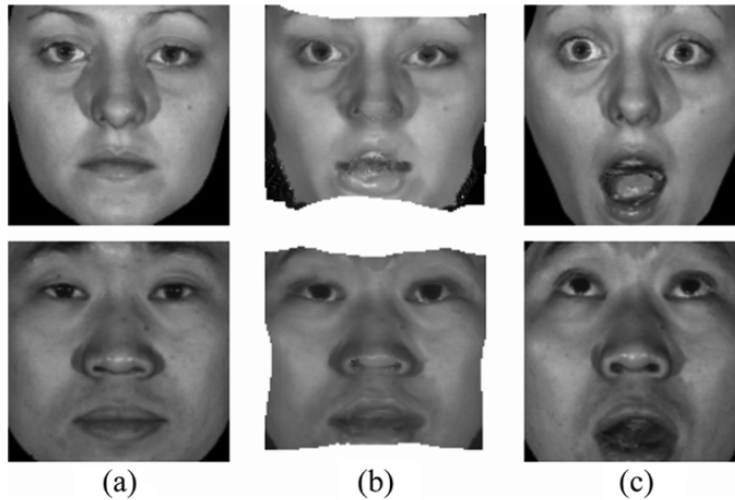


Figure 2.29: Example of expression morphing to neutral. (a) neutral picture for comparison, (b) neutralised picture, (c) picture to neutralised. From [Hsieh et al., 2009].

**Non-cooperative real life conditions** The main problem linked to non-cooperative face recognition is not faced in most of the papers published in this area because no non-cooperative real life database is available to test their methods. [Medioni et al., 2009] try their system in real life conditions. Long distance (3,6 and 9m) non-frontal video camera captures are used for 2D and 3D recognition. In a first step low resolution images are used to localise the face. Then high resolution images are extracted and used to construct a 3D model of the face. The model is then registered in a frontal position. Finally 3D and 2D-texture face matching are performed using commercial solutions. The recognition rate seems to be low compared to that obtained with the FRGC databases but the task is harder with this database, which contains only non cooperative captures taken at a distance. At some point, real non-cooperative 3D databases will be needed to allow progress on this matter.

### 2.3.3 Gap in the Research Literature

What we can conclude from this literature review is that one of the biggest challenge for face recognition techniques in the years to come will be to cope with variations that are under-represented in current research databases, where all subjects are cooperative. Dealing with non-cooperative subjects is a crucial step in order to transfer technology from research labs to real life situations where humans are busy and should be recognised without having to

pay attention to the machine and the position of the sensors. Examples of non-cooperative face processing scenarios are :

- Surveillance: For example, people walking in a public area checked against a watch-list.
- Robotics: A mobile robot which has to find someone in the room. The person should not have to look at the robot. Recognition should be possible when people are busy or even unconscious (e.g. robots for elderly care).
- Driver attention monitoring: The system will be looking for signs of drowsiness on the driver's face. This can be difficult when the face is not in a frontal position.
- Game player recognition: with new 3D sensor hardware like the Kinect camera from Microsoft, game players need to be recognised in all sorts of positions.

In the face recognition process (see Figure 2.4), both acquisition and matching stages are likely to remain unchanged whether the subject is cooperative or not. However, the feature extraction process is very likely to change. The pre-processing of the data for face recognition in non-cooperative cases is one of the bottlenecks that will have the greatest effect on the overall robustness of the framework. Indeed, most automatic methods make strong assumptions about the input data (mainly its orientation) that will make the initial feature detection and registration fail in most uncooperative cases.

Acknowledging this, we focused our effort on the detection of facial features in 3D data with the objective of being able to deal with non-cooperative cases. We constrained our research to 3D data only because it is where the gap in research seems to be the biggest.



## Chapter 3

# Detailed Strategy

Strategy without tactics is the slowest route to victory.

Tactics without strategy is the noise before defeat.

---

Popular quote often attributed to Sun Tzu, Chinese General and Philosopher,  
b.500 BC. No written records found.

In the literature review, it has been seen that one of the major limiting factors for non-cooperative automatic processing of the face is the localisation of its features in non standard situations where pose variation, occlusions, and the presence of non-face objects in the scene can occur (See Figure 3.1). Most face recognition systems are oblivious to these issues as the databases used for testing are usually the ones designed for face verification in which the subjects are usually cooperative.

In this chapter, the high level set of the decisions that frame our research project will be explained and justified. The approach taken to improve feature localisation on 3D faces is discussed and the pipeline of the whole process is presented. Our overall problem is divided into sub-problems on which performance can be measured independently from the whole framework's cumulative errors. The choice of databases, the modalities in performance measurement and the priorities between different aspects of the system are also discussed.

### 3.1 Preliminary Choices and Justifications

In this section, justifications for some of our major high level research choices are summarised.

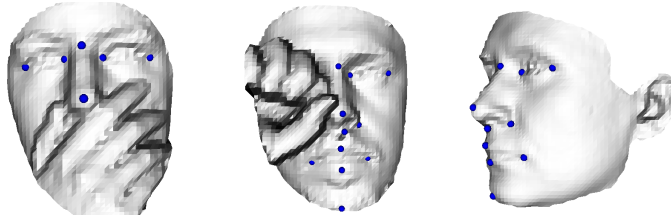


Figure 3.1: Example of landmark localisations (on models from the Bosphorus database) for which no robust automatic method currently exists.

**Why faces?** Most databases of 3D objects contain objects of different classes (coffee mugs, toys, tools, components, and so on). The number of objects per category is usually less than a few dozen. However, 3D face databases usually reach a few thousand captures. This makes faces (as an object class) the single category with the largest number of available samples for researchers. Furthermore, the natural variations inside that category (identities, sex, age, origins, expressions, and so on) are enormous, making it a challenging subject for research.

An incentive to the study of faces is also the wide range of possible applications in both industry and academia. Whether it is for surveillance (3D CCTV), human-machine interaction (especially for autonomous robots interacting with humans), anthropology or psychology studies. Our modern age has been build on automating repetitive tasks once performed by humans. To continue in that way, it is now required to mimic higher level skills of the human brain like face processing.

**Why treat faces differently from other objects?** A face is just another material object and yet it stands aside. Everything seems to indicate that faces need to be treated by specialised systems. Our main evidence for this is the human brain. While it is still controversial whether humans evolved a specific region of the brain (namely the fusiform gyrus) for face recognition it is agreed that this area is a “specialised” region in vision mainly used for face recognition. As so often in neurology, this was first observed due to brain injuries in that region leading to a condition called “prosopagnosia” (or face blindness) [McKone et al., 2006]. People suffering from this condition could no longer recognise faces, while keeping an undiminished ability to recognise other objects. This suggests that faces might be more difficult to recognise than other objects. It is why a lot of energy is spent on designing specific and specialised systems for faces. However this justification holds only for



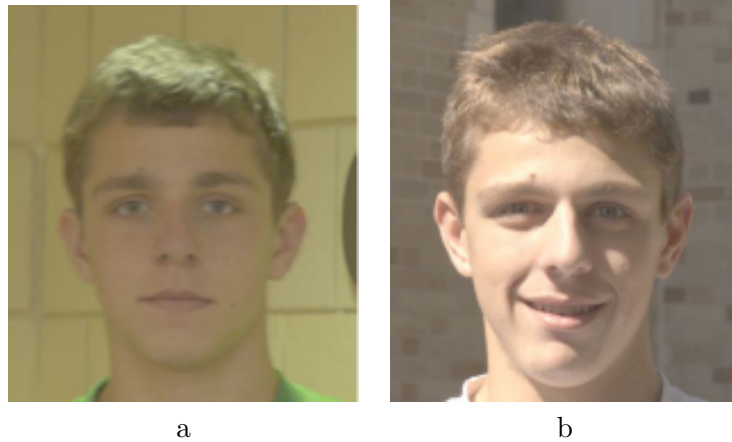


Figure 3.2: Limitation of 2D face recognition systems. Two captures of the same individual from the Multiple Biometric Grand Challenge Database (MBGC [Phillips et al., 2011]). This case is particularly difficult for current face verification systems. There is no certainty that the problem is solvable with these inputs as even a human might not be sure of his/her decision.



Figure 3.3: Example of difficulties for 2D face recognition systems: variations in pose, illumination and make-up/tanning/facial-hair/accessories. From [Liu et al., 2007].

face recognition and not for feature localisation on faces. It is why our framework does not make any assumptions about the class object it deals with and can be used for objects other than faces.

**Why 3D over 2D?** 3D sensors are usually expensive, noisy and low resolution compared to 2D sensors. So why expend so much effort on 3D processing? The reason is that 2D has inherent drawbacks. These start to appear as the datasets become bigger and less constrained. In figure 3.2 a practical example of such limitation is shown.

Changes in orientation, illumination, make-up and tanning (see Figure 3.3) put 2D recognition system in a very difficult position, up to the point where is it no longer known if the problem is solvable. The advantage of the 3D data is its robustness to most of those changes.

**Why not use texture in addition to 3D shape?** A good criticism of our approach is to say that while uncooperative face recognition needs 3D data, feature localisation might not need it and might always be performed better in 2D than in 3D. This is a very good point from an engineering perspective. From a research point of view, the challenge is to close the gap in performance between 2D and 3D landmarking systems. When this is done, and when acceptable localisation can be done with only 3D information, combining 2D and 3D will probably be the way forward for real-world applications when feature detection on 3D models is required.

**Why focus on feature localisation?** As seen in the literature review, feature localisation is a bottleneck. Every comparison of 3D structures needs a pre-registration or a set of initial correspondences. It is a challenging problem with numerous potential applications.

**What precision is targeted?** Finding an approximate position and finding the precise local position are two different problems. These two different problems are very likely to require inputs of different resolution, and while the coarse localisation requires both local and structural information, the precise localisation might only require local information. We argue that refining the local position of a labelled landmark is not as problematic as finding its global position in a scene.

**What priorities should be set between the different objectives?** There are intrinsic limits to how much an existing technique can be sped up. It is why it is important to take speed into account at an early stage in the conception of a method. As a rule, we always try to design methods that can run at under one second per face. A non-optimised method running in a few seconds is also acceptable if evidence shows that there is still room for significant speed improvements. Another obvious reason to avoid computationally expensive methods is that they are very difficult to evaluate as the smallest change of parameters on a big database will lead to a long wait before any feedback is available.

## 3.2 Problem Statement

The main problem that we address is *the localisation of zero-dimensional features on 3D face surfaces* (see Figure 3.4).

To simplify the problem, an additional condition is enforced: the selected points should be a subset of the input vertices. This can be justified in our case by the fact the resolution of the input is good enough compared to the acceptable error in positioning. Sub-pixel localisation of landmarks is an interesting problem for landmark refinement techniques but is not discussed in this thesis.

Let  $V$  be the set of vertices of the input mesh and  $V_i$  the  $i^{th}$  vertex in this set. Let  $M$  be the set of target labels and  $A_{M_i}$  the set of acceptable vertices for label  $M_i$  (based on a ground truth).

The problem of complete correspondence is defined as:

$$\exists S \subset V, \quad \exists \Lambda : S \rightarrow M \text{ bijective} \quad s.t. \quad \forall t \in S \quad t \in A_{\Lambda(t)}$$

In case of incomplete matching, we search for  $\Lambda$  injective <sup>1</sup>.

**Problem Breakdown** Presented in this form it appears that this problem can be decomposed in two sub-problems:

- Finding the subset  $S'$  - point positions.

$$\exists S' \subset V, \quad \forall t \in S', \quad \exists \lambda \in M \quad s.t. \quad t \in A_\lambda$$

- Finding the mapping  $\Lambda$  - point labels.

$$\exists S \subset S', \quad \exists \Lambda : S \rightarrow M \text{ bijective} \quad s.t. \quad \forall t \in S \quad t \in A_{\Lambda(t)}$$

Solving the problem requires finding  $S'$  a set of vertices and  $\Lambda$  a mapping to known labels. Of course the two are intimately linked and looping between the two might be necessary as

---

<sup>1</sup>  $\forall a, b \in S \quad \Lambda(a) = \Lambda(b) \implies a = b$



Figure 3.4: Problem: From a 3D mesh alone, detect the subset of vertices corresponding to 14 defined landmarks.

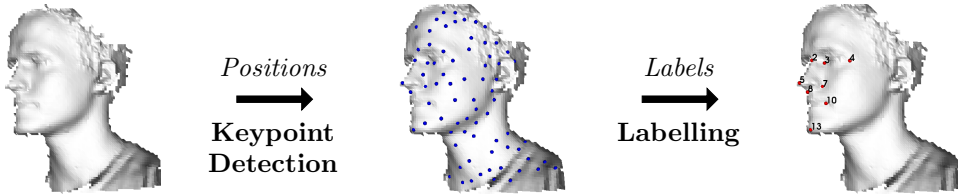


Figure 3.5: Problem Breakdown: the problem is split into two sub-problems that are solved independently. Keypoint detection and point labelling.

$S'$  should be big enough in order to contain  $S$  and small enough so that the labelling can be found in a reasonable amount of time.

Our starting point is to reduce the number of input vertices using local information before running structural matching techniques to select the final positions and labels. Our two requirements are therefore transformed into finding:

- a small set  $S' \supset S$  (wanting it small is just a matter of computational cost).
- and a mapping  $\Lambda'$  from  $S'$  to  $\{M \cup \emptyset\}$ .

The first one consists of finding good landmark candidate positions: it is a keypoint detection process. The second consists of finding a correspondence between a known structure and the query: it is a labelling process.

**Keypoint detection** When selecting the set  $S'$ , two opposing objectives are to be considered:

- $S'$  should be small (to reduce the search space to a manageable size).
- $S'$  should be big enough so that it is likely to contain a good proportion of good candidates (with regard to the ground-truth)

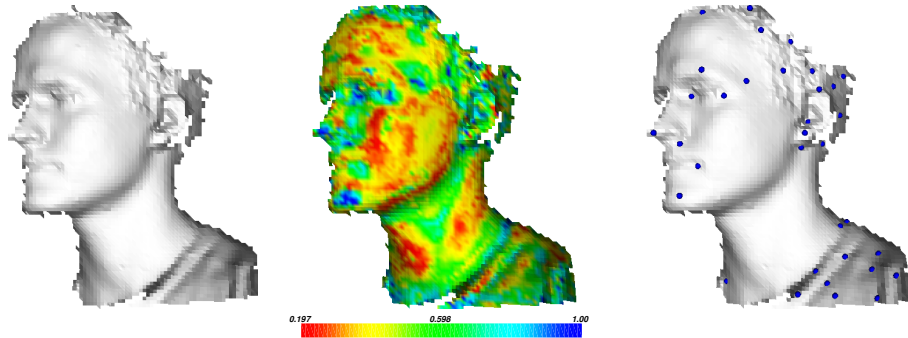


Figure 3.6: Keypoint detection example (cross database experiment). Input mesh (Left). Final score map (Center). Detected keypoints (Right).

The aim is therefore to detect keypoints that have a high probability of being labelled as landmarks later in the process. To do so, our best strategy to construct  $S'$  is to learn what landmarks look like locally and try to detect similar patterns in the query mesh.

Our first idea was to detect keypoints using local extrema of descriptor maps and to combine these using clustering. This approach gave interesting results but was discarded later on because of the arbitrary nature of selecting extrema (see Appendix D). Our second and retained idea was to compute score maps using a dictionary of local shapes and to combine these into a final map representing, at every vertex location, the likelihood of being a landmark (see Figure 3.6).

**Labelling** The labelling process consists of assigning labels to the input points. This process can use both local and structural information to do so. The two more common approaches are to use a bag-of-feature representation (when only local information matters) or a graphical representation (when local and pairwise structural information matters). We make the strategic choice of going beyond the graphical representation and using hypergraphs so that relations of any degree can be used to extract the correspondence. Our first idea was to extend the heuristic of *relaxation by elimination* to hypergraphs. While our idea of using hypergraph was, to our knowledge, original at the beginning of this PhD, several other researchers have published hypergraph matching related papers since (namely [Zass and Shashua, 2008], [Duchenne et al., 2009] and [Chertok and Keller, 2010]).

### 3.3 Decomposition to Sub-problems of Varying Difficulty

Unlike a lot of pattern recognition techniques, some of our techniques (especially the hypergraph matcher by relaxation) are “white boxes”: At every step, every decision made by the algorithm for keeping or rejecting candidates can be seen and explained. This is both a great advantage and a great drawback, as it is easy to lose oneself in the study of every unexpected behaviour of the system. A simple protocol to solve the problem would be to run the system, locate the errors, understand the reasons for those failures, adapt the system, and continue. However this approach can lead to loops in the development of our algorithm as by optimising the code for a particular problem one might reduce the performance for another set of data.

In order to avoid these never ending cycles, our main problem is split into smaller sub-problems of increasing complexity. By solving these problems in turn the risk of ‘early optimisation’ which is often responsible for those loops in the development cycle is limited. This approach can also help us categorise the behaviours of different techniques according to the imperfections observed in the input data. To do so, four kinds of input are considered for the labelling process (see Figure 3.7):

- **Synthetic data:** By generating pairs of hypergraphs with known correspondences and known imperfections (noise, number of missing point and number of additional point) the methods can easily be compared without being problem specific.
- **P1:** When trying to find landmarks on faces, an interesting input is to provide only hand-placed landmarks deprived of their labels to the system. If a method cannot replace the labels in those conditions, it could not do it with more realistic inputs. This also allows us to observe how face variation influences the matching (in terms of occlusion and expression). As all visible landmarks are present in the input, the maximum retrieval rate possible with this problem is 100%.
- **P2:** A more complex problem is then to look at inputs in which all detectable landmarks are present but hidden in a forest of other points, all statistically likely to be landmarks. This allows us to evaluate the candidate elimination capability of the

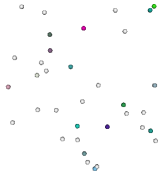

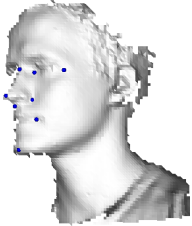
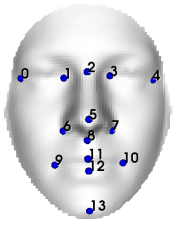
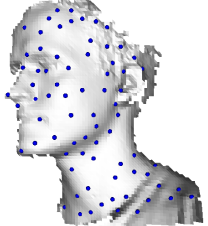
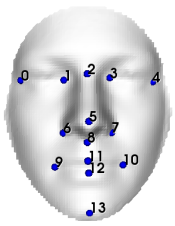
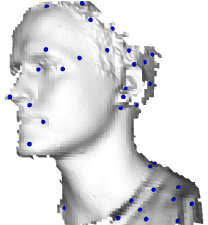
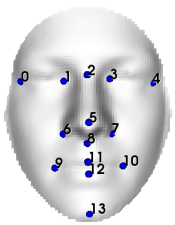
	Query	Model	Variations
Synthetic			<b>Noise:</b> Configurable <b>Missing:</b> Configurable <b>Extra:</b> Configurable
P1			<b>Noise:</b> Face variability + Human landmarking error <b>Missing:</b> Occlusions <b>Extra:</b> None
P2			<b>Noise:</b> Face variability + Human landmarking error <b>Missing:</b> Occlusions <b>Extra:</b> Dense automatic keypoints
P3			<b>Noise:</b> Face variability + Detection error <b>Missing:</b> Occlusions + Detection false negative <b>Extra:</b> Detection false positive

Figure 3.7: Different kinds of problems to evaluate the labelling process.

system in tougher conditions. As all visible landmarks are present in the input, the maximum retrieval rate possible with this problem is 100%.

- **P3:** To test the landmarking framework as a whole, the automatic points (as returned by our keypoint detector) need to be labelled. The keypoint detection being imperfect the maximum reachable retrieval rate will be lower than 100% for the labelling process.

**Sub-problem independence** In terms of methodology, each proposed method should be evaluated at least once without taking into account the pipelining cumulative errors induced by our framework. For example, if a graph matching technique is used on an input generated by our keypoint detection it would be impossible to say whether the measured errors resulting

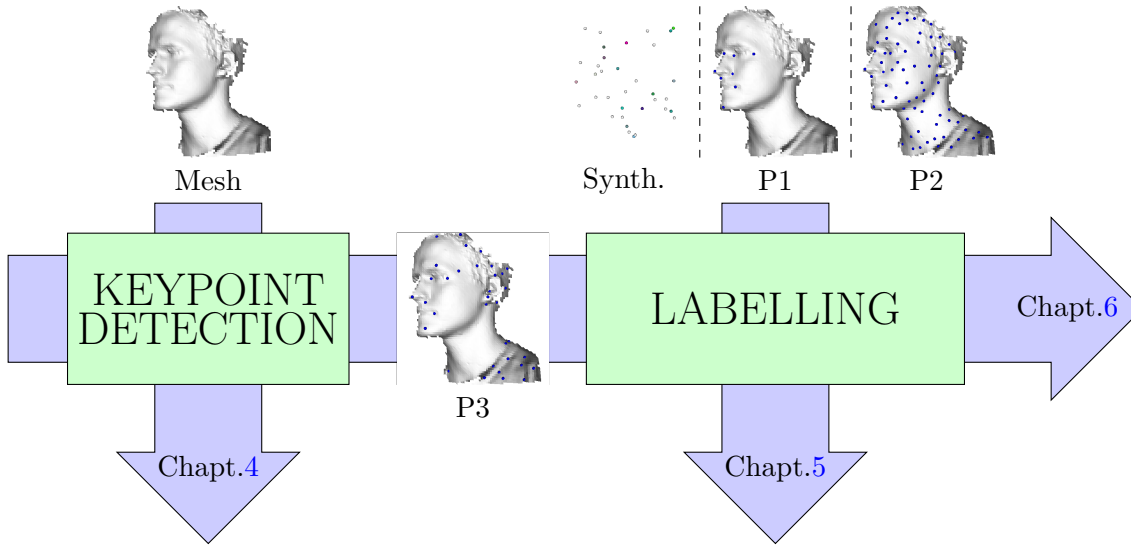


Figure 3.8: The two main components of our framework are first evaluated independently (no error accumulation due to pipelining). The whole landmarking framework is tested and optimised in a following chapter. In addition to synthetic data the labelling process will face three main kinds of unlabelled input data: sparse manual points (P1), a dense mixture of manual and automatic points (P2) and automatic points as returned by the keypoint detector (P3).

from the experiment are intrinsic to the method or induced by the spurious input data. To evaluate the methods on their own the inputs should always be good enough so that a success rate of 100% is reachable. When this is not possible, a baseline should be provided to indicate what the maximum expectable results would be. In many cases, human performance can be used as the baseline. Figure 3.8 shows how the components of our framework are evaluated independently before looking at the global framework performances.

### 3.4 Datasets

In order to test our ideas, two publicly available 3D face databases are used: the FRGC v2 and the Bosphorus databases. These two are used throughout the thesis and are presented here. Both have a little fewer than 5000 faces. The FRGC has more variation in terms of identity while the Bosphorus has more variation in terms of expression, pose variation and occlusions.



Table 3.1: Statistics per model of sex and coarse ethnic groups within the FRGC.

	Asian	Black	Hispa	Unknow	White	Total
Male	758 (15.31%)	29 (0.59%)	118 (2.38%)	54 (1.09%)	1774 (35.84%)	2733 (55.21%)
Female	710 (14.34%)	18 (0.36%)	25 (0.51%)	71 (1.43%)	1393 (28.14%)	2217 (44.79%)
Total	1468 (29.65%)	47 (0.95%)	143 (2.89%)	125 (2.52%)	3167 (63.98%)	4950 (100%)

Table 3.2: Statistics per identity of sex and coarse ethnic groups within the FRGC.

	Asian	Black	Hispa	Unknow	White	Total
Male	71 (12.75%)	7 (1.26%)	9 (1.62%)	11 (1.97%)	221 (39.68%)	319 (57.27%)
Female	59 (10.59%)	3 (0.54%)	5 (0.90%)	12 (2.15%)	159 (28.55%)	238 (42.73%)
Total	130 (23.34%)	10 (1.80%)	14 (2.51%)	23 (4.13%)	380 (68.22%)	557 (100%)

### 3.4.1 FRGC

The first database used is the Face Recognition Grand Challenge version 2 [Phillips et al., 2005] or FRGC v2 containing 4950 faces of 557 individuals. The data presents some variation in sex, ethnicity (see Table 3.1 and 3.2), age and expression (see Table 3.3) as well as small variations in pose (under 10 degrees). The database is fairly uneven in terms of capture per identity with some individuals appearing only once while others appear thirty times. This database is widely used in the research community and over the years has become the standard benchmark for face related computer vision systems.

In our experiments training is needed. 200 faces of different individuals are selected as our training set and all the rest are used as a test set (4750 faces).

Table 3.3: Statistics per model of coarse expression categories within the FRGC.

Blankstare	Disgust	Happiness	Other	Sadness	Surprise
3308 (66.83%)	202 (4.08%)	378 (7.64%)	543 (10.97%)	177 (3.58%)	342 (6.91%)

Table 3.4: Categories within the Bosphorus database.

Name	Description	Number	Percentage
N	Neutral Expression	299	(6.40%)
E	Happy/Sad/Surprise/Anger/Disgust	453	(9.71%)
AU	Action Unit Expressions [Savran et al., 2008]	2150	(46.07%)
O	Occlusions (hand, hair and glasses)	381	(8.17%)
YR45	Yaw Rotation 45° Right	105	(2.25%)
YL45	Yaw Rotation 45° Left	105	(2.25%)
YR90	Yaw Rotation 90° Right	105	(2.25%)
YL90	Yaw Rotation 90° Left	105	(2.25%)
YRlow	Yaw Rotation 10, 20&30° Right	315	(6.75%)
PR	Pitch Rotation up and down	419	(8.98%)
CR	Cross Rotation Pitch+Yaw	211	(4.52%)
IGN	Unclassified (Ignored)	18	(0.39%)

### 3.4.2 Bosphorus

The second database is the Bosphorus database [Savran et al., 2008]. It contains 4666 captures of 105 people. Unlike the FRGC, it contains large variations in pose (up to 90 degrees around the yaw axis) and occlusions (hand, hair and spectacles partially covering the face). Each individual appears between 29 and 54 times. Our experiments are performed on different subsets, as seen in table 3.4.

### 3.4.3 Data Preparation

Since pre-processing can greatly influence the results of a system and because it is really difficult to reproduce in detail, pre-processing has been avoided as much as possible for this project. One unavoidable preparation is to reduce the resolution of the original data and to format different databases into a common representation.

**File format** Most databases use either a depth map or a structured point cloud to store the output of the 3D sensors. As a structured point cloud cannot be represented by a depth-map without losing information, the decision was made to use a structure point cloud file format to bring different databases to the same representation. The `.abs` file format from the FRGC is used as a standard and files from the Bosphorus database (`.bnt`) are converted into `.abs` format.

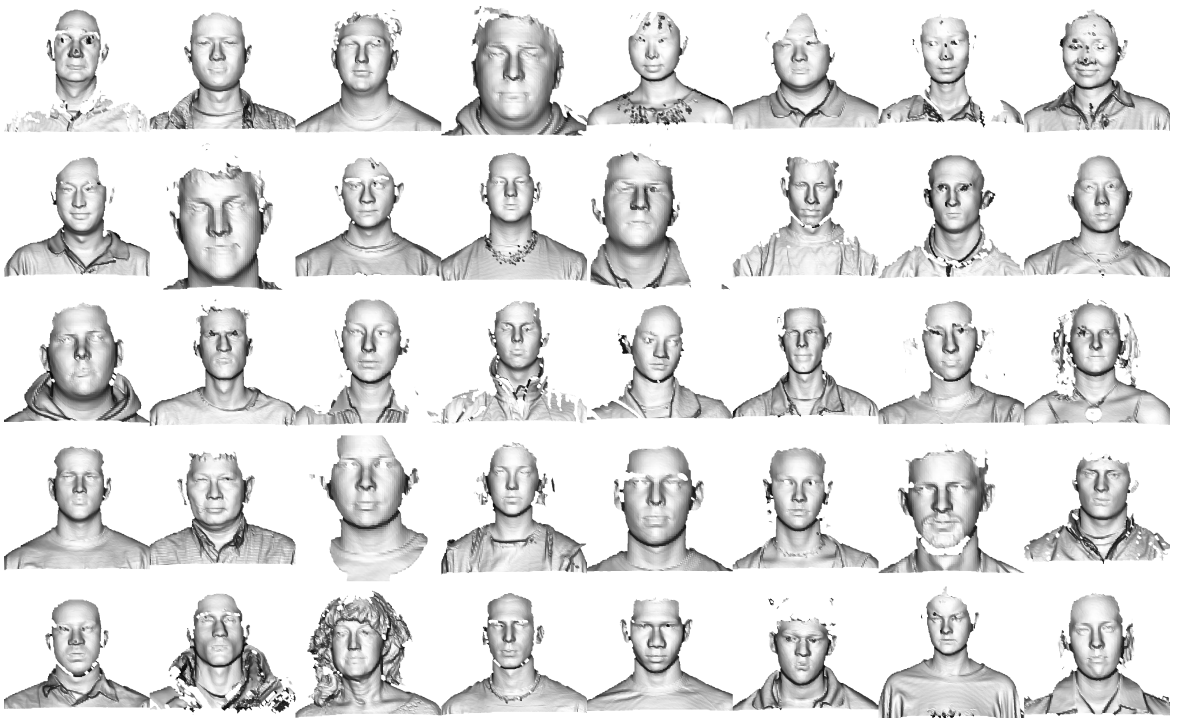


Figure 3.9: Sample of meshes from the FRGC dataset.

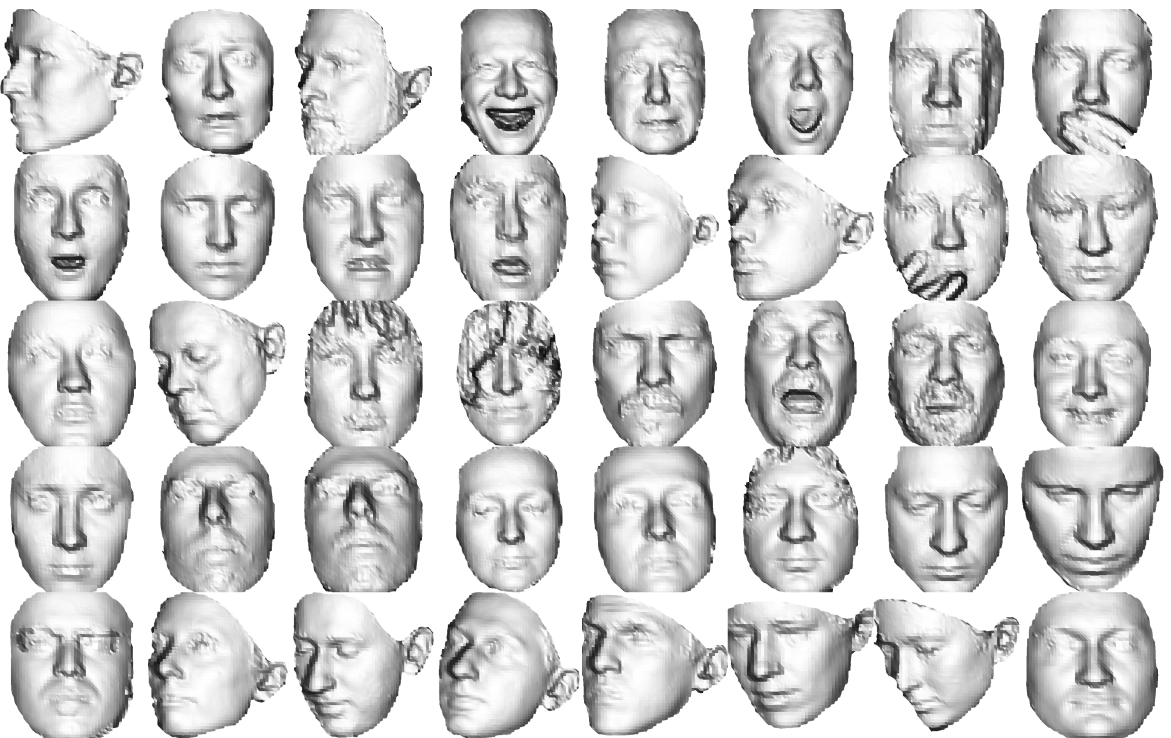






Figure 3.10: Sample of meshes from the Bosphorus dataset.

For our process inputs, Wavefront `.obj` file format are used to represent the query as a 3D mesh. The `.abs` structured point cloud are converted into `.obj` files by first extracting points on a grid and secondly by generating triangular faces between those points.





**Resolution reduction** In order to reduce the resolution when constructing the mesh from the 2D range image two techniques have been used.

- Fixed-capacity bins: The resolution is reduced by replacing each block of  $k \times k$  raw 3D data points with its average. (Usually  $k=4$ )
- Fixed-size bins: The resolution is reduced by binning input data into square bins of fixed size  $L$ . (Usually  $L=3\text{mm}$  or  $3.5\text{mm}$ )

These techniques are applied to the input range image by following two possible sampling protocols:

- Four Points Technique: Average  $z$ -position of points into squared non-overlapping bins forming a grid (   $\Rightarrow$   )
- Five Points Technique: Average  $z$ -position of points into squared bins of two grids (one being offset by 0.5 pixel size in  $x$  and  $y$ ) (   $\Rightarrow$   ) . This was used to get more accurate pseudo-geodesic distances between points.

**Mesh creation** Triangular faces are constructed from the set of points to facilitate the computation of neighbourhoods and normals. Two approaches have been used:

- Four points technique: Two triangular faces are defined for every group of four adjacent vertices (   $\Rightarrow$   ).
- Five points technique: Four triangular faces are defined for non-overlapping group of five vertices (   $\Rightarrow$   )

All created triangular faces are defined anti-clockwise from the camera view ( $z$  axis). The normals are therefore pointing outward from the face. Examples of meshes generated from the range image can be seen in Figure 3.11. Optionally, the conversion process can also associate each vertex with texture coordinates (see Figure 3.12). In this thesis, we only present results using meshes constructed with the four-points technique.

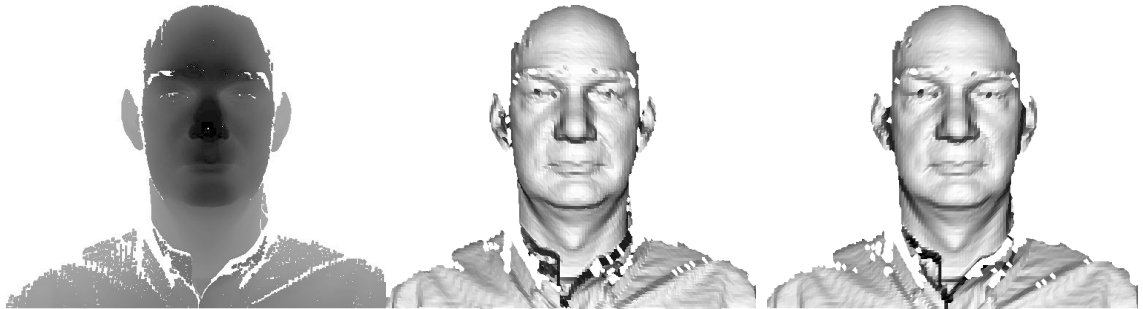


Figure 3.11: Depth-map generated from the `.abs` file (left). Mesh generated using fixed size bins (center). Mesh generated using fixed-capacity bins (right).



Figure 3.12: Plain mesh (Left). 2D texture mapping (Center). 2D contour mapping (Right).

## 3.5 Metrics and Performance Evaluations

A crucial element in planning this research project is to know how to evaluate and quantify successes and failures for every method considered. In this section, the metrics and cost functions used for this project are detailed. The selection of our targeted model and the construction of the ground truth used in one part for training and in another part for performance evaluation are explained.

### 3.5.1 Target Model

For this project 14 landmarks have been selected as targets for our pattern recognition system. These points (see Figure 3.13) are defined for almost every human being (except in some cases of people suffering from facial injuries and other facial deformations).

It is important to specify that we are looking at the local shape area surrounding these points because a point itself does not have a shape. Our main goal is to mimic the ability of average non-specialised human beings to locate features on the face. The goal of detecting

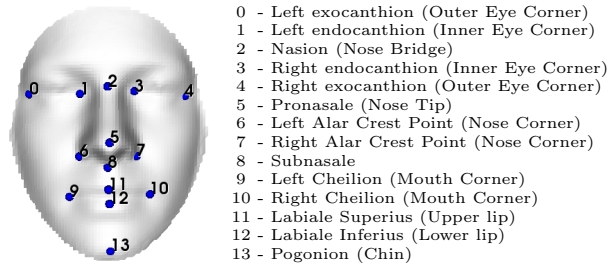


Figure 3.13: Position of the 14 landmarks that need to be retrieved.

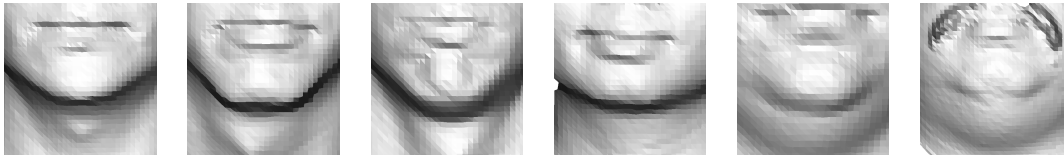


Figure 3.14: Examples of chin shapes. Detecting one single zero-dimensional location called “chin” doesn’t really make sense. What we want our system to do is to locate coarsely the region, and find its approximate centre as a non-anthropologist would do.

anthropological landmarks as defined in [Farkas, 1994] is a different problem as it involves (after a coarse detection) the use of human defined rules on precise inputs. However, the nomenclature used by anthropologists can be very useful for distinguishing points and explaining what the detection process is meant to detect. A concrete example is for example the distinction of the eye’s outer corner in 2D and 3D (see Figure 3.15): in 2D it is more easy to detect the exocanthion landmark (texture contour saliency) as being the corner of the eye while in 3D it is more easy to detect the lateral orbit landmark (geometric saliency). This vocabulary can therefore help us define our ground truth data and its limitations for particular tasks.

### 3.5.2 Landmarks’ Ground Truth

Our objective is to built a system that mimics the coarse feature localisation capability of human beings. Therefore an easy way to evaluate the success of localisation is to compare the landmarks found with the ground truth, as defined by human operators.

**FRGC** For the FRGC database, a mixture of landmarks provided by [Szeptycki et al., 2009] and [Romero and Pears, 2009a] are used with additional manual and semi-automatic points. In [Szeptycki et al., 2009], 15 landmarks are defined for the whole database, four of which

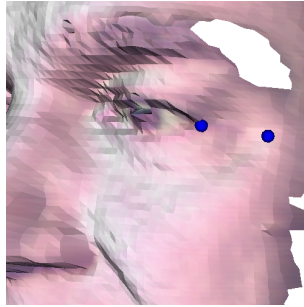


Figure 3.15: Difference between a right exocanthion and a right lateral orbit landmark. The first is a flesh/skin related landmark (corner of the lid) while the other is detected on the bone structure.

are upper and lower lids landmarks, which have no use for face recognition.

In [Romero and Pears, 2009a], 11 landmarks are defined for a subset of FRGC due to registration problem between the 2D texture and the 3D model. After merging the two databases and manually detecting missing points on the previously discarded models, a new landmark (the subnasal) was detected semi-automatically for the whole database as explained later.

**Bosphorus** For the Bosphorus, the landmarks used are the ones provided with the database [Savran et al., 2008] with additional manual and semi-automatic points and position refinements.

**Semi-automatic Refinements** In order to improve the localisation of the ground-truth data (especially the landmarks detected semi-automatically and landmarks detected on the texture before projection) an automatic refinement system was set up. It consists of using a local Iterative Closest Point (ICP) method [Besl and McKay, 1992] to register the local neighbourhood of a current landmark to the same local shape extracted from a mean model. The difference of position between the centre of the two patches after registration is used to correct the position of the input landmark (see Figure 3.16). This method was used for the nasion, the left and right corner of the nose and the subnasal.

While this method was used to speed-up the generation of the ground truth, every single model was checked and corrected manually.

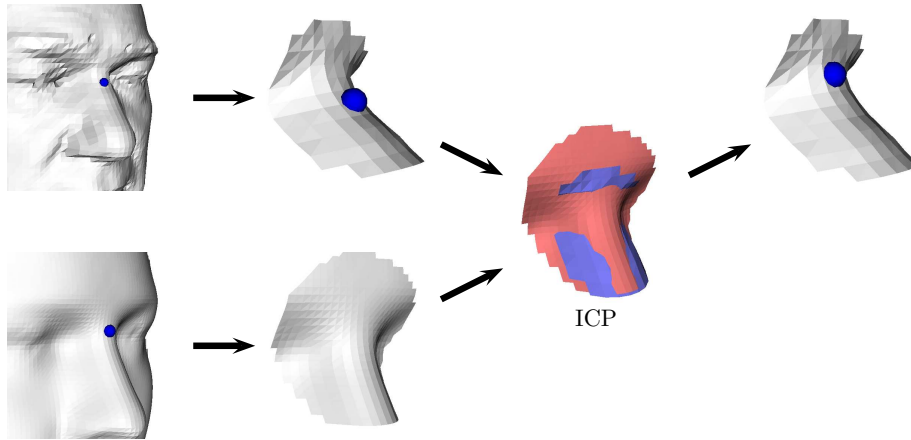


Figure 3.16: Position refinement process. The local region on the query mesh is cropped (top). The corresponding region in the model is cropped using a larger radius (bottom). Both are registered using ICP. The ground-truth position (on the model) is transferred to the query mesh.

**Errors in the ground truth** The ground truth cannot be absolutely perfect. It is sometimes difficult to know whether a landmark should be defined or not, especially when pose variation and occlusion are present. Since the landmarks were detected on 2D texture and projected onto the 3D model, some landmark positions are sometimes completely wrong (spikes, border of the mesh, and so on). We tried to correct most of these manually but errors may still be present.

In addition to obviously wrong detection, errors in terms of positioning distance will also appear. The same individual will not place the same landmark at the same position every time, and different individuals given the same instructions might place points at different positions. This cannot be measured in our case, because each face has been landmarked only once and it is impossible for us to quantify the human variability in hand-placing each class of landmark. However an upper bound for these errors can be defined with relative confidence. In this thesis, it is assumed that a human will always localise a landmark within 10mm of the defined ground truth position. Therefore, when giving a detection rate at 10mm, our results have to be compared to 100%.



### 3.5.3 Cost Functions

Here we present the main measure used to evaluate our system. Cost functions for particular problems will be introduced in the different chapters as necessary.

**Landmark retrieval rate:** We define the *landmark retrieval rate* for a particular landmark as the percentage of test models in which the system correctly retrieved its position given an error acceptance radius. For example, if the test set contains 1000 models, among which 950 has a ground truth landmark for the nose tip, and if we use an error acceptance radius of 10 mm, the landmark retrieval rate will be the percentage of the 950 models which present a detected “nose-tip” landmark within 10 mm of the known “nose-tip” position.

Usually it is not clear what radius should be used for such evaluation. As a good practice and to facilitate result comparison with other researchers, the retrieval rates are provided for an increasing acceptance radius (usually ranging from 2.5 mm to 25 mm in steps of 2.5 mm).

**Landmark positioning error:** Computing the distance from every localised landmark to the ground truth position of corresponding label is an interesting measure for the global landmark localisation framework. However this continuous measure is more meaningful for landmark positioning refinement than for coarse landmark localisation. This measure doesn’t allow notions of discrete failure except if coupled with an error acceptance radius, for example by measuring the positioning error only for points less than 20 mm from the ground truth.

**Global registration error:** Assuming a rigid representation of the face, a global measure of how well the two sets of landmarks correspond is to look at the mean registration error distance or at the difference in corresponding 3D transformation (translation, scale and rotation). If the computed correspondence contains errors, the global transformation of the output will be very different from the transformation produced using the ground truth set of landmarks. This give us an interesting non-landmark-related measure of the overall matching.

## 3.6 Expected Limitations

**Learning limitations** A first obvious limitation of our approach is that it relies entirely on explicit learning and consequently on a very narrow concept of face. Therefore it might not be able to deal with non-realistic facial representations. The concept of “face” as defined by humans is very large: faces can be seen in clouds or in emoticons. 3D animation movies provide us with lots of examples of anthropomorphised objects in which one can recognise facial components without prior training. This suggests that a meta-learning of what a face is is taking place. This is clearly out of the reach of any current computer vision system.

**Descriptor limitations** All local shape properties extracted from the query are extracted using sets of descriptors. Our method will not be able to overcome the intrinsic limitations of those descriptors. For example, if non-scale-invariant descriptors are used, our system cannot be scale invariant.

**Input quality** As our system uses both local and structural information a big proportion of the face should be present in the input (at least half of it). As our system doesn’t perform any hole filling, spike removal or denoising, the input data has to be of reasonable quality. We argue that for results comparison it is preferable to sometimes fail on publicly available inputs than to always succeed on non-publishable inputs (pre-processed 3D models).

**Optimisation limitations** Due to the complexity of the problem and the size of the search space it is not always possible to isolate the method from all the contextual parameters and variations of the input. We try to make explicit in each chapter the limitations of our evaluations by listing all variables that were not thoroughly scrutinised and which were set by empirical non-systematic tests and in some cases by mere educated guesses.

## 3.7 Conclusion

In this chapter, we presented the problematic of this thesis and some of the preliminary choices we made to constrain the problem. Our global strategy to solve the problem is based on the idea that the rules for detection defined by researchers should be learnt automatically

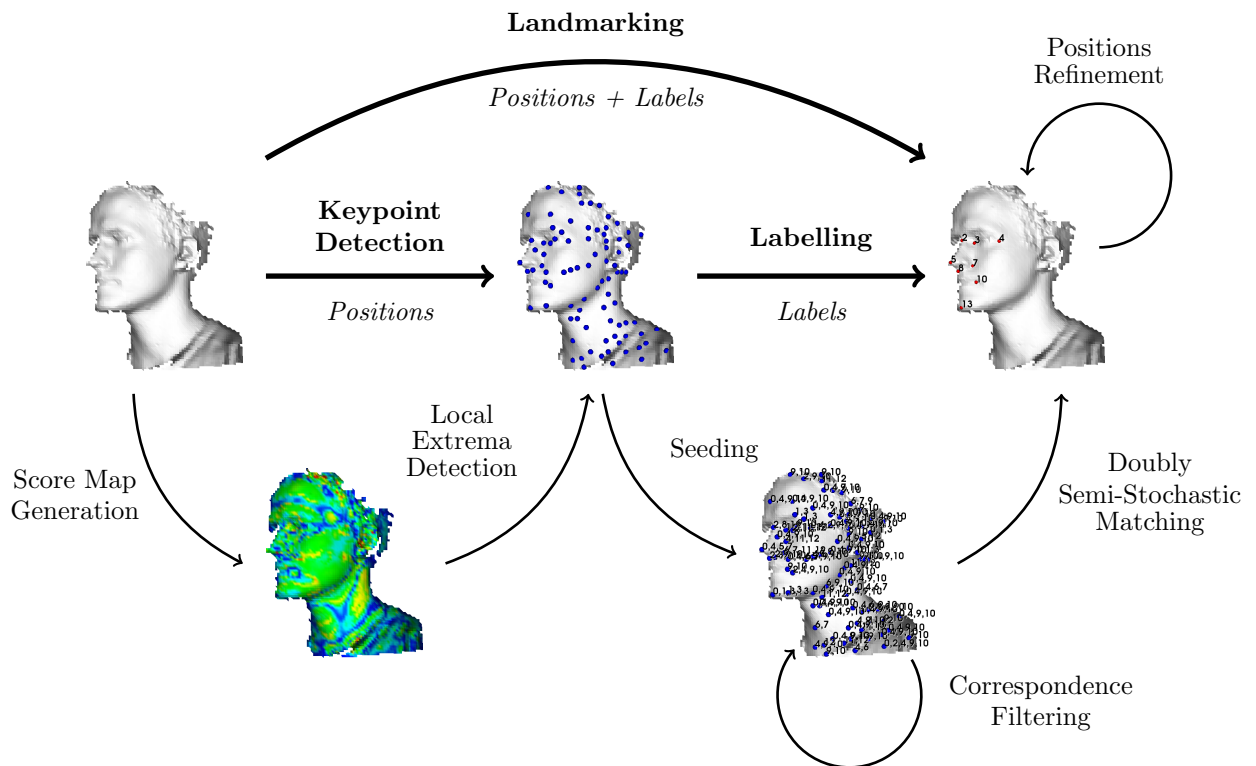


Figure 3.17: Synoptic diagram of our framework. The landmarking problem is split into a keypoint detection problem, a labelling problem, and, optionally, a local refinement problem.

rather than enforced, and on the intuition that candidate detection and labelling should be designed and optimised independently. We presented a non-iterative template framework for solving the problem that, we think, is very generic (see Figure 3.17). We also presented ways to evaluate the results obtained for the sub-problems we aimed to solve, taking advantage of two publicly available 3D face datasets on which a ground-truth positions of landmarks are known. The following chapters will look at explaining and evaluating solutions for our two main sub-problems (keypoint detection and labelling), before testing the whole framework for automatic 3D face landmarking.



## Chapter 4

# Learning-based Methods for Automatic 3D Keypoint Detection

A difference is a very peculiar and obscure concept. It is certainly not a thing or an event. This piece of paper is different than the wood of this lectern. There are many differences between them - of colour, texture, shape, etc. [...] Of this infinitude, we select a very limited number which become information. In fact, what we mean by information - the elementary unit of information - is a difference which makes a difference.

---

Extract from *Steps to an Ecology of Mind* by Gregory Bateson  
University of Chicago Press, 2000

In this chapter an automatic method for extracting keypoints on surface meshes is presented. After learning a set of local shape descriptors at manually labelled location on a training set, the system is able to rate every vertex of the input mesh with an interest score. The local maxima over this map are selected as keypoints. Many strategies are possible to combine information from different weak descriptors. Two are evaluated here: one where the maps are combined linearly, one where the scores are determined using a boosting technique. These approaches allow us to detect keypoints at locations of less distinctive shape which is often impossible with standard single descriptor techniques.

## 4.1 Introduction

While 3D vision has some advantages over 2D vision, it has also a set of specific drawbacks. One of which is that local descriptors in 3D are usually weak. The local curvature, for example, is not very discriminating by itself. This very reason has led to a strange phenomenon: while many researchers are detecting facial features with well defined machine learning techniques in 2D vision, researchers working with 3D faces are almost exclusively doing it using expert systems. These are usually landmark-dependent sequential recipes taking advantage of patterns detected by their designers. Such a recipe can be for example to take the tip of the nose as the most extreme point along one direction or as the most curved point in the input 3D model. These recipes give very good results for very salient points on most existing databases, but are bound to fail on unconstrained databases representing people in non-cooperative situations.

Here our contributions in methodology are three-fold:

1. To replace existing expert systems for 3D face feature localisation by machine learning techniques, such that the rules used for detection are learnt and not enforced by the designer of the system. This also allow the detection of less salient features for which humans struggle to create rules.
2. To relax some of the assumptions about the input data toward non-cooperative face recognition. Most detectors use a global approach trying to find extremal points in the input. These expert-designed patterns might be true for the face region but do not apply to all the unknown non-face part of the input data. Such approaches should therefore be discarded.
3. To increase the number of landmarks to be searched for in parallel in order to minimise the risk of task failure due to occlusions and spurious data in non-cooperative cases.

Our contributions at a practical level in this chapter are:

1. A new method for keypoint detection on meshes using a dictionary of learnt local shapes (see Figure 4.1).

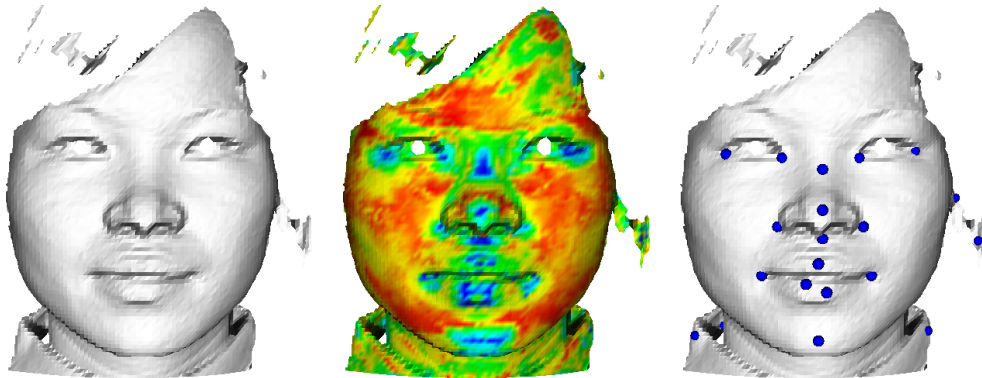


Figure 4.1: Example of keypoint detection on a model from the FRGC database. Plain mesh (centre), final keypoint score map (centre), detected keypoints.

2. An evaluation of its performance in different conditions.
3. An extension of this method to a non-linear boosting classifier.
4. A study of the behaviour of 10 common 3D descriptors over a range of scales at 14 landmark positions.

Here the scope is limited to the *detection* of probable landmark candidates. Selecting/labelling those candidates for a full landmarking system is the subject of chapter 5. To our knowledge nobody has evaluated the landmark candidate detection independently from the landmark localisation problem. In our view, these are two different problems, one is purely local, while the second is mainly structural. Evaluating the success of our methods against other work is almost impossible in this chapter as the nature of the problem is different to what is usually done in the literature (keypoints are not landmarks). However its use as a component of a landmarking system can be evaluated against state of the art feature localisation techniques, and this will be done in chapter 6. In this chapter, the term “keypoint” is justified as we try to detect unlabelled repeatable point of interest. However, our approach differs, as the scope for the targeted repeatability is larger. Our technique should be able to detect repeatable point of interest across the population and not only for several captures of the same individual. Therefore, our system is designed to extract macro-features (nose, eyes, mouth) common across a whole population of objects (faces), instead of discriminative micro-features (e.g. wrinkles) that are often specific to individuals.

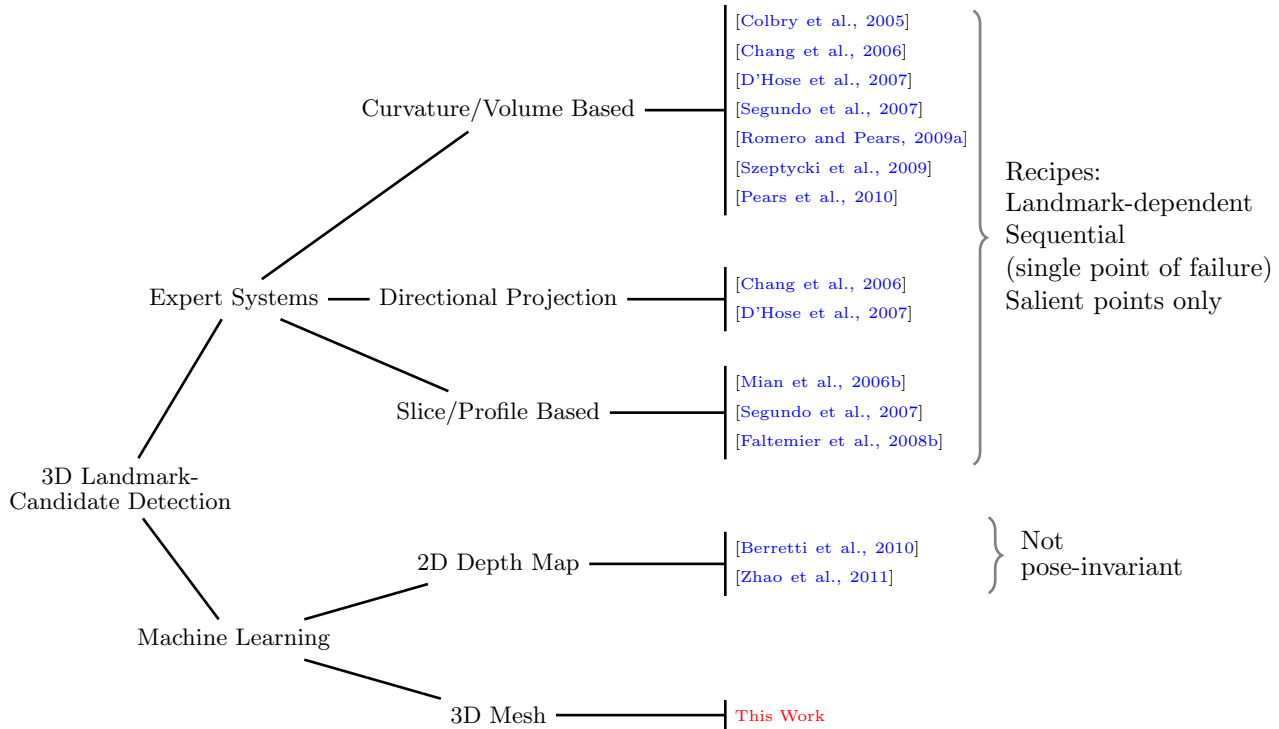


Figure 4.2: Taxonomy of related works on this problem. For more background information please refer to the literature review in chapter 2.

This chapter starts with a presentation of the local properties that can be computed on meshes. Explanation of a scoring scheme using learnt statistics of local shape is then presented. The following section looks at ways to convert score maps into sparse keypoints on the surface mesh using local maxima techniques. In section 4.6 a first method is proposed in order to detect keypoints at probable landmark positions by linearly combining descriptor scores. In section 4.7 an improved method is proposed introducing non-linear scoring for the vertices of the query mesh.



### Rationale

**What:** To detect keypoints on 3D faces similar to previously learnt local shapes.

**Why:** To provide meaningful and sparse landmark-candidates to a landmark localisation system.

**How:** By combining local scores obtained with several weak descriptors over the surface mesh.

**Constraint/Priorities:** Retrieval rate of the learnt local shape and speed are our main priorities.

**Precision:** A coarse localisation is our objective, a 10mm offset from the ground-truth can be considered a very good match.

## 4.2 Descriptor Maps

This whole chapter is based on the use of local 3D shape descriptors; in this section, details about the descriptor maps and their computation are presented. In our experiments, two kinds of descriptors are used, scalar value descriptors and histogram descriptors.

### 4.2.1 Normals

Several of the descriptors require a normal defined at each point. To compute the normals, a simple method using the adjacent triangle faces is used. When the mesh has been built from the 2D depth map, all the triangle faces have been defined anti-clockwise with regard to the camera position. Therefore all the normals of the faces are pointing outward. When setting the normal at a point  $i$ , a weighted sum of the normals of the neighbouring faces is computed. The weights given for each of the touching faces are computed using the Nelson Max technique [Max, 1999]:

$$cN_i = \sum_{k=0}^{n_i^e-1} \frac{V_k \times V_{k+1}}{|V_k|^2 |V_{k+1}|^2}$$

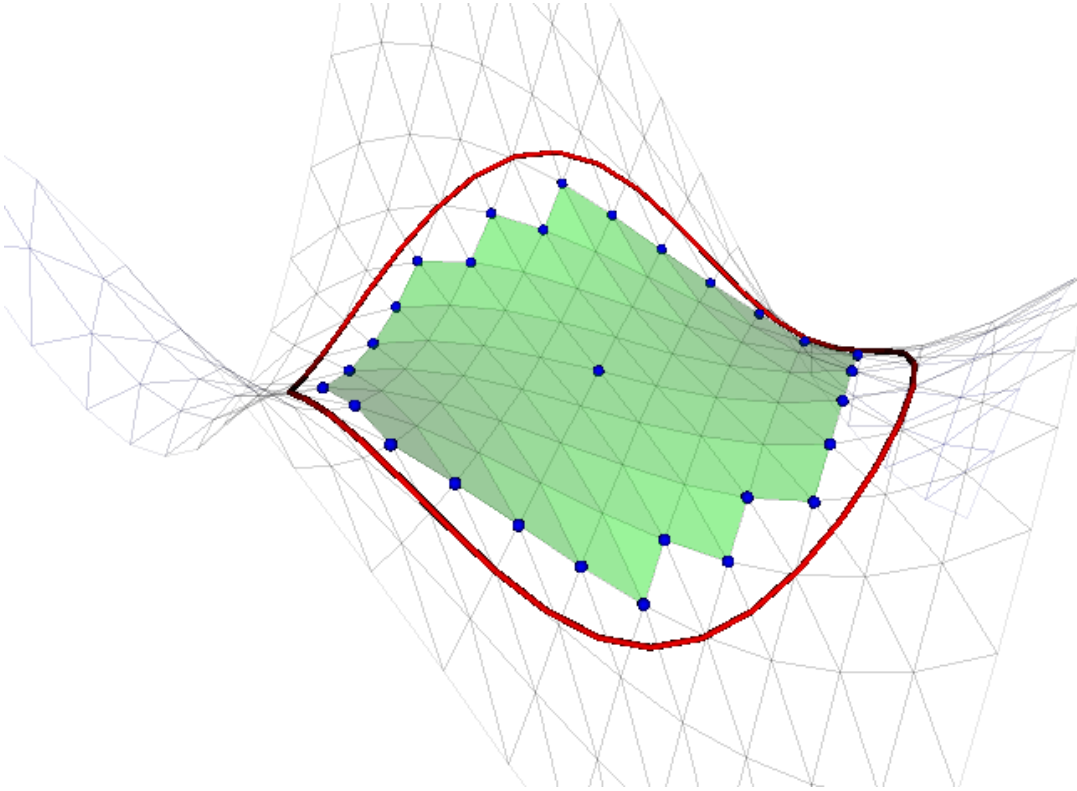


Figure 4.3: Neighbourhood computed using Euclidean distance. The red line is the intersection of the sphere of radius  $R$  with the surface. Every point inside the sphere is part of the local neighbourhood. The blue vertices represent the perimeter.

where  $c$  is a constant that disappears after normalisation,  $n_i^e$  is the number of edges adjacent to vertex  $i$  and  $V_k$  the vector corresponding to the  $k$ -th neighbouring edge. As the normal is computed in every point, the process is sped-up by looping over the triangle faces and accumulating the face normals with the corresponding weights on the three vertices of the face. This guarantees that the face normals are computed only once (instead of three times with a naive algorithm looping on vertices).

## 4.2.2 Neighbourhoods

A point by itself contains very little information: only its position in the space. To extract more information one needs to know how this point is positioned with regard to the other points around it. To determine this local neighbourhood the only thing that is needed is a metric and a threshold. All the points whose distance to the centre point is shorter than the

threshold are included in the neighbourhood. There are many kinds of distance that can be used to determine the local neighbourhood:

- Graph distance (ring level): the number of edges in the shortest path linking the two points (Dijkstra path).
- Euclidean distance: the classical distance (L2 norm) between the two point coordinates.
- Geodesic distance: the distance along the geodesic curve linking the two points, which is the shorter distance over the surface.

Computing geodesic distances over a discrete surface between all points is computationally expensive. An approximation of the geodesic distance (pseudo-geodesic) is easily computed from the Dijkstra path between two points by summing the length of the edges in the path. Another more precise technique is to use a Fast Marching Method over the mesh manifold to compute the distance map from a single source point for all vertices [Kimmel and Sethian, 1998]. The same method can be improved to efficiently find the geodesic distance between two given points (see [Surazhsky et al., 2005]). However, in you want to compute a geodesic neighbourhood you need to compute the geodesic distances for all vertices to all of their local neighbours. This can become computationally expensive when the locality radius gets larger. Furthermore, the advantages of using such neighbourhoods was not obvious. Only results using Euclidean neighbourhoods are presented in this thesis.

From now, when a local neighbourhood of radius  $R_b^a$  is defined,  $a$  will be the type of distance for the neighbourhood and  $b$  the descriptor which is computed from this neighbourhood. For example  $R_{SI}^{Eucl.}$  will be the radius (in Euclidean distance) of the neighbourhood used to compute the descriptor SI (Shape Index). Each computed neighbourhood is quite generic and contains three main components:

- the vertices present in the neighbourhood
- the faces present in the neighbourhood
- the arcs defining the border of the neighbourhood

In practise the Euclidean distance is almost always used. This Euclidean distance can be set to a fixed value or given as a ratio of an intrinsic property of the query mesh (for example a ratio of the bounding box dimensions or of a known distance).

In the literature other kinds of neighbourhood are sometime considered. For example, a neighbourhood containing a fixed number of vertices (exactly  $n$ ), which can be useful in some curvature computation algorithms (for example [Cazals and Pouget, 2003]). Figure 4.3 shows an example of a Euclidean neighbourhood.

### 4.2.3 Curvature

**Principal curvatures** The curvature is a simple notion of 2D geometry that measures the bending of a curve at a particular point. It is defined as the inverse radius of the osculating circle at that location. Therefore a circle has a constant non-zero curvature, while a straight line has a zero curvature as its osculating circle has its centre to infinity. Figure 4.4 illustrates this idea.

This 2D notion can be used with points on a 3D surface by extracting 2D plane curves at those points using an intersecting plane. As a rule the intersecting plane always contains the normal at point A. This leaves one degree of freedom, the angle around the normal, and therefore an infinity of possible 2D curves and curvature values. To constrain the problem to simple values, only two angles are selected: the ones giving the maximal and minimal curvature values known as first ( $k_1$ ) and second ( $k_2$ ) principal curvatures. If there is no direction in which the curvature is maximal ( $k_1 = k_2$ ) then the curvature is constant in all directions. These particular locations are called umbilical points.

Computing these two values for a discrete surface is not trivial. The curvature computation approach chosen for this project is the Adjacent-Normal Cubic Approximation method proposed in [Goldfeather and Interrante, 2004]. As the first and second principal curvatures are known analytically for continuous polynomial surfaces, the idea is to fit one of these (a cubic surface in our case) to a small neighbourhood of points on the discrete mesh. The additional idea is not only to use the neighbouring points to constrain the fitting, but also the normals at each of these points. The cubic surface is defined as:

$$f(x, y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2 + Dx^3 + Ex^2y + Fxy^2 + Gy^3$$

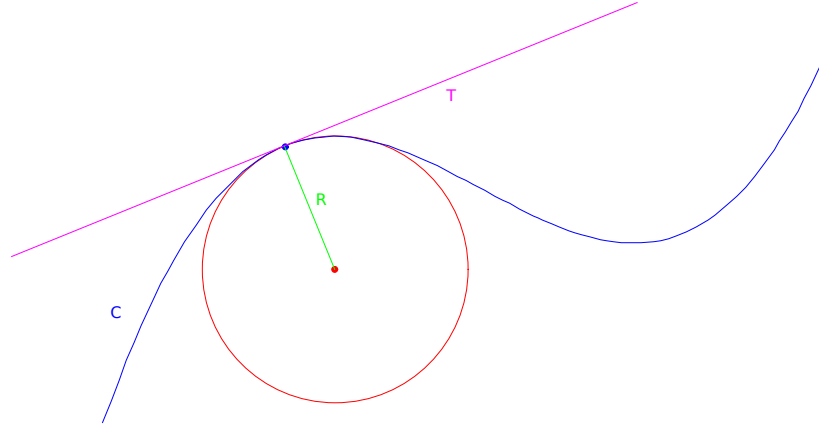


Figure 4.4: Notion of curvature in 2D. The curvature is the inverse of the osculating circle radius at one point.

Let's call  $\mathbf{x}$  the vector of unknown parameters to be determined for the fitting:

$$\mathbf{x} = (A \ B \ C \ D \ E \ F \ G)^T$$

First a local coordinate system is defined with the  $z$  direction along the normal. The point positions  $p_i$  and normals  $n_i$  for all neighbouring vertices ( $i$ ) are defined in this particular coordinate system. For any given  $x, y$  the position on the surface would be  $\mathbf{p} = (x, y, f(x, y))^T$  and the normal direction on the surface would be

$$N(x, y) = \partial_y(\mathbf{p}) \times \partial_x(\mathbf{p}) = \begin{pmatrix} \partial_x f(x, y) \\ \partial_y f(x, y) \\ -1 \end{pmatrix} = \begin{pmatrix} Ax + By + 3Dx^2 + 2Exy + Fy^2 \\ Bx + Cy + Ex^2 + 2Fxy + 3Gy^2 \\ -1 \end{pmatrix}$$

To avoid normalisation of these directions the input normals  $\mathbf{n}_i = (a_i, b_i, c_i)^T$  are rewritten so that the component along  $z$  is  $-1$ :  $\mathbf{n}'_i = (-\frac{a_i}{c_i}, -\frac{b_i}{c_i}, -1)^T$ . The number of equations for the normal fitting is therefore reduced from 3 to 2.

The fitting of the points  $\mathbf{p}_i$  and of the normal directions  $\mathbf{n}'_i$  is done simultaneously by minimising the least-square distance between the real data and the model. In total, three equations are defined for every neighbouring vertex:

- one for the point position:

$$\left(\frac{1}{2}x^2 \quad xy \quad \frac{1}{2}y^2 \quad x^3 \quad x^2y \quad xy^2 \quad y^3\right)\mathbf{x} = z_i$$

- two for the normal directions:

$$(x \quad y \quad 0 \quad 3x^2 \quad 2xy \quad y^2 \quad 0)\mathbf{x} = -\frac{a_i}{c_i}$$

$$(0 \quad x \quad y \quad 0 \quad x^2 \quad 2xy \quad 3y^2)\mathbf{x} = -\frac{b_i}{c_i}$$

The  $3n$  equations form a system :

$$\mathbf{U}\mathbf{x} = \mathbf{d}$$

This system can be solved using several different techniques as explained in the next paragraph. To determine  $k_1$  and  $k_2$  only  $A, B$  and  $C$  are needed (coefficients of the second fundamental form). The principal curvatures  $k_1$  and  $k_2$  are defined as the eigenvalues of the Weingarten curvature matrix of the surface  $W$ :

$$\mathbf{W} = \begin{pmatrix} A & B \\ B & C \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 \\ \mathbf{x}_2 & \mathbf{x}_3 \end{pmatrix}$$

Since the matrix is symmetric, eigenvalues are always real. The solution can even be expressed analytically (faster to compute than the eigenvalues in a general case) as  $\det(\mathbf{W} - \lambda I) = 0$  correspond to a simple quadratic equation:

$$k_1 = \frac{(A + C) + \sqrt{(A + C)^2 - B^2}}{2}$$

$$k_2 = \frac{(A + C) - \sqrt{(A + C)^2 - B^2}}{2}$$

**DIGRESSION:** Implementation Trick

Most people in our field use curvature computation as a black box. Once they have understood the maths, researchers usually use a ready-made library or rewrite a code following instructions from an article. These libraries or methods are (with reason) designed to be very accurate. In 3D face processing, the sensors used are usually quite coarse and the captures are often noisy. Using the same technique people used where curvature precision is critical would be, in our opinion, a waste of resources and time. Most of the techniques for curvature computation have been designed with an emphasis on quality over speed. Because speed is important in our study, and because the face models used are not precise anyway, finding the best approximation with a computationally expensive method is not the best approach.

The time expensive part of the curvature computation is the resolution of equation  $\mathbf{U}\mathbf{x} = \mathbf{d}$ . Most of the time, the LU, QR or SVD decompositions are used to help solve the system. These solvers (especially the one using SVD) can be time consuming. We initially used C++ code by Shin Yoshizawa [Yoshizawa et al., 2008] using SVD decomposition for the principal curvature computation. Switching later on to SVD-based solution using the C++ Armadillo library didn't improve speed performances. However switching to Armadillo `solve` operator (based on LAPACK `gesv`) really improved the performance. A last improvement was obtained by using a trick. The systems of equations are usually not ill formed (singular) except for points near the border of the mesh. As those points are expected to be noisy, it doesn't matter too much if the curvature is not perfect at those locations. Instead of decomposing the matrix to solve the system, the pseudo inverse of  $\mathbf{U}$  is computed while assuming linearly independent columns (LIC). Consequently, the computation is reduced to:

$$\mathbf{x} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{d}$$

Figure 4.5 shows the first principal curvature using different solvers and the relative error induced by switching methods. Similar evaluation has been done on the second principal curvature as well.

The mean execution time measured using these different techniques over 200 faces of 6000 vertices on average are given below:

SVD: 1.06s  
 SOLVE: 0.19s  
 LIC: 0.12s

**Conclusion:** When dealing with remotely acquired faces or other low resolution 3D objects, using expensive methods for the cubic surface fitting seems to be unnecessary. Evidence collected on a small set of 3D models indicates that the time of computation can be divided by more than 8 for meshes of around 6000 vertices ( $R_{curv.}^{Eucl.} = 15$  mm) by using a simple solver instead of the wide-spread, more precise, SVD based solver. Empirical verification on randomly selected meshes shows that the relative error introduced is negligible in comparison to the one usually observed due to sensor noise.

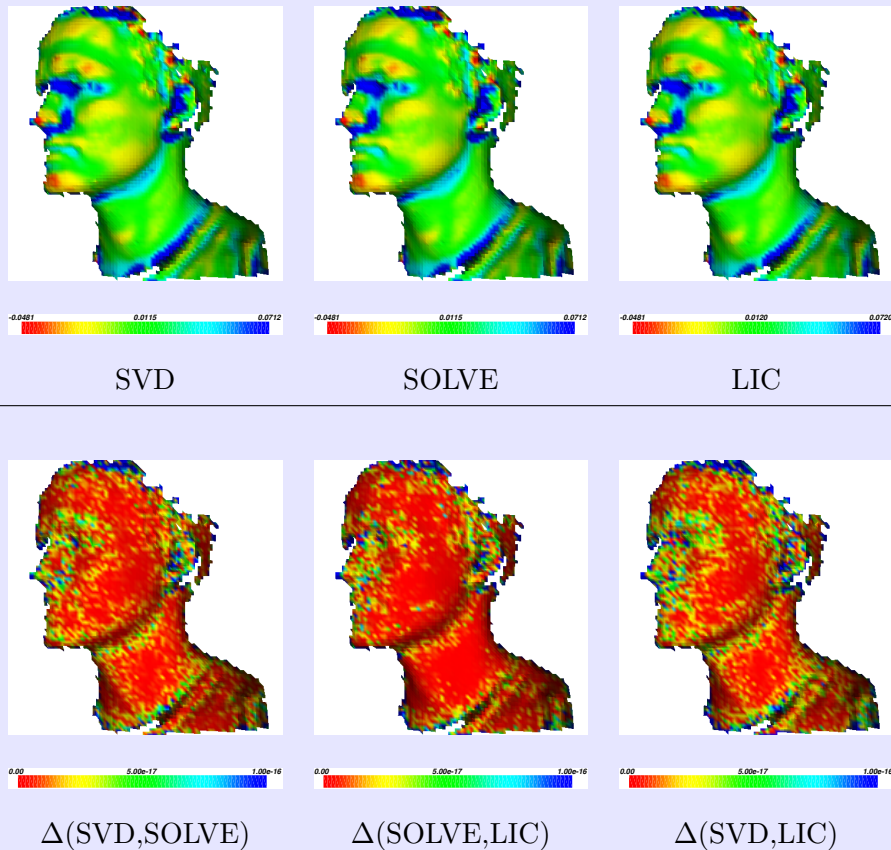


Figure 4.5: First Principal Curvature ( $k_1$ ) maps using 3 different solvers (top line). Visualisation of the absolute difference between  $k_1$  maps (bottom line). In the  $\Delta(\text{SVD}, \text{LIC})$  example, only 10 vertices show a relative error above  $10^{-10}$  and those cases occur on the border of the mesh.



### 4.2.4 Simple Descriptors

Here “simple descriptors” refers to descriptors that can be described by a single scalar value. Simple descriptors are ubiquitous in 3D keypoint detection, in particular curvature and volume related descriptors.

**Descriptors derived from the principal curvatures** The two principal curvatures  $k_1$  and  $k_2$  are rarely used directly. Most of the time they are used to compute other descriptors. Here we give a short-list of the most popular descriptors that can be written as a function of  $k_1$  and  $k_2$ :

- Gaussian Curvature (K):

$$K = k_1 \cdot k_2$$

- Mean Curvature (H):

$$H = \frac{k_1 + k_2}{2}$$

- Shape Index (SI): two flavours

$$SI_{0,1} = \frac{1}{2} - \frac{1}{\pi} \arctan \frac{k_1 + k_2}{k_1 - k_2} \quad 0 \leq SI_{0,1} \leq 1$$

or

$$SI_{-1,1} = \frac{2}{\pi} \arctan \frac{k_1 + k_2}{k_1 - k_2} \quad -1 \leq SI_{-1,1} \leq 1$$

- Curvedness (C):

$$C = \sqrt{\frac{k_1^2 + k_2^2}{2}}$$

- Log-Curvedness (LC):

$$LC = \frac{2}{\pi} \log \sqrt{\frac{k_1^2 + k_2^2}{2}}$$

- Willmore Energy (W):

$$W = H^2 - K = \frac{(k_1 - k_2)^2}{4}$$

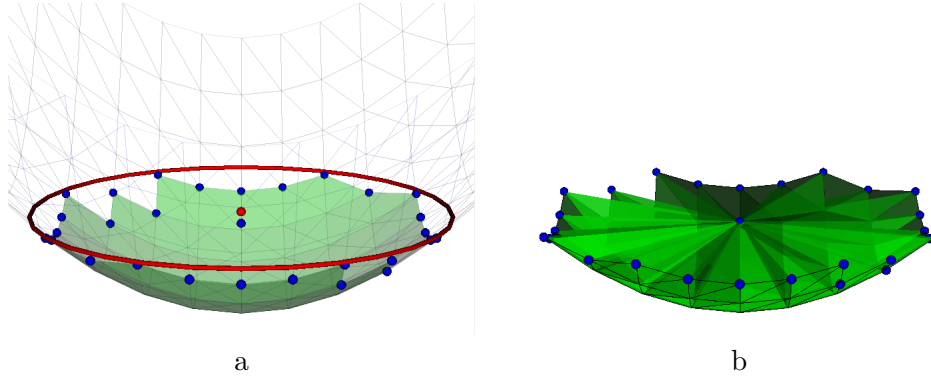


Figure 4.6: Example of local volume (VOL) computed at the extreme vertex of a hyperboloid surface.

a - The neighbourhood border points are used to compute the centroid point (blue) which is not far from the ideal centre of the intersection curve (red).

b - The signed volumes of all tetrahedron are summed.

- SC ([Kim and Choi, 2009]):

$$SC = SI.LC = \frac{4}{\pi^2} \log \sqrt{\frac{k_1^2 + k_2^2}{2}} \arctan \frac{k_1 + k_2}{k_1 - k_2}$$

- Log Difference map ([Dibeklioglu et al., 2008]):

$$D = \ln(K - H + 1) = \ln(k_1 \cdot k_2 - \frac{k_1 + k_2}{2} + 1)$$

### Other Scalar Descriptors

**Local volume (VOL)** Other kinds of measures that can be made from the local neighbourhood are the ones that use volume. It can be the volume under the surface within a ball of radius  $r$ . It can be the ratio of this volume to the whole ball volume. It can be other kinds of approximation of the volume.

A very fast and coarse approximation of the volume has been implemented for this study. First, the barycentre (point  $p_c(p)$ ) of the perimeter of the neighbourhood of  $p$  is determined. Then the volumes of the tetrahedra computed from this point and all the faces of the neighbourhood are summed. Figure 4.6 shows how the tetrahedra are computed. As the faces are oriented the volume can be positive (concave shape) or negative (convex shape).

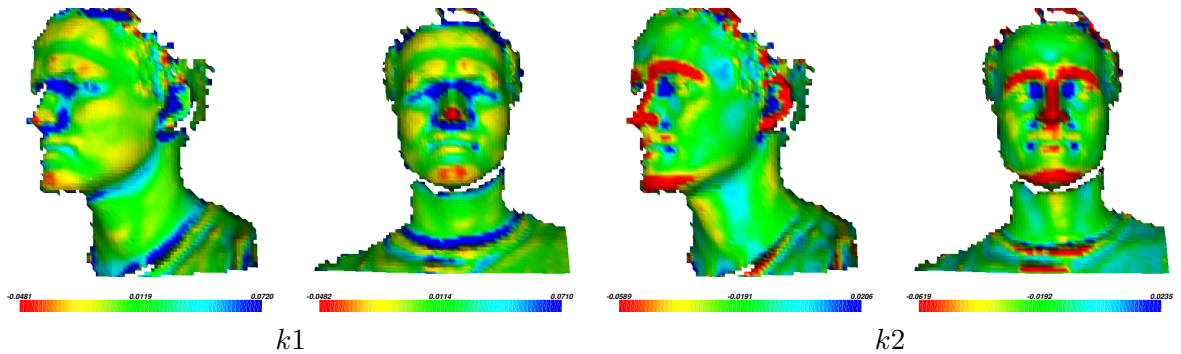


Figure 4.7: Examples of first and second principal curvature maps. ( $R_{scalar}^{Eucl.} = 15 \text{ mm}$ )

**Distance to Local Plane (DLP)** The distance to local plane is a coarse measure of the convexity/concavity at one point [Pears et al., 2010]. It is defined as the Euclidean distance between point  $P$  and the plane fitting its neighbouring points. This corresponds to projecting the centroid of the neighbourhood on the normal direction to the plane passing through  $P$ . In a general, case the neighbourhood used to compute the normal and the neighbourhood used to compute the target centroid can be different. However it is usually simpler to take them as equal.

#### 4.2.5 Histogram Descriptors

Simple scalar values are sometimes limited to describe surface shape. When trying to deal with larger and more complex surfaces, more information is needed than a simple scalar value. We call “histogram descriptors” any D-dimensional array of fixed dimensions containing information about the neighbouring surface at a point location.

It was important that our framework remains versatile and can use any sort of descriptors. However histogram descriptors are often computationally expensive and therefore are not always the best idea for our purpose. In this chapter only two histogram descriptors are used:

- **Spin image:** The spin-image descriptor introduced in [Johnson and Hebert, 1999], consists of a rectangular grid (each cell forming the pixel of an image) which is spun around the normal at the query point location. Each vertex of the mesh around this point is binned in each of the grid cells that have same altitude and distance to the axis of rotation (the normal). The cells (and therefore the resulting pixels) are not required

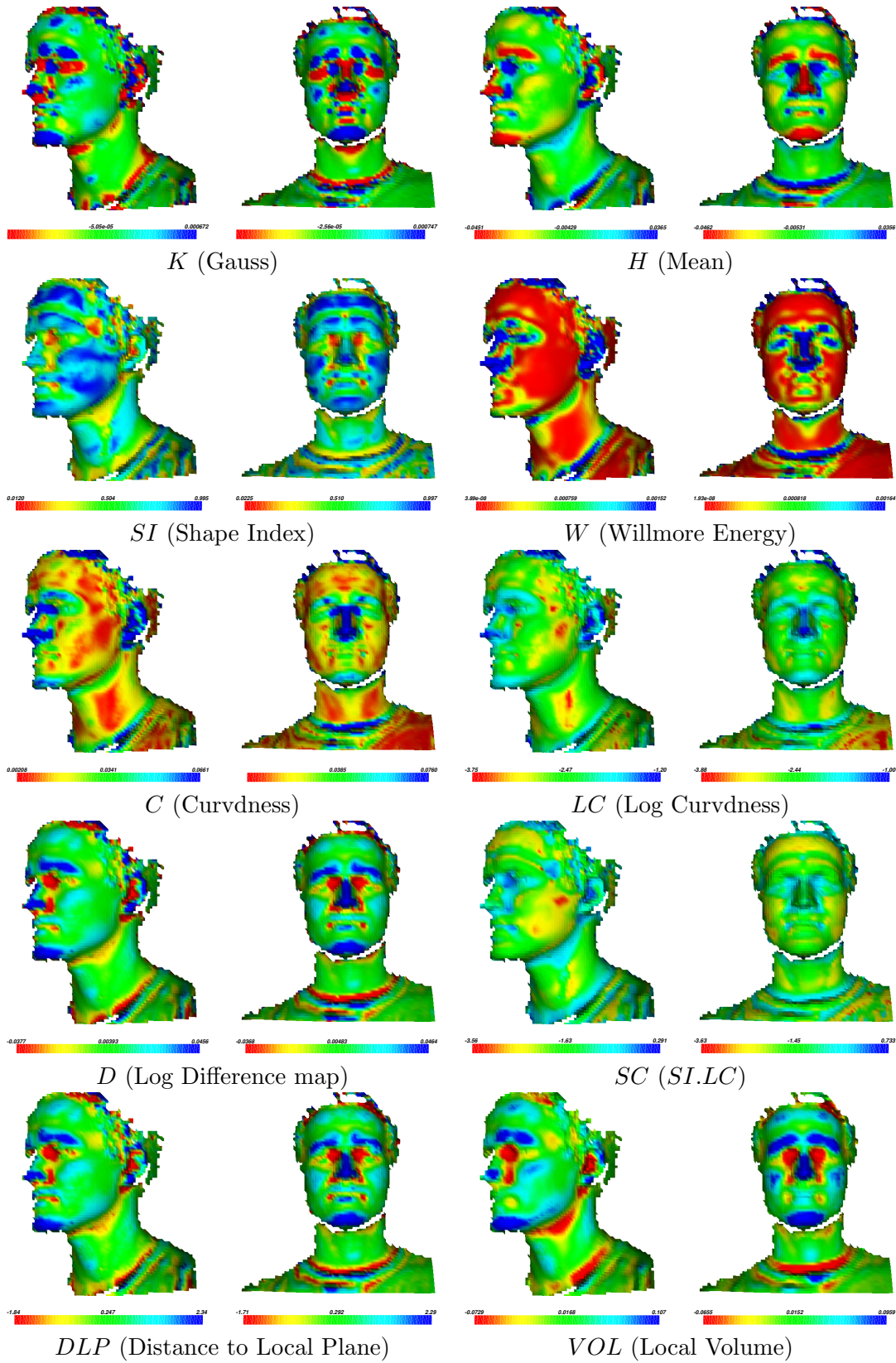


Figure 4.8: Examples of scalar fields computed on two models of the same individual with different orientations. ( $R_{scalar}^{Eucl.} = 15 \text{ mm}$ )

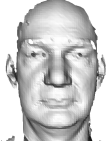































Landmarks			Landmarks		
Outer Eye (L) (00)			Nose Corners (R) (07)		
Inner Eye (L) (01)			Subnasale (08)		
Nasion (02)			Mouth (L) (09)		
Inner Eye (R) (03)			Mouth (R) (10)		
Outer Eye (R) (04)			Upper Lip (11)		
Nose (05)			Lower Lip (12)		
Nose Corners (L) (06)			Chin (13)		

Figure 4.9: Example of spin-image histograms computed for the 14 landmarks on two different faces from the FRGC. Here a bin size of 5 mm is used. The size of the image is 18x9 pixels. The middle top part of the image is the centre point. The left part of the image corresponds to points above the centre in the direction of the normal. The right part corresponds to point underneath the centre using this same direction.

to be square and the cell size can vary from cell to cell (for example by following a log function). In this document, only fixed size cells are considered. The parameters for this descriptor are the number of radial cells, the number of vertical cells (above and underneath the query point), and the radial and vertical cell sizes. If not otherwise specified the radial and vertical cell sizes are assumed equal (resulting in square pixels).

- **Spherical image:** The spherical image is a simplification of the spin image to a one dimensional vector of bins. Each cell represents the number of vertices present between two consecutive spheres centred on  $P_i$ .

## 4.3 Descriptor Matching

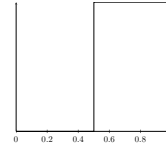
While correlations sometimes exist between descriptors' extremal values and the presence of landmarks (e.g. nose tip, eyes), this can not be extended to less well defined local shape. By learning a distribution of the values of a descriptor for a particular shape of interest, one can easily determine for any new point a score for matching a particular landmark. If the value of one descriptor at a particular point is correlated with the maximum of the probability density function of a known shape, it has a good chance of corresponding to this shape.

### 4.3.1 Distributions of Local Shapes

For each landmark in the training set, the distribution of the values for one descriptor can be observed and approximated with a simple function. A lot of possible idealised distributions and mixtures can be used to approximate the real distribution learnt on the training set. Examples of commonly used functions are:

**Heaviside Step function** (1 parameter:  $t$ )

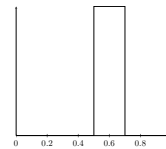
Most used descriptor distribution in literature:  
single threshold to select extrema. Null density.



**Top-Hat** (2 parameters:  $t_1$  and  $t_2$ )

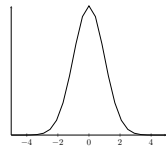
Select range of value.

$$\text{pdf}(x) = \begin{cases} \frac{1}{t_2 - t_1} & \text{if } t_1 < x < t_2 \\ 0 & \text{otherwise} \end{cases}$$



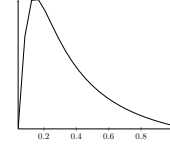
**Gaussian** (2 parameters:  $\mu$  and  $\sigma$ )

$$\text{pdf}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$



**Inverse Gaussian** (4 parameters:  $\mu, \lambda, origin, direction$ )

$$\text{pdf}(x) = \frac{\sqrt{\lambda}}{\sqrt{2\pi x'^3}} \exp -\lambda \frac{(x' - \mu)^2}{2\mu^2 x'}$$

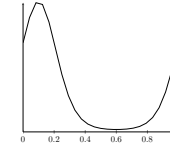


where  $x' = \text{sign}(direction) \cdot (x - origin)$

The variance of that function is  $\frac{\mu^3}{\lambda}$ . The observed deviation being  $\sigma$ , we have  $\lambda = \frac{\mu^3}{\sigma^2}$ .

**Von Mises** (2 parameters:  $\mu$  and  $k$ )

Measure natural distribution of value on toric domain (here between 0 and 1).



$$\text{pdf}(x) = \frac{1}{2.I_0(k)} \exp(k.\cos(\pi(x - \mu)))$$

where  $k$  is the concentration ( $\simeq \frac{1}{\sigma^2}$ ) and  $I_0(k)$  the Bessel function of order 0 in  $k$ .

In our case, a simple Gaussian conveys most of the variation from the natural data set in most cases (see Appendix A). For descriptors that have bounded or toric domains, other distributions might be preferable like the inverse Gaussian (for bounded domains) and the Von Mises distribution (for toric domains). In this document, and when not explicitly specified otherwise, we consider the descriptor distribution at one location to be approximated by a Gaussian. Examples of idealised distribution can be seen on figure 4.10 or more extensively for all pairs of descriptor/landmark in appendix A.

**Limitations:** It can be argued that the selected scheme for distribution representation is oversimplified. No mixture of distributions or sophisticated real-data-to-ideal-distribution fitting algorithms are used. The justification for this choice is twofold. First we believe that the gain in using such techniques will be marginal in most cases as the input data is relatively noisy. Second, our challenge is to combine weak feature descriptors rather than trying to design and match stronger single feature descriptors. The proposed method also has the advantage of being easily reproducible for result comparisons.

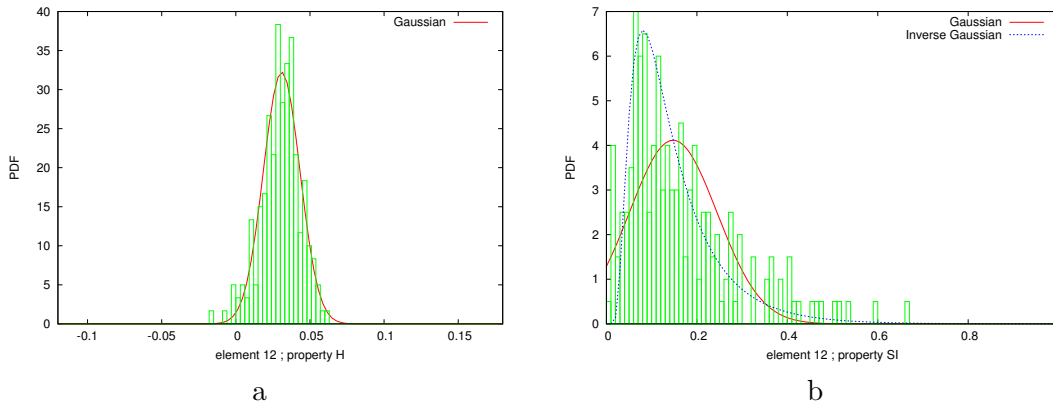


Figure 4.10: Examples of idealised distributions (computed from the mean and deviation) superimposed on the observed distribution for the lower-lip landmark (label:12). On the left, the mean curvature descriptor (H). On the right, the Shape Index descriptor (SI) bounded between 0 and 1.

### 4.3.2 Descriptor-Landmark Score Maps

Computing maps of descriptors is useful especially if the desired targets correspond to local extrema over those maps. In a general situation, this is not the case and the problem becomes the mapping of the initial descriptor values to a space where the extrema are correlated with the searched-for targets.

#### Rationale

**Hypothesis:** The “probability” of a vertex to be a “good” landmark candidate is correlated with the similarity between its descriptor value and the value distribution for the target landmark learnt in the training set.

**Limitation:** This is not true for random maps. This can only apply to maps that produce some form of information about the landmark. This supposes that a pattern exists for the descriptor/landmark pair. The notion of “good” landmark candidates corresponds to multiple objective functions (see section 4.5). Here the word “probability” is used in its popular meaning. It is not *stricto sensu* a probability as the sum over all vertices is not one. The global normalisation required to define probability cannot be done as there is no way to predict the number of “good” keypoints for a given query.



For each vertex  $v_i$ , the score value for the descriptor  $d$  and the local shape  $\lambda$  is defined as:

$$S_X^d(v_i) = \frac{\text{pdf}_X^d(d(v_i))}{\max_x(\text{pdf}_\lambda^d(x))}$$

where  $\text{pdf}_\lambda^d$  is the probability density function of the idealised distribution of descriptor  $d$  for the shape  $\lambda$  and where  $d(v_i)$  is the value of the descriptor  $d$  at vertex  $v_i$ .

In the case of a Gaussian distribution of mean  $\mu_\lambda$  and deviation  $\sigma_\lambda$ ,  $\max_x(\text{pdf}_\lambda^d(x))$  is reach at  $\mu_\lambda$  and we have:

$$S_\lambda^d(v_i) = \frac{\frac{1}{\sqrt{2\pi\sigma_\lambda^2}} \exp -\frac{(d(v_i)-\mu_\lambda)^2}{2\sigma_\lambda^2}}{\frac{1}{\sqrt{2\pi\sigma_\lambda^2}} \exp -\frac{(\mu_\lambda-\mu_\lambda)^2}{2\sigma_\lambda^2}} = \exp -\frac{(d(v_i) - \mu_\lambda)^2}{2\sigma_\lambda^2}$$

In figure 4.11 examples of descriptor-landmark score maps are presented. More descriptor-landmark score map examples are given in Appendix B with 10 descriptors for each of the 14 different local shapes used in this thesis.

### 4.3.3 Dealing with Histograms

Unlike scalar values, histograms are a bit more difficult to deal with. For our particular problem, a final scalar score map is required for each descriptor. Three approaches were considered:

- The first one consists of learning the distribution of values for each of the cells of the histogram for a particular tuple descriptor/local-shape. The score for the descriptor can be the mean or a weighted mean of those individual scores.
- A second method would be to take the Malahanobis distance between one histogram and the corresponding training distribution, however, this would imply a bigger number of operations for each comparison as the correlation matrix is involved ( $\mathcal{O}(n^2)$  instead of  $\mathcal{O}(n)$ ).
- A third method is to look at the difference to the mean of the target distribution and project in the direction that make more sense for the distinction between shapes in the dictionary.

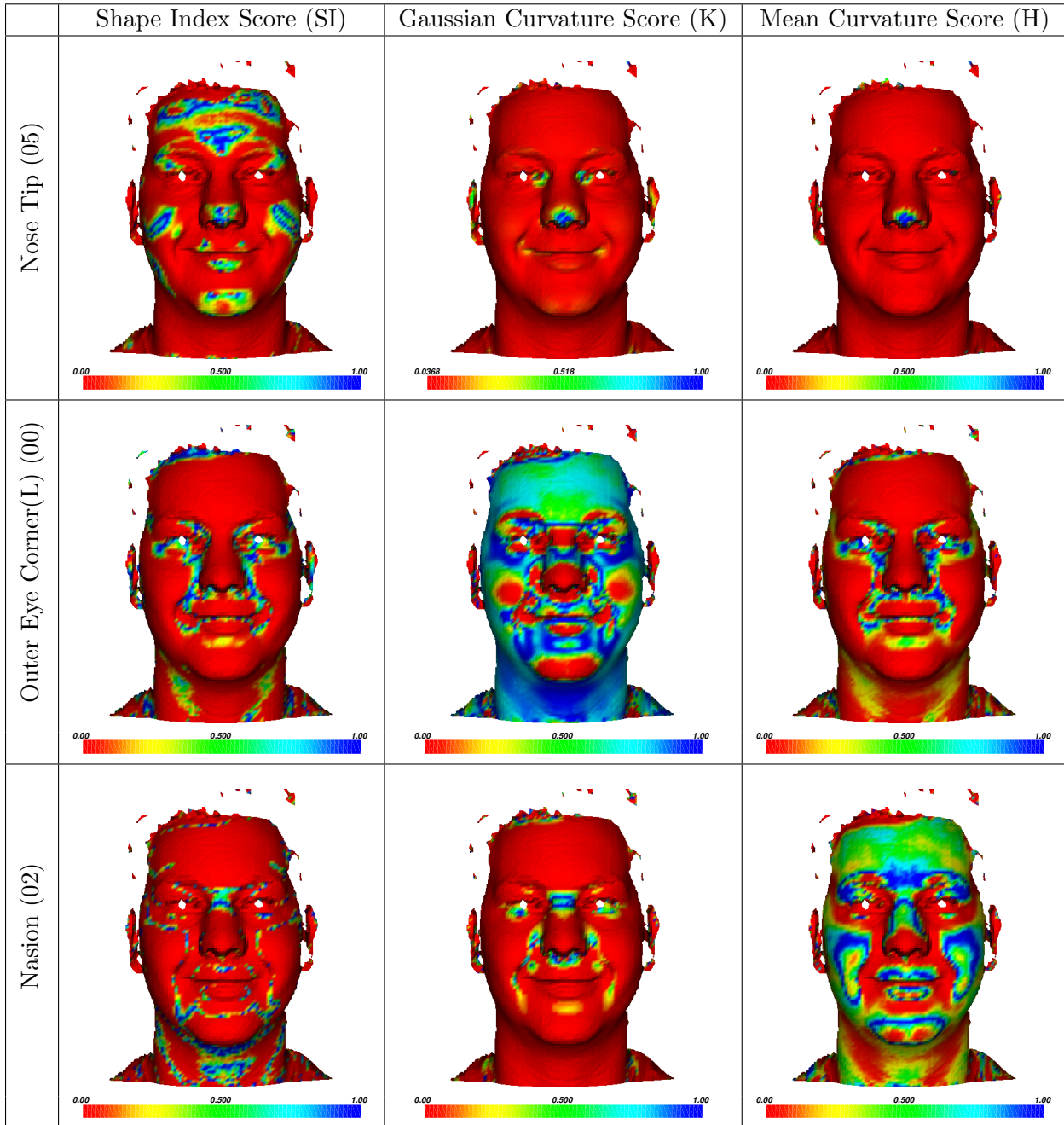


Figure 4.11: Examples of descriptor-landmark score maps for three descriptors at one scale and for three of the target shapes of interest. Each descriptor detects a fuzzy set of vertices. Our objective is to intersect those sets, i.e. to combine the individual scores into a single landmark score map per shape of interest.

In this chapter, the third method is used. For each vertex  $v_i$  the intermediate scalar descriptor is computed as follow:

$$d'(v_i) = \sum_c \omega_\lambda(c)(d(v_i)(c) - \mu_\lambda(c))$$

where  $d(v_i)(c)$  is the value of the cell  $c$  of the histogram descriptor  $d$  at vertex  $v_i$ ,  $\mu_\lambda(c)$  the mean value of the cell  $c$  for the local shape  $\lambda$ , and  $\omega_\lambda(c)$  the weight associated with each cell  $c$  for the local shape  $\lambda$ . The weights are learnt using 2-class Linear Discriminant Analysis on the training database of error histograms. One class is the local shape being treated, the other contains all the other local shapes in the dictionary. The score for the histogram descriptor  $d$  at a vertex  $v_i$  for a local shape  $\lambda$  is defined as:

$$S_\lambda^d(v_i) = S_\lambda^{d'}(v_i) = \frac{\text{pdf}_\lambda^{d'}(d'(v_i))}{\max_x(\text{pdf}_\lambda^{d'}(x))}$$

## 4.4 Local Maxima

This section deals with the problem of extracting keypoints from score maps. In the score map, vertices with higher values are more likely to be good keypoints than ones with lower values. If a simple threshold was applied to the map, the system would detect areas of vertices instead of single sparse points.

**Pitfalls** While selecting local maxima might seem the simplest part of our process it is in fact one of the most problematic. A naive solution to this problem would be to loop on all the vertices and keep only the ones having local maximal value. However this solution fails in cases where several vertices in the same neighbourhood share the same maximal value: if a “strictly greater” test is used, no point will be detected and if “greater or equal” test is used, several points will be detected. Such cases occur often with our system as blue areas in the final map contains several vertices with score 1. In order to enforce sparsity in the local maxima detection more complex methods have to be used.

**Implementation** Our local extrema detector works as follow: The system detects all points which have a score above a given threshold. For each candidate, using a neighbourhood of radius  $R$ , we look at whether a neighbour scores higher than the current selected point.

If several points match, the closest to the centroid of the set of matching point is selected. If this maxima is on the border of the neighbourhood it is not considered as a candidate. The search is repeated on the neighbourhood of the newly found point. The resulting point is stored as a candidate together with its associated score. Any other detection inside the neighbourhood of this point is forbidden by updating a flag map. This last condition was introduced relatively late in the project, the version without will be referred as *version 1* and the one that uses it as *version 2*. When all the candidate have been found, the best  $0.01n$  points are selected as keypoints, where  $n$  is the number of vertices. The reason for this pruning is to ensure that the labelling process will not face a too large number of keypoints, especially when complete-hypergraph matching techniques are used. For a face mesh of around 6000 vertices, about 60 keypoints will be returned.

## 4.5 Objective Functions

For 3D keypoint detection, a number of measures can be used to compare methods, and predict their usefulness. Results for these different measures are given later on in the chapter. A compromise between different objectives has to be found for this particular problem. These objectives include:

- **Sparsity:**

Firstly, a small number of points should be detected, as returning the whole set of vertices wouldn't be useful. In order to be manageable for a structural matching algorithm, the number of output points for our keypoint detector should be less than a few hundreds. As the meshes are usually are a few thousand vertices, a maximum ratio of 1 output point for every 100 input vertices is enforced.

- **Single Landmark retrieval rate:**

Secondly, as landmarks have been used as shapes of interest in the training, the system should be able to detect keypoints at those very locations. This is a measure of how well the training is working for each particular shape. For those landmarks that are visible, this is the percentage of test faces for which a keypoint is detected near the ground truth landmark position.

- **Repeatability:**

Thirdly, keypoints, by definition, should have a high intra-class (same subject's face) repeatability. This is a difficult measure as it implies that a correspondence between the two faces is known. It also has the drawback of being very dependant on the sparsity: for a given matching acceptance radius, the more detected points there are, the more the average repeatability will increase. However at a fixed sparsity it can give an interesting measure of how well the keypoints perform. The global transformations of the FRGC models used to compute repeatability are available on the author's webpage<sup>1</sup>.

- **Landmarks retrieval total number:**

Finally, the number of good retrievals per face is also an interesting measure for our system. Indeed, there will always be cases where 100% of the landmarks cannot be retrieved and this is expected in our framework. The important thing is that a large ratio of the learnt shapes are retrieved correctly (for example 10 landmarks retrieved on average out of 14).

There is no meaningful way of combining these cost functions into one. The trade-off between these measures cannot be automated at this stage. However, some of the cost functions are more meaningful to us than others because of our particular goal. The retrieval rates per landmark for a given sparsity are the most important measure for our framework.

## 4.6 Method 1: Linear Combination of Descriptor-Landmark Score Maps

Most of our descriptors are highly correlated. Indeed most of them are derived from the principal curvatures ( $k_1$  and  $k_2$ ). How can those different correlated clues be combined into a landmark score map for each of the targeted local shapes? A first simple idea to test our framework is to use linear combination of these maps. Results using a Linear Discriminant Analysis (LDA) method to merge the individual descriptor-landmark scores are presented in this section.

---

<sup>1</sup><http://www.cs.york.ac.uk/~creusot>

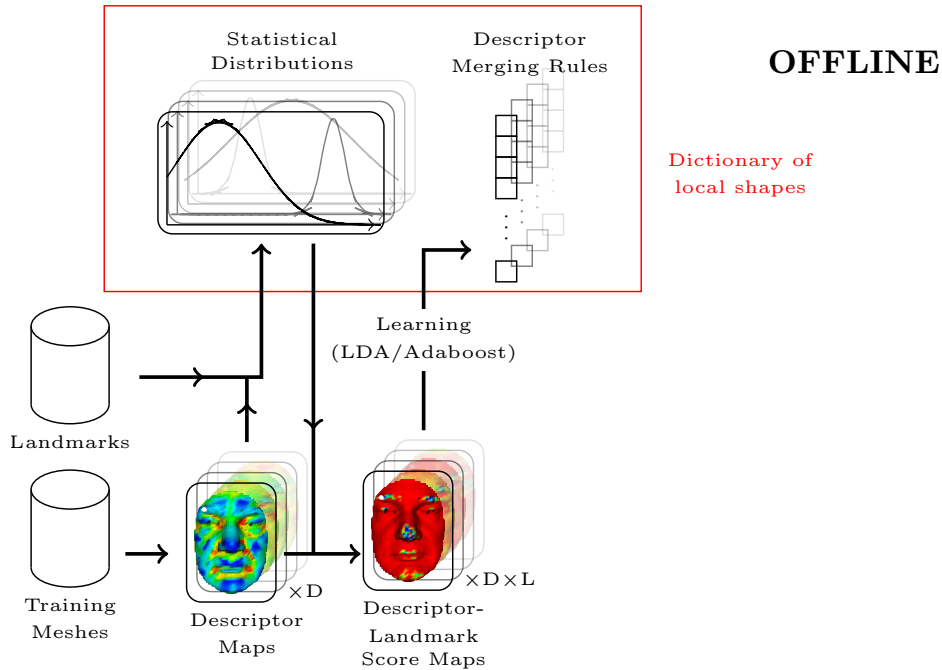


Figure 4.12: **Offline process:** Known landmark positions on the training set are used to learn idealised distributions of the descriptor values for each shape of interest. The descriptor-landmark scores computed using those distributions are used in a two-class Linear Discriminant Analysis (LDA) to determine linear weights for every descriptor.

#### 4.6.1 Workflow

Figures 4.12 and 4.13 explain the workflow of this method. The framework is composed of an offline process and an online process. The offline part is used to teach the system what is considered to be a shape of interest. For that purpose, 14 landmarks over the training set are used to define a dictionary of local shapes from which statistical distributions of descriptors are learnt.

The online part is composed of the following steps:

- $D$  descriptors are computed for all vertices
- For each learnt local shape (14):
  - $D$  descriptor-landmark score maps are computed by projecting the descriptor values of each vertex against the associated learnt distributions of the target landmark. The scores generated are between 0 and 1.

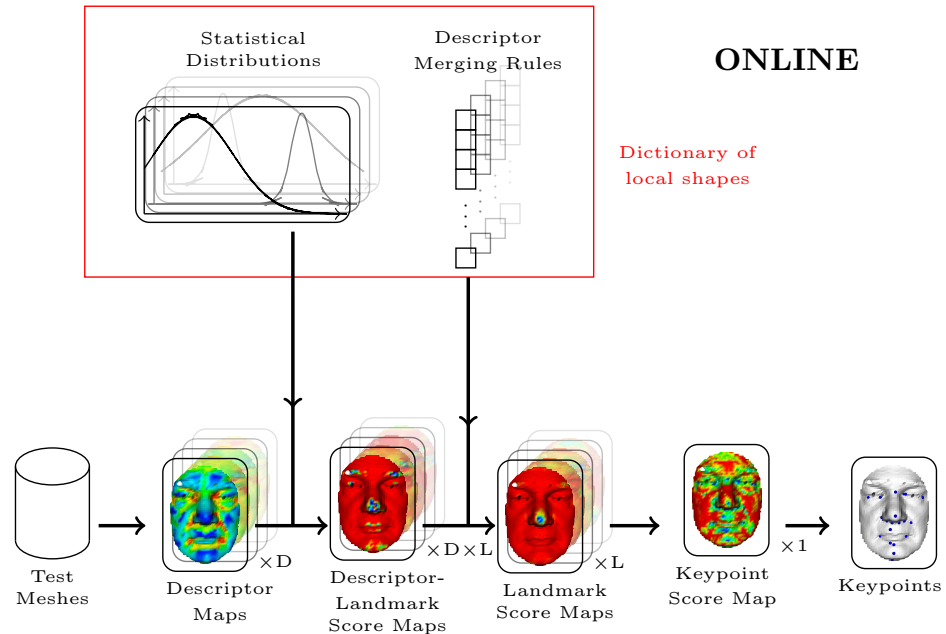


Figure 4.13: **Online process:**  $D$  descriptor maps are computed from the input mesh, each value is matched against the 14 learnt descriptor distributions to get descriptor-landmark score maps with values between 0 and 1. For each landmark, the  $D$  descriptor-landmark score maps are combined using the learnt linear weights. The 14 normalised Landmark score maps are combined into a single keypoint score map, using the maximal value (of 14 values) at each vertex. The output keypoints are the first  $0.01n$  local maxima detected on this final keypoint score map that are above some given threshold (0.85).

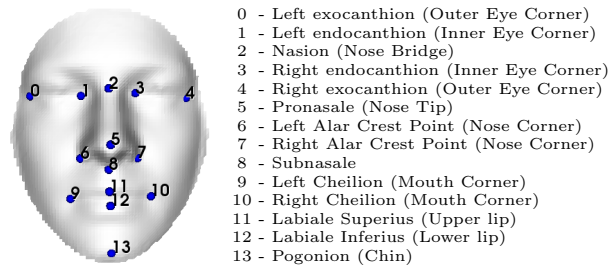


Figure 4.14: Position of the 14 landmarks used as “shapes of interest” for the training part of the system.

- A linear combination of those  $D$  maps is produced using landmark-specific weights learnt in the offline part. The result is one landmark score map per shape of interest. These maps are then normalised over the mesh to ensure the matching scores of different local shapes have the same impact on the final decision.
- All the landmark score maps are combined into one final keypoint score map, by using the maximum value (over all 14 landmark dictionary shapes) for each vertex (see figure 4.17).
- The keypoints are defined as the strong local maxima on this final map. An empirical threshold of 0.85 is used to discard weak candidates. Only the first  $0.01n$  best values are kept as output.

The variation of the descriptors at known landmark locations is learnt in the offline part by fitting an idealised distribution (a Gaussian except for the shape index where an inverse Gaussian is preferred) to the training data. The weights used to combine landmark score maps are defined using a Linear Discriminant Analysis (LDA) over a population of neighbouring and non-neighbouring vertices, relative to the relevant landmark. The population of neighbouring vertices is defined as those at a distance less than 5 mm from the specified landmark on all facial meshes in the training set. The population of non-neighbouring vertices is constituted of those between 15 and 45 mm from the same landmark (see figure 4.16 for the upper-lip landmark). These empirical radii have been selected to get two populations of manageable size on faces.

LDA applied to these two classes (neighbouring and non-neighbouring) returns the direction in  $D$ -dimensional score space that best separates the two sets. A schematic representation of the expected distributions of the two classes in an ideal case is presented in Figure 4.15. The real distributions observed for the nose tip landmark are also presented for comparison. The hypercube is projected using a naive linear classifier (the hypercube diagonal) (see Figure 4.15b) and LDA (see Figure 4.15c).

## 4.6.2 Experiments

**Descriptor and scale evaluation** For these experiments a shortlist of 10 descriptors were selected including two histogram descriptors (see 4.2 for more details):



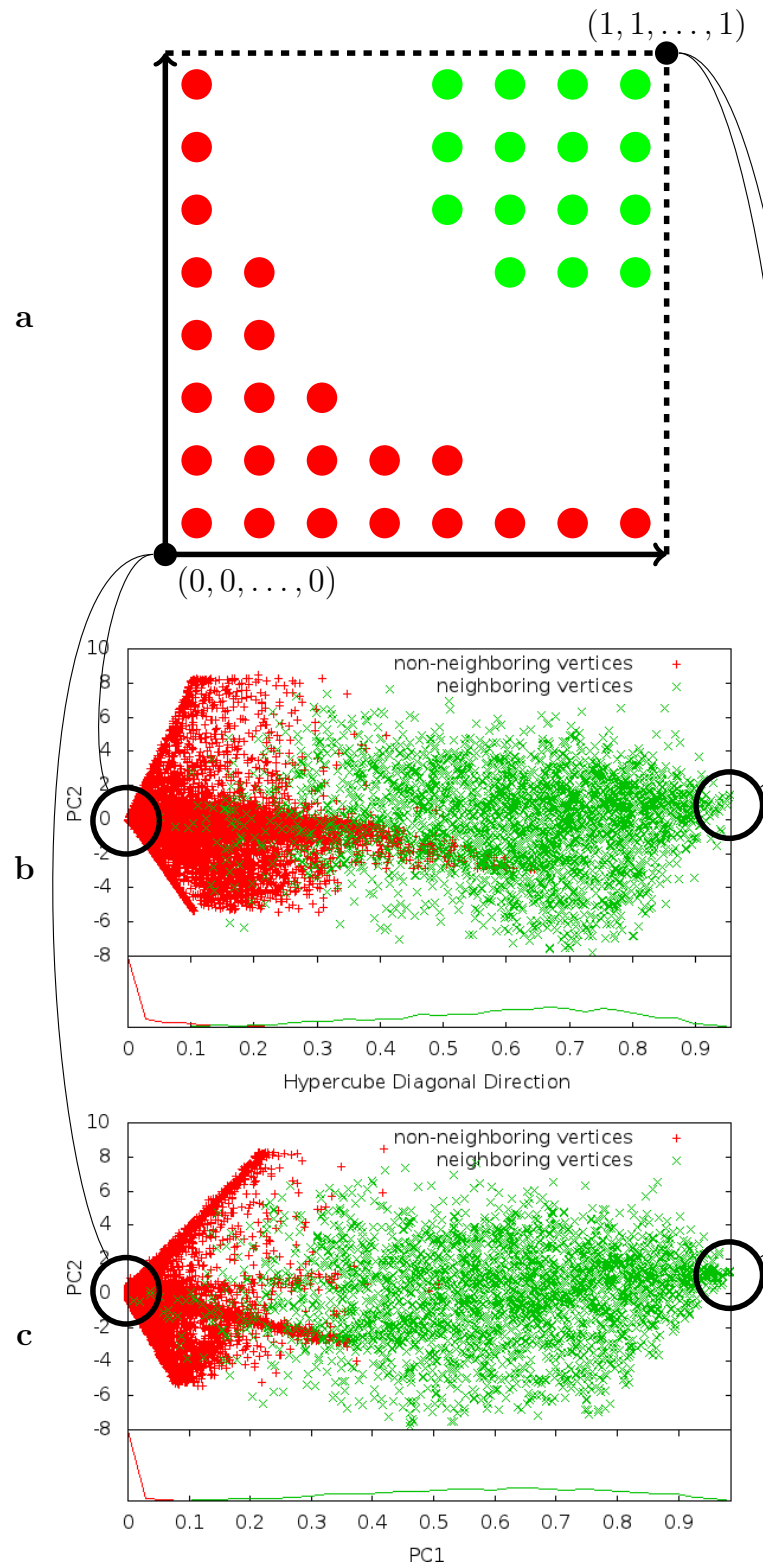


Figure 4.15: (a) Schematic representation of the two class distributions inside the score-hypercube for an ideal situation where the classes are separable by a single continuous class boundary. (b-c) Real capture of the two classes inside the DL-score hypercube for the nose tip landmark projected along (b) the hypercube diagonal direction, vector from  $(0, 0, \dots, 0)$  to  $(1, 1, \dots, 1)$ , and (c) the extracted LDA direction.

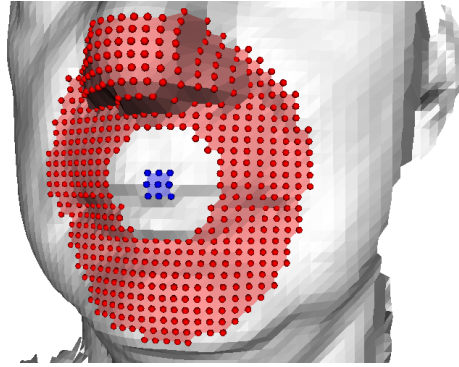


Figure 4.16: Example of class generation for LDA. Showing neighbouring (blue) and non-neighbouring (red) vertices for the upper-lip landmark on one face of the training set.

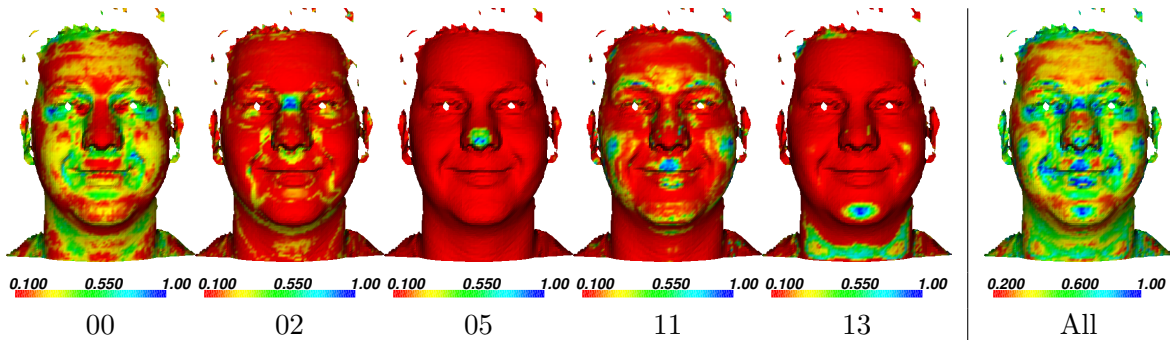


Figure 4.17: Example of normalised landmark score maps for landmarks 00 (left outer eye corner), 02 (nose bridge), 05 (nose tip), 11 (upper lip) and 13 (chin). The final keypoint score map, computed for the same subject using the 14 landmark descriptor maps, is shown on the right.

- First principal curvature ( $k_1$ )
- Second principal curvature ( $k_2$ )
- Gaussian curvature ( $K$ )
- Mean curvature ( $H$ )
- Shape Index ( $SI$ )
- Log Curviness ( $LC$ )
- Distance to Local Plane ( $DLP$ )
- Local Volume ( $VOL$ )

- Spin Image Histogram (SIH)
- Spherical Histogram (SH)

Those descriptors are computed over a range of different scales. The histogram descriptors are defined with 4 different bin sizes (from 2.5 to 10 mm). The others are defined with 6 different neighbourhood sizes (5, 10, 15, 30, 45 and 60 mm).

Using all the descriptors at all the scales and resolutions (Configuration 1) would be very time consuming and impractical for real-world applications. The most costly parts of the computation are the principal curvatures  $k_1$  and  $k_2$  over different scales, the computation of the histograms ( $\mathcal{O}(n^2)$ ) and the computation of the matching scores, as the number of maps is proportional to both the number of descriptors and the number of landmarks. In table 4.1, the weights returned by LDA for each landmark are given. The higher the value is, the more discriminating the descriptor is. From this table, the best descriptor and the best size of neighbourhood can be observed. They are represented by dark blue cells in the column corresponding to a given landmark. There is no common set of descriptors that works for all the landmarks. For example, not very salient points like the upper and lower lips (11 and 12) will prefer histogram descriptors to curvature descriptors, the corners of the mouth (9 and 10) will prefer to use the Shape Index with a small neighbourhood (15 mm), while the tip of the nose which is salient will best be detected with a bigger neighbourhood (60 mm). Figure 4.2 shows the mean results per neighbourhood size. It can be seen that the best neighbourhood size for our data resolution is around 15 mm. For the spherical histogram (SH) and the spin image histogram (SIH), the best bin dimension is between 5 and 7.5 mm. Regarding the type of descriptor, it can be seen in table 4.3 that the distance to local plane (DLP) often gives bad support to the distinction between neighbouring and non-neighbouring vertices when looking at relatively flat landmark local shape. The spin-image histogram (SIH), the Shape Index (SI) and the local volume are usually the more supportive descriptors for this set of landmarks.

From these observations, a new experiment (Configuration 2) is conducted using only one neighbourhood size (15 mm) for all scalar descriptors, and one bin size for the histogram descriptors (5 mm). In total, this comprises 10 descriptor maps and 140 descriptor-landmark score maps (far less than the original 672 of configuration 1).

Table 4.1: Snapshot of the weights corresponding to the first principal direction of the LDA for each of the 14 landmarks described in figure 4.14 (0 to 13). Dark blue cells represent the highest contribution per column. Red cells represent negative contributions (the minus symbol is omitted).

Size	Desc.	00	01	02	03	04	05	06	07	08	09	10	11	12	13	All
2.5 mm	SIH	.03	.08	.00	.10	.02	.04	.00	.04	.00	.03	.03	.01	.03	.11	.04
	SH	.00	.02	.00	.02	.02	.19	.00	.01	.02	.00	.01	.00	.00	.05	.01
5 mm	SIH	.04	.02	.04	.05	.04	.02	.04	.00	.09	.00	.00	.28	.44	.29	.09
	SH	.01	.00	.00	.01	.01	.00	.02	.00	.00	.04	.02	.01	.01	.37	.03
7.5 mm	SIH	.02	.03	.01	.02	.01	.00	.04	.01	.11	.02	.00	.40	.00	.09	.04
	SH	.05	.00	.01	.01	.04	.15	.01	.01	.00	.01	.00	.00	.07	.23	.04
10 mm	SIH	.02	.00	.08	.00	.00	.15	.02	.04	.05	.05	.05	.05	.20	.08	.04
	SH	.08	.01	.01	.00	.05	.12	.03	.03	.00	.01	.00	.00	.00	.00	.00
5 mm	k <sub>1</sub>	.00	.02	.02	.04	.00	.00	.05	.05	.01	.06	.10	.03	.09	.00	.02
	k <sub>2</sub>	.02	.04	.01	.01	.01	.00	.03	.03	.01	.02	.02	.01	.03	.00	.00
	H	.07	.02	.02	.00	.07	.00	.05	.00	.06	.01	.00	.05	.04	.01	.01
	K	.03	.06	.02	.05	.03	.00	.02	.02	.03	.03	.01	.01	.00	.00	.01
	SI	.01	.03	.01	.04	.01	.01	.02	.02	.01	.00	.04	.00	.00	.01	.00
	LC	.04	.00	.02	.00	.04	.00	.01	.06	.01	.09	.04	.01	.01	.00	.02
	Vol	.04	.00	.00	.01	.02	.00	.25	.04	.00	.02	.02	.00	.04	.00	.03
	DLP	.10	.10	.01	.02	.12	.00	.03	.03	.02	.06	.07	.11	.14	.00	.06
15 mm	k <sub>1</sub>	.01	.07	.05	.06	.00	.14	.00	.02	.04	.06	.06	.00	.17	.04	.02
	k <sub>2</sub>	.03	.39	.03	.14	.05	.00	.17	.22	.07	.07	.07	.02	.07	.01	.06
	H	.04	.00	.04	.02	.06	.01	.05	.22	.04	.02	.00	.00	.01	.04	.02
	K	.05	.07	.17	.04	.04	.00	.03	.02	.23	.07	.06	.01	.18	.00	.01
	SI	.08	.11	.08	.28	.08	.00	.06	.06	.00	.51	.57	.03	.03	.00	.13
	LC	.02	.03	.03	.02	.04	.00	.00	.03	.03	.04	.04	.00	.08	.00	.01
	Vol	.13	.01	.02	.12	.12	.16	.07	.11	.01	.11	.09	.03	.09	.02	.08
	DLP	.08	.00	.00	.04	.08	.00	.06	.13	.00	.08	.07	.00	.14	.00	.01
30 mm	k <sub>1</sub>	.00	.01	.02	.03	.01	.01	.00	.02	.04	.00	.01	.03	.05	.18	.01
	k <sub>2</sub>	.00	.02	.04	.08	.01	.01	.02	.02	.00	.03	.05	.05	.12	.01	.00
	H	.15	.09	.01	.10	.12	.00	.09	.11	.04	.05	.04	.05	.12	.08	.02
	K	.12	.00	.00	.01	.17	.03	.00	.01	.02	.02	.00	.02	.01	.20	.01
	SI	.17	.05	.08	.00	.15	.00	.05	.08	.02	.03	.02	.01	.04	.06	.04
	LC	.04	.01	.05	.00	.03	.01	.03	.02	.04	.06	.09	.08	.07	.00	.01
	Vol	.10	.03	.01	.08	.11	.00	.05	.05	.06	.06	.04	.01	.06	.02	.04
	DLP	.07	.41	.03	.33	.11	.05	.03	.05	.02	.04	.00	.01	.08	.01	.05
45 mm	k <sub>1</sub>	.05	.04	.01	.06	.05	.01	.00	.02	.00	.05	.05	.04	.01	.01	.00
	k <sub>2</sub>	.02	.00	.00	.02	.02	.00	.07	.05	.05	.01	.03	.08	.01	.01	.01
	H	.02	.01	.00	.03	.02	.01	.01	.02	.04	.12	.10	.01	.19	.00	.03
	K	.00	.00	.00	.00	.01	.00	.05	.02	.03	.01	.01	.06	.08	.01	.01
	SI	.04	.03	.04	.06	.02	.00	.01	.05	.00	.00	.00	.00	.02	.00	.02
	LC	.01	.03	.02	.03	.00	.00	.16	.09	.00	.15	.09	.10	.18	.02	.06
	Vol	.00	.01	.00	.01	.00	.00	.04	.06	.03	.02	.02	.03	.05	.01	.02
	DLP	.03	.14	.03	.02	.06	.24	.01	.01	.03	.15	.13	.03	.04	.00	.00
60 mm	k <sub>1</sub>	.08	.02	.04	.00	.08	.00	.05	.04	.00	.00	.00	.00	.04	.02	.01
	k <sub>2</sub>	.00	.15	.01	.09	.00	.00	.04	.07	.02	.03	.03	.01	.03	.04	.03
	H	.01	.03	.00	.00	.02	.00	.00	.02	.00	.02	.04	.00	.04	.00	.00
	K	.01	.05	.01	.04	.01	.00	.03	.00	.01	.00	.03	.00	.04	.02	.00
	SI	.09	.02	.04	.06	.08	.00	.01	.00	.02	.00	.00	.02	.05	.04	.01
	LC	.02	.02	.03	.01	.03	.00	.04	.07	.02	.03	.05	.02	.03	.04	.01
	Vol	.06	.00	.00	.00	.05	.02	.01	.03	.03	.05	.07	.01	.07	.00	.03
	DLP	.02	.03	.01	.02	.04	.33	.03	.00	.00	.05	.07	.01	.11	.01	.00

Table 4.2: Summed value per neighbourhood size of the weights returned by LDA (see table 4.1). The descriptiveness of the descriptors peaks around 15 mm for the selected landmarks.

Size	00	01	02	03	04	05	06	07	08	09	10	11	12	13	All
5 mm	.01	.10	.10	.07	.01	.01	.21	.01	.04	.10	.12	.07	.07	.02	.01
15 mm	.28	.41	.45	.54	.27	.33	.34	.51	.45	.53	.60	.09	.08	.03	.34
30 mm	.22	.38	.18	.20	.24	.12	.18	.14	.21	.01	.06	.08	.21	.12	.16
45 mm	.21	.01	.05	.10	.22	.26	.07	.12	.00	.15	.15	.10	.20	.01	.08
60 mm	.03	.14	.00	.01	.04	.35	.01	.07	.08	.04	.06	.04	.01	.05	.06

Table 4.3: Summed value per descriptor of the weights returned by LDA (see table 4.1).

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	All
DLP	.07	.19	.00	.35	.13	.14	.18	.24	.04	.00	.08	.08	.27	.02	.02
H	.11	.05	.08	.11	.12	.00	.10	.38	.10	.18	.09	.01	.39	.14	.01
K	.03	.17	.20	.12	.12	.04	.02	.05	.21	.08	.10	.07	.04	.19	.01
SH	.15	.01	.04	.03	.12	.20	.03	.03	.02	.04	.01	.01	.06	.55	.09
SI	.23	.14	.18	.23	.19	.02	.16	.17	.03	.48	.60	.06	.09	.10	.19
SIH	.07	.15	.14	.18	.09	.23	.12	.08	.26	.10	.09	.73	.67	.22	.22
k <sub>1</sub>	.15	.03	.16	.01	.16	.14	.11	.07	.05	.06	.10	.05	.18	.12	.07
k <sub>2</sub>	.03	.56	.00	.14	.05	.00	.08	.25	.03	.00	.02	.03	.03	.01	.09
LC	.15	.11	.15	.06	.16	.00	.13	.13	.08	.10	.05	.14	.32	.02	.10
Vol	.35	.07	.02	.20	.32	.19	.44	.31	.13	.29	.27	.08	.32	.01	.21

### 4.6.3 Results

Figure 4.18 shows examples of final keypoint score maps computed for configuration 1 and 2 and the corresponding detected keypoints. A visual check of these results gives a lot of indications about the system and its drawbacks. The central scan for example contains lot of false positive keypoints in the hair areas. However, in order to evaluate the results for the whole database, quantitative cost functions have to be used.

Here results for landmark retrieval rates and keypoint repeatability are presented.

**Landmark Retrieval** To evaluate the rate at which keypoints are localised near defined landmarks, the percentage of face meshes in which a keypoint is present in a sphere of radius  $R$  from the manually labelled landmark is computed. As there is no clear definition about what distance error should be considered for a match, this percentage is computed for an increasing acceptance radius ranging from  $R = 2.5$  mm to  $R = 25$  mm. Results for configuration 1 and 2 are given in figure 4.19. With configuration 2, at 10 mm, the nose tip is present in the detected keypoints 99.47% of the time, and the left and right inner eye corners in 90.50% and 92.56% of the cases. It can be seen that this method will not succeed in detecting all potential landmarks in all facial meshes. However, it aims to provide



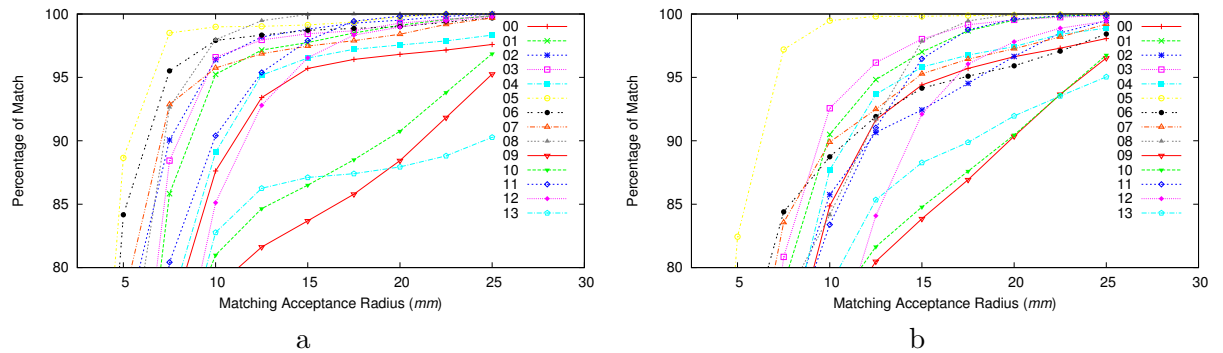


Figure 4.19: Matching percentage per landmark (0-13) with an increasing matching acceptance radius on the FRGC v2 test set. (a) using all descriptors (Configuration 1), (b) using a subset of descriptors (Configuration 2).

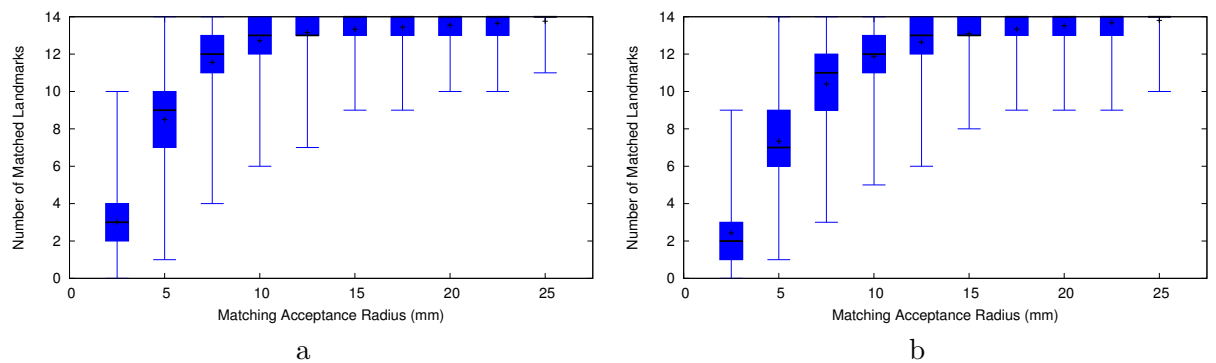


Figure 4.20: Number of matching landmark per file on the test part of the FRGC v2 database. (a) using all descriptors (Configuration 1), (b) using a subset of descriptors (Configuration 2).

tween shapes of interest that are linked to identity and those linked to change in expression or other variations.

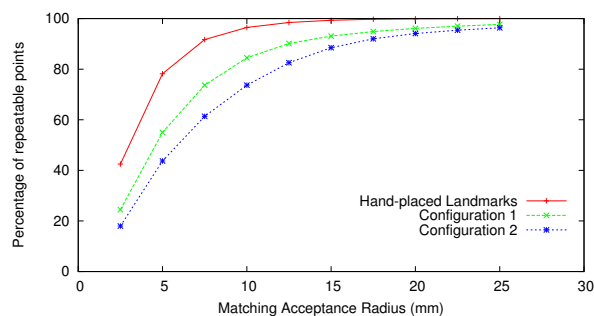


Figure 4.21: Percentage of points repeatable after registration at an increasing matching acceptance radius. The measure of the human hand-placed landmarks is used as a baseline.



**Computation Time performance** The per-scan computation time of the online process was on average 7.75 seconds on a laptop processor Intel Core I3 M350 (the mean number of vertices in our meshes being 6392). The more computational part is the neighbourhood and curvature computation (2.04 seconds using Yoshizawa method using SVD decomposition [Yoshizawa et al., 2008]) and the computation and projection of the histograms for each vertex (3.85 seconds). Using fewer descriptors on smaller meshes (using automatic face cropping) is a good way to reduce computation time. Later improvement of the curvature computation method (see Digression p. 121) and discarding the histogram descriptors reduced the time to under one second per scan for this particular method.

#### 4.6.4 Conclusion

A simple method has been proposed to deal with the problem of combining descriptors to detect unlabelled points of interest (keypoints) on 3D objects. This method gives interesting results on 3D faces as it detects weak features, but presents some limitations in its current state, such as the fact that only linear combinations are used for the descriptors or that it can be computationally expensive if too many descriptors are considered (especially histogram descriptors). However, a good point of our method is that it doesn't assume that the detected points should have an extremal value over a descriptor map. Instead, it assumes that the matching score of this descriptor against a learnt distribution should be maximal. Furthermore, this technique is very generic, as the set of descriptors, the sizes of the neighbourhoods and the dictionary of local shapes can be changed easily. Our method could therefore be used on other types of 3D objects without any modification.

The behaviour of different local descriptors competing with each other at different scales has also been studied. It provides us with better indications of which descriptor should be used with which parameters in order to detect each of the 14 common landmarks used as shapes of interest in our experiments (see table 4.1).



## 4.7 Method 2: Non-linear Combination of Descriptor-Landmark Score Maps

In order to improve the proposed method several paths can be taken. An obvious one is to use better, cheaper and more descriptive descriptors. Another is to try to better use the information present within a set of descriptor-landmarks score maps we already have.

To answer some of the limitations of method 1, a second method is proposed taking advantage of a non-linear classification technique, Adaboost. The aim here is to extract and use even more discriminative information from the set of descriptors. For this method, the configuration 2 (single scale) explained in the previous section is reproduced with two differences:

- When training the system using the two classes neighbouring and non-neighbouring vertices, a boosting technique is used to learn weak classifiers.
- In the online part, the weak classifiers are used to obtain a matching score per local shape for each vertex.

### 4.7.1 Boosting Technique

The boosting technique used here is really simple. The scalar component of the vectors (descriptor scores) are treated independently. Each weak classifier is composed of the following:

- The index  $i$  of the descriptor, i.e. the  $i^{th}$  element of the descriptor feature vector.
- A threshold  $T$  splitting the descriptor's 1D scalar space into two.
- A direction  $dir$  -1 ( $t < T$ ) or 1 ( $t > T$ ) stating which side of the space corresponds to the "match" class.
- A scalar  $\alpha$  describing the weight associated with this single weak classifier.

**Training** For the training of the weak classifiers, a simple Adaptive Boosting technique (or Adaboost [Freund and Schapire, 1997]) is used. At each additional iteration, the weights are assigned to the training data according to the classification error using the existing weak classifiers. Each dimension of the feature space is divided in small steps. The triplet

dimension/threshold/direction that best classifies the training set with the new weights is selected as the new classifier. To search the threshold along one dimension, a simple two-level coarse to fine approach is used. The range of observed values  $[min, max]$  is divided into 200 steps. Once the best step  $k$  is found using this discretisation, the range  $[k - 1, k + 1]$  is divided again in 50 steps. The new best step  $k'$  that is found is used as the candidate threshold for this dimension.

The weight of a current best classifier is defined relative to the current error in classification:

$$\alpha = \frac{1}{2} \log \frac{1 - err}{max(err, \epsilon)}$$

The influence of each input point is updated at each iteration by reducing the weights of the well classified points and retaining the weights of badly classified points:

$$w_{ij} = \begin{cases} w_{ij} \cdot \exp(-\alpha) & \text{if good classification} \\ w_{ij} & \text{otherwise} \end{cases}$$

**Projection** For the online part of the process, each input vertex has a vector of values, each value corresponding to one descriptor. The score for each vertex is computed from the scalar vector as explained in Algorithm 1. Instead of using the classification result ( $-1$  or

---

**Algorithm 1:** Boosting score computation.

---

**Data:** vector of values  $V$ , vector of weak classifier  $W$

**Result:** scalar  $score$

$score = 0.0$

**foreach** weak classifier  $(i, T, dir, \alpha)$  in  $W$  **do**

**if**  $(dir.V[i] > dir.T)$  **then**  
    |  $score += \alpha$   
    **else**  
    |  $score += -\alpha$

**return**  $\frac{1+score}{2}$

---

1), the output score is returned as a continuous value mapped into  $[0, 1]$ .

## 4.7.2 Number of Classifiers

Study of the variation of the number of classifiers on the training set shows that a plateau is reached relatively quickly (See figure 4.22). A short cross validation on the training set

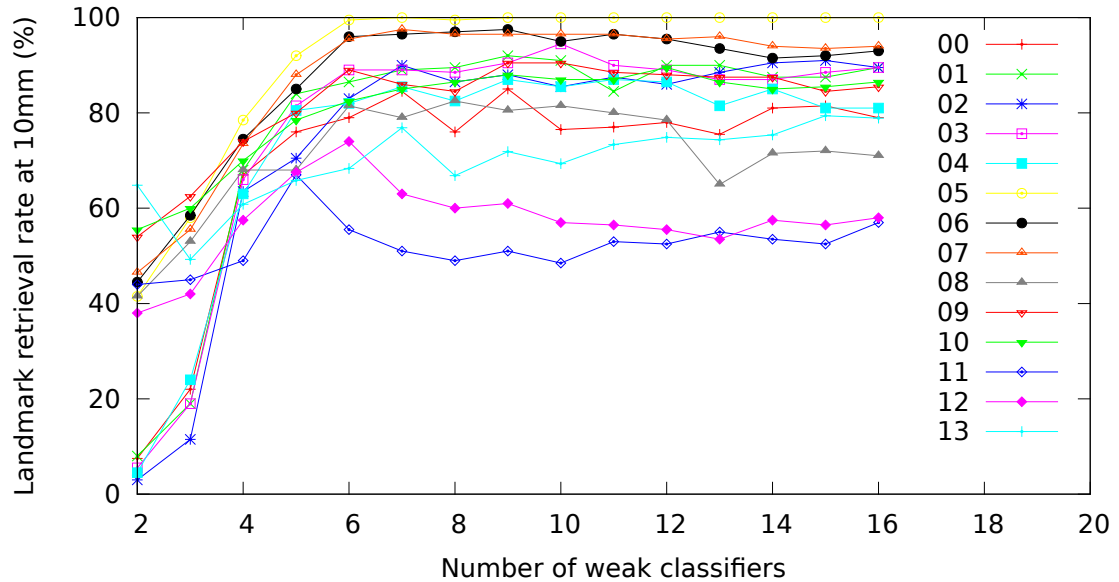


Figure 4.22: Variation of the retrieval rate with a 10 mm error acceptance radius for different numbers of classifiers on the training set.

showed that a upper limit on the number of classifiers is not really important: the system doesn't seem to over-fit the training data even with 160 classifiers (compared to the 10 descriptors used). This can be explained by the fact that the classifiers can be very similar to each other with the Adaboost technique. From figure 4.22, the following numbers of classifiers are selected:

Shape Id	00	01	02	03	04	05	06	07	08	09	10	11	12	13
Nb. of Classifier	9	9	7	9	9	9	9	9	9	9	9	5	6	7

### 4.7.3 Computing Distributions Separation

In order to compare quantitatively different classification techniques, a cost function is needed. Given two distributions densities  $D_0$  and  $D_1$  (respectively non-matching and matching) over a domain  $[0, 1]$  how can the separation between the two distributions be quantified? There are many possible solutions to this simple problem. In our particular case the distribution is not necessarily smooth nor continuous, therefore looking at the problem at one

abscissa is not meaningful. Because our two distributions are labelled, non-symmetrical relations can be introduced. For every threshold  $t$  set between  $[0, 1]$ , the following can be computed:

$$\begin{aligned} \text{True Negative Rate:} \quad & TNR(t) = \int_0^t D_0(x) dx \\ \text{False Positive Rate:} \quad & FPR(t) = \int_t^1 D_0(x) dx \\ \text{False Negative Rate:} \quad & FNR(t) = \int_0^t D_1(x) dx \\ \text{True Positive Rate:} \quad & TPR(t) = \int_t^1 D_1(x) dx \end{aligned}$$

Obviously our aim is to be able to detect methods with minimal  $FNR$  and  $FPR$  values. By integrating over all possible  $t$ , a global notion of the intersection between the two distributions is defined:

$$\begin{aligned} I(D_0, D_1) &= \int_0^1 FNR(t).FPR(t) dt \\ &= \int_0^1 (\int_0^t D_1(x) dx) \cdot (\int_t^1 D_0(x) dx) dt \\ &= \int_0^1 \int_0^t D_1(x) \cdot (1 - D_0(x)) dx dt \end{aligned}$$

#### 4.7.4 Matching Scores vs Raw Descriptor Values

As a non-linear learning technique is now used, one might question the legitimacy of mapping the original descriptor maps to descriptor-landmark score maps. This is a very good point that might lead to further speed improvement for the online part. However using the adaboost technique on the raw data presents some drawbacks: the number of classifiers increases more quickly with the number of dimensions. Indeed in the score space and using a given direction, the non-matching features are usually present in roughly one direction (the direction of the space origin). In the raw descriptor space the area that needs contouring is surrounded by non-matching features in all directions, which means that a larger number of hyperplanes is required to get the same performance. This might not be too expensive for the online process but becomes very costly for the training. This difference can be observed in figure 4.23 where the use of adaboost on score is compared to the use of adaboost on raw descriptors (the histograms being excluded in both cases). The distribution appears to be far less smooth for the raw descriptors if the same number of classifiers is used.

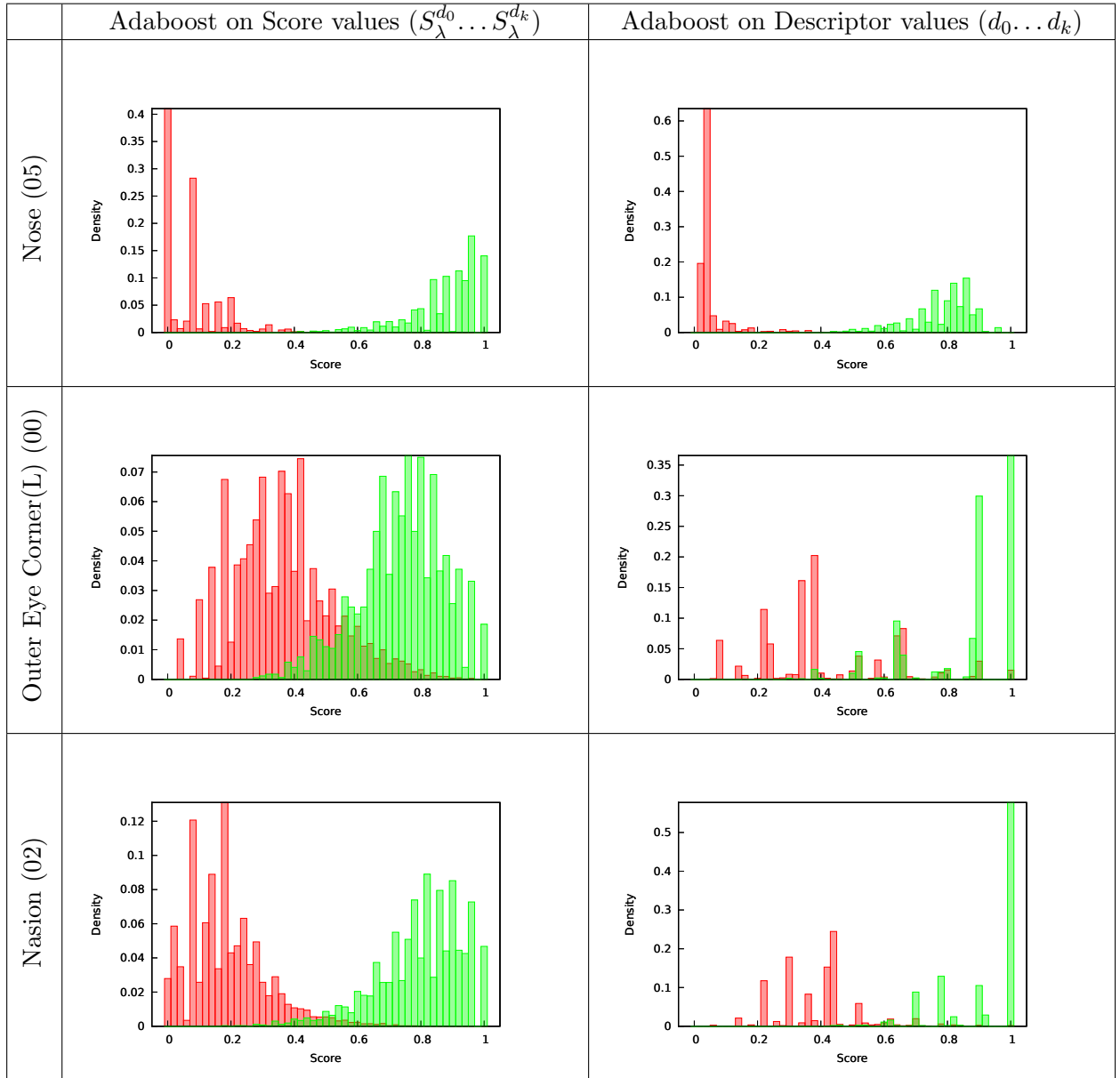


Figure 4.23: Examples of comparative distributions of neighbouring (green) and non-neighbouring (red) vertex classification for 3 landmarks using the adaboost method (20 hyperplanes) on the score values for each feature (left column) and on the raw descriptor values (right column).

Another advantage of using scores is that the histogram descriptors can be represented in the same fashion as scalar descriptors.

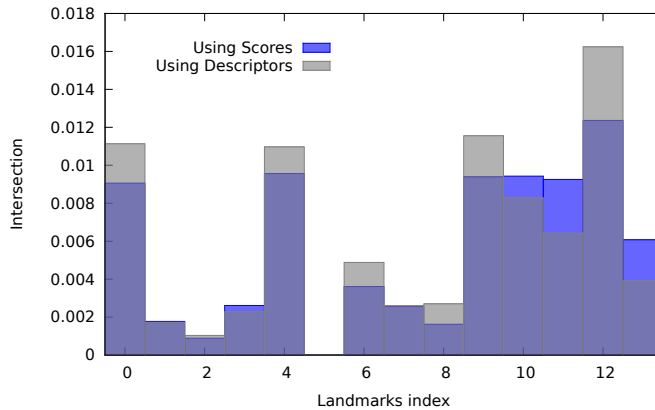


Figure 4.24: Comparison by landmark of the intersection between neighbouring and non-neighbouring vertices distributions. Both methods use Adaboost with 20 weak classifiers. The input data for the first method are matching scores. The input of the second method are the raw values of the descriptor. Because it is not trivial to use the histogram as raw data for the adaboost on raw values, the histogram descriptors have here been withdrawn from both experiments.

#### 4.7.5 Comparisons with LDA Scoring

A first way of comparing the LDA version to the Adaboost version is to see how well each of them separate the two classes used in training.

Figure 4.25 shows the different scoring results of both LDA and Adaboost methods when trying to differentiate neighbouring and non-neighbouring vertices. It can be seen that the neighbouring class is much more scattered with the LDA scoring than with adaboost. In Figure 4.26 the intersection of the density distributions are compared for each of the 14 shapes of interest. In all cases the adaboost performs clearly better than the LDA classification.

The two methods can also be compared when looking at the end result of the system: the keypoint detection. In term of repeatability and number of retrieved landmark positions both methods give similar results (see figure 4.27-a and 4.27-b). However, when looking at single landmark retrieval rates (see Figure 4.28) the adaboost method performs better for the same set of descriptors.

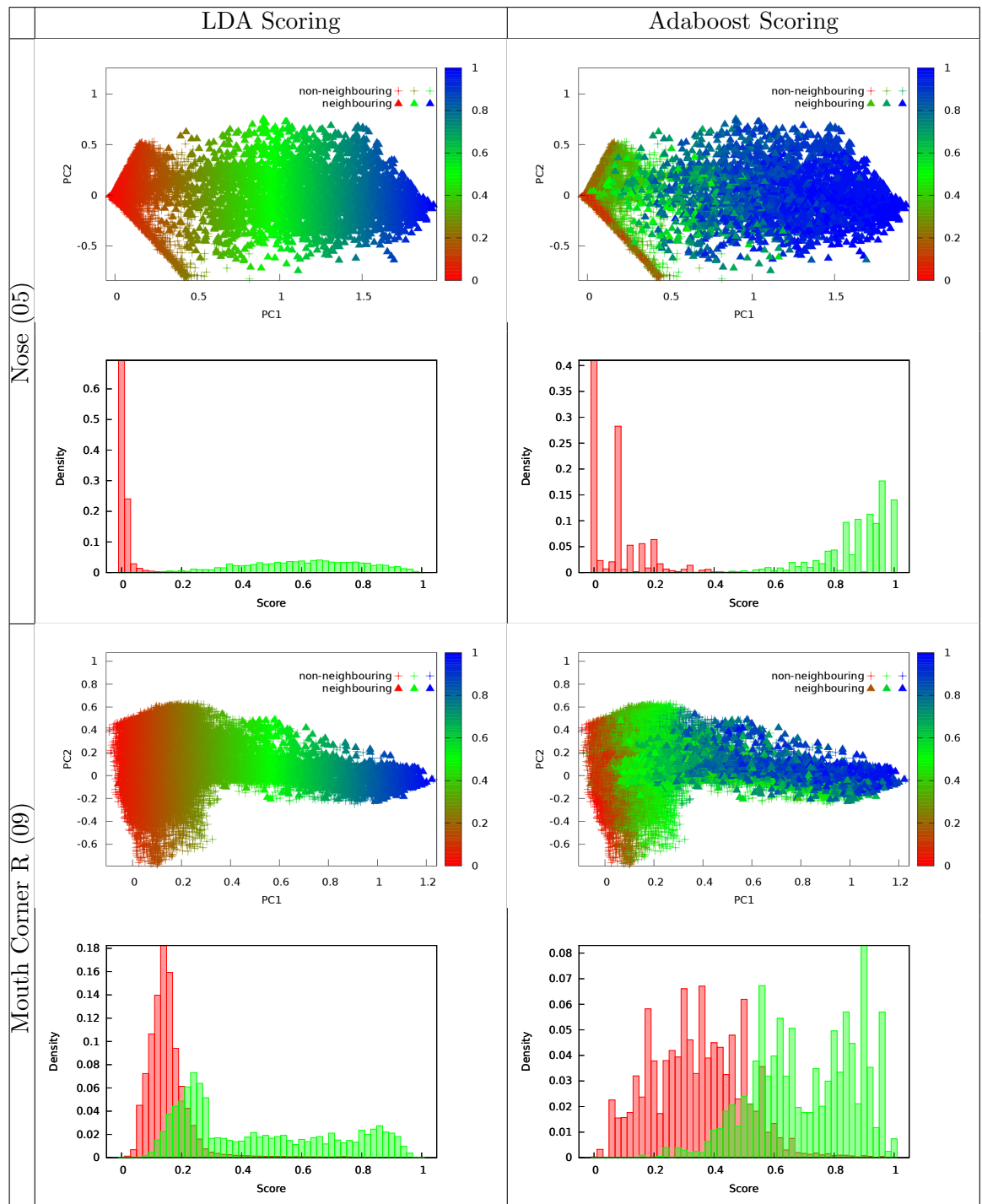


Figure 4.25: Examples of the differences in scoring using the LDA and the Adaboost method (20 classifiers) for two landmarks. The upper figure shows the feature points of the two types in the two first components of the LDA basis with scoring represented as colour. The figure below represents the density of the matching and non-matching classes through the scoring spectrum.

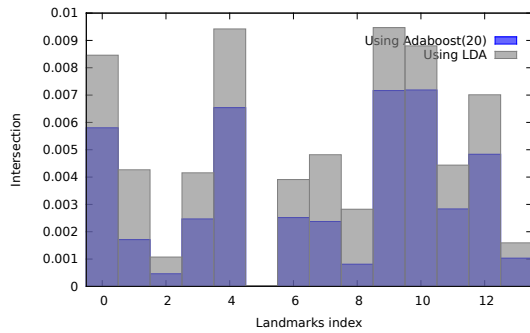


Figure 4.26: Comparison by landmark of the intersection between neighbouring and non-neighbouring vertices distributions. Adaboost is better at separating the two classes than LDA (intersection closer to zero).

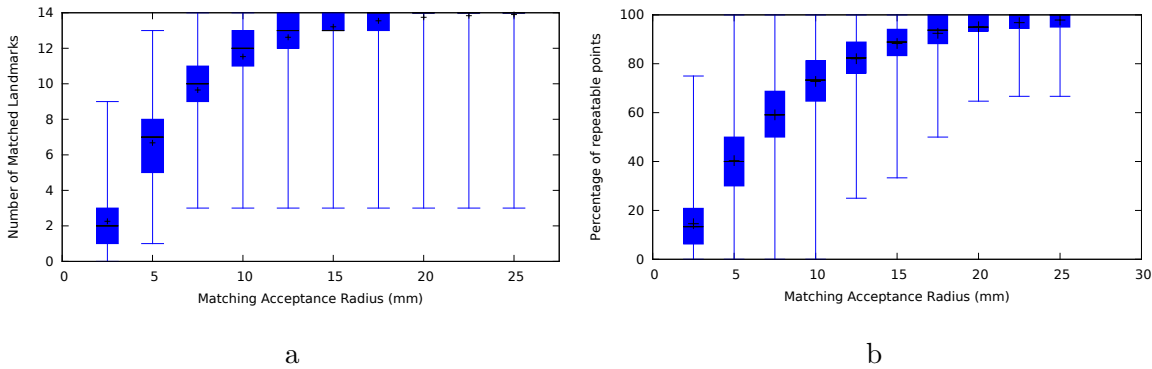


Figure 4.27: Number of retrieved landmarks (a) and repeatability (b) for an increasing matching acceptance radius on the FRGC v2 test set using method 2 (configuration 2).

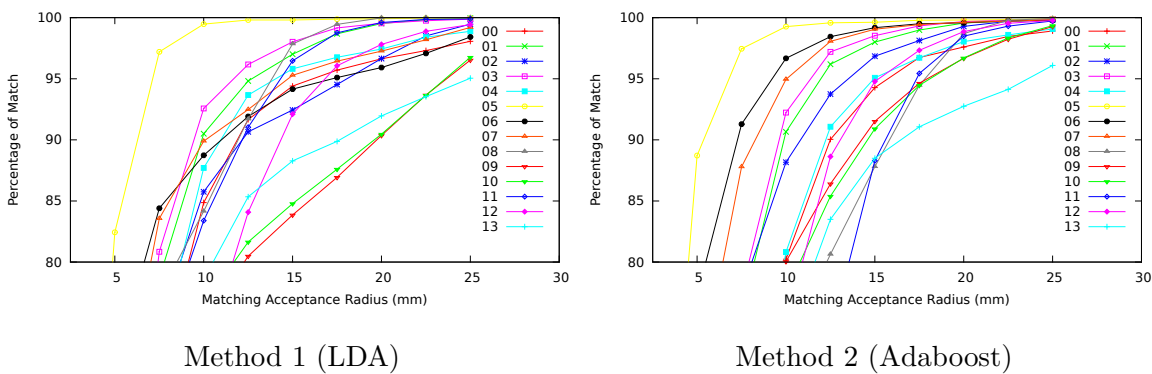


Figure 4.28: Matching percentage per landmark (0-13) with an increasing matching acceptance radius on the FRGC v2 test set. Adaboost gives better results than LDA for the same configuration (configuration 2).



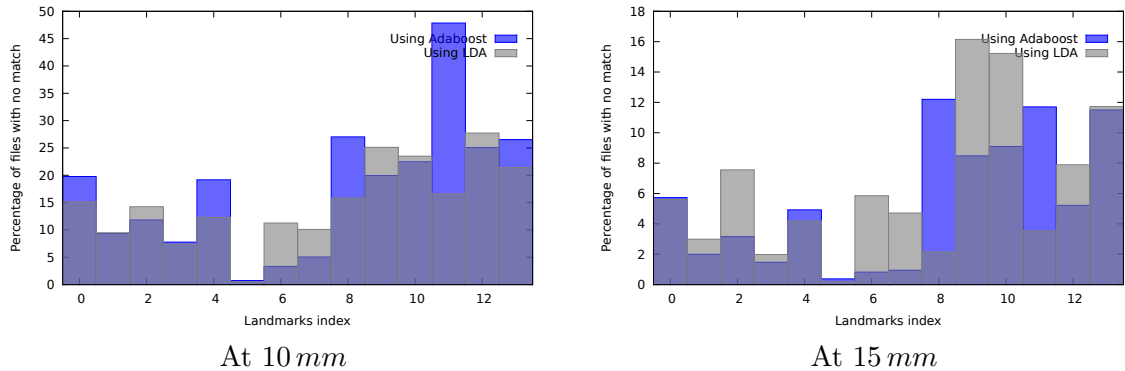


Figure 4.29: Retrieval error rates for the Adaboost and LDA merging method. The results are plotted for the 14 landmarks with two different acceptance radii.

## 4.8 Conclusion

In this chapter two new methods for landmark candidate detection have been presented. The system propagates the assumptions made when computing the descriptors but doesn't assume additional facts about the input data:

- The system is pose-invariant if the descriptors used are pose-invariant.
- The system would be robust to occlusion if the descriptor were robust to occlusion (this is not the case in our experiments). It is possible to imagine descriptors that cope with planar occlusion by extracting curve related descriptors instead of surface related descriptors from the local neighbourhood.
- The system is intrinsically landmark independent: the system is completely oblivious of the nature of the landmark used in training.
- The system is not sequential: it is not required to find the landmark candidates in a particular order. Every shape of interest can be treated in parallel and finding good candidates for one at the scoring stage does not depend on finding good candidates for another one.
- The system is able to learn patterns that describe weak local features where a human would struggle to design rules (e.g. for the corner of the mouth or the outer corner of the eyes).

While being more fuzzy (many-valued) compared to expert system methods, we believe that this kind of approach is necessary to deal with uncontrolled input data. In particular, it is more likely to be successful for non cooperative face pre-processing where there are great uncertainties about what is present in the query mesh.

Scoring all the individual vertices can be time consuming if computationally expensive descriptors are used. We noticed that for that kind of approach, it is best to avoid histogram descriptors. Weak scalar descriptors seem to contain enough information to reduce greatly the number of candidates for the landmarking process.

For our objectives, this method is greatly restricted by the intrinsic limitations of the descriptors we used: they are not occlusion invariant and incomplete local neighbourhoods lead to spurious descriptor values. For example, in a profile view, when a vertex is near the border of the mesh, its neighbourhood is incomplete and the descriptors computed on this neighbourhood are likely to be noisy. It might be interesting to develop different specialised descriptors that can detect keypoints near the border of the mesh using, for example, extracted 2D curves along the direction of occlusion. Combining the score using these new descriptors can easily be done within our framework.

In figure 4.30 an example of detection is presented on an input containing several faces. At this stage the system does not require any form of knowledge about the number of faces present in the scene. It can be seen that on the face viewed in profile, the system doesn't succeed in retrieving all features near the occlusion plane (e.g. nasion, subnasale). It is our belief that those points may be retrieved using special descriptors. Another interesting pattern in this example is that despite the fact it was not learnt in the training, the ophrion (mid-point of the fore-head) is detected on all four faces. This suggests that other landmarks might be good additions to the dictionary of local shape if it is possible to detect them on a large part of the database population. The advantage of our 14 landmarks is their ubiquity for all individuals in the database, making performance more easy to compute.

In our opinion, the main gain in performance in the future will come from adding new local descriptors that better deal with occlusions and profile views.

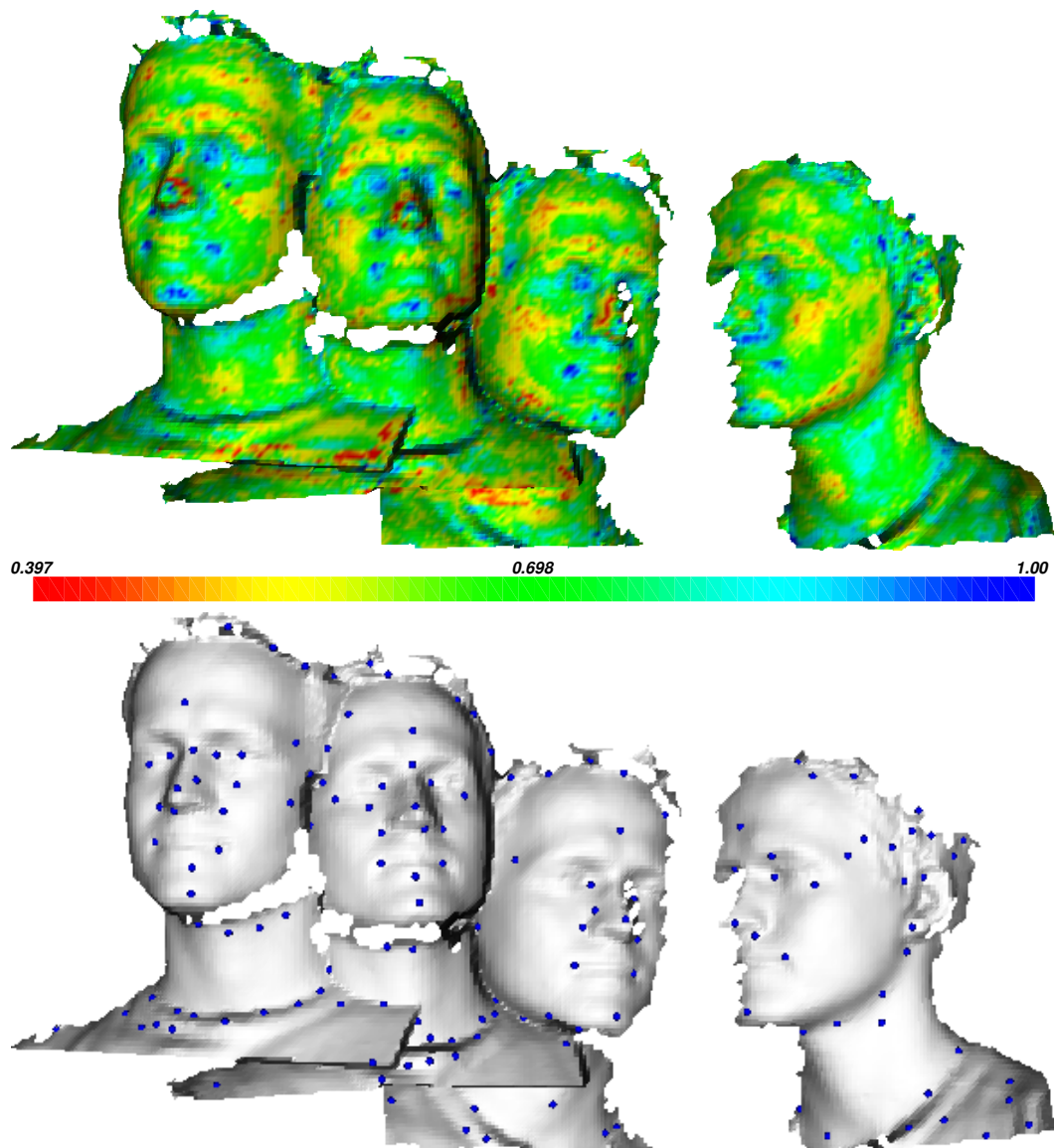


Figure 4.30: Example of keypoint detection on multiple faces (cross database experiment using training from the FRGC experiments). The faces were captured using a Cybula Ltd 3D face camera (2008) which has a lower resolution than the camera used for the FRGC database. The lips, for example are really flat in these captures and our system fails to detect them properly. When showing self occlusion, region near the vertical plane of symmetry become more difficult to detect (nasion, nose tip and subnasale). While blue regions are always visible around the eyes' outer corner, it doesn't always translate into a keypoint detection, which suggest that the local maxima detection can probably be improved. The eyes' inner corners, mouth's corners, nose's corners and chin are all correctly detected. A quite interesting pattern is that the ophrion point (middle of forehead) is detected on the four faces despite the fact the system was never trained for this landmark.



## Chapter 5

# Point Labelling using Structural Matching

We argue that for certain types of binary relations used, the label of an object is only affected by the values of the binary relations in which it is directly involved, and there is no need for the consideration of ternary and higher order relations.

---

W. J. Christmas, J. Kittler and M. Petrou in *Structural Matching in Computer Vision Using Probabilistic Relaxation* [[Christmas et al., 1995](#)]

Given a set of points on a 3D surface, can we select the ones corresponding to human-defined landmarks and determine a unique correspondence between this subset and a learnt model? In this chapter we investigate how local and configural information can be used to label a set of points using a graphical model of the class of object to be detected. An original contribution of this work is the way it looks at hypergraphs for the structural representation and its consideration of the matching systems as a succession of loose and fuzzy solution filters.

### 5.1 Introduction

Figure [5.2](#) illustrates the problem to be solved. A set of 14 landmarks on the 3D facial surface forms the basis of our graph model. Our ultimate aim is to be able to detect these landmarks

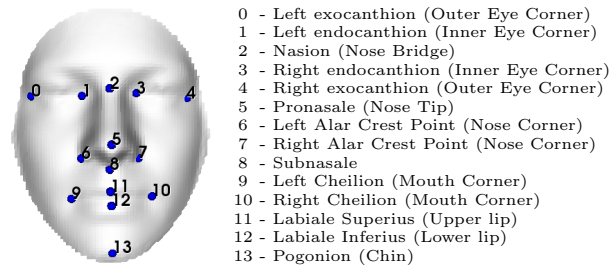


Figure 5.1: Model of the labels to be retrieved.

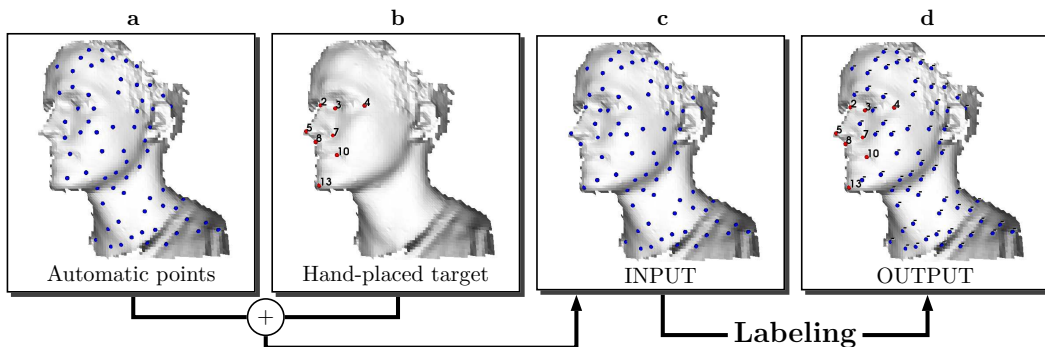


Figure 5.2: Problem to solve: Given a model and a set of points containing some hand-placed target landmarks, can the labels of the hand-placed landmarks be retrieved? Does it work when large parts of the face are missing? Here the automatic points are detected by taking the points that locally maximise the seeding score for any of the landmarks in the model.

automatically. The problem is complicated by the fact that not all of the landmarks will be visible, due to pose variations and occlusions. As a first step towards an automatic system, we want to demonstrate that if the *unlabelled* points that we want to detect are present within a large number of automatically detected points of interest, it is possible to relabel the landmarks correctly, as illustrated in figure 5.2.

**Controlled input:** By constructing artificial inputs for this problem we guarantee that the maximum landmark retrieval rate is 100% for all the targeted landmarks. This enables us:

- to evaluate the labelling system independently from any keypoint detection error,
- to allow comparison between individual landmark retrieval rates,
- to allow results comparison in the future by other researchers interested in this problem.

The main risk with mixing two types of points is the introduction of a bias towards the hand-placed landmarks that can be exploited by the system in the seeding process. We eliminate this risk by including automatic keypoints that have maximum local seeding scores. The local score of a hand-placed landmark can be, at most, the local maximum, while the local score of an automatic keypoint is *always* the local maximum. By doing this we guarantee that if a bias exists at the seeding stage, it is towards the automatic keypoints, which are incorrect solutions here. Therefore, a system using these inputs can only retrieve the landmarks using structural information, which is what we want to evaluate. The labelling of automatic keypoints (complete landmarking pipeline) is discussed in chapter 6.

**Contributions:** The contributions in terms of methodology in this chapter are:

- the use of a multi-attributed sparse hypergraph structure for data representation: while higher-degree structural matching papers have been published by other researchers during this PhD [Zass and Shashua, 2008] [Duchenne et al., 2009] [Chertok and Keller, 2010], using hypergraphs in a non-tensor form with multiple attributes per element is novel.
- the application of hypergraph matching techniques to 3D object (in particular 3D face) correspondences: all hypergraph matching techniques we have encountered are only applied to 2D image matching.
- the splitting of the matching process (usually considered as one step) in a succession of solution filters using different correspondence finding techniques.

At a practical level, our contributions are:

- an hypergraph matching relaxation algorithm alternating elimination on the hypergraph and its dual.
- a new framework for hypergraph matching based on successive correspondence filters.
- a scale-adapted rigid correspondence finder based on unit-quaternion clustering of triangle transformations.
- a scale-adapted rigid correspondence finder based on a RANSAC technique adapted for correspondence candidates.

- a proof-of-concept experiment showing that facial features can be retrieved even when a large number of false positive keypoints are present.

**Chapter structure:** In the first section, basic definition of graphs and hypergraphs will be recalled before examples of attributes that it is possible to use for nodes and hyperedges on 3D meshes are presented. In the second section, our particular problem will be explained and differentiated from other hypergraph matching problems. The basic structure of our framework is presented and some of its components explained.

Section 5.5 presents our proof-of-concept experiment confirming that this approach can cope with occlusion and pose variation. The following section studies the behaviour of different hypergraph matching techniques on synthetic data to optimise the overall matching according to some conditions (e.g. missing points, extra points, number of candidates per element, and so on). Finally, we discuss the principal issues encountered with our approach and give ideas of how it can be improved.

### Rationale

**What:** To retrieve points from the input corresponding to defined landmark labels.

**Why:** To convert previously detected keypoints into landmarks.

**How:** By finding the best correspondence between the set of input points and a learnt hypergraph representation of the target landmarks.

**Priorities:** Landmark-retrieval rates, robustness to missing elements and speed.

**Independence:** For experiments on faces, we assume the input landmark positions to be accurate so that the success of the methods is maximal at 100%.

## 5.2 Graph and Hypergraph Representation

In matching problems found in computer vision, three kinds of data type are important.

- the properties attached to the local regions to be matched
- the properties attached to the relationships between these regions.



- the presence, or absence, of a defined relationship between these regions.

A graph seems a good choice to represent such a set of information. Most of the time in the literature, the landmarks and their properties are attached to the vertices of the graph, while the relationship properties are attached to the edges. The absence or presence of relationships defines the topology of the graph.

This representation has a serious limitation in that only pairwise relationships are taken into account. The relationship between more than two landmarks (e.g. the area between three or the volume between four) cannot be represented by the data structure. A generalisation of the idea of a graph is the hypergraph in which the degree of connectivity of the edges is not limited to two. An hypergraph can be defined as  $H = (V, E)$  where  $V = \{v_1, v_2, \dots, v_{|V|}\}$  is a set of vertices and  $E = \{e_1, e_2, \dots, e_{|E|}\}$  a set of hyperedges such as  $\forall e_i \in E, e_i \subset V$ . Figure 5.3 shows some possible visual representations of hypergraphs. In our case, properties corresponding to features are attached to the vertices and properties corresponding to relationships between features are attached to the hyperedges. Another way to see the hypergraph is to imagine it as a bipartite graph (see Figure 5.3c) where the first set of nodes represents the vertices and the other set the hyperedges. In this case only nodes of the bipartite graph support properties. The symmetry of the bipartite graph representation allows us to easily consider the dual (where vertices and edges have been interchanged, see Figure 5.3e) in any non-specialised process trying to eliminate candidates during the matching. The generic term “element” will be used in this paper when speaking of concepts common to both vertices and hyperedges (for example the properties). In the rest of this thesis, the *neighbours* of a node  $N$  will be the hyperedges connected to it (*vice versa* for the dual graph). The *same-type neighbours* will be the nodes connected to neighbours of  $N$  (see figure 5.4).

Finding correspondences between two sets of elements is a general problem, the solution of which can be useful in lot of computer vision domains from stereo 3D reconstruction to object detection, recognition and tracking. In order to be general, a lot of applications do not use the raw data directly (pixels, mesh points) but higher level features that are extracted from the scene. However, for a huge quantity of problems, the features by themselves are not expressive enough for unambiguous matching. Using relational information between the

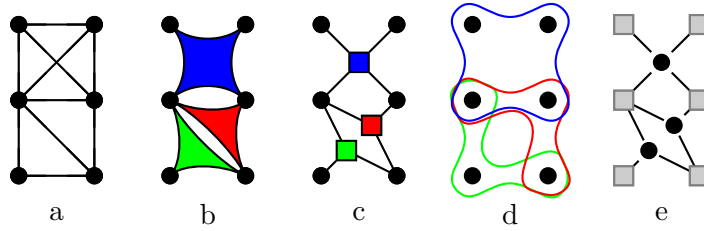


Figure 5.3: Examples of hypergraph representations.

- (a) A simple graph: a 2-uniform hypergraph (10 edges of degree 2)
- (b) An hypergraph (1 hyperedge of degree 4, 2 of degree 3)
- (c) A bipartite graph representation of this hypergraph
- (d) A set representation of this hypergraph
- (e) A bipartite graph representation of its dual

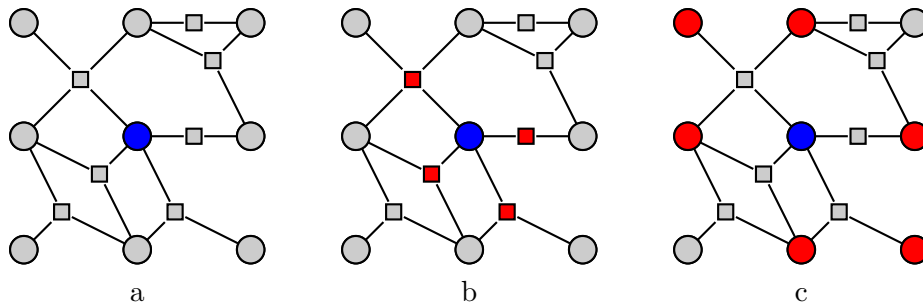


Figure 5.4: Example of direct neighbourhoods. Centre point (a), neighbours (b), same-type neighbours (c).

features helps to better describe the scene and, hopefully, brings enough constraints into the system to determine the correspondences.

While a high degree of connectivity can be useful for discrimination, the number of hyperedges increases dramatically with regard to the number of nodes.

Like graphs, hypergraphs can be oriented if the hyperedges are ordered lists. As most of the geometrical or arithmetic properties used here are symmetrical the decision was taken to implement a non-oriented hypergraph, which reduces the number of possible hyperedge matches. However, we introduced exceptions to this rule for hyperedges of degree 3 and 4.

### 5.2.1 Node Attributes

The nodes in our hypergraphs represent the points on the 3D model to be matched. The attributes (or properties) that can be attached to these nodes are numerous and can be

both scalar values or histogram structures. As a consequence of our framework design, every descriptor used in chapter 4 can be used as an attribute for the nodes and *vice versa*.

As a rule in our experiments, the number of nodes is much lower than the number of vertices in the mesh. This implies that we can use for the nodes, descriptors that are more computationally expensive than the ones used in Chapter 4. However, as our interest is to improve the relational matching when dealing with faces, we often prefer to use simple descriptors or no descriptor at all. The more discriminative the set of values attached to the node is, the better the seeding of the correspondences in the matching process will be. If the seeding is too good, the relational matching cannot be evaluated properly.

### 5.2.2 Hyperedge Attributes

Hyperedges attributes can play an important role in relational matching. However, the attributes are usually more ambiguous than those attached to the node, while the number of hyperedges is usually far more important. This implies that the seeding of the hyperedge correspondences is often inexact, produces a lot of false negatives and/or requires a lot of memory to store the candidates for each hyperedge in the query hypergraph. The hyperedge attributes depends on the degree of the hyperedge.

**Hyperedges of degree 2** Hyperedges of degree 2 (or simply edges) are the most common, as graphs are more popular than hypergraphs. On a 3D surface the number of attributes that can be attached to edges between landmarks are numerous. The obvious ones are the distances between the positions of the two extremities, whether the distance is Euclidean, geodesic, pseudo-geodesic, and so on. Other simple ones are the distances between any attributes of the nodes other than their positions. This can be the difference between scalar values (like curvature related descriptors) or that between histogram descriptors (like Spin Images). More sophisticated attributes might define values associated with a discretised path between the two extremities (e.g. integral of a scalar field along the geodesic path) or even use oriented local descriptors along the direction of the edge like the point-pair spin images [Romero and Pears, 2009b]. In this thesis, a few subsets have been selected for comparisons.

**Hyperedges of degree 3** For hyperedges of degree 3, simple descriptors can be the surface of the triangle area, its perimeter (Euclidean or geodesic), functions of the values of its edges and angles, and, of course, any meaningful ratio of the above (e.g. surface/perimeter, euclid.perim/geod.perim). A description of the interior of the triangle can also be used (Chapter 4 of [Romero, 2010]).

**Hyperedges of degree 4** A hyperedge of degree 4 correspond to a tetrahedron in the 3D space. Scalar descriptors that can be attached to it include the volume of the tetrahedron, its surface, the pair distance ratio between opposite arcs, the signed distance of one point to the plane defined by the 3 others (corner heights), and so on. Of course, any meaningful ratio of scalar descriptors can also be considered (e.g. surface area to volume ratio).

**Hyperedges of undetermined degree** Some of our ideas involve types of hyperedge of undetermined degree. The idea behind this is that surfaces or lines detected on the mesh help the grouping of landmarks into sets. These groups, stored as hyperedges, can be easier to match than individual landmarks. Therefore this reduces significantly the number of bad candidates. An example of the construction of hyperedges of undetermined degree is shown in Appendix D.

**Scalability** The size of the population of possible non-oriented hyperedges of degree  $d$  can grow as  $\frac{n!}{(n-d)!d!}$  with the number of nodes  $n$ . However, the filtering of attributes can easily be implemented to reduce such computational explosion. For example, on hyperedges of degree 2 with Euclidean distance, the range of possible values for the longest and smallest edge in the face can be learnt. By erasing edges that are too small or too long, the problem can be reduced to a reasonable size. The same principle applies to other hyperedge degrees and other descriptors. The problem of deciding which property and which threshold should be used is a difficult one as it depends on the kind of model we want to match. If we want to match an all-to-all connected graph it is quite easy. But there is no evidence that it is the best method to adopt as a sparser model can be easier to find (see Figure 5.5).

For hyperedges of degree 3, an order relation is enforced on the triplet to eliminate the uninformative permutation using the same set of points. To do so, the angles associated with each triangle's corners are sorted by increasing value. By fixing this order, the number

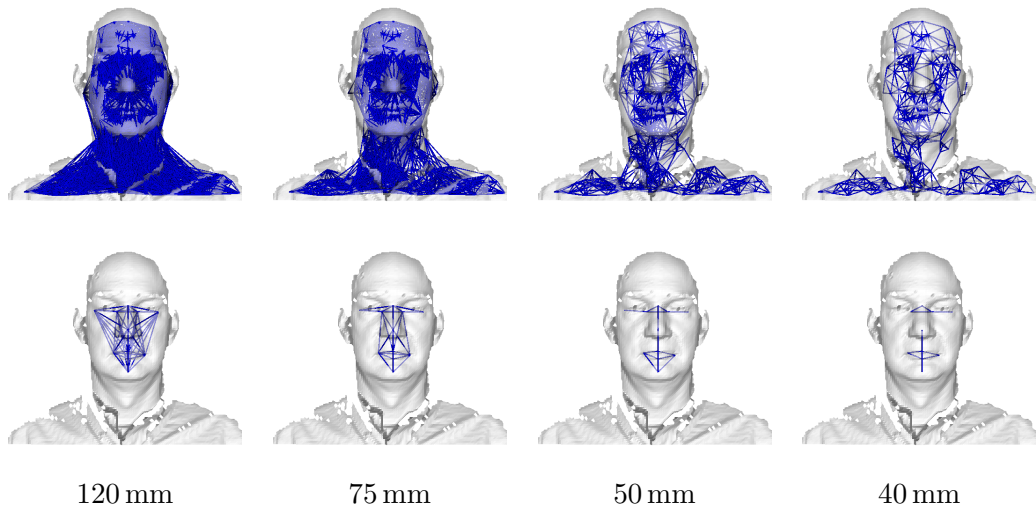


Figure 5.5: Query (top) and model (bottom) graphs generated using different maximum thresholds on the Euclidean distance between landmarks.

of hyperedges is divided by 6 and an extra attribute can be used for each triangle to reduce the number of possible candidates by 2. Indeed, the triangles of a manifold surface can always be oriented in direct or indirect order. In the case of range images produced by 3D sensors, the camera axis ( $z$ ) can be used to define this order. In the case of a complete 3D object, the normals are usually defined consistently relative to the interior/exterior of the object. After ordering by angles the current triangle is either direct (anti-clockwise) or indirect (clockwise). As direct triangles and indirect triangles cannot be matched, the candidate list per hyperedge of degree 3 can be reduced by 2.

However, ordering the angles of the triangle can cause instability: noise applied to the triangle corner positions can modify the shape of the triangle and therefore the ordering of the angles. This can not be totally avoided, but a simple technique can help reduce its occurrence. By not considering triangles that are almost isosceles, the probability of ordering inversion is significantly reduced. A margin of 0.2 radians has been selected, implying that the induced noise on the angle is expected to be lower in most cases. Figure 5.6 shows the part of the triangle space that is considered in our experiments. These selections can be performed independently on the query and the model. To further reduce the number of triangles, we can also consider only acute triangles (three times rarer than obtuse triangles for random-angle triangles).

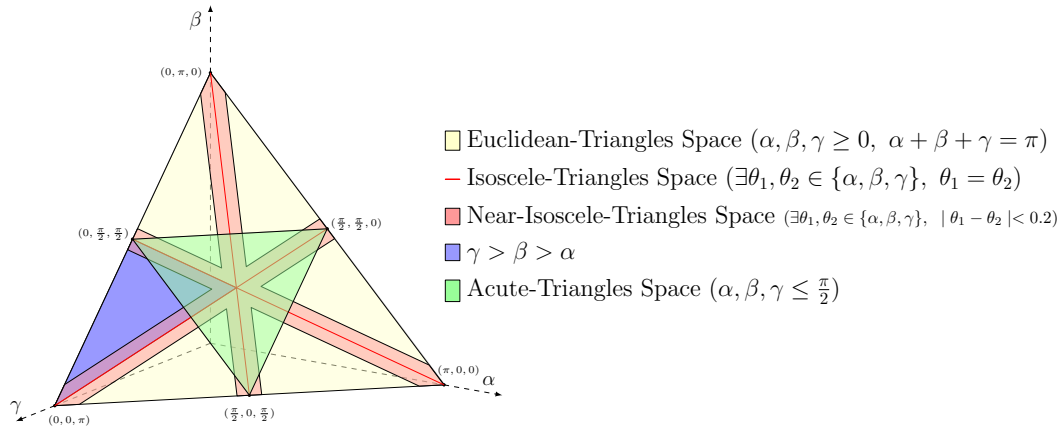


Figure 5.6: Random-angle triangle space representation. The whole triangle section of the plane represents all possible triangle shapes in Euclidean geometry. The blue area represent the increasing ordered-angle triangles. The red areas represent close to isosceles triangles that are discarded. The green area represents the acute triangles that can be used to further reduce the number of hyperedges.

### 5.3 Graph and Hypergraph Matching

Graphs are powerful tools for the representation of relational information. The correspondence problem applied to graphs is called the graph matching problem. A graph  $G$  matches another graph  $G'$  if there exists an edge-preserving mapping function  $f$  between  $V$  and  $V'$ .

The different constraints applied to the mapping function determine different flavours of exact graph matching between  $G$  and  $G'$  (see Figure 5.7):

- Graph Isomorphism: Match only if the two graphs have the same size and exactly the same edges.
- Induced Subgraph Isomorphism: Match if an induced subgraph of  $G$  is isomorphic to  $G'$ .
- Subgraph Isomorphism (or Graph Monomorphism): Match if a subgraph of  $G$  is isomorphic to  $G'$ . It means that  $G$  can have, between matching vertices, edges that do not appear in  $G'$ .
- Graph Homomorphism: Similar to a graph monomorphism but non-injective. Two vertices in  $G$  can be mapped to the same vertex in  $G'$ .

Problems	$G'$	$G$ +mapping(red)
<b>Isomorphism</b> ( $G \cong G'$ ): $\exists f$ bijective, $f : V \rightarrow V'$ $\forall u, v \in V \quad \{u, v\} \in E \Leftrightarrow \{f(u), f(v)\} \in E'$		
<b>Induced Subgraph Isomorphism:</b> $\exists V_S \subset V, \quad E_S = V_S \times V_S \cap E$ $(V_S, E_S) \cong G'$		
<b>Subgraph Isomorphism</b> (Graph Monomorphism): $\exists V_S \subset V, \quad \exists E_S \subset V_S \times V_S \cap E$ $(V_S, E_S) \cong G'$		
<b>Graph Homomorphism:</b> $\exists f$ surjective, $f : V \rightarrow V'$ $\forall u, v \in V, \quad (f(u), f(v) \neq \emptyset) \Rightarrow$ $(\{u, v\} \in E \Rightarrow \{f(u), f(v)\} \in E')$		
<b>Maximum Common Subgraph Isomorphism:</b> Maximum $K \in \mathbb{N}$ such that $\exists V_S \subset V, \quad \exists E_S \subset V_S \times V_S \cap E, \quad  V_S  = K$ $\exists V'_S \subset V', \quad \exists E'_S \subset V'_S \times V'_S \cap E'$ $(V_S, E_S) \cong (V'_S, E'_S)$		

Figure 5.7: Examples of Exact Graph Matching Problems

- Maximum Common Subgraph Isomorphism: Compute the largest subgraph of  $G$  which is isomorphic to a subgraph of  $G'$ .

The same problem exists for inexact graph matching in which small errors can be accepted. The errors can be due to topological differences or to attribute differences in cases of attributed graphs.

Here, the focus is set on problems of inexact multi-attributed hypergraph and sub-hypergraph matching. In the case of the hypergraph matching problem, the observed graph will be called “query graph” and the reference graph will be called “model graph”. In the case of the sub-hypergraph matching problem, the observed graph might sometimes be called “scene graph” and the reference graph “template graph”.

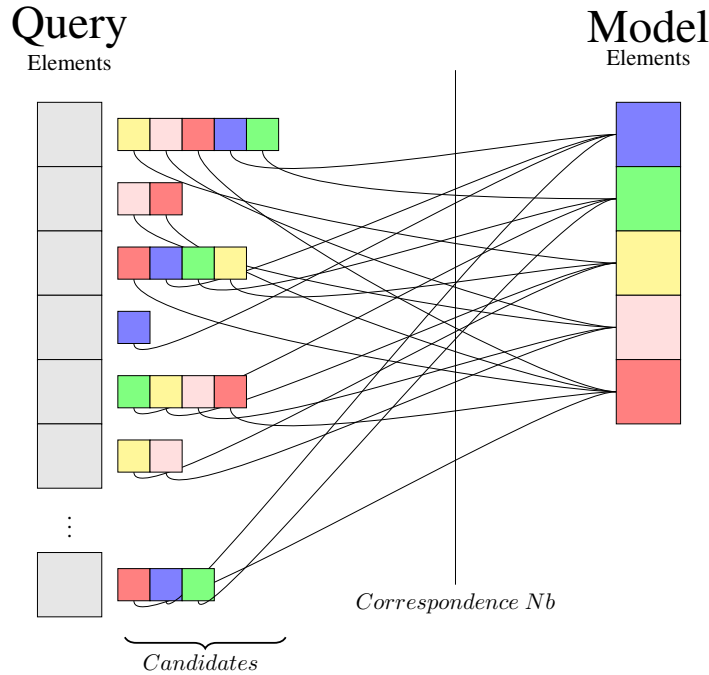


Figure 5.8: Hypergraph matching representation. The vertical vectors represent the query (left) and model (right) elements to be matched. The term element can be substituted by node or hyperedge. The horizontal vectors represent the list of current target candidates for each query element. For a given couple query/model, the set of list of candidates is called a *solution*. The aim is to generate the smallest solution possible containing the true correspondences.

### 5.3.1 Inexact Hypergraph Matching

In this thesis, and for the sake of simplicity, we will often use the term “graph matching” in a generic way. Most of the time, the term “graph matching” can be understood in its broader sense of “structural inexact matching” or more precisely in our case “multi-attributed inexact sub-hypergraph matching”.

**Inexact attributes** Due to the variability of the human face, occlusions, holes, spikes and noise in the input data, the attributes attached to the elements of the hypergraph might often be inexact. An inexact attribute in our system can lead to both discrete and continuous differences. A continuous difference is represented by a matching score giving the likelihood of the element to match a given target. A discrete difference occurs when the score is under a learnt threshold. When it happens the candidate is simply removed.



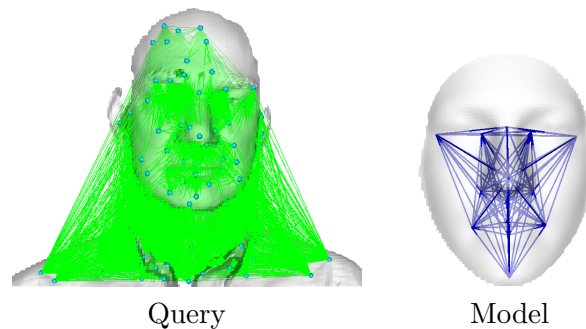


Figure 5.9: Visualisation of an example of input data for the graph matching problem. The aim is to find the subgraph of the query most similar to the model graph.

**Extra data** The input data always contains a lot of points that don't correspond to any label in the model (non-matchable). The ability to cope with this form of noise is an important characteristic of the system.

**Missing data** Because of pose variation, occlusion, or bad keypoint detection some points present in the model might be missing from the query. If we were working on an exact matching problem this would change the nature of the problem (from subgraph matching to maximum subgraph matching). While usually the scene is supposed to contain the model, here we have a case in which it contains less than the whole model. Therefore, we need to find the largest subset of the model present in the scene. This is a serious problem of doing practical inexact sub-hypergraph matching. None of the papers on hypergraph matching evaluate the effect of occlusion because of this problem. Most techniques are very stable in response to additional noise but are likely to be very sensitive to point removal.

One of our contributions in this chapter is to evaluate existing techniques with regard to occlusion on synthetic data.

**Complexity** The complexities we are interested in are those of both graph and subgraph isomorphism. The subgraph isomorphism problem has been known to be NP-complete since at least 1971 [Cook, 1971]. On the other hand, the graph isomorphism problem also belongs to NP but has not been proved to belong either to P or NP-Complete [Garey and Johnson, 1979]. It is one of the very few problems for which complexity remains an open question.

**Graph Matching Families** While classic pattern recognition mainly focuses on algorithms that work in the feature space, structural pattern recognition needs to find a way to combine both featural and configural information into the algorithm. The incompatibility between the two often requires a choice between:

- **Optimal methods on degenerate data:** For example, most matrix-based methods use only a fragment of the information contained by the input graph. Sometimes only the attributes of the edges are used, sometimes only the adjacency matrix is used. Besides, the problem is often simplified to guarantee that the matrices are squares.
- **Suboptimal methods on complete data:** For example, relaxation methods will use all the information from the graph but will not guarantee that the found solution is a globally optimum.
- **Optimal methods on complete data:** Due to a renewed interest in hypergraphs, some extension of matrix-based matching to tensors has led to interesting globally optimal methods on a complete data representation of the hypergraph matching problem in the similarity space [Duchenne et al., 2009] [Chertok and Keller, 2010] (not the feature space). The main problem with this is the fact than once again the human-defined global cost function doesn't necessarily describe what the human operator really wants the system to solve. These cost functions usually become unstable when some data is missing in the query (occlusion).

In conclusion, our system should enforce a trade-off between the algorithms, the data representation and the cost-function definition.

### 5.3.2 Two Different Kind of Matching

**Graph-to-graph matching** In cases in which two graphs of the same nature have to be matched, the problem is often to determine a similarity measure between the elements of the two graphs. This is the most widely solved problem in the literature where no model of the targeted objects is known. In [Zass and Shashua, 2008], for example, the scores between edges are computed from their length ( $d(e_i)$  and  $d(e_j)$ ):

$$S^d(e_i, e_j) = \exp(-|d(e_i) - d(e_j)|)$$

One obvious problem with this is the notion of scaling. The difference of attributes is usually normalised manually or not normalised at all. Consequently in [Zass and Shashua, 2008], changing measures from millimetres to metres would completely change the dissimilarity scores and change the outcome of the matching. Furthermore, all points have the same allowance of deviation. Despite the symmetry in the problem, the solution is not guaranteed to be symmetrical, depending on the method used.

**Graph-to-model matching** Sometimes the problem is to match the graph to a class of graph for which many instances are known. In this case, it is possible to extract a statistical model from the known set and try to match the graph to it. This kind of graph matching is intrinsically non-symmetrical. The advantage is that the distribution of each element’s properties are known which allows our similarity measures (scores) between elements to be scaled automatically using the deviation. An example in our case, assuming a Gaussian distribution, is:

$$S_{e_m}^d(e_i) = \exp\left(-\frac{(d(e_i) - \mu_m)^2}{2\sigma_m^2}\right)$$

The similarity measure doesn’t need to be normalised.

Because of our framework, the same distribution presented in Section 4.3.1 can be used here for the scores. Once again, for most elements and for most attributes, the Gaussian distribution is used as an idealised distribution. The reader should assume a Gaussian distribution is used except when explicitly notified otherwise.

### 5.3.3 Seeding

The seeding is the initial trimming of the candidates for each element of the query hypergraph. If the seeding is absent, each element of the query graph will have as candidates all the compatible elements in the model graph. Introducing a seeding for a multivalued element can be done in different ways.

**Combining attribute scores** As seen for vertices in Section 4.3, scores can be computed for each attribute  $D$  using the probability density function learnt on the training set. The same idea can be applied to any kind of element (node or hyperedge) in score computation for the matching process. Let  $e_i^Q$  be the  $i^{th}$  element of query hypergraph and  $e_j^M$  the  $j^{th}$  element

of the model hypergraph. Let  $D = [D_1, \dots, D_P]$  be the list of attributes (descriptors) for this type of element,  $D_k(e_i^Q)$  the value for the query element and  $\text{pdf}_{e_j^M}^{D_k}$  the probability density function associated with the attribute for a given model element.

For each potential query-model assignation  $(e_i^Q, e_j^M)$ ,  $P$  scores can be computed:

$$S_{e_j^M}^{D_k}(e_i^Q) = \frac{\text{pdf}_{e_j^M}^{D_k}(D_k(e_i^Q))}{\max_x(\text{pdf}_{e_j^M}^{D_k}(x))} \quad \forall k \in [1, 2, \dots, P]$$

Once again, the problem is to find a combine function  $\uplus$  that merges these  $P$  scores into one single score that can be used in the matching process. At the end, the seeding algorithm will look like:

$$IsCandidate(e_i^Q, e_j^M) = \begin{cases} True & \text{if } \uplus_{1 \leq k \leq P} S_{e_j^M}^{D_k}(e_i^Q) \geq T_{e_j^M} \\ False & \text{otherwise} \end{cases}$$

where  $T_{e_j^M}$  is a threshold learnt for this particular model element.

The combine function  $\uplus$  can be a simple product ( $\prod S_k$ ), the complement of the product of complement (multiply improbabilities) ( $1 - \prod (1 - S_k)$ ), the mean ( $\frac{1}{P} \sum S_k$ ), a linear combination using learnt weights ( $\sum \mu_k S_k$ ) or a non-linear combining function like, for example, Adaboost (see section 4.7).

Using a complex combining function here can be quite computationally expensive: while the number of nodes is far lower than the number of vertices used in chapter 4, the number of hyperedges can be similar or bigger. Moreover, the comparisons are not done  $P$  times like in the keypoint detector but  $|e^M|P$ , as one distribution is learnt for every model element and for every attribute.

**Retained solution** For our approach, the seeding scores are computed as a linear combination of each descriptor score for a particular target landmark (see Figure 5.10). The linear weights are computed using a simple 2-class LDA on the training data. The first class is the target element (node or hyperedge), and the other class is the set of remaining same-type elements in the model (see Figure 5.11). A simple threshold is learnt for each element such that at least 95% of the training points are true positives.

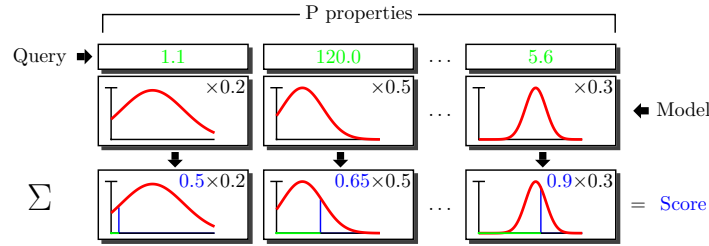


Figure 5.10: Diagram of the score computation for an element using a linear combination of descriptors. The query element owns a set of descriptor values (in green). The model provides a set of idealised distributions for each descriptor (in red) as well as weights (in black). For each descriptor, a score (in blue) is computed by looking at the image of the query value through the known distribution function. These scores are then summed with their corresponding weights to produce the final scalar score value.

### 5.3.4 Are Multi-Attributes Worthwhile?

A natural question at this stage is to ask whether using multi-attributed elements is useful. Our idea was to see how the landmarks, or the relationships between several landmarks, can be distinguished by the values of their properties.

As a first step, the assumed distribution of each property has been plotted for each landmark. The less the Gaussians of each element intersect, the better the property is for labelling (more discriminative). The figures 5.12 and 5.13 show examples of such plots. While this representation can give us an indication about which properties are less efficient for labelling (for example the Gaussian curvature  $K$ ), it is still difficult to take decisions with these results alone.

The second step was to see how the matching scores can help distinguish the landmarks. Each element of each graph was matched against each element of the model. Statistics for this metric were computed over the whole database. The results obtained are easier to analyse than those with the Gaussian curve. For example, in figure 5.14, the first three columns show, for a set of edges, the probability that they match a selection of other edges using a given property. The ideal situation is when all the candlesticks are close to zero except the one corresponding to the query edge.

The third step was to try to combine different properties together so as to make the elements more distinguishable. The fourth column of figure 5.14 shows a result using a simple mean on the matching probabilities greater than 0.1. We see that it helps eliminate

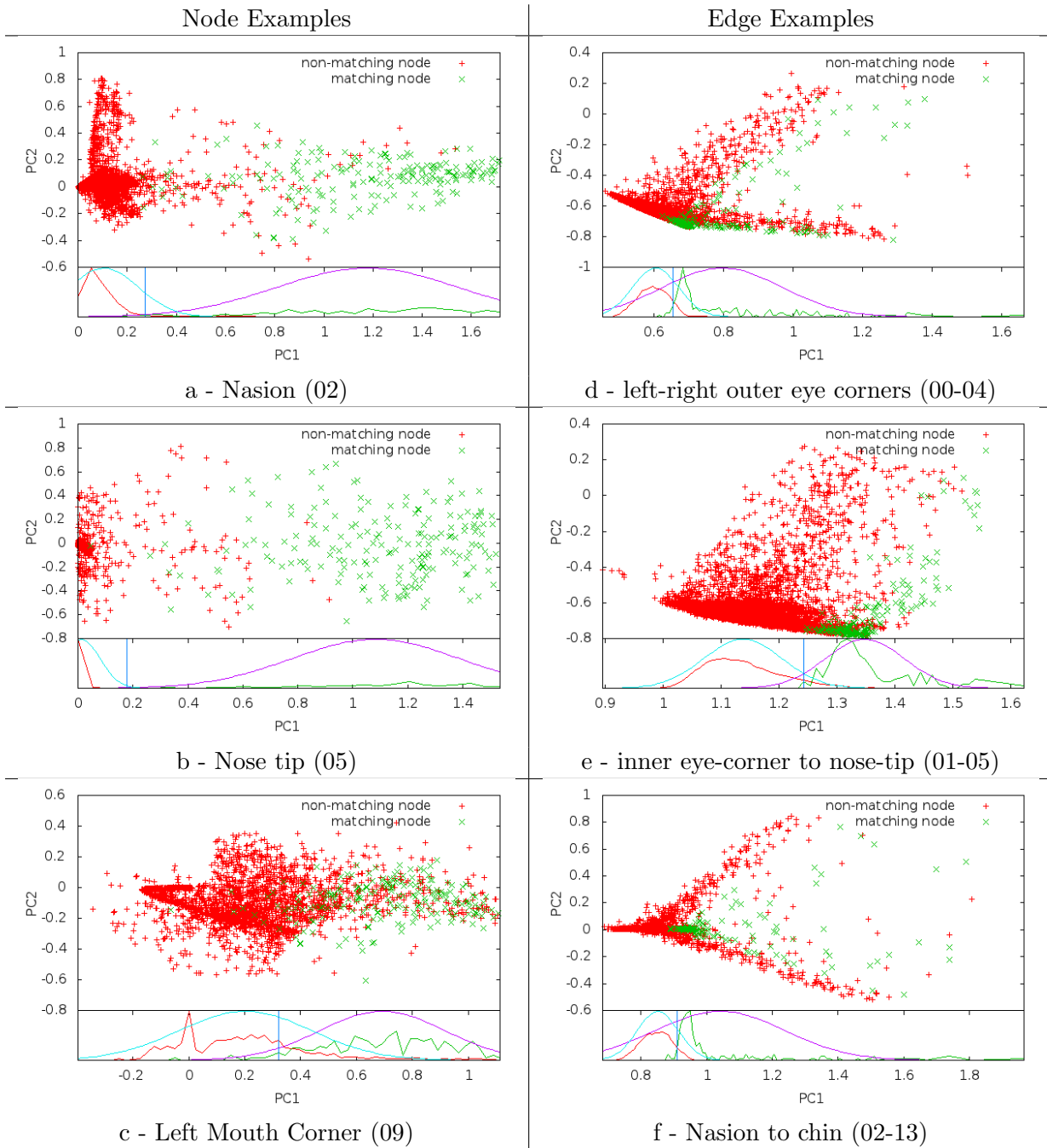


Figure 5.11: Examples of element correspondences (query-model) plotted in the LDA-reduced space of their score per attribute. The density functions below show the separation of the two classes along the main component of the transformed space (the second one is only used to help visualisation). The blue vertical line represents the selected seeding threshold.

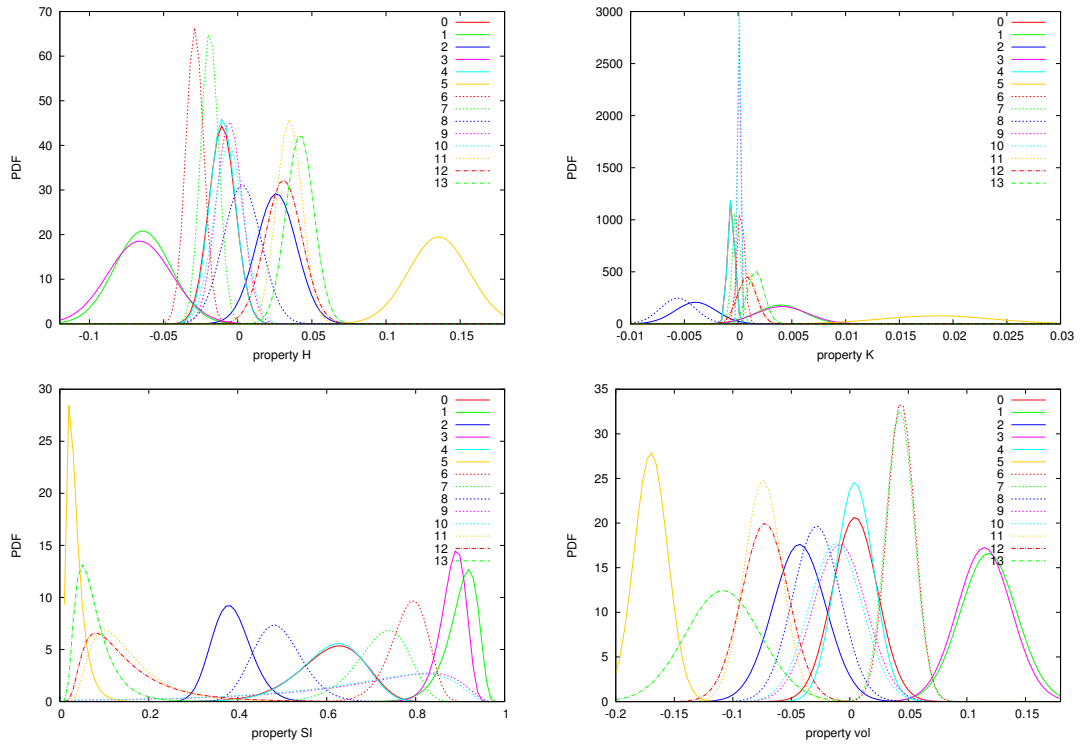


Figure 5.12: Probability Density Function (PDF) of different node properties (H, K, SI and VOL described in Section 4.2.4) over the FRGC database. The more the distributions are “separated”, the more easily the node can be distinguished using this property.

candidates (fewer blue sticks). The same data exploration can be performed in higher degrees but it becomes difficult to visualise without limiting the number of possible hyperedges. In figure 5.15, 7 properties on hyperedges of degree 3 are combined (once again we take the mean of all properties above 0.1) to improve the edge recognition. The more there are repeatable properties, the more likely the wrong candidates will be eliminated. It can also be noted that despite the symmetry of the face, the matching side elements obtain better scores than their counterparts.

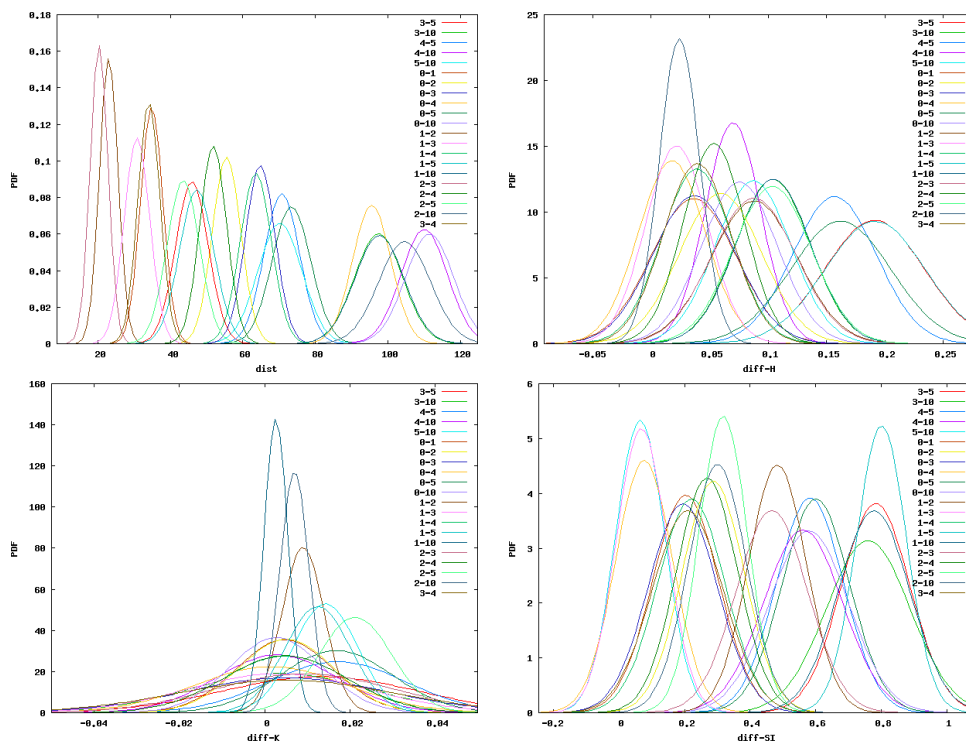


Figure 5.13: Probability Density Functions (PDF) of 4 properties for a subset of edges (between 11 landmarks) over the FRGC database. The more the Gaussians are “separated”, the more easily the edges can be distinguished using this property.



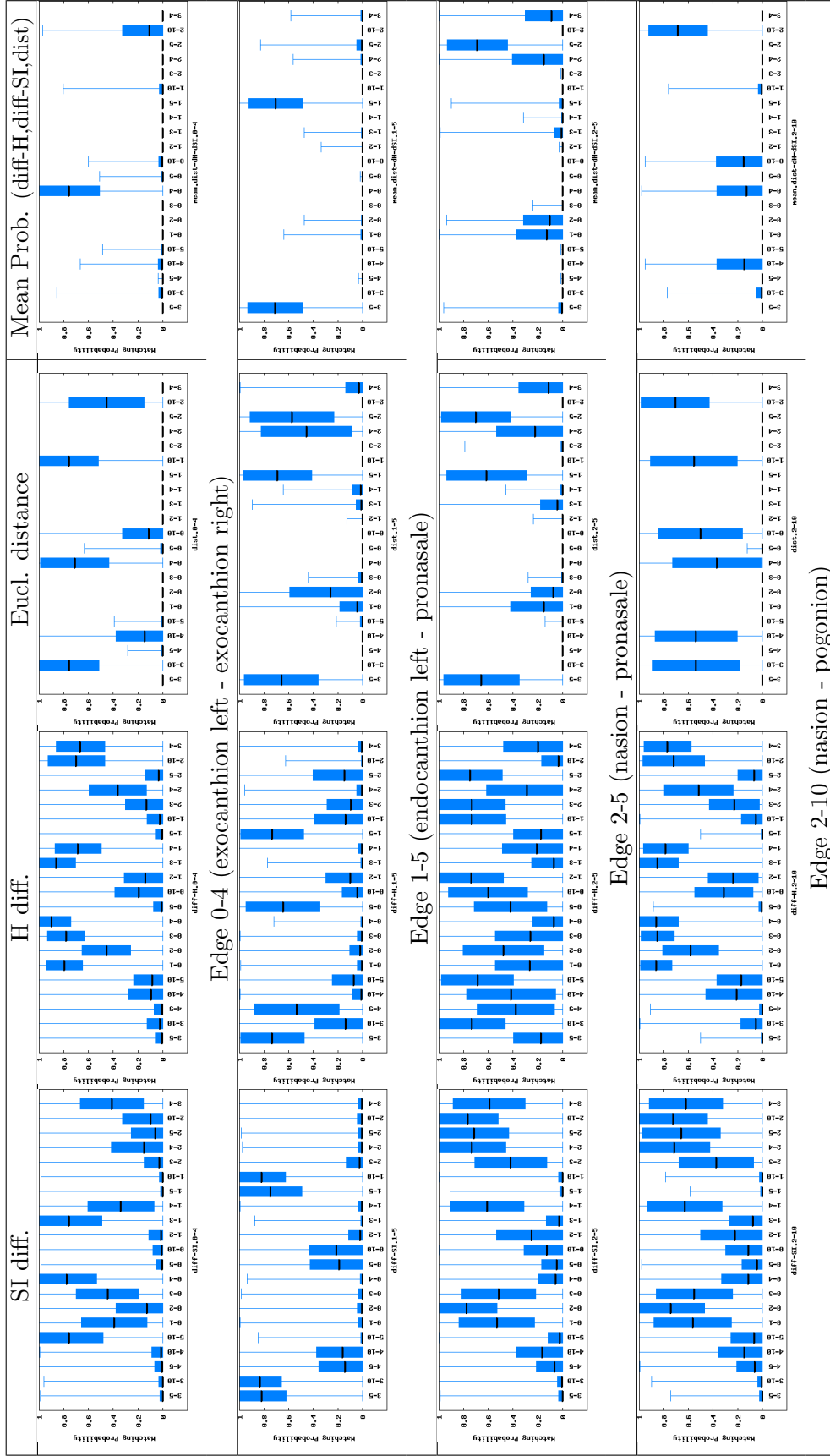


Figure 5.14: Example for four edges (rows) of matching probabilities against the other edges using different properties on the FRGC v2 database. We see that the combination of several properties (last column), computed by averaging properties that are above 0.1, improves the distinguishability of the edges. NB: the candlesticks represent the  $\{min, \mu - \sigma, \mu, \mu + \sigma, max\}$  values not the  $\{min, Q1, median, Q3, max\}$  values.

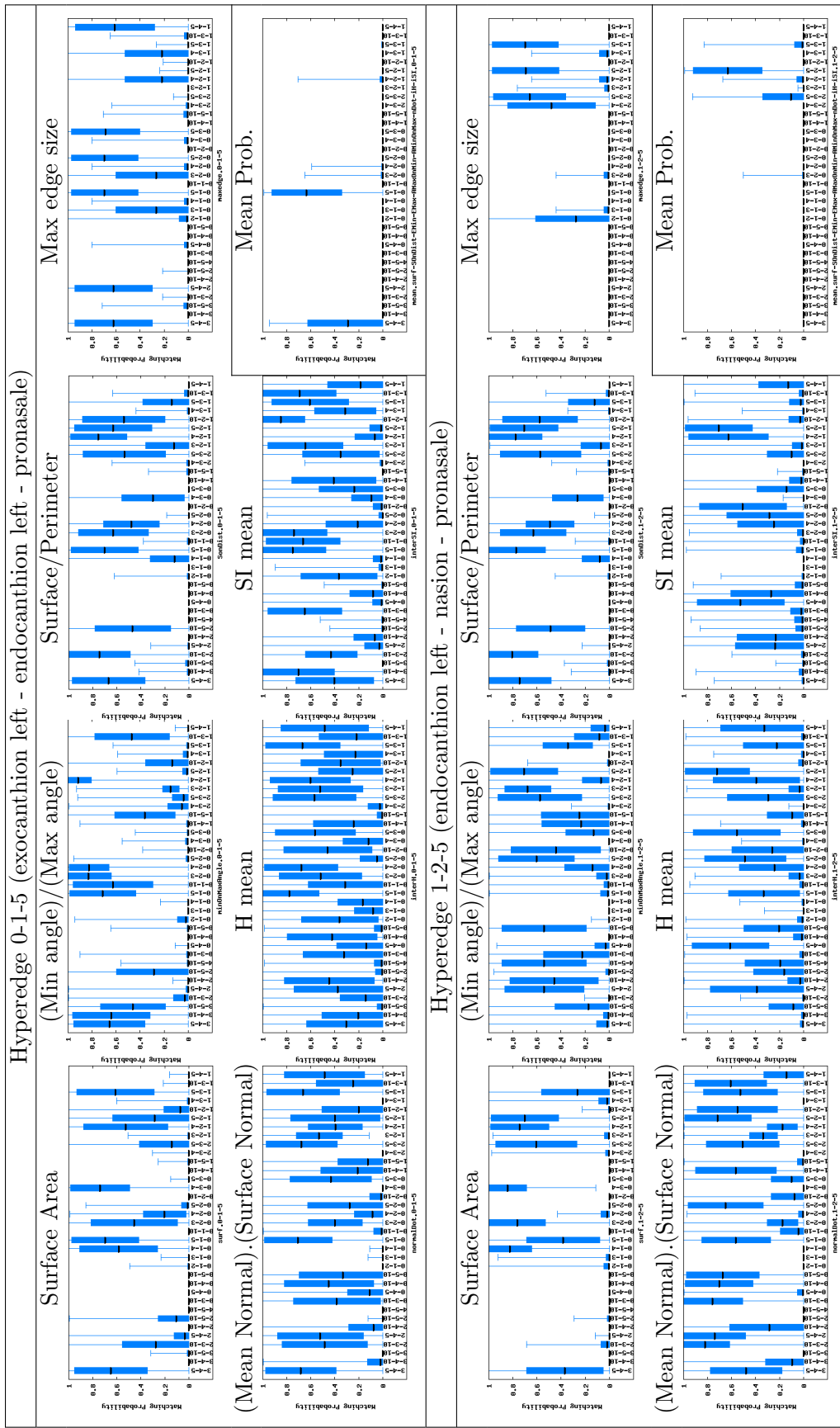


Figure 5.15: Examples (for two hyperedges) of matching probabilities against the other hyperedges using different properties on the FRGC v2 database. We see that the combination of several properties (last cell), computed by averaging properties that are above 0.1, improves the distinguishability of the hyperedges. NB: the candlesticks represent the  $\{min, \mu - \sigma, \mu, \mu + \sigma, max\}$  values not the  $\{min, Q1, median, Q3, max\}$  values.

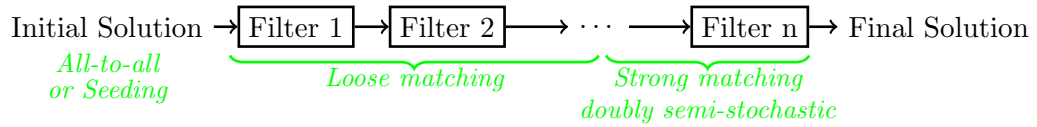


Figure 5.16: Workflow of our global correspondence systems. The initial local correspondences are filtered using different structural-matching techniques to reduce the ambiguities at each step.

### 5.3.5 Workflow

Our stand is that correspondence-finding processes should be seen as relaxation systems: taking as input sets of candidates for the correspondences and returning subsets of smaller or equal cardinality.

The framework of our system translates this idea. Each of our experiments consists of a pipeline of different methods working sequentially to reduce the correspondence ambiguities.

To avoid the computational cost of a global all-to-all correspondence, a seeding function is used to create the initial solution. In order to clean the solution the last filters are often doubly semi-stochastic processes. We do not always enforce the rule that the solution output should be a subset of the input (definition of a filter) to allow error correction. For example, a process composed of a registration method will extract correspondences that might not be present in the input.

### 5.3.6 Correspondence Stochasticity

Any exact-correspondence problem can be represented as a simple matrix of permutations<sup>1</sup>. This matrix is said to be doubly stochastic. Indeed, every column or row sums to 1. In our case, the two sets can have different sizes and the desired correspondence we seek doesn't need to be complete. Therefore, the sum of the columns or rows will be either zero or one. Thus, our problem is to find a doubly semi-stochastic solution.

### 5.3.7 Limitation

Because we use statistical matching, the model has to be constructed from a big enough training set. This learning part of the model object cannot be skipped. Consequently our

<sup>1</sup>The  $|V_Q| \times |V_M|$  matrix  $X = \{x_{i,j} \in \{0,1\}\}$ ,  $\sum x_{i,\cdot} = 1$  and  $\sum x_{\cdot,j} = 1$ . i.e.  $x_{i,j} = 1$  iff the vertex  $i$  of  $Q$  matches the vertex  $j$  of  $M$ .

technique cannot be used to find a particular object but only classes of objects. Our system cannot know what a mug is if only one instance of a mug is given. Otherwise you need to input manually the variation allowed for each attribute generated.

## 5.4 Postprocessing/ Correspondence Cleaning

The hypergraph matching process (especially by relaxation) is designed to eliminate the subtree of possibilities that are obviously wrong. It doesn't guarantee in any way that the remaining set of candidates in the output represent a coherent and unique solution.

In this section, different correspondence filters for label selection are presented. This section groups all the techniques that are not based on soft hypergraph matching but that can be used to reduce the set of correspondence candidates.

### 5.4.1 Hypergraph Matching as a Clustering Problem

The graph-matching problem can be seen as a clustering problem, provided the number of candidates for each match is relatively small. If a similarity measure between correspondence candidates exists, which gives high scores for individual correspondences of a good global match, and low scores for random correspondences, then finding valid global matches can be achieved by detecting the biggest clusters in this correspondence space.

Very often such a similarity measure is not easy to find. Luckily for hyperedges of degree 3, an obvious one can be computed: the distance separating global transformations between the query and model triangle of both correspondence candidates. Two different similarity measures based on rigid registration are presented in the following sections. In the first one, the similarity measure is the difference between the unit-quaternion component of the global transformation that are clustered using a cutting-tree clustering technique. The second uses the distance error of the model registration, clustered using a RANdom SAmple Consensus (RANSAC) technique.

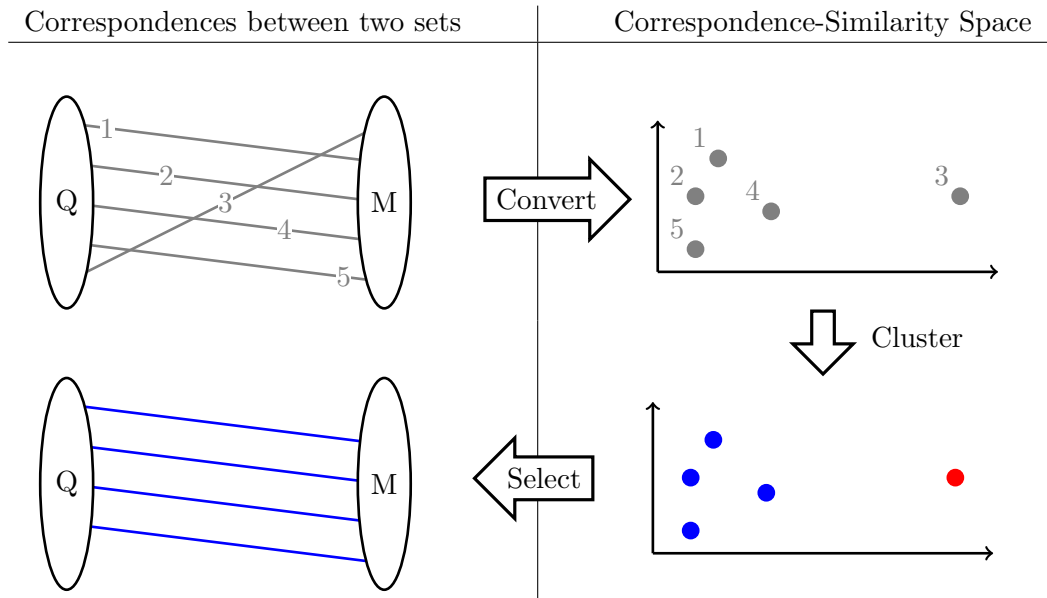


Figure 5.17: Description of the concept of hypergraph matching by correspondence clustering. If a similarity metric between correspondences exists and correlates with the two correspondences being part of the same global match, it is possible to cluster correspondences and eliminate outliers. The biggest cluster in the correspondence-similarity space is likely to correspond to the best global correspondence.

#### 5.4.2 Rigid Registration by Unit-Quaternion Clustering

In the case of 3D faces, the feature correspondence using individual landmarks and the global rigid correspondence using the mesh or a big set of features are usually close to each other. Finding a global rigid registration can therefore help us find the features and *vice versa*.

Given a set of query landmarks and a set of model landmarks, the registration is defined as the 3D transformation that minimises the mean square distance between the points of corresponding label. A closed-form solution to this problem is given by [Horn, 1987]. Hereafter, the registration that uses all of the landmarks of the model is referred to as “global” and the registration using a subset of three landmarks is called “triangle registration”. A 4x4 transformation matrix is used every time a rigid registration is computed. It is decomposed into its scaling, rotation and translation components in order to do independent clustering on those components. For the rotation part, a unit-quaternion representation is used. We

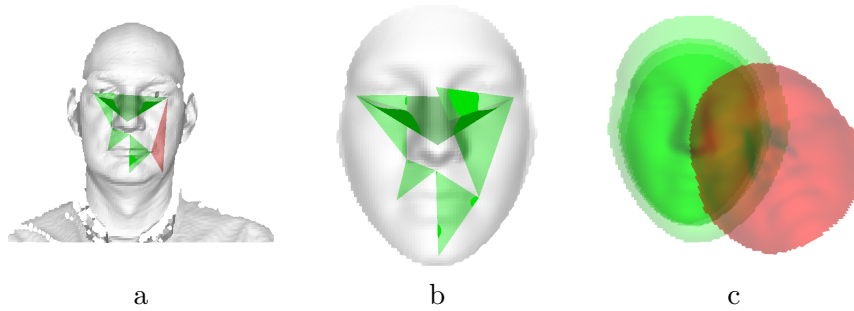


Figure 5.18: Example of corresponding triangles in the query (a) and model (b) with one error (the nasion is wrongly detected on the side of the face). Good correspondences between the corners of these triangles produce registrations close to each other on the unit-quaternion sphere while bad correspondences produce isolated rotations (c).

decompose the 4x4 transformation matrix as follows:

$$T = \left( \begin{array}{c|c} & \\ \hline R' & \vec{t} \\ \hline 0 & 1 \end{array} \right) \rightarrow \begin{array}{l} \dot{q} \quad \text{Unit Quaternion} \\ \vec{t} \quad \text{Translation} \\ s \quad \text{Scale} \end{array}$$

Our hypotheses at this stage are the following:

1. A significant proportion of the best query candidates for each model landmark in the input are good matches.
2. The transformation that registers the whole model to the whole face is very similar to the transformation that registers a sub-part of the model (a triangle) with the corresponding sub-part on the query face.
3. Bad correspondences are unlikely to produce a coherent transformation: different sub-parts will be registered in different ways.

Table 5.1 shows some evidence supporting the second hypothesis. Figure 5.18 shows an example of the principle that good landmark correspondence will produce good triangle correspondence. Good triangle correspondence can be detected by the fact that their associated registrations cluster in the unit quaternion space.

The smallest number of points to get a registration is three. However, when the triangle is very flat (close to a straight line), the transformation is less reliable. Triangles with an

Table 5.1: Statistics over the training set of the FRGC of transformation differences between triangle transformations (inter-T) and between triangle and global transformations (T/G) for the same face. The angles are given in radians, distances in millimetres.

Variable	Mean	Dev.	Min	Max
T/G scale ratio	1.000	0.069	0.671	1.367
T/G quaternion angle	0.055	0.036	8.59e-04	0.662
inter-T quaternion angle	0.078	0.048	6.89e-05	0.786
inter-T translation distance	5.594	3.436	0.018	47.172

angle of less than 15 degrees are discarded, which represents about 22% of the combinations within the model.

For the clustering used to get the final transformation, a very simple approach is adopted. First the distances between all pairs of elements are determined. A binary distance tree is created by selecting the smallest distance between the already computed sub-tree and the rest of the points. At each step, the distances involving the newly created sub-tree are updated using its new centroid. The distance tree is then cut using a distance threshold and/or a critical number of elements per sub-tree.

The final registration is used to assign the definitive labels by selecting the closest query point to each labelled landmark in the registered model.

**Unit-Quaternion Clustering** While clustering Euclidean vectors (translations) is not problematic, clustering quaternions that lie on a 4-dimensional sphere requires some precautions. As two quaternions  $\dot{q}$  and  $-\dot{q}$  represent the same rotation, it is important to check that the dot product between two compared quaternions is positive, otherwise we multiply one of them by  $-1$ . The metric used for the clustering is the following:

$$d(\dot{q}_1, \dot{q}_2) = \theta = 2 \arccos(\dot{q}_1 \cdot \dot{q}_2) \quad (5.1)$$

Finally, we determine the centroid of a subset of quaternions on the 4-sphere. However, as the clustering angles' thresholds are small, the distance can be approximated. In [Gramkow, 2001], Gramkow used the Taylor series of  $\cos \frac{\theta}{2}$

$$\cos \frac{\theta}{2} = 1 - \frac{1}{8}\theta^2 + o(\theta^4) \quad \cos \frac{\theta}{2} = 1 - \frac{1}{8}\theta^2 + o(\theta^4) \quad (5.2)$$

to approximate the distance for small angles.

$$\begin{aligned} d(\hat{q}_1, \hat{q}_2)^2 = \theta^2 &= 8(1 - \cos(\frac{2 \arccos(\hat{q}_1 \cdot \hat{q}_2)}{2})) + o(\theta^4) \\ &\simeq 8(1 - \hat{q}_1 \cdot \hat{q}_2) \end{aligned} \quad (5.3)$$

Under those assumptions, he proved that the centroid is equivalent to the normalised Euclidean barycentre of the unit-quaternions' coordinates. This allows us to average small rotations in a simple way.

### 5.4.3 RANSAC - Random Sample Consensus

As previously explained, the matching problem can often be seen as a clustering problem of the feature correspondences given an adequate similarity measure. Here a different clustering technique is used based on RANdom SAMple Consensus (RANSAC). RANSAC is a model-fitting algorithm introduced by [Fischler and Bolles, 1981] that consists of constructing a maximal size set of points by random sampling that agree on fitting the same instance (parametrisation) of a model.

Because of the discrete nature of our fitting problem, the method is slightly modified so that the selection occurs on correspondences instead of points. The target in our case is the mean rigid model of the 14 target landmarks. The parameters to fit the model are represented as a 4x4 transformation matrix  $\mathbf{T}$ . For any randomly picked correspondence  $(\mathbf{q}, \mathbf{m})$  the error in fitting the current model is defined as the Euclidean distance between the query point and the transformed model candidate of this correspondence:

$$error = d_{Eucl.}(\mathbf{q}, \mathbf{T} \cdot \mathbf{m})$$

If the error is under the acceptance threshold, the correspondence is added to the consensus and a new mean transformation  $\mathbf{T}$  is computed by using a least squares method. The least square method employs either a rigid transformation or a similarity (scale-adapted) transformation. In Chapter 6, this technique is slightly modified to take the angle between surface normals into account.



## 5.5 Proof of Concept - Hypergraph Matching by Relaxation with Unit-quaternion Clustering Disambiguation

Our labelling system is composed of a graph-matching stage and a post-processing stage, which uses a scale-adapted rigid registration. The graph-matching system consists of two graphs: firstly a fully-connected (complete) model graph created from the 14 landmarks in all of the hand-labelled training data set, and secondly a fully-connected query graph, generated from the (unlabelled) input points (box c in figure 5.2). Both of these graphs are attributed, with vertices having  $N$  descriptors (e.g. curvature) and edges having  $M$  descriptors (e.g. Euclidean distance).

In more detail, the attributes for each node in a graph are computed using a local spherical neighbourhood of radius  $15\text{ mm}$  within the face scan (training or testing set). Most of the attributes are derived from the maximal curvature ( $k_1$ ) and minimal curvature ( $k_2$ ) over this neighbourhood and the full list of 5 nodal attributes is as follows:

- Mean Curvature (H):  $\frac{k_1+k_2}{2}$
- Gaussian Curvature (K):  $k_1k_2$
- Shape Index (SI):  $\frac{1}{2} - \frac{1}{\pi} \arctan \frac{k_1+k_2}{k_1-k_2}$
- Rough Volume (Vol): Sum of the tetrahedron volumes from the centroid of the perimeter to all the triangles inside of the neighbourhood.
- Log Curvedness (LC):  $\frac{2}{\pi} \log \sqrt{\frac{k_1^2+k_2^2}{2}}$

The attributes for the edges of the graphs are:

- Euclidean Distance (dist)
- Coarse Geodesic Distance (distG): shortest path on the mesh
- Ratio between the two attributes above (ratioEucliGeod)
- Difference between vertex properties at either end of the edge ( $\Delta H$ ,  $\Delta LC$ ,  $\Delta Vol$ ,  $\Delta SI$ )

The labelling system is composed of an off-line part in which the graph model is trained (described later) and an on-line testing part. The online testing is divided into three main processes, as follows:

1. Compute an attributed graph (the query graph) from the unlabelled input points on the facial scan, as described above.
2. Run a graph-matching process using the generic model graph of the face. Here we determine initial candidates for each vertex and edge of the query graph. We then iterate our relaxation by elimination procedure to reduce the number of candidates.
3. Select the best labels using a scale-adapted rigid registration. The method employed here is: (i) Select current best labels using thresholding on scores. (ii) Compute registration transformations using combinations of 3 points. (iii) Cluster rotational and translational components to determine a good scale-adapted rigid registration. (iv) Use this registration to determine the best label assignment.

The graph matcher we have developed is in fact a hypergraph matcher which implies that the processes applied to nodes can also be applied to the edges. The term “element” is used hereafter to generally refer to either vertices or edges.

### 5.5.1 Offline Training Process

A set of facial scans, disjoint from the testing set, are selected for training. Using this data, the statistical distribution of each attribute value associated with each element (node or edge) in the model graph is collected and modelled using a Gaussian.

To determine a matching score between an element’s attribute *value* in the query graph and an element’s attribute *distribution* in the model graph, a normalised probability density function is used, as follows:

$$Score(P^{Query}, P^{Model}) = \exp \frac{-(P_{value}^{Query} - P_{\mu}^{Model})^2}{2 * P_{\sigma}^{Model}{}^2} \quad (5.4)$$

where  $P_{value}^{Query}$  is the value of the query attribute and  $P_{\mu}^{Model}$  and  $P_{\sigma}^{Model}$  are the mean and deviation of the trained model attribute distribution.

Note that this equation relates to one attribute, yet an element is described by a N-dimensional vector of attributes. Thus we need a method of composing a match score over this multidimensional space. In order to do this, we find the best linear combination of attributes (for every node/edge in the model graph) that discriminates between elements of the same label and those of a different label. To do this, we apply Linear Discriminant Analysis (LDA) to the training data, an example of which is given in figure 5.11. Shown at

the bottom of this figure is a blue vertical line that represents the seeding threshold. It is set such that at least 95% of matching landmarks in the training set are above it. When testing, this threshold is used to seed candidate labels for each query point.

The last parameters that need to be determined for the graph matcher are the thresholds used for the elimination decisions. For that, a simple heuristic is used: the thresholds are set to the maximal value that allows all training data to succeed.

The final part of offline modelling in our system generates a rigid face model used in our online post-processing stage. To retrieve relative coordinate positions from the set of statistics on pairwise Euclidean distances, a spring particle simulator is used. The landmarks are considered as particles having a random initial position. They are all linked by springs with their equilibrium length equal to the mean distance between the two points they represent. The simulator runs until it stabilises in a coherent configuration which is used as the rigid face model.

## 5.5.2 Graph Matching

To recap, the graph matcher takes as input a query graph and a model graph and returns, for each node of the query, a list of probable candidate labels in the model with associated scores.

### 5.5.2.1 Seeding

First, scores for each possible association are computed by projecting the vector of normalised attribute scores into the LDA space. Only the model elements for which the score reaches a given threshold (the blue line in figure 5.11) are added to the list of candidates.

### 5.5.2.2 Elimination Rules

The graph-matching heuristic used here consists of a loop of elimination processes that continues until the system stabilises. At each iteration (see Algorithm 2 and Figure 5.19), the less probable candidates are eliminated for each element. A candidate is thought to be improbable if its direct neighbourhood gives it very little support. Two kinds of support are considered: the number of matching neighbours and the score attached to those matches.

In most graph matchers [Christmas et al., 1995] the neighbours of a vertex will be other vertices. Here we use a hypergraph matcher which implies that the neighbours of a vertex are the edges connected to it and vice versa. The edge’s scores help erase node candidates, and the node’s scores help erase edge candidates.

The process is first run using static matching scores. Once the system stabilises and stops eliminating candidates, the scores are normalised so that their sum over the candidates of one element is equal to one. The support thresholds are replaced by those adapted to the dynamic elimination and then the flag *dynamicScore* is set to *true*. Once this set of iterations stabilises (the current iteration hasn’t eliminated any candidates), the graph matching ends and returns the list of candidates for each element.

**Unoriented hyperedges** We eliminate the candidate target for an element if this element doesn’t have enough neighbours supporting this candidate hypothesis. This rule is quite simple as we just have to count how many of the neighbouring hyperedges have a candidate included in the neighbouring hyperedges of the model target node.

This kind of elimination is completely dual:

- if the element is a Node we count the number of supporting neighbors among the neighbouring Hyperedges.
- if the element is a Hyperedge we count the number of supporting neighbors among the neighbouring Nodes.

If the number of supportive neighbours is greater than or equal to the number of neighbours in the model minus a small error tolerance, then the candidate is kept.

One limitation of this method is that the counting doesn’t take into account the permutations possible for each hyperedge. For example in figure 5.20, a candidate is not always eliminated as it should be if we were looking at “same-type” neighbours.

**Permutation Relaxation** One problem with hypergraphs is the determination of the correspondence between the set of connections of one query hyperedge and the set of the corresponding model hyperedge. When trying to know if a hyperedge  $e = \{u_1, u_2, \dots, u_n\}$  of degree  $n$  matches a model hyperedge  $e_M = \{v_1, v_2, \dots, v_n\}$  we have to verify whether there

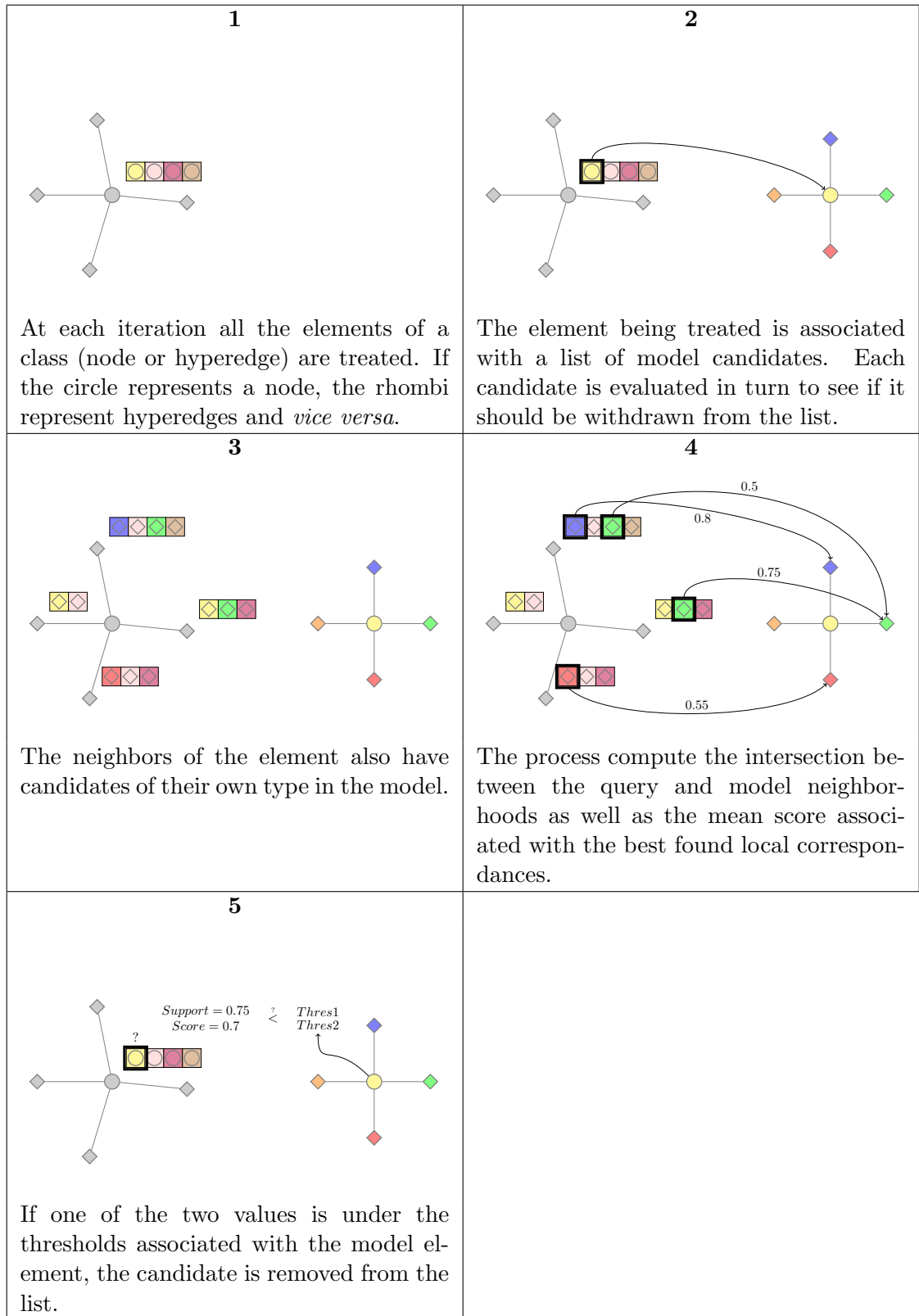


Figure 5.19: Visual explanation of the steps executed at each iteration of the relaxation by elimination process.



Figure 5.20: Example of one limitation of our simple relaxation system. Here the query graph (left) has two candidates (represented in curly brace) for its first node  $p$ . The elimination algorithm doesn't know that the other node  $q$  has only one candidate. The only thing known by the system is that one of the candidates ( $I$ ) for the touching hyperedge  $x$  belongs to the set of neighbouring hyperedges of both  $A$  and  $B$  in the model. Therefore the system will not erase  $B$  from the candidate list of  $p$ .

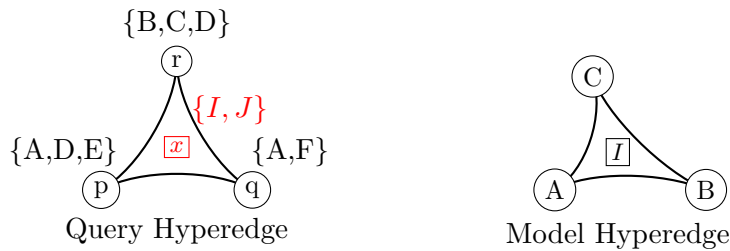


Figure 5.21: A model hyperedge of degree 3  $I$  (right) is a candidate for a query hyperedge  $x$  (left). Despite the fact the support is maximal for this hyperedge candidate, we see that it should be erased from the list of candidates of  $x$  because there is no permutation verifying equation 5.5. However, computing this lack of good permutation is computationally expensive in a dense non-oriented hypergraph.

exists a permutation  $\sigma$  such that

$$\forall i \in [1, n], \quad v_i \in \text{Cand}(u_{\sigma(i)}) \quad (5.5)$$

This makes it computationally expensive to use high degrees of connectivity for the hyperedges and therefore expensive to apply this technique to the dual hypergraph. The complexity becomes manageable if oriented hyperedges are used and/or if the set of hyperedges is relatively sparse. Figure 5.21 shows an example where it can be useful. This relaxation technique is implemented but not used in the proof-of-concept experiment.

### 5.5.3 Results

In controlled conditions on cropped, inexpressive frontal faces, our generic system labels the landmarks with an accuracy of 93.8%, with higher scores for the tip of the nose (97.0%) and

Table 5.2: Variables definition

$e$	Element type $\in \{Vertex, Edge\}$
$\bar{e}$	$\in \{Vertex, Edge\} \setminus e$ .
$e_i^Q$	$i^{th}$ query element of type $e$
$e_i^M$	$i^{th}$ model element of type $e$
$Cand(e_i^Q)$	list of model candidates $e_j^M$ for $e_i^Q$
$Neigh(e_i^X)$	list of element $\bar{e}_k^X$ connected to $e_i^X$
$Score(e_i^Q, e_j^M)$	Matching score between two elements
$dynamicScore$	Allow score to be updated
$TreshSup(e_j^M)$	Learnt Support Threshold
$TreshSco(e_j^M)$	Learnt Score Threshold

---

**Algorithm 2:** Pseudo-code for the elimination procedure called at each iteration of the graph matcher.

---

```

foreach  $e_i^Q$  in Query graph: do
   $totalScore = 0.0$  ;
  foreach  $e_j^M \in Cand(e_i^Q)$ : do
     $support = 0$ ;  $score = 0.0$  ;
    foreach  $\bar{e}_k^M \in Neigh(e_j^M)$ : do
       $sup = 0$ ;  $sco = 0.0$  ;
      foreach  $\bar{e}_l^Q \in Neigh(e_i^Q)$  do
        if  $\bar{e}_k^M \in Cand(\bar{e}_l^Q)$ : then
           $sup = sup + 1$ ;
           $sco = \max(sco, Score(\bar{e}_l^Q, \bar{e}_k^M))$ ;
        if  $sup > 0$ : then
           $support = support + 1$ ;
           $score = score + sco$  ;
       $score = score/support$ 
    if  $support < TreshSup(e_j^M)$  or  $score < TreshSco(e_j^M)$  then
       $Cand(e_i^Q) = Cand(e_i^Q) \setminus e_j^M$  # Erase candidate;
    else
      if  $dynamicScore$ : then # Update score
         $Score(e_i^Q, e_j^M) = Score(e_i^Q, e_j^M) * score$ ;
         $totalScore = totalScore + Score(e_i^Q, e_j^M)$ ;
    if  $dynamicScore$ : then # Normalise
      foreach  $e_j^M \in Cand(e_i^Q)$ : do
         $Score(e_i^Q, e_j^M) = Score(e_i^Q, e_j^M)/totalScore$ ;

```

---

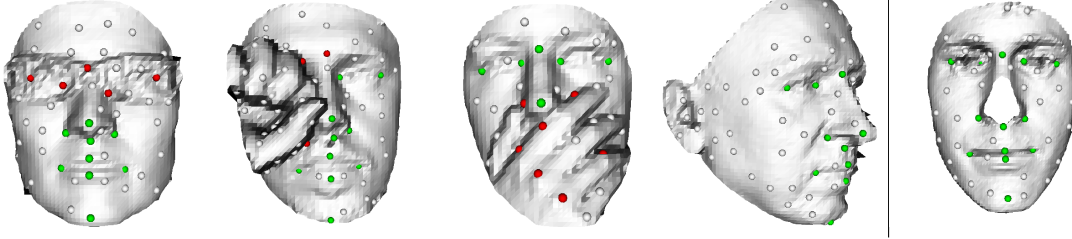


Figure 5.22: Example of results using the Bosphorus database and one using the FRGC database (right) where the nose is missing. Green dots represent good matches, red dots, false positives.

Table 5.3: Results per landmark (0-13) on the FRGC v2 database. The test set is split in two subparts: Neutral(-N) and Expression(-E), shown in the first two rows. The third row shows these results combined.

Train	Test	0	1	2	3	4	5	6	7	8	9	10	11	12	13
train(200)	test-N(3108)	<b>90.1</b>	<b>94.9</b>	<b>95.2</b>	<b>94.2</b>	<b>86.7</b>	<b>97.0</b>	<b>96.1</b>	<b>95.8</b>	<b>96.5</b>	<b>94.7</b>	<b>93.2</b>	<b>95.8</b>	<b>93.8</b>	<b>90.4</b>
train(200)	test-E(1642)	77.7	84.8	84.7	84.6	74.2	87.8	86.1	85.6	84.6	73.0	71.4	81.1	73.5	68.7
train(200)	test(4750)	85.8	91.4	91.5	90.9	82.4	93.8	92.6	92.3	92.4	87.2	85.7	90.8	86.8	83.0

Table 5.4: Results per landmark (0-13) on subsets of the Bosphorus database. The abbreviations used in the first two columns have been defined in Section 3.4.2.

Train	Test	0	1	2	3	4	5	6	7	8	9	10	11	12	13
N-train(99)	N-test(200)	<b>92.5</b>	<b>97.0</b>	<b>98.0</b>	<b>98.0</b>	<b>92.5</b>	<b>98.5</b>	<b>97.5</b>	<b>97.5</b>	<b>99.0</b>	<b>94.5</b>	<b>96.0</b>	<b>96.0</b>	<b>94.5</b>	<b>88.5</b>
N-train(99)	E(453)	77.0	87.1	86.5	87.6	75.0	90.2	88.9	88.3	87.1	68.4	65.5	81.6	71.0	55.4
N-train(99)	AU(2150)	84.8	92.2	91.7	92.5	86.5	92.7	92.0	92.2	91.2	75.7	72.2	82.5	74.1	65.5
N-train(99)	O(381)	68.9	78.8	74.5	78.9	69.9	83.4	82.6	82.7	84.8	82.4	81.5	84.8	81.0	73.0
N-train(99)	PR(419)	84.0	90.1	89.7	89.4	84.7	90.6	90.6	90.9	91.3	88.7	88.7	89.2	88.5	80.1
YR45-train(20)	YR45-test(85)	–	–	83.5	84.7	85.8	88.2	–	91.7	88.2	–	83.5	83.5	80.0	72.2
YL45-train(20)	YL45-test(85)	81.1	83.5	69.4	–	–	86.9	88.2	–	88.0	84.7	–	72.9	76.4	63.4
YR45-train(20)	CR(211)	–	–	70.1	72.3	70.1	77.6	–	79.5	77.7	–	72.0	75.3	71.5	65.2
Mean (3688-3984)		82.4	90.0	87.7	89.2	82.7	90.5	90.6	90.2	89.6	78.1	75.5	83.4	76.9	67.8



subnasale (96.5%), see table 5.3. The points that are more difficult to label are the outer corners of the eyes and chin (pogonion). This is not surprising, as these points don't have a very discriminating shape and can be easily misdetected if the final registration is not perfect.

On a more challenging database like the Bosphorus, we notice that our system does not give as good results as landmark-specific techniques (see Table 5.4). However, it performs quite consistently when occlusion and change in pose are considered, when most existing techniques will fail with such input. It should be emphasised that the hand-placed landmarks are not necessarily correlated with geometric extrema in this experiment; if the set of target landmarks is determined using such saliency, better results may be achieved.

Another interesting discovery is that the subnasale point, which is almost never used in automatic landmarking, is one of the most easily detected points in our experiments.

The time performance of the system when the graph has an average of 50 nodes is 0.67 s for the graph creation, 0.45 s for the graph matching and 0.04 s for the post-processing. The fact that complete graphs are used is costly when the number of nodes increases. Looking at different graph topologies (tessellation, etc) can help solve this problem.

#### 5.5.3.1 Conclusion

Our technique shows relatively good results on difficult test cases with changes in pose and occlusion. The main advantages of this technique are that it is landmark-independent and can learn several different models without human intervention and/or addition of specific rules.

The next step in our research will be to evaluate repeatability of automatic candidate detectors, and couple both point detection and labelling systems to produce an automatic landmarker. To our knowledge, this kind of global approach to face landmarking has never been evaluated before. While giving lower results than landmark-specific techniques for the tip of the nose and the eye corners, it is very promising for unconstrained 3D face landmarking when these points are difficult to localise. Finally, our method may also be used to landmark other kinds of objects such as bones, man-made objects and so on.

## 5.6 Global Matching Based on a Spectral Method

Because of its nature, hypergraph matching by relaxation can only eliminate bad candidates and is not usable on its own to detect final correspondences. Its main goal is to reduce the computational burden so that more expensive methods can be used on the remaining set of possible correspondences.

Strong correspondence finding (double semi-stochastic) can only be performed using global approaches. In this section, we present a global hypergraph matching technique very recently introduced by [Duchenne et al., 2009] and evaluate its advantages and drawbacks for our particular purpose. In order to test finely the robustness of the methods in dealing with spurious input data, we use synthetic data in this section where noise, sparsity and missed detections are controlled.

### 5.6.1 Duchenne Tensor Matching

In [Duchenne et al., 2009], an extension of the spectral matching method to hypergraphs is proposed. A previous global hypergraph-matching technique existed [Zass and Shashua, 2008] but it didn't use all the information of the input as the hyperedge matching scores were summed (independence assumption) into a 2D node-to-node matching matrix so that classic optimisation methods could be used. [Duchenne et al., 2009], on the other hand, use a tensor formulation. Their approach is to first find the main eigenvector (associated with the largest eigenvalue) of the tensor using a power method (continuous search of the node-to-node correspondence matrix) and then to use a greedy algorithm to select the doubly semi-stochastic solution (discrete projection). The solution  $V$  is computed from the tensor  $\mathbf{H}$  of the hyperedge matching scores using a power iteration method as shown in algorithm 3. It has to be noted that a tensor structure is not required to use this method. The code can easily be rewritten for our sparse hypergraph structure. Algorithm 3 is equivalent to algorithm 4 which is adapted to our sparse data structure. Every non-null value in  $H$  corresponds to a candidate score in  $Cand$ .

One obvious visible drawback of this approach compared to our relaxation scheme is that the candidate hyperedges have to be oriented, i.e.  $a_i$  should correspond to  $a_j$  and so on. No permutation is allowed. This works well for hyperedges of degree 3 corresponding

to triangles where the corners can be sorted using the angles. However, it doesn't work well for symmetric hyperedges such as hyperedges of degree 2 (using a simple Euclidean distance, for example). In that case, you want to keep the possibility that  $a_i$  is matched to  $b_j$  while  $b_i$  is matched to  $a_j$ . To select which permutation should be used, a simple test on the node candidate scores is performed: the permutation associated with the higher mean node assignation score is considered as correct (see Algorithm 5).

---

**Algorithm 3:** Tensor Power Iteration (Tensor Notation)

---

**Data:** Tensor of degree 3  $\mathbf{H}$  where  $\mathbf{H}_{a_i, a_j, b_i, b_j, c_i, c_j}$  is the score associated with the query hyperedge  $(a_i, b_i, c_i)$  matching the model hyperedge  $(a_j, b_j, c_j)$ .

**Result:**  $\mathbf{V}$  approximation of main eigenvector in  $\mathbf{H}$

initialisation;

**repeat**

$\mathbf{V} \leftarrow \mathbf{H} \otimes_1 \mathbf{V} \otimes_2 \mathbf{V}$  ;  
 $\forall k \quad \mathbf{V}[k, :] \leftarrow \frac{1}{\|\mathbf{V}[k, :]\|_2} \mathbf{V}[k, :]$  ;

**until** *Convergence* ;

---



---

**Algorithm 4:** Power Iteration for our Sparse structure of candidates in the case of degree 3 hyperedges.

---

**Data:** Lists of candidates  $Cand$ , Scores associated with each candidate

**Result:**  $\mathbf{V}_t$  approximation of main eigenvector in  $\mathbf{H}$

$\mathbf{V}_0 =$  initial score;

**repeat**

**foreach** Query hyperedge  $e_i^Q = (a_i, b_i, c_i)$  **do**  
**foreach** Model hyperedge  $e_j^M = (a_j, b_j, c_j) \in Cand(e_i^Q)$  **do**  
 $\mathbf{V}_t[a_i, a_j] \leftarrow Score(e_i^Q, e_j^M) \cdot \mathbf{V}_{t-1}[b_i, b_j] \cdot \mathbf{V}_{t-1}[c_i, c_j]$  ;  
 $\mathbf{V}_t[b_i, b_j] \leftarrow Score(e_i^Q, e_j^M) \cdot \mathbf{V}_{t-1}[a_i, a_j] \cdot \mathbf{V}_{t-1}[c_i, c_j]$  ;  
 $\mathbf{V}_t[c_i, c_j] \leftarrow Score(e_i^Q, e_j^M) \cdot \mathbf{V}_{t-1}[b_i, b_j] \cdot \mathbf{V}_{t-1}[a_i, a_j]$  ;

$\forall k \quad \mathbf{V}_t(k, :) \leftarrow \frac{1}{\|\mathbf{V}_t[k, :]\|_2} \mathbf{V}_t[k, :]$  ;

**until** *Convergence* ;

---

### 5.6.2 Data Generation

For every possible combination of parameters (noise, extra points, missing points), 25 pairs of 2D hypergraphs, each containing 25 points, are generated. The points are randomly picked in the unit square. Noise is then applied to the point positions, some points are erased and some random points are added.

---

**Algorithm 5:** Power Iteration for our Sparse structure of candidates for degree 3 (ordered) and 2 (symmetric).

---

**Data:** Lists of candidates  $Cand$ , Scores associated with each candidate

**Result:**  $V_t$  approximation of main eigenvector in  $H$

$V_0$  = initial score;

**repeat**

```

foreach Query hyperedge  $e_i^Q$  do
  foreach Model hyperedge  $e_j^M \in Cand(e_i^Q)$  do
    if  $degree(e_i^Q) = 3$  then # ordered
       $V_t[a_i, a_j] \leftarrow Score(e_i^Q, e_j^M) \cdot V_{t-1}[b_i, b_j] \cdot V_{t-1}[c_i, c_j]$  ;
       $V_t[b_i, b_j] \leftarrow Score(e_i^Q, e_j^M) \cdot V_{t-1}[a_i, a_j] \cdot V_{t-1}[c_i, c_j]$  ;
       $V_t[c_i, c_j] \leftarrow Score(e_i^Q, e_j^M) \cdot V_{t-1}[b_i, b_j] \cdot V_{t-1}[a_i, a_j]$  ;
    else if  $degree(e_i^Q) = 2$  then # symmetric
      if  $a_j \in Cand(a_i)$  and  $b_j \in Cand(b_i)$  then
         $score_1 \leftarrow Score(a_i, a_j) \cdot Score(b_i, b_j)$ ;
      if  $b_j \in Cand(a_i)$  and  $a_j \in Cand(b_i)$  then
         $score_2 \leftarrow Score(b_i, a_j) \cdot Score(a_i, b_j)$ ;
      if  $score_1 < score_2$  then
         $a_i, b_i \leftarrow b_i, a_i$ 
       $V_t[a_i, a_j] \leftarrow Score(e_i^Q, e_j^M) \cdot V_{t-1}[b_i, b_j]$ ;
       $V_t[b_i, b_j] \leftarrow Score(e_i^Q, e_j^M) \cdot V_{t-1}[a_i, a_j]$ ;

```

```

 $\forall k \quad V_t[k, :] \leftarrow \frac{1}{\|V_t[k, :]\|_2} V_t[k, :]$  ;

```

**until** Convergence ;

---

The noise parameter is a scalar value  $n$ . Each point position is offset by a vector of random angle and signed radius following a null-centred Gaussian distribution of deviation  $\sigma$  where  $\sigma$  is equal to  $n$  times the mean Euclidean distance between the 25 points.

The “missing points” parameter is an integer representing how many of the good matchable points are erased from the query graph. The “extra points” parameter is an integer representing how many non-matchable points are added to the query graph.

Here the problem to solve is a graph-to-graph matching problem and no longer a graph-to-model matching problem. However, unlike most experiments doing graph-to-graph matching, the degradation is only applied to the first graph as our final goal is to transpose these results for a graph-to-model matching of an imperfect query to a perfect model.

Once the points are defined, a 3-uniform hypergraph is constructed over each set. Each hyperedge is a set of three vertices ordered according to the sines of its corresponding angles.

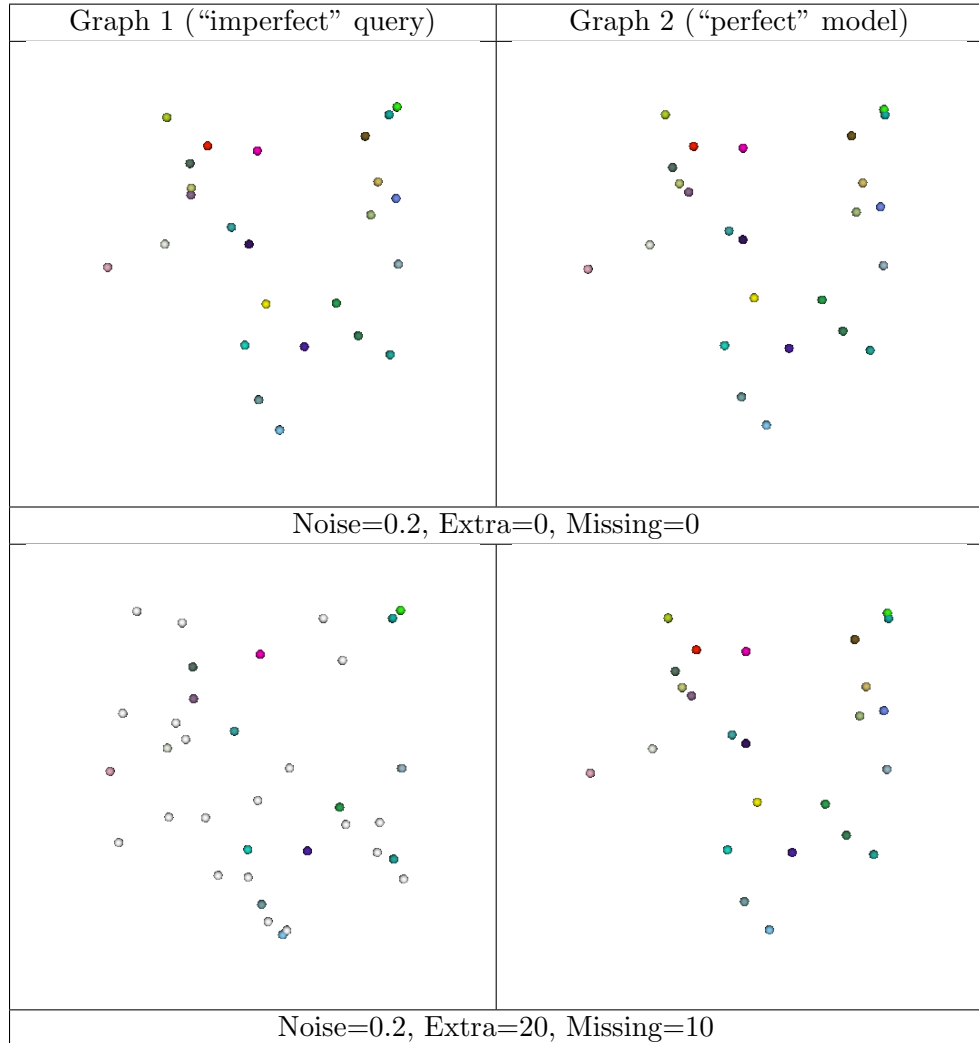


Figure 5.23: Example of synthetic set of points for hypergraph matching evaluations.

The descriptors are stored for each hyperedge are the three sines angles  $d_k$  plus the orientation of the triangle (direct or indirect) used as an eliminatory descriptor  $d_{DIR}$ . The matching score for a triangle is computed as the mean of the individual descriptor scores:

$$S(e_i, e_j) = \begin{cases} \frac{1}{3} \sum_k \exp(-|d_k(e_i) - d_k(e_j)|) & \text{if } d_{DIR}(e_i) = d_{DIR}(e_j) \\ 0 & \text{otherwise} \end{cases}$$

### 5.6.3 Effect of Missing Points

In figure 5.24 matching results for an increasing number of missing query points are shown when varying the parameters for noise and extra point number. While the RANSAC-base solution gives stable results when the number of missing points increases, the Tensor-base method produces more false positives. This was predictable as it tries to find a stochastic solution. What is interesting is that the false negatives are clearly lower with the Tensor-base method.

### 5.6.4 Effect of Stepping

The idea of stepping consists in completing the journey to the solution in several steps instead of just one. In the case of *Duchenne* Tensor Matching, stepping can be done by running the method a first time and projecting the discrete value in a looser way (for example, by accepting the  $N$  best match for each model element instead of just one). By eliminating the impossible hyperedges using this configuration and by running the method a second time on this new input we expect that better results can be achieved. In practice, we see a noticeable (but not large) improvement in going from one step to two, but very little more if a larger number of steps is used. Figure 5.25 shows results using 1-step and 2-step methods.

## 5.7 Conclusion

Existing matching techniques can be very robust in dealing with additional dummy points but remain very sensitive to missing data. The problem is the local management of the global amount of acceptable missing data. It is difficult for a distributed system to determine where the allowance for missing data should be used. If it is used everywhere at the same time, almost everything become acceptable and the matching process doesn't converge to an acceptable solution.

In this chapter we have shown that if keypoints corresponding to known landmarks are present in the input it is possible to retrieve their label even when a large number of dummy keypoints are present. To do this, we introduced a new hypergraph matching technique by relaxation alternating between node and hyperedge eliminations as well as two rigid registration methods adapted for hypergraphs constructed over 3D surface meshes. We also

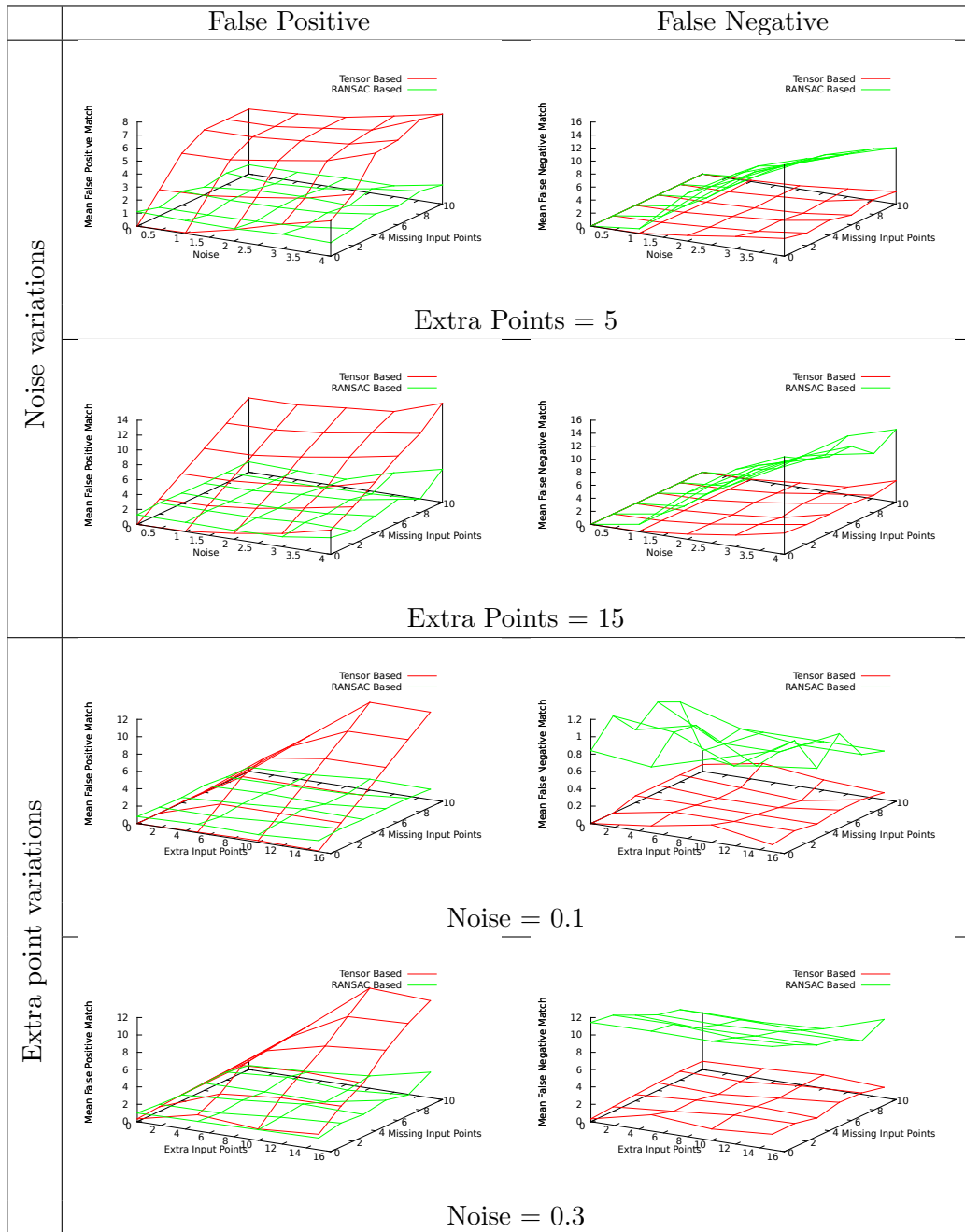


Figure 5.24: Effect of missing data in the query on the matching results.

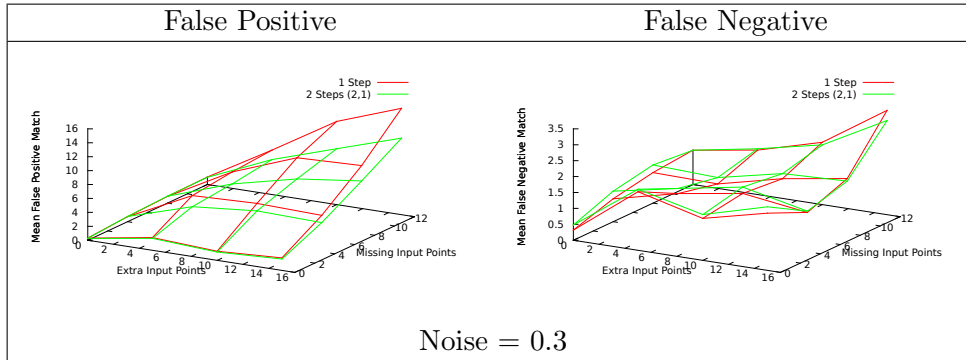


Figure 5.25: Effect of stepping with extra and missing nodes for the tensor-based method with a direct match and a two-step match.

highlight some advantages of using multi-degree multi-attributed hypergraphs over more classic schemes for structural matching.

Finally, we evaluated a tensor-based hypergraph matching technique on synthetic data and showed that it produced softer solutions than rigid-model-based techniques but cannot cope as efficiently with missing data. When dealing with inputs presenting occlusions, the use of classic registration methods for the final doubly semi-stochastic correspondence remains the best choice so far.

Retrospectively, it appears that our problem breakdown strategy had one flaw. We assumed that the artificial problem P2 was more simple than the real life problem P3. It is in fact harder as the automatic points used in the artificial data are much more dense than what can be obtained with our automatic keypoint detector. In the next chapter it appears that the keypoints in problem P3 are sparser and less ambiguous, making the matching part less problematic than expected.



## Chapter 6

# A Machine-Learning Approach to 3D Face Landmarking

The head is included in the body, but the face is not. The face is a surface: facial traits, lines, wrinkles; long face, square face, triangular face; the face is a map, even when it is applied to and wraps a volume, even when it surrounds and borders cavities that are now no more than holes. The head, even the human head, is not necessarily a face. The face is produced only when the head ceases to be a part of the body, when it ceases to be coded by the body, when it ceases to have a multidimensional, polyvocal corporeal code when the body, head included, has been decoded and has to be overcoded by something we shall call the Face. This amounts to saying that the head, all the volume-cavity elements of the head, have to be facialized. What accomplishes this is the screen with holes, the white wall/black hole, the abstract machine producing faciality.

---

Extract from *Thousand Plateaus* by Gilles Deleuze and Felix Guattari.

Editions de Minuit, 1980

Translated from the French by Brian Massumi.

In this chapter, we look at the automatic landmarking of 3D faces. The output of our keypoint detector is used as the input of a structural matching process of landmark labelling. This framework is able to robustly retrieve landmarks on new query faces, even in difficult conditions where salient regions of the face have been occluded. Our approach shows better

results than the state-of-the-art methods for 3D face landmarking, while being more generic and more efficient on non-standard pose capture.

## 6.1 Introduction

As explained in Chapter 3, our high-level strategy is to use an automatic keypoint detector to reduce the set of input vertices to a sparse set before running a labelling system so as to select the subset of points corresponding to known landmark labels. The positions of the points can then be refined using holistic or local optimisation methods. In Chapter 4 we presented a keypoint detector using a dictionary of local shapes, capable of giving a sparse selection of interesting points of 3D objects. In Chapter 5 we presented several techniques to determine correspondences between a learnt model and a query set of points over a 3D mesh and evaluated them for application in landmark labelling on faces. We also noted that graph and hypergraph matching techniques can be very efficient in reducing ambiguities but are difficult to use for final correspondences when occlusions are present in the query (missing data). The doubly semi-stochastic correspondences are often obtained using some form of rigid model, either piecewise registration (we used a unit-quaternion clustering technique) or holistic registration (we used a RANSAC technique). Here, we combine an instance of our keypoint detector and a modified RANSAC registration technique to obtain an automatic machine-learning-based 3D landmarking system. This framework outperforms the state-of-the-art landmarking systems on large, public datasets, such as the FRGC and the Bosphorus datasets.

### Rationale

**What:** To localise a set of landmarks on a 3D surface.

**Why:** To make sense of the input data as being a face, with a known orientation, and known facial features positions.

**How:** By pipelining the output of a keypoint detector into a labelling system.

**Priorities:** Robustness to occlusion and pose variation, speed.

## 6.2 Experimental Framework

In this chapter, the input of our system is a 3D face mesh and a learnt model of the face. The output are landmarks (a set of pairs: position+label). To evaluate the overall results for this experiment two kinds of measures are produced:

- The individual manual ground-truth landmarks are used as comparisons. This allows us to look at the retrieval rates and localisation errors (in mm). These measures are landmark-based metrics.
- The manual ground-truth landmarks used as a whole can provide us with a ground-truth global registration. Comparing the transformation obtained by the system with this ground-truth transformation provides us with a set of global metrics (rotation, translation and scale) for one query face.

### 6.2.1 Workflow

The landmarking framework is based on the keypoint detection system. The first steps are exactly the same as described in Chapter 4. However, this time, the landmark score maps are not merged into a final keypoint score map. The keypoints are detected on each landmark score map separately, providing us with landmark candidates directly (see Figure 6.1). This leads to more points, but with the advantage that only one label is associated with each landmark candidate. The final labels are selected by fitting a scale adapted rigid model of the targets to the query using a Random Sample Consensus (RANSAC) approach. The output landmarks are defined as the projection of the registered model points onto the face. The model fitting is not only labelling the point, but is also adjusting the position of the points using the global registration associated with the model parametrisation within RANSAC. This has the advantage of providing landmarks even in face regions that have missing or spurious data. The threshold for keypoint detection on the landmark score maps is set to 0.75 to reduce the number of false negative landmark candidates and therefore speed up the matching process.

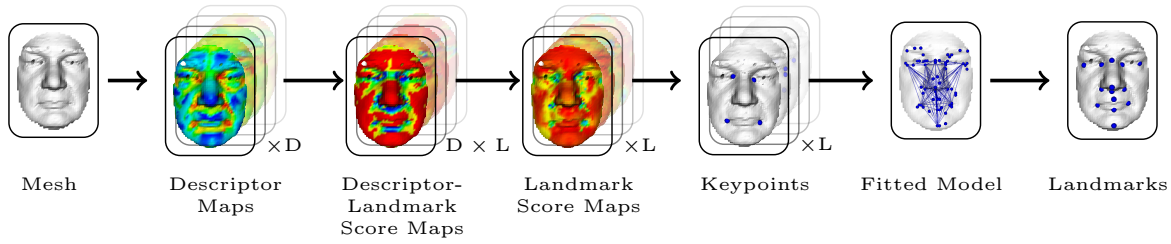


Figure 6.1: Workflow of the landmarking system.

## 6.2.2 Model-fitting

In order to establish a one-to-one correspondence between the query landmark candidates and the model landmarks, we need to use configural information related to the global geometry of the face. Here the simplest possible approach is chosen: to register a scale-adapted rigid model of the face<sup>1</sup> onto the query landmark candidates. The choice to not use more complex techniques for matching disambiguation as a first step came from the fact that the set of detected keypoints produced by our final detector was a lot sparser than we were expecting. This reduced the potential gain of more computationally expensive alternative techniques for this particular case.

### 6.2.2.1 Scale-Adapted Rigid Registration

A registration between two sets of labelled points is a geometric rigid transformation that can be represented as a 4x4 transformation matrix  $\mathbf{T}$ , with 7 degrees of freedom (3 for rotation, 3 for translation and one for scale). In our case, the query keypoints only have one candidate label each. Therefore, finding the best registration is equivalent to finding the best subset of keypoints in the query, i.e. those that best support a specific transformation. To do so a RANdom SAMple Consensus (RANSAC) approach is used. RANSAC is a model fitting meta-algorithm introduced in [Fischler and Bolles, 1981] that consists of selecting a set of inputs elements that agree with the same parametrisation of the input model. In our case the input elements are point correspondences, the model is a scale-adapted rigid model of the target landmarks, and its parametrisation is the 4x4 transformation matrix  $\mathbf{T}$ . To find the parametrisation with the largest number of supporting keypoints,  $N$  sets of keypoint triplets

<sup>1</sup>The rigid model was obtained using a spring-particle simulator where the zero-length of each spring between a pair of landmarks is fixed to the mean Euclidean distance between the two points.

are randomly sampled to instantiate the model. The “consensus” function of the algorithm tells whether a input point agrees with the current solution by looking at two conditions:

- the distance between the query point and its corresponding transformed model landmark should be under a given threshold
- the dot product between the unit normals of the query and model points should be above a given threshold.

If both conditions are true, the given correspondence is part of the consensus. At each iteration of the RANSAC method, a new transformation  $\mathbf{T}$  is computed by using a linear least-squares method.

#### 6.2.2.2 Dealing with Symmetry

A known problem with our approach is that the local shapes in our dictionary can be correlated, especially when associated with symmetric regions. For example, a keypoint detected near the outer right corner of the eye has a big chance of being a landmark candidate for its left counterpart. This can lead to upside-down face detection (where the registered model has a near  $180^\circ$  error in roll angle), or simply to imprecision in the global registration, as some points might be discarded in the fitting process. The RANSAC solution depends on a list of pairs (point, label) given as inputs. As the quality of the labels is uncertain at this point, running the RANSAC algorithm with different inputs can help cover a bigger search area. Selecting the best RANSAC solution among several can easily be done by looking at mean projection distance to the mesh.

Running RANSAC several times with different seedings can only improve the chances of finding the best match. However, designing meaningful seedings is not always easy. In this experiment, RANSAC is run twice with two different starting sets of correspondences. One in which the keypoints are associated with the labels derived from the score maps, and a second in which the keypoints are associated with one or two labels depending on whether the initial label belongs to a symmetric pair. One of the two transformations is selected as the best if its corresponding projection distance to the surface mesh is minimal.

On the 4750 scans from the test set on the FRGC, 2564 obtain better fitting with the first seeding, while 2186 do so with the second. In most cases the difference between the

mean projection distances is very small. In 99.31% of the cases, the mean distance between the two solutions is under 2 mm.

### 6.2.2.3 Projection

Once the transformation has been defined, all model landmarks are associated with the closest vertices on the query mesh. Those positions and their associated labels are defined as landmarks and are the outputs of our system. This part of the process is relatively naive, as the projection is always computed even if the distance to the mesh is unrealistic for a good registration. At this stage it would be possible to detect some of the bad registrations (with large differential between individual projection distances) and recompute the whole process with different parameters.

## 6.3 Results

The following tests have been executed using configuration 2 of our keypoint detection system (see Chapter 4) where 10 descriptors (8 scalars, 2 histograms) are used at a single scale/size. The landmarks obtained using our technique can be downloaded on my webpage<sup>2</sup> to help future results comparison. Figure 6.2 shows the landmark retrieval rate for an increasing acceptance radius.

Landmarking results are often difficult to compare due to the variety of datasets, pre-processing and performance metrics. Obviously, the bigger the dataset the more meaningful the results are. Here our method is compared with previous studies that give results on at least 4000 models from the FRGC v2 dataset. In Table 6.1, comparison with state-of-the-art methods is presented at the most commonly used acceptance radii for human face landmarking (10, 12, 15 and 20 mm).

Our system does not outperform all recent expert-system based techniques in terms of precision (at 10 mm), but it does so in terms of robustness (using commonly used acceptance radius) while presenting some unique capabilities. Firstly, the number of discrete failures is zero. The system always succeeds at coarsely registering the face and finding some correct landmarks. Indeed 100% of the models have at least three points retrieved at 20 mm.

---

<sup>2</sup><http://www.cs.york.ac.uk/~creusot>

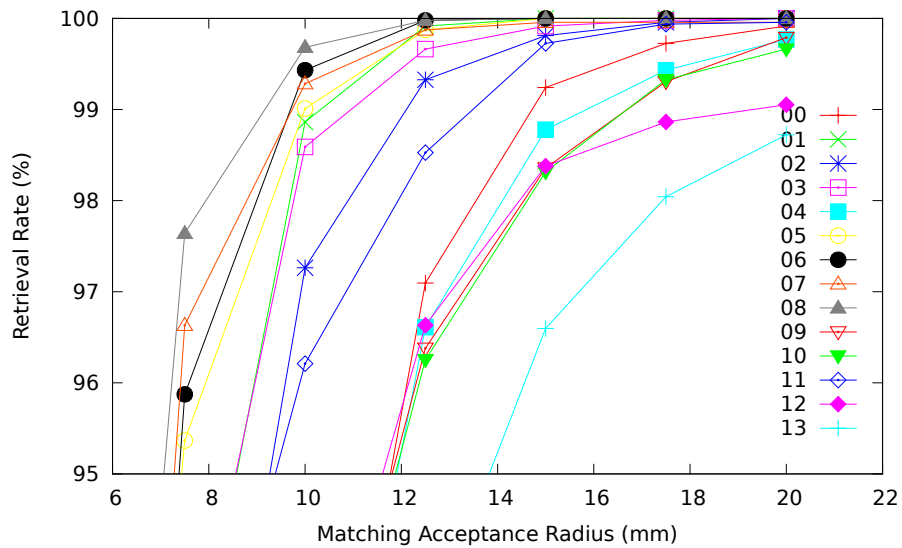


Figure 6.2: Landmark retrieval rate for the 14 landmarks on the FRGC test set.

Secondly, our system is generic, allowing us to detect  $L$  (14) shapes using exactly the same method for each of them. Thirdly, due to the landmark independence of the system and the non-sequentiality of the candidate search, our system has no trouble dealing with faces with missing key features, for example when the nose is missing (see Figure 6.3). Most existing techniques will fail in such cases, as landmark presence is required in the query scan. This confers a great advantage on our system when dealing with occlusions as observed in real life scenarios (for example, those that include pose variation, occlusions by hands, cell phone, spectacles, and so on). In addition, our system is invariant to rotation of the 3D surface as seen in Figure 6.4. Figure 6.5 shows results of our landmarking system on big input meshes from the FRGC. The error in positioning relative to the ground truth are presented in Figure 6.6.

The experiment has also been run on the Bosphorus dataset [Savran et al., 2008]. This dataset hasn't been used very often in the literature and is therefore less convenient in comparing results with existing techniques. However it is far more challenging in terms of non-standard pose capture with scans presenting large pose variation as well as occlusions. Results are given on this dataset for two main reasons:

- to highlight some limitations of our technique that could not have been detected with the FRGC dataset alone.

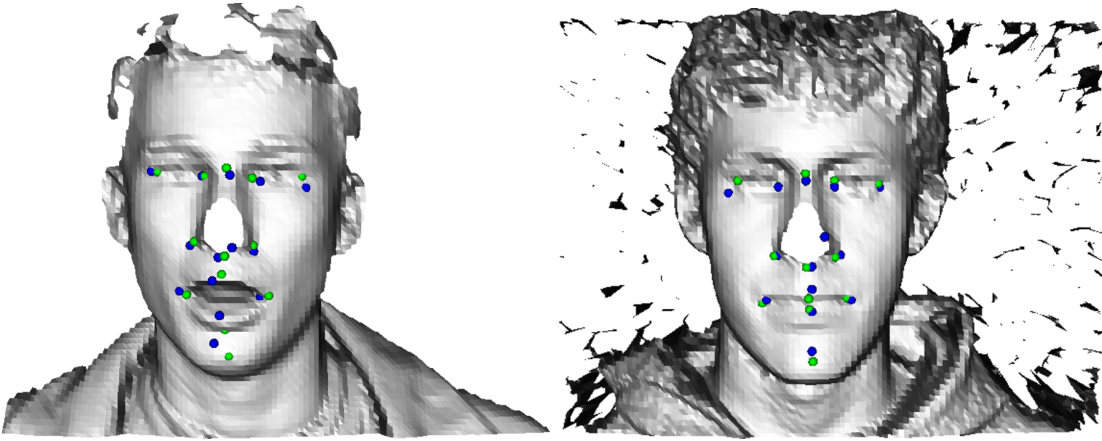


Figure 6.3: Examples of landmarking in cases with missing noses where expert systems usually fail (model 04814d22 and 04505d222 of the FRGC). Our system doesn't need the nose tip to be correctly detected in order to find the other landmarks (landmark independence). Blue points represent our results and green points represent the ground truth.

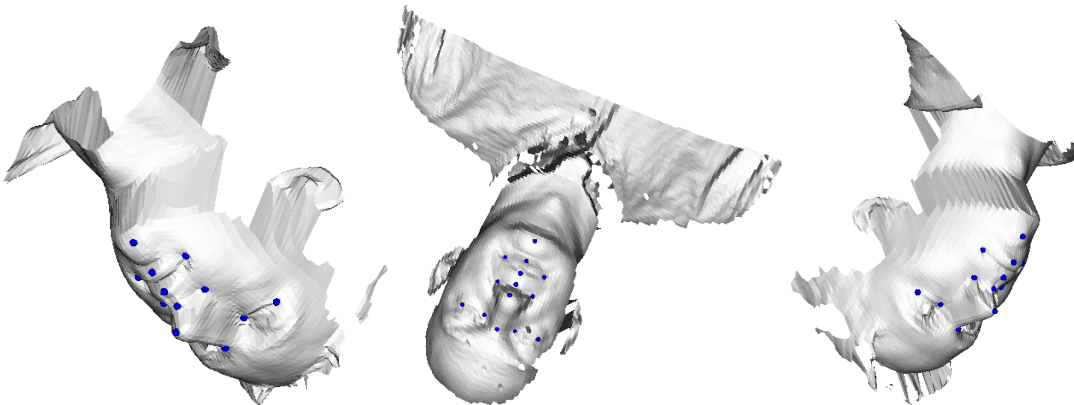


Figure 6.4: Examples of localisations on rotated meshes. Our system only uses relative vertex positions and normals and is therefore translation and rotation invariant (pose invariant).



Table 6.1: 3D face landmarking systems that are tested on more than 4000 models from the FRGC v2 dataset. Results using the same metric are coloured in the same colour. When a comparison is possible, results in bold font highlight the best system score for the given metric.

Authors	[Chang et al., 2006]	[Mian et al., 2006b]	[Segundo et al., 2007]	[Romero and Pears, 2009a]	[Alyuz et al., 2010]			[Segundo et al., 2010]		This Work, 2011			
#Landmarks	3	1	6	3	5			5		14			
Acceptance Radius	<?	<?	<?	< 12	< 10	< 12	< 20	< 10	< 15	< 10	< 12	< 15	< 20
Nose (05)	99.40	98.3	99.95	99.77	99.62	99.80	99.87	99.95	99.95	99.01	99.81	100.0	100.0
Eye Inner Corners (01,03)	-	-	99.83	96.82	96.59	98.54	99.54	99.02	99.64	98.73	99.71	99.96	100.0
Nose Corners (06,07)	-	-	99.76	-	98.60	99.29	99.87	99.35	99.95	99.36	99.87	99.98	99.98
Subnasale (08)	-	-	99.98	-	-	-	-	-	-	99.68	99.98	100.0	100.0
Mouth Corners (09,10)	-	-	-	-	-	-	-	-	-	91.33	95.63	98.34	99.73
Eye Outer Corners (00,04)	-	-	-	-	-	-	-	-	-	89.84	95.92	99.01	99.84
Nasion (02)	-	-	-	-	-	-	-	-	-	97.26	99.07	99.81	100.0
Upper Lip (11)	-	-	-	-	-	-	-	-	-	96.21	98.21	99.73	99.96
Lower Lip (12)	-	-	-	-	-	-	-	-	-	92.04	96.00	98.38	99.05
Chin (13)	-	-	-	-	-	-	-	-	-	84.94	91.96	96.60	98.72
Candidate Selection	ES	ES	ES	ES	ES			ES		ML			
Independence	no	n/a	no	yes	no			no		yes			
Test Size	4,485	4,950	4,007	4,013	4,007			4,007		4,750			
Train Size	-	-	-	-	-			-		200			
Pre-processing	S,C <sup>1</sup>	∅	H,C	S,H	S,H,C			S,H,C		∅			
Pre-processing Time	-	-	1.1s	-	-			1.0s		0s			
Processing Time	-	-	0.4s	-	-			0.3s		1.18s			

ES: Expert System, ML: Machine Learning, C: Cropped/Segmented, H: Hole Filling, S: Spike Removal

<sup>1</sup> In [Chang et al., 2006] the mesh were cropped using 2D texture (skin colour).



Figure 6.5: Examples of landmarks and associated graphs (helping visualisation) detected by our system on large input meshes from the FRGC.

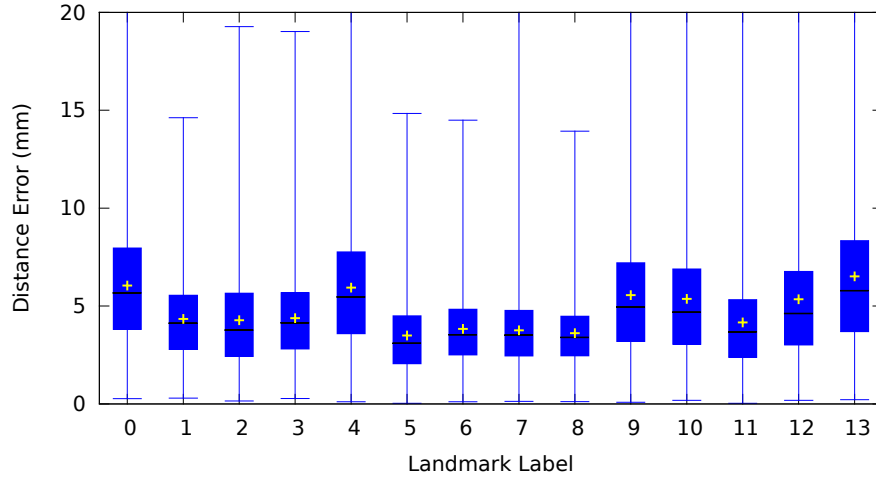


Figure 6.6: Distance error for the 14 landmarks on the FRGC test set. The candlestick represents the min/Q1/median/Q3/max values. The plus sign represents the mean.

- to provide enough data to allow results comparison with this dataset in future research.

Table 6.2 contains the landmark retrieval rates for the different parts of the dataset, as well as for the whole set. Scans of yaw rotation marked as  $90^\circ$  were not used due to their poor associated descriptor maps (higher resolution might be necessary to treat those cases). Moreover, scans marked as IGN (ignored) in the dataset have also been discarded. In total we tested 4339 face scans from the Bosphorus dataset. Figure 6.7 shows examples of landmark localisation on some of these scans.

#### 6.3.0.4 Limitations

A limitation of our system is that it relies heavily on local descriptors. Therefore, the robustness of the system strongly depends on the robustness of the local descriptors used. If these are not robust to occlusion the local score values might be spurious and detection quality will suffer. For example, in a profile view, when a vertex is near the border of the mesh, its local neighbourhood is incomplete and the descriptors computed on this neighbourhood are likely to be noisy. It might be interesting to develop different specialised descriptors that can detect keypoints near the border of the mesh using, for example, extracted 2D curves

Table 6.2: Results on the more challenging Bosphorus dataset using our landmarking method. The training set is composed of 99 faces from the neutral subset of the dataset. The results are provided for the four most commonly used acceptance radii in landmarking to facilitate future result comparison.

Test Sets (size)	Acceptance Radius	Eye Outer Corners (L) (00)	Eye Inner Corners (L) (01)	Nasion (02)	Eye Inner Corners (R) (03)	Eye Outer Corners (R) (04)	Nose (05)	Nose Corners (L) (06)	Nose Corners (R) (07)	Subnasale (08)	Mouth Corners (L) (09)	Mouth Corners (R) (10)	Upper Lip (11)	Lower Lip (12)	Chin (13)
Neutral (200)	< 10	93.00	100.0	96.50	99.50	90.50	98.50	99.00	99.50	100.0	97.00	96.50	99.00	96.00	58.50
	< 12	98.00	100.0	99.00	100.0	94.00	99.50	99.50	99.50	100.0	99.00	98.50	100.0	98.50	69.00
	< 15	99.50	100.0	99.50	100.0	99.50	100.0	100.0	100.0	100.0	100.0	99.50	100.0	100.0	84.00
	< 20	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	95.50
Emotions (453)	< 10	85.43	99.34	93.60	98.68	85.21	94.26	97.57	96.25	99.12	65.34	67.33	90.51	71.08	26.39
	< 12	92.27	99.56	97.35	99.34	93.38	98.68	99.12	99.78	99.56	75.50	76.60	95.36	79.47	34.59
	< 15	97.79	99.78	99.78	99.56	98.23	99.34	99.78	99.78	99.78	84.11	85.21	97.57	83.66	47.45
	< 20	99.56	99.78	99.78	99.56	99.56	100.0	99.78	99.78	99.78	95.81	94.92	99.56	86.09	66.52
Action Units (2150)	< 10	88.93	98.60	93.81	98.37	86.41	95.44	97.95	98.79	98.79	68.30	70.50	89.48	72.59	35.24
	< 12	95.21	99.63	97.49	99.40	93.30	97.91	99.16	99.72	99.67	78.03	80.97	95.30	81.06	44.62
	< 15	99.16	99.95	99.49	99.77	98.28	98.79	99.81	99.86	99.91	88.97	91.07	98.37	87.72	58.18
	< 20	99.86	99.95	99.86	99.91	99.77	99.12	99.91	99.91	99.91	97.30	97.77	99.53	90.88	75.62
Occlusions (381)	< 10	77.31	95.00	91.67	97.02	83.95	94.71	94.57	94.57	97.52	92.98	96.23	96.58	92.45	56.03
	< 12	86.15	96.67	96.39	98.10	88.25	96.83	97.43	97.01	98.45	97.54	97.60	97.95	95.68	67.38
	< 15	95.00	97.33	98.06	98.64	95.70	98.41	98.29	98.10	98.76	98.60	98.97	98.97	98.20	79.08
	< 20	98.08	98.33	99.17	98.92	97.71	98.68	98.57	98.91	100.0	98.60	99.32	99.32	99.28	92.20
Yaw Rotations (up to 45°) (525)	< 10	80.05	89.19	85.52	98.33	91.19	89.89	95.41	99.05	94.65	95.15	96.90	94.29	88.19	49.04
	< 12	90.29	91.08	91.81	99.05	94.76	95.23	97.25	99.52	96.75	96.94	97.86	96.19	93.90	61.73
	< 15	95.28	95.68	97.14	99.52	98.10	98.66	98.17	99.52	98.09	98.21	99.29	98.10	98.10	75.38
	< 20	97.90	97.84	97.90	99.52	99.29	99.05	98.62	99.52	99.04	99.49	99.52	99.05	98.29	91.15
Pitch Rotation (419)	< 10	89.98	98.33	94.27	99.28	84.25	93.32	98.33	98.81	98.79	94.51	96.90	97.37	94.98	55.56
	< 12	95.70	99.28	98.09	99.76	91.65	95.47	99.52	99.52	99.52	97.37	97.85	98.57	97.13	66.67
	< 15	98.57	99.76	99.05	99.76	97.37	97.85	99.76	99.76	99.76	99.28	98.81	99.76	98.56	80.43
	< 20	99.76	99.76	99.52	99.76	99.28	98.33	100.0	100.0	100.0	99.52	99.52	99.76	99.04	92.03
Cross Rotation (Yaw and Pitch) (211)	< 10	73.17	83.33	85.78	96.67	87.20	90.00	81.82	98.10	95.73	85.61	93.36	94.31	90.05	49.28
	< 12	84.15	85.29	92.89	100.0	91.94	94.76	90.91	99.52	98.58	91.67	95.73	97.16	94.79	62.32
	< 15	91.46	93.14	98.10	100.0	98.10	98.57	90.91	100.0	100.0	97.73	100.0	100.0	98.58	78.26
	< 20	98.78	99.02	99.05	100.0	100.0	100.0	93.94	100.0	100.0	100.0	100.0	100.0	100.0	91.30
All the above (4339)	< 10	86.87	97.17	92.40	98.34	86.65	94.28	97.44	98.20	98.17	77.04	79.80	92.16	79.74	41.08
	< 12	93.74	98.22	96.59	99.34	92.86	97.23	98.87	99.41	99.20	84.54	86.70	96.28	86.54	51.34
	< 15	98.15	99.15	99.03	99.64	98.02	98.75	99.50	99.67	99.58	91.93	93.48	98.63	91.47	64.99
	< 20	99.49	99.57	99.49	99.74	99.50	99.17	99.66	99.79	99.81	97.89	98.07	99.55	93.58	80.92

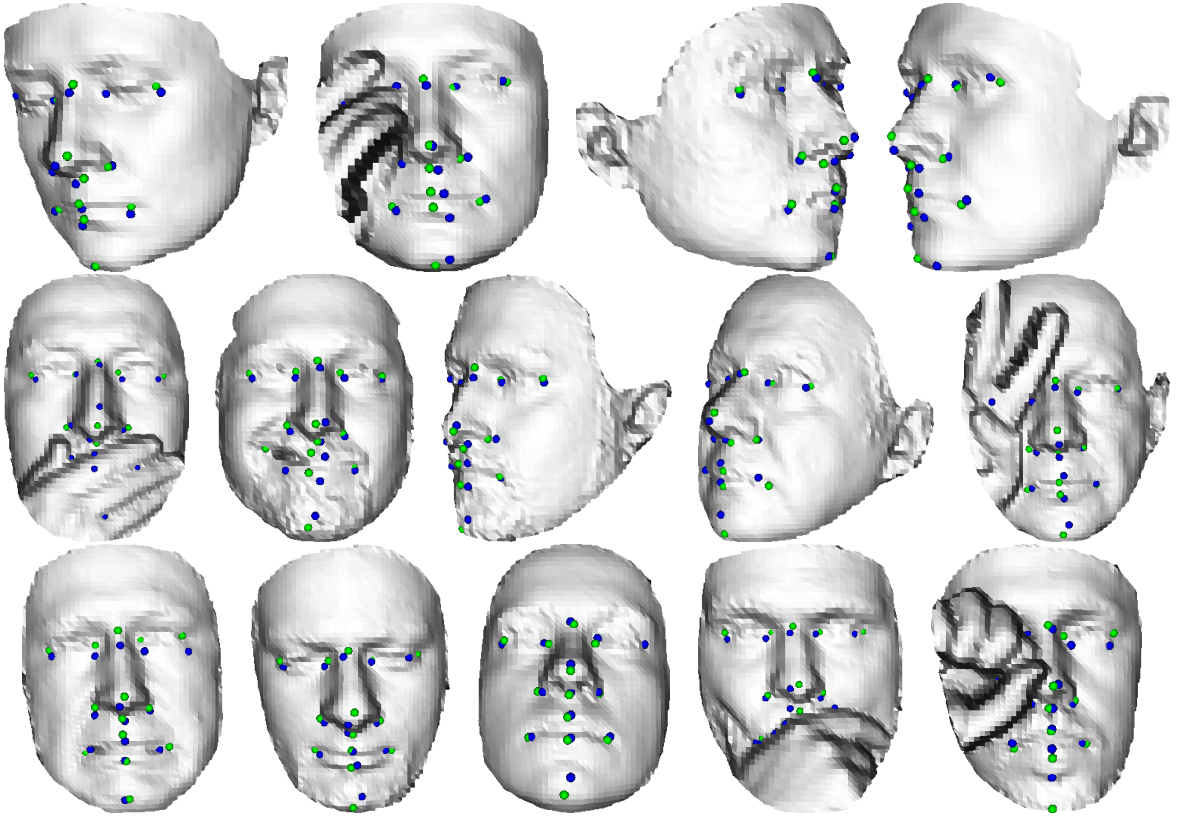


Figure 6.7: Examples of landmark localisation on the Bosphorus dataset. Blue points are our automatic results, green points are the ground truth (manually marked up).

along the direction of occlusion. Combining the score using these new descriptors can easily be done within our framework.

In Figure 6.8, the worst-case landmark localisations in terms of the three global registration metrics are presented for the FRGC test set. When registering the ground-truth and the localised landmarks, the transformation is decomposed by steps into a mean translation (aligning the centroids), a scale change (by scaling the mean edge length) and a final rotation, when translation has been cancelled and scale equalised. The mean error in translation is represented by  $\tau$  in millimetres. The scale difference is represented by the ratio  $\rho$  of the ground truth to the detected mean length. The final rotation error is defined as the angle  $\theta$  between the unit-quaternion representing the computed rotation and the one representing the identity. Figures 6.9 and 6.11 show the distribution of these errors for the FRGC and Bosphorus test sets.

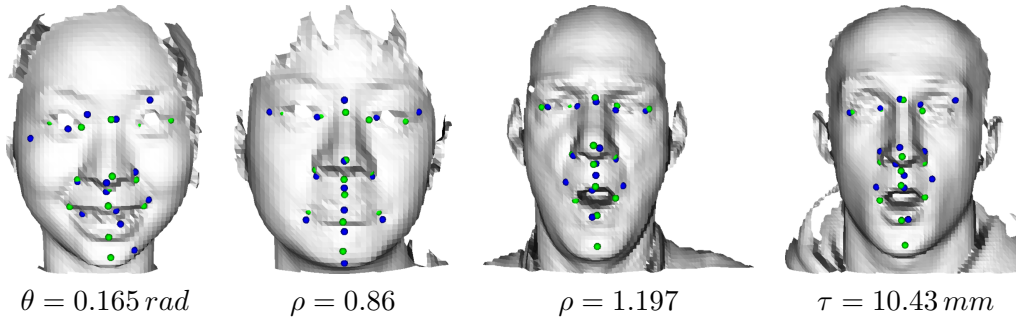


Figure 6.8: Four worst cases in the FRGC test set by global registration metric: largest angle, smallest scale ratio, largest scale ratio, largest mean translation.

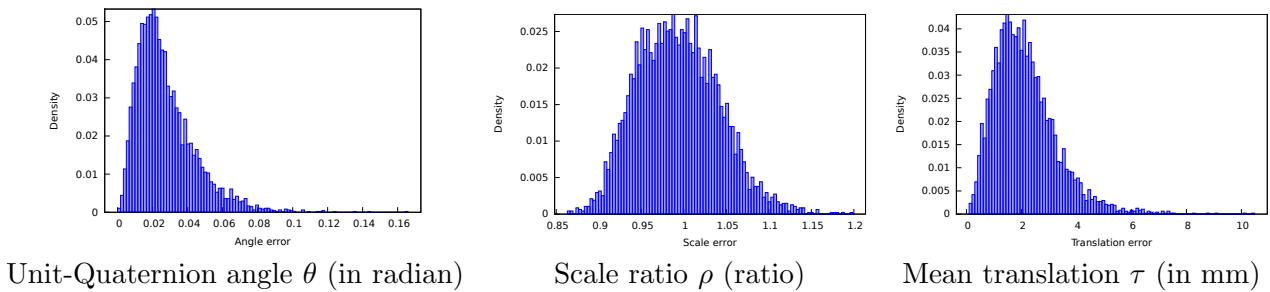


Figure 6.9: Distribution of global registration errors on the FRGC test set.

A limitation of our framework for landmarking is that the matching technique used is very naive and employs a rigid registration of the face to a common model. It is straightforward to build a more shape-adapted model using principal component analysis (PCA) and this may give us an improved landmarking system in terms of the precision of the localisations. However, it is likely that detection of the chin landmark with the mouth open will still be difficult (our current system always fails to detect the chin landmark when the mouth is open). Future work could look at new graph and hypergraph matching techniques to find a softer assignments between the keypoint and the target landmark labels. However, unlike existing techniques, they will need to be able to get stochastic assignments in cases presenting missing data. Figure 6.10 shows examples of discrete failure on the Bosphorus dataset, where a correct coarse registration is not found by the system. A discrete failure is declared if the rotation error  $\theta$  is above  $10^\circ$  ( $\sim 0.17 \text{ rad}$ ) or if the translation error  $\tau$  is above  $20 \text{ mm}$ .

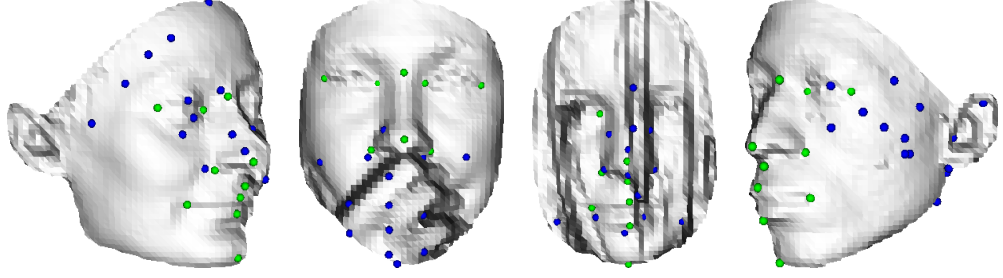


Figure 6.10: Examples of discrete failures on the Bosphorus dataset. The 17 failures detected on the 4339 scans (0.39%) are mainly due to occlusion (7 scans) and rotation (six ‘yaw’ and one ‘pitch’ scans). The three remaining failures are on scans that show exaggerated expressions.

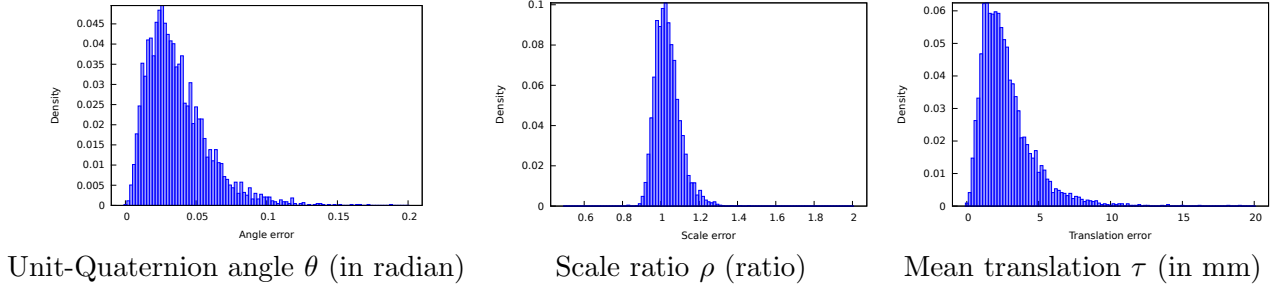


Figure 6.11: Distribution of global registration errors on the Bosphorus test set (all categories).

### 6.3.1 Computation Time Performance

The total computation time per query scan on the FRGC dataset is 1.18 seconds for meshes which have 3232 vertices on average. Most of the time is spent on the keypoint detection (0.97s). The most computationally expensive part of this is the histograms computation (0.70s). The neighbourhood computation costs 0.11s while the principal curvatures computation takes 0.06s on average. Big improvements in terms of computational speed have been achieved by modifying the curvature computation and using a Normal solver instead of SVD for the cubic surface fitting. However, some parts of the framework remain computationally expensive. The histogram computation for every vertex takes more time than all the rest put together. Indeed, the complexity of the naive algorithm used to produce the histograms is quadratic in the number of vertices. The speed can be improved by using better structures for locality retrieval, for example an octree and kd-tree structure. The computation of



DL-score and landmark score maps is performed in under 0.03s. The final matching using RANSAC takes 0.18s with 130 landmark candidates per face on average.

The total computation time per input scan on the Bosphorus dataset was 0.55 seconds for meshes that have 1879 vertices on average.

## 6.4 Conclusion

A simple method has been proposed to deal with the landmarking of learnt local shapes. A flexible aspect of our method is that it doesn't assume that the detected points should have an extremal value over a descriptor map. Instead, it assumes that the matching score of this descriptor against a learnt distribution should be maximal.

While other techniques are landmark-dependent, ours can be applied to any shape of interest as long as training is provided. The same method is used to detect the nose, an eye's corner or the chin. We detected 14 facial features, while expert system methods are usually limited to a few salient features (see Table 6.1).

While being more fuzzy (many-valued) compared to expert system methods, we believe that this kind of approach is necessary to deal with uncontrolled input data. In particular, it is more likely to be successful for non-cooperative face pre-processing where there are great uncertainties about what is present in the query scan.

In our opinion, the main gain in performance in the future will come from adding new local descriptors that deal with occlusions and profile views better. The RANSAC technique used in this paper works through registration, and therefore the labelling of the keypoints and the refinement of the positions occur in a single step. For the labelling process, using a less rigid and less global approach than the RANSAC method can help deal with cases in which most of the landmarks are not detected or spurious (e.g. profile view). Development of graph and hypergraph matching techniques robust to missing data could really help this aspect of the system. Local refinement of coarse landmark localisation will also be essential to gain precision and high retrieval rates at low acceptance radii.



## Chapter 7

# Conclusions

Did I say it before? I'm learning to see — yes, I'm making a start. I'm still not good at it. But I want to make the most of my time. For example, I've never actually wondered how many faces there are. There are a great many people, but there are even more faces because each person has several.

There are those who wear one face for years on end; naturally, it starts to wear, it gets dirty, it breaks at the folds, it becomes stretched like gloves that are kept for travelling. These are thrifty, simple people; they don't change their faces, and never for once would they have them cleaned. It's good enough, they maintain, and who can convince them otherwise?

Admittedly, since they have several faces, the question now arises: what do they do with the others?

---

Extract from *The Notebooks of Malte Laurids Brigge*

by Rainer Maria Rilke, 1910

Translated from the German by William Needham.

In this thesis, the problem of 3D landmarking has been split into different subproblems that have in turn been solved using sets of different techniques. We avoided the burden of computing all possible combinations of techniques within our framework by trying to understand the strengths and limitations of each individual method for particular tasks.

In this chapter, we summarise the high level conclusions that can be drawn from our work and explain, for each sub-problem, our major contributions as well as the new problems to solve that, we think, might lead to better 3D-surface understanding systems in the future.

## 7.1 A Global Picture in Chronological Order

In this section, we explain the chronological order in which we got the ideas and the results for our main development branch. This is a fairly high-level description of what we have done and in what order.

At the beginning of this project, we were mainly interested in 3D face recognition and shape analysis using geometric morphometrics, as used in anthropology. From the study of the literature of both these fields it has been highlighted that a lot of the systems still rely heavily on manual landmarking or show very fragile detection when automatised. From that point we focused on the landmarking of 3D surfaces. We concentrated on 3D faces because of the availability of data, but tried, during the whole project, to remain as general as possible.

Our first step towards improving landmarking was to analyse the existing automatic systems. They were mainly based on object-dependent and sequential recipes making lots of assumptions about what the input would be. Some, inspired by 2D machine learning techniques, were more generic, but lacked the pose-invariant capabilities that are expected from a 3D system. We therefore endeavoured to fill the gap in the research literature and produce 3D machine learning methods for landmarking.

The first idea, in order to be able to deal with occlusion and pose variation, was to detect more features than other existing techniques. A ridge-lines detector was first investigated. The idea behind this was to detect 1-dimensional curves on the face that are robust to pose variations before matching them to a model of the ridgeline of the face using graph matching techniques. Unfortunately, the two kinds of curve we investigated (ridgelines and isolines) were both quite sensitive to noise and holes in the mesh surfaces, making them very tricky to detect reliably. However, the curvature maps used to generate the ridgeline seemed repeatable enough across different individuals and poses. We therefore planned to improve landmarking systems by using points detected on curvature maps (which was not a new idea) but by selecting local maxima instead of global maxima, producing in turn more points on the face. The initial idea was to detect local maxima on different curvature-related maps, to cluster the points together to eliminate some of the false positives (a region was interesting only if different local descriptors produced a local maximum at its location), and finally to run a hypergraph matching technique to obtain the final correspondence. At that

stage, the idea of using hypergraphs instead of graphs was brought in because of the fact that the positions of the detected keypoints were not precise and that using more relational information of a wider range of type should help the disambiguation.

From that moment onward, we had a good idea of the problem separation: keypoint detection on one hand and point labelling on the other hand. A considerable amount of time was dedicated to the construction of tools for our framework to create a local descriptor map, visualise the data, detect features, create multi-attributed nodes and hyperedges over those features, and so on. We then started working on the problem that seemed the most complicated, the detection of a structural model within a query scene using graph and hypergraph matching techniques. For that problem, the set of keypoints is an input on which a graph is constructed. Each node and each hyperedge is associated with a set of descriptors. The query graph is then compared to a model graph for which distribution of the descriptor value over a training dataset is known. This leads to scores for every single combination of query-model assignation. Using a hypergraph-adapted relaxation by elimination scheme, we have been able to reduce the set of possible correspondences to a small number, at which point a unit-quaternion clustering technique using triangle registrations was used to obtain a one-to-one correspondence. We obtained encouraging results using this technique [Creusot et al., 2010], reassuring us that our approach was probably a good one. One of the aspects of the hypergraph matching technique is that the initial candidate list was generated using a seeding technique based on the distribution of learnt values for an underlying set of local shape descriptors from which a scores were computed. Because the number of candidates in the query was too large using the simple local maxima techniques we decided to focus back on the keypoint detection part of the framework and had the idea to use the seeding scores computed for the nodes of the hypergraph matching system for every single vertex in the input. By doing this using both scalar and histogram descriptors and using both linear and non-linear techniques for the construction of the final scores, we obtained very sparse sets of points for which the ambiguity was quite small. This was quite a surprise, as we didn't expect such good response maps. This method gave very good results in term of retrieval rates of hand-placed landmarks. The obtained keypoints were so sparse that the problem of additional false positives was no longer the main issue, highlighting that the main difficulties were linked to missing data. Therefore, we tried to

find ideas or existing work in hypergraph matching that can cope with missing data, but this was not conclusive. Thus, we headed back to registration techniques that are able to deal with missing data and constructed, for the first time during the project and close to the end of the PhD, a completely automatic landmarking system pipelining the output of our keypoint detector into our labelling system. This technique shows better results than the state-of-the-art techniques in landmarking on the two biggest publicly available 3D face datasets.

## 7.2 Summary of Contributions

The way we approach the problem of landmarking has been very pragmatic: we tried to uncover in other people's work the assumptions that lead them to fail in difficult cases and then we built a framework from scratch that doesn't make these assumptions.

In this section, we summarise the contributions made in the thesis. Our single most important contribution in this thesis is the introduction, for the first time, of a machine-learning system for 3D face landmarking. All previously published machine-learning methods were using 2D projections at some point, making them sensitive to pose variation. Our approach outperforms heuristic methods commonly used for landmarking on 3D surfaces while also being more generic.

### 7.2.1 Keypoint Detection Using a Dictionary of Local Shapes

By replacing existing expert systems for 3D face feature candidate localisation by machine learning techniques, we have enabled the rules used for detection to be learnt instead of enforced by the designer of the system. This has allowed us to try to detect more points, including ones that are less salient and for which humans struggle to create manual rules. The fact that the same system is used for every landmark makes each detection concurrent, while existing systems are sequential (e.g. requiring the nose to be detected before any other landmarks). We have relaxed some of the assumptions about the input data to enable non-cooperative face recognition. For example, we make no assumptions about the face being frontal, upward, or free from non-face objects (hair, hand). By using maxima on a score map instead of descriptor map, we allowed both scalars and histograms to be used in a

similar fashion toward the local detection of keypoints. This also helped to move away from the assumption that the nose or eye corners correspond to the most extremely curved point in the input scene. The increase in the number of landmarks to be searched helped minimise the risk of failure where not enough points have been detected for a registration to occur, for example when occlusions and spurious data are present, which is commonplace in non-cooperative captures.

Our method of keypoint detection has been evaluated on the FRGC database, on which performance has been measured using lots of different configurations involving up to 40 local shape descriptors (10 types of local descriptors for 4 different neighbouring sizes). We also extended the method to use both linear and non-linear score merging techniques (LDA and Adaboost). Unfortunately, the keypoint detection technique is not comparable in itself to anything published in the literature, because the points are unlabelled at the stage at which the detection rate is measured. We hope that researchers will publish this kind of intermediate results in the future, as they are essential to evaluate separately the local detection capabilities of the system from the structural labelling and eventually the global localisation of the landmarks.

### 7.2.2 Keypoint Labelling Using Learnt Structural Models

One originality of our labelling is its ability to use multi-attributed sparse hypergraph structures for data representation. While higher-degree structural matching papers have been published by other researchers during this PhD [Zass and Shashua, 2008] [Duchenne et al., 2009] [Chertok and Keller, 2010], using hypergraphs in a non-tensor form with multiple attributes per element is novel. Each element of our hypergraph (nodes and hyperedges) can be attached to a great number of properties. Matching the constructed graph against a model graph produced scores that are element dependent while most existing techniques use a fixed scoring scheme. Another original aspect of our matcher is that the hypergraph can be considered in its dual form without any computation. The application of hypergraph matching techniques to 3D object (in particular 3D face) correspondences is also new. All hypergraph matching techniques that we have encountered are only applied to 2D image matching. The use of higher degrees of connectivity for 3D can help to better represent the underlying data, by using surfaces, angles, volumes and so on. Splitting of the matching process (usually con-

sidered as one step) into a succession of correspondence filters using different correspondence finding techniques also helped us design a simpler matching framework. Most researchers publishing graph matching techniques present them as stand-alone processes while they are, in fact, pipelines of correspondence filters, some of which are continuous and some discrete.

At a practical level, we designed a new hypergraph matching relaxation algorithm alternating elimination on the hypergraph and its dual to reduce the set of possible correspondences when large amounts of extra nodes are present in the scene hypergraph. In a proof-of-concept experiment we show that facial features can be retrieved even when a large number of false positive keypoints are present, supporting the hypothesis that a landmarking system in two steps (keypoint detection and labelling) was achievable. We also developed a simple scale-adapted rigid correspondence finder based on unit-quaternion clustering of triangle transformations. This was our first technique to obtain one-to-one correspondences from a set of sparse landmark candidates. A second method for stochastic scale-adapted normals-aware rigid correspondence was based on the RANSAC meta-algorithm adapted for correspondence candidates. This technique was used in our final landmarking experiments.

### 7.2.3 Final Landmarking System

Our final landmarking system is an automatic method that takes, as input, a mesh and a learnt model set of landmarks and returns the positions and labels of the landmarks on the input mesh. This is the result of the pipelining of methods presented and evaluated in chapters 4 and 5. The evaluations of this technique on the two biggest publicly available 3D face datasets have shown that our approach works, even in difficult cases where important features are missing, when occlusions are present and where pose varies by up to  $45^\circ$ . Our approach, while taking a different path from state-of-the-art techniques, is not only gaining ground on them, but achieves better results for all metrics used (except for an acceptance radius of 10 mm). However, the improvement in terms of continuous values (e.g. localisation accuracy) is, in our opinion, less important than the gain in terms of discrete relaxation of hypotheses and improvement of the genericity. Table 7.1 shows some of the differences in hypotheses and capabilities of our system *versus* traditionally used expert system approaches.

Table 7.1: Comparisons between expert system and our machine learning approach.

Characteristic	Expert Systems Face Landmarkers	<b>This Work</b>
Object type	3D face only	Non-articulated objects
Landmarks number	Fix (often <5)	Arbitrary (tested up to 14)
Individual detection	Landmark dependent	Landmark-independent
Processing order	Sequential	Concurrent
Detections	Map extrema	Score map extrema
Landmark-Map correlations	Manually provided by researcher	Learnt automatically
Pre-processing needed	Yes	No
Local descriptors type	Scalar only	Scalar and histogram
Descriptors number	Fixed (<2)	Arbitrary (tested up to 40)
Descriptors combination	Manually fixed (linear)	Learnt (linear or non-linear)

### 7.3 Limitations and Future Research

While our work has by-passed some of the limitations of existing landmarking techniques, it is not as good and reliable as a human operator. In this section we summarise the main limitations of our work and propose ways in which these problems can be answered in the future.

#### 7.3.1 General Points

The errors made with our current system (see Section 6.3) are roughly of two types: failures and imprecisions, while the main cost is computation time. Here we discuss the limitations and possible improvements in terms of precision, robustness to failure and computation time.

**Localisation Precision** We believe precision can be dealt with by local refinement. If a landmark is correctly localised within a small radius from its ideal position, using local optimisation techniques can help retrieve the optimal, possibly 'sub-pixel', localisation. It has to be noted that the optimal resolution of the input data required to do precise localisation is probably quite different from that used for coarse localisation. Furthermore, this kind of local optimisation cannot be tested against human performance. Different human operators will landmark objects slightly differently and humans are not very precise at this kind of measurement. Developing ways to assess the quality of precise landmark localisations is an interesting problem. Intuition dictates that detected landmarks should be repeatable from

one shape to another but, even then, two questions can be asked: Firstly, how is the dense correspondence between the two objects computed in the first place? Secondly, even if the dense correspondence is known, do the landmarks move with the skin and flesh or are the landmarks attached to the underlying bone shape? These are easy questions for functional landmarks like the corner of the mouth or the corner of the lids but are very complicated for other types of landmarks like the tip of the nose, cheekbones, nose corners and so on. Finding solutions to these problems on faces is extremely difficult. Looking at less deformable objects (like large collection of bones with correspondences) would probably be useful, if the data were available.

**Robustness to Failures** The problem of detecting failure and getting a robust coarse registration all the time is both a local and structural problem for which optimal solution discovery requires, in the worst case, exponential computation. From the set of errors we detected with the Bosphorus database, the reasons for failure are usually linked to a combination of missing landmark candidates (due to occlusions or pose variations) and spurious local descriptor values (for example due to the fact that the neighbourhoods are not complete or clean near occlusions or non-face objects). We think discrete failure is the most important problem to solve. We think that the main progress in this area will come from using new local descriptors that can cope with missing data. One type of information that is known by the system but not used when computing the local descriptor is the holes border. When dealing with a nose viewed in profile the vertices that are next to the hole created by the self occlusion are known. Nonetheless the computation of the local descriptor is the same as everywhere else: a spherical neighbourhood is computed on which the function is applied. In case of symmetrical shapes with surface fitting, like the curvature computation, the values are not changed much. Other descriptors like the volume, the distance to local plane (DLP) or the histogram descriptor change more dramatically. In cases where the local shape is not symmetrical, all descriptors suffer from the missing data. One idea to change this in the future is to develop descriptors that are occlusion aware, such that the keypoints detected on the border of the mesh are more meaningful. This can be done using, for example, descriptors on curves detected in the overall direction of the occlusion. These are likely to be less descriptive than local area descriptors, but hopefully more robust near the mesh



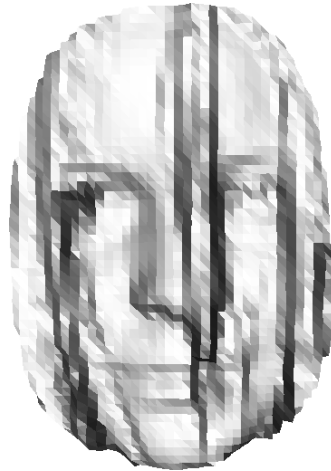


Figure 7.1: A difficult case from the Bosphorus dataset (bs101\_O\_HAIR\_0) for which, we believe, our method will always fail, but for which human operators can easily localise the face and its features.

border. This approach will not help in cases where occlusion is produced by a third-party object. For example, when the hair or hands are in front of the face, the system is not aware that there is occlusion. To some extent, this can be solved using 2D techniques of skin segmentation to discard hair, a cell phone, scarf, and so on. However 2D techniques cannot help in cases where a hand is in front of the mouth or the eyes. Another incentive not to use 2D segmentation techniques is the fact that these techniques don't work as well with darker skin. At a conceptual level, we know the problem is solvable with only 3D data, as a human operator can detect occlusion of the face very easily. Figure 7.1 shows one of the most difficult 3D faces on which to detect features, as almost every single landmark considered will have spurious local descriptors. Yet, humans seem to be able to segment the face from the occluding hair. Learning what process is used by humans to detect the feature on this picture can probably help improve our methods in the future.

**Computation Time** While the time of computation shown for our experiment is comparable to other existing techniques (around 1 second, see Table 6.1), it is still too expensive for real life application and far slower than 2D techniques (for example the one used in the OpenCV API based on Haar filters). As shown in section 6.3.1, around 60% of the computation time is used to compute histogram descriptors. While this can easily be improved by using better data structures to retrieve the local neighbourhoods, we think that complex

histogram descriptors should not be used in the keypoint detection process, but maybe only in the labelling process. The process can be easily sped up by only using simple and fast local descriptors. However the fewer descriptors, the more errors can appear.

The neighbourhood computation for scalar descriptors can be improved by also using an octree or a kd-tree. Optimising the computation of single descriptors can also significantly improve the performance. In chapter 4, we showed that using different solvers for the surface fitting of the curvature computation changes the performances dramatically (the time of computation has been divided by 8 for a neighbourhood of 15 mm) while having very little effect on the quality of the resulting descriptor map.

Once these changes have been implemented (mainly the kd-tree structure for the mesh vertices), future work should look at the differences in performance and time of computation when various combinations of descriptors are used.

### 7.3.2 Keypoint Detector

It has to be noted that in all training sets used in our experiments, only neutral faces are represented. We tested our systems on faces showing expressions, pose variation and occlusions, but they were not used in the training. The reason for this is that the local shape distribution is represented in our framework by a unimodal function (e.g. Gaussian). Two approaches can be used to deal with the problem of learning a wider range of face classes. First, a multimodal function can be used to represent the local descriptor values, such as a mixture of Gaussians, for example. The problems with this are: first, to know how many modes should be considered for a measured distribution; second, to fit the multivariate function to the observed data; and thirdly, to design a scoring function that makes sense for such a distribution. Indeed, taking the probability density function (*pdf*) divided by the maximum value of the *pdf* is no longer meaningful. Dividing by the closest local maximum might give good results but introduces a discontinuity for points in the middle of two local maxima. Interesting future work would look at real distributions of landmark local descriptor values for training based on different expressions and see how those values change and how they can be represented within the framework.

Another approach is to consider different landmark labels for different expressions, view poses, and so on. For example, the local shape of a cheekbone landmark will be learnt

independently for neutral and smiling faces. The problem with this approach is that it introduces a discrete separation that becomes difficult to justify when dealing with dynamic faces. If a neutral face starts to smile, at what point in time does the nature of the cheekbone landmark change from neutral to smiling? Mapping a continuous phenomenon to a discrete representation in this case introduces some conceptual issues.

Nonetheless, while this discrete approach can be criticised when dealing with expressions, it seems almost unavoidable when dealing with different identities. In this thesis we make the assumption that the set of landmarks in the model are universal. This assumption is almost certainly invalid to a certain extent. Local shapes that are optimally detectable for some individuals might not appear at all on other individuals. For example the ophrion<sup>1</sup> is not easily detectable on every forehead, one person might have a single peak nose tip while another has a two-peak nose tip, some people have squared chins while others have round or triangular chins. There is very little chance that a single face model will be optimal for everybody. Important future work would be to design a framework capable of extracting the best landmark to use for a part of the face and a subset of the database. At the end of the PhD, a prototype system was developed that is able extract “optimal” landmark models in an unsupervised settings. This system, which uses the same set of descriptors as used in Chapter 4, is outlined in Appendix C.

### 7.3.3 Labelling

The labelling part of this thesis focuses mainly on hypergraph matching techniques that, for now, have trouble dealing efficiently with missing data. The problem is that most graph matching techniques use sets of iterative local operations to deal with an otherwise intractable problem (very often the problem is exponential in complexity). The notion of missing data, on the other hand, is intrinsically global if the connectivity of the graph is global, which is often the case with faces, as distances between most pairs of points can be bounded by values with low deviations. Future work on graph matching should look at dealing with missing data. A mixture of rigid registrations and soft graph matching techniques might also be

---

<sup>1</sup>Points situated at the centre of the forehead just above the superciliary arches and under the frontal eminences.

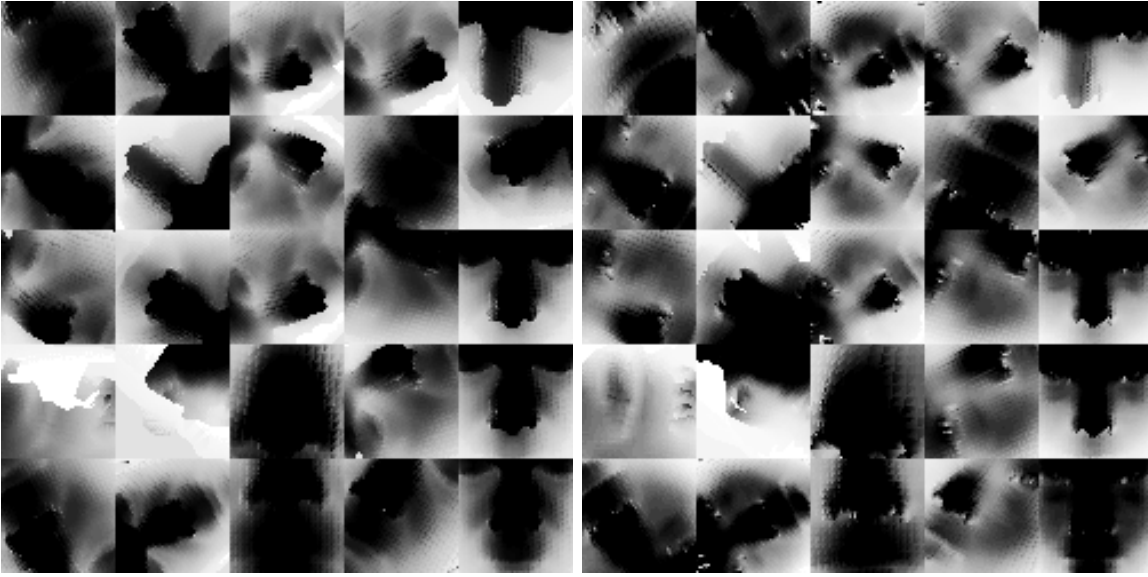


Figure 7.2: Multiview 2D depth maps. Two sets of 2D depth maps generated from 2 different individuals from the FRGC database using a set of hyperedges of degree 3 as the basis for the camera positioning.

interesting to consider for 3D objects on which rigid sub-parts can be detected (e.g. upper face, jaw).

Another interesting point is the use of hypergraphs for data representation. While designing hypergraph-based techniques is a bit more complicated than graph techniques the potential is great. Indeed, almost any form of information can be represented with a hypergraph. For example planar 2D information about the face can be attached to planes constructed from 3 nodes (hyperedges of degree 3), and this can be used to perform face recognition using 2D techniques in a pose invariant way (see Fig.7.2). Another example is to use hyperedges of degree 3 to constrain the angle at which a finger can move from the palm, making an hypergraph model of the hand much more constrained to realistic movements than a graph model. The great advantage of the hypergraph representation is that, by being versatile in terms of the nature of structural information it contains, it allows the same system to be used for many different objects and computer vision tasks. We think that hypergraph representation could play a major role in the unification of very specialised existing computer vision tasks.

### 7.3.4 Limitations at a Fundamental Level

It should be noted that the kind of geometric approaches presented in this thesis can never be used to see non-realistic faces. It seems that humans have an ability to see faces that is not based on a direct form of object-specific learning but on a high-level meta-learning. A human is able to facialise almost anything, from the faces of animals, to objects anthropomorphised in animation movies or even cars. This ability to facialise without the requirement for learning on each particular class of objects is very interesting. However, we can ask ourselves whether this bias toward faces is a good thing to reproduce in machines. Do we have this ability because it is required (the theory that only specialised systems can see faces) or is it just an artifact of our evolution, implying that a generic system could probably see faces in any of the cases that a human see faces without using a face-specific process? These questions are very unlikely to find an answer unless a better understanding of animal and human vision is achieved. It seems very unlikely that animal vision can be reproduced with existing approaches. Until a change of paradigm has occurred (e.g. evolutionary computer vision), we will have to find solutions for vision problems one at a time.

## 7.4 Final Conclusion

We have presented a 3D surface landmarking framework that alleviates a lot of the limitations of existing systems, while making as few assumptions as possible in order to solve the problem in an efficient and fast way. Our efforts have been focused on 3D faces that are, at the same time, very challenging and very useful for real life applications. At each stage of the PhD, a great deal of attention has been focused on finding a good trade-off between robustness and speed. Our framework, in its final version, is capable of dealing with lots of variations in the input data and produces better results than existing full 3D landmarking techniques.

Our system is able to interpret an input mesh as being a face, composed of different features in a given layout. Future work should look at improving the robustness to missing data, widening the learning phase of the process and designing faster and more descriptive local descriptors.

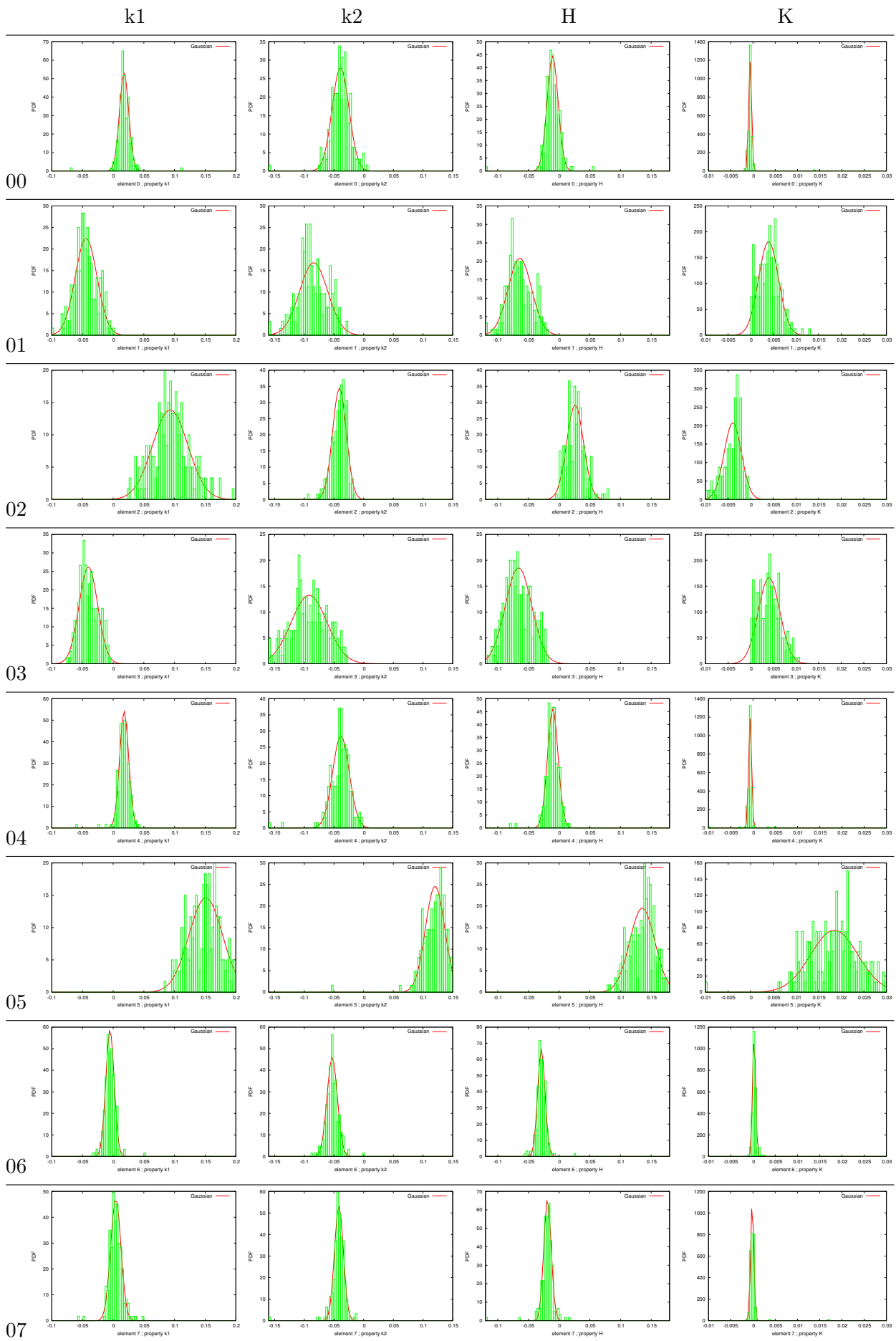


## Appendix A

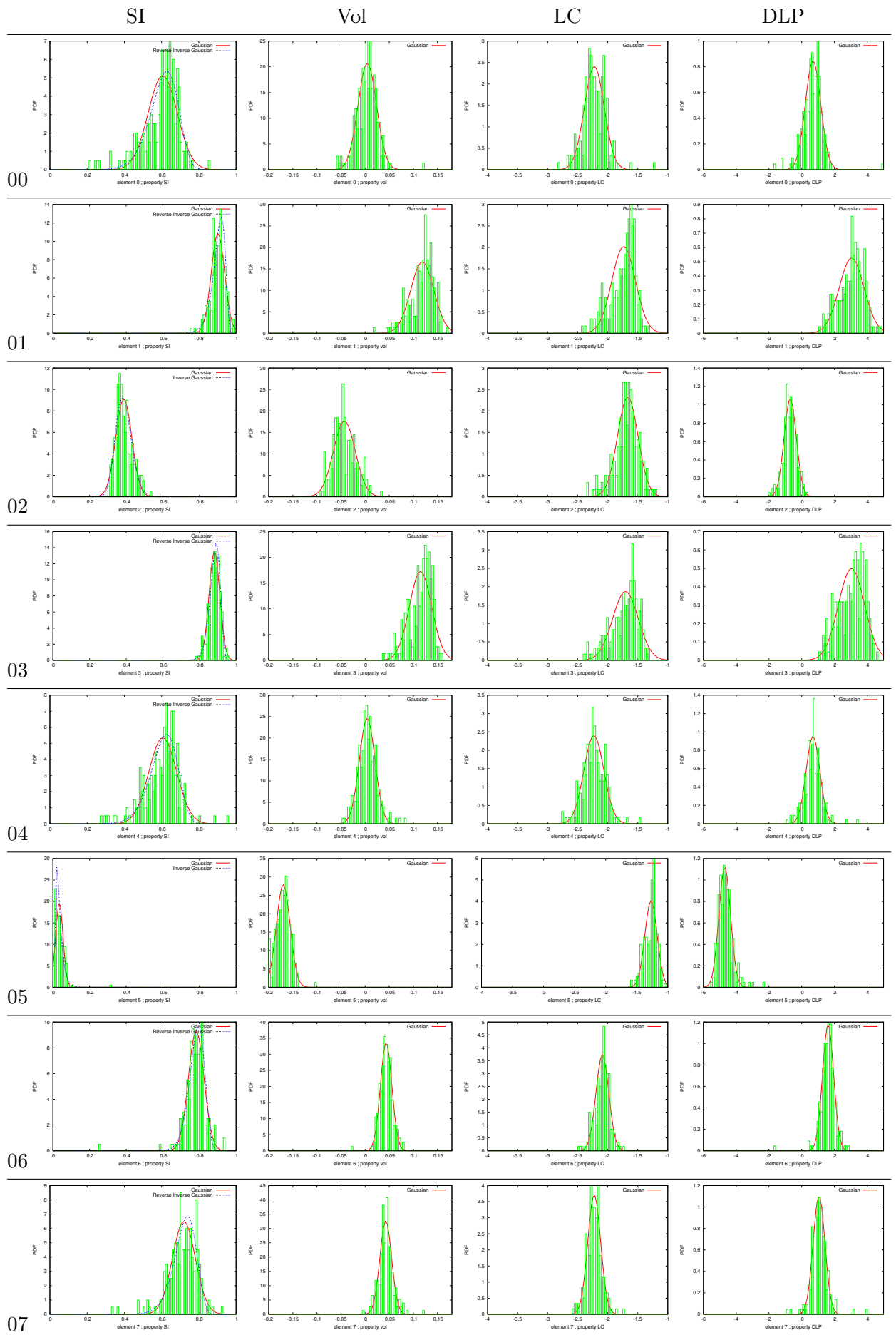
# Local Shape Descriptor Distributions

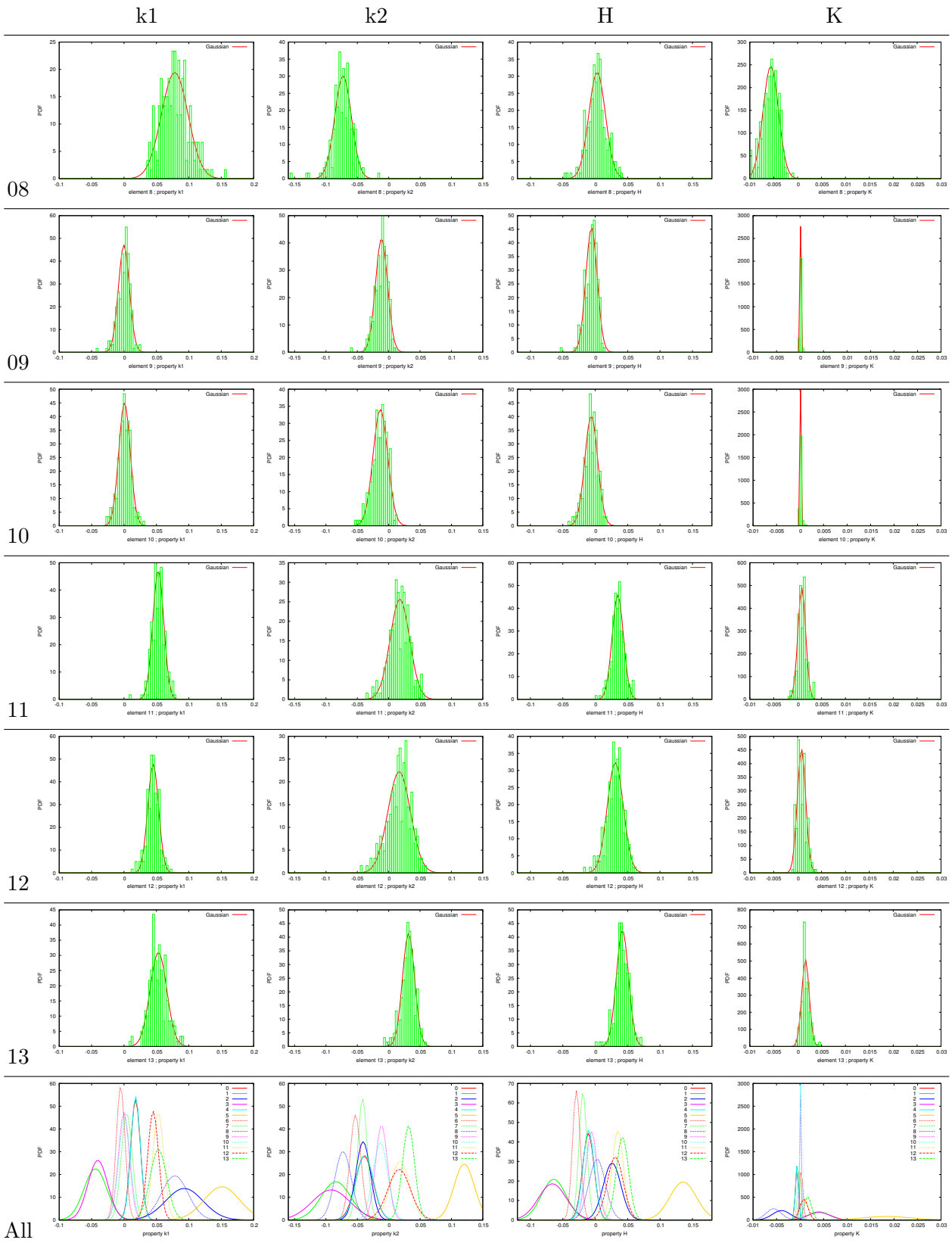
In Chapter 4, we presented our keypoint detection system that relies on computing a score for each vertex, based on learnt distributions of local shape descriptors for known landmarks over a training dataset. In the following, we show the observed distribution of 8 scalar local shape descriptors (columns) over the training set of the FRGC, for each of the 14 hand-placed landmarks (rows) used in this thesis (see Figure 4.14). For all the descriptors, the probability density function is approximated by a Gaussian curve, defined using the measured mean and deviation of the real distribution. For the Shape Index descriptor (SI) (0-1 range), an Inverse-Gaussian distribution is used (see Section 4.3.1). If the mean is lower than 0.5, the Inverse-Gaussian started at 0 with a positive direction, otherwise, it starts at 1 with a negative direction (here called Reverse Inverse-Gaussian).

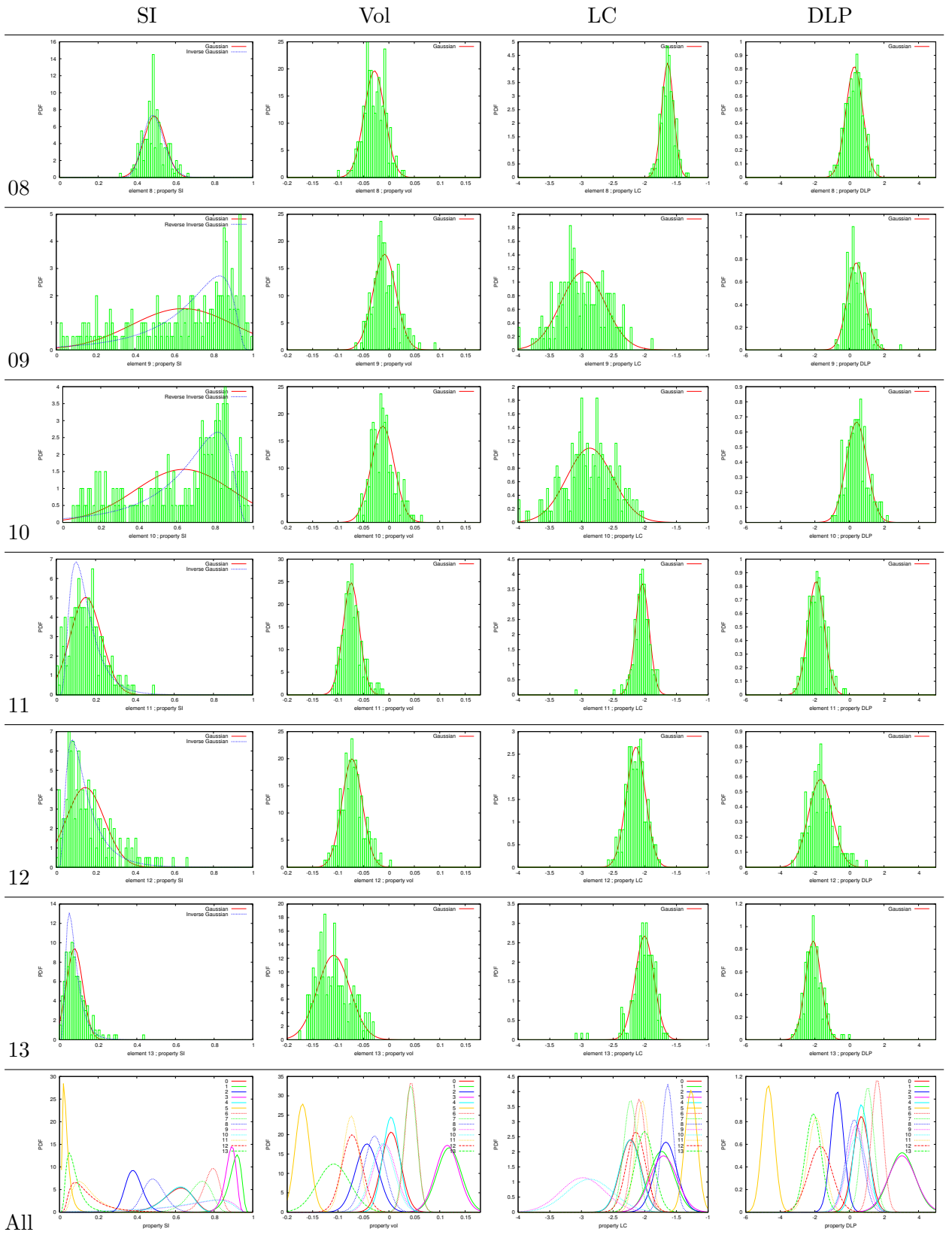
All of the distributions for a given descriptor are plotted within the same abscissa interval so that comparison can be made between landmarks. The last row of graphs shows the approximated probability density functions superimposed for the 14 landmarks.













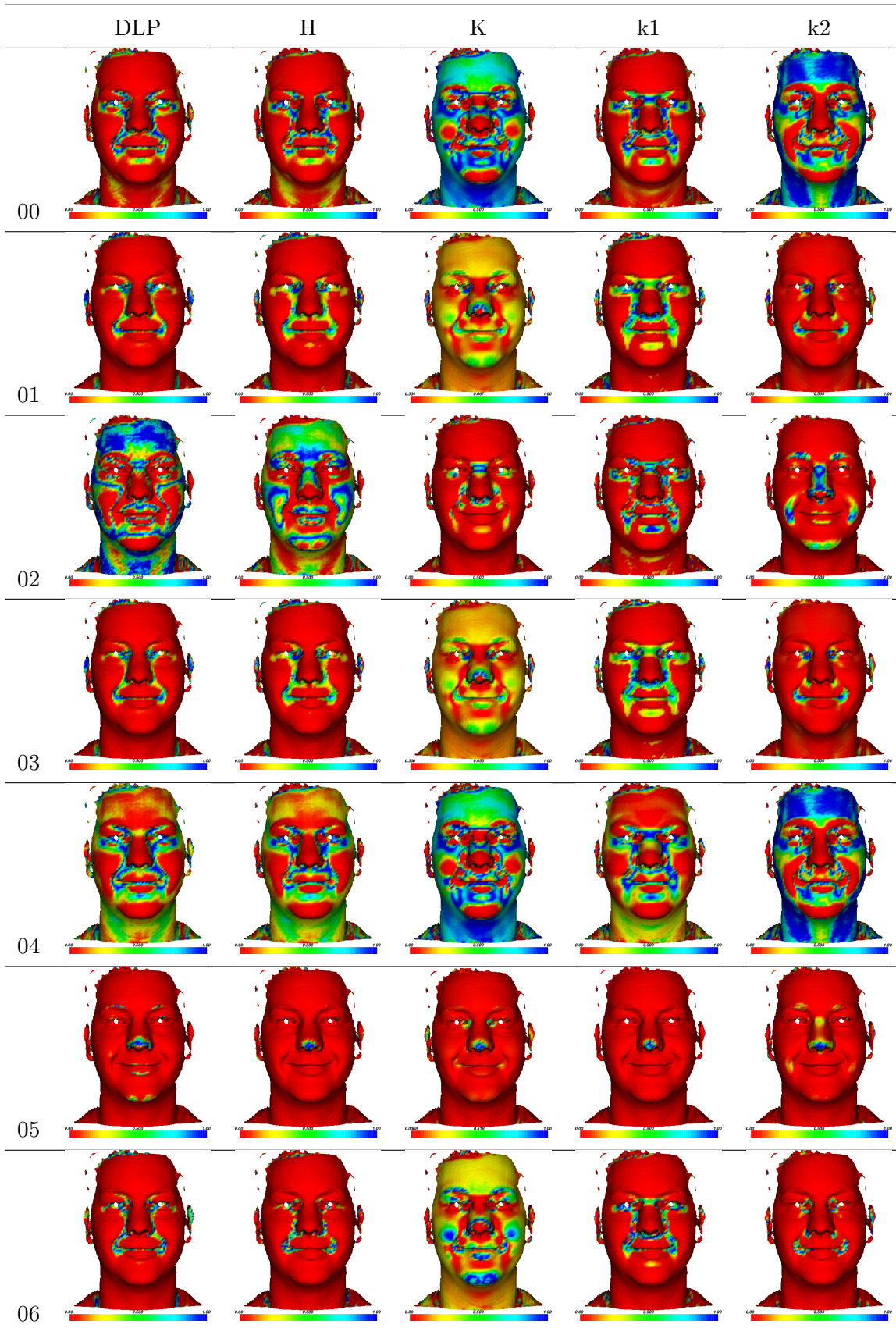
## Appendix B

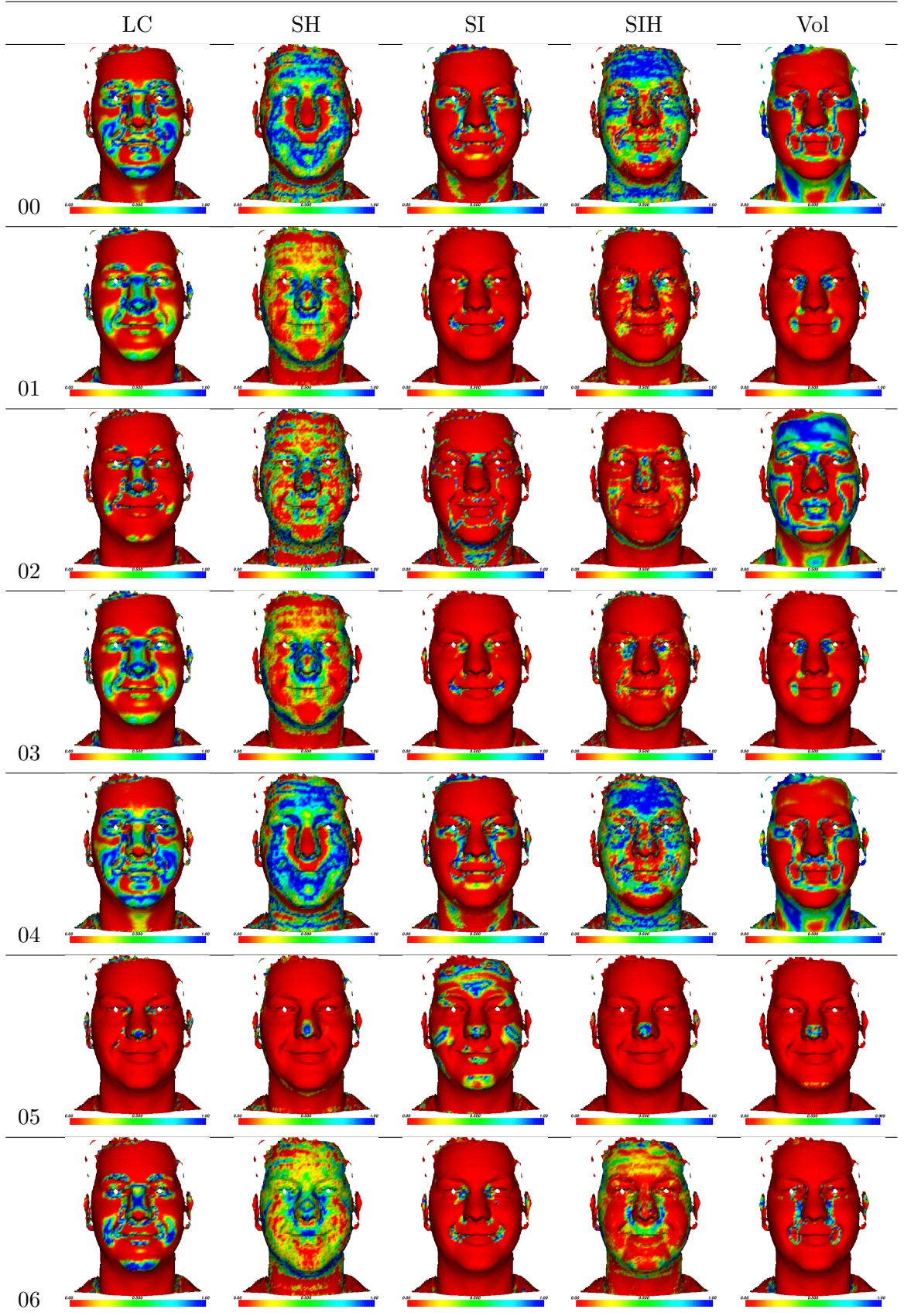
# Descriptor-Landmark Score Maps

To detect keypoints, the system described in Chapter 4 computes Descriptor-Landmark score maps. For each vertex, the value is a scalar ranging from 0 to 1. A value of 1 (blue vertex) means that the local shape for the given descriptor is very similar to the one learnt for the shape of interest being tested.

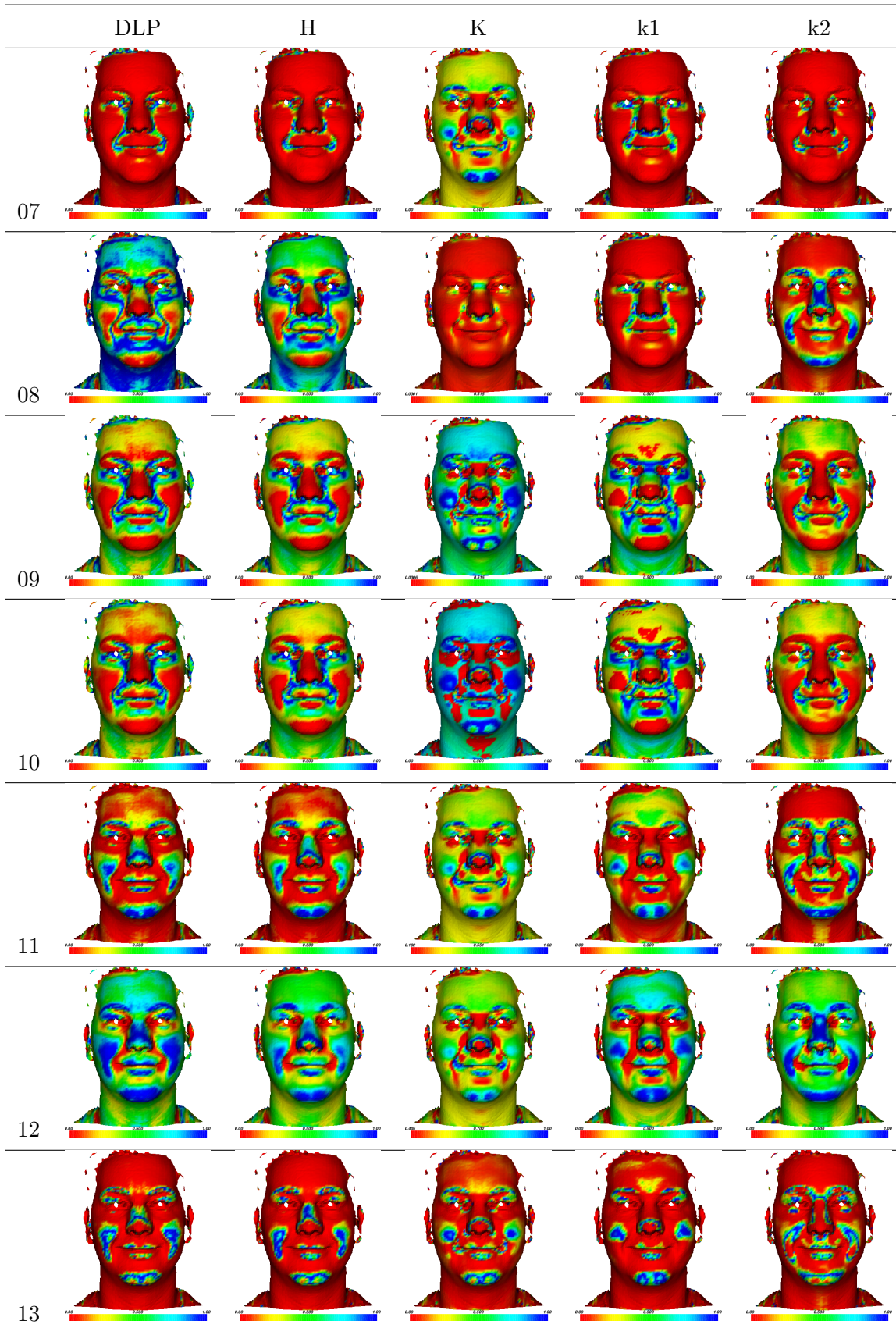
In the following, we show, for one face of the FRGC dataset, the descriptor-landmark score maps of the 14 landmarks used in the thesis for the 10 local shape descriptors used in the configuration 2 of Chapter 4.

On the last page, Table B.1 shows the landmark score maps, where the descriptor-landmark score maps have been combined to a single map using the LDA merging method (see Section 4.6). The last column of the last row shows the Keypoint score map, where each vertex is associated with the maximum value over the 14 Landmark score maps.











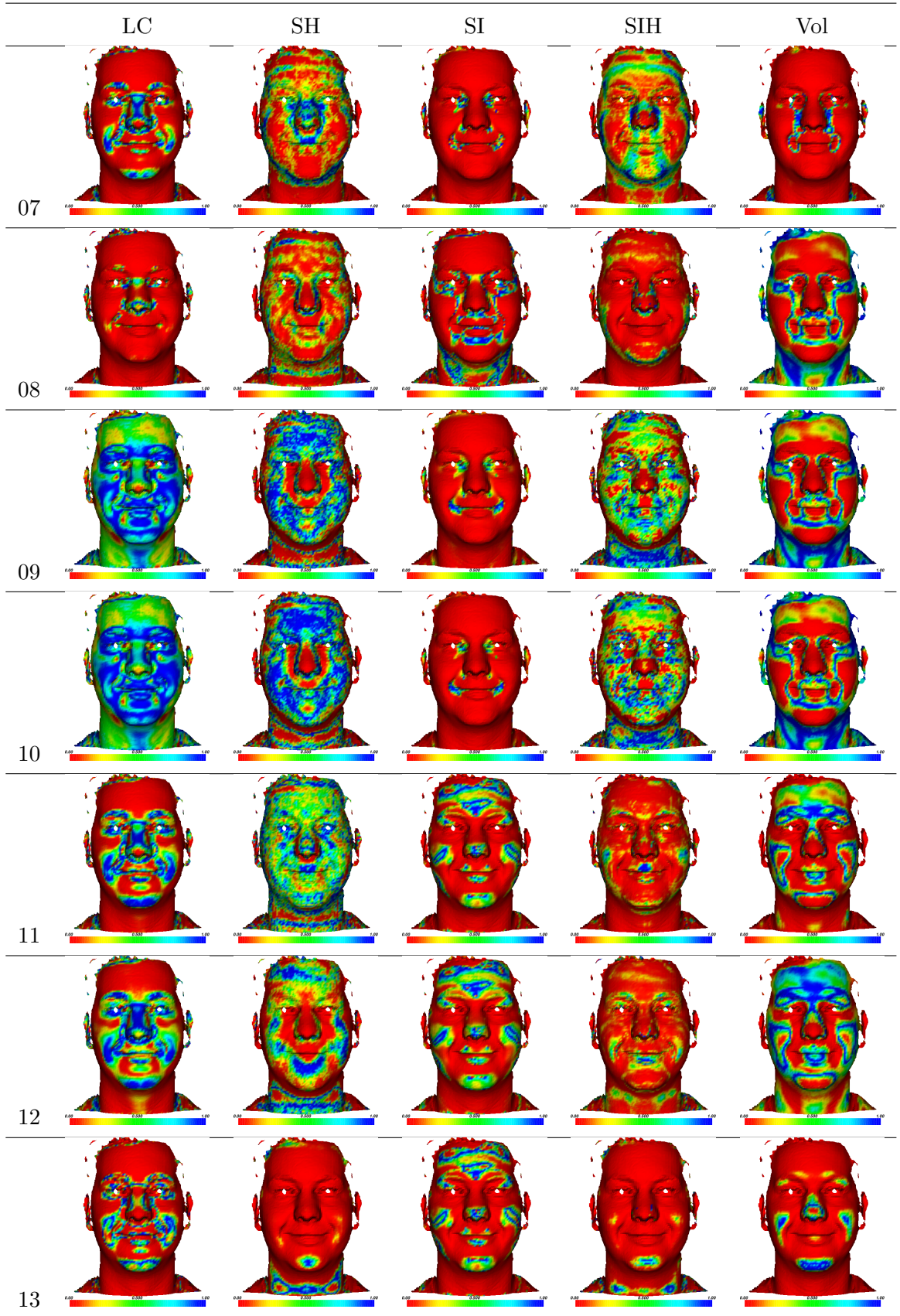
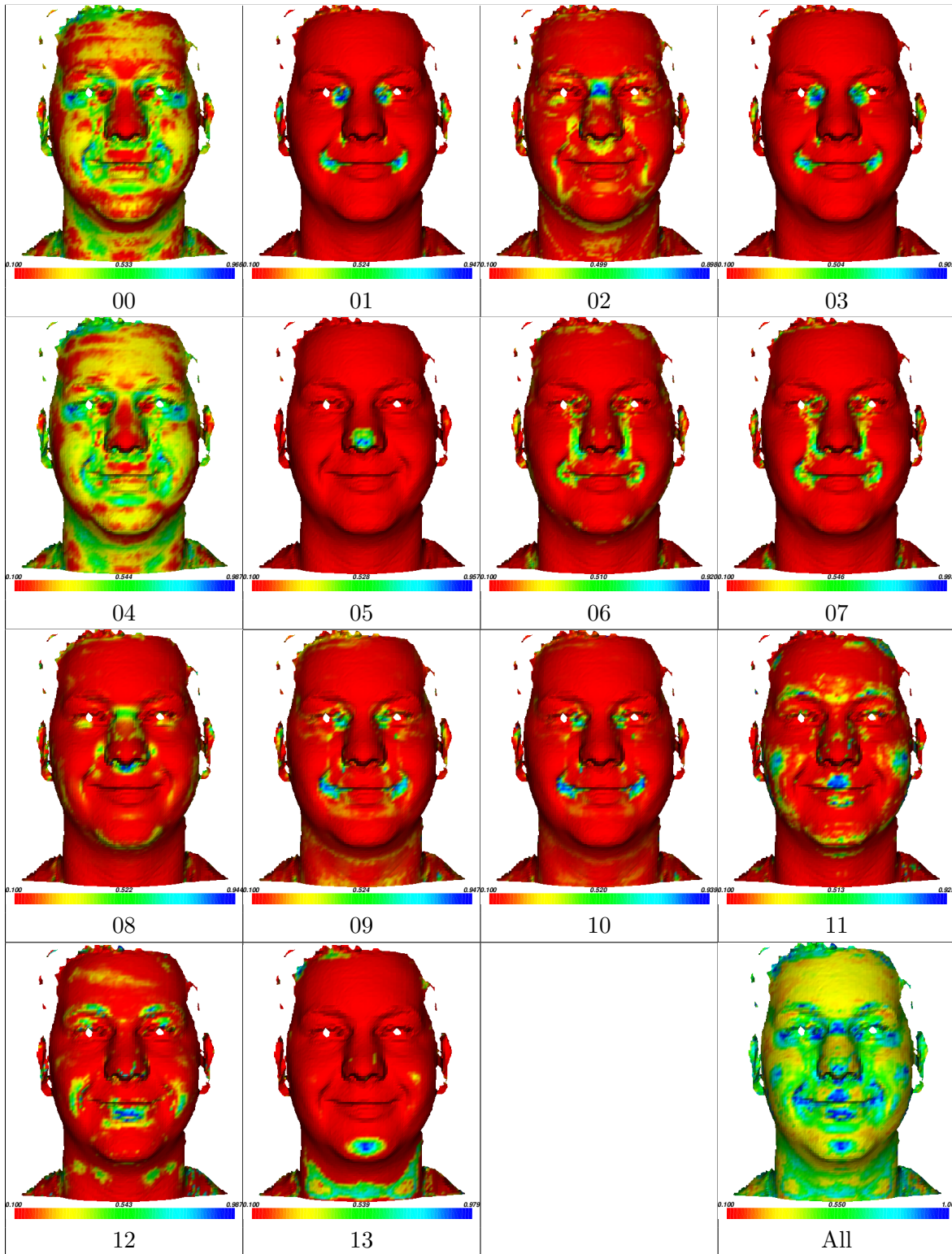


Table B.1: Example of Landmark score maps for one face of the FRGC dataset.



## Appendix C

# Unsupervised Learning of a 3D Face Landmark Model

“And the wheel,” said the Captain, “What about this wheel thingy? It sounds a terribly interesting project.”

“Ah,” said the marketing girl, “Well, we’re having a little difficulty there.”

“Difficulty?” exclaimed Ford. “Difficulty? What do you mean, difficulty? It’s the single simplest machine in the entire Universe!”

The marketing girl soured him with a look.

“Alright, Mr. Wiseguy,” she said, “if you’re so clever, you tell us what colour it should be.”

---

Extract from *The Restaurant at the End of the Universe* by Douglas Adams,  
Hitchhiker’s Guide to the Galaxy serie, 1980.

Here, we present a system for discovering saliency on 3D meshes, which allows the automatic, unsupervised generation of a model set of landmarks. This can then be used as the key component of a landmark-localisation system for the set of meshes belonging to some object class. In our experiments, we show that an unsupervised machine-learning framework can extract a symbolic representation of a class of objects, if is provided with enough registered data samples. Unlike the models designed by humans, the ones extracted by our framework can be optimised for a given set of local shape descriptors and locality definitions. While a lot of similarities can exist between the manual and automatic models, the automatic model has some intrinsic advantages; for example, the fact that repetitive shapes are automatically

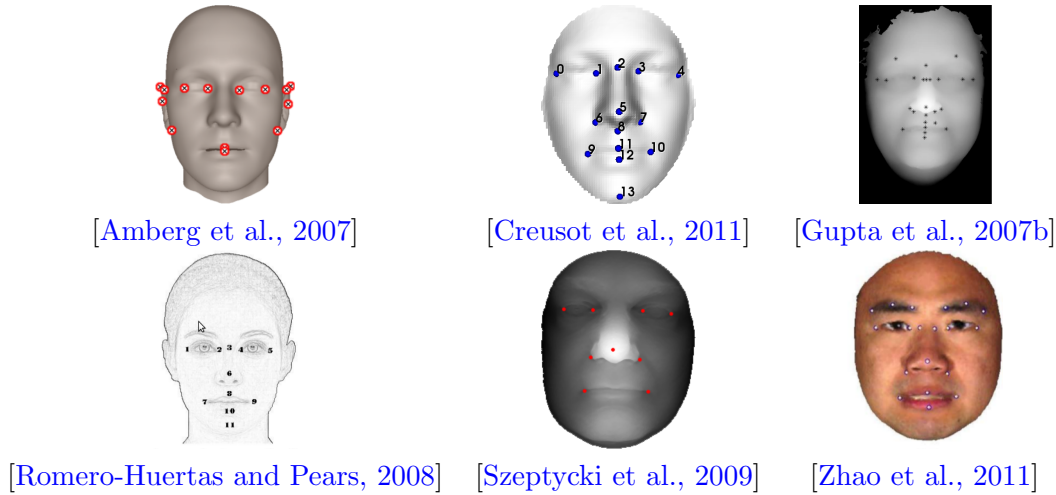


Figure C.1: Examples of sets of landmarks used in the literature as generic and sparse symbolic representations of 3D faces. In most cases, the points have been chosen because they can be explained through language to a human operator, not necessarily because of their optimality for a given purpose. In this chapter, we aim to find a less arbitrary generic model of 3D faces optimised for face detection and local feature localization.

detected and that the local shapes are ordered by their degree of saliency in a quantitative way. This small experiment is a first step in answering an important question: Can landmark models, learnt by machines, allow better landmark localisation on big datasets, such as the FRGC v2 3D face dataset? In Fig.C.1, it can be seen that there is no consensus on what landmarks to use for face analysis. A good detection system should focus on detecting the most salient landmark for the machine, instead of trying to find the most easily landmarkable by a human operator.

In the next section, we present our framework and the datasets used for the experiments. The following section presents results and comparison between automatic and manually defined models. The last section is used for conclusion and discussion of future work.

## C.1 Automatic Saliency Discovery

Our framework aims to learn what local shapes can easily be retrieved from a 3D scan of a given object using a predefined set of descriptors and a definition of locality for the feature detection.

### C.1.1 Local Shape Descriptors

We do not wish to rigidly prescribe a descriptor or combination of descriptors as those will often limit the degree of saliency that can be achieved, according to some saliency definition. Instead, we define a ‘bag of descriptors’ as an input to the system and our framework learns some linear or non-linear descriptor combination that maximises some saliency score.

In the experiments presented here, 8 local shape descriptors are used, as follows: the first and second principal curvature ( $k_1$  and  $k_2$ ), the Gaussian curvature ( $K$ ), the mean curvature ( $K$ ), the Shape Index (SI), the log-curvedness (LC), the local volume (Vol) and the Distance to Local Plane (DLP).

### C.1.2 Likelihood of Being a Landmark Given a Local Descriptor

We assume that we have a dense correspondence between 3D face scans in a training dataset. This can be achieved in an approximate way using ICP on the FRGC data, or more accurately using training generated from the Bassel Face Model (BFM).

Let  $i$  be a vertex index known across all of the training dataset and  $d$  be one local shape descriptor. The probability density function (pdf) of  $d$  across all the training set for this local shape centred on  $i$  can be learnt and approximated by a known function. For any new position  $j$  on a mesh  $k$ , a score can be computed:

$$S_i^d(j_k) = \frac{\text{pdf}_i^d(d(j_k))}{\max_x(\text{pdf}_i^d(x))} \quad (\text{C.1})$$

which, in the case where the pdf is a Gaussian, is equal to:

$$S_i^d(j_k) = \exp\left(-\frac{(d(j_k) - \mu_i)^2}{2\sigma_i^2}\right) \quad (\text{C.2})$$

where  $\mu_i$  is the mean value of the descriptor at vertex  $i$  and  $\sigma_i$  is its standard deviation.

Nothing prevents the neighbours of this point from having a similar shape and therefore a score as high as  $i$  itself. What is needed is a way to detect whether only the close neighbours of  $i$  are the points that look like  $i$ . We now define two metrics which we can employ to extract model landmarks.

### C.1.3 The Saliency Metric

Water covers around 71% of the earth surface. Therefore, if we compare a local patch of water to the entire world we will find that this patch is not very salient. Now if we compare this patch to a smaller neighbourhood, the conclusion can change significantly. Imagine that this patch of water is in the middle of the desert, the difference between the patch and its surrounding is now extremely large and the patch is considered very salient. This example highlights the problem of locality. For detection and localization purposes, the more globally salient elements might not necessarily be interesting or be part of the object that we want to extract from the input. For example, the tip of the nose is the most curved convex region on an average face, but it might be far less curved than other non-face elements (like hair locks) than can appear in the input. The objective is to determine locally discriminative features. *Saliency* is a generic word that usually refers to the level of differences between something and its neighbourhood, but it doesn't have a unique definition in the literature.

Here, we introduce a definition of saliency that can be learnt from a training set of registered surfaces. It relies on the definition of two classes of neighbouring vertex for a given input point:

- The local matching class: the set of neighbouring points that are thought to be similar to the input point.
- The remote non-matching class: the set of points surrounding the matching points that are used as a reference to which the matching class should be compared.

The first class is represented by a Euclidean inner sphere defined by radius  $R_A$  and the second by a Euclidean spherical outer shell defined by the two radii  $R_B$  and  $R_C$ . After training and applying a classifier on those two classes, a score between 0 and 1 can be computed for each input point using Eq.C.1. Let  $D_0$  and  $D_1$  be the distribution of these scores for the matching and non-matching class respectively. For every threshold  $t$  set between  $[0, 1]$ , the following can be computed:

$$\begin{aligned}
\text{True Negative Rate: } & TNR(t) = \int_0^t D_0(x) dx \\
\text{False Positive Rate: } & FPR(t) = \int_t^1 D_0(x) dx \\
\text{False Negative Rate: } & FNR(t) = \int_0^t D_1(x) dx \\
\text{True Positive Rate: } & TPR(t) = \int_t^1 D_1(x) dx
\end{aligned} \tag{C.3}$$

We define the saliency as  $SA = 1 - I$  where  $I$  is the oriented intersection of the score distribution of both classes. By integrating over all possible  $t$ , the global notion of the intersection  $I$  between the two distributions is defined as:

$$I(D_0, D_1) = \int_0^1 FNR(t).FPR(t) dt \tag{C.4}$$

The saliency can therefore be expressed as:

$$\begin{aligned}
SA &= 1 - I(D_0, D_1) \\
&= 1 - \int_0^1 \int_0^t D_1(x).(1 - D_0(x)) dx dt
\end{aligned} \tag{C.5}$$

This saliency score can be computed for every single vertex in the registered training set, providing us with a saliency map.

#### C.1.4 The Ubiquity Metric

Detecting the local saliency is sometimes not enough if the particular shape in question is commonplace on the object's surface. If a salient local feature is repeated too many times in the input data, the ambiguity to find them and assign a unique label will increase. An ideal model landmark should not only be locally salient but globally rare. Given a scoring function  $S_i$  for a given landmark  $i$ , we compute the ubiquity sum function as:

$$U(i) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{i=0}^{V-1} S_i(v_k) \tag{C.6}$$

where  $N$  is the number of scans in the training set and  $V$  the number of vertex in the model mesh. In an ideal situation, unlikely to be encountered in practice, only one vertex  $v$  per training mesh should trigger a non-near-zero score for landmark  $i$  and this score should be 1. A perfect landmark should therefore have a ubiquity score close to unity. In reality, many



vertices will have non-near-zero values and the ubiquity score will be significantly above 1. However, this does provide a second method to find potential model landmarks, which is to select those that generate very low ubiquity scores.

### C.1.5 Classification

In order to separate the two classes (matching and non-matching vertices), several methods can be used. For our experiments, two methods have been tested: Linear Discriminant Analysis method (LDA) and the Adaptive Boosting method (Adaboost). A motivation for using Adaboost, is its ability to perform non-linear classification and thereby improve performance.

In the case of LDA applied naively, the results can be quite unstable, as only one value is generated for per model vertex for all the training set. In order to have more meaningful local values, LDA is computed 20 times with different subsets of the training set. The final vector, which linearly weights the 8 descriptors, is the mean of these results. It appears, from our preliminary experiments, that the differences between the Adaboost and LDA methods are very small. Therefore we chose to use LDA, because it is significantly less computationally expensive.

### C.1.6 Selecting a Set of Landmarks

Once a saliency map has been constructed on the generic model mesh surface, it is possible to extract local maxima as shapes of interest for the system. As the desired output of such a system is a sparse set of points, the notion of locality should be taken into account at this stage. Furthermore, the object might present some repeated features for which we only want one instance in the model.

To avoid using parameters in this part of the system, a simple iterative scheme using the landmark response score map is implemented. Initially, we have one normalised saliency score map and we detect the single point with global maximum value on that map. Using this point as a training landmark, we compute the normalised landmark response score map on the training dataset. If the point was a singular point (e.g. the nose tip) only one high scoring region (coloured as a blue patch on the landmark score map) will appear; if it



was a symmetric point (e.g. an eye corner), two or more patches will appear. This map is subtracted from the first one and the resulting map is normalised. The second best shape of interest will then be the global maximum over this newly created map. By iterating in this fashion, the set of landmarks created will never contain similar (symmetric) shapes and will be relatively sparse.

### C.1.7 Input Parameters

In addition to the training datasets, consisting of registered surfaces, our system requires a number of input parameters that will directly influence the results of the experiments. One obvious source of variation is the number of descriptors, their nature, and the parameters for their computation. An advantage of our system is that those descriptors do not need to be independent. The correlation between descriptors is taken care of in the class separation part of the process. Therefore, we do not need to test the system with different subsets of descriptor. The biggest set of descriptors will always give better results. The only concern that remains is the time of computation in such cases. Here, we use 8 different scalar local-shape descriptors at a single scale. The neighbourhood size is fixed to 15 mm, which previous studies have shown to be adequate for this set of descriptors for hand-placed landmarks [Creusot et al., 2011]. We acknowledge that changing this scale can dramatically change the nature of the detected landmarks. Our goal here is merely to compare manual and automatic model set of landmarks under the same conditions and using the same descriptors. Optimising the set of descriptors and associated scales is another problem altogether.

Another source of variation is the definition of the locality of the features. Here, the differentiation between matching and non-matching vertices is done using Euclidean spheres and shells at different radii.

A further parameter of the system is the sparsity of the set of features (landmarks) to be detected. The system should know how many salient points it should be looking for, and what is the minimum distance between features that can be accepted. The number of features can either be fixed or given as a ratio of the number of vertices in the input model. The minimum separation between features is given as a Euclidean distance. Here, we present results looking at a maximum of 10 features with a minimum distance of 10 mm between any two landmarks.

### C.1.8 Summary of Workflow

For each of the 200 meshes in the dataset, 8 local shape descriptors are computed. For each model vertex, the distribution of values for each descriptor is learnt and approximated with a Gaussian. Landmark score maps are then computed for every model vertex on every training mesh. For a given model vertex, the separation between two classes of matching and non-matching vertices is computed in an 8-dimensional space, where each dimension represents a landmark score value for one descriptor. From this point two alternative methods are considered:

Method 1: The obtained distributions for the two classes are used to compute a saliency score for each vertex of the model, producing a saliency score map.

Method 2: Every single vertex in the model is used as a potential landmark in training so that a landmark score map is available for every vertex. The sum of the elements of this map is used to create a ubiquity score map. The higher the ubiquity score, the more the given landmark is likely to appear all over the mesh. The lower the ubiquity score, the more the landmark being considered is rare.

The maxima on the saliency score map and the minima on the ubiquity score map are our detected shapes of interest in method 1 and method 2 respectively.

## C.2 Datasets

Two different face datasets are used in our experiments. One is synthetic and has been generated from the Basel Face Model [Paysan et al., 2009]. The second is made of real human face scans from the FRGC dataset [Phillips et al., 2005], registered using the Iterative Closest Point (ICP) [Besl and McKay, 1992] technique.

### C.2.1 BFM

We generated 200 random faces from the Basel Face Model (BFM) as well as a null-parametrised face to be used as a mean model. To reduce the computation time, the model was cropped using a sphere of 100 mm around the nose. The inner mouth part was manually erased and the mesh resolution was reduced so that the number of vertices is 2000. The vertex indices on the modified model were used to reconstruct similar low resolution faces

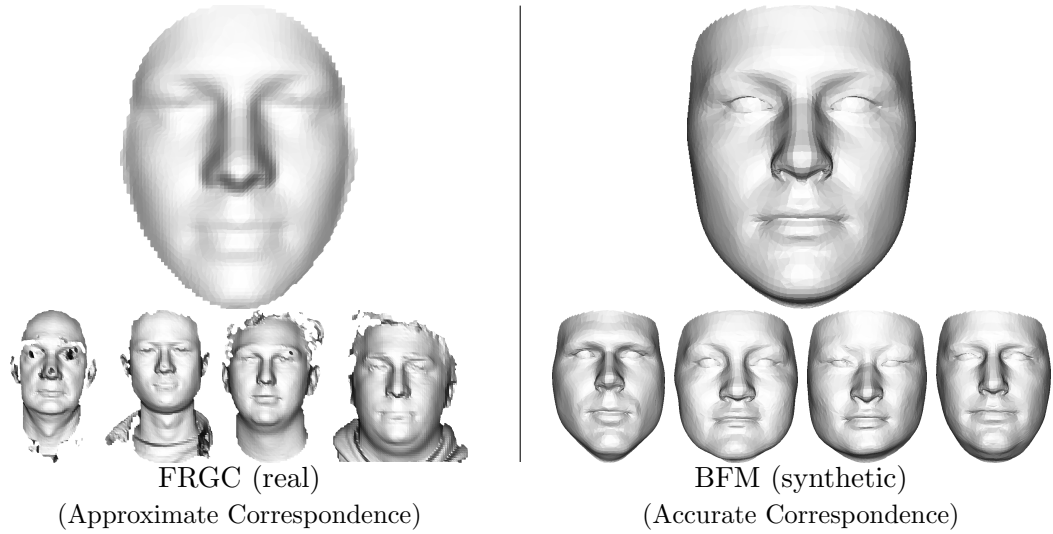


Figure C.2: Generic meshes derived from the FRGC (left) and BFM (right) on which results are projected. Examples from the 200 training meshes associated with these models are shown underneath.

from the 200 meshes in the training data. Every vertex in the model has a correspondence in every mesh of the training set.

### C.2.2 FRGC

For the FRGC, 200 random faces of different individuals were selected from the whole set. ICP-based registration was used to place cropped versions of the meshes into correspondence. The model mesh was generated by averaging depth values. The generated mesh is around 2000 vertices. For every model vertex, a correspondence is present in every training mesh by looking at the closest point in the  $(x, y)$  plane projection. Compared to the BFM, the registration is approximate for a single mesh. However it provides a good mean response, if enough meshes are present in the training set, and allows us to see if our method can be used without supervision, as the correspondence computation doesn't evolve manual manipulations of the data by an operator (no manual anchor points were used for these FRGC registrations).

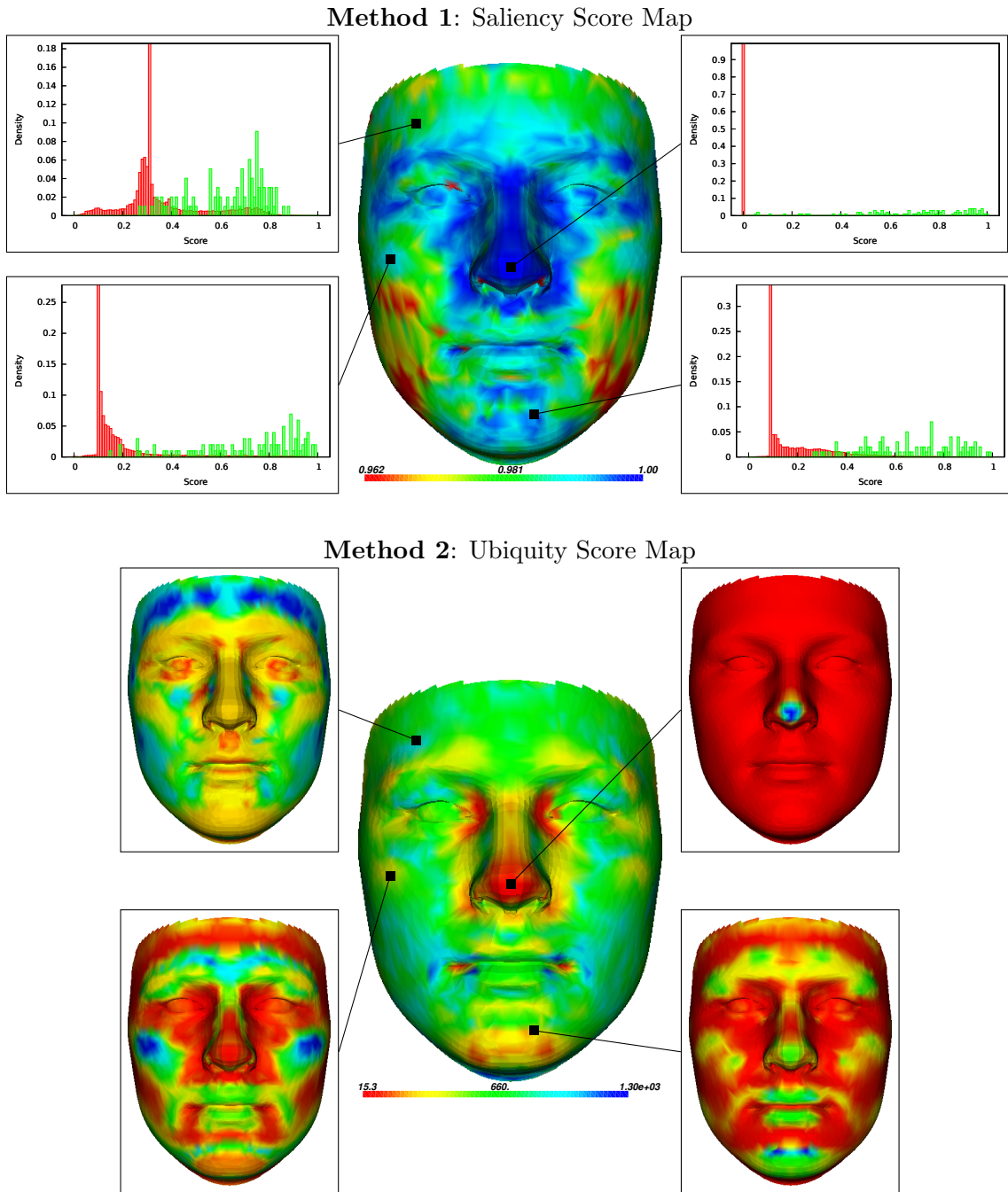


Figure C.3: Resulting map for our two methods. For four points we show, for the first method, a plot of the distribution of the scores associated with the matching local neighbourhood (green) and the non-matching local neighbourhood (red). The values on the central map are formed as the complement of the distribution intersection. For the second method, we show the associated response score map. The values on the central map are the averaged sum over all vertices of the corresponding response score map.

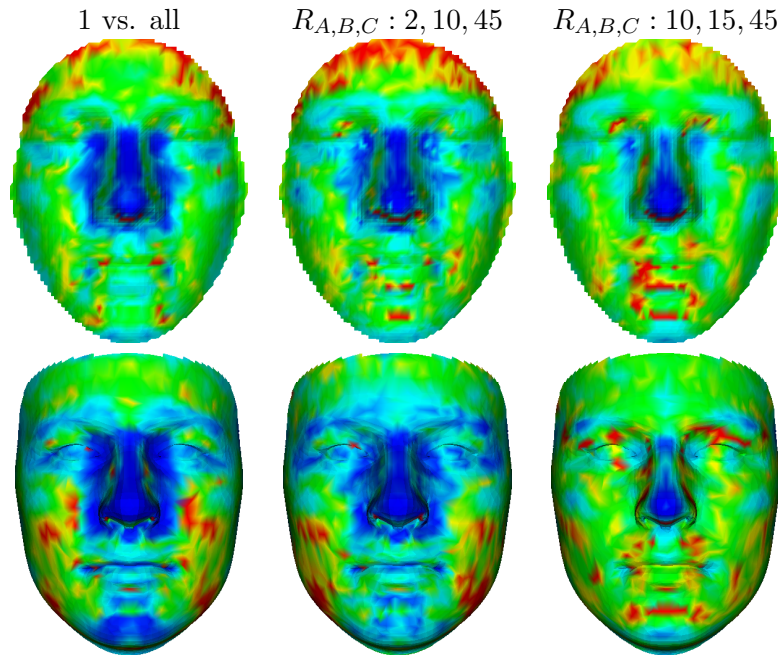


Figure C.4: Saliency map for the two datasets with different locality definitions. Regions of maximal saliency are represented in blue.

### C.3 Results

Figure C.3 shows maps computed using our two methods. A first encouraging result is that both techniques for both datasets find roughly the same regions of the faces as interesting. While the first method tries to find locally salient points, the second tries to find globally rarer and more discriminative points. It appears that, on faces, points that are locally salient are also globally rarer. This is something that we humans find obvious in faces, but that is not true in general.

Figure C.4 shows examples of computed saliency map using different locality definitions. The definition of the locality can not be optimised within the system and has to be provided in input. For the remainder of this chapter we use the middle configuration ( $R_A = 2, R_B = 10$  and  $R_C = 45$ ). When selecting the maxima on this map, 10 shapes of interest can be defined and compared to the 10 manual landmarks commonly used in the literature. One advantage of the automatic detection of the shapes of interest is that the centre of areas with similar shape can automatically be labelled. In Fig. C.5, the shapes of interest and corresponding symmetrical points are presented for both the manual and automatic sets of landmark. When

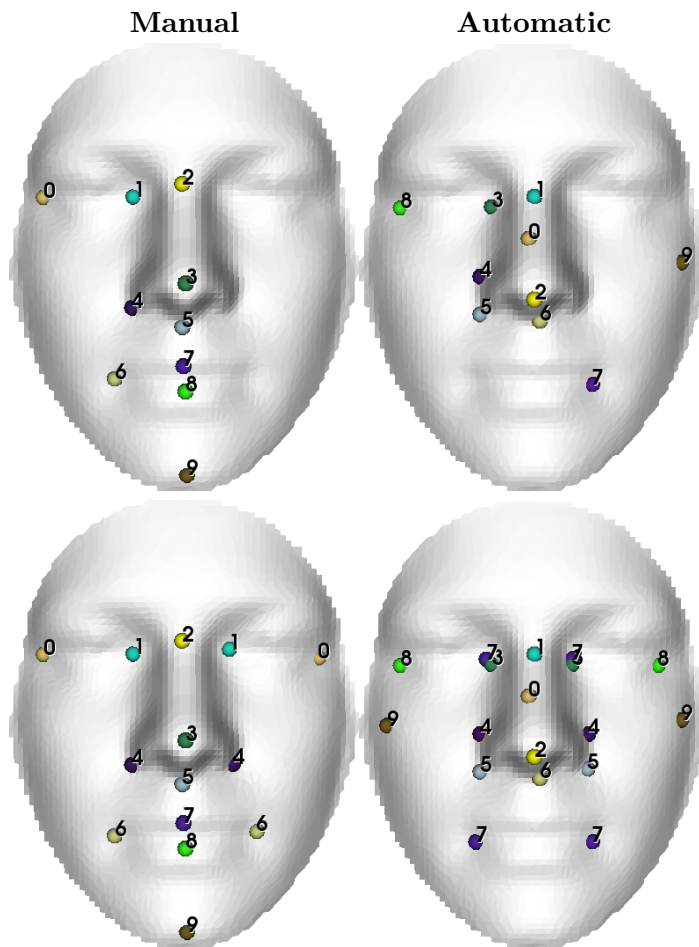


Figure C.5: Manual model set of 10 shapes used in the literature (top left) and the 10 best local shapes detected by our automatic method (top right). The bottom line shows the corresponding “symmetrical” detection (manual and automatic). With the automatic system, the correlations between the mouth and eye shape are detected.

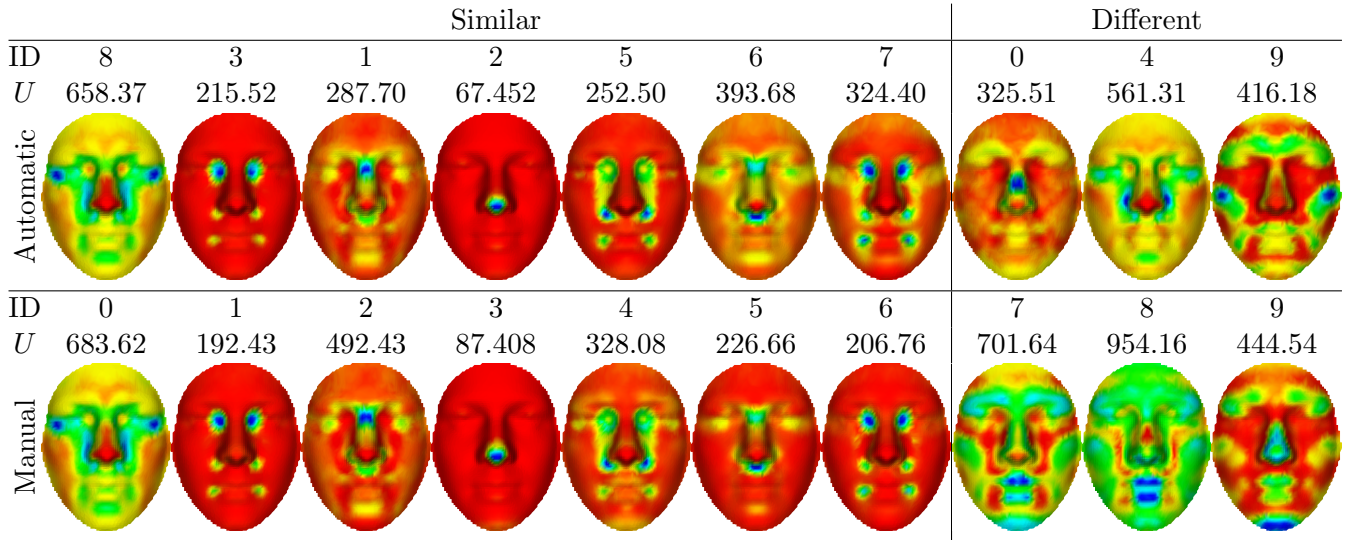


Figure C.6: Similarities and differences between automatic and manual landmarks. Most of the automatic landmark are very similar (in a qualitative way) to the ones picked by human specialists. However some of them are different. For the ones that are different, the definition of the automatic ones leads to tighter response maps (lower ubiquity score  $U$ ).

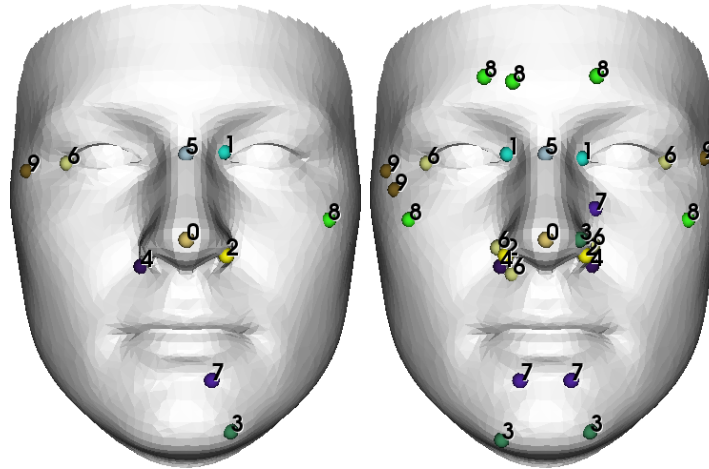


Figure C.7: Automatic model set of shape of interest discovered with the ubiquity score map (left). Corresponding “symmetrical” detection.

comparing both sets (see Fig.C.6), it can be noted that many of the coarse regions detected are similar in both solutions. For the ones that are different, it appears that the shapes selected by the automatic method have a ubiquity score far lower than the human ones, and are therefore more likely to be more easy to detect and label in a face landmarking system.

Since saliency and rarity seem to be correlated on faces, another way at looking at the problem is to try to find local minima of the ubiquity map. Figure C.7 shows minima detected



on the ubiquity score map of the BFM dataset and the detected centroids for the associated landmark score response map. Because the response map for the eye and the mouth corner are similar, our extrema detector does not detect the mouth corner as a shape of interest. It is also not detected when looking at the symmetrical shapes, as the response is not high enough at those points. Better extrema detection techniques might help solve this problem.

## C.4 Conclusion and Future Work

We have presented a way to automatically extract a model set of landmarks to be used as shapes of interest from a registered training set. We discovered some similarities and some differences in the landmark selection made by our automatic process and those typically made by humans. The most important thing is the fact that the landmarks from the automatic model, unlike the human ones, are optimised for a particular task with a given set of tools (local shape descriptors).

While our approach eliminates manual supervision for model landmark selections, it still makes two unsubstantiated assumptions:

- that points are the best things to detect on faces
- that one model should fit all faces

These are made to solve an otherwise too difficult problem. Future work should try to challenge these two points. Different face shapes, ethnic groups, sexes, age groups, might be associated with different optimal sets of landmarks. For example, some people have a shallow cup shape at the ophrion<sup>1</sup> while others have a perfectly monotonous (round or flat) forehead. The number and nature of the landmarks to be found on faces can vary a lot and trying to find a single model that fits everybody might be futile.

Other structures that can be searched for on the face include surface patches and curves (ridgeline, isoline). While these seem more adapted to the geometry of the face (nose bridge, superciliary arches, lips, jaw outline) it is still difficult to detect them reliably, as their lack of locality increases the chances of spurious descriptive values.

---

<sup>1</sup>Points situated at the centre of the forehead just above the superciliary arches and under the frontal eminences.



An other limitation of this work is the circular dependencies between the set of local shape descriptors and the set of landmarks to be used. If one is fixed the other can be found automatically. However finding the best combination of the two is a complex optimisation problem.

In addition to bringing automation to model design, the models generated are expected to be better than manually defined models, as they are produced from a quantitative optimisation process. Validating this hypothesis will be the subject of future work.



# Appendix D

## Abandoned Ideas

If at first you don't succeed, destroy all evidence that you tried.

---

Steven Wright - American comedian and actor

A lot of ideas we started investigating gave poor performance or presented problems too difficult to be treated in the allotted time for this project. By explaining the problem faced with these discarded ideas, we hope to save time for future research students in this area.

**Keypoint detection using descriptor map extrema** One of the first idea we had to find features was to use local extrema of local shape descriptor maps. Most existing techniques were using global-extrema detectors so that the set of keypoint they obtain is sparse. Unlike these techniques, we wanted to detect a more dense set of points and retrieve the labels using graph matching techniques. In figure [D.1](#) examples of keypoints detected as extrema on the maps of classic local shape descriptors are shown. Often, keypoints are detected near known manual-landmark positions.

As choosing one local shape descriptor over an other is sometimes a bit arbitrary, a simple heuristic was used to define a set of keypoints from a set of local shape descriptors: All the keypoints detected with all the descriptors are considered as a whole. Keypoints are generated at every location where more than  $x$  of the methods agree on the presence of a keypoint (see Figure [D.2](#)). To do so, we used a simple cutting tree clustering approach.

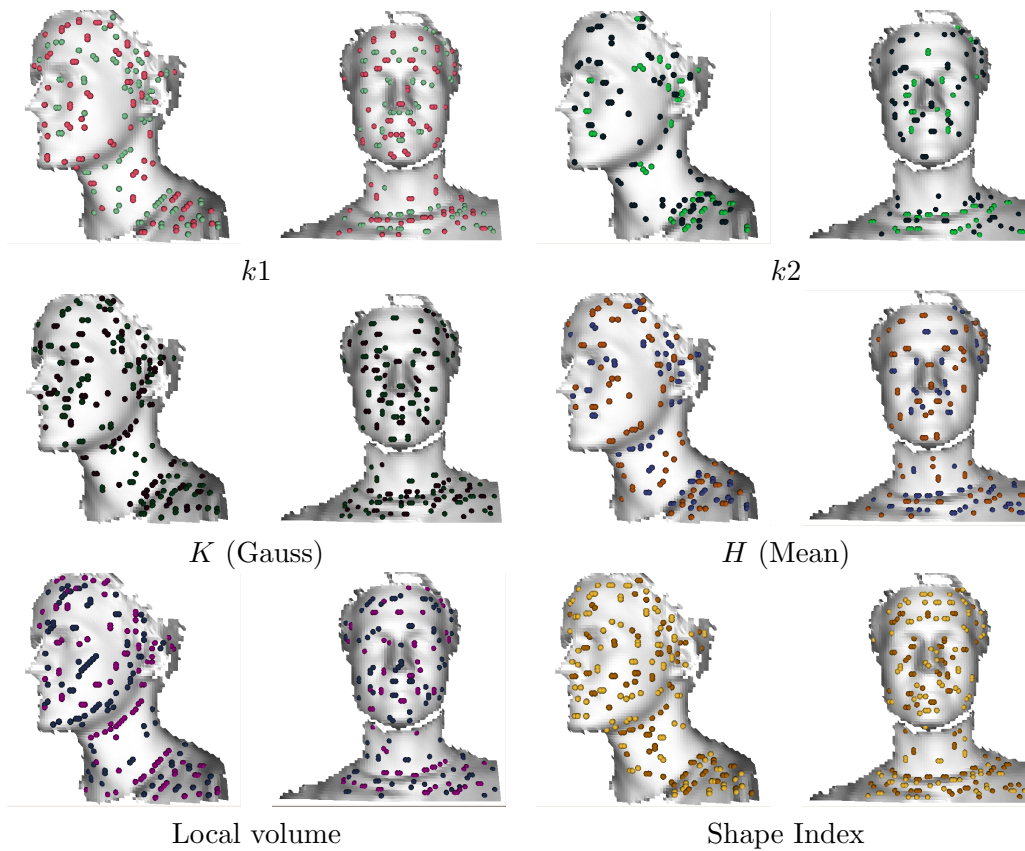


Figure D.1: Examples of local extrema computed on two models of the same individual with different orientation. ( $R_{scalar}^{Eucl.} = R_{extrema}^{Eucl.} = 10\text{ mm}$ )

The reason we abandoned this idea is because we developed a less arbitrary system for keypoint detection (See Chapter 4) based on score maxima instead of raw descriptor value extrema.

**The detection of feature as curves (not as points):** This idea was to detect ridge-lines (detected using the principal curvatures) and isolines (using any scalar map), before matching hypergraphs, where nodes correspond to the detected curves and hyperedges to the relationship between those curves.

While patterns were appearing with this representation (see Figure D.3), the discontinuity of the curves and the change of topology of the curves made the problem of detection really hard. We do think that curves are more meaningful than points to symbolically represent the face. Indeed, a 'smiley' icon (probably the most simplified representation of a face) is constructed with lines not points. However, we didn't succeed in detecting curves that are

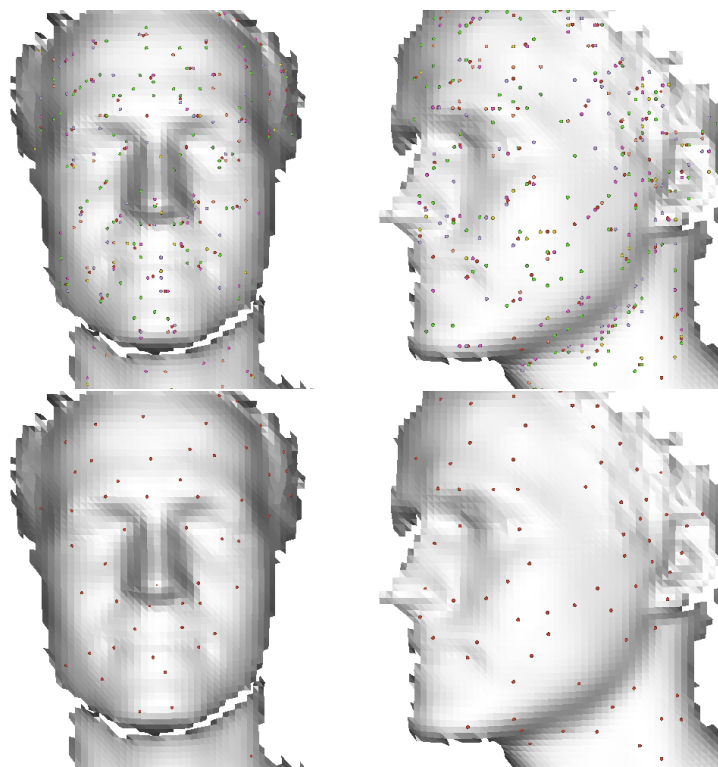


Figure D.2: Keypoints as point-cluster centroids: If a number of different methods indicate the presence of a landmark in an area, a keypoint is created.

easily repeatable from one person to another, or even for two captures of the same person. We discovered that the lack of locality of the curves make them very sensitive to noise and holes. The longer the curve, the higher the probability of problem occurring. Future research trying to deal with this problem will require heavy preprocessing to limit these errors. Moreover, finding ways to merge and split these curves in a robust fashion will help improve the repeatability of the feature detector.

Using a mixture of detected points and curves (see Figure D.5) can also ease the labelling process. Multi-dimensional features can be used as hyperedges between keypoints, bringing important information to the matching problem.

**Hyperedges of undetermined degree** Curve and surface detection on faces can also lead to the use of more discriminative matching techniques using hyperedges of arbitrary degree. Figure D.6 shows a face where both landmarks and curves (isolines in this case) are represented. By creating sets of landmarks that are close to a given curve, we define

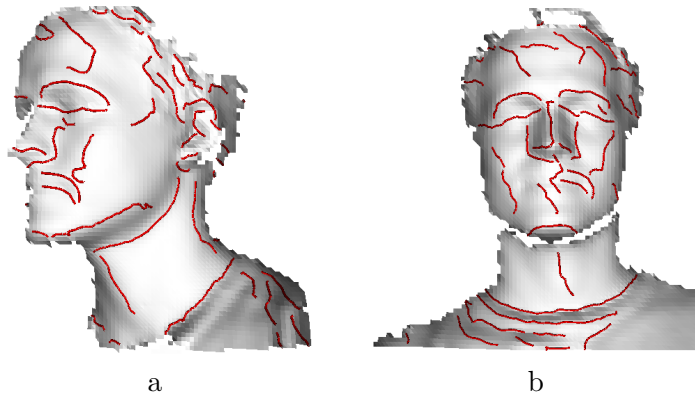


Figure D.3: Example of ridgelines detected on a frontal and profile scan of the same person. While some repeatability patterns can be observed, the curves are far too sensitive to noise and holes to be used as reliable features in our framework.

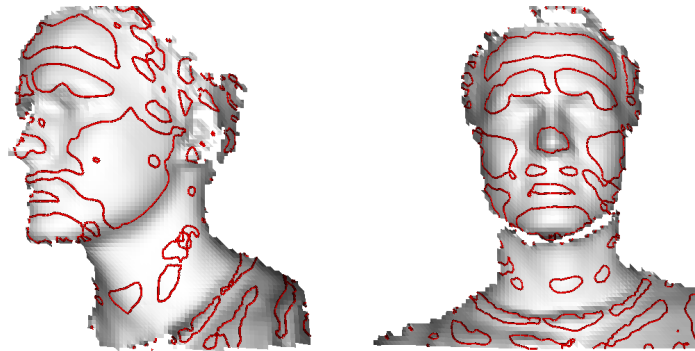


Figure D.4: Example of contour lines drawn on the  $k_1$  curvature at level  $C_0 = 0.0$ .

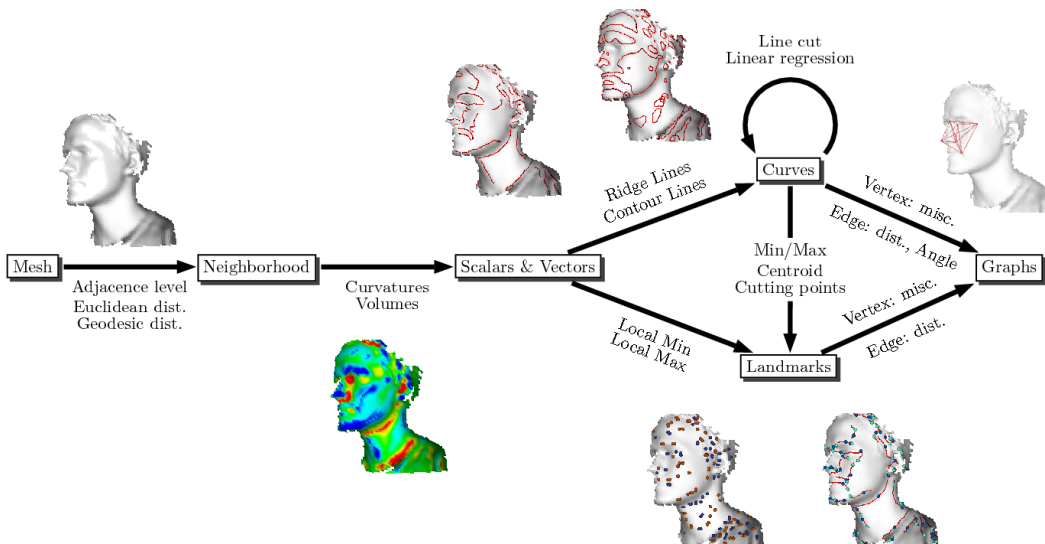


Figure D.5: Diagram of the use of curves. The plan was to use curves as features but also as an additional way to detect keypoints.

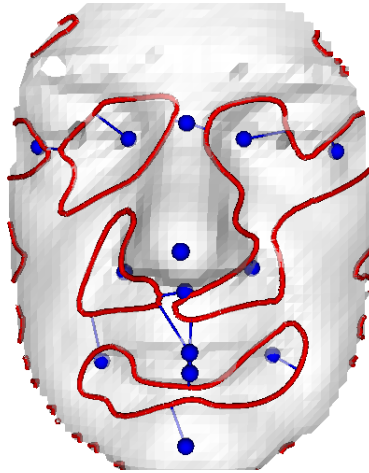


Figure D.6: Example of hyperedges of undetermined degree constructed using some curve detector on the face. Every landmark close to one curve are all connected through a hyperedge. Here matching can be done independently from the degree of the hyperedge (by using the curve length or other measure associated with the set of points).

hyperedges of arbitrary degree. The degree of a hyperedge is the number of landmark close to the curve (or the number of landmarks inside an area for a surface feature).

By using metrics that are common to all curves features (length, surface, map value integration along a path, and so on), our hypergraph matcher is able to associate hyperedges of different degree together. The advantage of using that kind of hyperedges is that there is no combinatorial explosion as the number of hyperedges of high degree is small and correspond to detected features on the query face. Once again, the main problem with this approach is to detect reliably curves or surface features on the face.

**Keypoints as contact points between rigid surfaces** If you place a cast of a face, face down on the floor, there are a very limited number of stable gravitational equilibrium. These equilibrium involved a minimum of 3 contact points between the face and the flat surface (usually: cheekbone, side of nose-tip and superciliary arch). The position of these points are very stable compared to single point detections. Our idea was to detect those points in order to detect other features. The plane surface can also be replaced by other shape, leading to different possible contact points (e.g. a sphere inside the face). One problem with this approach is that the mesh needs to be cleaned and preprocessed. Because of time constraints, we didn't investigate this idea.

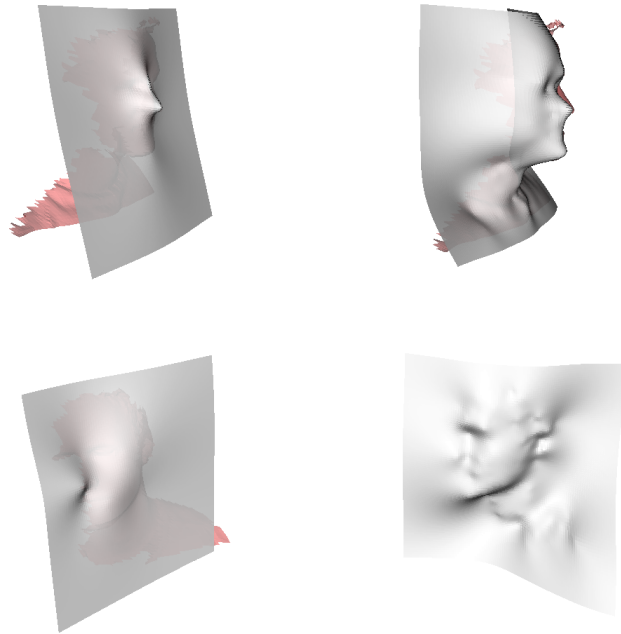


Figure D.7: Face recognition using Thin Plate Spline (TPS). Example of fitting on a frontal (top) and profile view (bottom). The left column uses 14 landmarks while the right column uses automatically detected keypoints (as extrema over the mean curvature map). An interesting point is that, in the last image, the identity of the person can be retrieved by a human operator only watching the TPS surface, implying that the set of points used to create the TPS “contains” the identity information required for face recognition.

**Using Thin-Plate-Spline (TPS) for face representation:** Some face processing systems use radial basis functions to smooth and fill holes in input data. The idea is often that all the vertices of the input contribute to the final model representation. An interesting idea is that maybe keypoints might contain all the information needed for face recognition. Therefore, it might be possible to reconstruct a clean face surface from a set of detected keypoints. The Thin Plate Spline technique was used to test this hypothesis. The plan was to fit a morphable two-dimensional sheet to a set of detected points. The idea behind it was to count how many parameters (dimensions) are necessary to represent identities in a big dataset and when the parameters take the form of a point on the surface of the face. As our project shifted from face recognition to landmarking, this idea hasn’t been properly investigated. It can be noted that this approach would not have been pose-invariant.



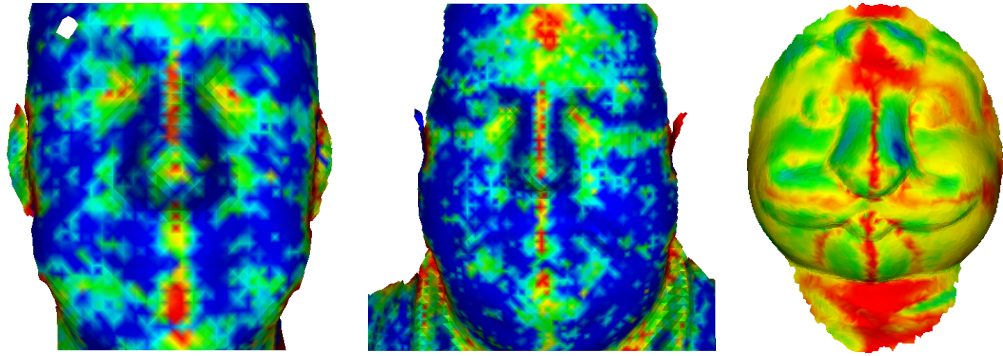


Figure D.8: Examples of Local Symmetry Descriptor (LSD) response maps. Red vertices shows region with high local symmetry.

**Local Symmetry Descriptor (LSD):** When trying to design new local descriptors, we came up with the idea of looking at the degree of symmetry of local shapes. To do so, we constructed, for each vertex  $v$ , a 2D polar histogram  $H^v$  (parametrised by  $\alpha$  in angles and  $\rho$  in radius) averaging the local neighbourhood  $z$  values in a basis including the local surface normal. The polar histogram is composed of  $2 \cdot k_\alpha$  slices of equal angle, each partitioned in  $k_\rho$  cells. We defined the Local Symmetry Descriptor as:

$$LSD(v) = \min_{q \in \{1..k_\alpha\}} \sum_{i \in \{1..k_\rho\}} \sum_{j \in \{1..k_\alpha\}} |H^v[i, sym(j, q)] - H^v[i, j]|$$

where  $sym(j, q)$  is the symmetric slice to  $j$  using the  $q^{th}$  symmetric plane of the histogram.

The min value over all the symmetric planes  $q$  of the histogram corresponds to the best plane of symmetry where the absolute difference between the two sides in terms of  $z$  values is minimum.

Tests performed with this descriptor doesn't seem to give interesting information for most of the landmarks we use. Figure D.8 shows that the obtained results are relatively noisy. Some high symmetry vertices are detected on the sagittal plane but the descriptor seems useless everywhere else. This kind of descriptor requires very good vertex normals and relatively smooth meshes, which is not always the case with our datasets.

**2D Contour** When we first used the FRGC datasets, we had to face a number of challenges linked to the data imperfections. One of which was the offset between the capture of the

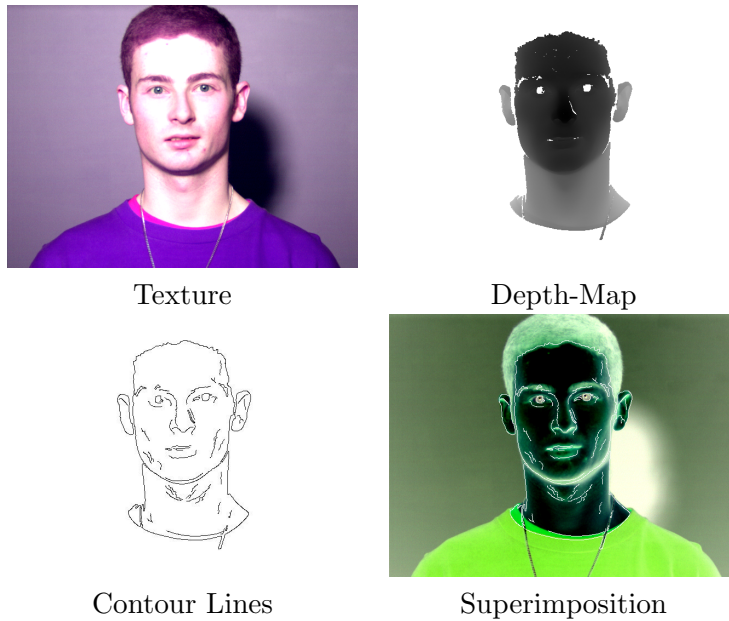


Figure D.9: Generation of depth-map contours.



Figure D.10: Example of error in the FRGC dataset where 2D and 3D data do not correspond.

2D texture image and the 3D structured point cloud. Most of the time this leads to small translations between the two data channels.

In an attempt to automatically correct these errors, we developed simple scripts to generate and superimpose 2D contours of the 3D depth map on the 2D texture image (see Figure D.9 and D.10). An automatic method would have computed contours on both the input images and registered them. However, registration of 2D curves is a difficult task. As we decided not to use the 2D texture for our main project, we didn't pursue this approach.

Another idea was to detect facial features using contours from the depth map, but that kind of method would not have been pose invariant, and the detections are usually poor in the eye region as it can be seen in Figure D.11.



Figure D.11: Example of depth-map contours. The eyes area often lack details.

**On-manifold Face recognition Using Vectorised Hypergraphs** One of the initial ideas of the PhD was to investigate face recognition using sets of detected features on the face surface. If a unique model is used to detect the facial features on the whole dataset (common model), it is possible to represent any hypergraph, generated from these features, as a vector. To deal with occlusion, a binary mask vector can be used in addition to the scalar vector.

By learning the correlations between identities and these vectors using space reduction techniques (e.g. Linear Discriminant Analysis), it would be possible to retrieve identities. The advantages of this approach are that the 2D face recognition techniques can be used (e.g. Fisherface technique) and that the system would be completely pose invariant and aware of possible occlusions. Some might argue that the amount of information contained in the hypergraph is too small to perform face recognition. This is not true if hyperedges of high degree are considered. In addition to local shape descriptors for the nodes and measured anthropological properties for the hyperedges (distances, surfaces, etc), it is possible to attach images to hyperedges of degree 3 or above. Indeed, in our framework, an image can easily be represented by a histogram.

In Figure D.12, depth maps are constructed from view points attached to hyperedge of degree 3. Running a Fisherface recognition technique with that kind of input would achieve better result than using only a single frontal-view depth-map. Furthermore, the presence of a given node or hyperedge can be used to mask some parts of the vector when occlusions are present. Generating 2D texture images with new view points is also possible, as the 2D-3D correspondence can be used. As such a system will require robust facial feature detection, we decided to focus on feature localisation instead of face recognition.

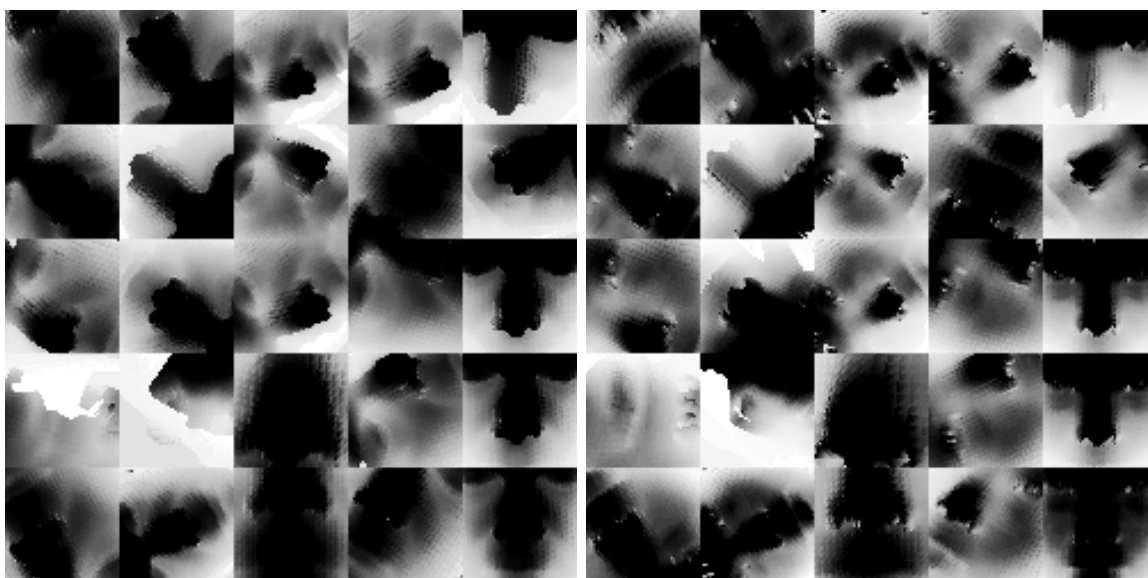


Figure D.12: Multiview 2D depth maps. Two sets of 2D depth maps generated from 2 different individuals from the FRGC database using a set of hyperedges of degree 3 as the basis for the camera positioning.

# List of References

- [Alyuz et al., 2010] Alyuz, N., Gokberk, B., and Akarun, L. (2010). Regional registration for expression resistant 3-d face recognition. *Information Forensics and Security, IEEE Transactions on*, 5(3):425–440, doi:[10.1109/TIFS.2010.2054081](https://doi.org/10.1109/TIFS.2010.2054081). 211
- [Amberg et al., 2007] Amberg, B., Romdhani, S., and Vetter, T. (2007). Optimal step non-rigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, doi:[10.1109/CVPR.2007.383165](https://doi.org/10.1109/CVPR.2007.383165). 246
- [Amberg and Vetter, 2011] Amberg, B. and Vetter, T. (2011). Optimal landmark detection using shape models and branch and bound. *Computer Vision, IEEE International Conference on*, 0:455–462, doi:<http://doi.ieeecomputersociety.org/10.1109/ICCV.2011.6126275>. 84
- [An and Chung, 2008] An, K. H. and Chung, M. J. (2008). 3d head tracking and pose-robust 2d texture map-based face recognition using a simple ellipsoid model. *Intelligent Robots and Systems, 2008. IROS 2008. International Conference on*, pages 307–312, doi:[10.1109/IROS.2008.4650742](https://doi.org/10.1109/IROS.2008.4650742). 84
- [Belhumeur et al., 1997] Belhumeur, P., Hespanha, J., and Kriegman, D. (1997). Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, doi:[10.1109/34.598228](https://doi.org/10.1109/34.598228). 51, 52, 53
- [Ben Amor et al., 2005] Ben Amor, B., Ouji, K., Ardebilian, M., and Chen, L. (2005). 3d face recognition by icp-based shape matching. In *The second International Conference on Machine Intelligence (ACIDCA-ICMI'2005)*. <http://liris.cnrs.fr/publis/?id=1963>. 52, 58, 79
- [Berretti et al., 2010] Berretti, S., Bimbo, A. D., and Pala, P. (2010). Recognition of 3d faces with missing parts based on profile networks. In *1st ACM Workshop on 3D Object Retrieval (ACM 3DOR'10)*, pages 81–86, Firenze, Italy, doi:[10.1145/1877808.1877825](https://doi.org/10.1145/1877808.1877825). 81, 83, 114
- [Berretti et al., 2008] Berretti, S., Del Bimbo, A., and Pala, P. (2008). 3d face recognition by spatial arrangement of iso-geodesic surfaces. *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008*, 1:365–368, doi:[10.1109/3DTV.2008.4547884](https://doi.org/10.1109/3DTV.2008.4547884). 52, 62, 63, 75

- [Besl and McKay, 1992] Besl, P. and McKay, N. (1992). A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, doi:10.1109/34.121791. 57, 105, 144, 252
- [Bevilacqua et al., 2008] Bevilacqua, V., Cariello, L., Carro, G., Daleno, D., and Mastronardi, G. (2008). A face recognition system based on pseudo 2d hmm applied to neural network coefficients. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, Volume 12(7):615–621, doi:10.1007/s00500-007-0253-0. 52, 56
- [Blanz and Vetter, 2003] Blanz, V. and Vetter, T. (2003). Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, doi:10.1109/TPAMI.2003.1227983. 84
- [Bledsoe, 1966] Bledsoe, W. (1966). Man-machine facial recognition. Technical Report Rep. PRI: 22, Panoramic Research Inc., Palo Alto, Cal. 45
- [Bookstein, 1989] Bookstein, F. (1989). Principal warps: thin-plate splines and the decomposition of deformations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(6):567–585, doi:10.1109/34.24792. 58
- [Bronstein et al., 2007] Bronstein, A., Bronstein, M., and Kimmel, R. (2007). Expression-invariant representations of faces. *Image Processing, IEEE Transactions on*, 16(1):188–197, doi:10.1109/TIP.2006.884940. 57, 85
- [Bronstein et al., 2004a] Bronstein, A., Bronstein, M., Kimmel, R., and Spira, A. (2004a). 3d face recognition without facial surface reconstruction. In *Proceedings of ECCV 2004*. [http://www.face-rec.org/algorithms/3D\\_Morph/CIS-2003-05.pdf](http://www.face-rec.org/algorithms/3D_Morph/CIS-2003-05.pdf). 57
- [Bronstein et al., 2004b] Bronstein, M. M., Bronstein, A. M., and Kimmel, R. (2004b). Three-dimensional face recognition. Technical report, Dept. of Computer Science, Technion, Israel, <http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2004/CIS/CIS-2004-04.pdf>. 57
- [Brunelli and Poggio, 1993] Brunelli, R. and Poggio, T. (1993). Face recognition: features versus templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(10):1042–1052, doi:10.1109/34.254061. 52, 59, 65
- [Callaghan and Mahony, 2010] Callaghan, K. and Mahony, M. (2010). An obstacle segmentation and classification system for the visually impaired. In *Optomechatronic Technologies (ISOT), 2010 International Symposium on*, pages 1–6, doi:10.1109/ISOT.2010.5687345. 22
- [Campbell et al., 1995] Campbell, R., Walker, J., and Baron-Cohen, S. (1995). The development of differential use of inner and outer face features in familiar face identification. *Journal of Experimental Child Psychology*, 59(2):196–210, doi:DOI: 10.1006/jecp.1995.1009. 44
- [Castellani et al., 2008] Castellani, U., Cristani, M., Fantoni, S., and Murino, V. (2008). Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Computer Graphics Forum*, 27(2):643–652, doi:10.1111/j.1467-8659.2008.01162.x. 82

- [Castellano et al., 2008] Castellano, M., Mastronardi, G., Daleno, D., Cariello, L., and Decataldo, G. (2008). Computing the 3d face recognition based on pseudo 2d hidden markov models using geodesic distances. *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, pages 335–338, doi:10.1109/IWSSIP.2008.4604435. 52, 56, 57
- [Cazals and Pouget, 2003] Cazals, F. and Pouget, M. (2003). Estimating differential quantities using polynomial fitting of osculating jets. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 177–187, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association. 118
- [Chang et al., 2006] Chang, K. I., Bowyer, K., and Flynn, P. (2006). Multiple nose region matching for 3d face recognition under varying facial expression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1695–1700, doi:10.1109/TPAMI.2006.210. 52, 65, 66, 74, 81, 83, 114, 211
- [Chertok and Keller, 2010] Chertok, M. and Keller, Y. (2010). Efficient high order matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(12):2205–2215, doi:10.1109/TPAMI.2010.51. 95, 161, 172, 223
- [Christmas et al., 1995] Christmas, W. J., Kittler, J., and Petrou, M. (1995). Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):749–764, doi:10.1109/34.400565. 159, 190
- [Chua et al., 2000] Chua, C.-S., Han, F., and Ho, Y.-K. (2000). 3d human face recognition using point signature. *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 233–238, doi:10.1109/AFGR.2000.840640. 52, 60
- [Chua and Jarvis, 1997] Chua, C. S. and Jarvis, R. (1997). Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, Volume 25:63–85, doi:10.1023/A:1007981719186. 60, 61, 79
- [Colbry et al., 2005] Colbry, D., Stockman, G., and Jain, A. (2005). Detection of anchor points for 3d face verification. In *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pages 118–118, doi:10.1109/CVPR.2005.441. 80, 81, 83, 114
- [Cook, 1971] Cook, S. A. (1971). The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA. ACM, doi:10.1145/800157.805047. 171
- [Cootes and Taylor, 1999] Cootes, T. and Taylor, C. (1999). Statistical models of appearance for computer vision. Technical report, University of Manchester., <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.1455>. 76
- [Craw et al., 1999] Craw, I., Costen, N., Kato, T., and Akamatsu, S. (1999). How should we represent faces for automatic recognition? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):725–736, doi:10.1109/34.784286. 74



- [Creusot et al., 2010] Creusot, C., Pears, N., and Austin, J. (2010). 3D face landmark labelling. In *Proceedings of the ACM workshop on 3D object retrieval*, 3DOR '10, pages 27–32, Firenze, Italy. ACM, doi:10.1145/1877808.1877815. 221
- [Creusot et al., 2011] Creusot, C., Pears, N., and Austin, J. (2011). Automatic keypoint detection on 3d faces using a dictionary of local shapes. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pages 204–211, doi:10.1109/3DIMPVT.2011.33. 246, 251
- [Cristinacce, 2004] Cristinacce, D. (2004). *Automatic Detection of Facial Features in Grey Scale Images*. PhD thesis, University of Manchester, [http://david.cristinacce.net/publications/cristinacce\\_thesis.pdf](http://david.cristinacce.net/publications/cristinacce_thesis.pdf). 74
- [D’Hose et al., 2007] D’Hose, J., Colineau, J., Bichon, C., and Dorizzi, B. (2007). Precise localization of landmarks on 3d faces using gabor wavelets. *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on*, pages 1–6, doi:10.1109/BTAS.2007.4401927. 75, 81, 83, 114
- [Dibeklioglu et al., 2008] Dibeklioglu, H., Salah, A., and Akarun, L. (2008). 3d facial landmarking under expression, pose, and occlusion variations. In *BTAS08*, pages 1–6, doi:10.1109/BTAS.2008.4699324. 82, 124
- [Duchenne et al., 2009] Duchenne, O., Bach, F., Kweon, I., and Ponce, J. (2009). A tensor-based algorithm for high-order graph matching. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1980–1987, doi:10.1109/CVPR.2009.5206619. 95, 161, 172, 196, 223
- [Dutagaci et al., 2006] Dutagaci, H., Sankur, B., and Yemez, Y. (2006). 3d face recognition by projection-based methods. In Delp, III, E. J. and Wong, P. W., editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6072 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 194–204, doi:10.1117/12.643089. 69
- [Elad et al., 2001] Elad, M., Tal, A., and Ar, S. (2001). Content based retrieval of vrml objects - an iterative and interactive approach. In *The 6th Eurographics workshop in Multimedia*, Manchester UK. 57
- [Ellis et al., 1979] Ellis, H. D., Shepherd, J. W., and Davies, G. M. (1979). Identification of familiar and unfamiliar faces from internal and external features: some implications for theories of face recognition. *Perception*, 8(4):431–439, doi:doi:10.1068/p080431. 12, 71, 72
- [Faltemier et al., 2008a] Faltemier, T., Bowyer, K., and Flynn, P. (2008a). A region ensemble for 3-d face recognition. *Information Forensics and Security, IEEE Transactions on*, 3(1):62–73, doi:10.1109/TIFS.2007.916287. 52, 66, 67
- [Faltemier et al., 2008b] Faltemier, T., Bowyer, K., and Flynn, P. (2008b). Rotated profile signatures for robust 3d feature detection. In *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–7, doi:10.1109/AFGR.2008.4813413. 81, 83, 114



- [Farkas, 1994] Farkas, L. G. (1994). *Anthropometry of the Head and Face*. Raven Press, New York, NY, USA. 75, 104
- [Feng and Yuen, 2000] Feng, G. and Yuen, P. (2000). Recognition of head-and-shoulder face image using virtual frontal-view image. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 30(6):871–882, doi:10.1109/3468.895926. 54
- [Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, doi:10.1145/358669.358692. 186, 206
- [Freund and Schapire, 1997] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, doi:10.1006/jcss.1997.1504. 147
- [Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA. ISBN:0716710455. 171
- [Gill-Robinson et al., 2006] Gill-Robinson, H., Elias, J., Bender, F., Allard, T. T., and Hoppa, R. D. (2006). Using image analysis software to create a physical skull model for the facial reconstruction of a wrapped akhmimic mummy. *Journal of Computing and Information Technology*, 14(1):45–51, doi:10.2498/cit.2006.01.05. 44
- [Gokberk et al., 2008] Gokberk, B., Dutagaci, H., Ulas, A., Akarun, L., and Sankur, B. (2008). Representation plurality and fusion for 3-d face recognition. *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, 38(1):155–173, doi:10.1109/TSMCB.2007.908865. 52, 53, 69, 71
- [Goldfeather and Interrante, 2004] Goldfeather, J. and Interrante, V. (2004). A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.*, 23(1):45–63, doi:10.1145/966131.966134. 118
- [Goldstein et al., 1971] Goldstein, A., Harmon, L., and Lesk, A. (1971). Identification of human faces. *Proceedings of the IEEE*, 59(5):748–760. 45, 46, 52, 59
- [Gordon, 1991] Gordon, G. (1991). *Face Recognition from Depth and Curvature*. PhD thesis, Harvard University, Division of Applied Sciences. 52, 59
- [Gordon and Vincent, 1992] Gordon, G. G. and Vincent, L. M. (1992). Application of morphology to feature extraction for face recognition. In Dougherty, E. R., Astola, J. T., and Boncelet, C. G., editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 1658 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 151–164, doi:10.1117/12.58390. 79
- [Gramkow, 2001] Gramkow, C. (2001). On averaging rotations. *International Journal of Computer Vision*, 42(1):7–16, 10.1023/A:1011129215388. 185

- [Gupta et al., 2007a] Gupta, S., Aggarwal, J., Markey, M., and Bovik, A. (2007a). 3d face recognition founded on the structural diversity of human faces. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–7, doi:10.1109/CVPR.2007.383053. 62
- [Gupta et al., 2007b] Gupta, S., Markey, M. K., Aggarwal, J., and Bovik, A. C. (2007b). Three dimensional face recognition based on geodesic and euclidean distances. In *IS&T/SPIE Symposium on Electronic Imaging: Vision Geometry XV, Proceedings of the SPIE*, doi:10.1117/12.704535. 52, 61, 62, 84, 246
- [Hallinan et al., 1999] Hallinan, P. W., Gordon, G. G., Yuille, A. L., Giblin, P., and Mumford, D. (1999). *Two- and three-dimensional patterns of the face*. A. K. Peters, Ltd., Natick, MA, USA. 75
- [Horn, 1987] Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, [http://people.csail.mit.edu/bkph/papers/Absolute\\_Orientation.pdf](http://people.csail.mit.edu/bkph/papers/Absolute_Orientation.pdf). 183
- [Hsieh et al., 2009] Hsieh, C.-K., Lai, S.-H., and Chen, Y.-C. (2009). Expression-invariant face recognition with constrained optical flow warping. *Multimedia, IEEE Transactions on*, 11(4):600–610, doi:10.1109/TMM.2009.2017606. 85, 86
- [Irfanoglu et al., 2004] Irfanoglu, M., Gokberk, B., and Akarun, L. (2004). 3d shape-based face recognition using automatically registered facial surfaces. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 4:183–186, doi:10.1109/ICPR.2004.1333734. 52, 61, 76
- [Itskovich and Tal, 2011] Itskovich, A. and Tal, A. (2011). Surface partial matching and application to archaeology. *Computers & Graphics*, 35(2):334 – 341, doi:10.1016/j.cag.2010.11.010. 82
- [Johnson and Hebert, 1999] Johnson, A. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):433–449. 78, 125
- [Kanade, 1973] Kanade, T. (1973). *Picture Processing System by Computer Complex and Recognition of Human Faces*. PhD thesis, Kyoto University, [http://www.ri.cmu.edu/pub\\_files/pub3/kanade\\_takeo\\_1973\\_1/kanade\\_takeo\\_1973\\_1.pdf](http://www.ri.cmu.edu/pub_files/pub3/kanade_takeo_1973_1/kanade_takeo_1973_1.pdf). 45, 46, 48, 52, 59
- [Kilgour and Lederman, 2002] Kilgour, A. and Lederman, S. (2002). Face recognition by hand. *Perception & Psychophysics*, 64(3):339–352, <http://www.ncbi.nlm.nih.gov/pubmed/12049276>. 22, 73
- [Kim and Choi, 2009] Kim, J.-S. and Choi, S.-M. (2009). Symmetric deformation of 3d face scans using facial features and curvatures. *Comput. Animat. Virtual Worlds*, 20:289–300, doi:10.1002/cav.v20:2/3. 124
- [Kimmel and Sethian, 1998] Kimmel, R. and Sethian, J. A. (1998). Computing geodesic paths on manifolds. In *Proc. Natl. Acad. Sci. USA*, volume 95, pages 8431–8435. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.3190>. 117

- [Koudelka et al., 2005] Koudelka, M., Koch, M., and Russ, T. (2005). A prescreener for 3d face recognition using radial symmetry and the hausdorff fraction. *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pages 168–168, doi:10.1109/CVPR.2005.566. 52, 66
- [Lawrence et al., 1997] Lawrence, S., Giles, C., Tsoi, A. C., and Back, A. (1997). Face recognition: a convolutional neural-network approach. *Neural Networks, IEEE Transactions on*, 8(1):98–113, doi:10.1109/72.554195. 52
- [Lederman et al., 2007] Lederman, S., Kilgour, A., Kitada, R., Klatzky, R., and Hamilton, C. (2007). Haptic face processing. *Canadian Journal of Experimental Psychology*, 61(3):230–241, doi:10.1037/cjep2007024. 73
- [Lee and Seung, 1999] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, doi:10.1038/44565. 49, 52, 53, 54
- [Lee et al., 2002] Lee, K., Byatt, G., and Rhodes, G. (2002). Caricature effects, distinctiveness, and identification: Testing the face-space framework. *Psychological Science*, 11(5):379–385, doi:10.1111/1467-9280.00274. 73
- [Lewis, 1999] Lewis, M. B. (1999). Are caricatures special? evidence of peak shift in face recognition. *European Journal of Cognitive Psychology*, 11(1):105–117, doi:10.1080/713752302. 73
- [Li et al., 2003a] Li, Y., Gong, S., and Liddell, H. (2003a). Constructing facial identity surfaces for recognition. *International Journal of Computer Vision*, 53(1):71–92, doi:10.1023/A:1023083725143. 55
- [Li et al., 2003b] Li, Y., Gong, S., and Liddell, H. (2003b). Recognising trajectories of facial identities using kernel discriminant analysis. *Image and Vision Computing*, 21(13-14):1077–1086, doi:DOI: 10.1016/j.imavis.2003.08.010. British Machine Vision Computing 2001. 54, 55
- [Liu and Wechsler, 2000] Liu, C. and Wechsler, H. (2000). Evolutionary pursuit and its application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(6):570–582, doi:10.1109/34.862196. 52, 53
- [Liu et al., 2007] Liu, C.-C., Dai, D.-Q., and Yan, H. (2007). Local discriminant wavelet packet coordinates for face recognition. *J. Mach. Learn. Res.*, 8:1165–1195. 91
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, doi:10.1023/B:VISI.0000029664.99615.94. 68, 82, 83
- [Lu et al., 2004] Lu, X., Colbry, D., and Jain, A. (2004). Three-dimensional model based face recognition. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 1:362–366 Vol.1, doi:10.1109/ICPR.2004.1334127. 52, 58, 76
- [Lu and Jain, 2006] Lu, X. and Jain, A. (2006). Deformation modeling for robust 3d face matching. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2:1377–1383, doi:10.1109/CVPR.2006.96. 76, 77

- [Lu and Jain, 2005] Lu, X. and Jain, A. K. (2005). Deformation analysis for 3d face matching. *Application of Computer Vision, 2005. WACV/MOTIONS '05. Seventh IEEE Workshops on*, 1:99–104, doi:10.1109/ACVMOT.2005.40. 52, 58
- [Max, 1999] Max, N. (1999). Weights for computing vertex normals from facet normals. *J. Graph. Tools*, 4:1–6, <http://portal.acm.org/citation.cfm?id=334709.334710>. 115
- [McKone et al., 2006] McKone, E., Kanwisher, N., and Duchaine, B. C. (2006). Can generic expertise explain special processing for faces? *Trends in Cognitive Sciences*, 11(1):8–15, doi:10.1016/j.tics.2006.11.002. 22, 90
- [Medioni et al., 2009] Medioni, G., Choi, J., Kuo, C.-H., and Fidaleo, D. (2009). Identifying noncooperative subjects at a distance using face images and inferred three-dimensional face models. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 39(1):12–24, doi:10.1109/TSMCA.2008.2007979. 86
- [Mian et al., 2006a] Mian, A., Bennamoun, M., and Owens, R. (2006a). 2d and 3d multimodal hybrid face recognition. In *Computer Vision - ECCV*, pages 344–355, doi:10.1007/11744078\_27. 52, 68, 76, 77
- [Mian et al., 2007a] Mian, A., Bennamoun, M., and Owens, R. (2007a). *Advances in Biometrics*, chapter Keypoint Identification and Feature-Based 3D Face Recognition, pages 163–171. Springer Berlin / Heidelberg, doi:10.1007/978-3-540-74549-5\_18. 52, 63, 79
- [Mian et al., 2007b] Mian, A., Bennamoun, M., and Owens, R. (2007b). An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(11):1927–1943, doi:10.1109/TPAMI.2007.1105. 52, 68, 74, 78, 79
- [Mian et al., 2010] Mian, A., Bennamoun, M., and Owens, R. (2010). On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2):348–361, doi:10.1007/s11263-009-0296-z. 82
- [Mian et al., 2008] Mian, A. S., Bennamoun, M., and Owens, R. (2008). Keypoint detection and local feature matching for textured 3d face recognition. *Int. J. Comput. Vision*, 79(1):1–12, doi:10.1007/s11263-007-0085-5. 63, 64
- [Mian et al., 2006b] Mian, A. S., Bennamoun, M., and Owens, R. A. (2006b). Automatic 3d face detection, normalization and recognition. In *3DPVT*, pages 735–742, doi:10.1109/3DPVT.2006.32. 68, 75, 77, 81, 83, 114, 211
- [Moreno et al., 2003] Moreno, A. B., Sánchez, A., Vélez, J. F., and Díaz, F. J. (2003). Face recognition using 3d surface-extracted descriptors. In *Irish Machine Vision and Image Processing*. <http://gavab.escet.urjc.es/articulos/imvip03.pdf>. 52, 67, 76
- [Nair and Cavallaro, 2008] Nair, P. and Cavallaro, A. (2008). Matching 3d faces with partial data. In *British Machine Vision Conference (BMVC)*. <http://www.comp.leeds.ac.uk/bmvc2008/proceedings/papers/286.pdf>. 52, 58

- [Nair and Cavallaro, 2009] Nair, P. and Cavallaro, A. (2009). 3-d face detection, landmark localization, and registration using a point distribution model. *Multimedia, IEEE Transactions on*, 11(4):611–623, doi:10.1109/TMM.2009.2017629. 76
- [Pan and Wu, 2005] Pan, G. and Wu, Z. (2005). 3d face recognition from range data. *Int. J. Image Graphics*, pages 573–594. 52, 64
- [Paysan et al., 2009] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. (2009). A 3d face model for pose and illumination invariant face recognition. In *Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS '09*, pages 296–301, Washington, DC, USA. IEEE Computer Society. 252
- [Pears, 2008] Pears, N. (2008). Rbf shape histograms and their application to 3d face processing. *Automatic Face & Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–8, doi:10.1109/AFGR.2008.4813442. 78
- [Pears et al., 2010] Pears, N., Heseltine, T., and Romero, M. (2010). From 3d point clouds to pose-normalised depth maps. *International Journal of Computer Vision*, 89(2):152–176, doi:10.1007/s11263-009-0297-y. 81, 83, 114, 125
- [Phillips et al., 2011] Phillips, P., Beveridge, J., Draper, B., Givens, G., O’Toole, A., Bolme, D., Dunlop, J., Lui, Y. M., Sahibzada, H., and Weimer, S. (2011). An introduction to the good, the bad, and the ugly face recognition challenge problem. In *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 346–353, doi:10.1109/FG.2011.5771424. 91
- [Phillips et al., 2005] Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., Marques, J., Min, J., and Worek, W. (2005). Overview of the face recognition grand challenge. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:947–954, doi:10.1109/CVPR.2005.268. 99, 252
- [Pouget, 2005] Pouget, M. (2005). *Geometry of surfaces : from the estimation of local differential quantities to the robust extraction of global differential features*. PhD thesis, Université Nice Sophia-Antipolis, France. 80
- [Rodríguez et al., 2008] Rodríguez, J., Bortfeld, H., Rudomín, I., Hernández, B., and Gutiérrez-Osuna, R. (2008). The reverse-caricature effect revisited: Familiarization with frontal facial caricatures improves veridical face recognition. *Applied Cognitive Psychology*, doi:10.1002/acp.1539. 73
- [Romero, 2010] Romero, M. (2010). *Landmark Localisation in 3D Face Data*. PhD thesis, University of York, UK, Department of Computer Science. 166
- [Romero and Pears, 2009a] Romero, M. and Pears, N. (2009a). Landmark localisation in 3d face data. In *Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on*, pages 73–78, doi:10.1109/AVSS.2009.90. 81, 83, 104, 105, 114, 211

- [Romero and Pears, 2009b] Romero, M. and Pears, N. (2009b). Point-pair descriptors for 3d facial landmark localisation. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS '09. IEEE 3rd International Conference on*, pages 1–6, doi:10.1109/BTAS.2009.5339009. 165
- [Romero-Huertas and Pears, 2008] Romero-Huertas, M. and Pears, N. (2008). 3d facial landmark localisation by matching simple descriptors. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–6, doi:10.1109/BTAS.2008.4699390. 246
- [Russ et al., 2006] Russ, T., Boehnen, C., and Peters, T. (2006). 3d face recognition using 3d alignment for pca. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2:1391–1398, doi:10.1109/CVPR.2006.13. 52, 58, 59
- [Savran et al., 2008] Savran, A., Alyüz, N., Dibeklioglu, H., Çeliktutan, O., Gökberk, B., Sankur, B., and Akarun, L. (2008). Bosphorus database for 3d face analysis. In *Biometrics and Identity Management: First European Workshop, BIOID 2008, Roskilde, Denmark, May 7-9, 2008.*, pages 47–56, Berlin, Heidelberg. Springer-Verlag, doi:10.1007/978-3-540-89991-4\_6. 100, 105, 209
- [Schaefer et al., 2006] Schaefer, K., Fink, B., Grammer, K., Mitteroecker, P., Gunz, P., and Bookstein, F. L. (2006). Female appearance: facial and bodily attractiveness as shape. *Psychology Science*, 48(2):187–204. 44
- [Schwaninger et al., 2002] Schwaninger, A., Lobmaier, J. S., and Collishaw, S. M. (2002). Role of featural and configural information in familiar and unfamiliar face recognition. In *Second International Workshop on Biologically Motivated Computer Vision*, pages 643–650, Berlin, Germany. Springer. 72, 73
- [Segundo et al., 2007] Segundo, M., Queirolo, C., Bellon, O., and Silva, L. (2007). Automatic 3d facial segmentation and landmark detection. *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 431–436, doi:10.1109/ICIAP.2007.4362816. 76, 81, 83, 114, 211
- [Segundo et al., 2010] Segundo, M., Silva, L., Bellon, O. R. P., and Queirolo, C. C. (2010). Automatic face segmentation and facial landmark detection in range images. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(5):1319–1330, doi:10.1109/TSMCB.2009.2038233. 211
- [Sirovich and Kirby, 1987] Sirovich, L. and Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America*, 4:519–524, doi:10.1364/JOSAA.4.000519. 46
- [Slater and Quinn, 2001] Slater, A. and Quinn, P. C. (2001). Face recognition in the newborn infant. *Infant and Child Development*, 10(1-2):21–24, doi:10.1002/icd.241. 44
- [Suganthan et al., 2008] Suganthan, S., Coleman, S., and Scotney, B. (2008). Combining gradient operators and dihedral angle for 2d and 3d feature extraction. *Machine Vision and Image Processing Conference, 2008. IMVIP '08. International*, pages 117–122, doi:10.1109/IMVIP.2008.22. 79



- [Sun and Yin, 2008] Sun, Y. and Yin, L. (2008). 3d spatio-temporal face recognition using dynamic range model sequences. *Computer Vision and Pattern Recognition Workshops, 2008. CVPR Workshops 2008. IEEE Computer Society Conference on*, 1:1–7, doi:10.1109/CVPRW.2008.4563125. 49
- [Surazhsky et al., 2005] Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S. J., and Hoppe, H. (2005). Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.*, 24(3):553–560, doi:10.1145/1073204.1073228. 117
- [Szeptycki et al., 2009] Szeptycki, P., Ardabilian, M., and Chen, L. (2009). A coarse-to-fine curvature analysis-based rotation invariant 3D face landmarking. In *International Conference on Biometrics: Theory, Applications and Systems*, pages 32–37, doi:10.1109/BTAS.2009.5339052. 81, 83, 104, 114, 246
- [Thompson, 1980] Thompson, P. (1980). Margaret thatcher: A new illusion. *Perception*, 9(4):483–484. 44
- [Turk and Pentland, 1991a] Turk, M. and Pentland, A. (1991a). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, doi:10.1162/jocn.1991.3.1.71, <http://www.face-rec.org/algorithms/PCA/jcn.pdf>. 46, 47
- [Turk and Pentland, 1991b] Turk, M. and Pentland, A. (1991b). Face recognition using eigenfaces. *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pages 586–591, doi:10.1109/CVPR.1991.139758. 74
- [Valstar et al., 2006] Valstar, M. F., Pantic, M., Ambadar, Z., and Cohn, J. F. (2006). Spontaneous vs. posed facial behavior: automatic analysis of brow actions. In *ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces*, pages 162–170, New York, NY, USA. ACM, doi:10.1145/1180995.1181031. 44
- [Wang et al., 2002] Wang, Y., Chua, C.-S., and Ho, Y.-K. (2002). Facial feature detection and face recognition from 2d and 3d images. *Pattern Recognition Letters*, 23(10):1191 – 1202, doi:DOI: 10.1016/S0167-8655(02)00066-1. 52, 61
- [Whitehill et al., 2008] Whitehill, J., Bartlett, M., and Movellan, J. (2008). Automatic facial expression recognition for intelligent tutoring systems. *CVPR 2008 Workshop on Human Communicative Behavior Analysis*, doi:10.1109/CVPRW.2008.4563182. 44
- [Whitehill and Movellan, 2008] Whitehill, J. and Movellan, J. (2008). Personalized facial attractiveness prediction. *Automatic Face & Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–7, doi:10.1109/AFGR.2008.4813332. 44
- [Wiskott et al., 1999] Wiskott, L., Fellous, J.-M., Krüger, N., and von der Malsburg, C. (1999). *Face Recognition by Elastic Bunch Graph Matching*, chapter Chapter 11, pages 355–396. CRC Press. ISBN 0-8493-2055-0. 52, 60, 61, 84
- [Xu et al., 2008] Xu, Z., Chen, H., Zhu, S.-C., and Luo, J. (2008). A hierarchical compositional model for face representation and sketching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6):955–969, doi:10.1109/TPAMI.2008.50. 65

- [Xu and Luo, 2006] Xu, Z. and Luo, J. (2006). Face recognition by expression-driven sketch graph matching. *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 3:1119–1122, doi:10.1109/ICPR.2006.525. 52, 65
- [Yang et al., 2009] Yang, J., Liao, Z.-W., Li, X.-N., and Wu, Z.-D. (2009). A method for robust nose tip location across pose variety in 3d face data. *Informatics in Control, Automation and Robotics, 2009. CAR '09. International Asia Conference on*, pages 114–117, doi:10.1109/CAR.2009.86. 75
- [Yoshizawa et al., 2008] Yoshizawa, S., Belyaev, A., Yokota, H., and Seidel, H.-P. (2008). Fast, robust, and faithful methods for detecting crest lines on meshes. *Computer Aided Geometric Design*, 25(8):545–560, doi:10.1016/j.cagd.2008.06.008. 121, 146
- [Zafeiriou and Pitas, 2008] Zafeiriou, S. and Pitas, I. (2008). Discriminant graph structures for facial expression recognition. *Multimedia, IEEE Transactions on*, 10(8):1528–1540, doi:10.1109/TMM.2008.2007292. 44
- [Zaharescu et al., 2009] Zaharescu, A., Boyer, E., Varanasi, K., and Horaud, R. (2009). Surface feature detection and description with applications to mesh matching. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 373–380, doi:10.1109/CVPR.2009.5206748. 82
- [Zass and Shashua, 2008] Zass, R. and Shashua, A. (2008). Probabilistic graph and hypergraph matching. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, doi:10.1109/CVPR.2008.4587500. 95, 161, 172, 173, 196, 223
- [Zhan et al., 2008] Zhan, C., Li, W., Ogunbona, P., and Safaei, F. (2008). A real-time facial expression recognition system for online games. *Int. J. Comput. Games Technol.*, 8(3):1–7, doi:10.1155/2008/542918. 44
- [Zhang et al., 2008] Zhang, T., Fang, B., Tang, Y. Y., He, G., and Wen, J. (2008). Topology preserving non-negative matrix factorization for face recognition. *Image Processing, IEEE Transactions on*, 17(4):574–584, doi:10.1109/TIP.2008.918957. 52, 53
- [Zhao et al., 2011] Zhao, X., Dellandrand, E., Chen, L., and Kakadiaris, I. A. (2011). Accurate landmarking of three-dimensional facial data in the presence of facial expressions and occlusions using a three-dimensional statistical facial feature model. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(5):1417–1428, doi:10.1109/TSMCB.2011.2148711. 81, 83, 114, 246
- [Zhou et al., 2006] Zhou, D., Yang, X., Peng, N., and Wang, Y. (2006). Improved-lda based face recognition using both facial global and local information. *Pattern Recognition Letters*, 27(6):536–543, doi:DOI: 10.1016/j.patrec.2005.09.015. 54
- [Zhou and Tang, 2008] Zhou, D. K. and Tang, Z. M. (2008). Kernel discriminant analysis with weighted schemes and its application to face recognition. *Machine Learning and Cybernetics, 2008 International Conference on*, 1:448–453, doi:10.1109/ICMLC.2008.4620447. 52, 54