Kirill Sidorov

Groupwise Non-rigid Registration for

AUTOMATIC CONSTRUCTION OF APPEARANCE MODELS

of the Human Craniofacial Complex
for Analysis, Synthesis, and Simulation

PhD Thesis

Cardiff University ◇ 2010

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Dedicated to my parents and grandparents for their endless support.

Abstract

In this thesis, the problem of automatic construction of statistical appearance models from examples is considered. The key step in statistical appearance modelling is establishing spatial correspondences between examples in order that statistics on the corresponding features may be computed. This is known as registration.

Groupwise registration methods, which aim to consider useful information from the entire ensemble at once when searching for correspondences, have been shown in the literature to be superior to pairwise methods. However, the groupwise approach to registration is generally computationally expensive due to the large dimensionality of the search space in which the globally optimal solution is searched.

A novel, fast and reliable, stochastic algorithm is proposed to solve the problem of groupwise non-rigid registration of large ensembles of images quickly and more accurately than state of the art methods. The efficiency of the proposed approach stems from novel dimensionality reduction techniques specific to the problem of groupwise image registration and from comparative insensitivity of the adopted optimisation scheme (Simultaneous Perturbation Stochastic Approximation (SPSA)) to the high dimensionality of the search space.

The proposed image registration algorithm is then generalised to the case of textured 3D surfaces, allowing groupwise non-rigid registration of 3D data, such as produced by widely available 3D surface scanners.

In evaluation of these approaches we show a high robustness and success rate, fast convergence on various types of test data, including facial images featuring large degrees of both inter- and intra-person variation, and show considerable improvement in terms of accuracy of solution and speed compared to traditional methods.

Finally, a novel application of 3D appearance modelling is proposed: a faster than real-time algorithm for statistically constrained quasi-mechanical simulation. Experiments demonstrate superior realism, achieved in the proposed method by employing statistical appearance models to drive the simulation, in comparison with the comparable state of the art quasi-mechanical approaches.

Acknowledgements

I AM greatly indebted to my principal supervisor, Prof. David Marshall, for creating the research environment that made this work possible, and without whom one would not conceive of undertaking such a challenging and interesting project.

I am also very grateful to the Cardiff University School of Dentistry and particularly to my second supervisor, Prof. Stephen Richmond, for generous funding of this research, his great sense of humour, as well as courage and patience when dealing with us, computer scientists.

I would also like to express my gratitude to the School of Computer Science for administrative and technical support. To Mike Daley for keeping all our equipment up and running and for his unique ability to provide us with virtually any instrument, tool, device, contraption, gadget, no matter how bizarre. To Robert Evans for his transcendental dedication to running and maintaining all the essential computing infrastructure in the most reliable and robust manner.

I thank Dr. Julia Hicks for helpful advice and encouragement, as well as many fruitful discussions over the past few years, very refreshing and intellectually stimulating.

I would also like to thank everyone in the Geometric Computing and Computer Vision group, and all the members of the *vlunch* seminar, particularly Dr. Paul Rosin and Prof. Ralph Martin, for their motivating input, bold ideas, comments and valuable feedback on my work.

Finally, I thank my parents and friends for their patience and unconditional support throughout.

Contents

List of Figures

List of Tables

List of Algorithms

Notation

> Proof by cumbersome notation — best done with access to at least four alphabets and special symbols.
>
> Folklore

Table 1 summarises the notational conventions used in this thesis for various types of quantities and variables. Table 2 summarises frequently used symbols, operators, functions and notational devices. All vectors are assumed to be column vectors, unless otherwise specified.

TABLE 1: Notational convention for quantities.

Alphabet	Quantity	Description
α, N, p, ...	Scalar quantities (constants, variables, functions).	Lower or upper case, italic font, Latin or Greek letters.
$\mathbf{x}$, $\mathbf{f}$, $\boldsymbol{\varphi}$, ...	Vector quantities (constants, variables, functions).	Lower case, bold font, Latin or Greek letters.
M, R, X, ...	Matrices.	Upper case, upright font, Latin letters.
$\mathcal{F}$, $\mathcal{E}$, $\mathcal{B}$, ...	Vector-valued matrices, vector fields, and tensors.	Upper case, bold font, calligraphic Latin letters.
$\mathbb{R}$, $\mathbb{Z}$, $\mathbb{N}$, ...	Number sets.	"Blackboard bold" upper case Latin letters.

TABLE 2: Commonly used symbols, operators, functions, notational devices.

Notation	Description
M^{-1}	Inverse of a matrix M.
M^{T}	Transpose of a matrix M.
$\mathrm{M}_{n\times k}$	Matrix of n rows and k columns.
$\mathbf{1}$	Matrix (or vector) of ones, of appropriate dimensions.
$\mathbf{0}$	Matrix (or vector) of zeros, of appropriate dimensions.
$\mathbf{1}_{n\times k}$	Matrix of n rows and k columns, filled with ones.
$\mathbf{0}_{n\times k}$	Matrix of n rows and k columns, filled with zeros.
$\mathrm{I}_{n\times n}$ or $\operatorname{diag}\mathbf{1}_{n\times 1}$	The identity matrix of n rows and n columns. The $\operatorname{diag}\mathbf{1}_{n\times 1}$ notation is preferred to "$\mathrm{I}_{n\times n}$" where confusion with image matrices is possible.
$\mathrm{A} \leftarrow \mathrm{B}$	In algorithms, assignment of the value of variable B to variable A.
$\{\mathrm{R}, \mathrm{S}\} \leftarrow \mathrm{poldec(X)}$	Polar decomposition (see Lorusso *et al.* [168]) of matrix X into R and S.
$\mathbb{R}^n$	n-dimensional Euclidean space.
$\mathrm{M} \in \mathbb{R}^{n\times k}$	An alternative way to indicate that matrix M has n rows and k columns, when using subscripts is undesirable, *e.g.* when other subscripts are present.
$\mathbf{x}\mathbf{1}_{1\times k}$	Matrix of k columns, in which every column is the (column) vector $\mathbf{x}$, *i.e.* repeat vector $\mathbf{x}$ "horizontally" k times. This is equivalent to MATLAB expression `repmat(x, 1, k)`.
$\underset{n\times k}{\text{reshape}}\,\mathrm{M}$	Denotes a matrix of n rows and k columns, whose elements are taken column-wise from M (which contains a total of $n \times k$ elements). This is equivalent to MATLAB expression `reshape(M, n, k)`.

continued on the next page...

TABLE 2: Commonly used symbols, operators, functions, and notational devices.

... continued from the previous page

Notation	Description
$\operatorname{diag}\mathbf{x}$	If $\mathbf{x}$ is a vector of n components ($\mathbf{x} \in \mathbb{R}^n$), this expression denotes a square matrix with elements of $\mathbf{x}$ on the main diagonal. This is equivalent to MATLAB expression `diag(x)`.
$\operatorname{diag}\mathrm{M}_{n\times n}$	Denotes a vector of n components, taken from the main diagonal of matrix M. his is equivalent to MATLAB expression `diag(M)`.
$expr(v_1, \ldots, v_n) \rightarrow \min_{v_1,\ldots,v_n}$	Denotes that the espression *expr* is to be minimised with respect to variables v_1, ..., v_n.
$\dim\mathbf{x}$	Number of elements in vector $\mathbf{x}$. This is equivalent to MATLAB expression `numel(x)`.
$\mathrm{M}(i,j)$	Denotes the element on the i-th row and j-th column of matrix M. Enumeration of rows and columns is 1-based. This notation is preferred over subscripting ("M_{ij}") in this thesis.
$\mathbf{x}(i)$	Denotes the i-th element of vector $\mathbf{x}$. Enumeration of elements is 1-based. This notation is preferred over subscripting ("$\mathbf{x}_i$") in this thesis.
$\mathrm{A}_{m\times k} \bullet \mathrm{B}_{n\times k}$	Hadamard product of matrices. For any two matrices A and B of the same size, $(\mathrm{A}_{n\times k} \bullet \mathrm{B}_{n\times k})(i,j) = \mathrm{A}(i,j)\mathrm{B}(i,j)$, *i.e.* element-wise multiplication.
$\mathrm{A}_{\bullet}/\mathrm{B}$	Hadamard division of matrices. For any two matrices A and B of the same size, $(\mathrm{A}_{n\times k\bullet}/\mathrm{B}_{n\times k})(i,j) = \mathrm{A}(i,j)/\mathrm{B}(i,j)$, *i.e.* element-wise division.
$_{\bullet}\sqrt{\mathrm{M}}$	Element-wise square root: $\left({}_{\bullet}\sqrt{\mathrm{M}}\right)(x,y) = \sqrt{\mathrm{M}(x,y)}$.
$\mathrm{A}_{m\times k} * \mathrm{B}_{(2n+1)\times(2n+1)}$	Discrete 2D convolution of matrices A and B: $(\mathrm{A}_{m\times k} * \mathrm{B}_{(2n+1)\times(2n+1)})(x,y) = \sum_{i=-n}^{i=n}\sum_{j=-n}^{j=n} \mathrm{B}(i+n+1, j+n+1)\mathrm{A}(x-i, y-j)$. Values outside A are treated as zeros.

continued on the next page...

TABLE 2: Commonly used symbols, operators, functions, and notational devices.

... continued from the previous page

Notation	Description
$\mathrm{A}_{m\times k} \circledast \mathrm{B}_{(2n+1)\times(2n+1)}$	Same as above, but values outside A are taken from the nearest "border" value. This is equivalent to MATLAB expression `imfilter(A, B, 'same', 'replicate', 'conv')`.
$\mathrm{M}(:,\cdot)$, $\mathrm{M}(\cdot,:)$, *etc.*	The colon notation semancically equivalent to that in MATLAB.
G_w^σ	Gaussian convolution kernel with SD= σ, stored in a w-by-w matrix. It is normalised so that $\sum_{i=1}^{i=w}\sum_{j=1}^{j=w}\mathrm{G}_w^\sigma(i,j)=1$.
$\nabla\mathrm{A}$	For matrices, $\{\mathrm{G}_x,\mathrm{G}_y\} = \mathcal{G} = \nabla\mathrm{A}$ denotes numerical approximation of the gradient.
$\min\mathrm{M}$, $\max\mathrm{M}$	When applied to a matrix or a vector, denotes the smalest (largest) element of the matrix or vector.
$_\bullet\|\mathcal{A}\|$	For vector-valued matrices the "dotted norm" operator denotes the element-wise norm: $_\bullet\|\mathcal{A}\|(i,j) = \|\mathcal{A}(i,j,:)\|$, $\forall i,j$.
$\mathrm{rescale}_a^b\,\mathrm{M}$	Rescale the data in matrix M so that all values lie in the interval $[a,b]$. $\mathrm{rescale}_a^b\,\mathrm{M} = a + \dfrac{(b-a)(\mathrm{M}-\mathbf{1}\min\mathrm{M})}{\max\mathrm{M}-\min\mathrm{M}}$.
$\lfloor x \rfloor$	The floor function.
$\mathrm{randn}(1\ldots m)$	A uinformly distributed natural radom number between 1 and m inclusively.

CHAPTER

1 Introduction

Poekhali! (Let's go!)

Yuri Gagarin

THIS thesis is a study of methods for automatic construction of models of craniofacial[1] appearance and dynamics, for purposes of analysis, synthesis, and simulation. The search for novel ways of modelling the human craniofacial complex *automatically*, in an unsupervised fashion, is inspired by a number of important applications in the orthodontic practice (Sidorov *et al.* [241], Kau *et al.* [140], Popat *et al.* [215], Popat and Richmond [216], Beldie *et al.* [18]), computer graphics (Blanz and Vetter [28]), psychology (Cosker *et al.* [69]), biometrics and security (Benedikt [20]), and other fields.

The problem of accurately representing craniofacial appearance and its temporal evolution is a very challenging one, considering the complexity, diversity and variability of the geometry, topology, texture, reflectivity, self-shadowing and other properties of the human head.

The dynamic nature of the craniofacial complex makes its study from the computational point of view a fertile field of research, comprising such areas as computer vision, image processing, statistical modelling, computational

[1]The term *craniofacial*, from Mediæval Latin *crānium* ("skull"), refers to the head or skull and the facial structures together.

geometry, non-rigid deformable object modelling, rheology, finite element (FE) modelling and simulation, biometrics and facial identification, forensics, facial expression recognition and facial tracking, visual speech analysis, emotion psychology, computer graphics (CG) animation and many others.

In this thesis, the crucial component of appearance modelling is considered — computing correspondences between examples, images or surfaces, in order that deformable models of appearance may be built.

1.1 Motivation

The work described in this thesis was inspired by the need to develop novel mathematical and engineering methods, models, and algorithms to capture, represent, manipulate and simulate craniofacial appearance, kinematics and dynamics. The motivation for this research comes from practising craniofacial surgeons and orthodontists who seek to employ the advances in the fields of computer vision and computational geometry to facilitate and automate the analysis of growth and development of the craniofacial complex, to study its abnormalities, to model and study variation in its appearance, and to simulate its kinematics and dynamics for interactive surgery simulation and teaching scenarios. Although the methods, models and algorithms presented in this work are very generic and applicable to a wide variety of imagery, in order to to illustrate the applicability of the proposed methods to problems of orthodontics and craniofacial surgery, the results are exemplified and the experiments are conducted mostly with craniofacial imagery, which also happens to be a characteristic example of "difficult" data, due to its inherent enormous variability.

The development of affordable non-invasive surface scanners (*e.g.* [6]) capable of capturing the shape and appearance of objects at video rate ("4D cameras") has recently provided the researchers in computer vision with a novel and very valuable source of data. These instruments and the previously unattainable data that they provide, in turn, have opened the doors to a flood of new ideas, inspired much interdisciplinary research, found a number of clever applications, scientific, industrial, and medical, as well as caused the revival of some of the familiar established techniques.

Methods for automatic construction of craniofacial appearance models will in the future underpin much of the orthodontics research, including realistic modelling of influences of orthodontic treatment and injuries, predicting the effects of ageing, simulation of surgical intervention, post-surgical evolution of a patient's appearance, optimisation of existing surgical procedures.

Outside of medicine, craniofacial appearance modelling also has a number of important applications. One is motion analysis, including extraction of high-level features (expression recognition) for purposes of biometrics and security (recognition and identification, see Benedikt [20]). Another is advanced user interfaces, which include facial expression driven control, lip-motion tracking to potentially augment existing speech recognition engines. These are concerned with facilitation of a man-machine interaction through reducing the need to resort to traditional input devices and manipulators, such as keyboard or mouse. Model-based video compression (Toelg and Poggio [271]) can be used to transmit videos through very low-bandwidth channels. This can be useful in video telephony if instead of transmitting a video stream only parsimonious model parameters are transmitted and used on the receiving side to synthesise the animated appearance of a person from a generative model. Synthesis of highly realistic faces via a generative model of appearance in computer graphics applications has commercial potential in video games and film industry.

Computer vision has been revolutionised by model-based methods which have originated from the early 80's. Instead of relying on some analytical — algebraic, algorithmic or some other — description of objects, contemporary model-based approaches are capable of describing appearance, properties or features of objects in a parsimonious model which is learnt directly from images of the objects (Gonzalez-Mora *et al.* [110]).

Currently, craniofacial modelling and animation are some of the most important domains in computer vision and graphics. In recent years, the major advances in digital imaging, both in acquisition and in processing, made it possible for the human craniofacial complex to be studied in much greater detail than has previously been possible. This study, for example, takes advantage of the recent progress in the 3D scanning technology to gather data from which to build computerised 3D models of human heads. With the advent of fast accurate 3D scanners [6] that provide non-invasive sequential capture capabilities at video rate, it is possible to collect large amounts of real geometrical and

colour data, which can be used to automatically build computerised models of appearance, kinematics and dynamics. Previously available data acquisition methods prevented research into craniofacial modelling that requires video-rate 3D data. Much of the relevant literature describes research using either static 2D images or short 2D sequences, and more recently — static 3D scans. Scarcity of publications taking advantage of these new data sources suggests that great potential exists now in the study of new methods, models and algorithms applicable to more advanced modelling of craniofacial appearance.

Due to the fact that statistical modelling allows for *automatic* construction of models (as opposed to hand-crafted ones which are inevitably inflexible and often inaccurate) describing real-world data, adaptable statistical methods draw more and more interest from the computer graphics and vision community.

While many kinds of appearance models have been proposed over the past two decades, the question of *automatic* construction of good models (accurate, with high specificity and generalisation ability) has only been raised recently.

The key step in statistical appearance modelling is establishing spatial correspondences between examples in order that statistics on the corresponding features may be computed — a process known as *registration*. This thesis discusses the problem of groupwise non-rigid image registration, and its generalisation — groupwise non-rigid registration of surfaces, for the purpose of appearance modelling.

1.2 Main Contributions

The main contributions of this thesis are:

- A novel efficient stochastic algorithm for groupwise non-rigid registration of images is presented. The proposed algorithm is shown to register sizeable image ensembles quickly and more accurately than state of the art methods. Experiments demonstrate the reliability of the proposed approach on data with very high variability, in particular pioneering the notoriously difficult case of inter-subject registration. See also Sidorov *et al.* [243].
- A generalisation of the above algorithm to the case of textured 3D surfaces. The proposed 3D registration algorithm retains all the desirable properties of the above 2D algorithm and allows for groupwise non-rigid

registration of 3D surfaces in a principled way. This opens new research prospects by allowing a new valuable source of data to be leveraged: textured 3D surfaces produced by video-rate surface scanners which have recently gained popularity. See also Sidorov *et al.* [241, 242, 244].

- To show the usefulness of the proposed registration framework in appearance model building, a novel application of statistical appearance modelling is presented: a faster than real-time quasi-mechanical simulator of deformable objects using statistical constraints. Experiments demonstrate the entire pipeline from acquisition, registration and model building, to physically realistic real-time simulation of deformable objects.

1.3 List of Relevant Publications

- K. SIDOROV, S. RICHMOND, D. MARSHALL. An Efficient Stochastic Approach to Groupwise Non-rigid Image Registration. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec. (CVPR '09)*, pages 2208–2213. IEEE Computer Society, Los Alamitos, CA, USA, 2009.
- K. SIDOROV, S. RICHMOND, D. MARSHALL. Efficient Groupwise Non-rigid Registration of Textured Surfaces. Accepted to *IEEE Conf. on Comp. Vis. and Pat. Rec. (CVPR 2011)*, 2011.
- K. SIDOROV, D. MARSHALL, S. RICHMOND. Nonrigid Image Registration Using Groupwise Methods. In C. H. Kau and S. Richmond (editors), *Three-Dimensional Imaging for Orthodontics and Maxillofacial Surgery*, pages 290–304. Blackwell Publishing Ltd., 2010. ISBN 9781405162401.
- K. A. SIDOROV, A. D. MARSHALL, P. L. ROSIN, S. RICHMOND. Towards Efficient 3D facial Appearance Models. In D. Metax, J. Popovic (editors), *ACM SIGGRAPH Symposium on Computer Animation*, 2007.
- K. A. SIDOROV, S. RICHMOND, A. D. MARSHALL. Statistically Constrained Real-time Meshless Simulation. *(in preparation)*, 2010.

1.4 Organisation of the Thesis

The rest of the thesis is organised as follows.

- Chapter 3 describes the proposed framework for groupwise non-rigid registration of ensembles of 2D images. A novel robust and efficient algorithm is proposed that is capable of rapidly establishing correspondences between a large set of images in a reliable and unsupervised fashion.
- Chapter 4 extends the findings of chapter Chapter 3 and generalises the proposed registration framework to registration of 3D surfaces.
- Chapter 5 discusses a novel approach to pseudo-mechanical simulation, based on employing statistical models, such as those build using the framework of Chapter 4, to provide a very computationally cheap, faster than real-time, unconditionally stable simulation of deformable objects.
- Chapter 6 summarises the contributions and findings of this study and establishes the foundation for potential future research.

CHAPTER

2 Background

> Before attempting to create something new, it is vital to have a good appreciation of everything that already exists in this field.
>
> Mikhail Kalashnikov

THE purpose of this chapter is to place the research described in this thesis into the broader context of the study of craniofacial modelling, including the problem of image registration and simulation methods. This thesis brings together and builds upon many ideas from computer vision, statistics, computational geometry and even computational mechanics. An all encompassing review of the literature related to these topics is therefore made impossible not only by the vast amount of material accumulated over the years in each of these fields, but also by the depth and breadth of these problems.

There are two main classes of approaches to craniofacial modelling. Approaches of the first class are concerned with modelling the craniofacial mechanics: measuring and modelling the material properties, construction of constitutive mechanical models, and physical simulation. While these approaches are perfectly valid and have found many applications, they are outside the scope of this thesis and will only be briefly reviewed below, for context.

Approaches of the second class are concerned with modelling the appearance, typically in a statistical framework, using only example imagery of the object

as the input. These approaches play a very important role in computer vision and are the main focus of this thesis. More precisely, this thesis focuses not on the modelling itself, but on the key step required in construction of appearance models — establishing correspondences between samples of a deformable object, or registration. This is essential, because unsupervised modelling typically involves analysis of multiple examples, and a meaningful analysis is only possible if the spatial correspondences between examples are known.

A brief overview of mechanical modelling, followed by the discussion of properties of craniofacial appearance, introduction to registration and a summary of appearance modelling techniques are given below.

2.1 Mechanical Modelling of the Craniofacial Complex

Existing methods of measuring the mechanical properties (Young's modulus, Poisson's ratio, Lamé parameters *etc.*) of living tissue are invasive to the degree of being incompatible with the life of the subject (for example, dissection of the head into small pieces before their Young's moduli can be measured with a dynamometer). Such measurements are indeed being made with dissected human cadavers and other animals (pigs), which allows for development of generalised, atlas-like mechanical models (Beldie *et al.* [18]). However, the latter are of little use in medical practice where accurate subject-specific models are required — in humans, the inter-subject variability in the craniofacial proportions, in the character of fat deposits, and even the layout of muscle tissue and ligaments is enormous (Wilkinson *et al.* [294]).

Modern non-invasive diagnostic tools, such as computed axial tomography (CAT), magnetic resonance imaging (MRI) and other scanning techniques are still of very limited use where automated measurement of tissue properties, and especially the analysis of internal organisation (muscle attachment points, fibre directions *etc.*) of the components of the head is required. Even if it was not the case, practical difficulties would still exist in so measuring mechanical properties of human soft tissue in *bulk quantities* by means of some robust automated process, such as for conducting large-population latitudinal studies.

Furthermore, the knowledge of mechanical properties of human soft tissues is still very limited. In particular, the behaviour of tissues in the presence of

large deformations is poorly understood, as is the evolution of their mechanical properties with time.

Despite acute interest in such mechanical craniofacial models and vast theoretical support from the related fields of materials science, computational mechanics, rheology *etc.*, the history of mechanical craniofacial modelling over the past 30 years has yielded poor results.

Hand-crafting of even a generic mechanical model of the human craniofacial complex, suitable for simulation with a degree of realism and detail sufficiently high for it to be useful in medical practice and research, is an extremely challenging task, if not outright impossible. Simplification of the anatomy represented by such hand-crafted models and numerous assumptions that will have to be made about the properties of tissues in order to make the problem tractable lead to insufficient degree of realism in simulation, despite the huge amount of effort invested in building the models.

The pioneering work in modelling deformable surfaces and solids, particularly in the domain of computer graphics, has been carried out by Terzopoulos *et al.* [264], which arose wide interest in new efficient ways of mechanical modelling and simulation within the CG community and stimulated further research in accurate representation of material properties, questions of stability in mechanical simulations and modelling advanced mechanical phenomena (such as plasticity, fractures, viscosity etc.) Terzopoulos *et al.* [264] were first to discuss practical (from the CG point of view) ways of discretisation of the motion equations using finite difference approximations which produce a linked system of ordinary differential equations (ODEs) and efficient ways to numerically integrate such system over time. They foretold the increasing importance of physically-based modelling of non-rigid curves, surfaces and solids with properties similar to that of elastic materials in computer graphics applications by demonstrating the computational tractability of that problem. A number of attempts were made to drive a head model by simulating the mechanical processes (e.g. muscle activations and contractions) thus providing a biophysically meaningful basis for such models. One of the earliest such studies is presented in Waters [292], where modelling of simple muscle contraction process suitable for generation of several varied facial expressions, controlled by a limited number of parameters, is presented.

FIGURE 2.1: Sifakis and Fedkiw [245] muscle structure and simulation mesh and their deformable model fit to motion capture input (red and green are simulated and captured markers respectively).

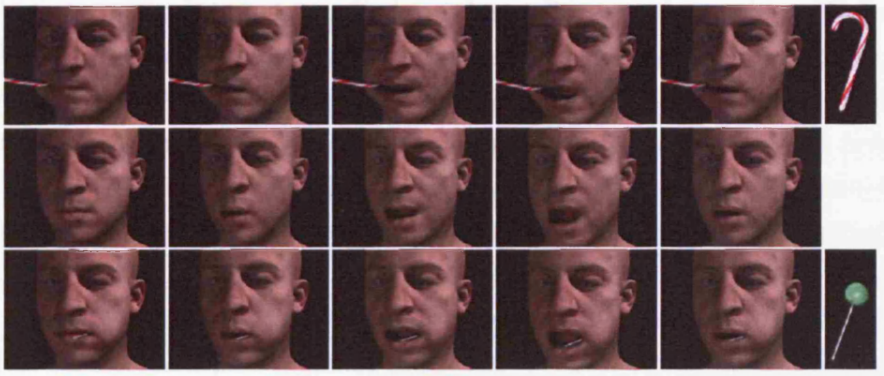


FIGURE 2.2: Illustration of the results by Sifakis *et al.* [246]. Middle row: synthesis of facial expression from speech. Top and bottom rows: simulation augmented with interaction of the model with external objects.

Some of the more recent advances include Sifakis and Fedkiw [245], see, where a finite element model simulation (configured by sampling deformations of the face surface over time) is used to determine the facial expression resulting from muscle activations driving the associated rigid bones. Their model is illustrated in Fig. 2.1. While simulating only 32 muscles of the face, this model is extremely computationally expensive (requires 840×10^3 simplexes, approximately 8 minutes per frame) and at the present time uses the quasistatic (*i.e.* assuming equilibrium of forces at any given time) approximation to the solution. Such models, however, can be trivially driven by various data sources; for example, success is reported in speech simulation using such models, see Sifakis *et al.* [246], also see Fig. 2.2. In reality, fast and accurate simulation of elastic solids is still an open problem with a vast field of applications. Teran *et al.* [263] present a novel quasistatic algorithm that alleviates geometric and material indefiniteness allowing one to use fast conjugate gradient solvers during Newton-Raphson iteration.

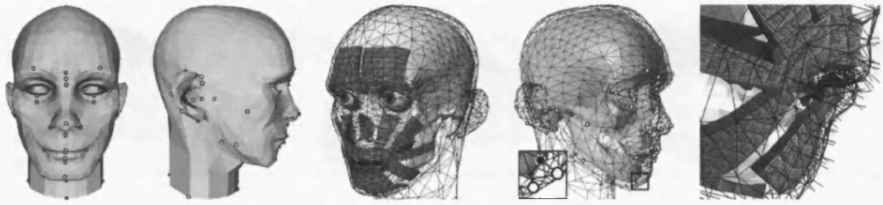


FIGURE 2.3: Model of Kähler *et al.* [138]. From left to right: head geometry with landmarks, front and side views, skull and facial components, skull landmarks related to subset of skin landmarks, detailed view.

A real time solution to realistic and non-linear deformations of elastic bodies is presented by Allen *et al.* [7], where a model of human body shape variation, learnt from a corpus of 3D range scans, is used to capture both identity-dependent and pose-dependent shape variation in a correlated fashion.

Another anthropometrically meaningful model with anatomical structure capable of real-time physics-based simulation and animation is presented by Kähler *et al.* in [138]. Their model is deformable through landmark data and adapts the underlying muscle and bone structure to match the deformed model. Their model (see Fig. 2.3) comprises a skin surface (approximated by a triangle mesh), a set of 24 virtual muscles capable of contracting in linear or circular fashion, a solid skull with rotatable mandible and a mass-spring model connecting skin, muscles and skull. In addition, they have experimented with fitting their model to imperfect scan data and also with simulation of head deformations due to ageing.

The problem of mechanical modelling of the craniofacial complex has been attempted by Teschner *et al.* [266, 267] with limited success. They present a simple system for interactive craniofacial surgery simulation, in particular of osteotomies of the facial and skull bones and for prediction of soft-tissue changes caused by bone movement. The system utilises radiometric data (CAT scanning) as well as pre-operative appearance data obtained from a laser scanner. Their system uses a simplistic elastic spring model to represent the mechanical properties of the multi-layer soft tissue. The model also attempts to represent additional features such as skin turgor, gravity and sliding bone contact.

In addition, a significant effort has been made in the recent years at INRIA (Epidaure Group) related to mechanical simulation of soft tissues for interactive

FIGURE 2.4: Laparoscopic surgery simulation (Picinbono *et al.* [210]).

surgery and other medical simulations. They have, in particular, developed a minimally invasive hepatic surgery simulator prototype (see Fig. 2.4). A vast list of publications resulted from this project includes Cotin and Delingette [70], Picinbono *et al.* [210, 211] and others. They are addressing two problems of virtual surgery simulation: first, the geometrical and physical model of the human organs must be very realistic; second, the model and simulation methods must be sufficiently efficient to allow for real-time simulations. They find a compromise in a novel deformable model based on non-linear elasticity and FE simulation. They utilise non-linear tensor-mass model. Stiffness tensors are pre-computed before the simulation. During the simulation, forces for each vertex, edge, face and tetrahedron are computed from the model state and are used to find the vertex positions in the next iteration. Like Teschner *et al.* [268], they use explicit integration scheme to compute vertex positions from elastic forces.

It is worth mentioning a dynamic simulation framework for topology-changing deformable material, presented in Gissler [107]. Their model is based on corotational FE approach for linear elasticity and plasticity, with geometric constraints. Topology changes that are modelled in the paper comprise fracturing and merging of deformable objects.

Koch and Bosshard [149] proposes a system for synthesis of facial expressions through superposition of facial actions in real-time. Notably, their approach utilises biometric data for the FE simulation and takes into account facial anatomy when defining muscle groups.

The challenges of the mechanical approach to craniofacial modelling lead us to search for the solution in the field of computer vision. Some of the existing computer vision techniques proved to be highly useful tools for modelling of appearance (see *e.g.* Section 2.5) and great potential for further development

FIGURE 2.5: Anatomical structures of the face can undergo highly non-rigid deformations when expressing emotions and cognitive state.

exists, as indicated by the strong interest of the scientific community in such methods. Looking at the problem from the computer vision point of view allows for better utilisation of available data sources, such as 3D scanners. Many of the modelling techniques in computer vision have a wider range of applications, for example in computer graphics, and so are not limited to medical problems.

2.2 Appearance of the Face

The face is the frontal part of the human craniofacial complex. It houses important sensory organs, the exterior part of the speech-production apparatus, and the entrance to the alimentary canal. The face is the most individual part of the human body and its appearance exhibits enormous inter-personal variability (faces convey identity) as well as temporal intra-personal variability: faces deform to convey emotions (Fig. 2.5), during speech production, and mastication.

In more detail, the variability of the craniofacial appearance is discussed by Pantic in [197], where a classification of "facial signals" is proposed. For example, relatively permanent features of the face, such as overall proportions or the layout of the fat tissues, are *static signals*, at least within the subject, but may significantly vary between subjects. Slow evolution of the facial appearance, such as development of wrinkles due to ageing, is classified in Pantic [197] as *slow signals*, and such signals are of significance in longitudinal studies. Noticeable changes in facial appearance due do neuromuscular activity, such as speech, expression of emotions and blushing, are called *rapid signals*. The human face owes its vast repertoire of possible deformations to the complexity of the underlying musculature, Fig. 2.6. In addition to the above time-varying

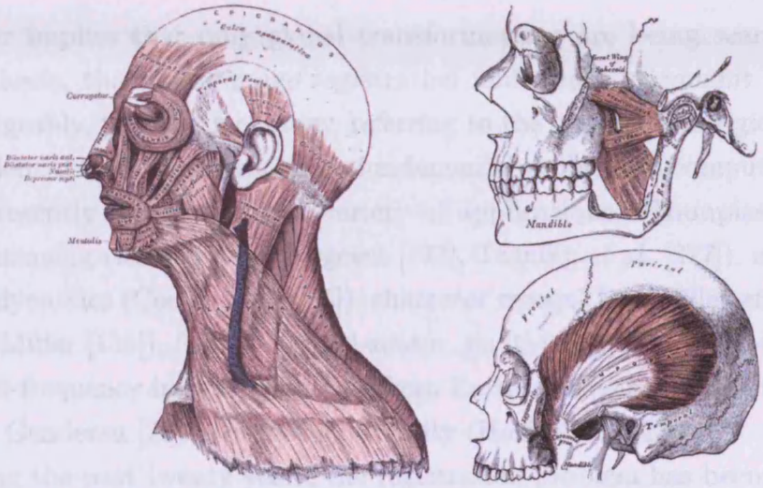


FIGURE 2.6: Muscles of head, face, and neck. Image from Gray's Anatomy [112].

signals, Pantic [197] also considers *artificial signals*. They include varying exogenous features of the face, such as make-up or glasses.

In order to study the variation in appearance of the face, one must learn to extract and aggregate useful information from *ensembles* of facial images that show the variation of the face. This topic is discussed next.

2.3 Image Registration

In many imaging and computer-vision problems, it is often the case that important information is contained in more than one image. To properly extract and integrate the valuable information from an ensemble of complementary images, a procedure called *image registration* is employed (Fischer and Modersitzki [98]).

Image registration is a process of computing the spatial transformations that bring two or more images into correspondence, so that the analogous features match. In the literature, image registration is occasionally referred to as *spatial normalisation*[1], particularly in the field of medical imaging (Park *et al.* [198]), *the correspondence problem*, or *image alignment*, though more frequently

[1]More correctly, spatial normalisation includes first registration, to find the transformations between images, and then actually applying these transformations to warp the images.

the latter implies that only global transformations are being searched for. In this thesis, the terms *image registration* and *image alignment* are used interchangeably, to avoid monotony, referring to the general, non-rigid, case of the problem. Image registration is a fundamental problem in computer vision and is presently being used in a variety of applications. Examples include: medical imaging (Maintz and Viergever [173], Twining *et al.* [277]), modelling of facial dynamics (Cootes *et al.* [66]), character recognition (Miller *et al.* [182], Learned-Miller [156]), fusion of multi-sensor, multi-resolution, multi-temporal and multi-frequency imagery obtained from Earth observation satellites (Pohl and Van Genderen [214]), augmented reality (Hoff *et al.* [128]).

During the past twenty years, the registration problem has been drawing increasingly more attention from researchers, as the increase in the available raw computing power made practical solutions to the problem a possibility, as well as opened a number of important applications.

So far, a general theory or a unified treatment for all aspects of the registration problem has not yet been established; and so, over the years a vast range of techniques have been developed for registration of various kinds of imagery and for various applications (Fischer and Modersitzki [98]).

While humans possess the remarkable ability for accurate and fast registration of images (including solving the stereopsis problem all the time in real time!), teaching computers the same skill turns out to be a very difficult task. Incidentally, human capacity for image registration sets the upper bound on the complexity of the problem: it is definitely soluble.

2.3.1 Taxonomy of Image Registration

Many of the vast number of registration techniques can be categorised according to the important properties of the algorithm used, and according to the nature of the imagery to which they are applicable. A crude classification is given below. A more detailed taxonomy and review can be found in Zitova and Flusser [303].

BY SEPARATION. Depending on the application domain, images being registered can be spatially or temporally separated, or both, and can be acquired with sensors of different modalities (Zitova and Flusser [303]). Spatial separation of images can be due to the relative motion of the camera and the scene,

or due to the acquisition being performed with multiple spatially separated cameras (Hartley and Zisserman [124]).

Examples of registration of images that are spatially separated include: solving the stereopsis problem (shape from stereo), which involves establishing dense correspondence between a pair of stereo images to compute the stereo disparities and, therefore, depth (Ogale and Aloimonos [195], Scharstein and Szeliski [230], Moravec *et al.* [185]); stitching panoramic or high-resolution images together out of several smaller images (Szeliski [259], Noirfalise *et al.* [192]), and similarly surfaces[2] (Levoy *et al.* [161], Curless and Levoy [75]).

Examples of the registration of temporally separated images include: medical imaging (especially longitudinal studies for change detection and quantification (Leow *et al.* [160]), monitoring of tumour development (Angelini *et al.* [10]), study of the effects of ageing on the craniofacial complex (Andresen *et al.* [9]), computer-aided surgery (Archip [11]), and others); surveillance and biometrics (Benedikt [20]); statistical appearance modelling (Cootes *et al.* [66]) for animation (Cosker [68]), model-based video compression (Toegl and Poggio [271]), and even study of emotion psychology (Cosker *et al.* [69]).

In any case, the scenes in the images to be registered undergo some kind of evolution, in time or space, including substitution of the subject (for example, inter-subject registration of facial images of several people in a latitudinal study).

BY INTERACTIVITY. Image registration can be performed in a manual, a semi-automatic, or an automatic way. Tedious manual registration is sometimes used in medical imaging, in applications for which automatic registration is not yet feasible. While manual annotation of 2D images is possible, despite being a laborious task, annotating higher-dimensional imagery is usually impractical. Furthermore, manual annotation leads to suboptimal models, in addition to being prone to subjective biases, as shown in Davies *et al.* [78]. Semi-automatic approaches typically involve a less laborious manual bootstrapping stage, such as placing a small number of landmarks or labelling the key features, to initialise, guide, or constrain the consequent automatic registration. This thesis is concerned only with the fully automatic methods.

[2]In this thesis, 3D surface scans are regarded just as a special kind of imagery.

BY MODALITY. According to whether a registration method is applicable to the images that were acquired with the same or different type of equipment or probes, the registration methods are classified into *multimodal* and *unimodal*. The various image modalities include: ordinary photographic imaging, Röntgen ray imaging, sonography, radar imagery, CAT, MRI, positron emission tomography (PET) and others. A special case of multi-modal registration is matching 2D images against 3D scenes (Walli and Rhody [288]).

Another special case of image registration is the registration of novel images to some prior model. This includes atlas lookup (for example registering and comparing CAT scans of a human brain with a healthy brain from an anatomical atlas), and even, in a much more general sense, certain types of model-based recognition and image interpretation tasks in computer vision: fitting an Active Appearance Model (AAM) (Cootes *et al.* [61], Cootes and Taylor [59]) or Morphable Model (MM) (Blanz and Vetter [28]) to novel images is an example, regarding AAM to be just a special way (modality) of representing images.

BY DEFORMATION MODEL. Image registration methods are also classified according to the space of admissible spatial deformations that is being searched. In some cases only rigid or affine[3] deformations are considered, such as when mosaicing images (Szeliski [259]), and in other cases a more complex model of deformations is used. The former class models only the global transformations (such as translation, rotation and scaling) which apply to the entire images. The latter broad class, termed *non-rigid* registration, or sometimes *elastic*, or *non-linear* registration, capable of modelling local geometric transformations between images, is of special interest in the context of this thesis. The various deformation models for non-rigid registration will be discussed in detail in Section 3.4.1.

BY FEATURE UNDERSTANDING. According to the way in which the image features are treated, the registration algorithms are classified into *feature-based* and *area-based* (sometimes also called *intensity-based*).

Feature-based methods for image alignment rely on detection and matching of salient features (such as edges, corners, points of high curvature) in the

[3]In the literature, the term *rigid registration*, as opposed to *non-rigid* registration, is often confusingly used to indicate that the affine deformation model is used, not necessarily rigid!

images. After the correspondence between a number of features in the images is found, the sought transformation, which brings images into dense point by point correspondence, is then computed by interpolation between features. For example in Benedikt [20] fiducial points on 3D scans of the face (nasion and the eye cavities) are extracted using Gaussian curvature and mean curvature invariants, and are used to bring a corpus of 3D scans into crude alignment. In Jiang and Yu [133], an interesting algorithm is proposed for simultaneous feature point detection, matching and estimation of global geometrical transformations for tracking of objects in videos. A classical method, Scale-Invariant Feature Transform (SIFT), is used in Péchaud *et al.* [201] to extract key points in vasculature images and use them for non-rigid registration; in Cheng *et al.* [56] a matching method based on Belief Propagation (BP) to improve upon the traditional SIFT-based registration methods is discussed.

In contrast, area-based approaches register the images without first detecting any salient features. Instead, intensity information from the entire images, or large areas thereof, is used to estimate correspondences. These methods rely on some kind of intensity-based objective function that evaluates the quality of alignment given the images and the computed spatial transformations between them. This information is used to drive the search for the solution. The various objective functions used in intensity-based registration will be discussed in Section 3.4.2, and the various approaches to searching for the optimal solution will be discussed in Section 3.4.3.

BY MULTIPLICITY PARADIGM. According to the number of images being registered simultaneously, image registration methods are subdivided into *pairwise* and *groupwise*. Given two images, pairwise registration finds the suitable transformations that bring one of the images, called the *target* (or *template*, or *sensed*) image, in correspondence with the other (*source* image), chosen as the reference. Note that when multiple images are being registered by aligning them one at a time to a single reference image, it still constitutes just the repeated pairwise registration, see discussion in Section 3.2.

In contrast, the groupwise registration, given an ensemble of images, aims to bring all the images into correspondence with each other simultaneously, by using as much as possible of the available information from *all* the images together to guide the registration.

Note that in practical problems, various combinations of the above orthogonal classes of registration may occur. For example, multimodal pairwise non-rigid registration of MRI cardiac scans against a reference CAT scan in O'Donnell *et al.* [194], or unimodal groupwise affine alignment of handwritten digits in Learned-Miller [156].

This thesis is primarily concerned with fully automatic groupwise non-rigid unimodal area-based registration of images (Chapter 3) and surfaces (Chapter 4). The background for non-rigid image registration and the review of literature are found in Section 3.4.

2.4 Dimensionality Reduction Techniques

When dealing with visual data, one is usually given a large set of very low-level features, such as measurements of light intensity[4], taken at sufficiently small spatial intervals — pixels or voxels. As the resolution of imaging increases, and so does the size of images, it becomes necessary to preserve only those attributes of images that are relevant for the task at hand, discarding unnecessary information (Gonzalez-Mora *et al.* [110]). This explains the usefulness of subspace dimensionality reduction methods in modelling imagery, the topic which is discussed next.

Let $\mathrm{I}_{r\times c}$ be an r-by-c grayscale image[5]. It can also be represented as a vector $\mathbf{v} \in \mathbb{R}^{r\times c}$ by scanning the pixels of the image in some predefined order and concatenating them into a vector, written using the notation from Table 2 as $\mathbf{v} = \mathrm{reshape}_{rc\times 1}\,\mathrm{I}$. So, an r-by-c image is a point in rc-dimensional Euclidean space and an ensemble of such images corresponds to a point cloud in this space.

Under the assumption that an ensemble of images is far from being randomly distributed in the $\mathbb{R}^{r\times c}$ space, dimensionality reduction techniques are applied to find a low-dimensional subspace spanned by the images.

Below the "workhorse" of the linear dimensionality reduction techniques, Principal Component Analysis (PCA), ubiquitous in computer vision, will

[4]Other modalities are, of course, possible, such as Röntgen ray imaging, MRI, PET, CAT, sonograms, *etc.*

[5]To simplify notation, it will frequently be convenient to assume the images to be grayscale, stored in matrices, with one real number per pixel describing its intensity. In cases when having multiple components (*e.g.* red, green, blue) per pixel is significant, the vector-valued matrix notation for such multi-channel images will be used.

be reviewed, leading to the explanation of some of the classical appearance modelling techniques which are based on PCA.

2.4.1 Principal Component Analysis

The probability density function (PDF) of a random variable $\mathbf{v} \in \mathbb{R}^n$ that has the multivariate normal distribution is expressed as a multivariate Gaussian:

$$G(\mathbf{x}, \boldsymbol{\mu}, \mathrm{C}_{n\times n}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathrm{C}_{n\times n}|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathrm{C}_{n\times n}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}, \tag{2.1}$$

where n is the dimensionality of the space, $\mathbb{R}^n$, $\mathbf{x}$ is any point in $\mathbb{R}^n$, and the distribution is characterised by its centre $\boldsymbol{\mu}$ and its positive-definite covariance matrix $\mathrm{C}_{n\times n}$.

The term *eigenspace model* or simply *eigenmodel* is sometimes used to refer to such a multidimensional Gaussian distribution (Hicks [126]). Under the assumption that points $\mathbf{x}_i \in \mathbb{R}^n$ have normal multivariate distribution, the eigenmodel can built,

$$\mathfrak{E} = \{\boldsymbol{\mu}, \mathrm{U}_{n\times n}, \mathrm{L}_{n\times n}\}, \tag{2.2}$$

which comprises the origin $\boldsymbol{\mu}$ of the data $\mathbf{x}_i$ in the original space, $\boldsymbol{\mu} \in \mathbb{R}^n$; a spanning basis of the eigenspace, defined by the matrix $\mathrm{U}_{n\times n}$ where columns are orthonormal basis vectors, called *eigenvectors*, coinciding with the axes of the Gaussian; and n *eigenvalues*, specifying the extent of the Gaussian along the corresponding basis vectors, for convenience stored on the main diagonal of matrix L.

Given an *observation matrix* O, the eigenmodel, Eq. (2.2), is built using the discrete Karhunen-Loève transform (KLT), more frequently referred to as PCA in computer vision literature. More correctly, KLT refers to the *transformation* of the observations, O, into the new coordinate space defined by the eigenvectors U in Eq. (2.2).

PCA is a classical technique in statistics, invented as early as 1901 (Pearson [200]). Given an observation matrix $\mathrm{O}_{n\times k}$, with columns corresponding to observations, its mean is expressed as $\boldsymbol{\mu} = \frac{1}{k}\mathrm{O}_{n\times k}\mathbf{1}_{k\times 1}$ and the corresponding

centered observation matrix is therefore $\tilde{\mathrm{O}} = \mathrm{O} - \boldsymbol{\mu}\mathbf{1}_{1\times k}$. The covariance matrix, C, is then

$$\mathrm{C}_{n\times n} = \frac{1}{k}\tilde{\mathrm{O}}\tilde{\mathrm{O}}^T = \frac{1}{k}\left(\mathrm{O} - \frac{1}{k}(\mathrm{O}_{n\times k}\mathbf{1}_{k\times 1})\mathbf{1}_{1\times k}\right)\left(\mathrm{O} - \frac{1}{k}(\mathrm{O}_{n\times k}\mathbf{1}_{k\times 1})\mathbf{1}_{1\times k}\right)^T. \tag{2.3}$$

The new basis, U, is obtained by decomposing $\tilde{\mathrm{O}}$ into $\tilde{\mathrm{O}} = \mathrm{ULU}^{-1}$ ($= \mathrm{ULU}^T$, since U is orthonormal) using eigenvalue decomposition (EVD), the numerical solution to which was first proposed by Кублановская (Kublanovskaya) [305]. Let $\boldsymbol{\lambda}$ be the vector of the eigenvalues: $\boldsymbol{\lambda} = \operatorname{diag}\mathrm{L}$. Assume also that the eigenvalues are sorted by magnitude in descending order, $\boldsymbol{\lambda}(1) \geq \boldsymbol{\lambda}(2) \geq \ldots \geq \boldsymbol{\lambda}(n)$, and the corresponding eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$, constituting the columns of U, are also sorted in the same order. Projection of a vector $\mathbf{x} \in \mathbb{R}^n$, into the eigenspace takes the form

$$\mathbf{p} = \mathrm{U}^T(\mathbf{x} - \boldsymbol{\mu}), \tag{2.4}$$

and since orthonormality of U implies $\mathrm{U}^{-1} = \mathrm{U}^{\mathrm{T}}$, the reverse operation, projection from the eigenspace to the original space, is then

$$\mathbf{x} = \mathrm{U}\mathbf{p} + \boldsymbol{\mu}. \tag{2.5}$$

The Empirical Rule (Ross [225]) states that $\approx 99.7\%$ of the normally distributed data lies within three SDs of the mean. Therefore, a linear combination of the eigenvectors with no eigenvector contributing more than three times the square root of the corresponding eigenvalue is sufficient to represent $\approx 99.7\%$ of the normally distributed data (Hicks [126]). Because of this, PCA can be used to reduce the dimensionality by simply discarding the eigenvectors that do not significantly contribute to the linear combination.

Having sorted the eigenvalues and having rearranged the corresponding eigenvectors (columns of U), one can discard the least significant ones, responsible for the least amount of variation, keeping only m most important basis vectors, $m \leq n$, thus: the reduced basis $\mathrm{U}'_{n\times m} = \mathrm{U}(:, 1 \ldots m)$. Each original data point $\mathbf{x}_i \in \mathbb{R}^n$ can be approximately described by a lower-dimensional parameter vector, $\mathbf{p}' \in \mathbb{R}^m$, by *projecting* it into the reduced dimensionality eigenspace:

$$\mathbf{p}' = \mathrm{U}'^T(\mathbf{x} - \boldsymbol{\mu}), \tag{2.6}$$

and, conversely, the original data points can be approximated using the reduced dimensionality parameters $\mathbf{p}'$, by *unprojecting* them back to the original space:

$$\mathbf{x} \approx \mathrm{U}'\mathbf{p}' + \boldsymbol{\mu}. \tag{2.7}$$

As it is often the case with applying eigenanalysis to ensembles of images, there are usually much fewer samples then there are dimensions in each sample. If each r-by-c image is represented by a vector $\mathbf{x} \in \mathbb{R}^{r \times c}$, with r and c typically a few hundred (pixels) each, and there are n images, yielding $n \ll rc$, the covariance matrix would be of size $rc \times rc$, *i.e.* very large. Since time complexity of finding eigenvectors and eigenvalues requires cubic time, $O((rc)^3)$, in the size of the covariance matrix (see Кублановская (Kublanovskaya) [305]), the direct approach to PCA described above becomes prohibitive. Luckily, as detailed in Cootes and Taylor [59], there is a procedure for computing the eigenvectors and eigenvalues from a much smaller $n \times n$ covariance matrix. Suppose $\tilde{\mathrm{O}}_{rc \times n}$ is a centered observation matrix and $\mathrm{C} = \frac{1}{n}\tilde{\mathrm{O}}\tilde{\mathrm{O}}^T$ is the corresponding covariance matrix, as in Eq. (2.3). Now let T be an n-by-n matrix $\mathrm{T} = \frac{1}{n}\tilde{\mathrm{O}}^T\tilde{\mathrm{O}}$. Then if $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$ are the eigenvectors of T then vectors $\tilde{\mathrm{O}}\mathbf{e}_1, \tilde{\mathrm{O}}\mathbf{e}_2, \ldots, \tilde{\mathrm{O}}\mathbf{e}_n$ are the eigenvectors of C (but they are not necessarily normalised to unit length).

Note, also, that a fast Expectation-Maximisation (EM)-based method for performing PCA without solving the eigenvalue problem has been reported by Roweis in [226]. It relies on probabilistic arguments and allows to very efficiently, in both space and time, compute a small number of eigenvectors and eigenvalues from large sets of data.

Further, methods for incremental PCA and for manipulating eigenspaces are presented by Hall *et al.* in [121–123]. In particular, in Hall *et al.* [121], a novel approach is proposed for merging two eigenmodels, each representing a set of observations, to yield a new model representing the union of these sets. These ideas can be used for making on-line modifications to the existing eigenmodel.

2.5 Statistical Models of Appearance

The early work of Sirovich and Kirby [247] has pioneered modelling of ensembles of images (there, they experimented with face images) using subspace methods (PCA). In Sirovich and Kirby [247], it was remarked for the first time that

the manifold of all faces has a much lower dimensionality than the space of all pictures of the same size (in their particular experiments, the space of all images is $\mathbb{R}^{128\times128}$ and they observe that a linear combination of fewer than 100 eigenpictures is usually sufficient to describe any picture of a face), and they also note that this is even more true if the face images are compartmentalised and a separate eigenmodel is used to model each part (left eye, right eye, nose, mouth, *etc.*). The economy of such compartmentalised model is compared to the way facial features of family members are described ("he has the eyes of his father"). Moreover, they speculate that this is perhaps related to the way the human visual system performs recognition tasks, although no proof is given.

EIGENFACES. Another notable early model of craniofacial appearance, called *eigenfaces*, was proposed by Turk and Pentland in [274]. There, PCA is also applied directly to a corpus of images that are not shape-normalised and the resulting statistical model is used for classification and recognition of faces. The early historical examples of Turk and Pentland [274] as well as Sirovich and Kirby [247] have demonstrated the usefulness of statistical methods for modelling of facial imagery and became a foundation of which later more advanced modelling techniques were devised. A related technique, also for face recognition, but based on Linear Discriminant Analysis (LDA) instead of PCA was proposed in Belhumeur *et al.* [19] and is called *Fisherfaces*.

Facial recognition is indeed an important application of facial modelling. In the seminal papers by Turk and Pentland [274, 275], a system is described that is capable of performing a near real-time location and tracking of subjects' faces, followed by recognition, based on projecting face images onto the feature space (eigenspace) that spans the significant variations in a training set of face images. The recognition is achieved by comparing the low-dimensional parameter vector, as in Eq. (2.6), of the face in question to that of the faces in the database. The term *eigenfaces* was also coined in Turk and Pentland [274]. Note that in Turk and Pentland [274] no registration of face images is performed when building the eigenmodel of the reference set, only a crude rigid alignment. Thus, the per-pixel statistical computations on images are very approximate; this might be sufficient for classification and recognition, but is not good enough for most other purposes. In addition, they make strong assumptions about the orientation of novel faces (faces are upright, frontal view), only scaling of faces is performed when matching the face parameters against the

database. Despite many limitations the original *eigenfaces* approach of Turk and Pentland [274] is still an important early example of modelling face images with dimensionality reducing statistical techniques.

Frequently, when the application at hand is model-based recognition or classification, it is possible to achieve satisfactory performance with a relatively poor model built with only a very rudimentary spatial normalisation of training examples. This is exemplified by the above mentioned work of Turk and Pentland [274], as well as by a more recent work of Chang *et al.* [52–54] and Bowyer *et al.* [37]. The latter works investigate, in particular, the advantages of using a combination of 2D and 3D models of appearance for recognition and show that such approach is superior to using either of the modalities alone.

FISHERFACES. Another approach to the problem of classification, comparable to Turk and Pentland [274] but based on Fisher's Linear Discriminant (FLD) instead of PCA to achieve greater between-class scatter in the low-dimensional projection and, thus, simplify classification, is presented in Belhumeur *et al.* [19]. Their approach is reported to be less sensitive than that of Turk and Pentland [274] to large variation in lighting direction and facial expressions, due to better separation of classes in the low-dimensional space achieved by FLD. The term *Fisherfaces* was also coined in Belhumeur *et al.* [19].

The application of kernel methods for learning low-dimensional representation of faces for the task of recognition is investigated in Yang [296], where two novel methods are proposed, termed Kernel Eigenfaces (KE) and Kernel Fisherfaces (KF), based on the Kernel Principal Component Analysis (KPCA) and Kernel Fisher's Linear Discriminant (KFLD) respectively, which are a generalisation of PCA and FLD in the sense that to find the projection directions they take the higher order correlation of samples into account. The experiments conducted in Yang [296] demonstrate the superior performance of kernel methods in face recognition over the classical approaches in terms of representation of ensembles of images and lower recognition error rates.

2.5.1 Active Appearance Models

A powerful generative method for modelling deformable objects, Active Appearance Models (AAMs) were first proposed by Cootes *et al.* [60, 61] as an extension to the earlier Active Shape Models (ASMs) of Cootes *et al.* [65].

The key idea behind AAMs, as a generative model of images, is to encode in a single *appearance parameter* vector both the pixel intensities (texture) and shape information (spatial deformations) of the images. Both models of shape and texture are linear, but their combination yields a nonlinear model (Kokkinos and Yuille [150]).

In natural time-varying imagery, videos, a lot of energy in pixel variation can be explained in terms of movement of parts of, or of the entire images — this is well known from the video compression literature (Netravali and Robbins [191]). For example, the Moving Picture Experts Group (MPEG) video compression algorithm uses motion compensation, the crudest form of shape normalisation, on small blocks of 8-by-8 or 16-by-16 pixels to parsimoniously explain and encode a significant portion of variation in pixel intensities arising from movement of objects in a video, before encoding the residual variation with a JPEG-like compression. Also, a curious example of this principle is found in Black *et al.* [24] where the change of appearance of a mouth is modelled as a mixture of the learnt motion (optical flow) and "iconic" model (texture variation). They experimentally show that a model based on factoring the changes in pixel intensities into two separate causes — smooth motion of pixels and "iconic" change in pixel intensities — has a much greater representation power.

This property of time-varying imagery, that it is often possible to encode local, relative, motion of parts of the images (such as when an object undergoes some deformation) and the residual appearance variation much more parsimoniously than to encode the appearance variation directly, is exploited AAMs use to compactly represent ensembles of images.

A good overview of AAMs is found in Cootes *et al.* [61] and in even more detail the AAMs are exposed in the ongoing report of Cootes and Taylor [59]. For completeness and to establish the nomenclature, the idea of AAMs is summarised here (for a 2D case, to simplify explanation).

Suppose an ensemble of n images $\mathrm{I}^{(i)}_{r\times c}$ of r-by-c pixels each is given. Assume also that the correspondence problem is somehow solved and that the correspondences between the positions of some control points have been established, k points in each image. Let the coordinates of the j-th control point in the i-th image be $\mathbf{p}^{(i)}_j \in \mathbb{R}^2$. Let vectors

$$\mathbf{s}_i = \left(\mathbf{p}^{(i)}_1(1), \mathbf{p}^{(i)}_1(2), \mathbf{p}^{(i)}_2(1), \mathbf{p}^{(i)}_2(2), \ldots, \mathbf{p}^{(i)}_k(1), \mathbf{p}^{(i)}_k(1)\right)^T \tag{2.8}$$

represent the ensembles of control points in each i-th image as a vector in $\mathbb{R}^{2k}$, call them the shape vectors. By solving the Procrustes problem (see Bookstein [33] or Seber [235]), the shapes $\mathbf{s}_i$ are brought into alignment and the average shape $\tilde{\mathbf{s}}$ is computed:

$$\tilde{\mathbf{s}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{s}_i. \tag{2.9}$$

The images are then warped, see Section 3.4.1, to the mean shape (a process called *shape normalisation* by Cootes and Taylor [59]) to obtain a new set of "shape-free" images $\hat{\mathrm{I}}^{(i)}$. Let vectors $\mathbf{t}_i = \mathrm{reshape}_{rc\times 1}\,\hat{\mathrm{I}}^{(i)}$ represent the vectorised shape-free images $\hat{\mathrm{I}}^{(i)}$ as points in $\mathbb{R}^{r\times c}$, and let the mean shape-free image be

$$\tilde{\mathbf{t}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{t}_i. \tag{2.10}$$

After concatenating the shape vectors and the texture vectors into the corresponding observation matrices, applying PCA as in Section 2.4 yields two linear models of the same form as in Eq. (2.7), one for the shape, and one for the shape-free texture:

$$\mathbf{t} \approx \mathrm{E}_t\mathbf{p}_t + \tilde{\mathbf{t}} \tag{2.11}$$

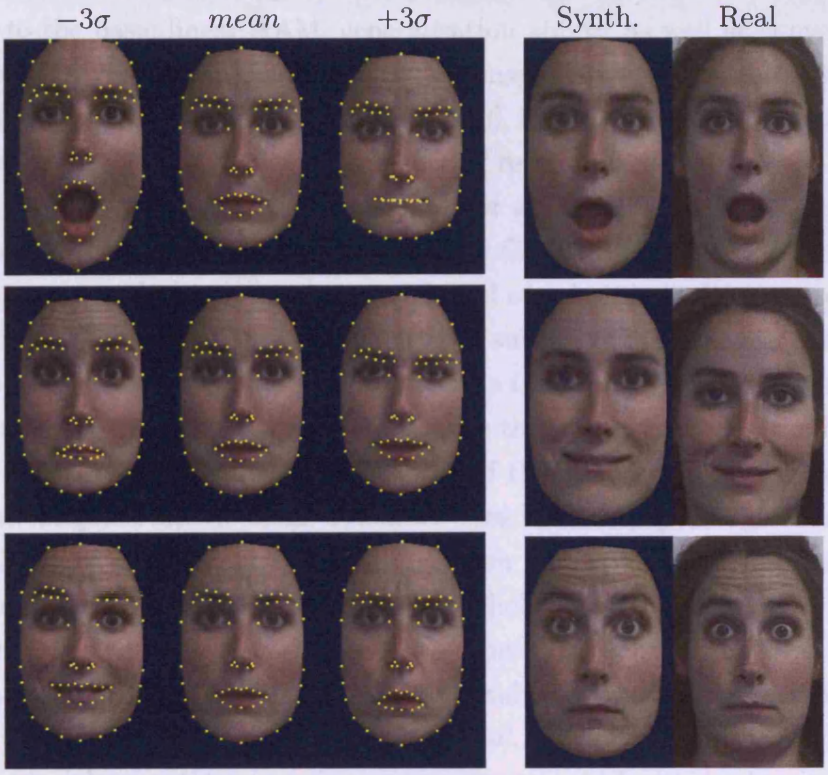
$$\mathbf{s} \approx \mathrm{E}_s\mathbf{p}_s + \tilde{\mathbf{s}}, \tag{2.12}$$

where E_s and E_t are the matrices whose columns comprise several most important basis vectors, $\mathbf{p}_s$ and $\mathbf{p}_t$ are vectors of parameters that parsimoniously summarise every example of shape and texture respectively; the "$\approx$" sign indicates that the new basis has reduced dimensionality and the unprojection is therefore inexact. To find further correlation between shape and texture, a combined appearance model is built by applying PCA to the concatenated shape and texture parameters vectors, to find the basis E_c of the combined model:

$$\mathbf{p}_c = \begin{pmatrix} \mathbf{p}_s \\ \mathrm{W}_t\mathbf{p}_t \end{pmatrix} = \mathrm{E}_c\mathbf{c} = \begin{pmatrix} \mathrm{E}_{cs} \\ \mathrm{E}_{ct} \end{pmatrix}\mathbf{c}, \tag{2.13}$$

where W_t is a scaling matrix to account for difference in units, $\mathbf{c}$ is a vector of parameters for the combined model. In practice (Cootes and Taylor [59]), W_t is simply set to

$$\mathrm{W}_t = \sqrt{\frac{\sum_i \boldsymbol{\lambda}_t(i)}{\sum_i \boldsymbol{\lambda}_s(i)}}\,\mathrm{diag}\,\mathbf{1}. \tag{2.14}$$



(a) First three modes of appearance variation, $\pm 3\sigma$ from the mean. Yellow dots are the control points used to shape-normalise the images.

(b) Comparison of real images against synthesised ones, using the first four eigenvectors.

FIGURE 2.7: Demonstration of images synthesised by the classical 2D AAM.

The shape and texture can now be controlled just by the parameter vector $\mathbf{c}$:

$$\mathbf{s}(\mathbf{c}) \approx \mathrm{E}_s \mathrm{E}_{cs} \mathbf{c} + \tilde{\mathbf{s}} \tag{2.15}$$

$$\mathbf{t}(\mathbf{c}) \approx \mathrm{E}_t \mathrm{W}_t^{-1} \mathrm{E}_{ct} \mathbf{c} + \tilde{\mathbf{t}}. \tag{2.16}$$

AAMs have been previously used for modelling of craniofacial appearance (Cootes and Taylor [64], Edwards *et al.* [92], Gross *et al.* [114–116]), with applications in face tracking, recognition, synthesis, video-assisted speech recognition (Matthews *et al.* [180], Lan *et al.* [153]) *etc.* Despite their simplicity, AAMs remain one of the state of the art modelling approaches due to their representational power and computational efficiency (Gonzalez-Mora [110]).

An extension to AAMs is proposed by Gonzalez-Mora *et al.* in [110] where it is termed Bilinear Active Appearance Model (BAAM). The idea behind

BAAM is to decouple changes due to pose and expression (or identity). Compared to the basic linear AAM, generalisation ability, as well as convergence performance, when fitting the model to new images, are improved with BAAMs, as demonstrated in Gonzalez-Mora *et al.* [110], in addition to better robustness to pose changes when applied to the task of recognition. An excellent review of AAMs, which includes a summary of their applications and various recent advances and improvements, is presented by Gao *et al.* in [102].

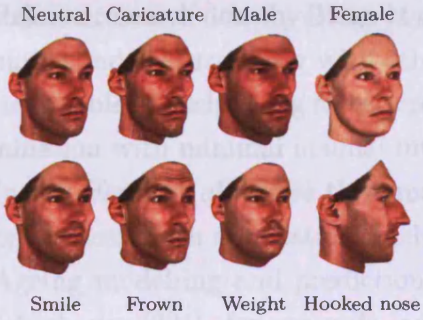
Cosker *et al.* [69] have used a special kind of AAM, built from an artificial training ensemble composed of a number of subsets of face images, such that within each subset only a specific part of the face (left eye, right eye, mouth, forehead, *etc.*) undergoes deformation while the rest of the face is artificially set to a neutral expression. The purpose of the model in Cosker *et al.* [69] is to investigate realistic facial dynamics based on the psychological analysis of real people and to determine the relative contribution of various facial actions to the resulting perception and psychological judgement, see Fig. 2.9. Cosker *et al.* [69] argue that there exists a significant difference between simply recognising a facial expression or action and truly believing in their genuineness. Applications of the results due to Cosker *et al.* [69] include synthesis of ultra-realistic facial animations (e.g. in computer games), and, conversely, automatic evaluation of authenticity of facial expressions and actions in existing video sequences.

Bettinger and Cootes [22] further applies AAMs to model and investigate facial behaviour which is regarded as a sequence of short actions (samples from a statistical model representing the action's variability). Variable-length Markov models are used to define the ordering of actions, and are trained from long video sequences of a talking face.

A conceptually similar geometry-driven facial expression synthesis system is presented by Zhang [300], capable of automatically augmenting the performance-driven appearance with additional details (*e.g.* wrinkles).

2.6 3D Models of Appearance

The author of this thesis has first proposed an efficient extension of 2D appearance models for modelling of 3D surfaces in Sidorov *et al.* [242].



(a) Various modes of variations in a model of a single face [28].

(b) *Top row:* fitting a MM to an image *(left). Bottom row:* rendering MM back into the image.

FIGURE 2.8: Morphable Models of [28].

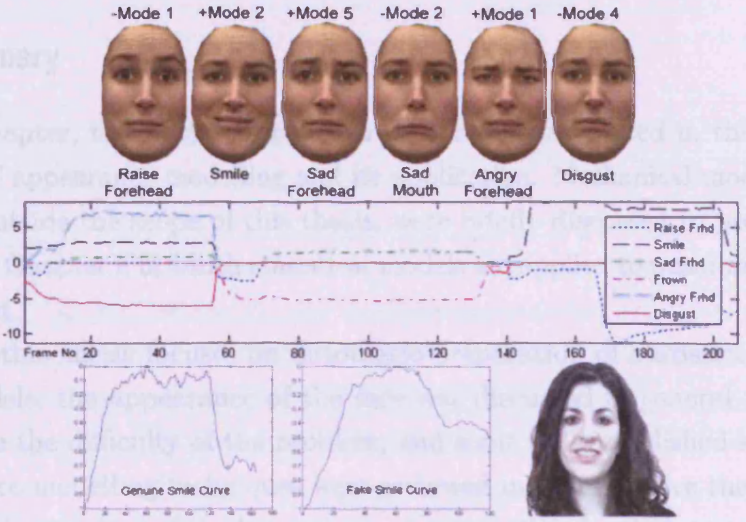


FIGURE 2.9: *Top*: Distinct facial actions synthesised by the appearance model. *Middle*: Result of tracking of a real person's performance using the appearance model (value of the feature parameters). *Bottom:* Value of the most significant feature parameter tracked from a real person performing a real and a fake smile. Image from Cosker *et al.* [69].

A closely related class of models, a powerful technique for modelling textured 3D faces, proposed in the seminal paper by Blanz and Vetter [28] is named Morphable Model (MM). It is similar to AAM, except it uses separate models for shape and texture. Blanz and Vetter [28] derive a morphable face model from an example set of 3D faces by transforming the texture and shape of the samples into a vector space representation. The model is then capable of synthesising new faces by forming the linear combination of the prototypes.

Related research done by Blanz *et al.* [27] resulted in a system that estimates 3D shape and texture along with other scene parameters from single images and is capable of exchanging faces across large differences in the viewpoint and illumination with minimal manual interaction.

In [26] Blanz *et al.* utilise their morphable model for PCA-based representation of faces, with applications including face recognition from 3D scans.

Ageing modelling and prediction is also an expanding area of research. In Scherbaum [231], for example, an automated algorithm is described for prediction of children's facial growth based on an example-driven approach. They claim that it is possible to estimate an age-progressed 3D head of a person from a single photograph at the present age.

2.7 Summary

In this chapter, the work described in this thesis was placed in the broader context of appearance modelling and its application. Mechanical models, even though outside the scope of this thesis, were briefly discussed to prepare the reader for Chapter 5 in which statistical models are applied to quasi-mechanical simulation.

Since this thesis focuses on automatic preparation of statistical appearance models, the appearance of the face was discussed in general terms, to emphasise the difficulty of the problem, and some well established statistical appearance modelling techniques were reviewed in order to give the reader a better understanding of the importance of registration (spatial normalisation) for the task at hand.

The problem of registration itself was reviewed and a brief taxonomy of registration methods was given, to establish terminology and to place the subsequent chapters in the broader context of image and surface registration literature.

The next chapter, in which a novel stochastic algorithm for groupwise non-rigid image registration is proposed, begins with a more thorough review of techniques specific to this particular problem.

CHAPTER

3 Groupwise Registration of Images

In Soviet Russia, images register you.

Yakov Smirnoff

In this chapter, the groupwise registration of ensembles of 2D images is discussed. After presenting the background and the relevant work, the exposition proceeds to introduce the main contribution: a novel, fast and reliable, fully unsupervised stochastic algorithm to search for optimal dense groupwise correspondence in large sets of unlabelled images.

The efficiency of the proposed approach stems from novel dimensionality reduction techniques specific to the problem of groupwise image registration, intimate integration of the deformation model and the optimisation regime, and from comparative insensitivity of the adopted optimisation scheme (Simultaneous Perturbation Stochastic Approximation (SPSA)) to the high dimensionality of the search space.

The chapter concludes with the evaluation of the proposed method, which demonstrates high robustness and success rate, fast convergence on various types of test data, including facial images featuring large degrees of both inter- and intra-subject variation. Further, considerable improvement in terms of accuracy of solution and speed compared to traditional methods is shown.

Due to the robustness of the proposed approach it is capable of performing *inter-subject* groupwise registration of face images: a corpus of individual face

images is taken and successfully registered. This is a pioneering achievement: for the first time in the world the automatic non-rigid registration of data possessing such variety has been reported in our paper, Sidorov *et al.* [243] (in CVPR '09), on which this chapter is based.

Additionally, the proposed algorithm is formulated in a way which admits efficient implementation. In particular, it is readily suited to implementation on graphics processing units (GPUs), see Section 3.5.12.

3.1 Introduction

One of the primary concerns of computer vision is understanding of images. A special case of that is understanding of *ensembles* of images. Either to directly analyse variation across the ensemble, or to construct a statistical model explaining the variation, the key technique (Cootes *et al.* [67]) in computer vision is to first establish the dense, pixel-to-pixel, correspondences between the images, in other words to register them. The fundamental challenge is to find the dense correspondences between images of deformable objects *automatically.*

Registration of image ensembles has now become an important problem in computer vision, with numerous applications ranging from character recognition (Learned-Miller [156], Miller [182]), medical imaging (Marsland and Twining [174], Twining *et al.* [277]), to modelling of facial dynamics (Cootes *et al.* [66]). Typically, such applications involve the analysis of deformable structure in groups of images and the construction of some statistical model of appearance (Davies *et al.* [76]). Registration allows the information about the deformations between images, implicitly contained in the image ensemble, to be quantitatively studied (Marsland and Twining [174]). In medical imaging, for example, groupwise registration is frequently used for direct analysis of the variation across a group of images: to assess change or to compare different examples within a group (Guimond *et al.* [119], Twining *et al.* [277]). Model-based computer vision methods, such as those used for image interpretation and require a statistical model to be built from a corpus of images (Baker *et al.* [14], Cootes *et al.* [61,62]), also benefit from automatic groupwise registration methods.

Furthermore, unsupervised groupwise non-rigid registration is especially important when dealing with large data sets for which manual or even semi-automatic annotation is either too time consuming or impractical. It is, therefore, of special importance in the context of study that the existence of such methods makes it possible to automatically construct statistical models of appearance in an entirely unsupervised fashion from a set of example images or 3D scans.

Groupwise registration may be regarded as an inverse problem (Fischer and Modersitzki [98]) that aims to recover the underlying process or phenomenon which explains the variation between the images. For example, in the case of temporally separated images of a deforming object, the aim of registration is to recover and model the spatial transformations that lead to the (usually highly non-linear) changes in pixel intensities in the resulting images.

The fact that image registration is an inverse problem, makes its general case solution a very difficult task, which is true of many inverse problems, such as that of inverse kinematics. The analogy can be drawn between registration and some related inverse problems: shape-from-X, where a process or phenomenon is being sought that transforms one modality into another (*e.g.* shape into shadow); or that of stereopsis, wherein one aims to compute the disparity, or depth, for each pixel, given a set of spatially separated images, and so recovering the underlying structure of the scene that transforms one image into another.

An intimate relationship exists between groupwise registration and the problem of manifold learning (see Samko [228] for a review of manifold learning methods). Consider an idealised example in Fig. 3.1 where two types of interpolation between images are shown. In the top row, the interpolation is performed without any prior knowledge of the manifold of valid hand images: the middle image is simply a point half-way along the shortest path between the left and the right images in the Euclidean space of all possible images of that size. However, groupwise registration of a corpus of deforming hand images would have shed light on the shape of the manifold of valid hand images, for example by constructing an AAM from the ensemble of registered hand images. Then, taking the middle image to be half-way along the shortest path *on such manifold*, ideally a result akin to that in the bottom row would be obtained.

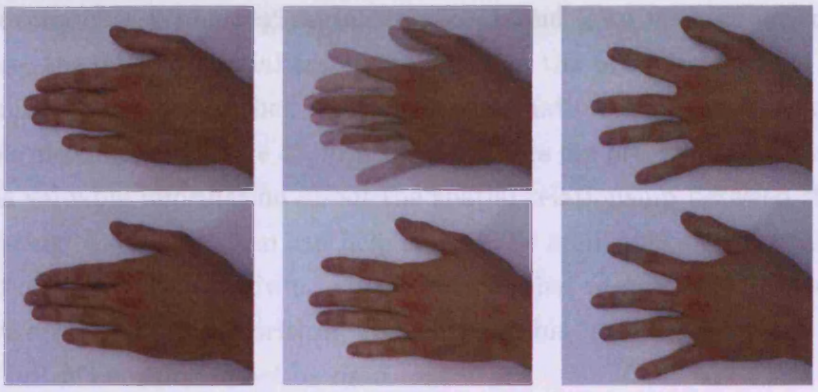


FIGURE 3.1: Interpolation between the images in the leftmost and the rightmost columns. The result of the interpolation is shown in the middle column. *Top row:* linear interpolation in the Euclidean space. *Bottom row:* hypothetical ideal interpolation on the manifold of valid hand images.

FIGURE 3.2: Illustration of the benefits of information propagation for groupwise registration.

3.2 Motivation

When registering together multiple images, as mentioned in Section 2.3.1, one of two general approaches can be adopted. The first approach is to select one of the images as a *reference* and then repeatedly apply a pairwise registration algorithm (for a review see Zitova and Flusser [303]) to align each of the images in turn with this reference, thus decomposing and reducing the problem of registering multiple images to a sequence of simpler subproblems or registering two images.

While this naïve procedure might work in uncomplicated cases, it suffers from an important drawback. At any point in the algorithm, information from only *two* images is being used, and no propagation of information between the subproblems (of registering the reference and the i-th image) ever occurs.

That this is indeed a problem can be illustrated with the following example. In Fig. 3.2, the right eye in image 1 is occluded, and in image 2 the left

eye is occluded. When registering images 1 and 2 to each other, or to a reference, the desired spatial transformations in the occluded regions cannot be unambiguously established if no other information is given. However, the ensemble also includes image 42, in which both eyes are present, and it therefore contains valuable information about the spatial relationship between the eyes. Propagating this information can help resolve the ambiguities that would have arisen if image 42 was not given. This illustrates that *ideally*, in a well-designed groupwise registration algorithm, *all* the available information from *all* the images in the ensemble *must* be used.

In contrast, simple repeated application of a pairwise registration algorithm will inevitably be affected by the choice of the reference image; this leads to errors and inaccuracies in the final alignment (Marsland *et al.* [176]). In addition, an unfortunate choice of the reference, for example an image that is missing features or is not characteristic of the rest of the ensemble, will corrupt the alignment further: the results will be statistically biased (Marsland *et al.* [176]).

To combat such issues, *groupwise* approaches have been recently developed. They consider the entire group of images simultaneously when bringing the images into alignment (Cootes *et al.* [62, 66], Cristinacce *et al.* [74], Davies *et al.* [80] Marsland *et al.* [176], Petrovic *et al.* [205], Twining *et al.* [278]). Broadly speaking, the idea of groupwise registration is to utilise as much information as possible from the *entire* ensemble of images. Or, looking at it in another way, to somehow propagate information from one image to another, or from one subproblem to another, to increase the quality and robustness of registration.

So, in such groupwise approaches, the information from the *entire* data set is being utilised at each stage, rather than from only a pair. Only by considering *multiple examples* simultaneously can the corresponding structures be reliably and accurately identified. Additionally, only when the images in an ensemble have been aligned in a groupwise fashion, to a common reference frame, the correspondences between any pair of the images can be consistently deduced, via the common reference frame (Marsland *et al.* [176]).

Indeed, the groupwise paradigm to finding dense correspondence across a set of unlabelled examples (images or shapes) has been experimentally shown to be superior to pairwise methods (Cootes *et al.* [66]).

3.3 The Challenge of Groupwise Registration

The advantages of the groupwise approach come at a cost. Unlike with pairwise methods, the dimensionality of the space in which the search for the optimal solution has to be performed, grows very rapidly with the number of samples in the set when a groupwise approach is employed.

The groupwise image registration problem is typically formulated in terms of optimisation of an intensity-based objective function over a certain spatial transformation space and so may be decomposed into three subproblems (Davies *et al.* [80], Cootes *et al.* [67], Sidorov *et al.* [243]):

- A mechanism for representing and manipulating dense correspondence between images (the model of admissible deformations).
- An objective function F with a minimum at a point corresponding to the desired good registration.
- A global minimisation algorithm which optimises F.

Global minimisation of the objective function, F, whose arguments are correspondences between all images and whose value measures the quality of registration, solves the problem.

In practice, the very high dimensionality of the search space presents a significant obstacle to finding the optimal solution (Davies *et al.* [76, 78]). Suppose n images are to be registered, and the correspondences between images are controlled by k degrees of freedom per image, yielding $2nk$ degrees of freedom in total; even for a modest data set (say, hundreds of images) and a modestly flexible model of deformations (say, tens of degrees of freedom per image) the dimensionality of the space in which the solution is to be found is measured in thousands.

The problem of efficient optimisation of the very high dimensional objective function in the context of groupwise image registration has not been extensively explored in the literature. Most traditional optimisation algorithms, applied naïvely, cannot reliably deal with an optimisation problem of such magnitude and tend to converge to local minima (Cootes *et al.* [62,66], Davies *et al.* [76,78]). Some stochastic algorithms (for example, Simulated Annealing (SA) and genetic algorithms (GAs)), which attempt to avoid local minima, have impractical computation times even for small data sets, see Section 3.4.3.

I contend that the problem of optimisation of the objective function, the key component of groupwise registration, needs to be addressed explicitly.

The main contribution of this chapter is to describe an efficient optimisation framework for many-dimensional groupwise objective functions for non-rigid image registration that can quickly and reliably find very good (and in practice almost always the best) minima.

The approach proposed in this chapter alleviates the "curse of dimensionality" on two fronts:

- A novel solution that implicitly reduces the dimensionality of the search space as the search progresses by incrementally learning optimal deformations is proposed.
- A novel application of stochastic optimisation algorithms that do not significantly degrade in performance as the dimensionality grows is proposed.

Additionally, the algorithms are formulated in a way that is amenable to efficient implementation, including harnessing the processing power of modern GPUs. Indeed, apart from the control logic, all steps in the proposed algorithms can be performed on a GPU, see Section 3.5.12.

3.4 Groupwise Registration Background

As discussed above, groupwise registration can be decomposed into three subproblems: modelling of deformations, evaluation of the quality of alignment (objective function), and optimisation. In the literature, many various combinations of approaches to each of these subproblems are found. Because of this, rather than linearly discuss background by listing the contributions of individual papers, here the literature will be reviewed combinatorially, according to the approach taken to solve each of the subproblems.

3.4.1 Deformation Models

The deformation models employed in groupwise approaches are typically borrowed from the comparable pairwise methods, and so the vast literature of results on deformation modelling for pairwise registration remains relevant. Below some of the most important techniques are reviewed.

THE AFFINE TRANSFORM. The simplest useful model of transformations is the affine one. The affine transformation is a map $f\colon \mathbb{R}^n \to \mathbb{R}^n$ of the form

$$\mathbf{x}' = f(\mathbf{x}) = \mathrm{A}\mathbf{x} + \mathbf{t}, \qquad \mathbf{t} \in \mathbb{R}^n, \tag{3.1}$$

where A is a linear transformation of $\mathbb{R}^n$, ant $\mathbf{t}$ is the translation. For example, in $\mathbb{R}^2$, a stretch by a factor (q_x, q_y) along each axis, followed by shear (s_x, s_y) along each axis, then by rotation by the angle α counterclockwise around the origin, and finally by translation by (t_x, t_y) yields

$$\begin{aligned}\mathbf{x}' = &\begin{pmatrix}\cos(\alpha) & -\sin(\alpha)\\ \sin(\alpha) & \cos(\alpha)\end{pmatrix}\begin{pmatrix}1 & 0\\ s_y & 1\end{pmatrix}\begin{pmatrix}1 & s_x\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & q_y\end{pmatrix}\begin{pmatrix}q_x & 0\\ 0 & 1\end{pmatrix}\mathbf{x} + \begin{pmatrix}t_x\\ t_y\end{pmatrix} = \\ &\begin{pmatrix}q_x(\cos\alpha - s_y\sin\alpha) & -q_y(\sin\alpha - s_x(\cos\alpha - s_y\sin\alpha))\\ q_x(\sin\alpha + s_y\cos\alpha) & q_y(\cos\alpha + s_x(\sin\alpha + s_y\cos\alpha))\end{pmatrix}\mathbf{x} + \begin{pmatrix}t_x\\ t_y\end{pmatrix}.\end{aligned} \tag{3.2}$$

The 2-by-2 linear transformation matrix, A, together with the 2-by-1 translation vector $\mathbf{t}$ give the total of six degrees of freedom. In the case of rigid transformations, the number of degrees of freedom is three (two for translation and one for rotation). This is a global transformation, *i.e.* it is applied to the entire images: it represents the gross relative motion of the images, but is of course incapable of describing non-rigid, relative local deformations, which are discussed next.

Several popular non-rigid deformation models rely on defining a translation at a small number of control points, interpolating the deformations smoothly everywhere in between. This can be regarded as scattered data interpolation problem for which many solutions are known from scientific and engineering literature (Späth [251], Lee *et al.* [157], Renka [221]).

PIECE-WISE AFFINE MODEL. The simplest local deformation model is the piece-wise affine one (Amidror [8], Cootes *et al.* [66,67], Petrovic *et al.* [205], Twining *et al.* [278]). A set of control points are selected such that their convex hull [219] covers the entire region of interest (RoI). A suitable tessellation of the RoI is then computed, usually by means of Delaunay triangulation (see, for example, Делоне (Delaunay) [304], Скворцов (Skvortsov) [306]).

Given a triangle with vertices $\mathbf{v}_1 = (x_1, y_1)$, $\mathbf{v}_2 = (x_2, y_2)$, and $\mathbf{v}_3 = (x_3, y_3)$, the barycentric coordinates (Coxeter [73]), call them $(\lambda_1, \lambda_2, \lambda_3)$, of a point $\mathbf{x} = (x, y)$ can are expressed as

$$\lambda_1 = \frac{1}{D}\begin{vmatrix}(x_2 - x_3) & (x_3 - x)\\ (y_2 - y_3) & (y_3 - y)\end{vmatrix}, \qquad \lambda_2 = \frac{1}{D}\begin{vmatrix}(x_3 - x) & (x_1 - x_3)\\ (y_3 - y) & (y_1 - y_3)\end{vmatrix}, \tag{3.3}$$

$$\lambda_3 = 1 - \lambda_1 - \lambda_2, \qquad \text{where } D = \begin{vmatrix}(x_1 - x_3) & (x_2 - x_3)\\ (y_1 - y_3) & (y_2 - y_3).\end{vmatrix} \tag{3.4}$$

The symbol $\mathfrak{B}_{\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3}$ shall be used to denote the mapping between the Cartesian and the barycentric coordinates, given the triangle vertices $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$:

$$\boldsymbol{\lambda} = \begin{pmatrix}\lambda_1 & \lambda_2 & \lambda_3\end{pmatrix}^T = \mathfrak{B}_{\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3}(\mathbf{x}). \tag{3.5}$$

Conversely, the coordinates of a point with barycentric coordinates $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ in a triangle with vertices $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ are simply

$$\mathbf{x} = \mathfrak{B}^{-1}_{\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3}(\boldsymbol{\lambda}) = \lambda_1\mathbf{v}_1 + \lambda_2\mathbf{v}_2 + \lambda_3\mathbf{v}_3 = \begin{pmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3\end{pmatrix}\begin{pmatrix}\lambda_1 & \lambda_2 & \lambda_3\end{pmatrix}^T. \tag{3.6}$$

(The concept of barycentric coordinates easily generalises to simplexes of higher dimensions).

The piece-wise affine transformation $\mathbf{x}' = W_{\text{PWA}}(\mathfrak{T}, \mathbf{x})$, $W_{\text{PWA}}\colon \mathbb{R}^2 \to \mathbb{R}^2$, on a triangulated domain $\mathfrak{T}$ is performed as follows. For each point $\mathbf{x}$ in the RoI, its encompassing triangle is found by searching through all the triangles in the tessellation $\mathfrak{T}$ to find the unique[1] triangle for which $0 \leq \lambda_i \leq 1$, $\forall i$ (a point $\mathbf{x}$ is inside or on the boundary of a triangle if and only if its barycentric coordinates are all in [0,1]). Once such triangle $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is found, and so is the corresponding transformed triangle $\{\mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_3\}$ from the deformed tessellation $\mathfrak{T}'$, the transformation can be written as

$$\mathbf{x}' = W_{\text{PWA}}(\mathfrak{T}, \mathfrak{T}', \mathbf{x}) = \mathfrak{B}^{-1}_{\mathbf{v}'_1,\mathbf{v}'_2,\mathbf{v}'_3}(\mathfrak{B}_{\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3}(\mathbf{x})). \tag{3.7}$$

For reasons discussed in Section 3.5.8, it is necessary that the deformations be invertible and the inverse should be easily computable. In addition to that, in non-rigid registration literature it is usually assumed that if a structure is

[1]Ignoring for simplicity the case when a point is on a shared boundary between two triangles.

present in one image, it is also present in all other images (Marsland *et al.* [176], but see the discussion in Section 3.7).

Because of this assumption, deformations models that are not just bijective but *diffeomorphic* are used in some algorithms (*e.g.* Marsland and Twining [174]). (A bijective map $f\colon M \to N$ is called diffeomorphic if it and its inverse $f^{-1}\colon N \to M$ are both differentiable to some order.)

While the piece-wise affine model is not diffeomorphic, as it is not differentiable, it is desirable to ensure at least bijectivity. It is easy to see that within each individual triangle this piece-wise affine transformation is bijective. To ensure that it is also invertible across the entire triangulated RoI, it is necessary to ensure that the no triangles ever overlap. This can easily be done by ensuring the consistency of signs of the signed areas of all triangles (Cootes *et al.* [67]). It should also be noted that interpolation on a triangulated domain need not in general be affine (Amidror [8]).

THIN PLATE SPLINES. A classical technique for interpolation of scattered data in many dimensions, Thin Plate Splines (TPSs), introduced by Duchon [90], has found an application as a deformation model for image registration, (Johnson and Christensen [135]). An excellent introduction of TPSs is given in Bookstein [32]. This interpolant minimises the bending energy of a thin metal plate subject to control points constraints, hence the name. At the core of TPS interpolation in $\mathbb{R}^n$ is the kernel function (Marsland and Twining [174], Sprengel *et al.* [252], Bookstein [32])

$$U(\mathbf{x},\mathbf{c}) = \begin{cases} |\mathbf{x}-\mathbf{c}|^{4-n}\log|\mathbf{x}-\mathbf{c}|, & \text{when } n \text{ even and } \mathbf{x} \neq \mathbf{c} \\ |\mathbf{x}-\mathbf{c}|^{4-n}, & \text{when } n \text{ odd or } \mathbf{x} = \mathbf{c}. \end{cases} \tag{3.8}$$

The TPS interpolant $W_{\text{CPS}}\colon \mathbb{R}^n \to \mathbb{R}^n$ then, given a sparse set of control points $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$, takes the form (Sprengel *et al.* [252])

$$\mathbf{x}' = W_{\text{CPS}}(\mathbf{x}) = (\mathbf{t} + \mathbf{A}\mathbf{x}) + \sum_{i=1}^{k} \mathbf{w}_i U(\mathbf{x}, \mathbf{c}_i). \tag{3.9}$$

The constants $\mathbf{t}$, $\mathbf{A}$, that control the affine part of the warp, as well as the vector-valued coefficients $\mathbf{w}_i$ are computed from constraints: the interpolant must pass through the control points. This is a linear problem and its solution is discussed in *e.g.* Sprengel *et al.* [252] and Chui [58].

It should be noted that the TPS interpolant is based on radial basis functions (RBFs), because the kernel $U(\mathbf{x}, \mathbf{c})$ in Eq. (3.8) depends only on the distance $r = |\mathbf{x} - \mathbf{c}|$ between a point $\mathbf{x}$ and a control point $\mathbf{c}$. So, TPS is just a particular example of interpolation with RBFs. Kernels that can be used in place of Eq. (3.8) include: linear $U(r) = r$, cubic $U(r) = r^3$, multiquadrics $U(r) = (1 + r^2/\sigma^2)^{1/2}$, the frequently used Gaussian $U(r) = e^{(-r^2/2\sigma^2)}$ whose effect is more local than that of Eq. (3.8), and others. For more information on RBF kernels and scattered data interpolation in general, the reader is referred to Wendland [293] and Liu [167].

As discussed above, the deformation model in some approaches is assumed to be diffeomorphic. Below, two popular techniques for which diffeomorphicity can be guaranteed under well-known conditions are reviewed.

FREE-FORM DEFORMATIONS BASED ON B-SPLINES. A frequent choice in the field of medical imaging, the B-spline free-form deformation (FFD) model, given a regular lattice of control points $\mathbf{c}_{i,j}$, can be formulated (in $\mathbb{R}^2$) as follows (Lee *et al.* [158]):

$$\mathbf{x}' = W_{\text{FFD}}(\mathbf{x} = (x, y)) = \sum_{k=0}^{3} \sum_{l=0}^{3} B_k(x - \lfloor x \rfloor) B_l(y - \lfloor y \rfloor) \mathbf{c}'_{\lfloor x \rfloor - 1 + k, \lfloor y \rfloor - 1 + l}, \quad (3.10)$$

where $B_k(\cdot)$ and $B_l(\cdot)$ are the uniform cubic B-spline basis functions, and $\mathbf{c}'_{i,j}$ are the displaced control points. See Lee *et al.* [158] and Rueckert *et al.* [227] for derivation of the coefficients B_l, B_k given $\mathbf{c}_{i,j}$ and the discussion of a more useful model, there termed Multi-level Free-form Deformation (MFFD). What is more important is that Lee *et al.* [158] prove an important bijectivity theorem: the mapping W_{FFD} in Eq. (3.10) is bijective if

$$-0.48 \leq \mathbf{c}'_{i,j}(k) - \mathbf{c}_{i,j}(k) \leq -0.48, \qquad \text{for all } i, j, \text{ and } k. \quad (3.11)$$

Under these conditions the B-spline model is also diffeomorphic, as B-splines are maximally differentiable (except possibly at the control points). A generalisation of B-spline FFD to $\mathbb{R}^3$ applied to registration of volumetric brain scans by Balci *et al.* in [15, 16] illustrates the usefulness of the B-spline FFD as a deformation model.

CLAMPED-PLATE SPLINES. Twining *et al.* [276] introduced a representation of deformation fields based on polyharmonic Clamped-Plate Splines (CPSs). This representation is applied for groupwise registration of medical imagery by Marsland *et al.* [175]. The CPS-based deformations are diffeomorphic: bijective and differentiable to some order (Twining *et al.* [276]). They are also bounded, hence the name, to a spherical region. Unlike the TPS interpolant, which affects the entire space and decays only asymptotically, CPSs satisfy both Dirichlet and Neumann boundary conditions: the effect of the CPS interpolant vanishes smoothly at the boundary of the spherical region (Twining *et al.* [276]). This is due to the fact that on the boundary of a unit ball in $\mathbb{R}^n$, the value and the first normal derivative of the general Green's function, G, given in Eq. (3.12), of the biharmonic clamped plate equation (see Twining *et al.* [276] and Marsland *et al.* [175] both citing Boggio [30]), on which the CPSs are based, are both zero. In two dimensions G has the form

$$G(\mathbf{x},\mathbf{y}) = |\mathbf{x}-\mathbf{y}|^2 \left(\frac{1}{2}\left(A(\mathbf{x},\mathbf{y})^2 - 1\right) - \log A(\mathbf{x},\mathbf{y})\right), \tag{3.12}$$

$$\text{where } A(\mathbf{x},\mathbf{y}) = \frac{\sqrt{|\mathbf{x}|^2|\mathbf{y}|^2 - 2\mathbf{x}\cdot\mathbf{y} + 1}}{|\mathbf{x}-\mathbf{y}|}, \tag{3.13}$$

and in three dimensions (*tri*harmonic clamped-plate spline), see Marsland and Twining [174] citing Boggio [30]:

$$G(\mathbf{x},\mathbf{y}) = |\mathbf{x}-\mathbf{y}| \left(A(\mathbf{x},\mathbf{y}) + \frac{1}{A(\mathbf{x},\mathbf{y})} - 2\right), \text{ with } A(\mathbf{x},\mathbf{y}) \text{ as in Eq. (3.13).} \tag{3.14}$$

Then, given a set of control points $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$, the CPS-based transformation takes the form (omitting the affine part)

$$\mathbf{x}' = W_{\text{CPS}}(\mathbf{x}) = \mathbf{x} + \sum_{i=1}^{k} \boldsymbol{\alpha}_i G(\mathbf{x}, \mathbf{c}_i), \tag{3.15}$$

where the vector-valued coefficients $\boldsymbol{\alpha}_i$ are found by solving the exact matching conditions for the control points (Marsland *et al.* [175]).

This deformation model is used by Twining and Marsland in [277] for groupwise registration of brain images. The bounding ball of the CPS in Twining and Marsland [277] is chosen to be the circumcircle of the images. They use 10 control points equally spaced around the skull.

SUPERPOSITION OF ELEMENTARY WARPS. An approach closely related to the RBF-based interpolation is the superposition of elementary warps. The main difference is that the kernels of elementary warps are chosen to only affect a bounded region of the image and the deformations induced by the elementary warps are treated independently: the contributions of individual elementary warps can be arbitrary, not necessarily such that they interpolate a given function between control points.

For example, Lötjönen and Mäkelä [169] model the deformation of the brain with a superposition of spherical elementary warps of the form

$$\mathbf{x}' = W_{\text{Lötjönen}}(\mathbf{x}, \mathbf{c}, \mathbf{d}, r, k) = \mathbf{x} + \frac{e^{\frac{|\mathbf{x}-\mathbf{c}|^2}{r^2}} - e^{-k}}{1 - e^{-k}}\mathbf{d}, \tag{3.16}$$

where k is the sharpness parameter, $\mathbf{d}$ is the free parameter controlling the direction and magnitude of the warp inside the sphere of radius r with the centre at $\mathbf{c}$. As can be easily seen from Eq. (3.16), the deformation at the centre of the sphere ($|\mathbf{x} - \mathbf{c}| = 0$) is $\mathbf{d}$, and at the boundary of the sphere it is zero.

An elementary warp used by Cootes *et al.* in [67] has the form

$$\mathbf{x}' = W_{\cos}(\mathbf{x}, \mathbf{d}) = \mathbf{x} + \left(\prod_{i=1}^{\dim \mathbf{x}} k(\mathbf{x}(i)) \right) \mathbf{d}, \tag{3.17}$$

where the kernel $k(r)$ is

$$k(r) = \begin{cases} (1 + \cos(\pi r))/2, & |r| < 1 \\ 0, & |r| \geq 1. \end{cases} \tag{3.18}$$

It affects only the hypercubic region $\overbrace{[-1,1] \times [-1,1] \times \cdots \times [-1,1]}^{\dim \mathbf{x} \text{ times}}$, and a suitable affine transform can be used to apply Eq. (3.17) to some other region. As in Eq. (3.16), $\mathbf{d}$ is the parameter controlling the magnitude and direction of the elementary warp.

A similar elementary warp, acting on a unit ball and bounded by it, is proposed in Cootes *et al.* [62]:

$$W_{\text{Cootes}}(\mathbf{x}, \mathbf{d}) = \begin{cases} \mathbf{x} + k(|\mathbf{x}|), & \text{when } |\mathbf{x}| < 1 \\ \mathbf{x}, & \text{otherwise,} \end{cases} \tag{3.19}$$

where $\mathbf{d}$ is the displacement of the centre of the ball, $|\mathbf{d}| < 1$, controlling the magnitude and direction of the deformation; $k(|\mathbf{x}|)$ is a smooth kernel

satisfying $k(0) = 1$, $k(1) = 0$, $k'(0) = 0$, $k'(1) = 0$. Cootes *et al.* [62] point out that $W_{\text{Cootes}}(\mathbf{x}, \mathbf{d})$ in Eq. (3.19) is diffeomorphic if

$$|\mathbf{d}| < \frac{1}{\max_{0<|\mathbf{x}|<1}|k'(|\mathbf{x}|)|}. \tag{3.20}$$

This elementary warp is useful because it can be guaranteed to be diffeomorphic and works in arbitrary dimensions. Again, a suitable affine transform can be used to apply Eq. (3.19) to an arbitrary ellipsoidal area.

If only one control point, placed at the centre of the CPS bounding ball, is used, the CPS described above can also be regarded, in $\mathbb{R}^2$, as an elementary bounded warp of the form in Eq. (3.19) if the following kernel is chosen:[2]

$$k(|\mathbf{x}|) = 1 - |\mathbf{x}|^2 + |\mathbf{x}|^2 \log(|\mathbf{x}|^2). \tag{3.21}$$

According to Cootes *et al.* [62], the warp in Eq. (3.19) with the kernel from Eq. (3.21) is guaranteed to be diffeomorphic if $|\mathbf{d}| < 0.25e$.

A computationally cheaper kernel $k(|\mathbf{x}|) = (1 - |\mathbf{x}|^2)^2$ in place of Eq. (3.21) is also proposed in Cootes *et al.* [62] for which the diffeomorphicity of Eq. (3.19) is ensured when $|\mathbf{d}| < 3\sqrt{3}/8 \approx 0.650$.

The number of degrees of freedom in the above control points based deformation models is nk, where n is the dimensionality of the space, and k is the number of control points.

DENSE FIELDS AND FLUID MODELS. Deformation fields can be represented simply by a dense vector-valued matrix, a *deformation map*, specifying displacements of individual pixels. Such is the approach taken in the proposed algorithm, see Section 3.5.2.

Dense representation of deformations is also used in methods based on the ideas of fluid mechanics, which have recently been applied to image registration (Bro-Nielsen and Gramkow [40], Christensen *et al.* [57]). At the core of such methods is the compressible fluid flow equation (Bro-Nielsen and Gramkow [40]):

$$\mu\nabla^2\mathbf{v}(\mathbf{x}) + (\lambda + \mu)\nabla(\nabla \cdot \mathbf{v}(\mathbf{x})) = \mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x})), \tag{3.22}$$

where $\mathbf{v}(\cdot)$ is the velocity field, $\mathbf{f}(\cdot)$ is the force field that drives the flow, $\mathbf{u}(\cdot)$ is the displacement field, μ and λ are viscosity constants. Under this model,

[2] Defining additionally $\log(0) = 0$.

pixels are compressible, they can change volume, this is allowed by the term $\nabla(\nabla \cdot \mathbf{v}(\mathbf{x}))$. The viscous term $\nabla\mathbf{v}(\cdot)$ spatially constraints the velocity field. The force field $\mathbf{f}(\cdot)$ is chosen such that it encourages the parts of the image that are yet poorly aligned to move more. The numerical solution of Eq. (3.22) is discussed by Bro-Nielsen and Gramkow in [40]. Viscous flow of a template image towards alignment with the reference solves the above Eq. (3.22). However, this approach, combining the deformation model and the optimisation procedure, does not fit in the framework of explicit optimisation and will not be discussed further.

Note that the dense non-rigid deformations discussed above need not be applied to the pixel grid directly. To gain computational efficiency, they can be applied to the nodes of a sufficiently dense, but less dense than the grid of pixels, triangular tessellation instead. The transformations of individual pixels can be then computed with the piece-wise affine interpolation between the nodes of the tessellation, which is usually a much cheaper operation. This optimisation has been adopted by Cootes *et al.* in [66, 67].

It is important to note that the spline-based deformation models (TPS and CPS) are computationally expensive: each time the positions of the control points are changed, the parameters of the interpolant need to be recomputed. This is not a cheap operation (*e.g.* a large linear system has to be solved in the case of TPS), and this is a significant drawback if spline-based interpolation is to be used within the objective function.

In contrast, the piece-wise affine interpolation is extremely cheap (especially if delegated to a GPU). The deformation models based on a composition of elementary bounded warps are relatively cheap also, as the parameters for each elementary warp are specified independently, without ensuring that some form of interpolant smoothly passes through all the control points.

It is worth mentioning an interesting exception. In Miller [182] and Learned-Miller [156] a problem very similar to groupwise registration (there termed "congealing") is addressed but applied to the optimisation of also *non-spatial* modes of "deformations" (for example, brightness).

3.4.2 Objective Function

Given a set of images $\{\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_n\}$ and the corresponding deformation fields $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$, the purpose of the objective function is to evaluate the quality of alignment into which the deformation fields bring the images. The objective function should be such that its global extremum corresponds to the desired good alignment. Devising an efficient objective function is on its own a challenging task considering the high dimensionality of the image set and the parameterisation of deformations (Cootes *et al.* [62]).

It is common in groupwise registration literature (Cootes *et al.* [62, 66], Marsland *et al.* [176], Twining *et al.* [278], Petrovic *et al.* [205], Davies *et al.* [80], Cristinacce *et al.* [74]) that algorithms operate on one image "at a time", for all images in the ensemble in turn, repeating the entire process several times until convergence. In such cases, efficiency can be achieved if the objective function is chosen such that its value depends only on the deformation field that is "currently" being optimised. For example, assume the objective function has the form

$$F = \sum_{i=1}^{n} G(\mathcal{D}_i, M(\mathcal{D}_1, \ldots, \mathcal{D}_{i-1}, \mathcal{D}_{i+1}, \ldots, \mathcal{D}_n)), \tag{3.23}$$

where $M(\cdot)$ is some model built from the rest of the ensemble and independent from the "current" $\mathcal{D}_k$, and the summation is done over all the images, treated equally, in the ensemble. When optimising w.r.t. $\mathcal{D}_k$, Eq. (3.23) can then be rewritten as

$$F = G(\mathcal{D}_k, \text{const}) + \sum_{\substack{i=1 \\ i \neq k}}^{n} G(\mathcal{D}_i, M(\mathcal{D}_1, \ldots, \mathcal{D}_{i-1}, \mathcal{D}_{i+1}, \ldots, \mathcal{D}_n)). \tag{3.24}$$

If the model $M(\cdot)$ is chosen such that it changes very little compared to the change in $G(\cdot)$, when $\mathcal{D}_k$ is slightly changed to $\mathcal{D}'_k$,

$$|G(\mathcal{D}_k, \cdot) - G(\mathcal{D}'_k, \cdot)| \gg |G(\text{const}, M(\mathcal{D}_k, \cdot)) - G(\text{const}, M(\mathcal{D}'_k, \cdot))|, \tag{3.25}$$

then the second term in Eq. (3.24), which is the sum of dominating G's independent of $\mathcal{D}_k$ is approximately constant:

$$F \approx G(\mathcal{D}_k, \text{const}) + \text{const}, \tag{3.26}$$

and these constants can be precomputed before optimising F w.r.t. to $\mathcal{D}_k$.

In many algorithms the role of $M(\cdot)$ is played by the evolving reference (*e.g.* Cootes *et al.* [67]), or some other model, computed from all images and the corresponding deformation fields except the one "currently" being optimised. The simplest, but surprisingly well working, choice of $M(\cdot)$ is the average of images (Cristinacce *et al.* [74]). It should be noted that, indeed, in such case if the number of images is large, the individual contribution of each image to $M(\cdot)$ is small, and therefore the approximation in Eq. (3.26) is valid.

The function $G(\cdot)$ then is based on comparison of the "current" image I_k, deformed with $\mathcal{D}_k$, with the reference: $G(\mathcal{D}_k) = Q(W(\mathrm{I}_k, \mathcal{D}_k), M(\cdot))$. Leaving the discussion of other models until the end of this section, the options for the image comparison function $Q(\cdot)$ will now be discussed. Since $Q(\cdot)$ compares only a pair of images the abundant literature on pairwise registration is relevant here.

The simplest, but well working in practice (Cootes *et al.* [66,67]), approach is to compare all the corresponding pixels in the images and aggregate the error:

$$Q(\mathrm{A}, \mathrm{B}) = \frac{1}{rc} \sum_{j=1}^{c} \sum_{i=1}^{r} P(\mathrm{A}(i,j) - \mathrm{B}(i,j)), \tag{3.27}$$

where $P(.)$ is some per-pixel cost function, discussed below, and the scaling of the sum by $1/rc$ serves to make $Q(\cdot)$ insensitive to the image size.

The choice of $P(\cdot)$ is based on the assumed distribution of the per-pixel differences. The two most common assumptions are: the exponential distribution, $p(d) \propto e^{-|d|/\sigma}$, for which $P_{\mathrm{AD}}(d) = |d|$, and the Gaussian distribution, $p(d) \propto e^{-d^2/(2\sigma^2)}$, for which $P_{\mathrm{SD}}(d) = d^2$, see Cootes *et al.* [66]. The former is more long-tailed and therefore leads to more robustness to outliers than the Gaussian assumption (see Cootes *et al.* [66,67] where this is confirmed experimentally). The corresponding sums in Eq. (3.27) in the literature are then called the sum of absolute differences (SAD) and the sum of squared differences (SSD) respectively. Under the assumption of Cauchy distribution (Sebe *et al.* [234]),

$$p(d) \propto \frac{d}{d^2 + \alpha^2}, \text{ the cost function is } P_{\mathrm{Cauchy}}(d) = \frac{\beta^2}{2} \ln\left(1 + \frac{d^2}{\beta^2}\right). \tag{3.28}$$

In stereo matching literature (*e.g.* Sun *et al.* [257]), the following robust metric is often used:

$$P_{\mathrm{Sun}}(d) = -\ln\left((1-\gamma)e^{-|d|/\sigma} + \gamma\right), \tag{3.29}$$

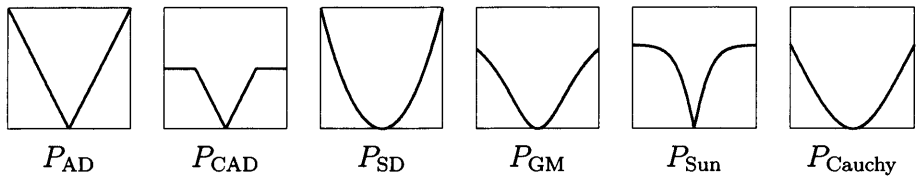


FIGURE 3.3: Comparison of the shapes of the various per-pixel cost functions.

as well as clamped absolute difference: $P_{\text{CAD}}(d) = \min(\varphi, |d|)$. The Geman-McClure function (Black and Rangarajan [25])

$$P_{\text{GM}}(d) = \frac{d^2}{1 + d^2/\mu^2}, \tag{3.30}$$

is also a popular choice because it behaves quadratically near zero, but quickly becomes less steep for larger d. In general, the choice of the per-pixel comparison function is dictated by the desire to make it less sensitive to noise. The shapes of the various such functions are summarised in Fig. 3.3. See also Black and Rangarajan [25] for discussion of other robust error functions. In Klaus *et al.* [148], gradient information is used addition to pixel intensities:

$$\begin{aligned} P(\mathrm{A}(i,j), \mathrm{B}(i,j)) = (1-\omega)|\mathrm{A}(i,j) - \mathrm{B}(i,j)| + \\ \omega \left(|\nabla_x \mathrm{A}(i,j) - \nabla_x \mathrm{B}(i,j)| + |\nabla_y \mathrm{A}(i,j) - \nabla_y \mathrm{B}(i,j)| \right). \end{aligned} \tag{3.31}$$

A similar gradient term in the objective function is employed in Lötjönen and Mäkelä [169].

A comparison of three similarity measures, SSD, SAD, mean absolute difference (MAD), for the problem of registration is presented in Ulysses and Conci [280]. They conclude that, at least for registration of medical imagery of the same modality, their performance in terms of the resulting mean square error (MSE), correlation coefficient (CC) and peak signal to noise ratio (PSNR) is very similar. An analogous result is reported in Cootes *et al.* [66, 67] with SSD slightly outperforming SAD.

The objective function for joint alignment of Miller *et al.* [182], where the joint alignment is called "congealing", is the sum of univariate entropies along

pixel stacks. For binary images (they apply their algorithm to handwritten digits) the joint entropy is defined as (Miller *et al.* [182])

$$E = \sum_{\forall \mathbf{p}} H(v(\mathbf{p})), \tag{3.32}$$

where $\mathbf{p}$'s are coordinates of pixels, $v(\mathbf{p})$ is a binary random variable defined by the stack of pixels, across all images, at position $\mathbf{p}$, and $H(\cdot)$ is the discrete entropy function (Cover and Thomas [71]) of that variable. More precisely, the entropy estimator of Miller *et al.* [182] for a binary pixel stack has the form

$$\hat{H}(\cdot) = -\left(\frac{N_0}{N}\log_2\frac{N_0}{N} + \frac{N_1}{N}\log_2\frac{N_1}{N}\right), \tag{3.33}$$

where N_0 and N_1 are the number of occurrences of the black and white pixels respectively.

This approach is justified because when the images are well-aligned, the intensities of pixels at the corresponding locations (in pixel stacks) form a low entropy distribution, and vice-versa (Balci *et al.* [15]). Note that in Miller *et al.* [182] the $v(\mathbf{p})$'s, from Eq. (3.32), are treated as independent random variables, not accounting for the "lateral" redundancy between pixels, and so Eq. (3.32) is an *upper bound* on the true entropy; minimisation of Eq. (3.32), assumes Miller *et al.* [182], minimises the true entropy of the image distribution.

To apply the stack entropy measure to real-valued images, the objective function, Eq. (3.32), of Miller *et al.* [182] is generalised by Learned-Miller[3] in [156]. To do so, he uses the estimator of Vasicek [281] given in Learned-Miller [156] as

$$\hat{H}_{\text{Vasicek}}(z_1, z_2, \ldots, z_n) = \frac{1}{n-\varphi(n)}\sum_{i=1}^{n-\varphi(n)}\log\left(\frac{n}{\varphi(n)}(z_{(i+\varphi(n))} - z_{(i)})\right), \tag{3.34}$$

where n is the number of pixels in the stack, z_i are the pixel intensities, $z_{(i)}$ are the same intensities in rank order, and $\varphi(n)$ is a function such that $\varphi(n)/n \to 0$ as $n, \varphi(n) \to \infty$, in [156] Learned-Miller uses $\varphi(n) = \lfloor\sqrt{n}\rfloor$. This has the advantage of estimating the entropy directly from samples (intensities of pixels in a stack) without first estimating the distribution itself, which is an expensive

[3]Erik G. Miller has changed his name to Erik G. Learned-Miller sometime between the publication of [182] and [156].

operation. A similar cost function, but for 3D voxel stacks and with a different entropy estimator, is used by Balci *et al.* in [15].

Another classical measure is normalised cross correlation (NCC) [39, 162]

$$S_{\mathrm{NCC}}(\tilde{\mathrm{A}}, \tilde{\mathrm{B}}) = \frac{\sum_{i,j} \tilde{\mathrm{A}}(i,j)\tilde{\mathrm{B}}(i,j)}{\sqrt{\sum_{i,j} \tilde{\mathrm{A}}(i,j)^2 \tilde{\mathrm{B}}(i,j)^2}}, \tag{3.35}$$

where the tilded letters $\tilde{\mathrm{A}}$ and $\tilde{\mathrm{B}}$ indicate intensity-centered images:

$$\tilde{\mathrm{I}} = \mathrm{I}_{r\times c} - \frac{1}{rc}\sum_{i,j} \mathrm{I}_{r\times c}(i,j). \tag{3.36}$$

There are cases when an objective function based on comparing image intensities cannot be devised, such as when the images have different modalities, and so the functions like Eq. (3.27) or cross correlation, Eq. (3.35), cannot be used (Rueckert *et al.* [227]). An information-theoretical measure, termed mutual information (MI), which expresses the amount of information that one image contains about another, has been proposed by Viola and Wells in [287]. Let $H(\mathrm{A})$, $H(\mathrm{B})$ denote the marginal entropies of images A and B. Then $H(\mathrm{A},\mathrm{B})$ denotes their joint entropy, computed from the joint histogram of A and B. Thus the MI between A and B is expressed (Rueckert *et al.* [227], Viola and Wells [287]) as

$$C_{\mathrm{MI}}(\mathrm{A},\mathrm{B}) = H(\mathrm{A}) + H(\mathrm{B}) - H(\mathrm{A},\mathrm{B}), \tag{3.37}$$

or, alternatively (Maes *et al.* [172])

$$C_{\mathrm{MI}}(\mathrm{A},\mathrm{B}) = \sum_a \sum_b p_{AB}(a,b) \log \frac{p_{AB}(a,b)}{p_A(a)p_B(b)}, \tag{3.38}$$

where for image intensity values, a and b, of the corresponding voxels, the joint, $p_{AB}(a,b)$, and the marginal, $p_A(a)$, $p_B(b)$, distributions are estimated by normalisation of joint and marginal image histograms (Maes *et al.* [172]).

Pointing out that the expression in Eq. (3.37) for MI is sensitive to the amount of overlap between the images, Studholme *et al.* [254] propose an overlap-insensitive measure for MI, termed normalised mutual information (NMI):

$$C_{\mathrm{NMI}}(\mathrm{A},\mathrm{B}) = \frac{H(\mathrm{A}) + H(\mathrm{B})}{H(\mathrm{A},\mathrm{B})}. \tag{3.39}$$

Note that MI is a measure of *similarity*: when images are well aligned, their mutual information (MI) is maximised. Naturally, MI can also be used to

compare images of the same modality. Legg [159] discusses the application of MI to non-rigid registration of retinal images in great detail. See also a brief review and discussion of NMI by Bhatia *et al.* in [23].

Myronenko *et al.* [189] propose a supposedly superior, novel intensity-based similarity measure, termed Residual Complexity (RC), that does not assume independence of the intensities from pixel to pixel. In short, RC measures the coding complexity of the difference image. Examples of artificial and real-world problems for which RC-based registration succeeds while the classical measures fail are also provided in Myronenko *et al.* [189].

In a number of works (Cootes *et al.* [66,67], Marsland *et al.* [176], Twining *et al.* [279]), an objective function based on the Minimum Description Length (MDL) principle (see the original work by Rissanen [223]) has been proposed. More specifically, Cootes *et al.* [66,67], Marsland *et al.* [176], Twining *et al.* [279] use the "old", two-part coding formulation of MDL. In these works, the objective function represents the length of the message required to transmit images, using some encoding scheme based on some statistical model. The total length of the message is then the sum of lengths of the data, transmitted using the model, plus the length of the model:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{model}}. \tag{3.40}$$

According to Shannon [239], the optimal number of bits required to communicate an occurrence of event whose probability is $p(x)$ is $\mathcal{L}(x) = -\log p(x)$, and so when computing the message lengths in Eq. (3.40) the probability distribution of the messages being sent has to be known.

Rather than aggregating the information from the image ensemble into an evolving reference model, by applying the MDL principle the cost of encoding the entire ensemble can be used as an objective function, thus using the information from all the images directly, as done in Marsland *et al.* [175,176]. The MDL approach is elegant from the information-theoretical point of view, but is very computationally expensive, as its full implementation requires the construction of a statistical model at each objective function evaluation, as well as estimation of the message length, Eq. (3.40), given the model: it is especially expensive if the probability distributions are estimated empirically from data.

In discussing the MDL approach, Cootes *et al.* [67] point out that the simplest model of texture is just the average of the deformed images, and is works well in practice. The data term in Eq. (3.40) typically dominates, because its contribution scales up with the number of images in the ensemble, and so the approximation $\mathcal{L} \approx \mathcal{L}_{\text{data}}$ is valid [67]. A computationally cheaper way of estimating the message lengths in Eq. (3.40) is to make an assumption on the model PDF rather than estimate it empirically, as is done in Cootes *et al.* [66, 67]. MDL then simply generalises the various per-pixel metrics discussed in Section 3.4.2 depending on the assumptions on the form of PDF

3.4.3 Optimisation

> Nothing takes place in the world whose meaning is not that of some maximum or minimum.
>
> Leonhard Euler

As discussed in Section 3.3, the new challenge encountered by the groupwise registration methods is that of a much greater magnitude of the optimisation problem. This topic has not been extensively explored in the groupwise registration literature, while the conventional pairwise methods, for which extensive literature exists (*e.g.* Zitova and Flusser [303]), have never faced such a problem.

I argue that optimisation is perhaps the dominant component of the registration framework, and it is surprising that the problem of optimisation for groupwise registration has not received well deserved attention in the literature.

By a good optimisation strategy I do not mean merely plugging in the best available general-purpose global optimiser and hoping for the best. On the contrary, I argue that an intimate relationship exists between the optimiser and the other two components of the problem: the deformation model and the objective function. This is in accordance with the "No Free Lunch" theorem (Wolpert and Macready [295]).

The deformation model and the objective function should therefore be designed in such a way as to simplify optimisation, and vice-versa, the optimiser should be tailored for a particular choice of the objective function and deformation model. This is well known (Kolmogorov and Zabin [152],

Scharstein and Szeliski [230]) in the field of stereo matching, the field whose competitive nature (see the Middlebury challenge [1]) has driven researchers to look for more and more ingenious ways to intimately integrate optimisation with the other components of the problem.

An unfortunate choice of the deformation model, for example such that makes it hard to represent the optimal alignment, will make it hard, if not impossible, for the optimiser to find it (see Davies *et al.* [80]). A poor choice of the objective function, for example such that is very noisy and contains many local minima, also makes its global optimisation difficult.

This is generally true for groupwise registration problems: the behaviour of the objective function over the space of possible deformations is *highly* non-linear and contains many local minima. This has led Davies *et al.* [78] to adopt a stochastic optimisation approach. Without elaborating, Davies *et al.* state in [78] that they have experimented with SA (Kirkpatrick [145]) and GAs (Goldberg [109]) before settling on a GA in their experiments.

While these stochastic optimisers are capable of escaping local minima (and so are truly global optimisers), the key disadvantage of SA and GA is that they typically require orders of magnitude more objective function evaluations than deterministic algorithms while exploring the space of solutions of the same dimensionality. This makes their use prohibitive if the evaluation of the objective function is itself a computationally expensive operation, as is the case with the groupwise registration. In addition, SA is *very* sensitive to the choice of parameters, notoriously hard to tune, and cannot easily detect when the solution has been found. Particle Swarm Optimisation (PSO) has been applied by Li *et al.* [164] as a global optimiser in the problem of registering "shape images".

The well-known deterministic global optimiser, the DIRECT algorithm (Jones *et al.* [136]), also suffers from the problem of prohibitively large number of the function evaluations.

An alternative is to use a local optimiser in the hope that it will find the global optimum. Deterministic local optimisation algorithms that operate on evolving a solution towards the optimum, rather than on sampling the entire search space, can be classified into gradient-based and gradient-free methods (Press *et al.* [220]). The former rely on the knowledge of the gradient of the objective function, either analytical or numerically estimated, to move in the

downhill direction. Numerical estimation of the gradient in $\mathbb{R}^n$ requires $O(n)$ evaluations of the objective function (Press *et al.* [220]), which alone makes this approach prohibitive.

The latter class advances without an explicit computation of the gradient. A well-known example of an algorithm of this class is the Nelder-Mead method (also known as the Downhill Simplex method, see Press *et al.* [220]). Note that it operates by evolving a simplex with $n + 1$ vertices in $\mathbb{R}^n$ and so the number of function evaluations grows as $O(n)$.

Gradient-based methods are rarely useful if the objective function is noisy or contains many local minima, for the estimation of the gradient is then unreliable and naïvely moving down the hill quickly leads to a local minimum. Another obstacle is called the "zero-gradient" problem, discussed in Miller *et al.* [182] and Learned-Miller [156], which occurs when the "current" solution happens to be on a flat plateau and so the gradient of the objective function is zero in the vicinity. This throws off course many downhill-descend algorithms (*e.g.* the Nelder-Mead algorithm), and especially those based on the direct estimation of the gradient (*e.g.* gradient descent (GD)). Miller *et al.* [182], suggest blurring of the images as a crude solution to overcome this problem.

The higher is the dimensionality of the search space, the more prone are the local optimisers to getting stuck in local minima. I found that Nelder-Mead algorithm becomes increasingly useless in "more than a few"[4] dimensions, but is quite robust for small-dimensional problems. For example, Nelder-Mead algorithm it is perfectly suited to the estimation of affine parameters, such as when initially aligning the images, as is done in Cootes *et al.* [67] (note however, that to estimate translation they use an exhaustive search first).

A common heuristic (used, for example, in Davies *et al.* [79], Twining and Marsland [279], Lötjönen and Mäkelä [169]), which allows to simultaneously reduce the dimensionality of the search space and to apply a comparatively fast local optimiser instead of a global one, is to perform optimisation along a small subset of dimensions first, then along a different subset of dimensions and so on, repeatedly. This heuristic is based on the assumption that along the selected few dimensions the objective function has a single global minimum in

[4]The exact number depends, of course, on the nature of the function being optimised. To illustrate, when optimising the piece-wise affine deformation model as in Section 3.5.2, the Nelder-Mead algorithm becomes useless if the dimensionality is higher than 6–8 (corresponding to 3–4 control points).

the vicinity of the current solution. This heuristic is analogous to the classical Powell's method (see Press [220] for description), where the optimisation is done along one axis at a time, and inherits its disadvantage of attaching to a local minima when the direction towards the true solution is not found along the selected dimensions.

For example, when aligning shapes, Davies *et al.* [79] deal with an enormous configuration space of dimensionality $12n_s \times 4^{k-2}$, where n_s is the number of shapes, with $12 \times 4^{k-2}$ parameters per shape. They found that robust and reliable global optimisation in the space of such dimensionality is difficult and they use the above heuristic to reduce the dimensionality of the search: they optimise a small number of parameters at a time using the classical Nelder-Mead algorithm (Press *et al.* [220]). The same heuristic is adopted by Twining and Marsland in [279] where they optimise the positions of a few control points at a time, in one image at a time (the optimisation method not specified).

Even with the above heuristic, optimisation is not trivial. This is why, when optimising the displacement of the deformation spheres' centres, Lötjönen and Mäkelä [169] employ a combination of the above heuristic and the brute force(!) approach. Several values are tried and the best one is then selected. Lötjönen and Mäkelä [169] report that GD method proved ineffective, as it easily gets stuck in local minima, and is outperformed by brute force optimisation of one deformation sphere at a time.

Cootes *et al.* [67], borrowing many ideas from Lötjönen and Mäkelä [169], also use the brute force approach, at least in the early stages of the registration, to estimate the optimal parameters of the elementary warps. They also report it to be less prone to sticking in local minima than downhill descend techniques. In the final stages of the registration, to refine the solution, Cootes *et al.* [67] perform the line search along the direction of the gradient.

Brute force optimisation is, of course, a poor choice for the optimiser for the same reason as SA or DIRECT are: it takes too many cost function evaluations to explore the space of solutions adequately. This number grows exponentially with the dimensionality of the search space, remarks Hicks [126] citing MacCormick and Isard [171].

There are a few exotic examples in the literature in which traditional continuous optimisation methods are not employed. In the highly original

paper by Kokkinos and Yuille [150], global shape models and articulated models are constructed by registering edge and ridge primal sketches, and an EM-style optimisation regime is used.

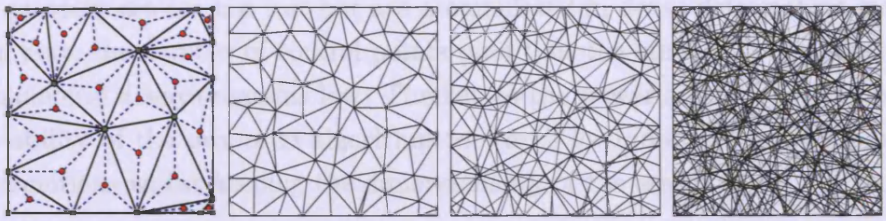
An innovative approach to apply discrete optimisation instead of continuous optimisation is proposed by Glocker *et al.* in [108]. There, to solve a non-rigid volume registration problem, the authors replace the continuous optimisation problem with a discrete one. The algorithm due to Glocker *et al.* [108] aims to optimally assign discrete labels, corresponding to predefined displacements, to control points on a grid, in the sense of minimising the Markov Random Field (MRF) energy. Smoothness is modelled by the edges between neighbouring vertices, and the cost of an assignment of labels to the grid nodes, once projected to the entire volume domain, serves as an objective function. The approach due to Glocker *et al.* [108], although promising, is applied only to pairs of images and their framework cannot be readily extended to operate on ensembles of images in a groupwise fashion. A conceptually similar treatment was proposed for the problem of stereopsis, which is intimately related to registration, for example in Kolmogorov and Zabin [151] as well as Yang *et al.* [297].

3.5 The Proposed Groupwise Registration Algorithm

Below, the main contribution of this chapter is presented: an efficient stochastic algorithm for groupwise non-rigid registration of image ensembles.

The input to the registration algorithm is a (possibly unordered) set of N images $\{\mathcal{I}_i, i = 1 \dots N\}$ of different examples of a deformable object or a deformable structure in an object. Automatically, without user intervention, dense spatial correspondences between the examples should be derived. Deformation fields, one for each image, define spatial correspondences between the images, by specifying where each pixel on the underlying object structure is located on that image.

As mentioned in Section 3.3, the problem of groupwise registration can be regarded as an optimisation problem. Its three components will be addressed next. Before explaining the optimisation regime, the deformation model and the objective function which is to be minimised will be defined.



(a) C_D and C_S (b) 1 mesh (c) 2 meshes (d) 8 meshes

FIGURE 3.4: (a) A dense mesh and its subset. (b)–(d) Superposition of random meshes.

3.5.1 Incrementally Learning Optimal Deformation

Let there exist a dense set of control points, C_D, and a sparse subset of C_D, $C_S \in C_D$. Suppose, for ease of explanation, that they are vertices of a triangular mesh describing the piece-wise linear deformation of an image. In Fig. 3.4a, C_D is all points, C_S is points on the solid lines. Let C_{D-S} denote the set of points in C_D that are not in C_S (in Fig. 3.4a they are solely on dashed lines). If it was possible to express the optimal position of points in C_{D-S} as a function of the optimal position of points in C_S, this would obviously yield a dimensionality reducing reparameterisation of the deformation: it would be possible to control more complex deformations with the same number of control points (increased resolution) or to control the deformations with a smaller number of control points keeping the same resolution (dimensionality reduction).

Unfortunately, such a function is not known in advance. Instead, the proposed algorithm incrementally learns and accumulates the optimal dense deformation everywhere *between* the sparse control points. The power of the proposed algorithm comes from this fact. The control points are used only when searching for an optimal piece-wise linear improvement for an already established *dense* deformation map. Moreover, instead of using only one set of control points, as other approaches (*e.g.* Cootes *et al.* [66,67]) do, a completely new set of control points at each iteration (see Alg. 3.2) is generated. This allows the algorithm to approximate more and more complex deformation fields as a sum of simple deformation fields (parameterised by control points at each stage), as the algorithm progresses. Figs. 3.4b to 3.4d illustrate this idea: as a new random parameterisation is used at each next iteration, the range of representable deformations progressively grows. It is critical to note that when a new set of control points is generated, the improvements obtained using

the previous set are not lost but are accumulated in dense deformation maps. Also note, the number of control points (and thus the number of optimisation parameters) always remains low. The key to the power stems therefore from the ability of the proposed algorithm to search for more and more complex deformations whilst keeping the dimensionality of the search space constantly low.

3.5.2 The Model of Deformations

The proposed algorithm represents the sought for deformation fields as a sum of randomly chosen piecewise linear basis functions. This is performed in an iterative fashion: each iteration, by adding a contribution of yet another random piecewise linear function to a deformation field, gradually improves the solution. This allows to keep the dimensionality of the search problem constantly low (by maintaining only a small number of control points at each iteration), while still computing smooth and detailed deformation fields in the end. Small number of parameters being optimised additionally acts as a regulariser.

The heuristic used by Cootes *et al.* [67][5], that of representing the deformation field as a sum of elementary warps of the form Eq. (3.17), is loosely comparable. The important difference is that the basis functions of Cootes *et al.* [67] are predefined and chosen *ad hoc* to approximately cover the space of deformations of the human face. The question of what is the range of admissible deformation under their model was not discussed by Cootes *et al.* [67], and it is easy to construct an example for which the basis of Cootes *et al.* [67] might be inappropriate. The proposed algorithm, on the other hand, makes no such assumptions and, given sufficient time, will automatically explore the space of all possible basis functions. This also excludes to possibility of getting stuck with a poor choice of deformation model (Sidorov *et al.* [243]).

Since the proposed algorithm computes the optimal deformation field by iteratively accumulating improvements to it, it is necessary that the deformation fields be additive.

The deformation fields are stored on a dense discrete grid, or *deformation map*, as in fluid models. The resolution of the deformation maps can be

[5]It should be also noted that Cootes *et al.* [67] was published one year after Sidorov *et al.* [243] on which this chapter is based.

arbitrary (for example, it can be the same as the resolution of the images, in which case every "pixel" of a deformation map stores the displacement, $(\Delta x, \Delta y)$, of the corresponding pixel in an image). The addition of deformations is then trivial: it can be performed by ordinary element-wise algebraic addition of the deformation maps.

The choice to store deformation fields on dense discrete grids is explained, therefore, by the following reasons. First, additivity of the fields is trivial, which allows for the final solution to be built gradually. Second, application of a deformation map to an image, or warping, is computationally cheap: it is sufficient, for each pixel of the image to look up the deformation map at the corresponding location to determine the displacement of the pixel. Third, a multi-resolution regime is easily possible: images can be registered at reduced resolution first, which is quicker, then the results (deformation maps) can be reused, by simply rescaling them, to refine the registration at higher resolution, and so on until the original resolution is reached. Finally, the regularity of the deformation map can be exploited to efficiently perform the above operations on a GPU (essentially, a parallel computer).

Parametric deformations are only used in Alg. 3.2, to parsimoniously represent the computed improvements. Each such improvement can be rasterised (converted to a discrete map) and added to the solution. Any of the deformation models based on control points discussed in Section 3.4.1 can, therefore, in principle be plugged into the proposed framework.

For computational efficiency, the piecewise affine model (Section 3.4.1), controlled by a sparse set of control points, was chosen to represent the incremental improvements to the solution. Although in some works this representation has been criticised for being insufficiently smooth (Cootes *et al.* [62]), the computational efficiency with which it can be manipulated outweighs this minor drawback (as acknowledged by Cootes *et al.* [67]). This is even more true if the GPU is employed: the operations of piece-wise affine interpolation is fundamental in computer graphics, and all, even very old, GPUs can perform it very quickly.

What is more important, is that because in the proposed algorithm the final solution is constructed by adding together many piecewise linear models, the resulting deformation fields can be arbitrarily smooth. And so, the choice of the piecewise affine model to represent incremental improvements is justified

by its computational efficiency and the fact that the incremental nature of the proposed algorithm does not lead to sacrifices in smoothness of the final solution (as will also be shown in Section 3.6).

Let $\mathcal{D}_{h\times w\times 2}$ denote a dense deformation map, with two components per pixel representing pixel displacements, $(\Delta x, \Delta y)$. In the exposition of the algorithm, two fundamental operations are needed. The first operation is the application of a deformation map to an image (possibly followed by an affine transform), or *warping*, and its inverse. Given an image $\mathcal{I}$, a dense deformation map $\mathcal{D}$, and an affine transform A, let the warped image $\mathcal{I}'$ be abstractly denoted as $\mathcal{I}' = W(\mathcal{D}, \mathrm{A}, \mathcal{I})$. This operation is illustrated in Fig. 3.5. Additionally, let $W^{-1}(\cdot)$ denote the inverse of this operation: $\mathcal{I} = W^{-1}(\mathcal{D}, \mathrm{A}, W(\mathcal{D}, \mathrm{A}, \mathcal{I}))$.

The second required operation is the generation of a dense deformation map from a sparse set of control points at which the displacements are specified. As discussed above, this is done by piecewise linear interpolation (see *e.g.* Berg *et al.* [21]). Given a set of n_p control points, with coordinates stored as columns in matrix $C_{2\times n_p}$, and the interpolated values at those points, stored as columns in matrix $\mathrm{V}_{2\times n_p}$, let the interpolated deformation map be abstractly denoted as $\mathcal{D}_{h\times w\times 2} = L(\mathrm{C}, \mathrm{V})$.

3.5.3 Objective function

Let $\mathcal{D}_i$ denote the corresponding deformation map, and A_i the corresponding affine transform for image $\mathcal{I}_i$. Then the groupwise objective function, F_{glob}, measuring the overall quality of alignment of the entire ensemble is defined as follows:

$$F_{\text{glob}}(\mathcal{D}_1, \ldots, \mathcal{D}_N) = \frac{1}{N}\sum_{i=1}^{N} G(\mathcal{I}_i, W(\mathcal{D}_i, \mathrm{A}_i, \mathcal{R}))\,. \tag{3.41}$$

This amounts to computing the average discrepancy, using a discrepancy function $G(\cdot)$, between every original image in the ensemble and the model of pixel intensities $\mathcal{R}$ warped to conform to each of the original images using the current estimate of $\mathcal{D}_i$ and A_i. This function measures how well the appropriately deformed model "explains" each of the original images. A minor detail omitted in Eq. (3.41) for clarity is this: when an affine transform is applied to $\mathcal{R}$, the result will in general only partially overlap with $\mathcal{I}_i$ and so $G(\cdot)$ should correctly compare the overlapping regions only.

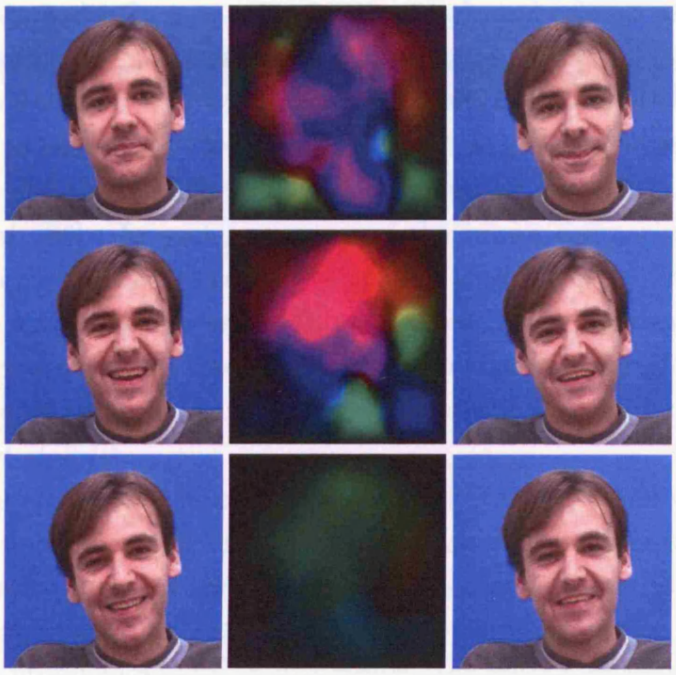


FIGURE 3.5: Samples from the data set [97] together with the computed deformation maps and shape-normalised images. In deformation maps, colour indicates direction, and brightness — the magnitude of the displacement.

It should be noted that for the purposes of registration only the arg min of the objective function is important, not its actual values. Therefore, the choice of the type of model $\mathcal{R}$ and the comparison function $G(\cdot)$ is governed by two factors: the objective function should be as cheap as possible to compute, and it should have the minimum at a point corresponding to a good registration.

As discussed in Section 3.4.2, efficiency may be achieved by visiting each image in turn and optimising its associated deformation map. The registration may then be regarded as repeated optimisation of *local* objective functions

$$F_{\text{loc}}(\mathcal{D}_i, \mathrm{A}_i, \Delta\mathcal{D}_i) = G(\mathcal{I}_i, W(\mathcal{D}_i + \Delta\mathcal{D}_i, \mathrm{A}_i, \mathcal{R}_i)), \tag{3.42}$$

where $\mathcal{R}_i$ is a local model computed as

$$\mathcal{R}_i = \frac{1}{N-1} \sum_{\forall j \neq i} W^{-1}(\mathcal{D}_j, \mathrm{A}_j, \mathcal{I}_j). \tag{3.43}$$

and $\Delta\mathcal{D}_i$ is the optimised improvement to the deformation map $\mathcal{D}_i$ obtained during previous iterations. Repeatedly optimising F_{loc} for each image in turn optimises F_{glob}.

The above model $\mathcal{R}_i$, is the simplest model of pixel colours: the average of the shape-normalised original images. This model works well in practice (as confirmed in Section 3.6, as well as by experiments of Cootes *et al.* [67]) and is very cheap to compute. The exclusion of the i-th image when computing the average serves to exclude a local minimum at zero $\Delta\mathcal{D}_i$, as done in Cootes *et al.* [66]. In such case, any of the image comparison functions discussed in Section 3.4.2 can in principle be used for $G(\cdot)$. For reasons of efficiency, a simple pixel-wise MAD, normalised by the number of pixels in the overlapping region, is used in the proposed algorithm for $G(\cdot)$. This was shown in the literature to work well (Sidorov *et al.* [243] and later confirmed by Cootes *et al.* [67]).

More importantly, as mentioned above, only the arg min of the objective function is important. In the experiments described in Section 3.6 values of the objective function using various choices for $G(\cdot)$ are plotted as the algorithm progresses. It is evident from these progress plots that optimising a MAD-based objective function also optimises the other measures (mean pixel stack entropy, MI, and NMI). Therefore, since the arg min of a MAD-based objective function is at the same point, or very close for all practical purposes, to that of other objective functions, as empirically verified in Section 3.6, the reason of computation efficiency prevails and MAD is chosen for $G(\cdot)$.

MI or NMI can be trivially substituted for $G(\cdot)$. In order to efficiently use the mean pixel stack entropy measure of Learned-Miller [156], a different kind of model $\mathcal{R}_i$ is required. Recall from Eq. (3.34) the efficient way of estimating pixel stack entropies. The model would then consist of sorted pixel stacks for all images except the i-th, and evaluation of the cost function would constitute a fast update of the model and evaluation of Eq. (3.34).

It should be noted that there is no shape constraint (a term dependent solely on $\mathcal{D}_i$) in the objective function. While various options for the shape terms have been proposed (Cootes *et al.* [67]), experiments in Section 3.6 as well as in Sidorov *et al.* [243] show that when enough features are present in the images no additional shape constraints (*e.g.* to encourage more smooth deformation maps) are needed. Even in "flat" regions smoothness is still achieved because the improvements to the deformation maps are strongly regularised. Additionally, the inclusion of a shape term into the objective function would necessitate a scaling coefficient in front of it, to account for

difference in units and to control the relative contribution of the intensity-based and the shape term. The need to tune this coefficient would have made the algorithm less automatic.

3.5.4 Optimisation regime

A naïve attempt to minimise F in Eq. (3.41) would be to exhaustively search in the space of all possible deformation maps, $\mathcal{D}_i$, for all n images. Instead, as shown below, the proposed algorithm visits each image in turn to find an optimal improvement for its corresponding deformation map using a small set of parameters, which leads to minimisation of Eq. (3.41). (If there are n images in the ensemble, and at each stage the improvement is controlled by k parameters, this is equivalent to optimising a nk-dimensional objective function along k dimensions at a time and is loosely analogous to Powell's classic optimisation method.)

The complete description of the registration procedure, which is summarised in Alg. 3.1, is given below. Assume the optimal affine transforms A_i have been found previously. The groupwise affine alignment, preceding the non-rigid stage, is done in the same fashion as the non-rigid alignment, described below in Alg. 3.1, except that search is performed for the optimal affine transformation parameters for each image, and instead of removing the embedding bias in line 14 the affine parameters are normalised so that the average translation and rotation across the ensemble is 0 and the average scaling is 1. Henceforth, assume that all images are affinely aligned.

The deformation maps $\mathcal{D}^a_{i_0}$ for all images are initialised to identity transform (line 2). The algorithm then operates by incrementally improving the accumulated deformation maps, $\mathcal{D}^a_{i_k}$, in an iterative fashion. The iterative body (lines 4–15) is repeated until no further improvement is possible. In order to avoid biasing the algorithm, the order in which the images in the set (and their corresponding deformation maps) are processed is randomised: at each iteration the set is randomly permuted.

When optimising deformations for each image $\mathcal{I}_i$, first compute an estimate of the texture model $\mathcal{R}_{i_k}$ by averaging all images $\mathcal{I}_j$ (except $\mathcal{I}_i$) transformed to the reference space using the deformation learnt at the previous stages $\mathcal{D}^a_{j_{k-1}}$ and the improvement $\Delta\mathcal{D}_{j_k}$ learnt at the current step. For computational

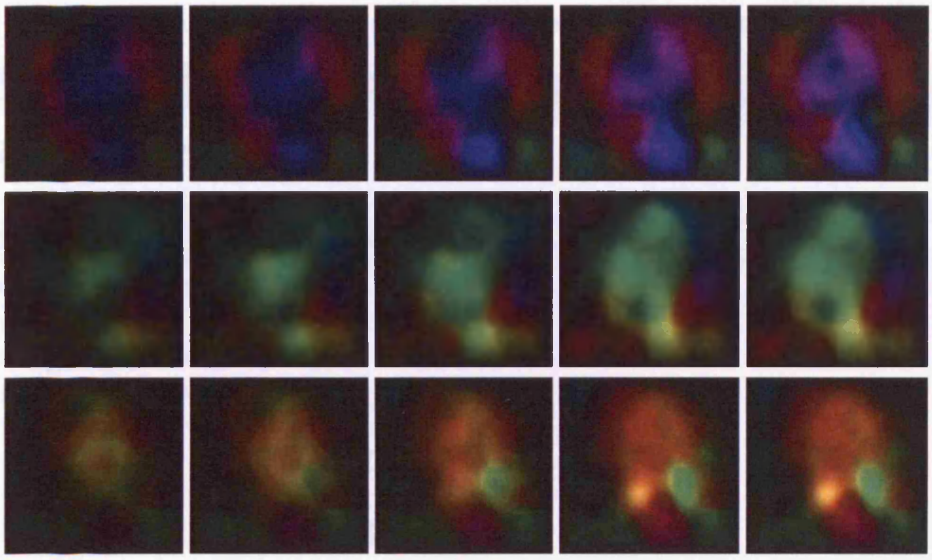


FIGURE 3.6: Evolution of the deformation map for selected images as the algorithm progresses.

efficiency this is accomplished by first computing the sum of all transformed images and corresponding masks (accounting for missing pixels due to affine transformation) in lines 5–6. Then, when visiting each image in turn the "current" image is subtracted from the sum, and its corresponding mask is subtracted from the sum of masks, and a weighted average is then computed (lines 9–10).

The optimal improvement $\Delta\mathcal{D}_{i_k}$ is then computed using Alg. 3.2 by minimising $G(\cdot)$ over the space of all possible improvements $\Delta\mathcal{D}$ (line 11). After the deformations of all images have been improved, the improvements are added to the previously learnt deformations (line 12) and the process repeats. The evolution of the deformation map for an image is illustrated if Fig. 3.6.

Line 14 serves to remove the deformation bias, the procedure discussed in Section 3.5.10. To save processing time, removal of the deformation bias can be performed less often than every iteration.

It is important that Alg. 3.2 is not allowed make a solution worse, it can only improve it.

Next, it is necessary to address the outstanding problem of minimising the objective function

$$G(\mathrm{I}_i, W(\mathrm{R}_{i_k}, \mathcal{D}^a_{i_{k-1}} + \Delta\mathcal{D})) \to \min_{\Delta\mathcal{D}} \tag{3.44}$$

in line 11 as a function of $\Delta\mathcal{D}$ which is exacerbated by its non-linear nature and many local minima.

3.5.5 Stochastic optimisation

Algorithm 3.1 Register a batch of images

Require: Images $\mathcal{I}_i$, $i \in \{1 \dots N\}$

1: $k \leftarrow 1$
2: $\mathcal{D}^a_{i_0} \leftarrow \mathbf{0}$, $\forall i \in \{1 \dots N\}$
3: **while** not happy **do**
4: Randomly permute the order of images.
5: $\mathcal{T} \leftarrow \sum_{i=1}^{N} W^{-1}(\mathcal{D}_{i_k}, \mathrm{A}_i, \mathcal{I}_i)$
6: $\mathcal{S} \leftarrow \sum_{i=1}^{N} W^{-1}(\mathcal{D}_{i_k}, \mathrm{A}_i, \mathbf{1}_{h\times w\times d})$
7: $\Delta\mathcal{D}_{i_k} \leftarrow \mathbf{0}$, $\forall i \in \{1 \dots N\}$
8: **for** $i = 1$ to N **do**
9: $\mathcal{S}' \leftarrow \mathcal{S} - W^{-1}(\mathcal{D}_{i_k}, \mathrm{A}_i, \mathbf{1}_{h\times w\times d})$
10: $\mathcal{R}_{i_k} \leftarrow (\mathcal{T} - W^{-1}(\mathcal{D}_{i_k}, \mathrm{A}_i, \mathcal{I}_i))_\bullet/(_\bullet\max(1, \mathcal{S}'))$
11: Using Alg. 3.2, compute
$\Delta\mathcal{D}_{i_k} \leftarrow \arg\min_{\Delta\mathcal{D}} G(\mathcal{I}_i, W(\mathcal{D}^a_{i_{k-1}} + \Delta\mathcal{D}, \mathrm{A}_i, \mathcal{R}_{i_k}))$
12: Learn improved deformation map:
$\mathcal{D}^a_{i_k} \leftarrow \mathcal{D}^a_{i_{k-1}} + \Delta\mathcal{D}_{i_k}$
13: **end for**
14: Remove deformation bias, see Section 3.5.10.
15: $k \leftarrow k + 1$
16: **end while**

SPSA is an attractive choice for the optimiser: it is capable of evading local minima due to its stochastic nature and, when adapted for the proposed framework, is orders of magnitude more efficient (Spall *et al.* [248]) than the traditional stochastic algorithms. Moreover, while in traditional gradient-based methods the number of function evaluations required to estimate the gradient at a point grows linearly with the dimensionality of the space, SPSA offers independence of the number of function evaluations at each iteration on the dimensionality of the space. An overview of the SPSA algorithm can be found in Spall [250], for completeness the idea of the algorithm is summarised below.

Let $f(\boldsymbol{\phi})$ be a real-valued function and $\boldsymbol{\phi}$ be a $p-$dimensional vector of parameters. Assume that only the direct measurements of $f(\boldsymbol{\phi})$ are available, but not of its gradient. Measurements of $f(\boldsymbol{\phi})$ might also be noisy, and p might

Algorithm 3.2 Optimise improvement $\Delta\mathcal{D}$ to deformation $\mathcal{D}^a_{i_{k-1}}$ of I_i into M_{i_k} using SPSA [249,250]

Require: R_{i_k}, $\mathcal{I}_i$, $\mathcal{D}^a_{i_{k-1}}$, $m_{\max}$, c_0, a_0, α, γ, A
1: $m \leftarrow 1$
2: **while** not converged and $m < m_{\max}$ **do**
3: Set gains $a_k \leftarrow \frac{a_0}{(A+m)^\alpha}$ and $c_m \leftarrow \frac{c_0}{m^\gamma}$
4: Select control points C_R randomly using FPS, see Section 3.5.7, and initialise $\hat{\boldsymbol{\phi}}_m \leftarrow \mathbf{0}$
5: Generate $\boldsymbol{\delta}_m$, $\delta_{m_i} \leftarrow$ Bernoulli(-1 or 1)
6: Using $G(\hat{\boldsymbol{\phi}}_m) := G\left(\mathcal{I}_i, W(\mathrm{R}_{i_k}, \mathcal{D}^a_{i_{k-1}} + L(\hat{\boldsymbol{\phi}}_m, C_R))\right)$
and ensuring that $\mathcal{D}^a_{i_{k-1}} + L(\hat{\boldsymbol{\phi}}_m, C_R)$ is invertible (else reject it), see Section 3.5.9
estimate $\mathbf{g}_m \leftarrow \mathbf{g}(G(\hat{\boldsymbol{\phi}}_m), \hat{\boldsymbol{\phi}}_m, c_m, \boldsymbol{\delta}_m)$, see Eq. (3.45)
7: Update $\hat{\boldsymbol{\phi}}_{m+1} \leftarrow \hat{\boldsymbol{\phi}}_m - a_m \hat{\mathbf{g}_m}$, see Eq. (3.46)
8: $m \leftarrow m + 1$
9: **end while**
10: **return** The optimal $\Delta\mathcal{D} \leftarrow L(\hat{\boldsymbol{\phi}}_m, C_R)$

be very large. The aim is to minimise $f(\boldsymbol{\phi})$ to find $\underline{\boldsymbol{\phi}} = \arg\min f(\boldsymbol{\phi})$. Let $\boldsymbol{\delta} = (\delta_1, \delta_2, \ldots, \delta_p)^T$ be a vector of independent random variables with symmetric Bernoulli distribution: $\delta_i = \pm 1$ and $\Pr(\delta_i = 1) = \Pr(\delta_i = -1) = 1/2$.

Let $\hat{\mathbf{g}}(\boldsymbol{\phi})$ denote the stochastic approximation of the gradient $\mathbf{g}(\boldsymbol{\phi})$:

$$\hat{\mathbf{g}}(f(\cdot), \boldsymbol{\phi}, c_k, \boldsymbol{\delta}_\mathbf{k}) = \begin{pmatrix} \frac{\left(f(\hat{\boldsymbol{\phi}}_k + c_k\boldsymbol{\delta}_\mathbf{k}) - f(\hat{\boldsymbol{\phi}}_k - c_k\boldsymbol{\delta}_\mathbf{k})\right)}{2c_k\delta_{k_1}} \\ \vdots \\ \frac{\left(f(\hat{\boldsymbol{\phi}}_k + c_k\boldsymbol{\delta}_\mathbf{k}) - f(\hat{\boldsymbol{\phi}}_k - c_k\boldsymbol{\delta}_\mathbf{k})\right)}{2c_k\delta_{k_p}} \end{pmatrix}, \tag{3.45}$$

and let $\hat{\boldsymbol{\phi}}$ denote the "current" estimate for $\underline{\boldsymbol{\phi}}$.

The SPSA algorithm incrementally updates $\hat{\boldsymbol{\phi}}$ by the following process:

$$\hat{\boldsymbol{\phi}}_{k+1} = \hat{\boldsymbol{\phi}}_k - a_k \hat{\mathbf{g}}(\hat{\boldsymbol{\phi}}_k). \tag{3.46}$$

The gain sequences c_k and a_k are chosen as follows:

$$a_k = \frac{a_0}{(A+k)^\alpha} \qquad \text{and} \qquad c_k = \frac{c_0}{k^\gamma}. \tag{3.47}$$

Note that at each iteration k only *two* $= O(1)$ evaluations of f are required, as opposed to $O(p)$ in traditional gradient-based methods. Extensive convergence theory (Maryak and Chin [178]) establishes performance guarantees for SPSA and shows that $\hat{\boldsymbol{\phi}}_k \to \arg\min f(\boldsymbol{\phi})$ as $k \to \infty$.

Alg. 3.2 summarises the proposed adaptation of SPSA for the task at hand. As stochastic algorithms are notoriously hard to tune, the issue of tuning SPSA will be addressed next. It should be noted that the maximal number of iterations in Alg. 3.2 should be limited (to $m_{\max}$) and instead Alg. 3.2 should be called more often, in order to allow for the information to faster propagate between images by more frequently updating the texture model. In practice, values $m_{\max} = 10 \ldots 50$ work well.

In Alg. 3.2, the tuning parameters c_0 and a_0 are measured in units of image size (say, pixels) and control the "greediness" of the algorithm — larger values correspond to less greedy search. The parameter c_0 is chosen to be about 1–4% of the image size and experimentally choose a_0 to be of the same order of magnitude. The decay parameters α and γ are set to the theoretically optimal values $\alpha = 0.602$ and $\gamma = 0.101$ as discussed in Spall [249] which also covers the choice of tuning parameters in various settings.

3.5.6 Normalisation of Images

While it is possible to perform registration of images using the RGB intensities of pixels directly, it has been reported in Cootes *et al.* [67] that better results may be achieved if the edge information in images is used more explicitly as well as if some normalisation is used to remove the effects of lighting variation. This has been known before; indeed, the invariance to lighting changes and, in general, insensitivity to modality, is a desirable property of algorithms for registration of medical imagery, and is achieved, for example, by using MI-based cost functions (Pluim *et al.* [213], Maes *et al.* [172], Viola and Wells [287]). Utilisation of edge information for groupwise registration has been shown to be useful in Kokkinos and Yuille [150].

Departing from these observations, Cootes *et al.* [67] discuss various forms of image representation that enhance the quality of registration. Global linear normalisation is considered in Cootes *et al.* [67], akin to the approach of Cootes *et al.* [61], as well as local normalisation and incorporation of the gradient information as some of the image channels. They touch on the subject very briefly, without giving detailed instructions nor providing complete formulæ and parameters for this procedure. So, below it is necessary to explicitly summarise what was experimentally found to work well in the experiments conducted in Section 3.6.

Let r be the window radius for local normalisation. It was found experimentally that for face images r = 10–30% of the image width is a good value (80–160 pixels for a 512-by-512 image). Then let

$$\mathrm{K}_{2r+1\times 2r+1} = \frac{1}{(2r+1)^2}\mathbf{1}_{2r+1\times 2r+1} \tag{3.48}$$

be the averaging kernel (a circular kernel may be used instead). Then, using the notation from Table 2, the local average A of an image I (for each pixel, the average of all pixels within a square window of radius r), and the local average of the squared image Q are found as

$$\mathrm{A} = \mathrm{I} \circledast \mathrm{K} \quad \text{and} \quad \mathrm{Q} = (\mathrm{I} \bullet \mathrm{I}) \circledast \mathrm{K}. \tag{3.49}$$

The local bounded standard deviation is then computed as

$$\mathrm{S} = {}_\bullet\max\left({}_\bullet\sqrt{\mathrm{Q} - (\mathrm{A} \bullet \mathrm{A})}, \sigma_{\min}\right). \tag{3.50}$$

Clipping with the parameter $\sigma_{\min}$ serves to reduce the amplification of noise in "flat" regions. Experiments show that good values are $\sigma_{\min}$ = 0.15–0.25, assuming the pixel intensities are in the range 0.0–1.0. The locally normalised image is finally obtained as

$$\mathrm{N} = (\mathrm{I} - \mathrm{A})_\bullet / \mathrm{S}. \tag{3.51}$$

For multi-channel images this normalisation is performed on each channel independently. Further, let $\{\mathrm{G}_x, \mathrm{G}_y\} = \nabla\mathrm{N}$ be the gradient of the normalised image N. Smoothing the gradient by convolution with a smoothing kernel and scaling the result to compensate for difference in the units yields

$$\mathrm{G}'_x = \gamma(\mathrm{G}_x \circledast \mathrm{G}^\sigma_w) \quad \text{and} \quad \mathrm{G}'_y = \gamma(\mathrm{G}_y \circledast \mathrm{G}^\sigma_w). \tag{3.52}$$

In the case of face images, the value for σ in Eq. (3.52) of about 1% of the image size (4-6 pixels for a 512-by-512 image) work well with a Gaussian mask of size w = 3% of the image size (about 5 and 15 pixels respectively for a 512-by-512 image).

The resulting multi-channel image, $\mathcal{I}'$ is composed by concatenation of N (converted to grayscale if necessary), G'_x and G'_y, so that $\mathcal{I}'(:,:,1) = \mathrm{N}$, $\mathcal{I}'(:,:,2) = \mathrm{G}'_x$ and $\mathcal{I}'(:,:,3) = \mathrm{G}'_y$. The scaling factor γ in Eq. (3.52) serves

(a) Original image. (b) Normalised image. (c) Smoothed gradient. (d) Combined image (false colour).

FIGURE 3.7: Illustration of the image preparation process.

to scale the values of the smoothed gradient relative to the intensity values N and can be adjusted to control the relative importance of the gradient and intensity information in the registration process. Typical values of γ range from 0.1–10.0 depending on the type of imagery. Figure 3.7 illustrates this process. On the left, Fig. 3.7a, the original image is shown; the locally normalised image, preserving the important features, but agnostic of lighting variation is shown in Fig. 3.7b; the smoothed gradient (colour-coded, brightness indicates the magnitude and colour indicates direction) is seen in Fig. 3.7c; finally, Fig. 3.7d shows the complete result of the image preparation process, with the normalised intensity and the gradient components combined into a single false-colour image (with $\gamma = 2.0$).

3.5.7 Selection of Control Points

The question of selecting the positions of the control points in a sensible fashion will now be addressed.

In the very simplest case the control points may be positioned at the nodes of a regular grid and then randomly perturbed. In Cootes *et al.* [66] a better performance is reported if the control points, placed initially on a regular grid, are moved to nearby strong edges and those points that happened to be in the areas of low variance are removed. This is unsurprising: removing the control

points from low variance, "flat", regions reduces the dimensionality of the optimisation problem at virtually no cost; and positioning the control points around the strong edges is a good heuristic if one wants to cover the most important areas of the image using as few control points as possible, again reducing the dimensionality of the optimisation problem. However, in images of biological objects, such as faces or brain imagery, where pixel intensities usually vary smoothly, edges tend to be an unreliable and unstable feature. Additionally, the above approach of simply moving the control points to the nearby edges leaves little control over the relative density of control points in "interesting" and "flat" regions.

A more elegant solution to the problem of sensible positioning of control points based on the concept of geodesic farthest point sampling is proposed below. An adaptive FPS strategy that yields higher density of control points in areas with finer details and covers the smoother parts less densely has been first proposed in Eldar *et al.* [95]. The idea of using farthest point sampling strategy to intelligently sample images has been successfully used in Bougleux *et al.* [35], for image compression and approximation. These approaches are made feasible by the advances in Fast Marching (FM) methods (see Sethian [238] for a comprehensive overview, as well as Sethian [237], and also the discussion in Chapter 4) which, among many other things, allow to efficiently compute geodesic distances on discrete scalar fields. Specifically, given a scalar cost function $F(\mathbf{y})$ (a scalar field) with positive values, the geodesic distance $G(\mathbf{y}) = g(\mathbf{x}, \mathbf{y})$ to a point $\mathbf{y}$ from a starting point $\mathbf{x}$ satisfies the Eikonal equation (Arnold [12])

$$\|\nabla G(\mathbf{r})\| = F(\mathbf{r}), \mathbf{r} \in \Omega \qquad \text{(subject to } G|_{\partial\Omega} = 0), \tag{3.53}$$

which on discrete domains can be efficiently solved using the FM methods.

Given a scalar field $F\colon \mathbb{R}^n \to \mathbb{R}$, and an arbitrary smooth curve C, bijectively parameterised as $\mathbf{r}\colon [a, b] \to C$, with $\mathbf{r}(a)$ and $\mathbf{r}(b)$ corresponding to endpoints of C, the total effect of the field F along the curve C can be expressed as a line integral

$$L(C) = \int_C F\, ds = \int_a^b F(\mathbf{r}(t))\|\mathbf{r}'(t)\|\, dt. \tag{3.54}$$

Among all possible curves joining the points $\mathbf{x} = \mathbf{r}(a)$ and $\mathbf{y} = \mathbf{r}(b)$, the curve Γ that has the minimum length $d(\mathbf{x}, \mathbf{y}) = L(\Gamma)$ is called the *geodesic curve*. If the

(a) Original image.

(b) Cost of movement (smooth gradient) with control points superimposed.

(c) Original image with control points superimposed.

FIGURE 3.8: Seeding control points with the FPS strategy. The number of control points ($N = 1000$) is greatly exaggerated to illustrate the tendency. $\varepsilon = 0.20$.

"cost of movement" through $\mathbb{R}^n$ is used as the field F, the physical meaning of Eq. (3.54) becomes the length of the optimal path, which has a tendency to pass through areas where F is small.

If F is selected to represent the "cost of movement" through an image, such that F has higher values in the areas with fine detail and lower values in flat regions, it is then possible to use the FPS algorithm to seed the control points on the image, the farthest distance being understood in the geodesic sense on the field F, as proposed in Bougleux *et al.* [35]. This will amount to seeding more points in the areas of interest and fewer points in flat regions.

The simplest choice of F would be simply the magnitude of the gradient of a smoothed image, essentially emphasising the edges, which was found to work well in the experiments. A more sophisticated metric would be some per-pixel local statistical measure of "interest", such as local variance or entropy in the neighbourhood of each pixel.

Figure 3.8 illustrates this idea. On the left, Fig. 3.8a, an image of the human face is shown. In the middle, Fig. 3.8b, the control points generated using the FPS strategy are superimposed on top of the colour-coded values of the field F, whose values are stored in matrix F, which in this example is set to be

$$\mathrm{F} = \mathrm{rescale}_0^1\left(\bullet \|\nabla(\mathrm{I} \circledast \mathrm{G}_w^\sigma)\|\right) + \varepsilon, \tag{3.55}$$

using the shorthand notation from Table 2. By varying the parameter ε

Algorithm 3.3 Perform greedy FPS sampling (*e.g.* Eldar *et al.* [95]).

Require: $\mathrm{D}_{m\times m}$ — a dissimilarity, or distance, matrix. N — number of samples to draw, $1 \leq N \leq m$.

1: $\mathbf{s}_{N\times 1} \leftarrow \mathbf{0}_{N\times 1}$, $\mathbf{s}(1) \leftarrow \mathrm{randn}(1 \ldots m)$
2: $\mathbf{d}_{m\times 1} \leftarrow \mathbf{1}_{m\times 1}\infty$
3: **for** $i = 1$ to $N - 1$ **do**
4: **for** $j = 1$ to m **do**
5: $\mathbf{d}(j) \leftarrow \min(\mathbf{d}(j), \mathrm{D}(\mathbf{s}(i), j))$
6: **end for**
7: $\mathbf{s}(i+1) \leftarrow \arg\min(\mathbf{d})$
8: **end for**
9: **return** indices of selected samples, $\mathbf{s}$.

in Eq. (3.55) it is possible to finely control the degree to which the FPS strategy prefers edges over flat regions, as illustrated in Fig. 3.9. It is important to note that it is essential that the selection of control points with FPS is random, in order to allow the algorithm to explore the space of all possible configurations of control points as the registration progresses.

Since FPS strategy is used several times in this chapter and in Chapter 4, it is convenient to write down the FPS algorithm abstracting from the nature of relative "distances" being maximised. The simplest, yet very well working in practice, form of FPS is a greedy algorithm (*e.g.* Eldar *et al.* [95]). It is presented abstractly in Alg. 3.3 and greedily samples N farthest samples, the "distances" between which are given in matrix D (or, more generally, a distance function of two variables).

In Cootes *et al.* [67] the total number of control points used is $16 \times 20 = 320$. They investigate the quality of registration as a function of the number of control points used, and conclude that, at least for face images, noticeable degradation of quality occurs if the number of control points is less than $\approx 8 \times 8 = 64$, but also that beyond a certain density, $\approx 11 \times 11 = 121$, no further improvement is observed, thus giving the golden number of control points in the region $\approx$ 60–120. Note, however, that in Cootes *et al.* [67] deformations are defined solely by the displacements of the control points. In the proposed algorithm this is not the case, because the final deformation maps are superpositions of simpler deformation maps, and so are capable of modelling more complex deformations with smaller number of control points. The estimation of Cootes *et al.* [67] is thus an overly conservative upper bound on the number of control points

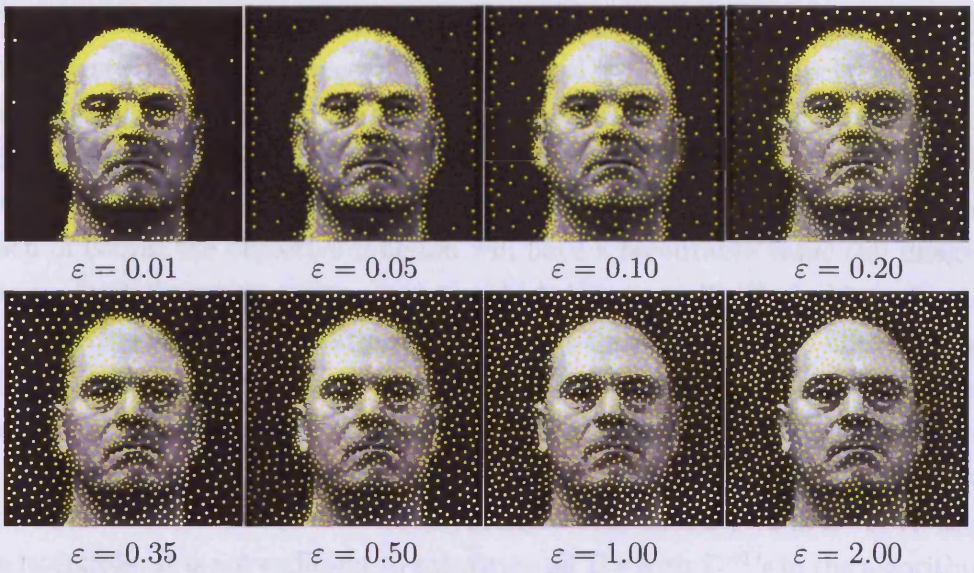


FIGURE 3.9: The effect of varying ε in Eq. (3.55) on control points generation. The number of control points ($N = 1000$, same in all images) is greatly exaggerated to show the tendency.

needed. Unlike with methods of Cootes *et al.* [66, 67] which rely on placing control points on a grid in order to adequately cover the entire RoI, and so do not allow an arbitrary number of control points to be used, with the FPS-based strategy the number of points can be gradually varied from 1 to infinity. It has been shown in Cootes *et al.* [66] that it is preferred to choose control points that are on strong edges and not on flat areas.

3.5.8 Target vs. Reference Frame

There are two ways in which the images can be compared with the evolving reference. The first way involves deforming the images, using the current estimate of the deformation fields, to bring them into alignment with the current estimate of the reference. The second way involves deforming the reference image to align it with each of the original images. For the sake of brevity, call these approaches: "comparison in the reference frame" and "comparison in the target frame" respectively. They have both advantages and disadvantages which are discussed next.

Comparison in the reference frame involves only the application of the deformation maps $\mathcal{D}$, which are readily available in Alg. 3.2 and which are used to compute the reference image in the first place, to the original images.

This is therefore a computationally relatively cheap operation and is easy to implement. The disadvantage of this is method is that unless special measures are taken, it is possible that the registration algorithm may converge to a trivial solution. For example, in such a failure mode, the algorithm may produce such deformation maps that map all images into a point in the reference frame; then of course the objective function will have a favourable value (all images are successfully registered to a point), which is not at all a desirable outcome.

Comparison in the target frame can be used instead, to alleviate the above problem. This, as Cootes *et al.* [67] put it, can be understood as "explaining" the pixels in the original image using the model (estimate of the reference). To use this method, the algorithm needs to know the inverse transformations, $\mathcal{D}^{-1}$. Moreover, the deformations themselves must obviously be invertible. It is not sufficient to substitute all $\mathcal{D}$'s with $\mathcal{D}^{-1}$'s in the algorithm and deal directly with the inverted deformations, because the forward transform is still required to compute the reference image. Experimental confirmation that comparison in the target frame is indeed better, is given in Cootes *et al.* [67].

3.5.9 Invertibility of Deformation Maps

The question of the invertibility of deformations will now be discussed. The deformation is invertible if it is a bijective mapping: each point in the deformed image corresponds to only one point in the original image and vice-versa (Tiddeman *et al.* [270]). There are two ways to ensure that the deformations are always invertible. First method is to use such reparameterisations of deformations for which there are known analytical bounds on the values of parameters for which the deformation remains invertible, and to ensure that the combinations of such deformations are also invertible. The latter requirement, that the combinations of deformations are invertible, is important for incremental accumulation of the final result out of multiple improvements to the deformations. This is discussed in Section 3.4.1.

This approach is not readily suitable for the proposed framework. The second method to ensure invertibility of deformations is found in Tiddeman *et al.* [270]. Its advantage is that it is independent of the nature of the deformation mechanism being used, as long as the deformation field can be rasterised into a deformation map, as is exactly the case in the proposed framework.

Suppose a transformation is defined by two functions, one for each coordinate, $T_x(\mathbf{r})\colon \mathbb{R}^2 \to \mathbb{R}$ and $T_y(\mathbf{r})\colon \mathbb{R}^2 \to \mathbb{R}$ that map the points

$$\mathbf{r} = (x, y) \to \mathbf{r}' = (T_x(\mathbf{r}), T_y(\mathbf{r})). \tag{3.56}$$

Tiddeman *et al.* [270] show, citing Meisters and Olech [181], that the so defined deformation is invertible if and only if the Jacobian determinant

$$J = \frac{\partial T_x}{\partial x}\frac{\partial T_y}{\partial y} - \frac{\partial T_x}{\partial y}\frac{\partial T_y}{\partial x} \tag{3.57}$$

is positive. This argument is also extended to higher dimensions in Tiddeman *et al.* [270].

This fact can be used in Alg. 3.2 when evaluating hypothetical improvements whether they, when added to the accumulated deformation map, will render the deformation man non-invertible. In such case the hypothesis is rejected.

Alternatively, Alg. 3.2 may be allowed to produce "slightly" non-invertible deformation maps and the approach described in Tiddeman *et al.* [270] can then be applied, every few iterations, to recover from this failure mode by forcing the non-invertible deformation maps to the closest invertible ones.

3.5.10 Removing Deformation Bias

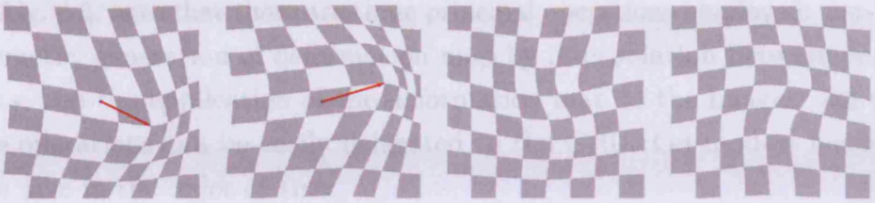
It is possible that during the non-rigid registration stage the correspondences between the images and the common reference space may become systematically biased, which is equivalent to common reference space becoming distorted. If not handled correctly, this effect might become a runaway process and completely ruin the registration.

To preclude the above problem from happening, the deformation bias is removed periodically (not necessarily at each iteration) in Alg. 3.1 in 14 by adjusting the improvements $\mathcal{D}_i$, so as to annihilate the bias.

This is done in two steps. For each deformation map $\mathcal{D}_i$ its inverse $\mathcal{D}_i^{-1}$ is computed (such that $W(\cdot, \mathcal{D}_i^{-1}) = W^{-1}(\cdot, \mathcal{D}_i)$. The inverse deformation maps are then averaged and the average subtracted from each:

$$\tilde{\mathcal{D}}_i^{-1} = \mathcal{D}_i^{-1} - \frac{1}{N}\sum_{i=1}^{N}\mathcal{D}_i^{-1}. \tag{3.58}$$

Finally, the biasless deformation maps are computed by inverting the $\tilde{\mathcal{D}}_i^{-1}$. Fig. 3.10 illustrates the idea. Two images are deformed (Fig. 3.10a, Fig. 3.10b),



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

FIGURE 3.10: Removing deformation bias. (a), (b) — deformations applied to two different images, in both cases biased to the right. (c), (d) — result of removing the bias.

using a deformation disk (as in Eq. (4.14)) placed at the centre, by $(128, 64)^T$ and $(160, -48)^T$ units respectively. Observe that the average deformation is biased to the right. The result of removing the bias using the above procedure is shown in Fig. 3.10c and Fig. 3.10d.

3.5.11 Applying Deformation Maps to Images

Invertibility of deformations, discussed above, is also important even when just applying them to images. Suppose there is a transformation W that maps points in image $\mathcal{I}$ to points in image $\mathcal{I}'$. Then the image $\mathcal{I}'$ can be computed by resampling:

$$\mathcal{I}'(\mathbf{r}) = \mathcal{I}(W^{-1}(\mathbf{r})), \qquad \forall \text{ pixel coordinates } \mathbf{r}. \tag{3.59}$$

Resampling is usually preferred, since it avoids holes and overlaps in the target image $\mathcal{I}'$ as the intensity values are computed for each of its pixels sequentially. A technique for forward warping, without the knowledge of the inverse transformation, is called "splatting" and is more computationally expensive.

Note that when sampling images, a good sampling scheme must be used (such as bilinear sampling) to avoid small plateaus in the objective function (and so potential zero-gradient problems) caused by discrete nature of the images.

3.5.12 GPU-based Implementation

The proposed registration framework is amenable to an implementation harnessing the power of modern programmable graphics hardware. Observing Alg. 3.1

and Alg. 3.2, note that there are three principal operations employed: per-pixel arithmetic, generation of deformation map by interpolation between control points, and the application of the deformation map to the images. All these three operations can be easily delegated to the GPU. On modest hardware, those take in the order of 1ms.

Surprisingly, the most expensive operation is the aggregation of the value aggregation across the pixels. This requires a technique know as "gather" in literature, which requires $O(log(w))$ passes, where w is the image width. On modest GPUs this takes in the order of 10ms per evaluation of the cost function.

Technical improvements here include simultaneous evaluation of the cost function at two values in SPSA (to save an extra "gather" operation); combination of deformation map generation and application into one step.

3.6 Experiments

In this section, the experiments that were carried out to evaluate the efficiency of the proposed optimisation framework are described.

Several registration experiments were conducted with artificial and real data, including inter-subject registration, and performance of the proposed algorithm was compared to the ground truth, manual annotation, and the state of the art methods.

The following data sets were used during the experiments. The publicly available FGNET "Talking Head" frontal face images of a single person, with manual annotation [97]. The publicly available `xm2vts` (Session 1) frontal face images of multiple individuals [4], also with manual annotation [5]. Note that in this data set of notorious difficulty 41% of images include facial hair, glasses, and other features making the registration very difficult. The publicly available IMM face database (only frontal images) with manual annotation [193] was also used. As an example of non-facial imagery, binary images of handwritten zeros and twos used in Miller *et al.* [182] from NIST data base [118] were used. Data captured by the author and artificial data sets were also used (see below).

3.6.1 Registration of Synthetic Data

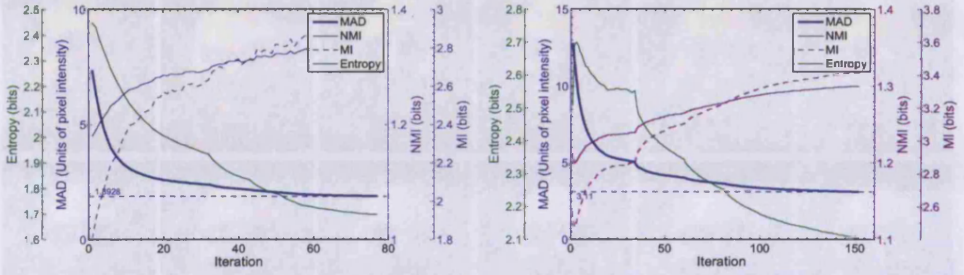
This experiment aims to evaluate the performance of the proposed algorithm with respect to the ground truth. One way to compare the results of the registration with the ground truth is to generate a synthetic dataset, for which the deformations that bring images into alignment are known *a priori*. In this experiment, one image was taken as a template, then deformation fields were randomly generated and applied to this template, thus producing a synthetic dataset with the known ground truth deformation field for each image.

The artificial deformation fields were generated by a superposition of 8 deformation balls as in Eq. (3.19), with the kernel from Eq. (3.21), each of radius 1/4 of the image width. The centres of the deformation balls were randomly chosen over the RoI of the image (uniformly distributed). The displacement directions were also chosen randomly (uniformly distributed) and the magnitudes of the displacement were set to 0.4 of the ball radius. This way, 64 artificially deformed images were generated.

It should be noted that, in general, registration of the above synthetic data set will not yield the original images *exactly*, for two reasons. The first reason is that warping of the template images itself introduces errors, in other words $W^{-1}(\cdot,\cdot,W(\cdot,\cdot,\mathcal{I})) \neq \mathcal{I}$, but only approximately so. The second reason is more important and has to do with the warping bias. Unless the average warp across all images is zero, the resulting reference image, after the algorithm has converged, will not be the same as the original template image. (Imagine two synthetic images produced by warping the template image "to the right". The algorithm then will converge to a reference that is also warped "to the right".)

The former effect can be alleviated by comparing the results of the registration with the images warped forward and then back, to account for quality loss during warping. The latter effect can be approximately alleviated by ensuring that the average warp is zero, as detailed in Section 3.5.10.

The algorithm was run on two artificial ensembles produced from the "Dave's Head" and the "Chequerboard" image. Selected images from the artificial ensembles are shown in the first row in Fig. 3.12 and Fig. 3.13. The second row shows the reconstructed images, in other words the deformed images warped to the reference frame after the correct deformation fields were established. The third row shows the absolute difference between each reconstructed image



(a) "Dave's Head" (b) "Chequerboard"

FIGURE 3.11: Registration progress for the synthetic data sets.

and the original template image (scaled to $0 \ldots 1$ for display). The fourth row shows the ground truth deformation fields that were used to produce the synthetic images from the template. Below them, in the fifth row, are shown the deformation fields computed by the algorithm. Finally, in the sixth row are shown the differences between the computed deformation fields and the ground truth (scaled to $0 \ldots 1$ for display).

The progress plots for the both ensembles are found in Fig. 3.11. (When reading the results, remember that NMI is a measure of similarity (bigger values are better), while the other metrics are measures of dissimilarity (smaller values are better)).

The affine alignment stage was not performed in this experiment. The algorithm was stopped when the relative change in the objective function value became less than 0.1%. (The results would be improved further if the algorithm was allowed to run longer.)

The same two artificial data sets were registered using the algorithm of Cootes *et al.* [67]. The results of running the Cootes *et al.* [67] algorithm are shown in Fig. 3.14 and Fig. 3.15.

The MAD between the ground truth images and their reconstructed counterparts were computed. Additionally, the average Euclidean spatial reconstruction error was computed. The latter is defined as

$$E = \frac{1}{N|\Omega|} \sum_{i=1}^{N} \sum_{\forall x,y \in \Omega} \sqrt{(\mathcal{D}_i^{\text{ref}}(y,x,1) - \mathcal{D}_i(y,x,1))^2 + (\mathcal{D}_i^{\text{ref}}(y,x,2) - \mathcal{D}_i(y,x,2))^2}, \tag{3.60}$$

which amounts to averaging the Euclidean distance between the ground truth deformation maps, $\mathcal{D}_i^{\text{ref}}$, and the reconstructed deformation maps, $\mathcal{D}_i$, for all

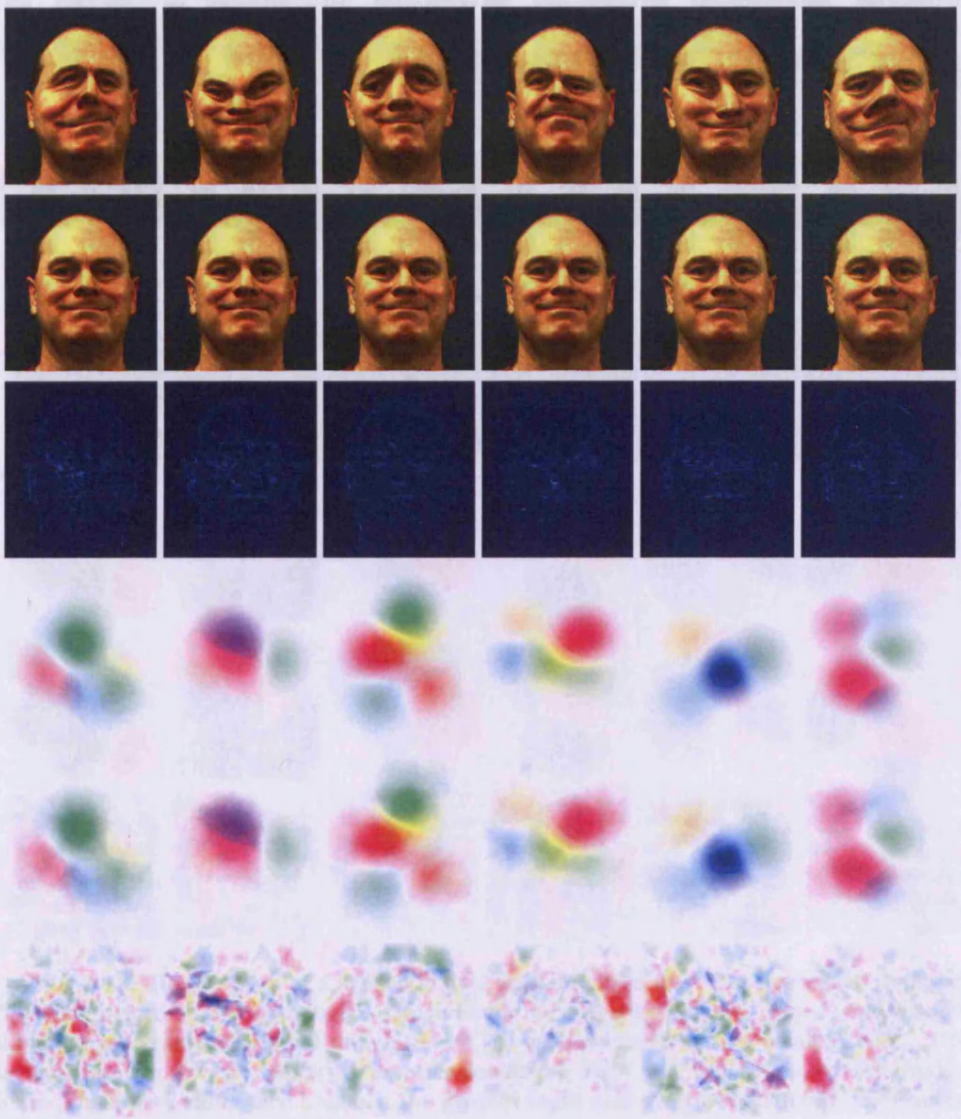


FIGURE 3.12: Registration of a synthetic data set built from the "Dave's Head" template. *First row:* deformed template image. *Second row:* reconstructed images. *Third row:* absolute difference between the original template and the reconstructed images (scaled to $0 \ldots 1$ for display). *Fourth row:* ground truth deformation fields. *Fifth row:* reconstructed deformation fields. *Sixth row:* difference between the ground truth and the reconstructed deformation fields (scaled to $0 \ldots 1$ for display).

pixels. The results for both artificial data sets and for both algorithms are summarised in Table 3.1 and Table 3.2.

Comparison shows that the proposed algorithm significantly outperforms the method of Cootes *et al.* [67] both in terms of spatial and intensity reconstruction

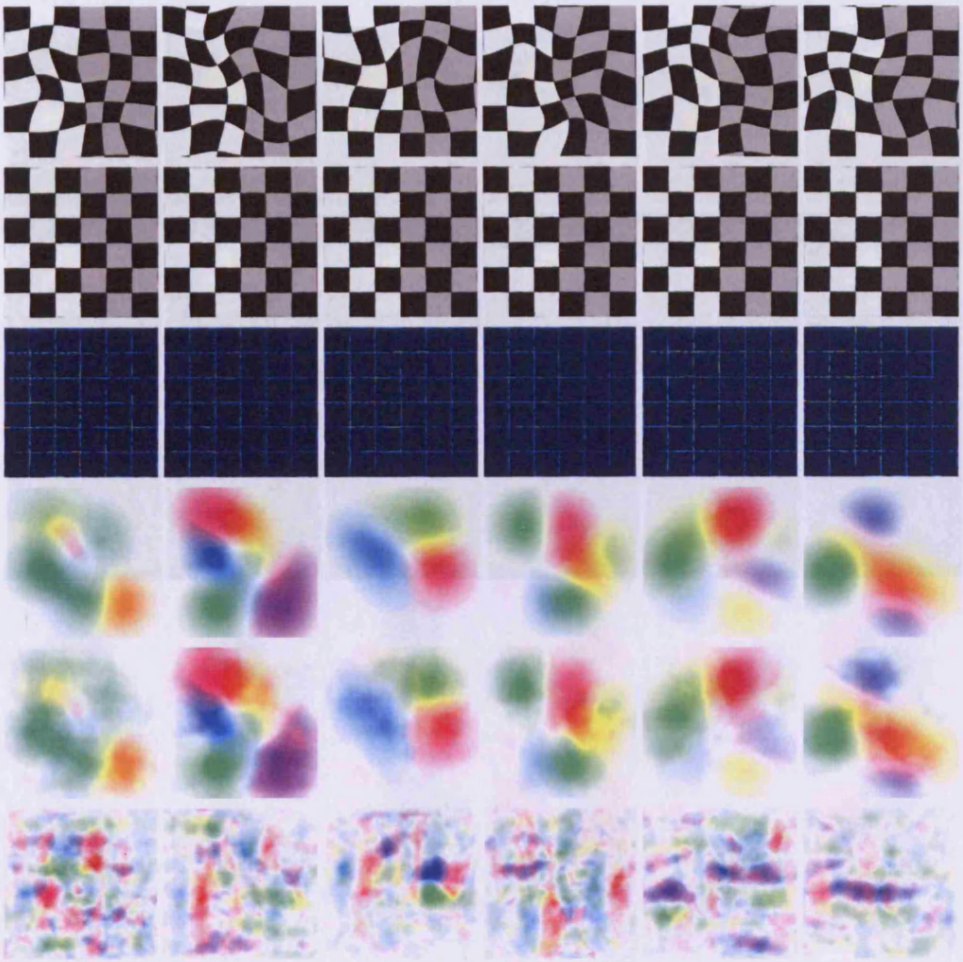


FIGURE 3.13: Registration of a synthetic data set built from the "Chequerboard" template. *First row:* deformed template image. *Second row:* reconstructed images. *Third row:* absolute difference between the original template and the reconstructed images (scaled to $0 \ldots 1$ for display). *Fourth row:* ground truth deformation fields. *Fifth row:* reconstructed deformation fields. *Sixth row:* difference between the ground truth and the reconstructed deformation fields (scaled to $0 \ldots 1$ for display).

errors. For the "Dave's Head" data, the proposed method produced a 68.71% lower mean spatial error than that of Cootes *et al.* [67], and 43.81% smaller intensity error. For the "Chequerboard" data these improvements are 37.53% and 48.14% respectively.

This can be attributed to two factors. First is a better model of deformations in the proposed method. Inspecting closely the difference images for the recovered deformation fields, one can notice the triangular-shaped error patterns

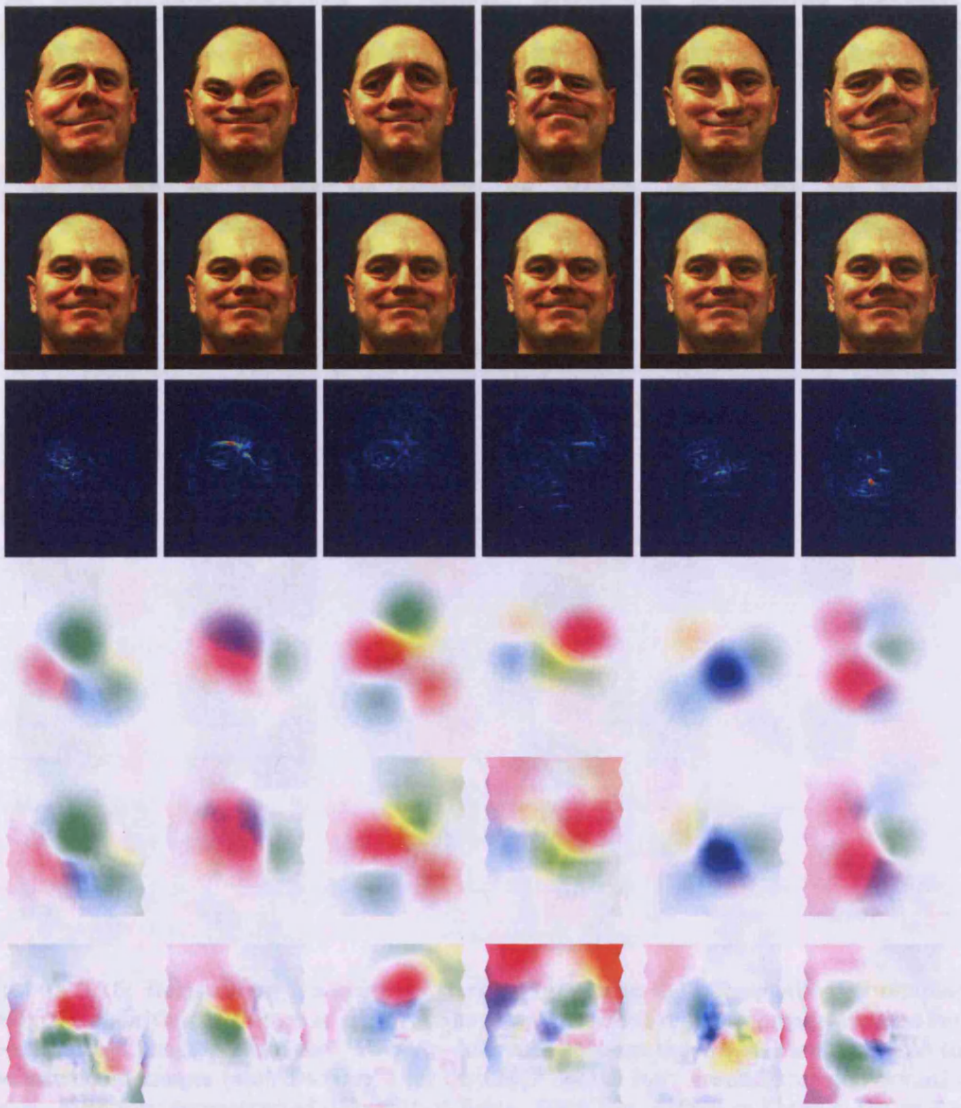


FIGURE 3.14: Registration of a synthetic data set "Dave's Head" using the algorithm of Cootes *et al.* [67]. *First row:* deformed template image. *Second row:* reconstructed images. *Third row:* absolute difference between the original template and the reconstructed images (scaled to 0...1 for display). *Fourth row:* ground truth deformation fields. *Fifth row:* reconstructed deformation fields. *Sixth row:* difference between the ground truth and the reconstructed deformation fields (scaled to 0...1 for display).

in the case of Cootes *et al.* [67] method: their deformation model is limited to one piece-wise affine field and cannot explain smooth deformations as well as the proposed method does. Additionally, the optimiser of Cootes *et al.* [67] seems to be worse. Inspecting the difference images reveals occasional crude

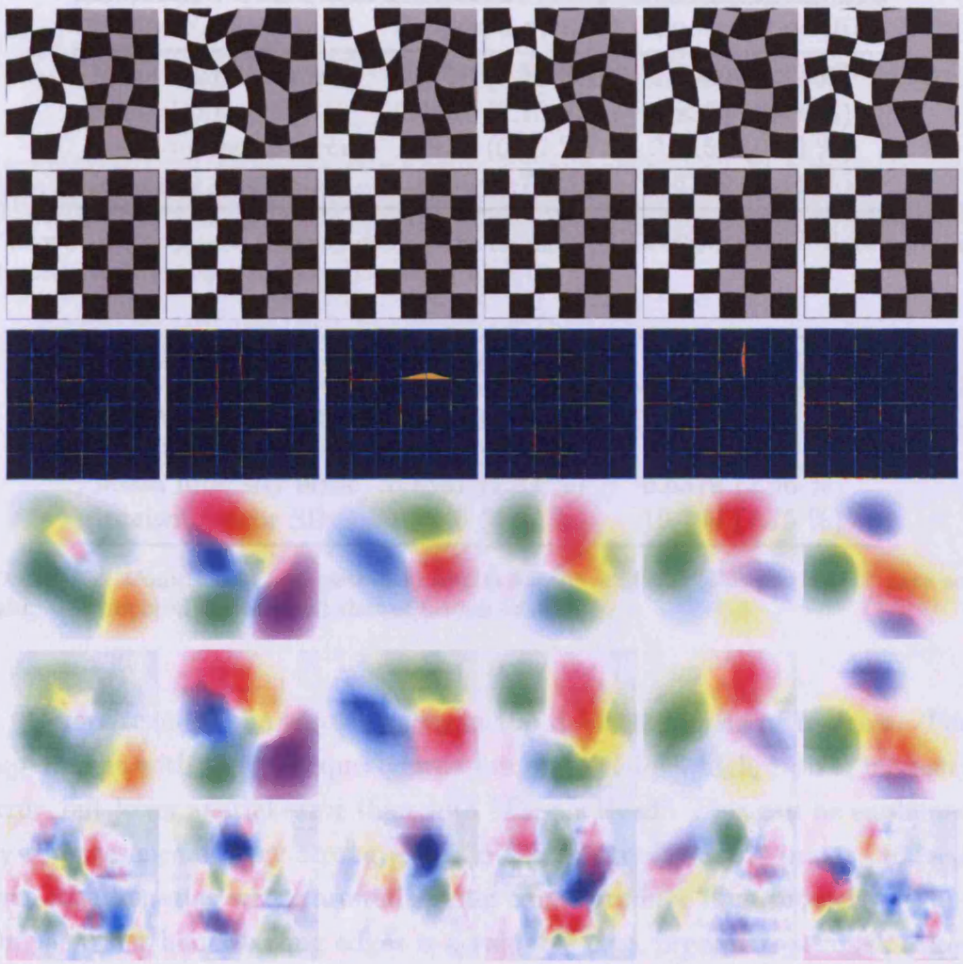


FIGURE 3.15: Registration of a synthetic data set built from the "Chequerboard" template using the algorithm of Cootes *et al.* [67]. *First row:* deformed template image. *Second row:* reconstructed images. *Third row:* absolute difference between the original template and the reconstructed images (scaled to 0...1 for display). *Fourth row:* ground truth deformation fields. *Fifth row:* reconstructed deformation fields. *Sixth row:* difference between the ground truth and the reconstructed deformation fields (scaled to 0...1 for display).

errors in alignment which their optimiser failed to deal with (*e.g.* right eye in the second column in Fig. 3.14, or third and fifth columns in Fig. 3.15. Further, in areas where there are no texture clues (*e.g.* monotone background) the algorithm of Cootes *et al.* [67] produces systematic spatial errors (*e.g.* columns four and six in Fig. 3.14), whereas the proposed method does not.

[6]Reported in pixels and, in brackets, as a percentage of the image height.

[7]Reported in units of intensity, with the full range being 0...255, and, in brackets, as a percentage of the intensity range.

Measure	Proposed method	Cootes *et al.* [67]
Mean spatial error[6]	0.4675 (0.12 %)	1.4939 (0.39 %)
Spatial error SD	0.4857 (0.13 %)	1.8364 (0.48 %)
Mean intensity error[7]	1.4136 (0.55 %)	2.5157 (0.99 %)
Intensity error SD	1.8669 (0.73 %)	3.6799 (1.44 %)

TABLE 3.1: Comparison of registration results for the artificial "Dave's Head" data set using the proposed method and that of Cootes *et al.* [67].

Measure	Proposed method	Cootes *et al.* [67]
Mean spatial error	0.6073 (0.24 %)	0.9722 (0.38 %)
Spatial error SD	0.5668 (0.22 %)	1.1119 (0.43 %)
Mean intensity error	3.3881 (1.33 %)	6.5379 (2.56 %)
Intensity error SD	10.246 (4.02 %)	19.769 (7.75 %)

TABLE 3.2: Comparison of registration results for the artificial "Chequerboard" data set using the proposed method and that of Cootes *et al.* [67].

Comparison of the results between the datasets is also of interest. The experiment with the "Chequerboard" template yielded higher pixel intensity error, but lower spatial error than with "Dave's Head". This can be explained by the presence of very strong edges in the "Chequerboard" image. Indeed, even small spatial errors around strong edges produce high intensity errors. On the other hand, strong edges is a valuable clue, present to a much lesser extent in "Dave's Head", that helps the algorithm to better find the correct alignment, hence the lower spatial error with the "Chequerboard" image.

Overall, the reconstruction errors with the proposed approach are sufficiently small to call the results a success.

3.6.2 Comparison With Manual Annotation

To evaluate the quality of the results produced by the proposed registration algorithm, comparison with the manual annotation has been performed. The FGNET talking head dataset comes with publicly available annotation, with 68 control points placed in each image at corresponding locations across the ensemble. The control points in the FGNET annotation cover the facial region of the talking head. The IMM dataset is also annotated, with 58 control points per image, also covering the facial area.

METHOD The following metrics are computed and compared: MAD between the reference and the images, mean NMI between reference and each image as well as mean, pairwise MAD and pairwise NMI between every pair of shape normalised images, and pixel stack entropies of the aligned ensemble. The procedure to compute the above metrics for the alignment defined by sparse set of control points is explained below, followed by the procedure to compute these metrics given the deformation model of Section 3.5.2. The results are compared in the end.

Since the control points in the manual annotation cover only the facial region of the images, a fair comparison requires the results to be evaluated only within the region covered by the control points (convex hull of the cloud of control points).

First, all sets of control points (from each image) are aligned using Generalised Procrustes Analysis (GPA) and the reference configuration of points is found by averaging the aligned point sets. To obtain shape-free patches, the reference configuration of points is first triangulated and each image $\mathcal{I}_i$ is warped to the reference configuration, yielding $\mathcal{I}'_i = W_i(\mathcal{I}_i)$, using the piecewise affine deformation model (see Eq. (3.7)) defined by the above triangulation. This procedure (shape normalisation) follows Cootes and Taylor [59] exactly. The average image $\mathcal{R}_{\text{CP}}$ is computed by averaging the shape-free patches: $R_{\text{CP}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{I}'_i$.

Let $\text{MAD}_\Omega(\mathcal{A}, \mathcal{B})$ be the MAD between images $\mathcal{A}$ and $\mathcal{B}$ over the domain Ω:

$$\text{MAD}_\Omega(\mathcal{A}, \mathcal{B}) = \frac{1}{|\Omega|}\sum_{\forall x,y\in\Omega}\left(\frac{1}{3}\sum_{c=1}^{3}|\mathcal{A}(x,y,c) - \mathcal{B}(x,y,c)|\right). \tag{3.61}$$

The mean absolute difference in the reference frame (MAD_{ref}) is then computed by warping each image in turn to the reference frame, computing the absolute difference with the reference, and averaging the result (dividing the sum by the number of images, pixels, and channels):

$$\text{MAD}_{\text{ref}} = \frac{1}{N}\sum_{i=1}^{N}\text{MAD}_{\Omega_{\text{ref}}}(\mathcal{R}_{\text{CP}}, W_i(\mathcal{I}_i)), \tag{3.62}$$

where N is the number of images, Ω_{ref} is the domain of the reference image (pixels covered by the convex hull of the control points).

Similarly, the mean absolute difference in the original frame is computed by warping the reference back, to align with each image in turn, computing the difference between them, and similarly averaging the result:

$$\mathrm{MAD}_{\mathrm{orig}} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{MAD}_{\Omega_i}(W_i^{-1}(\mathcal{R}_{\mathrm{CP}}), \mathcal{I}_i), \tag{3.63}$$

where, again, N is the number of images, Ω_i is the domain of the i-th original image and $W_i^{-1}(\mathcal{R}_{\mathrm{CP}})$ denotes the reference warped back to conform with the i-th original image $\mathcal{I}_i$.

By analogy with Eq. (3.62) and Eq. (3.63), mean NMI (see Eq. (3.39)) is computed in the reference and the original frame:

$$\mathrm{NMI}_{\mathrm{ref}} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{NMI}_{\Omega_{\mathrm{ref}}}(\mathcal{R}_{\mathrm{CP}}, W_i(\mathcal{I})_i), \tag{3.64}$$

$$\mathrm{NMI}_{\mathrm{orig}} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{NMI}_{\Omega_i}(W_i^{-1}(\mathcal{R}_{\mathrm{CP}}), \mathcal{I}_i). \tag{3.65}$$

Also computed is the the average pixel stack entropy (see Eq. (3.34)):

$$\mathrm{PSE}_{\mathrm{ref}} = \frac{1}{|\Omega_{\mathrm{ref}}|}\sum_{x,y\in\Omega_{\mathrm{ref}}} H(\mathbf{s}(W_i(\mathrm{I}_i), x, y)), \tag{3.66}$$

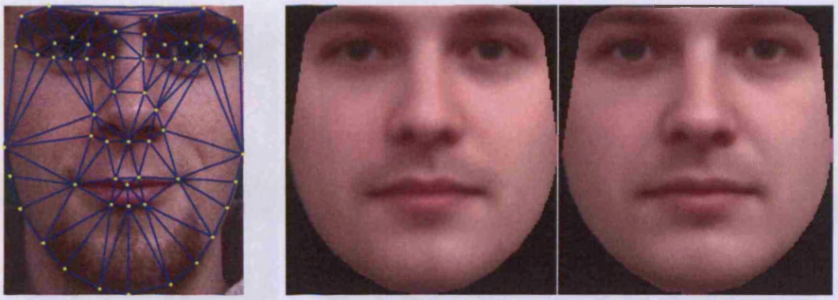
where I_i are images $\mathcal{I}_i$ converted to gray scale, $\mathbf{s}(\mathrm{I}, x, y)$ is a pixel stack obtained by sampling all images I_i at location (x, y), and $H(\cdot)$ is the entropy.

Finally, the MAD and NMI are computed between every pair of shape-normalised images and the result averaged:

$$\mathrm{MAD}_{\mathrm{ref}}^{\mathrm{p.w.}} = \frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{i-1}\left(\mathrm{MAD}_{\Omega_{\mathrm{ref}}}\left(W_i(\mathcal{I}_i), W_j(\mathcal{I}_j)\right)\right), \tag{3.67}$$

$$\mathrm{NMI}_{\mathrm{ref}}^{\mathrm{p.w.}} = \frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{i-1}\left(\mathrm{NMI}_{\Omega_{\mathrm{ref}}}\left(W_i(\mathcal{I}_i), W_j(\mathcal{I}_j)\right)\right). \tag{3.68}$$

Given the deformation fields $\mathcal{D}_i$ and the affine transforms A_i, computed by the proposed algorithm, that bring the images $\mathcal{I}_i$ into correspondence, the $\mathrm{MAD}_{\mathrm{orig}}$ metric from Eq. (3.63) can be computed trivially, by using the warp $\mathfrak{W}^{-1}(\mathrm{A}_i, \mathcal{D}_i, \cdot)$ in place of $W_i^{-1}(\cdot)$ to warp the reference image to each of the original images. This is possible because the domains Ω_i are already known (from the control points defined on the original images).



(a) Example image with annotation. (b) Mean using manual annotation. (c) Mean using the proposed algorithm.

FIGURE 3.16: Comparison of the registration with the proposed algorithm vs. manual annotation (IMM data set).

Computing the metrics in the reference frame, given $\mathcal{D}_i$ and A_i, is slightly more involved, because for a fair comparison the domain in the reference frame, Ω'_{ref}, needs to be estimated first. It should be noted that, in general, $\Omega'_{\text{ref}} \neq \Omega_{\text{ref}}$. In other words, the location of features in the average of the shape-normalised images (reference) depends, to an extent, on the registration algorithm being used. So, to estimate Ω'_{ref} the following simple procedure was employed. The control points $\mathbf{c}_{i,j}$ in each image were first transformed to the reference frame using the computed deformation fields $\mathcal{D}_i$ and the affine transforms A_i: $\mathbf{c}'_{i,j} = \mathfrak{W}^{-1}(\mathrm{A}_i, \mathcal{D}_i, \mathbf{c}_{i,j})$, $\forall i, j$. The transformed sets of control points were averaged to establish the reference configuration and so Ω'_{ref}.

The above metrics that are computed in the reference frame can now be computed analogously, by substituting $\mathfrak{W}(\mathrm{A}_i, \mathcal{D}_i, \cdot)$ for $W_i(\cdot)$, and Ω'_{ref} for Ω_{ref} in the above equations.

COMPARISON Figure 3.16, Fig. 3.17, and Fig. 3.18 show the layout of the control points in the manually annotated images, and the two averages of the shape-normalised ensembles: the one obtained with manual annotation and the one obtained with the proposed algorithm.

Table 3.3 summarises the comparison metrics for the FGNET dataset (sample of 128 images). Table 3.4 summarises the same results for the IMM dataset (all frontal colour images, 37 in total). In all cases the proposed method noticeably outperforms manual annotation. The quantitative improvement over manual annotation in the case of the `xm2vts` data set is more modest (Table 3.5), because of much greater variation in pixel intensities that cannon

(a) Example image with annotation. (b) Mean using manual annotation. (c) Mean using the proposed algorithm.

FIGURE 3.17: Comparison of the registration with the proposed algorithm vs. manual annotation (FGNET "Talking Head" data set).

(a) Example image with annotation. (b) Mean using manual annotation. (c) Mean using the proposed algorithm.

FIGURE 3.18: Comparison of the registration with the proposed algorithm vs. manual annotation (`xm2vts` Session 1 data set).

TABLE 3.3: Comparison of registration results: proposed algorithm vs. manual annotation. FGNET "Talking Head" data set (sample of 128 images).

Metric	Manual annotation	Proposed algorithm	Improvement
MAD_{ref}	6.7197	5.8299	13.24 %
MAD_{orig}	7.2824	6.1324	15.79 %
$MAD_{ref}^{p.w.}$	9.1288	7.9491	12.92 %
NMI_{ref}	1.2304	1.2528	1.82 %
NMI_{orig}	1.2233	1.2440	1.69 %
PSE_{ref}	4.1768	4.0620	2.75 %
$NMI_{ref}^{p.w.}$	1.2069	1.2213	1.19 %

TABLE 3.4: Comparison of registration results: proposed algorithm vs. manual annotation. IMM data set (all colour frontal images, 37 in total).

Metric	Manual annotation	Proposed algorithm	Improvement
MAD_{ref}	15.7575	13.8267	12.25 %
MAD_{orig}	16.4288	14.7785	10.04 %
$MAD_{ref}^{p.w.}$	22.6327	19.8833	12.15 %
NMI_{ref}	1.1910	1.2175	2.22 %
NMI_{orig}	1.1823	1.2005	1.54 %
PSE_{ref}	4.7194	4.6705	1.04 %
$NMI_{ref}^{p.w.}$	1.1606	1.1812	1.77 %

TABLE 3.5: Comparison of registration results: proposed algorithm vs. manual annotation. `xm2vts` Session 1, 295 images.

Metric	Manual annotation	Proposed algorithm	Improvement
MAD_{ref}	19.3037	18.3413	4.99 %
MAD_{orig}	20.1811	18.8650	6.52 %
$MAD_{ref}^{p.w.}$	27.7852	26.4328	4.87 %
NMI_{ref}	1.0905	1.0996	0.84 %
NMI_{orig}	1.0888	1.0958	0.64 %
PSE_{ref}	6.1772	6.1261	0.83 %
$NMI_{ref}^{p.w.}$	1.0664	1.0695	0.29 %

be explained by deformation alone, due to the nature of imagery and greater length of the data set. However, the qualitative improvement in this case is noticeable: observe the shady lines around the eyes in Fig. 3.18c. These are the rims of the glasses present in some images of the data set. The proposed algorithm, unlike the manual annotation, managed to correctly align the rims of the glasses which, being in alignment, manifest themselves in the average image. Note also the visually noticeable better alignment of the deep wrinkles which run from the nose to the corners of the mouth.

3.6.3 Registration of Various Data Sets

3.6.3.1 FGNET "Talking Head" (Within-subject Registration)

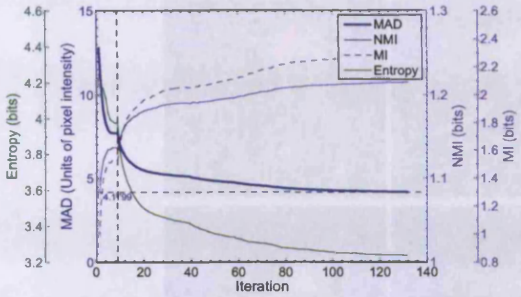
In this and the following experiments, in order to visually inspect the progress of registration, at each iteration the current estimate of the reference image was saved. Initially, averaging unregistered images leads to a blurry reference,

FIGURE 3.19: Example images from the FGNET "Talking Head" data set.

Initial 8 (affine) 11 16 32 131 (final)

FIGURE 3.20: Evolution of the texture model for the FGNET "Talking Head" data set.

as the averaging operates on stacks of pixels that do not yet correspond do each other. As the algorithm progresses and incrementally establishes the correct correspondences, the shape-normalised average of the images converges to a true, crisp picture of the underlying structure. A sharp final reference image means that all images have been well aligned by the groupwise registration. This is a useful technique to visualise the progress. Figure 3.20 shows the evolution of the reference after k iterations (the iteration number is shown below each image). The leftmost image shows the average of the unregistered images, the next image shows the reference after the completion of the affine stage, and the remaining images show the evolution during the non-rigid stage. Note that iteration numbers are given *inclusive* of the affine stage, *e.g.* the third image in Fig. 3.20 shows the reference after iteration 3 of the non-rigid stage (11-th overall), as the affine stage required 8 iterations. This convention is followed throughout the thesis in other analogous figures. (Note that the texture model obtained by shape normalisation and averaging of the *original* images is shown, even though the registration relies on the model built from the preprocessed images.) To analyse the registration progress more quantitatively, at every iteration the following measures were computed and recorded: the value of the cost function (average MAD between each shape-normalised image and the reference), similarly the average MI and NMI between each image and the reference, and finally the mean pixel stack entropy of the entire shape-normalised ensemble. It should be noted that while pixel



Measure	Initial	Final	Impr.
MAD	12.839	4.1799	67.45 %
MI	0.9024	2.2778	152.40 %
NMI	1.0758	1.2148	12.92 %
Entropy	4.2492	3.2340	23.89 %

FIGURE 3.21: Registration quality measures (FGNET "Talking Head").

stack entropy and MAD are measures of *dissimilarity*, and so expected to decrease the algorithm progresses, the MI and NMI are measures of *similarity*, and so expected to increase. The plot in Fig. 3.21 illustrates the evolution of these measures. The dotted vertical line demarcates the affine and the non-rigid stages. The dotted horizontal line show the final value of the cost function. This convention is followed throughout in all similar plots.

The entire FGNET "Talking Head" video comprises 5000 frames. For this experiment, a subset of 256 images was sampled from the full set using the procedure described in Section 3.6.5.

Having registered the images, it is possible to construct statistical appearance models, using deformation maps directly to build high resolution shape model. If memory is a concern, however, one might obtain traditional control point-based representation of shape in the end by sampling the deformation maps. The first two modes of variation of combined model of the FGNET data set are shown in Fig. 3.22.

To show that the proposed algorithm can be applied not only to facial imagery, the experiment on registration of handwritten digits from Miller *et al.* [182], Learned-Miller [156] was replicated. The example images from these data set are shown in Fig. 3.23 and Fig. 3.25. The evolution of the texture model as the registration progresses are shown in Fig. 3.24 and Fig. 3.26. The progress plots are found in Fig. 3.27 and Fig. 3.28. It is evident that the proposed algorithm admirably copes with this task.

FIGURE 3.22: An Active Appearance Model of a talking head obtained by registering the images using the proposed method.

FIGURE 3.23: Example images from the handwritten zeros data set.

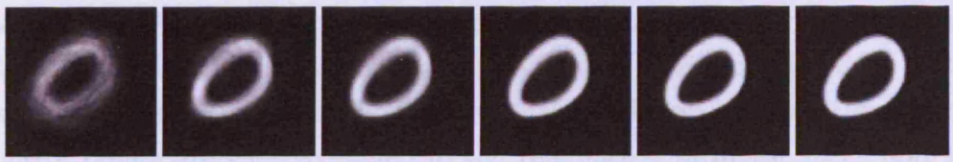
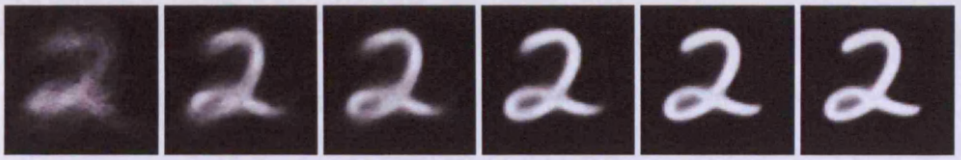


FIGURE 3.24: Evolution of the reference image for the handwritten zeros data set.

3.6.3.2 Inter-subject Registration

For this experiment, frontal images from Session 1 of the `xm2vts` data set were used. The total number of images is 295. Some of the example images from this data set are shown in Fig. 3.29. Note the presence male and female faces, glasses in some images as well as significant variation in facial hair. Images were cropped to 342×366 pixels to contain the facial region. The images were then registered. The evolution of the texture model is shown in Fig. 3.30 and the progress plots in Fig. 3.32. To illustrate the usefulness of the proposed

FIGURE 3.25: Example images from the handwritten twos data set.



Initial 16 (affine) 19 31 50 67 (final)

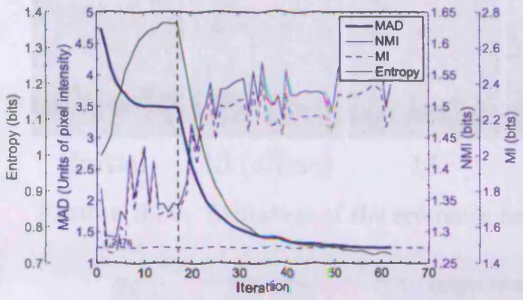
FIGURE 3.26: Evolution of the reference image for the handwritten twos data set.

registration framework for appearance model building, an AAM was built from registered images, as in the previous experiment. The first three modes of variation are shown in Fig. 3.31.

COMPARISON WITH STATE OF THE ART. Cootes *et al.* [67] also experiment with the `xm2vts` data set and provide quantitative results. It seems interesting, therefore, to compare the output of their method with the algorithm proposed in this chapter. After registering the images, Cootes *et al.* [67] warp the manually placed control points (more precisely, their subset of 20 points defined in the paper) using the computed deformation fields. The warped positions of the control points are then averaged in the reference coordinate space and the average is warped back to the space of the original images. These are compared to the manually placed control points: Euclidean distance is measured and normalised by the interocular distance. To illustrate, for a single image the error measure of Cootes *et al.* [67] is

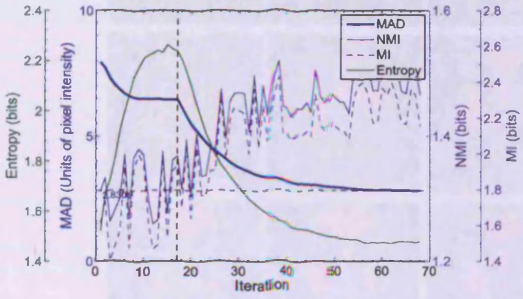
$$E_i = \frac{1}{N\gamma}\sum_{j=1}^{N}|\mathbf{p}_{i,j} - W_i(\frac{1}{N}\sum_{i=1}^{N}W_i^{-1}(\mathbf{p}_{i,j}))|, \tag{3.69}$$

where N is the number of control points, γ is the interocular distance, $\mathbf{p}_{i,j}$ is the position of the j-th manually placed control point in the i-th image, W_i and W_i^{-1} are the computed deformation field and its inverse for the i-th image. Cootes *et al.* [67] report the values of this spatial error measure for various variants of their algorithm, and their best median value is 3.5 pixels.



Measure	Initial	Final	Impr.
MAD	4.7422	1.2476	73.69 %
MI	1.8156	2.2468	23.75 %
NMI	1.3670	1.5349	12.28 %
Entropy	1.0006	0.7242	27.62 %

FIGURE 3.27: Registration quality measures (handwritten zeros).



Measure	Initial	Final	Impr.
MAD	7.9473	2.8301	64.39 %
MI	1.6118	2.1553	33.72 %
NMI	1.3123	1.4665	11.75 %
Entropy	1.5212	1.4786	2.80 %

FIGURE 3.28: Registration quality measures (handwritten twos).

FIGURE 3.29: Example images from the `xm2vts` (Session 1) data set.

The algorithm proposed in this chapter yields the median value of 3.56 pixels, which compares favourably (subject to minor differences in the experimental setup). Importantly, the best result (3.5 pixels) in Cootes *et al.* [67] is obtained using an ad hoc *shape term* in the cost function, while the algorithm in this chapter does *not* use one and still shows the same performance. Without the shape term, the method of Cootes *et al.* [67] produces a much worse value of 4.3 pixels, a 18% difference.

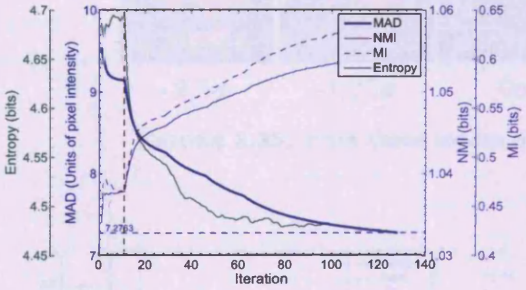
To ensure that the algorithm can perform inter-subject registration reliably, the experiment was repeated using IMM data set. The example images from this ensemble are shown in Fig. 3.33, the evolution of the texture model

Initial 10 (affine) 13 43 70 127 (final)

FIGURE 3.30: Evolution of the reference image for the `xm2vts` (Session 1) data set.

-2.5σ -1.25σ 0σ $+1.25\sigma$ $+2.5\sigma$

FIGURE 3.31: First three modes of variation for the `xm2vts` (Session 1) data set.



Measure	Initial	Final	Impr.
MAD	9.5460	7.2760	23.78 %
MI	0.4359	0.6365	46.00 %
NMI	1.0341	1.0544	1.97 %
Entropy	4.6869	4.4733	4.56 %

FIGURE 3.32: Registration quality measures (`xm2vts` Session 1).

in Fig. 3.34, the progress plots in Fig. 3.36, and the first three modes of variation of the resulting AAM in Fig. 3.35.

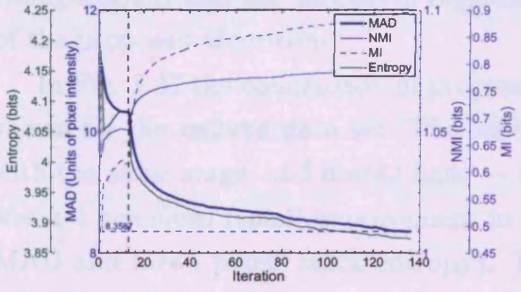
FIGURE 3.33: Example images from the IMM data set.

Initial 12 (affine) 15 32 70 111 (final)

FIGURE 3.34: Evolution of the reference image for the IMM data set.

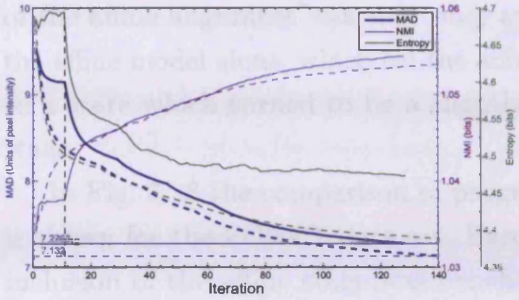
-2.5σ -1.25σ 0σ $+1.25\sigma$ $+2.5\sigma$

FIGURE 3.35: First three modes of variation for the IMM data set.



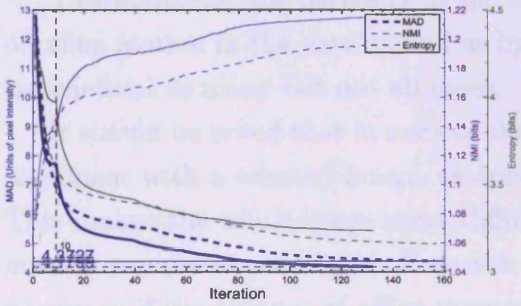
Measure	Initial	Final	Impr.
MAD	11.1349	8.3587	24.93 %
MI	0.4911	0.8949	82.23 %
NMI	1.0394	1.0761	3.53 %
Entropy	4.1843	3.8737	7.42 %

FIGURE 3.36: Registration quality measures (IMM).



Measure	Without	With	Impr.
MAD	7.1320	7.2763	-2.02 %
MI	0.6041	0.6365	5.35 %
NMI	1.0535	1.0544	0.09 %
Entropy	4.3883	4.4734	-1.94 %

FIGURE 3.37: Effect of the affine stage (`xm2vts` Session 1).



Measure	Without	With	Impr.
MAD	4.3727	4.1799	4.41 %
MI	2.0968	2.2778	8.63 %
NMI	1.1970	1.2148	1.48 %
Entropy	3.1605	3.2340	-2.32 %

FIGURE 3.38: Effect of the affine stage (FGNET "Talking Head" data set).

3.6.4 The Effect of The Affine Stage

This experiment aims to determine the contribution of the affine stage of registration to the overall progress. To determine this, several data sets were registered with and without the affine stage of the algorithm, and the results compared. Additionally, the exclusion of the affine stage stresses the algorithm more (as the more difficult non-rigid stage begins with a less favourable initial configuration) and the successful registration further confirms the robustness of the proposed algorithm.

In Fig. 3.37 the comparison of progress with and without the affine stage is shown for the `xm2vts` data set. The solid lines correspond to the experiment with the affine stage, and dotted lines — without. In this case, the affine stage was not beneficial (small improvement in MI and NMI and slightly worse final MAD and mean pixels stack entropy). This is because the amount of affine movement in the data set was not significant for the affine stage to have a significant positive effect; on the other hand, the initial model in the beginning

of the affine alignment was very poor and could not be improved much with the affine model alone, which led the affine stage to greedily drive the situation to a state which turned to be a slightly worse initial state for the non-rigid stage.

In Fig. 3.38 the comparison of progress with and without the affine stage is shown for the FGNET data set. Here, unlike in the previous example, the inclusion of the affine stage was beneficial: the original data contains a fair amount of affine motion but significantly less texture variation than the `xm2vts` data set.

To summarise, the necessity of the affine stage is governed by the amount of affine motion in the data as well as by the amount of texture variation, and is beneficial in many but not all cases.

It should be noted that in none of the experiments the crude pairwise affine alignment with a selected image, as done in Cootes *et al.* [67], was performed. This makes the affine stage more difficult, but is more congruent with the overall groupwise paradigm. However, such an initialisation might still be necessary if the amount of affine movement in the ensemble is very high.

3.6.5 Scalability

To investigate how the performance of the algorithm changes as a function of the number of images in the ensemble, the algorithm was run several times on ensembles of varying lengths and the results compared. The experiment was performed with two data sets: FGNET "Talking Head" and DAVE, as these are sufficiently long.

To draw N samples from a data set, the following procedure was used. First, a matrix of pairwise differences, D, between each possible pair of the unregistered images was computed and saved: $\mathrm{D}(i,j) = |\mathcal{I}_i - \mathcal{I}_j|$. Then, to draw N samples, the FPS strategy was employed to select N *most dissimilar* images from the dataset, using D to determine the relative "distances" between images. This way of sampling produces subsets which are more characteristic of the entire set, than, say, taking the first N, or every k-th frame. This also makes the problem *harder*, because the images are selected that are most dissimilar to each other.

The registration was done on the subsets of lengths from 16 to 256 in increments of 16, totalling 16 experiments with each dataset.

Since increasing the number of images in the ensemble typically leads to more variance in pixel colours, even in perfectly registered images, the measures of the final alignment quality are expected to get worse as the length of the ensemble increases, *regardless* of the performance of the algorithm. And so, for a fair evaluation, this effect needs to be accounted for. To do so, in this experiment the final quality measures for each registration run were divided by those of the corresponding unregistered ensemble. In other words, the *relative* change was examined.

It should be noted, parenthetically, that while larger data sets take more time to register, their registration is not necessarily harder. Indeed, the more "intermediate stages" are present between the images in the ensemble, the easier it is for the algorithm to construct an accurate model (as more useful information is present) and so is easier to register the images.

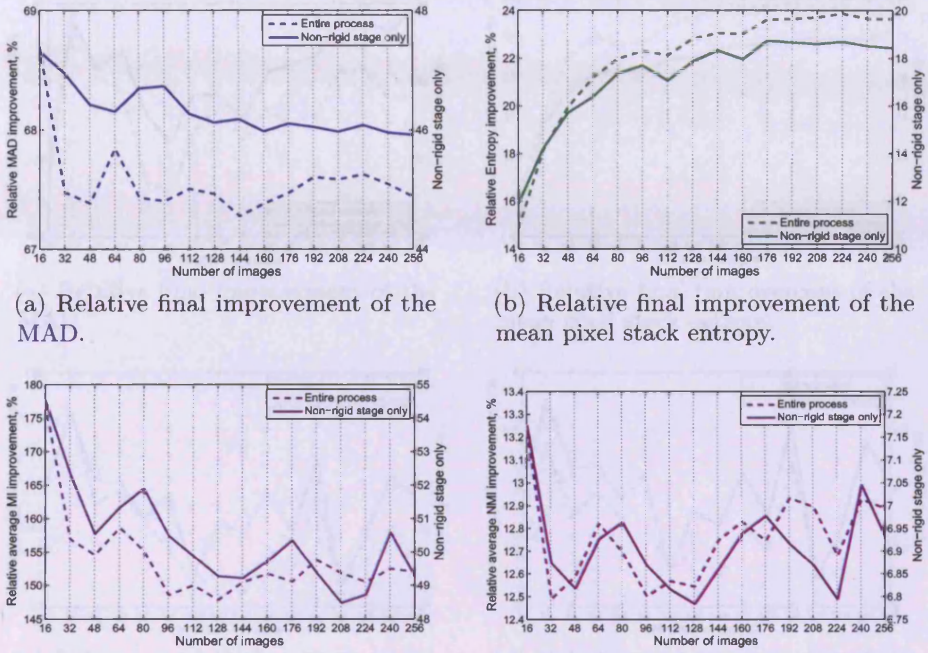
The performance measures for the above experiments are given in Fig. 3.39 and Fig. 3.40. The solid lines in the plots indicate the results of the non-rigid stage only, and the dotted lines —the results inclusive of the affine stage.

In both cases, there is no evidence of the performance degrading significantly as the number of images grows, despite the selection of samples that maximised inherent texture variation in the data sets.

In the case of the FGNET data (Fig. 3.39) the trend is better discernible: as the number of images increases, so does the amount of the inherent texture variation (indicated by the growing mean pixel stack entropy), which cannot be explained, even in principle, by spatial deformations alone. This leads to the decreasing relative improvement, but only to some extent. Above a certain number of images (about 160), the trend stabilises and no further degradation of performance is noticeable. This trend is less discernible in the case of "Dave's Head" data (Fig. 3.40).

The experiment was repeated with the same data sets, but this time the samples were drawn in sequential order. The corresponding results are shown in Fig. 3.41 and Fig. 3.42.

In this case, as the number of images increases, the relative improvement (compared to the unregistered data) also *increases*, but only to some extent. This is due to the fact that in in this experiment the samples were not drawn to maximise the inherent texture variation in the data, and the measured intensity errors are largely due to misalignment of images (which the algorithm



(a) Relative final improvement of the MAD.

(b) Relative final improvement of the mean pixel stack entropy.

(c) Relative final improvement of the average MI between each image and the reference.

(d) Relative final improvement of the average NMI between each image and the reference.

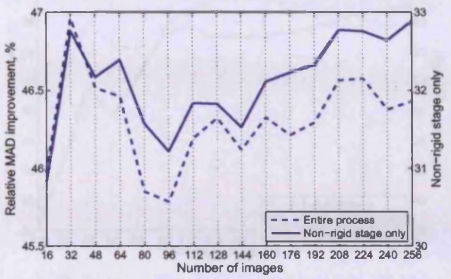
FIGURE 3.39: Results of the scalability experiment using the FGNET "Talking Head" data set. Dotted lines indicate the results of the entire registration process, solid lines are indicate the results of the non-rigid stage only.

later improves), not due to texture variation inherent in the data. As the size of the ensemble grows, assuming the algorithm neutralises most of the spatial alignment error, the remaining error due to inherent variation does not grow as fast as in the previous experiment. This leads to increasing relative improvement. This behaviour is the expected and intuitively pleasing result.

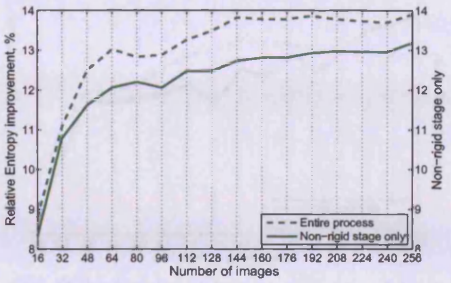
3.6.6 Comparison of The Optimisers

To justify the choice of SPSA as the optimiser, the algorithm was run on two image ensembles (FGNET and `xm2vts`) with the SPSA and then with the Nelder-Mead method as the optimisers, and the results compared. To make the comparison fair, both algorithms were allowed to use the same number of the objective function evaluation.

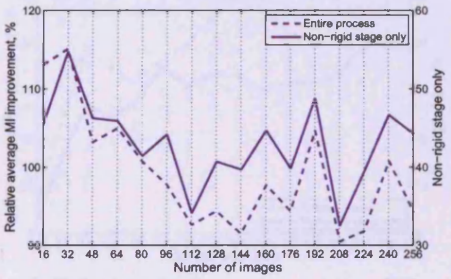
The comparative progress plots are found in Fig. 3.43 and Fig. 3.44. The solid lines correspond to SPSA and dotted lines — to Nelder-Mead. While



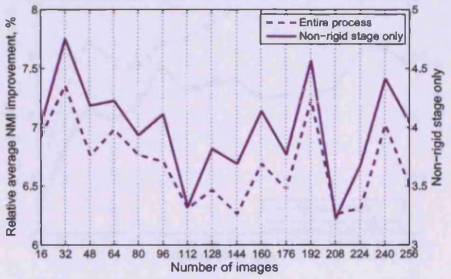
(a) Relative final improvement of the MAD.



(b) Relative final improvement of the mean pixel stack entropy.



(c) Relative final improvement of the average MI between each image and the reference.



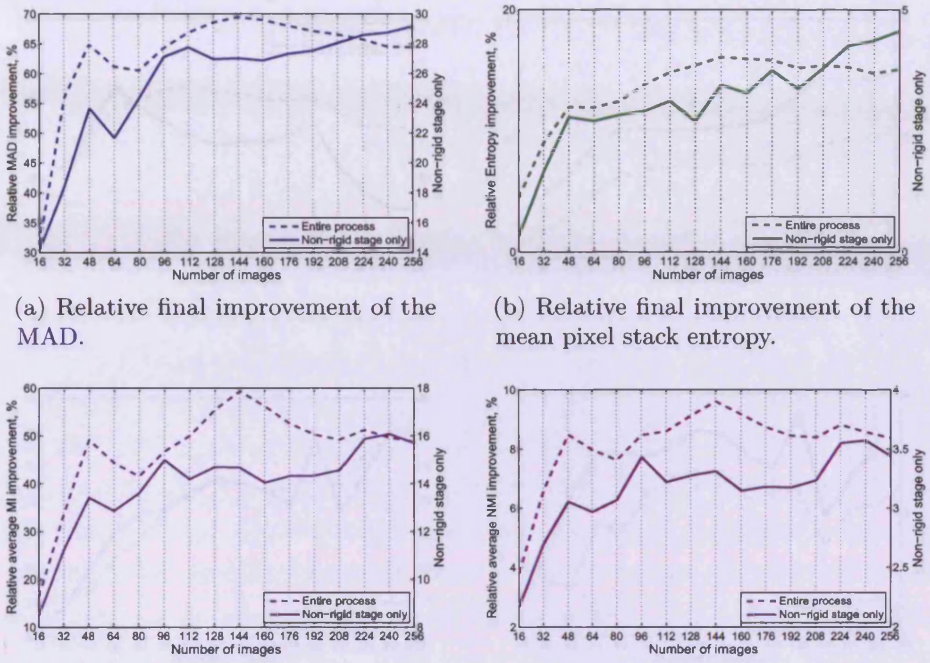
(d) Relative final improvement of the average NMI between each image and the reference.

FIGURE 3.40: Results of the scalability experiment using the "Dave's Head" data set. Dotted lines indicate the results of the entire registration process, solid lines are indicate the results of the non-rigid stage only.

the final quality measures are comparable (SPSA slightly outperforming), the important result here is the total number of iterations (running time) is *significantly* smaller (relative difference was 178.03% and 107.69% in the two experiments) when SPSA is employed. This confirms its advantages in the proposed registration framework.

3.7 Future Work

Apart from the incremental improvement of each part of the registration framework — deformation modelling, objective function, and optimisation — which is a natural continuation of this research, several important outstanding problems were identified, solutions to which, it can be speculated, would greatly advance the field of groupwise image registration and automatic appearance model building. They are outlined below.



(a) Relative final improvement of the MAD.

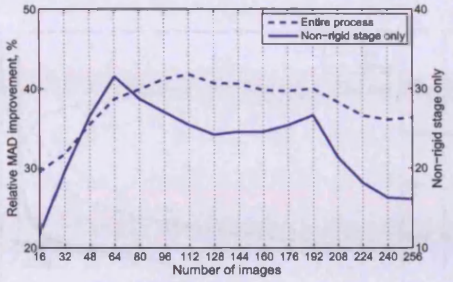
(b) Relative final improvement of the mean pixel stack entropy.

(c) Relative final improvement of the average MI between each image and the reference.

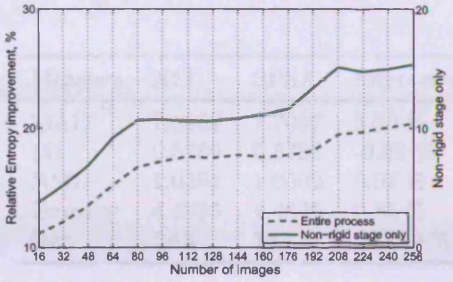
(d) Relative final improvement of the average NMI between each image and the reference.

FIGURE 3.41: Results of the scalability experiment using the FGNET "Talking Head" data set, using sequential samples. Dotted lines indicate the results of the entire registration process, solid lines are indicate the results of the non-rigid stage only.

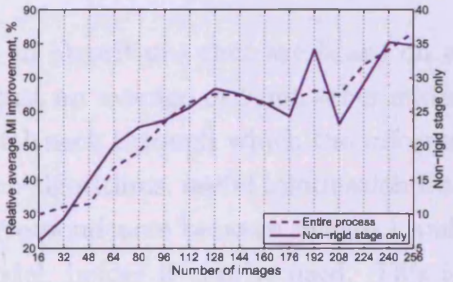
The first problem with current groupwise registration algorithms is that the objective function is based on some sort of per-pixel statistics. Be it an evolving estimate of a "reference" image, obtained by averaging the partially registered images (Sidorov *et al.* [243]), or the entropies of the pixel stacks (Miller *et al.* [182]), or some other kind of statistical method — the problem is that such per-pixel computations are not entirely appropriate before the images are fully registered and the correspondences between pixels are determined. This is a "chicken and egg" problem. In such algorithms, it is assumed that if the images are initially approximately aligned, then the use of such per-pixel statistics on "approximately corresponding" pixels is sufficient to slightly improve the knowledge of correspondence between pixels on the first iteration; then this improved correspondence can be used to do more accurate statistics on the second iteration and so on. This leads to the question: how exactly



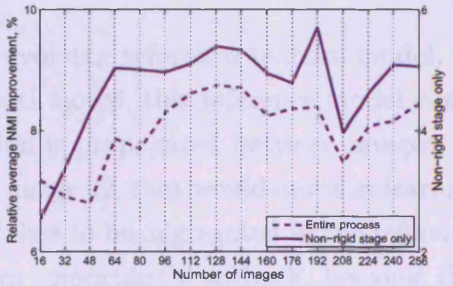
(a) Relative final improvement of the MAD.



(b) Relative final improvement of the mean pixel stack entropy.

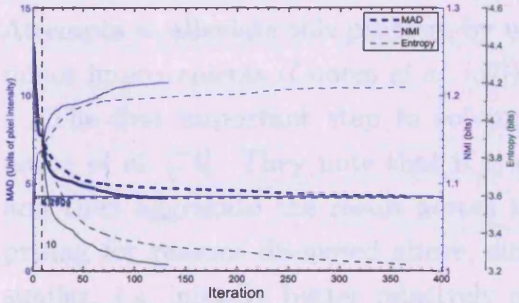


(c) Relative final improvement of the average MI between each image and the reference.



(d) Relative final improvement of the average NMI between each image and the reference.

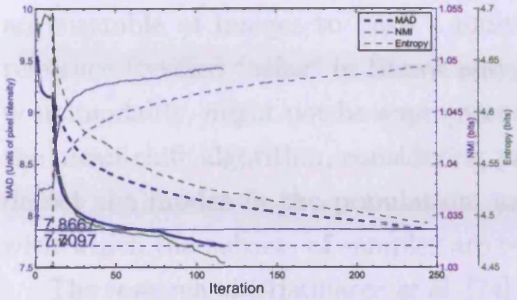
FIGURE 3.42: Results of the scalability experiment using the "Dave's Head" data set, using sequential samples. Dotted lines indicate the results of the entire registration process, solid lines are indicate the results of the non-rigid stage only.



Measure	NM	SPSA	Impr.
MAD	4.2601	4.1799	1.88 %
MI	2.2276	2.2778	2.25 %
NMI	1.2101	1.2148	0.39 %
Entropy	3.2334	3.2340	-0.02 %
Iter.	**367**	**132**	178.03 %

FIGURE 3.43: Comparison of the optimisers (FGNET "Talking Head" data set).

does this assumption influence the basin of convergence? Of interest would be a comparative quantitative study evaluating the basin of convergence for various choices of the objective function, deformation models and optimisation regimes in groupwise registration algorithms.



Measure	NM	SPSA	Improvement
MAD	7.8667	7.7097	2.00 %
MI	0.5789	0.5788	-0.02 %
NMI	1.0494	1.0502	0.07 %
Entropy	4.4925	4.4546	0.85 %
Iter.	**243**	**117**	107.69 %

FIGURE 3.44: Comparison of the optimisers (`xm2vts` data set).

In algorithms that are based on an evolving reference texture model, be it just an average or some other statistical model, this reference model is the bottleneck through which the information is propagated between images. In such algorithms, useful information from image 42, that would assist in learning correspondences between images 1 and 2, has to be aggregated in the reference model, before it can be used. This is an important drawback, because this approach does not scale up well. Since the exact correspondences between pixels are not known in advance, computing an aggregate reference model with per-pixel statistics has the effect of filtering out useful high-frequency information (*e.g.* average image is blurry). This undesirable effect increases with the number of images being registered. (Note that the other side of the medal is the desirable effect of filtering out noise and transient features). Attempts to alleviate this problem by using more robust statistics yield only minor improvements (Cootes *et al.* [67]).

The first important step in solving this problem was made by Cristinacce *et al.* [74]. They note that it is easier to register similar images first and then aggregate the result across the entire ensemble. (This is unsurprising for reasons discussed above, since when averaging small number of similar, *i.e.* initially better relatively aligned, images, the loss of valuable high-frequency information due to averaging is less significant). The approach of Cristinacce *et al.* [74] is to cluster the images and construct the shortest path tree over the clusters. The order in which the images are registered with an evolving estimate of the mean is dictated by this tree. Cristinacce *et al.* [74] show that a more accurate result can be achieved with this method than when all images are registered at once. This quantitatively confirms the argument outlined above.

Furthermore, Blezek and Miller [29] also point out that it is common for an ensemble of images to have a multi-modal distribution and so a single reference (termed "*atlas*" in Blezek and Miller [29]), essentially an assumption of unimodality, might not be appropriate. They address this problem by using the mean-shift algorithm, considering pairwise distances between samples, to detect the modes in the population, and determine the references (atlases) with which the subsets of samples are registered.

The research of Cristinacce *et al.* [74] and Blezek and Miller [29] are the first steps in the right direction and it would be interesting to further investigate the possible approaches to alleviate the above problem. Consider an extreme case with no loss of information: when no per-pixel statistics are ever computed, and the hypothetical objective function only ever compares the original images between themselves directly. The important question is whether such algorithm is possible. The search for such method is one of the topics of potential future research.

While so far primarily the linear appearance models and their automatic construction were discussed, recent developments in the field of *multilinear* modelling suggest that potential exists in the application of multilinear algebra methods to modelling of appearance. The key papers exploring such possibility include Vasilescu and Terzopoulos [282–284, 286] and especially Vasilescu and Terzopoulos [285]. Conventional linear modelling approaches, such as PCA, assume that the apparent variation in an ensemble of images is due to a single contributory factor. Natural images, however, result from the contribution of multiple factors, such as scene orientation, subject identity, deformation, illumination *etc.* Exploiting multilinear algebra, Vasilescu and Terzopoulos [285] elegantly and effectively deals with the multifactor variation in image ensembles, by introducing a powerful tensor-based modelling framework. In Vasilescu and Terzopoulos [285], the primary proposed applications are recognition and synthesis.

The multilinear modelling paradigm is very promising, but a number of outstanding question remains. The work of Vasilescu and Terzopoulos [282–286] concentrates on modelling the pixel intensities only, the spatial transformations (such as due to rotation of the subject) being modelled as contributing factors. The first question then is: is it possible to augment the multilinear models with the idea of explicitly modelling the spatial deformations, together with

the intensity variation, in a fashion similar to AAM? Groupwise registration of images can be used to create an explicit model of deformations, just like for AAM. The second question is then: can an algorithm be developed that would, given an assorted ensemble of images, automatically extract the contributing factors? (Presently, in multilinear methods, it is assumed that the contributing factors are known, and the training images can be arranged into an observation tensor, with dimensions corresponding to factors). Can this process be integrated with groupwise registration, perhaps to give "groupwise registration and factor decoupling"? Can such models be used for something more demanding that recognition and classification? These questions could be a basis for potential future research.

As groupwise image registration has been related to the problem of manifold learning in Section 3.1, it would be interesting to investigate exactly which ideas can be borrowed from one area and applied to the other. It can be speculated that merging ideas from both fields can be fertile.

Another potential direction of future research is modelling of deformations that are inherently non-diffeomorphic. This is of interest in medical imaging, as well as in craniofacial modelling. This problem is mentioned, but not addressed, by Cootes *et al.* [62]: *"In cases where structures appear or disappear between one image and the next, these should be explicitly modelled as creation or destruction processes."* Indeed, the appearance and growth of a tumour in CAT brain scans, or the appearance of teeth when the mouth is opened, cannot be adequately modelled with existing techniques — the non-diffeomorphic features are explained as simply intensity variation, not a spatial process. It can be speculated that such imagery may be modelled with composite, or layered, AAMs, a technique proposed by Jones and Soatto [137] or a variation thereof. However, an automatic construction of such layered models appears to be an extremely difficult task: on top of the existing challenges of the groupwise registration, the algorithm will have to automatically separate each of the images into layers, simultaneously align the contents of each layer. This is a "chicken and egg" problem: if the demarcation of the images into layers was known, the groupwise registration of each layer can be done relatively easily; if the images were registered, by measuring the degree of "non-diffeomorphicity" the layers could be detected. Perhaps an iterative solution that would perform

demarcation and registration simultaneously is possible. This question can also be a promising direction of potential future research.

3.8 Conclusion

A novel approach to groupwise non-rigid image registration which is fast, reliable and requires no manual initialisation was proposed. Methods that implicitly reduce the dimensionality of the search space by representing increasingly complex deformations as a superposition of simpler deformations were developed. Due to this formulation it was possible to take advantage of the simplicity and efficiency of piece-wise affine interpolation to represent deformations and overcome previous limitations of this model due to limited smoothness and flexibility. A novel efficient and reliable, fully unsupervised stochastic optimiser — an adaptation of SPSA — whose performance in terms of the number of function evaluations at each iteration is independent on the dimensionality of the space was intimately integrated into the groupwise registration framework and proved to be a very efficient solution.

In evaluation of the proposed method, high robustness and success rate were demonstrated, as well as fast convergence on various types of test data which shows considerable improvement in terms of accuracy of solution and speed compared to existing methods. Due to the robustness of the proposed approach, *inter-subject* registration is possible. At the time of publishing the CVPR '09 paper (Sidorov *et al.* [243]), this was the first time that the groupwise registration of data possessing such variety (faces of multiple people) had been reported.

Due to the efficient formulation of the proposed approach, it is easily amenable for GPU implementation — in the experiments, apart from the control logic, all steps were performed on a GPU.

CHAPTER

4 Registration of Textured Surfaces

I remember, once, Peter the Great had a problem like that...

Pavel Chekov (*Star Trek*)

REGISTRATION of textured[1] surfaces, such as those obtained with the modern commercially available photogrammetric surface scanners (*e.g.* [6]), is a problem which is closely related to the problem of 2D image registration, but presents a number of challenges that prevent it from being immediately amenable to a solution by existing 2D registration methods. In this chapter, a methodology to effectively reduce the problem of groupwise registration of textured surfaces to a problem similar to that of registering 2D images is proposed.

More specifically, the method proposed in this chapter, focuses on the registration of textured genus-0 disk-like orientable open surfaces (Massey [179]) represented by triangulated manifolds, or meshes, defined in Section 4.3.1, since this is the most common type of surface data in orthodontic practice as well as in the fields of computer graphics and vision. However, with some modifications, the proposed approach can be adapted to surfaces of other kinds (such as closed surfaces, as discussed in Section 4.7).

[1]Texture, in this chapter, refers to any scalar or vector field (for example, RGB colours) defined on a surface. See also the discussion on page 135.

In short, this chapter discusses a variation on the method described in Chapter 3 which is adapted to the registration of textured 3D surfaces, using primarily texture information, and incorporating additionally geometrical information which is not present in 2D images. Note that this problem is related to, but is in general different from the problem of surface registration based on purely geometric information.

4.1 Motivation

The solution to the above problem is important for craniofacial appearance modelling for several reasons. Readily available textured surface scans are becoming an important and highly practical noninvasive diagnostic tool in orthodontics (Kau *et al.* [140], Popat *et al.* [215], Popat and Richmond [216], Sidorov *et al.* [241]). Having an automatic pipeline from acquisition to registration and modelling of ensembles of textured 3D face scans would allow for such diagnostic procedure to be performed on a massive scale, very cheaply and rapidly, providing valuable information to the clinician within minutes after scanning the patients (currently, manual analysis takes many hours of clinicians' time).

In longitudinal studies, the ability to reliably establish correspondences between features in multiple face scans of a patient would provide a valuable insight into dynamics of craniofacial development, disease progress, or post-surgical recovery. In latitudinal studies, large databases of patient face scans, which are presently being actively created, could be subjected to registration and then to statistical analysis, in order to discover novel facts about the variability and properties of the human craniofacial complex.

In biometrics, identification methods based on facial dynamics (see *e.g.* Benedikt [20] and references therein) will also greatly benefit from automatic registration of textured meshes as this would allow for a quick and easy creation of subject databases.

In computer graphics, and especially in the game industry, registration of textured surfaces from photogrammetric scanners, which are already ubiquitous in the field, would enormously facilitate the otherwise laborious process of creating animated head and face models by artists. The entire procedure, from scanning an actor's face with a non-invasive surface scanner, without using the

traditional physical landmarks (such as special lipstick or reflective labels glued to the face) to establishing the correspondences between features in multiple frames, to the final polygonal model of the head, will be streamlined by the proposed method for automatic registration of textured surfaces.

Additionally, in Chapter 5 a novel application of statistical 3D appearance models is proposed: statistically constrained meshless mechanical simulation. This application also greatly benefits from an automated pipeline for registering textured surfaces.

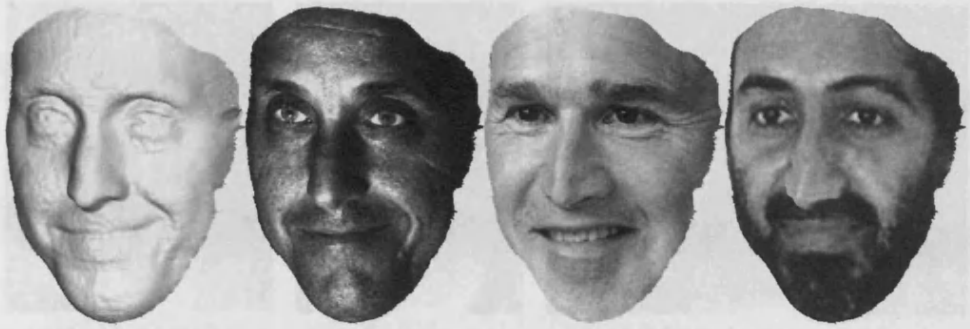
4.1.1 Information Content of Texture and Shape

Contemporary 3D surface scanners most frequently employ either passive (using only ordinary visible light cameras) or active (additionally using projected infrared patterns) photogrammetry to achieve 3D shape reconstruction [6], in addition to capturing the appearance (texture) of the object. These are the only types of scanners that can presently operate at video-rate. They typically sample 3D surfaces at a much lower resolution than that of the corresponding textures (many pixels of a texture per triangle of a mesh). This is unsurprising, not simply because high-resolution digital photography is ahead in the resolution race, but because such scanners compute the 3D shape of an object (typically by solving the simplified stereopsis problem) from images obtained with the same class of cameras as those that capture the texture.

Additionally, even in the state of the art commercial photogrammetric scanners, the errors in determining the 3D shape from images are still significant. This results in noisy surfaces and makes registration based on shape features (such as those discussed in Section 4.7) unreliable.

What is more important, in the case of craniofacial imagery, is that the 3D geometry (shape) of the head and face is relatively smooth, contains few details, and its information content is lower than that of the images (texture) of the head: the shape of the head is essentially a low-frequency signal. Therefore, since the texture data is readily available from photogrammetric scanners and usually has high information content (including important detail not found in shape data), it *must* be used to guide registration.

Incidentally, the higher information content of textures might be the reason why humans have evolved to be more sensitive to the texture of the face than to



(a) Original surface. (b) Original texture. (c) Same surface, texture of G. Bush. (d) Same surface, texture of Osama bin Laden.

FIGURE 4.1: Human visual system is more sensitive to texture than shape. Image from Bronstein *et al.* [46].

its shape when performing facial recognition tasks (Bronstein *et al.* [46]). This property of the human visual system is exemplified by Fig. 4.1 as well as by make-up artists in theatres who are known to radically change the appearance of actors by altering the facial texture with make-up, without altering the shape of the face (Bronstein *et al.* [46]).

For synthesis then, a good quality model of texture is more important than a good model of shape, so it seems important to drive the registration predominately by texture information.

4.2 Background

One approach to registration of textured meshes is to parameterise the meshes (see Section 4.2.3), thus mapping the corresponding textures to a plane, and then register the resulting flat textures as images, using, for example, the approach of Sidorov *et al.* [243] or Cootes *et al.* [66]. Using the computed correspondences between the flattened textures, the correspondences between points on the original meshes can be established.

This approach was used by the author in Sidorov *et al.* [241]. While it proved to work in many cases, the planar model of deformations in this approach has an important drawback: it is agnostic of the 3D geometry of the meshes. In other words, small displacements in the flattened textures, induced by the deformation model, do not necessarily correspond to small displacement

(a) Original surface (b) Surface projected onto a cylinder (c) Cylinder unwrapped

FIGURE 4.2: A naïve mapping of a surface onto a rectangle as in Sidorov *et al.* [243] or Blanz and Vetter [28].

on the original 3D surfaces and vice versa. The more the surfaces differ from a flat disk, the more pronounced is this effect.

The above drawback was further exacerbated by a naïve choice of the flattening scheme in Sidorov *et al.* [241], where a simple cylindrical unwrapping (see Fig. 4.2) was used instead of a more elaborate scheme discussed in Section 4.2.3 below. This introduced additional unpredictable deformations dependent on the orientation of the meshes and, therefore, sensitive to the initial alignment with respect to the cylinder. These are essentially noise mixed with useful signal, the genuine deformations of the original meshes that are to be recovered. It should be noted that cylindrical parameterisation was also used in the classic work by Blanz and Vetter [28], in which the cylindrical representation was due to the scanning process which sampled facial geometry $r(h, \varphi)$ as a function of angle φ and height h, and similarly sampled the colour.

Note that in the approach of Sidorov *et al.* [241], the cumulative result of first embedding the meshes into 2D, followed by computing the 2D deformations that bring them into alignment, is still an embedding of 3D meshes into a 2D plane, except performed in two steps.

With this in mind, this chapter offers a more principled approach (an improvement of the approach due to Sidorov *et al.* [241]).

The main idea is to maintain the correspondences between surfaces and to operate with textures in a common flat reference space while performing optimisation on the original 3D surfaces. This can be regarded as gradually

computing the embeddings $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ of the surfaces into a plane such that they also bring all surfaces into alignment simultaneously. To accomplish this, the algorithm begins by computing maximally isometric embeddings (reviewed in Section 4.2.3) which implicitly define correspondences between surfaces and continues iteratively improve them.

This is an improvement because the "embedding model" (the analogue of the deformation model) in the proposed method is optimised on the *original* manifolds, with geodesic distances along the surface of the manifold being used instead of the Euclidean ones in the plane. This makes the registration (embedding) algorithm aware of the shape of the meshes and the above drawback is eliminated. Additionally, performing optimisation on the original 3D meshes gives easy access to the 3D information, such as curvature.

The method described in this chapter is then, technically, a *groupwise embedding* (or a *groupwise parameterisation*) algorithm, but for consistency and parallelism with Chapter 3 it shall continue to be called *registration*, and so shall the changes in embedding be called *deformations*.

The idea of doing registration of surfaces by mapping them to a plane first and applying planar registration has been described in the literature before. Li *et al.* [164], addressing the problem of shape registration via the *shape images* representation, remark that image-based representations of shapes are more useful for registration, than point-based representations, as they provide more constraints and supporting information from neighbouring areas of shape. This, essentially, means that the application of area-based registration methods becomes easier. While Li *et al.* [164] focus on registration of shapes based on geometry only, via shape images, this argument equally applies to photometry-based registration.

Below, the computational tools and concepts which are needed for the exposition of the proposed algorithm are reviewed: geodesic operations on meshes, embedding of meshes into a plane, and filling holes.

4.2.1 Geodesic Distances and Fast Marching

One of the early algorithms for computing geodesic paths on triangulated manifolds was proposed by Mitchell *et al.* in [183]. On a mesh with n edges, their algorithm runs in $O(n^2 \log n)$ time and $O(n^2)$ space. After the initial

pass, the "single source to single destination" distance can be found in $O(\log n)$ time, and the actual path recovered in $O(k + \log n)$, where k is the number of faces crossed by the path.

Later, Chen and Han [55] improved upon this result by proposing an algorithm to compute exact geodesics in $O(n^2)$ time in the worst case.

A breakthrough occurred when a method for solving the Eikonal equation (Eq. (4.3)) on a regular grid of m points in $O(m \log m)$ steps, called Fast Marching Method (FMM), was originally proposed by Sethian [236]. Two years later, Kimmel and Sethian proposed an extension to this method in their seminal paper [144], in which their technique is applied to triangulated domains and has the same computational complexity $O(m \log m)$. The Fast Marching Method (FMM) resembles Dijkstra's algorithm [87] in that a moving front is advanced outward from the source in all directions. The reader is addressed to Kimmel and Sethian [144] and Sethian [238] for full details. It should be noted that a similar algorithm has been proposed even earlier, by Tsitsiklis [273].

Importantly, the appearance of FMM on triangulated domains, such as the original approach of Kimmel and Sethian [144] and their extensions, made the solution to the problem of finding "single source to all targets" and even "all sources to all targets" geodesic paths computationally very cheap.

Following the exposition of Peyré and Cohen [208], the main idea is summarised below. Given a manifold $\mathfrak{M}$, two points on it, $\mathbf{x}_0$ and $\mathbf{x}_1$, and a strictly positive metric $P(s)ds$ defined on $\mathfrak{M}$, the weighted geodesic distance between $\mathbf{x}_0$ and $\mathbf{x}_1$ is defined as (Peyré and Cohen [208])

$$\mathfrak{G}(\mathfrak{M}, \mathbf{x}_0, \mathbf{x}_1) = \min_{\gamma} \left(\int_0^1 \|\gamma'(t)\| P(\gamma(t)) dt \right), \tag{4.1}$$

where γ's are all possible piecewise linear curves on $\mathfrak{M}$ such that $\gamma(0) = \mathbf{x}_0$ and $\gamma(1) = \mathbf{x}_1$. Fixing the point $\mathbf{x}_0$ as the starting point, the distance $U(\mathbf{x}) = \mathfrak{G}(\mathfrak{M}, \mathbf{x}_0, \mathbf{x})$ to all other points, $\mathbf{x}$, can be computed by propagating the level set curve $\mathcal{L}_t = \{\mathbf{x} \colon U(\mathbf{x}) = t\}$ using the evolution equation

$$\frac{\partial \mathcal{L}_t}{\partial t}(\mathbf{x}) = \frac{\mathbf{n}_x}{P(\mathbf{x})}, \tag{4.2}$$

where $\mathbf{n}_x$ is the exterior unit normal to $\mathcal{L}_t$ at point $\mathbf{x}$ and $U(\mathbf{x})$ satisfies the Eikonal equation

$$\|\nabla U(\mathbf{x})\| = P(\mathbf{x}). \tag{4.3}$$

The FMM method of Kimmel and Sethian [144] does just that on triangulated meshes.

A number of improvements to the original FMM has been proposed. For example, Giard and Macq [105] point out redundancies in the previous Fast Marching solutions to the geodesic distance problems and propose a method to estimate geodesic distances between some of the vertices by reusing the information obtained during computation of the geodesic distances between other vertices. This makes the solution even cheaper.

Pointing out that the method of Mitchell *et al.* in [183] typically runs much faster than the worst case analysis suggests (rather, typical time complexity is better than $O(n^2)$), Surazhsky *et al.* [258] extend it and propose a family of fast and accurate approximation algorithms (with bounded error) for computing geodesics in $O(n \log n)$ time. Bommes and Kobbelt [31] generalise the algorithm of Surazhsky *et al.* [258], maintaining its properties, but further allow the geodesic distances from an arbitrary, possibly open, curve on the surface, not just from points, to be computed.

Grossmann *et al.* [117] proposed a voxel-based method for computing geodesic distances on surfaces that are not represented as polygonal meshes. This is useful, for example, when the surface in question is based on some voxel data, such as a cortex boundary in an MRI brain scan.

In the remainder of this chapter, abstracting from the particular algorithm used to compute the geodesic distances, the notation $\mathfrak{G}(\mathfrak{M}, \mathbf{a}, \mathbf{b})$ denotes the geodesic distance between points $\mathbf{a}$ and $\mathbf{b}$ on a mesh $\mathfrak{M}$.

In practice, several good implementations of geodesic path algorithms exist. For example, an implementation of the approach due to Mitchell *et al.* [183] by Kirsanov is found at [147] and [146]. An implementation of the FMM of Kimmel and Sethian [144] by Bronstein is found at [41]. The same method was implemented by Peyré and is available at [206].

4.2.2 Multidimensional Scaling

One of the approaches to embedding a point cloud into a Euclidean space, possibly of lower dimensionality, well known in statistics, is termed Multidimensional Scaling (MDS), see *e.g.* Borg and Groenen [34], Cox *et al.* [72].

Several variants of MDS have been proposed, differing in the objective function and optimisation framework. Below, the main idea behind the classical MDS algorithm, first proposed by Young and Householder in [299], is summarised for completeness, following the exposition given by Platt in [212].

Given n points $\mathbf{x}_i \in \mathbb{R}^k$, $i = 1 \dots n$, a dissimilarity matrix $\mathrm{D}_{n\times n}$ is constructed, such that $\mathrm{D}(i,j) = |\mathbf{x}_i - \mathbf{x}_j|$. The classical MDS algorithm attempts to find a set of n points $\mathbf{y}_i \in \mathbb{R}^m$, $i = 1 \dots n$, with $m \leq k$, to minimise

$$\sum_{i<j} \left(\|\mathbf{y}_i - \mathbf{y}_j\| - \mathrm{D}(i,j) \right)^2 \rightarrow \min_{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n} . \tag{4.4}$$

To do so, first a Gram, or *kernel*, matrix K is computed by "double-centering" the distance matrix $\mathrm{D}_{n\times n}$ (see Platt [212]):

$$\mathrm{K}(i,j) = -\frac{1}{2} \left(\mathrm{D}(i,j)^2 - \frac{1}{n} \sum_{i=1\dots n} \mathrm{D}(i,j)^2 - \frac{1}{n} \sum_{j=1\dots n} \mathrm{D}(i,j)^2 + \right.$$
$$\left. + \frac{1}{n^2} \sum_{i=1\dots n} \sum_{j=1\dots n} \mathrm{D}(i,j)^2 \right) = \left(\mathrm{I}_{n\times n} - \frac{\mathbf{1}_{n\times n}}{n} \right) \left(-\frac{1}{2} \mathrm{D} \bullet \mathrm{D} \right) \left(\mathrm{I}_{n\times n} - \frac{\mathbf{1}_{n\times n}}{n} \right) . \tag{4.5}$$

If D is a Euclidean (in $\mathbb{R}^n$) distance matrix, then K is a positive semi-definite symmetric matrix of dot-products of points' radius vectors in the same space (Schölkopf [232]). The matrix K is then decomposed using EVD into $\mathrm{K} = \mathrm{ALA}^T$, where the eigenvectors (columns of A) constitute the derived coordinates up to a similarity transform. The embedding then consists of simply selecting m ($m \leq k$) eigenvectors from A corresponding to the m largest eigenvalues in L.

When the size, n-by-n, of the dissimilarity matrix becomes large, the solution of the eigenvalue problem (with time complexity $O(n^3)$, see the original paper by Кублановская (Kublanovskaya) [305]) may become unfeasible due to time and space limitations. This can be overcome by employing a variant of MDS that scales better with the size of the dissimilarity matrix. Options include: Landmark MDS (De Silva and Tenenbaum [82]), FastMap (Faloutsos and Lin [96]), MetricMap (Wang *et al.* [291]). These are compared and discussed by Platt in [212].

Alternatively, the size of the dissimilarity matrix may be reduced by reducing the size of the point cloud. Since, as explained below, MDS is used in the proposed method for embedding of mesh vertices into a plane, this can be accomplished by strategic decimation of the meshes, as discussed in Section 4.5.1.

It is worth mentioning that embedding can be done onto another manifold instead of a Euclidean space. A method for doing so, called Generalised Multidimensional Scaling (GMDS) was proposed by Bronstein *et al.* [47–49]. The possible applications of GMDS in the context of this chapter are discussed in Section 4.7.

4.2.3 Flattening, Parameterisation and Bending Invariants

Parameterisation of triangulated manifolds, or meshes, is a fundamental and frequently used operation in computational geometry and computer graphics (Desbrun *et al.* [85]). Embedding and parameterisation of surfaces are two closely related concepts. Both are concerned with bijective mapping of a surface to a lower dimensional Euclidean space (usually a plane, in which case the term *flattening* is sometimes used) while trying to preserve some desirable properties, local, global, or both, such as distances, angles, areas, connectivity *etc.* The term *parameterisation* usually refers to a special case[2] of embedding that bijectively maps the surface in question to a *unit square* (sometimes a unit disk) on a plane, thus mapping the Euclidean coordinates on a unit square to the corresponding curvilinear coordinates (Moon and Spencer [184]) on the original surface. Embedding of a triangulated manifold, or mesh, amounts to embedding its vertices in such a way that the resulting planar mesh is isomorphic to the original mesh (Desbrun *et al.* [85]).

The purpose of embedding or parameterising meshes is to simplify processing, by performing operations in the flat Euclidean space rather than on the curved surface. The benefits of doing so were originally realised by Schwartz *et al.* [233], who applied MDS to flattening of the surface of a macaque monkey's visual cortex (a highly convoluted surface).

Not all surfaces can be bijectively mapped to a plane, so it must be stipulated that from now on the discussion is focused on $\mathbb{R}^2$-embeddable surfaces, in other words surfaces that can be bijectively embedded into $\mathbb{R}^2$.

There are presently many methods for mesh flattening and parameterisation, and some theoretical breakthroughs have been made in the last decade. A classical approach, due to Floater [99], is an early example of a shape

[2]However, in Peyré and Cohen [208] the meaning is reversed: flattening, or embedding in $\mathbb{R}^2$, is understood as a particular approach to parameterisation.

preserving parameterisation based on the graph drawing theory. A number of improvements to Floater's original parameterisation scheme has been proposed. For example, the heuristic method of Yoshizawa *et al.* [298] starts with Floater's parameterisation and proceeds to improve it via an iterative process which minimises the weighted quadratic stretch energy. Another approach due to Floater *et al.*, is found in [100]. The main idea of their approach is to partition a complex triangulated surface into geodesically-triangular patches (*i.e.* patches bounded by three geodesic curves) and then parameterise each patch individually. The so obtained coarse triangulation is then parameterised globally.

Desbrun *et al.* [85] proposed a family of approaches, called *Intrinsic Parameterisations* that are capable of finding parameterisations minimising the distortion of some intrinsic measure in a linear-algebraic framework. An important contribution of Desbrun *et al.* [85] is the detailed discussion of various types of energies that can be used to govern the flattening of a mesh.

Sander *et al.* [229] propose a method of parameterisation, especially useful in the case of textured surfaces, that minimises the texture stretch ensuring homogeneous sampling density of the texture. The method of Sander *et al.* [229] is shown to deal with closed surfaces as well as disks: a procedure for partitioning the surfaces into "charts" is proposed as well is a procedure for combining the individual charts into an atlas.

Of special interest in the context of this chapter is a family of parameterisation methods that try to *preserve distances*. For such purpose, MDS is a useful technique as it computes the embedding that minimises the distortion of distances (stress) supplied to it in the dissimilarity matrix. Such embeddings that preserve distances are called *isometric*. If MDS is fed a matrix of pairwise *geodesic* distances between points on a mesh, the resulting Euclidean distances between the points on the embedded mesh will approximate the original geodesic distances as closely as possible (*e.g.* in least squares sense, depending on the particular formulation of MDS). In case the embedding is performed into $\mathbb{R}^3$, the result is a *bending invariant*, a concept first proposed by Elad and Kimmel [93] (but see also Elad and Kimmel [94]).

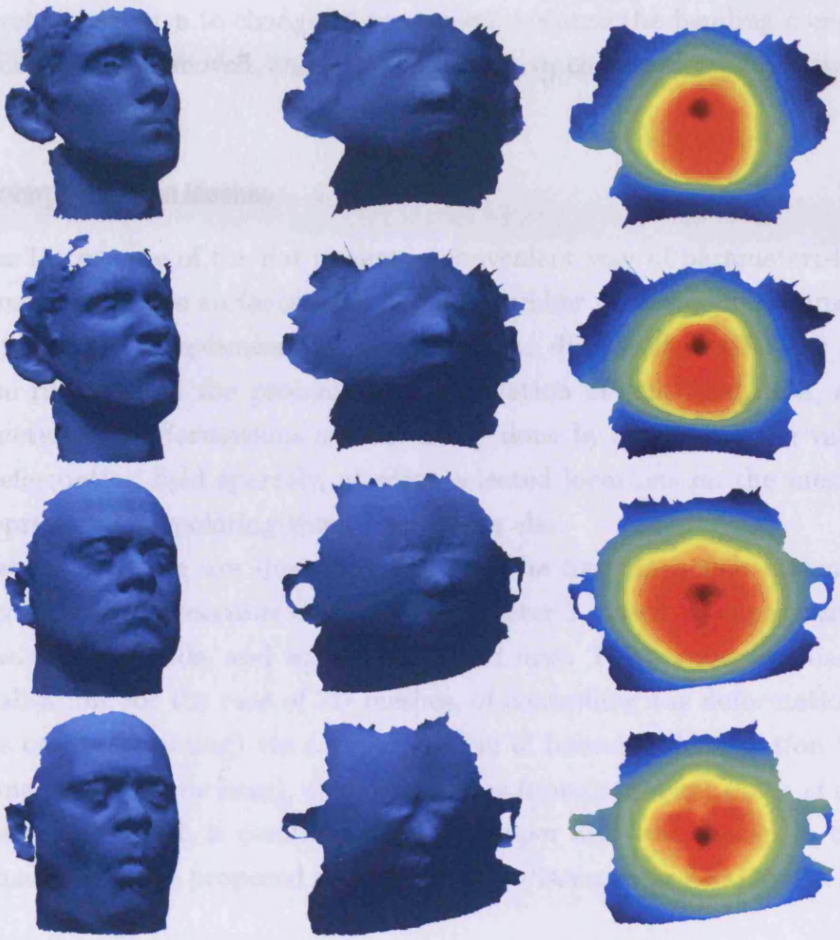
Parenthetically it should be noted that, in general, a surface cannot be embedded into *any* $\mathbb{R}^n$ perfectly isometrically (without distortion of distances), unless it is of a special kind (see Linial *et al.* [166]). For example, mapping

of a 3D surface onto a 2D plane can only be isometric if the surface has zero Gaussian curvature (see *e.g.* Do-Carmo [88] or O'Neil [196]), otherwise distortion of distances occurs. MDS computes the best approximation to isometry.

Bronstein *et al.* [42–44, 50, 51] showed empirically that geodesic distances on the surface of a face are significantly less sensitive to changes in expression than Euclidean distances. This means, therefore, that bending invariants of Elad and Kimmel [93], in which Euclidean distances between points correspond to the geodesic distances on the original surfaces, change only slightly due to the change in facial expression and so can be used as an excellent starting point with which to initialise the proposed registration algorithm. This is indeed a useful property, because a small part of the registration job is essentially done just by embedding alone. (Of course the invariance of geodesic distances to expression change is only very approximate, as the geodesic distances between corresponding points on the face still do sometimes change significantly, for example when yawning). This useful property was applied by Bronstein *et al.* [42–45, 50, 51] to the task of face recognition. In principle, this property of bending invariants is useful not only in the case of craniofacial imagery, but, in general, in all cases in which removing the bending component of the deformation, leaving only stretching, assists in establishing correspondences.

It should be noted that isometric embedding need not necessarily be performed into a flat Euclidean space: Bronstein *et al.* [44, 45] discuss embeddings into spaces with spherical and hyperbolic geometries, the choice of the space being governed by the anticipated embedding error. Bronstein *et al.* [44, 45] indeed show that embedding of face scans onto a sphere leads to smaller embedding errors and, in turn, leads to better performance of their recognition algorithm.

The bending invariants of some meshes can be seen in Fig. 4.3. In the left column (Fig. 4.3a) the original meshes in $\mathbb{R}^3$ are shown, in the middle (Fig. 4.3b) — their bending invariants (choose any two points on the original mesh and compare the geodesic distance between them with the geodesic distance between the corresponding points on the bending invariant), and in the right column (Fig. 4.3c) — the result of embedding the meshes into $\mathbb{R}^2$, with the discarded third component of the bending invariant shown in colour.



(a) Original mesh. (b) Bending invariant. (c) Embedding into $\mathbb{R}^2$, with discarded depth colour-coded.

FIGURE 4.3: Bending invariants and embedding of meshes into $\mathbb{R}^2$.

The distance-preserving property of MDS has been exploited in a number of works on mesh parameterisation and flattening. For example, in the classic paper by Zigelman *et al.* [302], MDS is applied to the problem of distortion-minimising texture mapping. The method of Zigelman *et al.* [302] is shown to compare favourably to that of Floater *et al.* in [100].

Grossmann *et al.* [117] discuss the application of MDS to flattening of surfaces that are not represented by polygonal meshes, with their main contribution being a novel voxel-based method for estimating geodesic distances.

To summarise, the main advantage of MDS-based parameterisation, in the context of this chapter, is that the embedding is maximally isometric. This

allows to compute bending invariants of meshes. The bending invariants are relatively insensitive to change in expression, because the bending component of deformation is removed, and only the stretch, or change of geodesic distances, remains.

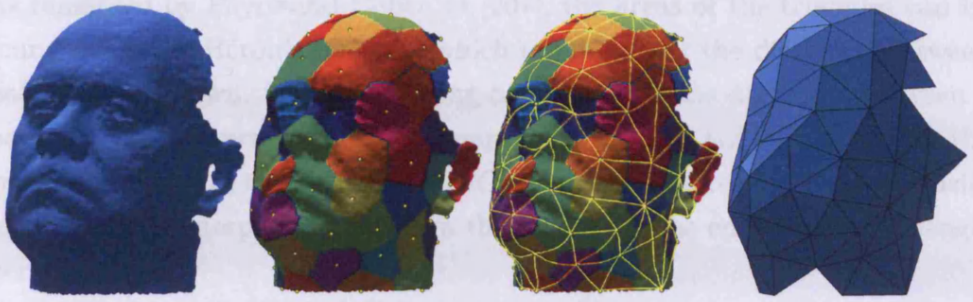
4.2.4 Interpolation on Meshes

Just as in the case of the flat images, a convenient way of parameterising the deformations of the surfaces with a small number of parameters, amenable to numeric global optimisation, is needed. As discussed in Chapter 3, this can be regarded as the problem of interpolation of scattered data, and so parameterising deformations is most easily done by specifying the values of the deformation field sparsely, at some selected locations on the mesh, and appropriately interpolating them everywhere else.

Two approaches are discussed below. The first approach attempts to closely mimic the procedure discussed in Chapter 3, based on superposition of piecewise affine fields, and will be addressed first. The second approach is a generalisation, for the case of 3D meshes, of controlling the deformations (or, in this case, embedding) via a superposition of bounded deformation "disks" (residing on the 3D surfaces), analogous to the formulation in Cootes *et al.* [62]. The latter approach is computationally cheaper and was chosen to control deformations in the proposed algorithm. It is discussed in Section 4.4.1.

4.2.5 Geodesic Delaunay Triangulation on Meshes

The idea to introduce an analogue of Delaunay triangulation on an arbitrary triangulated manifold has first been discussed in the remeshing literature (*e.g.* Peyré and Cohen [207]), where it is used as a straightforward solution to the problem of resampling and remeshing a triangulated manifold. The simple yet effective approach of Peyré and Cohen [207] is to first select a set of points on a surface, using some appropriate sampling technique, then to compute an analogue of the Delaunay triangulation of those points (on the original manifold, with edges of the "triangles" being the geodesic curves between vertices), thus yielding a new triangulated mesh. A conceptually similar idea was used by Floater *et al.* in [100] to partition a complex mesh into large geodesic triangles which are then parameterised individually.



(a) Original mesh. (b) Geodesic Voronoi cells with centres at yellow dots. (c) Geodesic Delaunay triangulation. (d) A very coarse mesh obtained from this triangulation.

FIGURE 4.4: Geodesic Delaunay triangulation on meshes, using the approach due to Peyré and Cohen [207, 209].

Given the operation $\mathfrak{G}(\mathfrak{M}, \mathbf{a}, \mathbf{b})$ which computes geodesic distances between any two point, geodesic Delaunay triangulation can be easily obtained by first computing a geodesic Voronoi diagram (essentially, finding for each vertex on the original dense mesh the geodesically nearest selected control point) and computing its dual (Peyré and Cohen [207]). This is illustrated in Fig. 4.4: the geodesic Voronoi diagram, Fig. 4.4b, of the original mesh, Fig. 4.4a, is used to produce the geodesic Delaunay triangulation, Fig. 4.4c, which Peyré and Cohen [207] use for decimation, Fig. 4.4d, of the original mesh.

This idea can be adapted to interpolation on meshes. Piecewise affine interpolation on a plane, discussed in Section 3.4.1, can be to some extent generalised to triangulated manifolds. Recall, from Section 3.4.1, that in the case of piecewise affine interpolation two operations are required to find the interpolated value at some point on the triangulated domain: the point-in-triangle test to find the encompassing triangle, and a way to interpolate between the values defined at vertices of that triangle.

A geodesic analogue of these operations was proposed by Peyré and Cohen in [207]. Indeed, since distances between any two points on a mesh can be readily computed with fast marching methods, an analogue of *areal coordinates* can be used instead of barycentric coordinates.

Denote by $S_{\triangle ABC}$ the area of the a triangle $\triangle ABC$. Assume that a point P is inside the triangle $\triangle ABC$, and $D = S_{\triangle PAB} + S_{\triangle PBC} + S_{\triangle PCA}$. Then the areal coordinates of P are: $\{\lambda_1 = S_{PAB}/D, \lambda_2 = S_{PBC}/D, \lambda_3 = S_{PCA}/D\}$.

As remarked by Peyré and Cohen in [207], the areas of the triangles can be found using the Heron's formula, which requires only the distances between points to be known. Therefore, using only the geodesic distances between a point P and the vertices of the encompassing triangle $\triangle ABC$, as well as the geodesic lengths of the sides of $\triangle ABC$, areal coordinates of P can be easily found and the interpolation between the vertices of the encompassing triangle performed.

The deformation fields defined by the geodesic Delaunay triangulations as in Fig. 4.4c can be added together, by summing their influence at each vertex. This allows for a compete analogue of the algorithm proposed in Chapter 3 to be implemented. However, computing geodesic Delaunay triangulation is not a cheap operation and so cannot be performed frequently, as required by the method of Chapter 3. So, while elegant, this solution is not very practical. Instead, a superposition of radial basis functions, detailed in Section 4.4.1, which is cheaper to compute, is used in the proposed method.

4.2.6 Filling Holes

Finally, a method of dealing with imperfect data will be needed in the proposed algorithm, particularly for resampling the registered meshes in Section 4.4.5. Holes in the textures and shape images, inevitably arising from the deficiencies of the scanning process, can be filled using Poisson interpolation as proposed in the seminal paper by Pérez *et al.* [204].

The main idea behind Poisson interpolation, presented here following the exposition of Pérez *et al.* [204], is to complete the missing areas in an image as smoothly as possible. Let $F^* = F^*(\mathbf{r})$ be a scalar function from pixel coordinates, $\mathbb{R}^2$, to pixel values, representing the pixel values in the known area of the image. Similarly let $F = F(\mathbf{r})$ be the image in the unknown area, and let Ω represent the domain of the unknown area. The desired interpolant F is the solution to the following minimisation problem (Pérez *et al.* [204]):

$$\arg\min_F \iint_\Omega |\nabla F|^2, \quad \text{subject to } F|_{\partial\Omega} = F^*|_{\partial\Omega}, \tag{4.6}$$

with the minimiser F satisfying the corresponding Euler-Lagrange equation:

$$\nabla^2 F = 0 \text{ over } \Omega, \quad \text{subject to } F|_{\partial\Omega} = F^*|_{\partial\Omega}. \tag{4.7}$$

One solution to this problem, a somewhat simplified variant of the procedure due to Pérez *et al.* [204], is summarised below, following the exposition in Leyvand [163].

On a discrete pixel grid of an image F, the analogue $\nabla^2\mathrm{F}$ of the Laplacian $\nabla^2 F$ can be approximated as

$$\begin{aligned}\nabla^2\mathrm{F}(x,y) &\approx \mathrm{F}(x+1,y) - 2\mathrm{F}(x,y) + \mathrm{F}(x-1,y) + \\ &\qquad \mathrm{F}(x,y+1) - 2\mathrm{F}(x,y) + \mathrm{F}(x,y-1) = \\ &= \mathrm{F}(x+1,y) + \mathrm{F}(x-1,y) + \mathrm{F}(x,y+1) + \mathrm{F}(x,y-1) - 4\mathrm{F}(x,y) = 0 \quad (4.8)\end{aligned}$$

because the partial derivatives can be approximated, in the discrete case, with finite differences:

$$\frac{\partial \mathrm{F}}{\partial x} \approx \mathrm{F}(x+1,y) - \mathrm{F}(x,y) \quad \text{and} \quad \frac{\partial^2 \mathrm{F}}{\partial x^2} \approx \mathrm{F}(x+1,y) - 2\mathrm{F}(x,y) + \mathrm{F}(x-1,y). \tag{4.9}$$

Suppose there are n pixels in the "unknown" area of the image. One can construct a sparse linear system, relating the values of the missing pixels to the values of known pixels, with n equations and n unknowns, in which each unknown corresponds to the value of a missing pixel. The solution of such a system gives the desired interpolation. Let the unknowns be $u^{(1)}, u^{(2)}, \ldots, u^{(n)}$ corresponding to some unknown pixel values at $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. If the value of a pixel at (x, y) is unknown and the value of a pixel just above it is also unknown, denote by $u^{\uparrow}_{x,y}$ the unknown variable corresponding to the pixel above (x, y). Similarly denote by $u^{\downarrow}_{x,y}$, $u^{\leftarrow}_{x,y}$ and $u^{\rightarrow}_{x,y}$ the unknown variables corresponding to pixels below, to the left, and to the right of the pixel at (x, y) respectively. An unknown pixel at (x, y), surrounded by four unknown pixels (in the sense of the 4-connected neighbourhood), gives rise to an equation

$$u^{\uparrow}_{x,y} + u^{\leftarrow}_{x,y} - 4u_{x,y} + u^{\rightarrow}_{x,y} + u^{\downarrow}_{x,y} = 0. \tag{4.10}$$

If, say, the value of a pixel directly above it is known ($= \mathrm{F}(x, y-1)$) then the equation becomes

$$\mathrm{F}(x,y-1) + u^{\leftarrow}_{x,y} - 4u_{x,y} + u^{\rightarrow}_{x,y} + u^{\downarrow}_{x,y} = 0, \text{ or, rearranging,} \tag{4.11}$$

$$u^{\leftarrow}_{x,y} - 4u_{x,y} + u^{\rightarrow}_{x,y} + u^{\downarrow}_{x,y} = -\mathrm{F}(x,y-1). \tag{4.12}$$

For all other situations of known and unknown pixels the equations are formed in a similar fashion, substituting the unknown variables with known values.

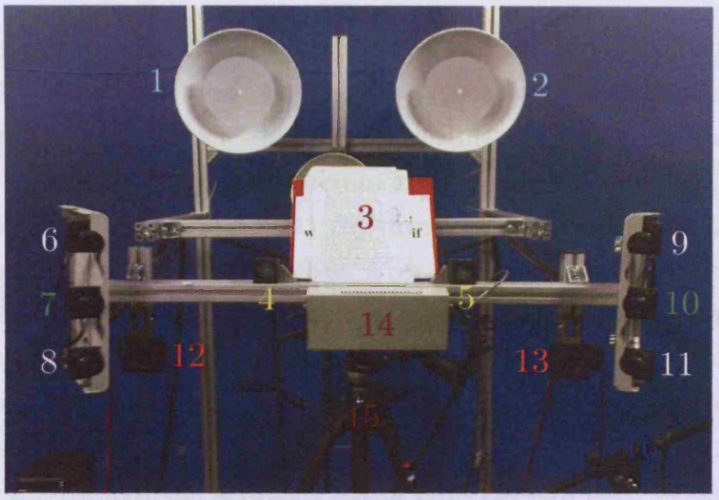


FIGURE 4.5: Set up of the video-rate 3D surface scanner [6], frontal view. 1, 2 — powerful "cold" lights; 3 — stand to hold the instructions for the subject; 4, 5 — "warm" lights; 6, 8, 9, 11 — infrared cameras; 7, 10 — colour (texture) cameras; 12, 13 — infrared pattern projectors; 14 — power supply; 15 — tripod and support frame; 16 — microphone stand.

The known pixels values contribute, therefore, to the right hand side of the equations after rearranging the terms, as, for example, in Eq. (4.12). The complete set of such equations, one for each unknown pixel, has the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{4.13}$$

where A is a sparse matrix of coefficients, $\mathbf{x}$ is a vector of unknowns and $\mathbf{b}$ is the right hand side — contributions from the known pixels. Solving Eq. (4.13) for $\mathbf{x}$ gives the interpolated values for the missing pixels.

It is possible to apply the same concept to fill holes directly on triangulated meshes, as, for example, proposed by Zhao *et al.* [301].

4.3 Data Acquisition and Preparation

Figure 4.5 illustrates the set up of the video-rate 3D surface scanner ("4D camera") that was used for data acquisition in the experiments in Section 4.6. This is a slightly customised version of a commercial photogrammetric system [6]. The four infrared cameras (ordinary grayscale cameras with an infrared filter fitted), two on the left (6, 8 in Fig. 4.5) and two on the right (9, 11 in Fig. 4.5), with sufficient vertical separation, capture the speckle pattern

which is projected onto the subject by the two infrared projectors (12, 13 in Fig. 4.5). More than one projector is typically required to better cover the side areas of the subject's head. The resulting speckled infrared images are shown in Fig. 4.6, top and bottom rows, together with the texture of the subject (Fig. 4.6, middle row) captured by two horizontally separated ordinary colour cameras (7, 10 in Fig. 4.5). The detail of the infrared patter is shown in Fig. 4.8, the same small area of the face (in this example, bridge of the nose) as it is seen from the four infrared cameras. For completeness, Fig. 4.7 shows the side view of the experimental setup, supplemented with an adjustable seat for the subject, a microphone for recording synchronous audio and, more importantly, the arrangement of light sources used to create a more controllable light situation. Powerful "cold" gas-discharge lights on both sides of the subject, as well as two in front (1, 2 in Fig. 4.5 and also in Fig. 4.7) serve to provide favourable lighting conditions without overpowering the infrared emissions from the projectors. The spectrum of the light is fine-tuned by addition of small "warm" incandescent light sources (4, 5 in Fig. 4.5).

4.3.1 Mesh Representation

There are many ways of representing triangle meshes. Here, for ease of explanation, the simplest useful representation — the *Face-Vertex* mesh — is adopted. A triangle mesh of this form is a set of n_f faces (triangles) and n_v vertices: $\mathfrak{M} = \{\mathrm{V}_{3\times n_v}, \mathrm{F}_{3\times n_f}\}$, where the matrix $\mathrm{V}_{3\times n_v}$ contains as columns the coordinates of all vertices, and each column of the matrix $\mathrm{F}_{3\times n_f}$ contains the three indices (into columns of V) of the three vertices of each triangle. Later in this chapter, when dealing with textured meshes, the above structure is augmented to additionally contain a texture map, $\mathcal{T}$, and, for each vertex, the corresponding texture coordinates, U, in the space of the texture map: $\mathfrak{M} = \{\mathrm{V}_{3\times n_v}, \mathrm{F}_{3\times n_f}, \mathrm{U}_{2\times n_v}, \mathcal{T}_{w\times h\times c}\}$.

A mesh can also be regarded as a graph, with vertices of the mesh corresponding to the vertices of the graph and the edges of the mesh faces corresponding to the edges of the graph (*vertex-edge* connectedness). Alternatively, a mesh can be regarded as a graph in which mesh faces correspond to the vertices of the graph and the edges shared between two adjacent faces of the mesh correspond to edges of the graph (*face-edge* connectedness).

FIGURE 4.6: Images acquired by the six cameras. Layout of the images corresponds to the layout of the cameras in Fig. 4.5.

FIGURE 4.7: Same 3D scanner as in Fig. 4.5, side view.

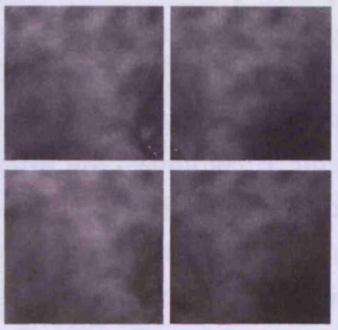


FIGURE 4.8: Patches of the infrared speckle (bridge of the nose) as seen by the four infrared cameras.

FIGURE 4.9: *Left:* the original texture map composed of two separate views of the subject. *Right:* the corresponding resampled continuous texture map.

4.3.2 Mesh cleaning

The raw meshes that come from scanners are often imperfect and are not readily suited to the processing discussed later in this chapter. The geometry is served as a disorganised triangle soup. Due to imperfections of the scanning process the surfaces contain holes (missing triangles), there are disjoint regions, duplicate vertices *etc.* The preliminary step is therefore to "clean" the input

meshes. Without such cleaning, algorithms that assume the meshes to be well-formed manifolds, for example the geodesic path algorithms (Section 4.2.1), will not produce the expected results.

First, the duplicate vertices are removed and the triangle soup is converted to a Face-Vertex representation (see Section 4.3.1).

Since the textures in the experimental setup are acquired with two spatially separated cameras, see Section 4.3 and Fig. 4.5, the texture map is discontinuous, Fig. 4.6 (middle row), and so it is impossible to trivially remap the original texture coordinates to the Face-Vertex representation: in the original triangle soup more than one vertex can exist with the same 3D coordinates, but with different texture coordinates, corresponding to the disjoint parts of the texture. Therefore, as the second step, the textures are resampled to produce one continuous texture map. This procedure is optionally combined with mesh decimation, see Section 4.5.1, in which case the texture is resampled so as to conform to the decimated mesh. The discontinuous raw texture from cameras and the resampled continuous texture map are shown in Fig. 4.9.

The input meshes might contain several disjoint objects, for example the main subject and small bits of the background scene. To filter those out, the connected components of the mesh graph (see Section 4.3.1) using vertex-edge connectedness are computed and the connected component with the largest number of vertices in it is selected. The largest connected component is assumed to contain the main subject (RoI), and the other connected components are discarded. The procedure is repeated using the face-edge connectedness. Finally, other pathological cases, *e.g.* more than two triangles sharing an edge, are detected and removed.

After this cleaning procedure the meshes are assumed to be well formed manifolds, with one connected component containing the subject, and with a continuous texture map.

Note that the input meshes contain different number of vertices and have vastly different topologies, see Fig. 4.10, and so no assumptions can be made about the correspondence of vertices between meshes, even if they are of the same object.

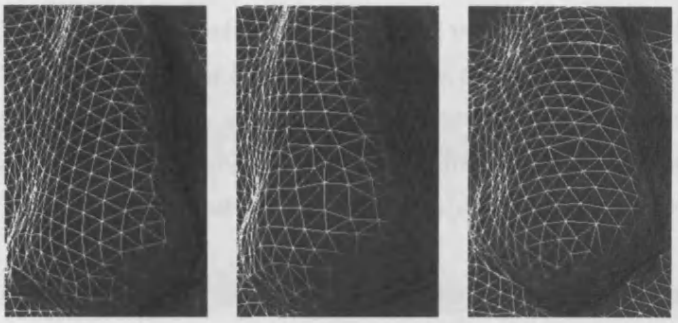


FIGURE 4.10: Illustration of the mesh topology problem. Three meshes obtained from a 3D scanner approximating the same human nose, at small time intervals. Notice how the topology of the meshes is very different even though variation of shape is minimal.

4.4 Groupwise Registration of Textured Meshes

The proposed algorithm takes as input as set of N textured meshes, $\mathfrak{M}_i$, or, more precisely, using the symbols defined in Section 4.3.1, $\mathfrak{M}_i = \{\mathrm{F}_i, \mathrm{V}_i, \mathrm{U}_i, \mathcal{T}_i\}$.

For every mesh, the algorithm maintains a $2 \times n_{v_i}$ matrix E_i of the embedded vertex coordinates in the common reference plane. The algorithm aims to find such embeddings E_i that bring the analogous features in all meshes into alignment. Having done this, one can easily recover the correspondences between points on the original 3D meshes, assuming the embedding is bijective.

As in Sidorov *et al.* [243], Davies *et al.* [80], the problem of groupwise registration is regarded here as an optimisation problem consisting of three components: a mechanism for representing and manipulating deformations (changes in embedding), an objective function F measuring the alignment error, and a global minimisation algorithm which optimises F. These components are addressed below.

4.4.1 Model of Deformation (Embedding)

Radial basis interpolation on meshes proves useful in applications like defining surface vector fields or mesh watermarking (Praun *et al.* [217, 218]) in which certain values are specified at specific points on a surface and have to be interpolated elsewhere. (Praun *et al.* [217, 218] use Dijkstra's algorithm [87] to approximate true geodesic distances.)

For the method proposed in this chapter, deformations need only be *specified* at a small number of control points and smoothly interpolated elsewhere, but

not necessarily interpolated *between* the control points. It is therefore sufficient to represent the total deformations as a superposition of some radial basis functions, without necessarily solving for the contribution coefficients to ensure that the superposition interpolates some function between the control points. This was the approach successfully adopted by Lötjönen and Mäkelä [169] and Cootes *et al.* [62].

Suppose n_b vertices on the mesh are selected as control points (centres of the "deformation disks", in this algorithm residing *on the surfaces* of the meshes). Radii of the disks are selected to be proportional to the average distance between the nearest disks. By analogy with Eqs. (3.20) and (3.21), using the elementary warp formulation of Cootes *et al.* [62], let the influence of a disk be a function of the *geodesic* distance along the surface of the mesh from the disk's centre. For a vertex $\mathbf{v}$ on a mesh $\mathfrak{M}$, the influence of a disk with geodesic radius r and centre at $\mathbf{c} \in \mathfrak{M}$ can be written, using the symbol $\mathfrak{G}$ from Section 4.2.1, as

$$B(\mathfrak{M}, \mathbf{v}, \mathbf{c}, r) = \begin{cases} 1 - d^2\left(1 + \log(d^2)\right), & d \in (0, r) \\ 1, & d = 0 \\ 0, & d \geq r \end{cases} \quad \text{where } d = \mathfrak{G}(\mathfrak{M}, \mathbf{c}, \mathbf{v}). \tag{4.14}$$

Note the desirable properties of this representation: since the magnitude of influence of a disk depends only on the geodesic distance from its centre, there is no need to define a coordinate system on the mesh, unlike, for example, with Eq. (3.18).

If $\mathbf{p}_{2\times 1}$ is a vector of parameters controlling the contribution of this disk (displacement at its centre), the coordinates of the embedded vertices E_i are affected by $\Delta\mathrm{E}_i = \mathbf{p}B(\mathfrak{M}, \mathbf{v}, \mathbf{c}, r)$. Given n_b disks, the complete configuration space (the space of all possible deformations) is described by a matrix of parameters $\mathrm{P}_{2\times n_b}$, where the columns are parameters (contributions) of the individual disks $\mathrm{P}(:, i) = \mathbf{p}_i$.

Once the disks $\{\mathbf{c}_i, r_i\}$ are selected, the magnitudes of their influences on vertices of the mesh can be precomputed and stored in the *influence matrix* $\mathrm{Q}_{n_b \times n_v}$, where the i-th row stores the influence of the i-th disk on the vertices:

$$\mathrm{Q}(i, j) = B\left(\mathfrak{M}, \mathrm{V}(:, j), \mathbf{c}_i, r_i\right). \tag{4.15}$$

FIGURE 4.11: *Left:* magnitude of influence of several randomly placed individual disks (using Eq. (4.14) with $r = 110$ mm). *Right:* colour-coded mixture of their influence.

Given Q, the effect of all disks together on all vertices of the mesh is then simply $\Delta D = PQ$. The effect of individual deformation disks as well as their superposition is illustrated in Fig. 4.11.

4.4.2 Objective Function

Now, the objective function is addressed. The purpose of the objective function is to measure how well the correspondences between the analogous features on different surfaces have been established. Instead of operating on the correspondences between surfaces directly, the algorithm operates with correspondences between the surfaces and a common reference space (a Euclidean plane, assuming the surfaces can be bijectively embedded in it). Mapping the textures to this reference space makes it possible to adopt any suitable intensity-based objective function from the 2D image registration literature: the textures in the flat parametric space can be manipulated as ordinary images. This also facilitates GPU-based implementation.

The standard practice in groupwise image registration literature is to maintain an evolving model of pixel colours (*e.g.* average of shape-normalised images) in some reference space to which all samples are aligned. In the proposed algorithm, such a model of texture can also be easily computed by mapping the textures from curved surfaces onto the common reference plane using the estimated correspondences between surfaces and the reference plane and averaging them.

Figure 4.12 illustrates the idea: vertices of the mesh are mapped to the reference plane, in which the evolving model of texture is maintained. The correspondences between any points on any two meshes can be consistently deduced given the correspondences between the points and the reference

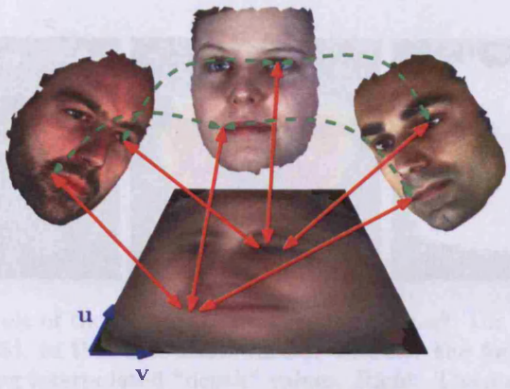


FIGURE 4.12: Correspondences via a flat parametric space.

plane. In practice, the mapping between surfaces and the reference plane can be represented by specifying for each vertex of a mesh the corresponding coordinates in the reference plane, storing these as columns in a $2 \times n_{v_i}$ matrix E_i, and interpolating between the vertices.

In practice, operations on the textures in the reference plane can be most easily performed on a discrete grid. Two operations are now defined which map the textured surfaces to such a discrete buffer in the reference plane.

Let $\{\mathcal{B}_{h\times w\times 4}, \mathcal{M}\} = \mathfrak{R}(\mathrm{F}, \mathrm{E}_{2\times n_v}, \mathrm{U}_{2\times n_v}, \mathcal{T}_{h_t\times w_t\times 4})$ denote the result of rasterising a textured mesh, with connectivity defined by faces F, and a 4-channel texture $\mathcal{T}$ into a $w \times h$ buffer $\mathcal{B}$ with 4-component pixels, using columns of E as target vertex coordinates in the reference plane, and columns of U as the texture coordinates. (This essentially amounts to piecewise affine warping of texture $\mathcal{T}$, using the triangulation defined by F and using U and E as the source and the destination coordinates of vertices respectively.) A stencil mask, $\mathcal{M}_{h\times w\times 4}$, which records the pixels of the buffer affected by the rasterised mesh as 1's, with 0's elsewhere, is also returned. The size of the buffer may be chosen to accommodate the entire rasterised mesh or just the RoI. This is illustrated in Fig. 4.13.

Additionally, let $\{\mathcal{B}_{h\times w\times 4}, \mathcal{M}\} = \mathfrak{R}'(\mathrm{F}, \mathrm{E}_{2\times n_v}, \mathrm{U}_{2\times n_v}, \mathrm{Z}_{1\times N_v}, \mathcal{T}_{h_t\times w_t\times 3})$ similarly denote the result of rasterising a textured mesh, but this time the first three channels of the buffer, $\mathcal{B}(:,:,1{:}3)$, is the result of rendering the textured mesh (equipped with a 3-channel texture), and the fourth channel of the buffer, $\mathcal{B}(:,:,4)$, receives the interpolated values of depth Z, appropriately scaled. Incorporating the depth component in addition to pixel colours helps to more

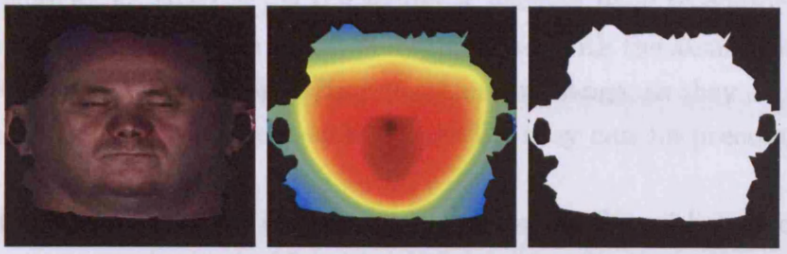


FIGURE 4.13: The result of the rasterisation operation $\mathfrak{R}$. *Left:* the first three components of the buffer, $\mathcal{B}(:,:,1{:}3)$, in this case RGB values. *Middle:* the fourth component of the buffer $\mathcal{B}(:,:,4)$ receiving interpolated "depth" values. *Right:* The stencil mask $\mathcal{M}$.

quickly perform rough alignment in the early stages of registration, but is detrimental in the later stages, where it is not used.

Note that the above operations can be trivially performed on any GPU and are easily implemented on a CPU.

Suppose at some point the "current" estimate of the model of texture is $\mathcal{R}$. For a mesh $\mathfrak{M}_i$ with the initial embedding E_i and a computed improvement D_{i_k} to this embedding, the quality of alignment (embedding) can be evaluated by comparing the rasterised versions of the embedded mesh with the model $\mathcal{R}$. As advocated in Cootes *et al.* [67], it is preferable to compare the model, warped using the current estimate of the correspondences, with the original undeformed samples: in other words, measuring how well the model of texture "explains" the original samples.

The purpose of the *local* objective function is to evaluate a particular embedding hypothesis

$$\mathrm{H(P)} = \mathrm{E}_i + \mathrm{D}_{i_k} + \Delta\mathrm{D}_{i_k} = \mathrm{E}_i + \mathrm{D}_{i_k} + \mathrm{PQ}_{i_k}, \tag{4.16}$$

where E_i is the initial embedding, D_{i_k} is the "current" accumulated embedding improvement, and $\Delta\mathrm{D}_{i_k} = \mathrm{PQ}_{i_k}$ is the hypothetical improvement due to geodesic deformation disks, with the influence matrix Q_{i_k} and parameters P. The affine component of the transformation is omitted for brevity: assume henceforth that before the non-rigid stage all E_i are already affinely aligned.

Using the above notation, the local objective function which evaluates a hypothesis H(P) is

$$C(\mathcal{R}, \mathrm{H(P)}) = S(\{\mathcal{B}_{\mathrm{src}}, \mathcal{M}_{\mathrm{src}}\}, \mathcal{B}_{\mathrm{ref}}), \tag{4.17}$$

where $\{\mathcal{B}_{\mathrm{src}}, \mathcal{M}_{\mathrm{src}}\} = \mathfrak{R}'(\mathrm{F}_i, \mathrm{E}_i, \mathrm{U}_i, \mathrm{Z}_i, \mathcal{T}_i)$ is the rasterisation of the *original* flattened mesh, which is to be explained by the deformed model, and

$\{\mathcal{B}_{\text{ref}}, \cdot\} = \mathfrak{R}(\mathrm{F}_i, \mathrm{E}_i, \mathrm{H}(\mathrm{P}), \mathcal{R})$ is the model $\mathcal{R}$ warped back to conform to the original mesh in the reference space. Note that since with the assumption above, after the affine alignment stage $\{\mathcal{B}_{\text{src}}, \mathcal{M}_{\text{src}}\}$ never change, as they depend only on the initial, affinely aligned, embeddings E_i, they can be precomputed in advance.

Function $S(\{\mathcal{B}_{\text{src}}, \mathcal{M}_{\text{src}}\}, \mathcal{B}_{\text{ref}})$ compares the buffer $\mathcal{B}_{\text{ref}}$ with respect to $\mathcal{B}_{\text{src}}$, such that only pixels masked by $\mathcal{M}_{\text{src}}$ are considered. As in Chapter 3, the exponential distribution of pixel intensity errors is assumed, and, consequently, the mean absolute difference between masked pixels is used for $S(\cdot)$. More precisely, let the comparison of two buffers, $\mathcal{A}_{h\times w\times c}$ and $\mathcal{B}_{h\times w\times c}$, with respect to the buffer $\mathcal{A}$ which has an associated stencil mask $\mathcal{M}_{h\times w\times c}$, be defined as:

$$S(\{\mathcal{A}, \mathcal{M}\}, \mathcal{B}) = \frac{1}{|\mathcal{M}|} \sum_{x=1}^{w} \sum_{y=1}^{h} \sum_{i=1}^{c} |\mathcal{B}(y, x, i) - \mathcal{A}(y, x, i)| \mathcal{M}(y, x, i). \quad (4.18)$$

The multiplication by mask $\mathcal{M}$ in Eq. (4.18) serves to ensure that $S(\cdot)$ compares only the content (masked pixels) of $\mathcal{A}$. The result is normalised by scaling by the number of affected pixels and channels, $|\mathcal{M}|$.

Repeated optimisation of $C(\cdot)$ for one mesh at a time, and evolving the model $\mathcal{R}$ appropriately, optimises the groupwise alignment of the whole ensemble, which can be expressed as a *global* objective function

$$C_{\text{glob}} = \frac{1}{N} \sum_{i=1}^{N} C(\mathcal{R}, \mathrm{E}_i + \mathrm{D}_{i_k}). \quad (4.19)$$

4.4.3 Optimisation Regime

The first stage of the process is to compute the bending invariants (Elad and Kimmel [93]) of the meshes using MDS on the pairwise geodesic distances between all vertices in each mesh. The first two components of the bending invariants (or, equivalently, the result of embedding the vertices of the mesh with MDS in $\mathbb{R}^2$) form the initial maximally-isometric embeddings E_i, and the third component, call it *depth*, is kept in a matrix Z_i (Fig. 4.3). Note also that no initial centering or other alignment of the meshes in the 3D space is needed, as MDS operates only on the relative distances between vertices and so is insensitive to the absolute position and orientation of the meshes. In addition, MDS produces the embeddings centered around the origin.

Note that textures need not necessarily be RGB images, but can in general be any features (vector or scalar) associated with every point on the surface. As advocated in Cootes *et al.* [67] and as was done with images in Chapter 3, better performance can be achieved if local brightness normalisation is applied to images (textures) and the gradient information is also incorporated as image channels. This idea can be also applied here: assume henceforth that textures $\mathcal{T}_i$ are in this form (but see also Section 4.5.2).

The registration begins with a crude alignment of the embedded meshes to a template (say, the first mesh). Since MDS performs the embedding up to a similarity transform, including reflection, this needs to be accounted for. In practice, for the crude alignment brute force search is used to test for the eight possible reflection combinations (by 1 or -1 along each of the three dimensions of the bending invariant) and to approximately estimate rotation (trying all angles in increments of 10°).

The crude alignment is followed by a groupwise affine alignment stage. This is done in the same fashion as the non-rigid alignment, described below in Alg. 4.1, except that search is performed for the optimal affine transformation parameters for each embedded mesh, and instead of removing the embedding bias in line 18 the affine parameters are normalised so that the average translation and rotation across the ensemble is 0 and the average scaling is 1. Henceforth, assume that all E_i are affinely aligned.

The most important, non-rigid alignment stage, is addressed next. The idea from Sidorov *et al.* [243] and Chapter 3 that proved to work well is used in the proposed algorithm: accumulate the solution additively, gradually composing the resulting optimal embeddings over several iterations.

The non-rigid registration procedure is summarised in Alg. 4.1. The algorithm maintains the improvements to the initial embedding in a matrix D_{i_k} for each mesh. They are initialised to zero (line 1). This is the analogue of the deformation fields that in Chapter 3 were stored on a dense discrete grid ("deformation maps").

The iterative body of the algorithm (lines 3–20) is repeated until convergence. Each iteration begins by computing the current estimate of the texture model in the reference plane by rasterising and summing all embedded meshes using the current estimate of the optimal embedding $\mathrm{E}_i + \mathrm{D}_{i_{k-1}}$, lines (5–8). The embedding of each mesh in turn is then improved (lines 9–17). In

Algorithm 4.1 Perform non-rigid registration of an ensemble of textured meshes.

Require: Textured meshes $\mathfrak{M}_i = \{\mathrm{F}_i, \mathrm{V}_i, \mathrm{U}_i, \mathcal{T}_i\}$, their initial embeddings E_i (produced by MDS in advance and affinely aligned), depth components of the bending invariants Z_i, $i \in \{1 \dots N\}$. User-controllable parameters: termination conditions (line 2) and the schedule to decrease the sizes of the deformation discs (and increase their number, n_b), lines 14 and 19.

1: Initialise: $k \leftarrow 1$; $\mathrm{D}_{i_0} \leftarrow \mathbf{0}$, $\forall i$
2: **while** not happy **do**
3: Randomly permute the order of meshes.
4: $\mathcal{B}_{\text{sum}} \leftarrow \mathbf{0}_{h \times w \times 4}$; $\mathcal{M}_{\text{sum}} \leftarrow \mathbf{0}_{h \times w \times 4}$
5: **for** $i = 1$ to N **do**
6: $\{\mathcal{B}, \mathcal{M}\} \leftarrow \mathfrak{R}'(\mathrm{F}_i, \mathrm{E}_i + \mathrm{D}_{i_{k-1}}, \mathrm{U}_i, \mathrm{Z}_i, \mathcal{T}_i)$
7: $\mathcal{B}_{\text{sum}} \leftarrow \mathcal{B}_{\text{sum}} + \mathcal{B}$; $\mathcal{M}_{\text{sum}} \leftarrow \mathcal{M}_{\text{sum}} + \mathcal{M}$
8: **end for**
9: **for** $i = 1$ to N **do**
10: $\{\mathcal{B}_{\text{this}}, \mathcal{M}_{\text{this}}\} \leftarrow \mathfrak{R}'(\mathrm{F}_i, \mathrm{E}_i + \mathrm{D}_{i_{k-1}}, \mathrm{U}_i, \mathrm{Z}_i, \mathcal{T}_i)$
11: $\mathcal{M}_{\mathcal{R}} \leftarrow \mathcal{M}_{\text{sum}} - \mathcal{M}_{\text{this}}$
12: $\mathcal{R}_{i_k} \leftarrow (\mathcal{B}_{\text{sum}} - \mathcal{B}_{\text{this}})_\bullet / \max(1, \mathcal{M}_{\mathcal{R}})$
13: $\mathbf{b}_{n_b \times 1} \leftarrow \mathrm{FPS}(\mathfrak{G}(\mathfrak{M}_i, \cdot, \cdot), n_b)$
14: $\mathrm{Q} \leftarrow \mathrm{influence}(\mathfrak{M}_i, \mathbf{b})$
15: Using $C(\cdot)$ from Eq. (4.17) and with $\mathrm{H(P)} = \mathrm{E}_i + \mathrm{D}_{i_k} + \mathrm{PQ}$, optimise w.r.t. P to compute the optimal improvement $\Delta\mathrm{D}_{i_k} \leftarrow (\arg\min_{\mathrm{P}} C(\mathcal{R}_{i_k}, \mathrm{H(P)}))\,\mathrm{Q}$
16: $\mathrm{D}_{i_k} \leftarrow \mathrm{D}_{i_{k-1}} + \Delta\mathrm{D}_{i_k}$
17: **end for**
18: Remove embedding bias, see Section 4.4.4.
19: If improvement becomes slow, increase the number, n_b, and decrease the size of the deformation disks according to schedule.
20: $k \leftarrow k + 1$
21: **end while**
22: **return** $(\mathrm{E}_i + \mathrm{D}_{i_{k-1}})$ — the optimal embedding of V_i into $\mathbb{R}^2$, that brings all meshes, $\mathfrak{M}_i$, into alignment.

order to avoid a local minimum around the zero improvement hypothesis, the "current" sample is excluded from the model (lines 11–12), as suggested in Cootes *et al.* [66]. At each iteration a random deformation model is selected, comprising n_b deformation disks on the current 3D mesh. Using the FPS strategy n_b mesh vertices with indices $\mathbf{b}_{n_b \times 1}$ are randomly selected as the centres of the disks (line 13, see also Alg. 3.3).

This selection of the centres of the deformation disks is performed using adaptive FPS strategy, using mesh curvature to control the sampling density, analogously to using gradient information in Chapter 3. The reader is referred to Peyré and Cohen [207, 209] for details. Result of adaptive FPS sampling is illustrated in Fig. 4.14. As in Chapter 3, this allows to more frequently visit and improve the embedding of the "more interesting" parts of the surface which are assumed to be areas with higher curvature. It should be noted that adaptive FPS sampling based on texture gradient, as in Chapter 3, can also be used.

The radii of the deformation disks are chosen such that the adjacent disks overlap by one radius. FPS sampling ensures that the entire area of the mesh is covered evenly. The reason for choosing a random deformation model each is to allow the algorithm the progressively explore the space of all possible deformation models and to exclude to possibility of getting stuck with a poor choice of deformation model, as proposed in Sidorov *et al.* [243].

The influence matrix Q describing the effect of the disks on each vertex is then computed (line 14). In practice, to avoid geodesic computations in line 14, memory can be traded for speed. If memory permits, for a given radius r the influences of the deformation disks can be precomputed at each vertex as its potential centre and stored in a sparse matrix. Since the influence of each disk is bounded its influence on most vertices is zero and the above matrix is sparse. (An alternative but less memory efficient way is to precompute the pairwise distances between vertices in each original 3D mesh to avoid geodesic computations in the main loop).

Optimising over all possible parameters $\mathrm{P}_{2\times n_b}$, inducing a hypothetical embedding $\mathrm{H(P)} = \mathrm{E}_i + \mathrm{D}_{i_k} + \mathrm{PQ}$, the algorithm computes the optimal improvement $\Delta\mathrm{D}_{i_k}$ to the embedding (line 15). In experiments, the following optimisation scheme proved to work well. During the first few iterations, when the disks are large, optimise each disk, one at a time (a 2-dimensional optimisation problem). First, brute-force search (as in [67]) is performed, trying several displacements within a given evaluation budget and selecting the best one. Then the solution is refined with the Nelder-Mead method.

At later stages, when the deformation disks become small and the correspondences are already roughly established, the hypothesis is refined by optimising all disks at once using the stochastic optimiser, SPSA (see Spall [249], Maryak

and Chin [178]), as proposed in Sidorov *et al.* [243]. Recall from Chapter 3 that the advantage of SPSA is that its performance, in terms of the number of objective function evaluations, is relatively insensitive to the dimensionally of the search space. Optimisation with SPSA, is summarised in Alg. 4.2 as is similar to Alg. 3.2. The symbol $\mathfrak{S}$ in line 8 denotes the gradient of the local objective function (which takes a matrix P_m of parameters), estimated by sampling it at two points as before, reshaped as a matrix of the same size as P_m.

Finally, the computed improvement is learnt (line 16).

Parenthetically it is worth noting that when evaluating hypotheses during optimisation, the heuristic discussed in Section 3.4.3 can be also used, especially in the early stages when the disks are large. In such case the rasterisation operations can be sped up by only considering the triangles that are affected by the disks being optimised.

Line 18 serves to remove the embedding bias, the procedure discussed in Section 4.4.4. To save processing time, removal of the embedding bias can be performed less often than every iteration.

As the algorithm approaches the solution and the improvement slows down, the number of deformation disks is increased and their radii are accordingly decreased (line 19), to allow the algorithm to finesse the improvements with a progressively detailed deformation model.

The algorithm is stopped either when a maximum number of iterations has exceeded, or when relative improvement to the value of the overall cost function becomes less than a certain threshold. As the result, the algorithm returns the optimal embeddings for each mesh: $(\mathrm{E}_i + \mathrm{D}_{i_{k-1}})$. After the registration is complete, correspondences between any point on one mesh and any point on any other mesh are known via the common reference frame. So, for applications that require only the correspondences to be found nothing else needs to be done. To build an appearance model from the registered meshes are resampled at corresponding locations yielding a set of topologically consistent meshes and corresponding surfaces, see Section 4.4.5.

4.4.4 Removing Embedding Bias

It is possible that during the non-rigid registration stage the correspondences between the surfaces and the common reference plane may become systemati-

Algorithm 4.2 Improve embedding of a mesh

Require: $\mathfrak{M} = \{\mathrm{V}_{3\times n_v}, \mathrm{F}_{3\times n_f}, \mathrm{U}_{2\times n_v}, \mathcal{T}\}$ — mesh the embedding of which to improve, $\mathcal{R}$ — model, n_b — number of control points, $\mathrm{Q}_{n_b\times n_v}$ — influence matrix for control points, $\mathrm{E}_{2\times n_v}$ — initial embedding of the mesh, D — difference from initial embedding accumulated so far, n_{opt} — number of control points to optimise at once. User-controllable parameters α, γ, a_0, c_0, A, $m_{\max}$ are discussed on page 67.

$\mathbf{m}_{n_{\text{opt}}\times 1} \leftarrow$ randomly choose n_{opt} indices of control points to optimise.
$\mathrm{Q}'_{n_{\text{opt}}\times n_v} \leftarrow \mathrm{Q}(\mathbf{m}, :)$
$\mathrm{P}_0 \leftarrow \mathbf{0}_{2\times n_{\text{opt}}}$
$m \leftarrow 1$
while not converged and $m < m_{\max}$ **do**
  $a_m \leftarrow \frac{a_0}{(A+m)^\alpha}$ and $c_m \leftarrow \frac{c_0}{m^\gamma}$
  Generate $\Psi_m \in \mathbb{R}^{2\times n_{\text{opt}}}$, with $\Psi_m(i,j) \leftarrow$ Bernoulli(-1 or 1)
  Using $C(\mathrm{P}, \mathrm{E} + \mathrm{D} + \mathrm{P}_m\mathrm{Q}')$ from Eq. (4.17), estimate the gradient (reshaped as matrix Γ_m)
  $\Gamma_m \leftarrow \mathfrak{S}(C(\mathrm{P}_m, \cdot), \mathrm{P}_m, c_m, \Psi_m)$, by analogy with in Eq. (3.45)
  Update $\mathrm{P}_{m+1} \leftarrow \mathrm{P}_m - a_m\Gamma_m$, by analogy with Eq. (3.46)
  $m \leftarrow m + 1$
end while
return the optimal improvement parameters P_m.

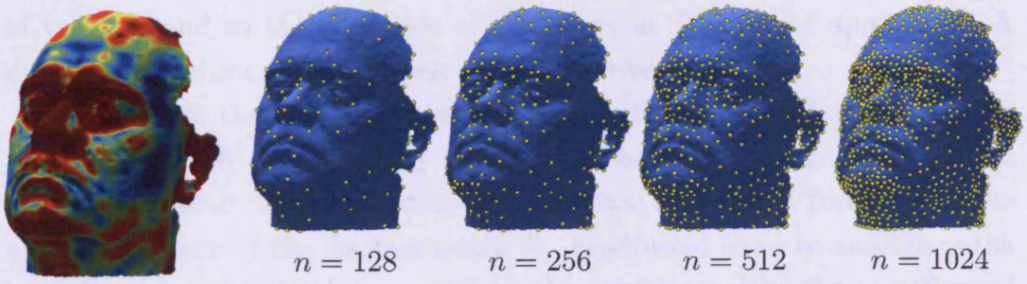


FIGURE 4.14: Adaptive, curvature based, farthest-point sampling on a mesh. *Left:* magnitude of the mesh curvature. *Right:* various number, n, of points seeded using FPS with the curvature magnitude as cost of movement in FM, as proposed in Peyré and Cohen [207, 209].

cally biased, which is equivalent to common reference space becoming distorted. If not handled correctly, this effect might become a runaway process and completely ruin the registration.

To alleviate this problem, Cootes *et al.* [67] apply GPA (see *e.g.* Dryden [89]) to align together all sets of control points, from each image, and then average them, thus computing a biasless reference coordinate frame. The approach of Cootes *et al.* [67] is feasible in their case since the number of control points

Algorithm 4.3 Remove embedding bias

Require: Ensemble of n meshes $\mathfrak{M}_i$ (each having n_{v_i} vertices), their initial embeddings E_i, affine transforms A_i, non-affine displacements D_i.

1: Initialise accumulators $\mathrm{S}_i \leftarrow \mathbf{0}_{2\times n_{v_i}}$, $\forall i = 1 \ldots n$
2: Initialise counters $\mathbf{c}_i \leftarrow \mathbf{0}_{n_{v_i}\times 1}$, $\forall i = 1 \ldots n$
3: **for** $i = 1$ to n **do**
4: $\quad \mathrm{X} \leftarrow \mathrm{affine}(\mathrm{A}_i, \mathrm{E}_i + \mathrm{D}_i)$
5: $\quad$ **for** $j = 1$ to n **do**
6: $\quad\quad \mathrm{Y} \leftarrow \mathrm{affine}(\mathrm{A}_j, \mathrm{E}_j + \mathrm{D}_j)$
7: $\quad\quad$ **for** all points $\mathbf{p}_k = \mathrm{X}(:, k)$ **do**
8: $\quad\quad\quad$ **if** $\mathbf{p}_k$ is on the mesh defined by $\{\mathrm{F}_j, \mathrm{Y}\}$ **then**
9: $\quad\quad\quad\quad \mathbf{c}_i(k) \leftarrow \mathbf{c}_i(k) + 1$
10: $\quad\quad\quad\quad \mathrm{S}_i(:, k) \leftarrow \mathrm{S}_i(:, k)$+(sample D_j on mesh $\{\mathrm{F}_j, \mathrm{Y}\}$ at point $\mathbf{p}_k$).
11: $\quad\quad\quad$ **end if**
12: $\quad\quad$ **end for**
13: $\quad$ **end for**
14: **end for**
15: **return** biasless $\tilde{\mathrm{D}}_i \leftarrow \mathrm{D}_i - \mathrm{S}_{i\bullet}/(\mathbf{c}_i \mathbf{1}_{1\times 2})^T$, $\forall i = 1 \ldots n$.

in each image is the same and so GPA and averaging are defined. In the framework described in this chapter, each mesh may contain a different number of vertices, and so the approach of Cootes *et al.* [67] is not applicable. A different procedure, discussed below, is used instead.

To preclude the above problem from happening, the embedding bias is removed (line 18) by adjusting the improvements D_i, so as to annihilate the bias. Instead of manipulating the reference coordinate frame, the non-affine component of the displacements, D_i, is adjusted so as to annihilate the bias. First, a point cloud $\mathrm{A}_{3\times n_a}$ is formed by concatenating the transformed embedded vertices $(\mathrm{E}_i + \mathrm{D}_{i_k}), \forall i$ from all meshes. Each embedded mesh $\mathfrak{M}_i$ in turn is sampled to determine the non-affine displacements at points A due to $\mathfrak{M}_i$ (for points in A that lie on the mesh). The contribution from all meshes is then averaged and subtracted from the displacements D_{i_k}. Since the transforms are invertible, the above procedure ensures that for any point in the reference coordinate frame, its inverse non-affine transform, averaged across all meshes, remains zero. The entire procedure is summarised in Alg. 4.3. To save on computation time, bias removal need not necessarily be performed after each iteration, but every few iterations instead.

This procedure is analogous to removing the deformation bias in Chapter 3 (see also Fig. 3.10).

4.4.5 Resampling

After the registration is complete, correspondences between any point on one mesh and any point on any other mesh are known. So, for applications that require only the correspondences to be found nothing else needs to be done.

However, in case building of a statistical model of shape or appearance, or both, is required, it is necessary to produce a set of meshes, approximating the original meshes, with the same number of vertices (at corresponding locations) and the same topology, in order that statistics on corresponding vertices can be computed.

There are two ways of doing it. One way is to select one mesh as a reference, and repeatedly warp it to conform to all other meshes in the ensemble, using the computed correspondences. The second, more flexible, approach is to *resample* the registered meshes and is discussed below.

Recall that after the registration is complete, the result is a set of embeddings of meshes into the common reference plane with associated embedding improvements D_{i_k} that bring all the meshes into alignment. For any point on a mesh embedded into the common reference plane it is therefore trivial to recover the corresponding position in the original 3D space, because the original 3D coordinates of all vertices are known (stored away before flattening), and so for any point on an embedded mesh the corresponding 3D coordinates can be found by barycentric interpolation between the 3D coordinates of the encompassing triangle.

The resampling is again done using the FPS strategy to select the points in the 2D plane which to map back to 3D. To improve the quality of the resulting meshes, the seeding is again done adaptively — this time using the gradient of the depth image as a movement cost. This serves to create more vertices in areas that steeply go "into the plane", and vice-versa.

When resampling the stack of aligned meshes, it is likely that due to imperfections in the original data (holes) there will be some points in the common reference plane for which the 3D coordinates and pixel colours are not known for all meshes. If the original data is very poor, it might even be

that at *most* points the picture is incomplete. Two strategies can be used here. The first is to use the fact that the meshes are now aligned and, therefore, statistical methods can be used to fill in the missing information. For instance, a hole in a particular mesh can be patched with the weighted average taken from the other meshes. This works well for pixel colours (textures), but doesn't always work for the 3D coordinates. The second strategy is to use the Poisson interpolation, described in Section 4.2.6.

4.5 Performance and Space Complexity Considerations

4.5.1 Mesh Decimation

It is now worth addressing the problem of time and space complexity when dealing with sizable meshes. Consider the problem of applying the MDS algorithm, as discussed in Section 4.2.3, to the matrix of pairwise geodesic distances between the vertices of a mesh. The time complexity of MDS is $O(n^3)$, where n is the number of vertices in the mesh (and, therefore, the number of rows in the square distance matrix). This is due to the fact that MDS requires solving an eigenvalue problem of complexity $O(n^3)$, see Кублановская (Kublanovskaya) [305], and this alone makes operating on the raw high-resolution meshes undesirable.

On top of that, the space complexity of MDS is at least $O(n^2)$, because one needs to store the n-by-n distance matrix. This makes the maximum useful size of the mesh bounded by the amount of available memory. The meshes that come from scanners can easily have up to $N_v \approx 20000$ vertices. The storage requirement for the distance matrix alone is then at least $n_v^2 \times 8$ bytes ≈ 2.98 GB, which usually exceeds the available contiguous memory size on desktop PCs, not to mention the additional storage required for the actual computations.[3]

Efficiency can be achieved if the dense meshes are decimated first. In Zigelman *et al.* [302], for example, MDS is performed on a subset of vertices, but using the full model for geodesic computations, and the result is interpolated to the remaining vertices.

[3]The most widely used double precision floating point representation of numbers, the IEEE 754-2008 Standard for Floating-Point Arithmetic, requires 64 bits = 8 bytes per number.

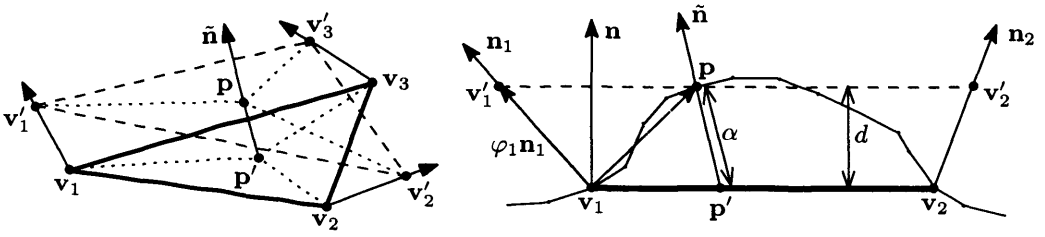


FIGURE 4.15: Projection of the dense mesh (violet line) onto a coarse mesh.

In Sidorov *et al.* [242] a way to operate on coarse decimated meshes without significantly sacrificing the final result was shown. (In Sidorov *et al.* [242] a special kind of combined appearance model is discussed, in which the shape of the coarse mesh, the texture, and the difference between the coarse and the dense mesh were modelled). The idea is simple: the decimated mesh is chosen to consist of a subset of vertices of the original dense mesh, appropriately triangulated, and additionally the *difference* between the coarse and the dense meshes (see below) is computed and stored. The necessary operations are then performed on the coarse mesh, significantly reducing storage requirements and running time. Finally, using the previously computed difference, the results are interpolated to the missing points to give the final high-resolution mesh.

The projection of the dense mesh onto the coarse mesh is a simple geometric operation which is now derived. Assume it has been established that a point $\mathbf{p}$, belonging to the dense mesh (drawn in violet in Fig. 4.15), projects onto a triangle $T = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ of the coarse mesh. Denote by $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$ the vertex normals, usually computed as the weighted sum of the normals to the faces that share the vertex, at vertices $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$. The interpolated normal, $\tilde{\mathbf{n}}$, at point $\mathbf{p}'$ inside the triangle is defined as $\tilde{\mathbf{n}} = (\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3)\mathfrak{B}_{\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3}(\mathbf{p}')$. Projection of the point $\mathbf{p}$ along the interpolated normal (light blue lines in Fig. 4.15 on the right) then involves simply finding the corresponding point $\mathbf{p}'$ on the triangle T such that the vector $(\mathbf{p} - \mathbf{p}')$ is parallel to $\tilde{\mathbf{n}}$. Denote by $T' = \{\mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_3\}$ a triangle whose vertices are the intersection points of the plane parallel to T and passing through $\mathbf{p}$, and the rays from $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ in the direction of normals $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$, shown in red dashed lines in Fig. 4.15. The "side view" of the situation is shown in Fig. 4.15 on the right. Since $(\mathbf{v}'_i - \mathbf{v}_i)$ is parallel to $\mathbf{n}_i$, one can write $\mathbf{v}'_i$ parametrically: $\mathbf{v}'_i = \mathbf{v}_i + \varphi_i\mathbf{n}_i$. If $\mathbf{n}$ is a unit normal

to T, then $0 = \mathbf{n} \cdot (\mathbf{p} - \mathbf{v}'_i) = \mathbf{n} \cdot (\mathbf{p} - \varphi_i \mathbf{n}_i - \mathbf{v}_i)$, because $T \parallel T'$. Therefore, $\varphi_i = (\mathbf{n} \cdot (\mathbf{p} - \mathbf{v}_i)) / (\mathbf{n} \cdot \mathbf{n}_i)$ and finally

$$\mathbf{v}'_i = \mathbf{v}_i + \mathbf{n}_i \frac{\mathbf{n} \cdot (\mathbf{p} - \mathbf{v}_i)}{\mathbf{n} \cdot \mathbf{n}_i}. \tag{4.20}$$

Having found the vertices $\mathbf{v}'_i$ of T', the barycentric coordinates of $\mathbf{p}$ in T' can be found used to find the position of the projected point $\mathbf{p}'$:

$$\mathbf{p}' = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{pmatrix} \mathfrak{B}_{\mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_3}(\mathbf{p}). \tag{4.21}$$

To determine, for each point on the dense mesh, onto which of the triangles T_i of the coarse mesh it should be projected, the vertices of T'_i are computed for each T_i and the barycentric point-in-triangle test is applied, see Section 3.4.1.

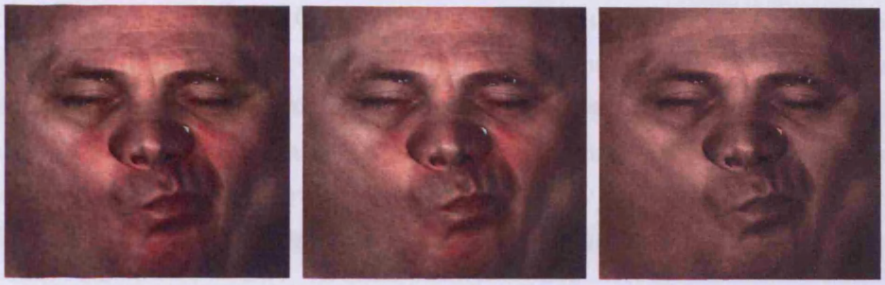
The signed displacements, α, along the interpolated normals, are stored away for all the projected points. After performing some operations on the coarse mesh, the projected points of the dense mesh can be reconstructed by elevating them by α along the corresponding interpolated normals, as summarised in Sidorov *et al.* [242].

The method of Peyré and Cohen [207], which involves selecting the subset of vertices, of the dense mesh, using the FPS strategy can be used. The resulting subset of vertices is then triangulated: geodesic Voronoi tessellation is computed and so is its dual — the geodesic Delaunay triangulation, as detailed in Section 4.2.5.

There are, of course, many other ways of decimating meshes, *e.g.* QSLIM [2], Garland [103] *etc.* The reader is referred to a survey by Talton [260]. The FPS based approach was chosen for two reasons. The first reason is that it can be used for adaptive decimation Peyré and Cohen [207] in a way similar to the way the control points in Section 3.5.7 are seeded, but using curvature information instead. The second reason is that FPS is used elsewhere in the algorithm, and so the implementation and the explanation become easier.

4.5.2 Compressing Texture Data

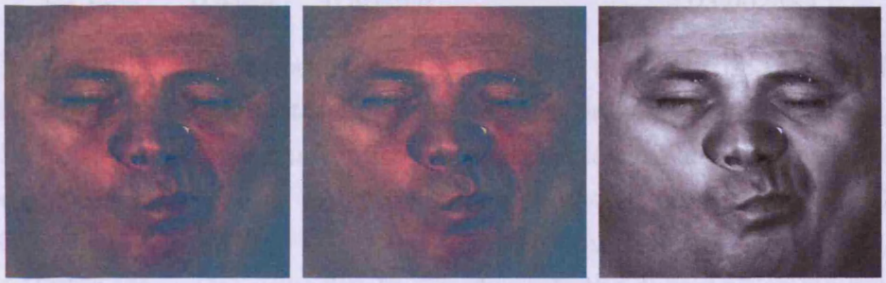
Another trick to reduce storage requirements is based on the observation that in face images the colour variation does not usually span the entire RGB space but is typically constrained to a limited gamut ("skin tones"). This suggests



(a) Original texture.

(b) Texture recovered from 2-channel colour space.

(c) Texture recovered from 1-channel colour space.



(d) Projection into the new colour space (all 3 channels).

(e) Projection into the reduced 2-channel colour space.

(f) Projection into the reduced 1-channel colour space.

FIGURE 4.16: Colour space compression.

that the texture colour information can be compressed into a smaller number of channels without any noticeable degradation of registration quality.

One approach to colour space compression, proposed by Cosker in [68] is to simply construct a look up table of colours using the luminance as a key, reducing three channels to just one. This amounts to projecting colour values from $\mathbb{R}^3$ to a $\mathbb{R}^1$ subspace using a predefined projection operator that is independent of the data.

In the experiments of Section 4.6 a more general procedure in was used. The projection operator from $\mathbb{R}^3$ (RGB) to a lower dimensional colour space is computed from a sample of pixel colours using PCA. This ensures that the projection retains maximum variance.

The procedure is straightforward. First, RGB colours of pixels from all images are sampled and concatenated into an observation matrix $O_{colours}$. (To avoid wasting memory, 2000 randomly selected pixels are sampled from each

image.) Applying PCA to $O_{colours}$ yields the basis vectors for the new colour space. Discarding one or two least significant ones yields a data-dependent projection operator. Retaining just one basis vector would amount to a conversion to grayscale that retains maximum variance. In the experiments of Section 4.6 two channels were retained. For the particular gamut of texture in Fig. 4.16 the eigenvectors E and the corresponding eigenvalues $\boldsymbol{\lambda}$ are

$$\mathrm{E} = \begin{pmatrix} 0.8091 & -0.5877 & 0.0036 \\ 0.4771 & 0.6532 & -0.5879 \\ 0.3432 & 0.4774 & 0.8089 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\lambda} = \begin{pmatrix} 0.053365 \\ 0.001659 \\ 0.000238 \end{pmatrix}. \tag{4.22}$$

The result is illustrated in Fig. 4.16. The example texture is shown in Fig. 4.16a. The projection of pixel colours onto the new basis with all the three, then two, and finally just one basis vector are shown (appropriately scaled for display) in Fig. 4.16d (with the projected components displayed as RGB), Fig. 4.16e (same as above, with the blue channel set to zero), and Fig. 4.16f (as grayscale) respectively. To illustrate the degree of lossiness due to colour space compression, also shown are the images obtained by unprojection of the pixel colours back into the original RGB space: from the 2-channel colour space (Fig. 4.16b) and from the 1-channel colour space (Fig. 4.16c).

Note that more than 96.5% of the eigenenrgy corresponds to just the first eigenvector. This result is typical for skin textures. It should be noted that no significant loss occurs when compressing the colours to 2-channels (compare Fig. 4.16b with Fig. 4.16a) because the third eigenvalue $\boldsymbol{\lambda}(3)$ is so small.

4.5.3 Distance Computations on Meshes

Despite fast marching on meshes being relatively cheap, for very large meshes it might still be too slow. If the mesh in question is very dense (compared to the typical density of control points), geodesic path computations can be approximated with the computations of shortest paths on a graph, whose vertices are vertices of the mesh and edges are the edges of the polygons of the mesh, which amounts to restricting the valid paths on a mesh to be along the edges only. This allows one to use classical shortest path algorithms on graphs, *e.g.* Dijkstra's algorithm [87], that are faster than geodesic computations

as well as more memory efficient. Such approximation to the true geodesic distances was employed by Praun *et al.* [217,218]. In addition, there has been research into efficiently solving shortest path problems on graphs using the highly parallel architecture of modern GPUs, see *e.g.* [139].

The approximation of geodesic paths with paths on the mesh graph is very crude and might lead to incorrect results, because the true geodesic paths can cut across faces and so cannot be found by Dijkstra's algorithm.

This approximation was not used in the experiments of this chapter. However, it might be useful as a last resort in problems of very large magnitude.

4.6 Experiments

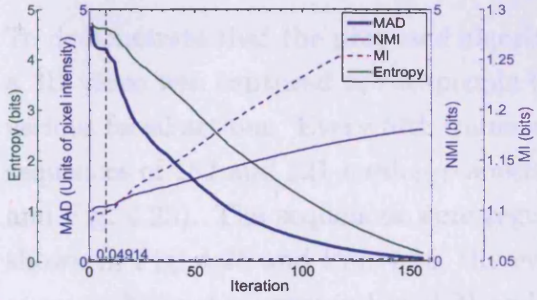
In order to validate the proposed approach, several registration experiments were conducted with artificial and real 3D data, including inter-subject registration. For all experiments the values of a various alignment quality measure at each iteration as the algorithm progresses, were plotted to monitor improvement. These are the values of the C_{glob} from Eq. (4.19) (MAD), mean average mutual information and normalised mutual information between the texture model and each shape normalised sample (MI and NMI), and average pixel stack entropy across the shape normalised ensemble (by analogy with Section 3.6). To visually inspect the registration progress, the evolution of the model of texture and average shape is also shown: as the algorithm establishes the correspondences more and more accurately these converge to a true crisp representation of the underlying structures.

4.6.1 Comparison with the ground truth

For this experiment, one mesh was selected as a template and randomly deformed by selecting 32 control points on it, displacing each control point randomly by ± 24 mm (uniformly distributed) and interpolating the deformation with thin-plate splines. The obtained 64 synthetic meshes (examples shown in Fig. 4.17), with the ground truth correspondences known, were then registered. Figure 4.18 shows the evolution of the average shape and texture as the registration progressed, and Fig. 4.19 show the progress plots. In order to evaluate the accuracy of the registration, two measures were computed.

FIGURE 4.17: Example meshes from the artificial data set.

FIGURE 4.18: Evolution of the mean surface and texture for the artificial data set.



Measure	Initial	Final	Impr.
MAD	4.6706	0.0491	98.95 %
MI	1.0907	1.2875	18.04 %
NMI	1.0047	2.6123	160.00 %
Entropy	4.7295	0.1537	96.75 %

FIGURE 4.19: Registration quality measures (ground truth experiment).

The average pairwise distance between corresponding vertices in the aligned meshes was 0.838 mm (median 0.633 mm, $\sigma = 0.773$ mm). The algorithm was stopped after 160 iterations (the results would be improved even further if the algorithm was run for longer), see the progress Fig. 4.19.

The final spatial errors between every shape-normalised mesh and the template warped to the mean of the shape-normalised meshes were also measured. The average pairwise distance between corresponding vertices in the aligned meshes was 0.570 mm (median 0.408 mm, $\sigma = 0.570$ mm).

These results show that the proposed method performed well and converged to within the expected accuracy (subject to the finite number of iterations, flat areas in the texture, and small imperfections due to texture warping).

FIGURE 4.20: Example meshes from the PERSON1 data set.

FIGURE 4.21: Evolution of the mean surface and texture for the PERSON1 data set.

4.6.2 Within-subject registration

To demonstrate that the proposed algorithm can register a sequence of meshes, a 3D video was captured of two people (PERSON1 and PERSON2) performing various facial actions. Every fifth frame was taken from each video yielding two sequences of 182 and 221 meshes respectively (examples are shown in Fig. 4.20 and Fig. 4.23). The sequences were registered. The progress of registration is shown in Fig. 4.25 and Fig. 4.26, the evolution of the texture model and the average shape are shown in Fig. 4.21 and Fig. 4.24: observe the crisp texture in the final stage of alignment. Having registered the sequences, a 3D appearance model for each person was built. The first three modes of variation are shown in Fig. 4.22 and 4.28. The results are excellent, demonstrating the usefulness of the proposed algorithm for automatic 3D appearance model building.

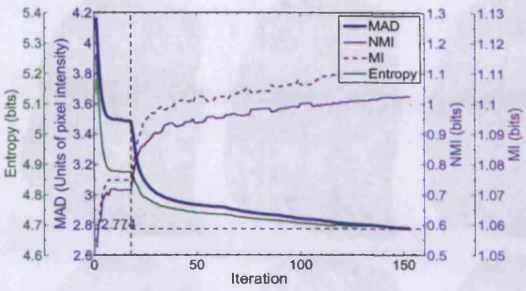
4.6.3 Inter-subject registration

To demonstrate that the proposed algorithm can easily handle inter-subject registration, a corpus of facial scans of 32 different individuals, 11 of which are women, was captured. Some examples from this data set are shown in Fig. 4.27. Note the degree of variation, both in shape and texture (*e.g.* facial hair). The algorithm successfully registered this data set. The progress plot is shown in Fig. 4.32, and the evolution of the texture model and average shape in Fig. 4.30 and Fig. 4.31. The appearance model from registered samples

FIGURE 4.22: The first three modes of variation ($\pm 3\sigma$) of the 3D AAM built from the registered PERSON1 data set.

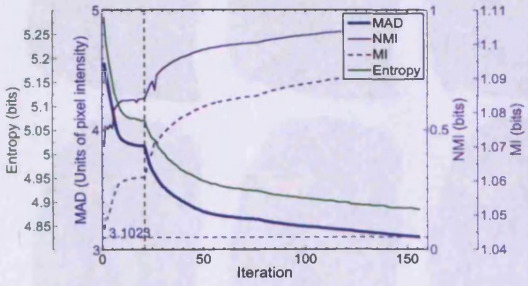
FIGURE 4.23: Example meshes from the PERSON2 data set.

FIGURE 4.24: Evolution of the mean surface and texture for the PERSON2 data set.



Measure	Initial	Final	Impr.
MAD	4.1669	2.7740	33.43 %
MI	1.0547	1.1123	5.46 %
NMI	0.5266	1.0241	94.49 %
Entropy	5.2624	4.6896	10.88 %

FIGURE 4.25: Registration quality measures (PERSON1 data set).



Measure	Initial	Final	Impr.
MAD	4.7667	3.4423	27.78 %
MI	1.0443	1.0898	4.36 %
NMI	0.4142	0.9043	118.29 %
Entropy	5.3093	4.9442	6.88 %

FIGURE 4.26: Registration quality measures (PERSON2 data set).

FIGURE 4.27: Example meshes from the inter-subject data set.

were also built. The first three modes of variation are shown in Fig. 4.29. Inter-personal registration is a notoriously challenging problem, with which the proposed algorithm admirably copes.

FIGURE 4.28: The first three modes of variation ($\pm 3\sigma$) of the 3D AAM built from the registered PERSON2 data set.

4.7 Future work

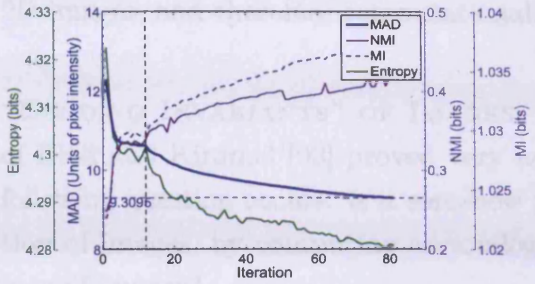
Below, some possible improvements to various parts of the proposed method and some possible directions of future research are outlined.

MORE EXPLICIT USE OF SHAPE INFORMATION. Note that in the proposed algorithm only the depth of the bending invariant was used in addition to pixel colours. This depth is a very crude feature and in case of face data is a low frequency signal. However, without any changes to the algorithm, other shape-based features can be added as additional channels. One example would be the gradient of the depth component of the bending invariants. Some of the other features (essentially scalars or vectors associated with every point on the surface) that can be used, include: spin images (Johnson and Hebert [134]), curvature (Gal and Cohen-Or [101]), moments and spherical harmonics (Sharp *et al.* [240]), integral descriptors (Gelfand *et al.* [104]), Fast

FIGURE 4.29: The first three modes of variation ($\pm 3\sigma$) of the 3D AAM built from the registered inter-subject data set.

FIGURE 4.30: Evolution of the texture model in the flat parametric space for the inter-subject data set.

FIGURE 4.31: Evolution of the mean surface and texture for the inter-subject data set.



Measure	Initial	Final	Impr.
MAD	12.3036	9.3095	24.33 %
MI	1.0226	1.0368	1.39 %
NMI	0.2483	0.4186	68.59 %
Entropy	4.3225	4.2809	0.96 %

FIGURE 4.32: Registration quality measures (inter-subject data set).

Fourier Transform (FFT) and Discrete Cosine Transform (DCT) coefficients (Li and Guskov [165]), cluster signatures (Huang and Pottmann [130]) and others. These features are discussed in great detail in the review paper of Tam *et al.* [261].

STRESS TERM IN THE COST FUNCTION. It is also possible to augment the cost function with an additional term that measures the quality of embedding. The *stress* the embedded point cloud, as used in MDS, can serve for this purpose. In the proposed method, such a term was not used for two reasons. First, adding an extra term that is not commensurate with the texture-based term would require a scaling coefficient that is hard to determine automatically, and so the algorithm would gain one extra parameter which requires tuning. Second, as the experiments demonstrate, in the case of 3D face surfaces and similar data, there is usually enough clues in texture for registration to succeed without the additional stress term. It is possible that the inclusion of the stress term could be useful if the registration was guided by the above discussed shape features alone, without the textures.

OTHER KINDS OF SURFACES. It is possible to modify the proposed algorithm in order to register surfaces other than those homeomorphic to a disk. A related work here is the groupwise approach applied to the registration of 3D data (in this case, closed genus-0 surfaces) via parameterisation by Davies *et al.* [77, 81]. In Davies *et al.* [77], the surfaces are mapped to a unit sphere and Davies *et al.* [81] discuss a computationally cheaper option of first mapping a closed surface to a unit sphere, then to an octahedron which is then cut and unfolded to a square. This mapping of a closed surface to a square makes the manipulation of deformation fields analogous to the case of

2D images, and therefore computationally cheap.

"Bending Invariants" of Images. Seeing how the bending invariants of Elad and Kimmel [93] proved very useful for registration of surfaces, the following question occurs: is it somehow possible to apply this idea to registration of images, by computing an analogue of the bending invariants for the case of images?

One possible way of doing so is evoked by the stereo vision literature. Remarkably, the 11-th ranking algorithm in the Middlebury Stereo Evaluation rank [1] is an algorithm based on local window matching. It was proposed by Hosni *et al.* [129]. (In contrast to all other state of the art stereo disparity algorithms which these days typically rely on some global optimisation framework.) The clever idea of Hosni *et al.* [129] is to compute the support weights, for a square window being matched, using geodesic distances between the centre of the window and all other pixels in it. The geodesic distance between two adjacent pixels ("cost of movement" from one pixel to the next) depends on the difference in their colours. To illustrate: moving across a strong edge then results in higher geodesic distance and lower support weight for pixels demarcated by that edge, which, in turn, results in the block matching algorithm treating the regions demarcated by the edge as potentially different objects having potentially different disparities.

Combining this idea with the concept of bending invariants, it is conceivable that "bending invariants" of images can be computed, using the geodesic distances understood in the above sense. It is possible that such image "bending invariants" can be used for classification and recognition straight away (as ordinary surface bending invariants were used by Bronstein *et al.* [44,45]) and possibly could be used to initialise image registration algorithms.

Registration Without Embedding. It is conceivable that all operations could be performed directly in 3D, without resorting to flattening first. Bronstein *et al.* [47–49] considered the problem of reducing the distortion of distances introduced by embedding of a manifold into $\mathbb{R}^n$, in order to improve the performance of their face recognition system. In doing so, they developed a method, called Generalised Multidimensional Scaling (GMDS), which allows for embedding of a manifold to be performed into another manifold.

On the other hand, in Section 3.7 a possibility of performing groupwise registration without creating an aggregate model, by only ever comparing the original samples between themselves, was discussed.

Fusing these two ideas, it is possible to contemplate a groupwise textured surface registration algorithm that would not need to resort to embedding of the surfaces into $\mathbb{R}^2$ first. Instead, using the scheme from Section 3.7 the hypothetical algorithm would perform comparisons between (deformed) original surfaces by embedding one into another with GMDS. This, it can be speculated, may improve the performance of registration.

4.8 Conclusion

A novel, efficient and reliable, fully automatic method for performing groupwise non-rigid registration of textured surfaces was presented. Using a novel combination of ideas from geodesic mesh processing and traditional registration methods, it was shown how to reliably, in a principled manner, solve the problem of registering 3D surfaces in a fashion analogous to the previously solved (Chapter 3, Sidorov *et al.* [243]) problem of 2D image registration. The resulting algorithm is computationally efficient, reliable, fully automatic, and is, additionally, readily amenable to a GPU implementation. Its usefulness in accurately establishing correspondences between textured meshes and, especially, in building high quality 3D appearance models was experimentally demonstrated. The proposed method copes with data exhibiting significant variation in shape and texture, such as in the case of notoriously difficult inter-subject registration, with which the proposed algorithm copes admirably.

CHAPTER

5 Statistically Constrained Real-time Meshless Simulation

> Ideas become a force when they control the masses.
>
> Vladimir Lenin

In this chapter, the topics discussed earlier are combined, and a novel application of automatic craniofacial appearance modelling — statistically driven simulation — is proposed.

Inspired by statistical modelling, and leveraging the surface registration methods for automatic construction of models, this chapter improves upon the ideas from recent works on meshless geometrically based quasi-mechanical simulation methods. A new real-time approach is proposed to simulate deformable objects, using a learnt statistical model to achieve a higher degree of realism while retaining the advantages of geometrically based meshless simulation methods. The improvement in realism over the state of the art approaches is attained by capturing important nuances of an object's kinematics, and additionally its dynamic texture variation, in a statistical appearance model, and using it to drive the simulation.

In the previous chapters, all the required components of the automated pipeline, from data acquisition to modelling, were presented and will now be supplemented with fast quasi-mechanical simulation, as a natural extension.

In Section 5.5, examples of non-trivial biomechanical objects simulated on a desktop machine in real-time are presented, demonstrating superior

realism of the proposed method over current geometrically motivated simulation techniques.

5.1 Background

The earliest mention of applying mechanical principles in computer graphics is found in the discussion by Lasseter [154], who summarised the principles which were well known in the field of hand drawn animation ("cartoon physics"), and suggested that they may be used to enhance the realism of animated 3D computer graphics.

The history of numerical mechanical simulation for computer graphics begins around the same time with the pioneering work on elastic models by Terzopoulos *et al.* [265] and continues to this day. In the past two decades the field of mechanical simulation and, in general, physically motivated modelling for computer graphics has made a remarkable progress, with breakthroughs in fundamental numerical methods (solution of partial differential equations (PDEs), numerical integration, modal analysis, fast real-time approximation algorithms *etc.*), as well as modelling of various phenomena and object characteristics (fracture, plasticity, in non-Newtonian mechanics, modelling of gases, liquids, thin shells, cloth and hair in addition to solids).

In this field, some of the landmarks that applicable to computer graphics are briefly given below. Baraff and Witkin [17] address the problem of large time steps in cloth simulation via implicit integration. Desbrun *et al.* [86], using a clever approximation to implicit integration, developed a stable and efficient algorithm for simulating mass-spring systems. Pentland and Williams [203] describe an approach to couple the model of mechanics based on modes of vibration with a volumetric geometrical model, trading accuracy for efficiency. James and Pai [132] use Boundary Element Method (BEM) to simulate linear elastic objects at interactive rate, including elastic interactions between objects. Efficiency is achieved in their method through precomputation of state space dynamics and impulse response functions. A number of works focuses on simulation of particle systems. Desbrun and Gascuel [84], for example, represent objects as clouds of massive particles smeared in space and use explicit "leap-frog" integration to update particle positions given forces. Self-organising particle systems for fluid objects are proposed by Tonnesen in [272]. Müller *et al.*

[188] use particle system to simulate plasticity and melting. The traditional FE and finite volume (FV) methods from computational mechanics have also been widely employed. Müller and Gross [186] simulate elasto-plastic deformations and fractures in real time. Debunne *et al.* [83] proposed an approach which by adaptively changing the resolution of the FE model is capable of animating visco-elastic deformable objects with a guaranteed frame rate. Teran *et al.* [262] use the FV method to simulate contracting muscle tissue using a quasi-incompressible, transversely isotropic, hyperelastic constitutive model.

Even a brief summary of this vast field is difficult here. For a comprehensive review of key developments see Gibson and Mirtich [106] followed by a more recent review of Nealen *et al.* [190], and also references within Müller *et al.* [187].

To summarise, the main focus of the traditional deformable object modelling approaches has always been on increasing the fidelity and accuracy of modelling properties of materials, increasing the range of behaviours that can be simulated (realistic collision detection and response, modelling of fractures, melting, plastic deformations *etc.*), and improving the stability of numerical simulation methods. In other words, physical realism and fidelity have been receiving more emphasis than speed.

However, the interactive applications of mechanical modelling (such as computer games or virtual surgery simulators) have been neglected until recently. This is evident from the observation of mechanical modelling in computer games — the industry which has been a dominant stimulus for the development of computer graphics in the past two decades. While some recent games feature plausible simulation of cloth, vegetation *etc.*, in general the game physics is still dominated by rigid, possibly articulated, objects.

Müller *et al.* [187] discuss the reasons why more complex mechanical phenomena cannot yet be modelled in interactive scenarios. One reason is the hard performance constraint imposed by interactive scenarios: only a fraction of computational resources can be dedicated to the mechanical simulation, in other words the performance of the simulator has to be *faster* than real-time. The other, more important, reason is that interactive scenarios require numerical stability under all circumstances. While there are ways to ensure the stability of simulation by using implicit integration schemes (*e.g.* Irving *et al.* [131]), for objects with complex mechanical properties and non-trivial complexity

they are prohibitively computationally expensive. Additionally, volumetric representations of objects, required for FE modelling, are hard to produce and are rarely, if ever, used in the computer games industry.

Recently, approaches have been proposed (Müller *et al.* [187, 188], Guo and Qin [120]) which attack the problem on two fronts: by employing a point-based, or meshless, representation of bodies, and by replacing a physically based simulation paradigm with a geometrically motivated one. These approaches have been drawing more and more attention as a possible solution to the faster than real-time interactive simulation of deformable objects. In this sphere, a minor degradation in physical realism is a small price to pay for computational efficiency as long as visual realism is preserved. Efficiencies may be achieved by abandoning the physical model (*e.g.* elastic energies and forces) and replacing it with a geometrically based model.

Point-based, or meshless, representations of surfaces and solids have some distinct advantages in interactive applications. First, point-based representations are much easier to obtain than volumetric ones since commercially available 3D scanners only sample the geometry of the surface (and capture surface texture). Second, data from 3D scanners typically comes as a point cloud, and so no preprocessing, such as meshing the surfaces or domain meshing, is required. Third, meshless methods offer higher spatial adaptivity: node insertion and deletion, modelling of fractures *etc.* do not require remeshing. Finally, meshless methods require minimal storage and data manipulation overhead: complex, memory intensive data structures are not required during the simulation.

The work that is most relevant in the context of this chapter is that by Müller *et al.* [187]. To achieve efficiency, they reject the idea of implicit integration (which is stable, but requires a solution to a large system of equations at each step, making it prohibitive in real-time scenarios), and use explicit integration. To guarantee stability of explicit integration, the core idea of the approach due to Müller *et al.* [187] is to replace physically based simulation with a geometrically based one: their method relies on a generalised shape matching (Kent *et al.* [143]) between an undeformed, or rest, state and a deformed state of a point cloud. In other words, the main idea of Müller *et al.* [187] is to replace energies with geometric constraints and forces

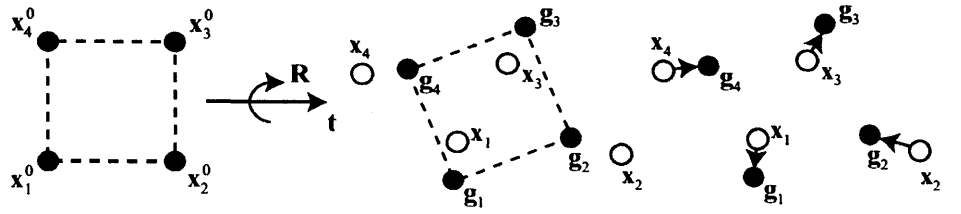


FIGURE 5.1: Illustration of the shape matching process in Müller *et al.* [187].

by distances of current positions to goal positions which are determined via a shape matching algorithm.

The shape matching process of Müller *et al.* [187] can be summarised as follows. Let $\mathbf{x}_i^0$ be the initial, and $\mathbf{x}_i$ the target positions (displaced due to external forces, say) of particles with masses m_i. Let $\mathbf{x}_{\text{cm}}^0$ and $\mathbf{x}_{\text{cm}}$ be the mass centres of the point clouds $\mathbf{x}_i^0$ and $\mathbf{x}_i$ respectively. A rigid transformation $\{\mathrm{R}, \mathbf{t}\}$ is computed that best matches the cloud $\mathbf{x}_i^0$ to $\mathbf{x}_i$ in the least squares sense:

$$\sum_i m_i(\mathrm{R}(\mathbf{x}_i^0 - \mathbf{x}_{\text{cm}}^0) + \mathbf{t} - \mathbf{x}_i)^2 \rightarrow \min_{\mathrm{R},\mathbf{t}}. \tag{5.1}$$

The optimal translation $\mathbf{t}$ is simply $\mathbf{x}_{\text{cm}}$, and the optimal rotation R is found via polar decomposition (see Lorusso *et al.* [168]):

$$\mathrm{R} = \mathrm{A}_{pq}\sqrt{\mathrm{A}_{pq}^T\mathrm{A}_{pq}}^{-1}, \text{ where } \mathrm{A}_{pq} = \sum_i m_i(\mathbf{x}_i - \mathbf{x}_{\text{cm}})(\mathbf{x}_i^0 - \mathbf{x}_{\text{cm}}^0)^T, \tag{5.2}$$

see Müller *et al.* [187] for full derivation. The *goal* positions of points can then be expressed as $\mathbf{g}_i = \mathrm{R}(\mathbf{x}_i^0 - \mathbf{x}_{\text{cm}}^0) + \mathbf{x}_{\text{cm}}$. The matching process is illustrated in Fig. 5.1. The modified Euler integration step in Müller *et al.* [187] uses these goal positions to avoid overshooting:

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \alpha\frac{\mathbf{g}_i(t) - \mathbf{x}_i(t)}{\Delta t} + \Delta t f_{\text{ext}}(t)/m_i, \tag{5.3}$$

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t\mathbf{v}_i(t + \Delta t), \tag{5.4}$$

where α is a parameter controlling stiffness. Müller *et al.* [187] derive that for $0 \leq \alpha \leq 1$ this integration step never overshoots and the system always remains stable. Compare this with ordinary Euler integration:

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t\mathbf{v}_i(t), \tag{5.5}$$

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \Delta t f_{\text{ext}}(t)/m_i. \tag{5.6}$$

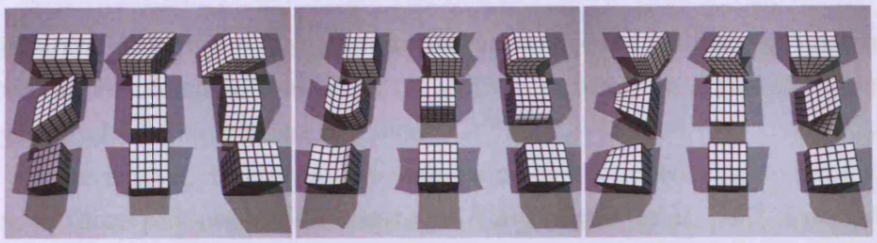


FIGURE 5.2: Modes of deformation admissible under the model of Müller *et al.* [187].

The latter can become unstable if Δt is insufficiently large to account for the displacements occurring during each step, leading to overshooting and erroneous increase in the total energy of the system.

The above rigid shape matching model, Eq. (5.1), is further extended in Müller *et al.* [187] to allow for linear and quadratic deformations (shear, stretch, bend and twist, illustrated in Fig. 5.2). Regardless of the deformation model the principle remains the same: find the optimal matching configuration of points and use it to constrain the Euler integration step.

The approach of Müller *et al.* [187] has been adopted for animating skin deformation (Park and Hodgins [199]) and has recently been extended by Henriques *et al.* [125] for interactive medical applications. A point-based approach is presented by Guo and Qin [120] where a solid volumetric octree-based interior is simulated using Meshless Moving Least Squares shape functions.

One issue with such models is the level of deformation detail that can be represented, and a variety of approaches have been designed to remedy this, for example subdividing objects into clusters in Müller *et al.* [187] or warped modal analysis in Guo and Qin [120]. Fast Lattice Shape Matching (FLSM) is a procedure developed by Rivers and James [224] to account for many more degrees of freedom than Müller by overlapping many rigid clusters of points in a lattice and using the regularity of the lattice to achieve efficient shape matching. The FLSM approach has been extended (Steinmann *et al.* [253]) with an octree-based fast adaptive shape matching algorithm.

These techniques help to make the degree of representable deformations of the objects more detailed, but this does not necessarily lead to more *realistic* behaviour. The main problem remaining is how to measure mechanical properties of an object. In the proposed system, a more sophisticated geometric model is adopted, which is based on learnt statistics of an observed *real* deforming

object, see Section 5.4, to combine the advantages of Müller's geometrically driven approach (speed and stability) and the high degree of realism typical of physically-based simulation methods.

To summarise, data-driven meshless simulation approaches to simulation, such as those proposed in Guo and Qin [120], Müller *et al.* [187], have proved to be extremely valuable in interactive scenarios, such as computer games or virtual surgery simulators, as they allow for unconditionally stable dynamic simulation at much lower computational expense of more traditional methods, *e.g.* finite element (FE) modelling. As demonstrated in Müller *et al.* [187], the idea of shape matching with a quadratic model of deformations (or a piecewise combination of such models) is plausible when simulating simple elastic objects.

However, the range of deformations which their approach affords is very limited and is not well suited to simulating complex objects, such as human faces and other biomechanical entities which are known to undergo very complex deformations and are non-trivially constrained.

The proposed system allows for the capture of idiosyncratic characteristics of an object's dynamics which for many simulations (*e.g.* facial animation) is essential. In existing geometrically motivated animation methods the assumptions about the mechanical properties of objects are too generic and, therefore, preclude this.

The proposed method allows for the plausible simulation of mechanically complex objects without the knowledge of their inner workings. This is especially useful where an object's mechanical properties are hard to measure directly (*e.g.* human face, see the discussion in Section 2.1): to drive realistic simulations, it is sufficient that the typical surface behaviour is learnt. The main idea of the proposed approach is to utilise a flexible statistical model to achieve a geometrically-driven simulation that allows for arbitrarily complex yet easily constrained deformations while at the same time preserving the desirable properties (stability, speed and memory efficiency) of current geometrically driven shape-matching simulation systems.

Statistical modelling, as discussed in Chapter 1, has a long tradition in computer vision, and there is now a considerable literature on applications of statistics-based deformable models. As discussed earlier, PCA has been the mainstay of a variety of linear approaches (*e.g.* Point Distribution

Model (PDM), ASM for modelling of shape, and AAM, MM allowing to model texture in addition to shape, see, for example, Cootes and Taylor [59], Blanz and Vetter [28]). Other linear and non-linear approaches (Linear Discriminant Analysis, Kernel PCA, Multidimensional Scaling, Isomap, non-linear PCA, Locally Linear Embedding and many others) (van der Maaten [170]) have also been utilised. The advantage of linear models such as PCA eigenmodels is their ability to represent principal modes of variation within the data with components spanning the reduced dimensionality space for which two-way projection/unprojection is computationally cheap, which is important for the purposes of this chapter. Some work has been reported on building statistical models from 3D time-varying point clouds (Kaus *et al.* [141], Sübamuth *et al.* [255], Wand *et al.* [289,290]) but these have mostly concentrated on only reconstructing the geometry of the articulated rigid objects.

Recently, fully automated registration methods have been developed, such as the one described in this thesis or Davies *et al.* [76], Sidorov *et al.* [241,243]. These new non-rigid groupwise image and surface registration techniques allow for automatic construction of statistical models of shape and appearance and therefore can be used to prepare models for the proposed quasi-mechanical simulation.

Learning motion from statistical models has also been an active area of research with many approaches utilising Hidden Markov Models (HMMs) to encode motion within a statistical model framework, *e.g.* Brand and Hertzmann [38]. A Gaussian Process Latent Variable Model (GPLVM), described in detail in Lawrence [155], is used in Grochow *et al.* [113] to learn a model of complex articulations of the body from motion capture data to solve the inverse kinematics problem. However, this approach is not suitable for the purposes of this chapter as it does not provide a cheap mapping between the data space and the reduced dimensionality space — it has to solve an optimisation problem each time a mapping is required. While GPLVM is an excellent method for estimating the likelihood function from very few samples, for reasons of efficiency the approach due to Gray and Moore [111] is used instead to estimate kernel density in the reduced dimensionality space, see Section 5.3. Meshless inverse kinematics (Sumner *et al.* [256]) attempts to overcome the potential explosion in degrees of freedom when using meshes instead of skeleton configurations (as

in Grochow *et al.* [113]) by relying on the non-linear multi-way interpolation of unstructured meshes using a deformation-gradient based feature space.

Parenthetically it is worth mentioning an early historic example of combining mechanical ideas with statistical ones in which traditional FE modal analysis methods (Pentland and Sclaroff [202]) have been combined with a statistical model (Martin *et al.* [177]). Both are underpinned by eigenanalysis. Modal analysis can generate a set of vibrational modes from a single shape example, point based statistical methods can model variation between more than one example. In Cootes and Taylor [63] a combined model is built where they generate a large set of new examples from individual variational modes which are then used to augment a point distribution model. These is an early historic example of combining mechanical ideas with statistical ones.

5.2 The Proposed Approach

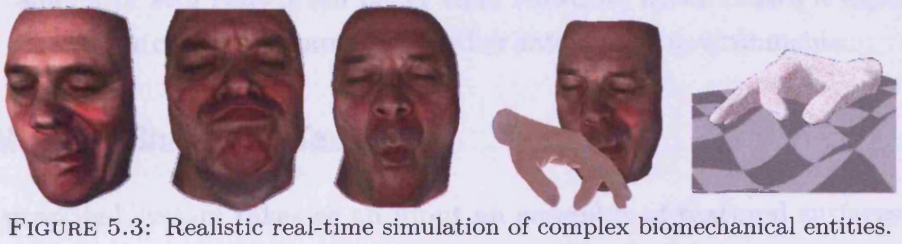


FIGURE 5.3: Realistic real-time simulation of complex biomechanical entities.

By extending the aforementioned techniques of Müller *et al.* [187] and, in particular, by integrating a statistical model into the meshless dynamic simulation paradigm, a system was built with an automated pipeline from capturing characteristic object deformations, encoding these deformations into a learnt statistical model, to finally rendering a faithful simulation (Fig. 5.3). The proposed system preserves all the desirable properties of the algorithm due to Müller *et al.* [187], which are: stability of the dynamic simulation under all circumstances and for all deformed geometry configurations, applicability to a large variety of objects, and computational efficiency in terms of memory requirements and speed. At the same time, the approach is augmented to account for the following:

More realistic simulation. Provided with multiple samples of deformed shapes of an object, for example by observing its evolution in a dynamic

3D surface scanner, important nuances in the underlying space of representable deformations are captured, and, by using this information, much more realistic simulation is achieved.

Texture Synthesis. The additionally available corresponding texture information for each sample of the deforming object is incorporated into the statistical model and then the correct texture appearance of the object, as it undergoes deformation in the proposed simulator, is resynthesised. This greatly increases visual realism.

Automatic Customisability. Mechanical properties of objects are notoriously hard to measure. Therefore, current mechanical simulation approaches are forced to either use generic or tediously hand-crafted models. In contrast, the proposed system enables the capturing of an object's idiosyncrasies into fully automatically constructed customised models using commercially available 3D surface scanners.

Efficiency and Speed. While providing exceptional realism, the proposed simulator still runs much faster than real-time which makes it especially useful for computer games and other interactive environments.

5.3 Model of Shape and Texture

The proposed system takes as an input an ensemble of textured surfaces. For training of the model, samples representative of typical deformations that are to be captured in the model are selected. In order that a statistical model be built, the correspondences between samples are established by using the approach of Chapter 4. Taking advantage of the embedding of meshes into $\mathbb{R}^2$, an artist can additionally specify masses at this point by "painting" a map of masses in the same 2D texture space.

For simulation purposes, the manifold of all plausible configurations that an object can assume has to be represented. A data driven approach was adopted, as it is desirable to be able to build such a model of plausible object shapes from measurements of a real physical object undergoing characteristic deformations.

Consider a parameterised generative model of the form $\mathbf{x} = \mathcal{Q}(\mathbf{b}_s)$, where $\mathbf{b}_s$ is a vector of model parameters, that can be used to generate new instances of shape $\mathbf{x}$ and given an instance of $\mathbf{x}$ would give estimates of parameters

$\mathbf{b}_s = \mathcal{Q}^{-1}(\mathbf{x})$. By enforcing constraints on parameters $\mathbf{b}_s$ it can be ensured that the model generates only plausible shape configurations.

While in general $\mathbf{x} = \mathcal{Q}(\mathbf{b}_s)$ can be any generative model of shape with the above properties, in the proposed approach it is exemplified with a classical linear model, of the same type as AAM of Cootes and Taylor [59], or in 3D case of Sidorov *et al.* [242], which use the well established technique of PCA for dimensionality reduction. The linear reduced dimensionality model of the form $\mathbf{x} \approx \mathrm{E}\mathbf{b}_s + \boldsymbol{\mu}_s$ is now derived for completeness.

Suppose the surface of a deforming object has been sampled at corresponding locations over time as the object undergoes deformation (or, equivalently, an ensemble of surfaces of a deforming object taken over time has been registered using the approach of Chapter 4 and later resampled).

This yields a set of N_s point clouds of N_p points each: $\mathrm{P}_i \in \mathbb{R}^{3\times N_p}$, $i = 1 \ldots N_s$. Each point cloud is centered around its centre of mass and groupwise Procrustes Analysis is applied thus compensating for linear motion in the training data, yielding $\tilde{\mathrm{P}}_i$.

Following the exposition given in Chapter 1, the $3N_p \times N_s$ observation matrix O_s is constructed by reshaping $\tilde{\mathrm{P}}_i$'s into column vectors (observations) and concatenating them together. Let $\boldsymbol{\mu}_s$ be the mean of observations and let mean-centred observations be $\tilde{\mathrm{O}}_s = \mathrm{O}_s - \boldsymbol{\mu}_s \mathbf{1}_{1\times N_s}$. The basis E_s is then simply the first D eigenvectors of the covariance matrix $\mathrm{C} = \tilde{\mathrm{O}}_s \tilde{\mathrm{O}}_s^T$ corresponding to the D largest eigenvalues. Thus the statistical model of shape $\{\mathrm{E}_s, \boldsymbol{\mu}_s\}$.

The proposed method is also concerned with recovering the realistic deformation dependent texture of an object. This follows the standard AAM technique (Cootes and Taylor [59], Sidorov *et al.* [242]). Let O_a be the observation matrix for the texture, constructed by reshaping textures into column vectors and concatenating them together. Applying PCA to O_a results in a linear model of texture $\mathbf{a} = \mathrm{E}_a\mathbf{b}_a + \boldsymbol{\mu}_a$.

For texture recovery, the relationship between shape and texture must be learnt. To accomplish this, a combined model of appearance (Cootes and Taylor [59]) is build. For each example in the training set, compute the corresponding parameters $\mathbf{b}_s$ and $\mathbf{b}_a$, concatenate them in a vector $\mathbf{b}_c = [\mathbf{b}_s^T \ \mathrm{W}_a\mathbf{b}_a^T]^T$, with scaling coefficients W_a to account for difference in units, and construct

the observation matrix for the combined parameters O_c by concatenating $\mathbf{b}_c$'s. Finally, applying PCA again to O_c produces a linear model $\mathbf{b}_c = E_c\mathbf{c}$:

$$\mathbf{b}_c = \begin{pmatrix} \mathbf{b}_s \\ W_a\mathbf{b}_a \end{pmatrix} = E_c\mathbf{c} = \begin{pmatrix} E_{cs} \\ E_{ca} \end{pmatrix}\mathbf{c} \tag{5.7}$$

It is important to note that the above model of shape is linear, while in reality the behaviour of objects will typically be highly non-linear. It is convenient to postpone the discussion of modelling non-linearity (Section 5.4.2) until after the simulation algorithm is explained.

5.4 Simulation

The proposed algorithm for simulation, summarised in Alg. 5.1, will now be described. The main idea behind the proposed method is to use explicit Euler time integration, using a statistical model that defines plausible deformations, to ensure that it never overshoots and is unconditionally stable. The algorithm iteratively evolves a point cloud X, representing the current shape of the object, over time. Each iteration begins with the Euler integration step (lines 5 and 6) which updates the velocities given the accelerations due to external forces (such as gravity). These updated velocities are then used to find the deformed shape X_{new} (line 6). The mass centre, $\hat{\mathbf{x}}_{new}$, of X_{new} is then computed and the points are centered around it to obtain $\tilde{X}_{new}$ (lines 7 and 8). Using Alg. 5.2, described in Section 5.4.1, the best *legitimate* configuration of points, S_{nr}, permitted by the statistical model, is computed, as well as a vector of model parameters, $\mathbf{b}_s$, corresponding to this configuration.

If it is also desirable to maintain the tendency of the object to return to its "undeformed" shape, it is needed to also compute (see below) the best match S_{rigid} between an "undeformed" shape of the object, X_{typ}, and X_{new}. (The choice of X_{typ} is application dependent, it could be any plausible shape (typically one at a rest state) or even a mean shape $\boldsymbol{\mu}_s$ provided this is a likely configuration).

After S_{nr} and S_{rigid} are computed, they are blended linearly (line 12) with parameter β to give a goal shape, X_{goal}. Since X_{goal} is a mixture of the deformed and undeformed shapes, the object will have a tendency (controlled by $\beta \in [0 \ldots 1]$) to return to its undeformed state. (When $\beta = 0$, the object

Algorithm 5.1 Perform simulation

Require: $\{\mathrm{E}_s \in \mathbb{R}^{(3N_p \times D)}, \boldsymbol{\mu}_s \in \mathbb{R}^{(3N_p \times 1)}\}$ — statistical model, Δt — time slice, X_0 — $(3 \times N_p)$ initial configuration of points, $\mathbf{m}_{(1\times N_p)} = \{m_i\}$ — masses of points; $\mathrm{X}_{\mathrm{typ}}$ — "typical" undeformed shape; $\mathbf{g}$ — accelerations due to external "forces", *e.g.* gravity; precomputed texture operator Z; the stiffness parameter $\alpha \in [0 \ldots 1]$ and the shape blending weight $\beta \in [0 \ldots 1]$ (α and β have the same meaning as in Müller *et al.* [187]; see also discussion on pages 168–169).

1: $\mathrm{V}_0 \leftarrow \mathbf{0}_{3\times N_p}$
2: $\hat{\mathbf{x}}_{\mathrm{typ}} \leftarrow \left(\mathrm{X}_{\mathrm{typ}}\mathbf{m}^T\right) / \left(\mathbf{1}_{1\times N_p}\mathbf{m}^T\right)$
3: $\tilde{\mathrm{X}}_{\mathrm{typ}} \leftarrow \mathrm{X}_{\mathrm{typ}} - \hat{\mathbf{x}}_{\mathrm{typ}}\mathbf{1}_{1\times N_p}$
4: **loop**
5: Apply accelerations, *e.g.* $\mathrm{V}_{t+\Delta t} \leftarrow \mathrm{V}_t + (\mathbf{g}\Delta t)\mathbf{1}_{1\times N_p}$
6: $\mathrm{X}_{\mathrm{new}} \leftarrow \mathrm{X}_t + \mathrm{V}_{t+\Delta t}\Delta t$
7: $\hat{\mathbf{x}}_{\mathrm{new}} \leftarrow \left(\mathrm{X}_{\mathrm{new}}\mathbf{m}^T\right) / \left(\mathbf{1}_{1\times N_p}\mathbf{m}^T\right)$
8: $\tilde{\mathrm{X}}_{\mathrm{new}} \leftarrow \mathrm{X}_{\mathrm{new}} - \hat{\mathbf{x}}_{\mathrm{new}}\mathbf{1}_{1\times N_p}$
9: Using Alg. 5.2 compute:
$\{\mathrm{S}_{\mathrm{nr}}, \mathbf{b}_s\} \leftarrow \mathrm{match}(\{\mathrm{E}_s, \boldsymbol{\mu}_s\}, \tilde{\mathrm{X}}_{\mathrm{new}})$
10: $\{\mathrm{R}, \mathrm{S}\} \leftarrow \mathrm{poldec}(\tilde{\mathrm{X}}_{\mathrm{new}}\,\mathrm{diag}(\mathbf{m})\tilde{\mathrm{X}}_{\mathrm{typ}}^T)$
11: $\mathrm{S}_{\mathrm{rigid}} \leftarrow R\tilde{\mathrm{X}}_{\mathrm{typ}}$
12: $\mathrm{X}_{\mathrm{goal}} \leftarrow (\beta\mathrm{S}_{\mathrm{nr}} + (1-\beta)\mathrm{S}_{\mathrm{rigid}}) + \hat{\mathbf{x}}_{\mathrm{new}}\mathbf{1}_{1\times N_p}$
13: $\mathrm{N} \leftarrow \alpha\mathrm{X}_{\mathrm{goal}} + (1-\alpha)\mathrm{X}_{\mathrm{new}}$
14: $\mathrm{X}_{\mathrm{new}} \leftarrow \mathrm{collision}\,(\mathrm{world}, \mathrm{X}, \mathrm{N})$
15: $\mathbf{b}_a \leftarrow \boldsymbol{\mu}_a + \mathrm{E}_a(\mathrm{W}^{-1}\mathrm{E}_{ca}\mathrm{E}_{cs}^T\mathbf{b}_s) = \boldsymbol{\mu}_a + \mathrm{Z}\mathbf{b}_s$
16: $\mathrm{V}_{t+\Delta t} \leftarrow (\mathrm{X}_{\mathrm{new}} - \mathrm{X})/\Delta t$
17: $\mathrm{X}_{t+\Delta t} \leftarrow \mathrm{X}_{\mathrm{new}}$
18: $t \leftarrow t + \Delta t$
19: **end loop**

will return to the undeformed state immediately, and when $\beta = 1$ there will be no such tendency at all.)

As in Müller *et al.* [187], the points are moved α-way towards their goal potions (line 13) to simulate stiffness. The fact that $0 \leq \alpha \leq 1$ ensures that the points never overshoot their legitimate goal positions $\mathrm{X}_{\mathrm{goal}}$. (Note that when $\alpha = 1$ it is a rigid body simulator, and when $\alpha = 0$ there are no internal "elastic" forces at all).

At this point in the algorithm, the interactions with the external world (line 14) need to be considered. The new positions, $\mathrm{X}_{\mathrm{new}}$, are updated to account for collisions. The application-dependent routine $\mathrm{X}_{\mathrm{new}} \leftarrow \mathrm{collision}\,(\mathrm{world}, \mathrm{X}, \mathrm{N})$

here applies hard constraints (such as those arising from collision of the deformable object with obstacles) and returns the deformed configuration of points subject to such constrains. The topic of collision detection and response between multiple deformable objects is outside the scope of this chapter, and is shown only schematically here (the exact mechanism is very application-dependent), for recent innovations see Keiser *et al.* [142], Teschner *et al.* [269].

Computing $\mathrm{S}_{\mathrm{rigid}}$ involves finding the pose parameters (rotation and translation) that best map in the least squares sense the undeformed shape, $\mathrm{X}_{\mathrm{typ}}$, to the deformed shape $\mathrm{X}_{\mathrm{new}}$. The translation is already known ($= \hat{\mathbf{x}}_{\mathrm{new}}$) since $\mathrm{X}_{\mathrm{typ}}$ has already been centered (lines 2–3), and so was $\mathrm{X}_{\mathrm{new}}$. The rotation between $\mathrm{X}_{\mathrm{typ}}$ and $\mathrm{X}_{\mathrm{new}}$ is found by solving the orthogonal Procrustes problem via polar decomposition of the weighted correlation matrix $\tilde{\mathrm{X}}_{\mathrm{new}}\,\mathrm{diag}(\mathbf{m})\tilde{\mathrm{X}}_{\mathrm{typ}}^{T}$, see Lorusso *et al.* [168], Müller *et al.* [187]. The notation $\{\mathrm{R},\mathrm{S}\} \leftarrow \mathrm{poldec}(\mathrm{A})$, in line 10, denotes the polar decomposition of a matrix A. See Higham [127] for definition and a detailed discussion of efficient computation (but see also Müller *et al.* [187]). The point "masses", $\mathbf{m}$, can be used to fine tune the dynamic behaviour of the model by specifying the relative importance of points in the shape matching stage. These, together with the coefficients α, β and γ, allow the artist to tweak the response of the object to external forces. As mentioned above, having a 2D reparameterisation of surfaces (during the registration) provides an artist with a convenient way of tweaking the point "masses": the artist can simply "paint" them.

Further, the most plausible texture corresponding to the current shape $\mathrm{X}_{\mathrm{new}}$ is computed (line 15). Given a vector of shape parameters $\mathbf{b}_s$ it is possible to take advantage of the linear nature of the models to estimate the corresponding texture parameters $\mathbf{b}_a$ by first computing $\mathbf{c} = \mathrm{E}_{cs}^{-1}\mathbf{b}_s = \mathrm{E}_{cs}^{T}\mathbf{b}_s$ and using it to estimate $\mathbf{b}_a = \mathrm{W}^{-1}\mathrm{E}_{ca}\mathbf{c}$. Precomputing $\mathrm{Z} = \mathrm{W}^{-1}\mathrm{E}_{ca}\mathrm{E}_{cs}^{T}$ off-line, the texture parameters $\mathbf{b}_a = \mathrm{Z}\mathbf{b}_s$ can be very quickly estimated. Texture is then recovered at runtime using the linear model $\mathbf{a} = \mathrm{E}_a\mathbf{b}_a + \boldsymbol{\mu}_a$. Note that this amounts to computing a linear combination of basis textures E_a and adding the mean; this can be straightforwardly accomplished on a GPU, by keeping E_a in video memory and synthesising novel textures on the fly. This has an added benefit of having to store only a relatively small basis set in video memory to synthesise a variety of novel textures. This simple approach to texture recovery assumes a bijective (and, moreover, linear) relationship between shape and texture.

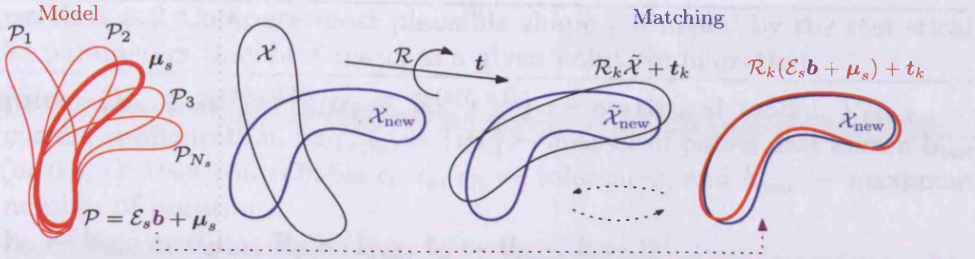


FIGURE 5.4: Find the most plausible shape given the statistical model and constraints.

Despite the strong assumption, in practice this approach is a good compromise between computational efficiency and realism.

Finally, the velocities V are updated to reflect by how much the points have actually moved during this iteration.

5.4.1 Shape Matching

The problem of computing the most plausible shape permitted by the statistical model that best matches a given point configuration with *a priori* known correspondences (Alg. 5.2) is now addressed. Solving this problem involves minimising the expression

$$\|\mathrm{P} - (\mathrm{R}\mathcal{Q}(\mathbf{b}_s) + \mathbf{t})\|^2 \rightarrow \min \qquad \text{w.r.t. R, } \mathbf{b}_s \text{ and } \mathbf{t}. \tag{5.8}$$

Here, $\mathcal{Q}(\mathbf{b}_s)$ denotes an instance of shape generated by the model given parameters $\mathbf{b}_s$. In the case of a linear model, $\mathcal{Q}(\mathbf{b}_s)$ is simply $\mathrm{E}_s\mathbf{b}_s + \boldsymbol{\mu}_s$. This is an optimisation problem and it is solved in an iterative fashion similarly to Cootes and Taylor [59]. To avoid conflict of subscripts, in Alg. 5.2 shape parameters $\mathbf{b}_s$ are denoted as simply $\mathbf{b}$. The search begins by initialising the model parameters, $\mathbf{b}_0$, to zero (or the last known value $\mathbf{b}_{\text{last}}$) and the pose parameters to the identity transform and, also, ensuring that input points are centred around the centre of mass (lines 2 and 3).

The iterative body of the algorithm operates as follows. Constraints (Fig. 5.6) are applied to $\mathbf{b}_{k-1}$ to ensure that it corresponds to a plausible shape (line 6). The constraints are made less hard with a mixing coefficient γ to allow for the object to assume any pose while still possessing a strong tendency to evolve towards valid shapes (line 7, see below). An instance of shape Q_k is then computed using the model controlled by parameters $\mathbf{b}_{k-1}$ and is centred around

Algorithm 5.2 Compute most plausible shape permitted by the statistical model parameters that best matches a given point configuration

Require: $\{\mathrm{E}_s \in \mathbb{R}^{(3N_p \times D)}, \boldsymbol{\mu}_s \in \mathbb{R}^{(3N_p \times 1)}\}$ — statistical model, $\mathrm{P}_{(3\times N_p)}$ — current configuration, $\mathbf{m}_{(1\times N_p)} = \{m_i\}$ — masses of points, last known $\mathbf{b}_{\text{last}}$ (or $\mathbf{0}_{D\times 1}$); User-controllable ϵ_t, ϵ_b, ϵ_R — tolerances, and $k_{\max}$ — maximum number of iterations.

1: $\mathbf{b}_0 \leftarrow \mathbf{b}_{\text{last}}$ or $\mathbf{0}_{D\times 1}$, $\mathrm{R}_0 \leftarrow \mathrm{I}_{3\times 3}$, $\mathbf{t}_0 \leftarrow \mathbf{0}_{3\times 1}$, $k \leftarrow 0$
2: $\hat{\mathbf{p}} \leftarrow \left(\mathrm{P}\mathbf{m}^T\right) / \left(\mathbf{1}_{1\times N_p}\mathbf{m}^T\right)$
3: $\tilde{\mathrm{P}} \leftarrow \mathrm{P} - \hat{\mathbf{p}}\mathbf{1}_{1\times N_p}$
4: **repeat**
5: $\quad k \leftarrow k + 1$
6: $\quad \mathbf{b}_{\text{cons}} \leftarrow \text{constrain}(\mathbf{b}_{k-1})$
7: $\quad \mathbf{b}_{k-1} \leftarrow (1-\gamma)\mathbf{b}_{\mathbf{k-1}} + \gamma\mathbf{b}_{\text{cons}}$
8: $\quad \mathrm{Q}_k \leftarrow \text{reshape}_{[3\times N_p]}\left(\mathrm{E}_s\mathbf{b}_{k-1} + \boldsymbol{\mu}_s\right)$
9: $\quad \hat{\mathbf{q}}_k \leftarrow \left(Q_k\mathbf{m}^T\right) / \left(\mathbf{1}_{1\times N_p}\mathbf{m}^T\right)$
10: $\quad \tilde{\mathrm{Q}} \leftarrow \mathrm{Q}_k - \hat{\mathbf{q}}_k\mathbf{1}_{1\times N_p}$
11: $\quad \{\mathrm{R}_k, \mathrm{S}\} \leftarrow \text{poldec}(\tilde{\mathrm{Q}}\,\text{diag}(\mathbf{m})\tilde{\mathrm{P}}^{\mathrm{T}})$
12: $\quad \mathbf{t}_k \leftarrow \hat{\mathbf{p}} - \hat{\mathbf{q}}_k$
13: $\quad \mathrm{Y} \leftarrow \mathrm{R}_k(\tilde{\mathrm{P}} - \mathbf{t}_k\mathbf{1}_{1\times N_p})$
14: $\quad \mathbf{b}_k \leftarrow \mathrm{E}_s^T\left(\text{reshape}_{[3N_p\times 1]}(\mathrm{Y}) - \boldsymbol{\mu}_s\right)$
15: $\quad \Delta\mathrm{R} \leftarrow \mathrm{R}_k - \mathrm{R}_{k-1}$
16: **until** $k \geq k_{\max}$ **or** $(\|\mathbf{t}_k - \mathbf{t}_{k-1}\| < \epsilon_t$ **and** $\|\mathbf{b}_k - \mathbf{b}_{k-1}\| < \epsilon_b$ **and** $\sqrt{\text{trace}(\Delta\mathrm{R}^T\Delta\mathrm{R})} < \epsilon_R)$
17: **return** $\{\mathrm{S}_{\text{nr}} \leftarrow \mathrm{R}_k^{-1}\mathrm{Q}^k + \mathbf{t}_k\mathbf{1}_{1\times N_p}, \mathbf{b}_k\}$

its centre of mass (lines 8–10). The pose parameters, R_k and $\mathbf{t}_k$, are then found that best map P to hypothesis Q_k as described above (lines 11–12). Note that the physical meaning of computing the optimal rotation and translation parameters, R_k and $\mathbf{t}_k$, in addition to the non-rigid shape parameters, $\mathbf{b}$, is to make sure that the angular momentum and momentum are preserved. Using this pose estimate, Y is computed, which is the new position of P, in the model coordinate frame (line 13). New model parameters $\mathbf{b}_k$ are computed that best approximate Y. This process is illustrated in Fig. 5.4. Lines 5–15 are repeated until convergence when there are no significant changes in $\mathrm{R}, \mathbf{t}$, and $\mathbf{b}$. In practice convergence occurs after only a few iterations.

Finally, the last estimate of Q_k is transformed back into the coordinate frame of P to give to most plausible shape S_{nr}.

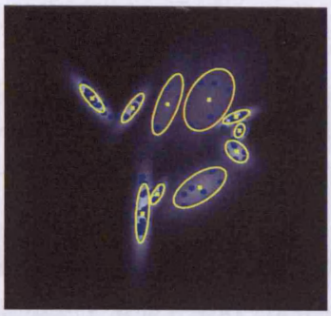


FIGURE 5.5: A 10-component GMM fitted to shape data in the reduced dimensionality space. Colour indicates PDF (annealed, for display, by raising it to power 0.1).

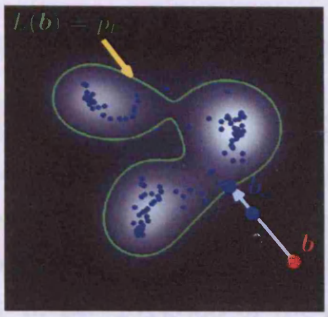


FIGURE 5.6: Probability density estimated using the method of Gray and Moore [111], and its level set serving as constraints on **b**.

5.4.2 Modelling Non-linearity and Constraining the Shape Parameters, b

The distribution of shape parameters **b** will not in general form a simple Gaussian assumed by the linear eigenmodel of shape. One of the established techniques for modelling non-linear data sets is to assume that although the whole data set is non-linear it can be approximated with a mixture of locally linear models (Hicks [126]). A classical approach to such approximation are GMMs which have the form (Duda *et al.* [91])

$$p(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i G(\mathbf{x}, \boldsymbol{\mu}_i, \mathrm{C}_i), \tag{5.9}$$

where n is the number of Gaussians in the mixture, $\boldsymbol{\mu}$ are the centres of the Gaussians, and C_i are the corresponding covariance matrices, and α_i are the prior probabilities. The parameters of the GMM are estimated using the well-known EM algorithm (Duda *et al.* [91], Press *et al.* [220]). Fitting a GMM to the original data set would be impractical (not only because of the high running time, but also because the fitting procedure is fragile in high dimensions), so a standard (Hicks [126], Aubrey [13], Cosker [68]) approach is to fit a GMM to the data in the reduced dimensionality space. Fitting of a GMM to a reduced dimensionality shape eigenmodel is illustrated in Fig. 5.5.

In practice, fitting a GMM to data suffers from a number of drawbacks: potential danger of over- and under-fitting, the need to select the number of components in the mixture, and the danger of the EM procedure getting stuck in local minima. These problems become particularly prominent if the number

of data points, approximating the "true" distribution, is small. These problems are also pointed out in Grochow *et al.* [113].

Having estimated the distribution of parameters $\mathbf{b}$, with PDF $p(\mathbf{b})$, define a configuration $\mathbf{b}$ as plausible if $p(\mathbf{b})$ is greater than some threshold p_t, see Fig. 5.6. One way to ensure that $\mathbf{b}$ tends to assume values corresponding to plausible shapes is to estimate the likelihood at each point in the reduced dimensionality space and then move it uphill in the direction of the gradient. Such is the approach adopted in Grochow *et al.* [113], where GPLVM is used to estimate the PDF of the distribution. This is an optimisation problem which is solved with an iterative algorithm whose convergence properties depend on the data and are hard to predict. It is also possible to accomplish the same using a GMM, especially since the gradient of the PDF of the mixture can be derived analytically (Rennie [222]).

Since GMMs and GPLVMs have the above shortcomings, a different approach was adopted, remarking that the actual PDF is of no interest: only a method (a *classifier*) to separate plausible shapes from implausible ones is required, together with a procedure to move an implausible shape towards the nearest plausible one.

When enforcing constraints on values of $\mathbf{b}$ the aim is to achieve absolute stability and predictability, which is important in interactive scenarios, as well as to make enforcement of constraints computationally very cheap. In the proposed algorithm, this is achieved by pre-computing the likelihood $L(\mathbf{b})$ at each point (on a regular grid, say) in the reduced dimensionality space off-line at the model building stage. To do so, the procedure described in Gray and Moore [111] for kernel density estimation is used. It proves to work very well even if the data set is sparse and doesn't suffer from the drawbacks of the GMM approach. The level set of this function $L(\mathbf{b}) = p_t$ (Fig. 5.6) is then computed and stored. For low dimensional $\mathbf{b}$'s this level set is stored as a set of polygons (or polygonal surfaces) that divide the reduced dimensionality space into regions of plausible and implausible values for $\mathbf{b}$.

This approach is computationally very cheap, however polygonal constraints do not generalise well if the dimensionality of $\mathbf{b}$ is higher than three. In practice, it is sufficient to apply constraints only to the first 2–3 most significant modes which capture most of the variation. During the simulation, implausible

parameters $\mathbf{b}$ are adjusted by forcing them γ-way towards the nearest point (line 7) on the level set (polygon).

An alternative approach, due to Bowden *et al.* [36], is to approximate the manifold with an ensemble of locally bounding boxes. Bowden *et al.* [36] apply cluster analysis to the points in the reduced dimensionality space and, having clustered the points, compute a bounding box for each cluster. The superposition of bounding boxes thus constraints the subspace of valid configurations.

5.5 Experiments

A simulator has been constructed with which the superior realism of the proposed statistical approach and its potential can be demonstrated. The results are illustrated on a selection of biomechanical objects: a human face, an artificially created model of a human hand and a human abdomen. Results from applying time-varying hard constraints to objects as well as constraints due to interactions (collisions) between them are shown. Also shown is the inclusion of texture resynthesis as an object undergoes deformation which is not currently afforded by any meshless simulation algorithms. All the simulations were performed on a typical PC and run much faster than real-time (collision detection, a topic not discussed here, being the only bottleneck).

5.5.1 Human Head Simulation

For this experiment, 3D video data from a 3D dynamic scanner capturing high resolution ($\approx$ 20k–30k triangles) meshes and accompanying texture maps at 48 frames per second was acquired, preprocessed, and registered. A human face undergoing deformations that sampled enough variation to model were captured: the head making typical chewing, cheek blowing, lip pursing and moving pursed lips sideways, were sampled thus creating automatically customised models of kinematics. See Chapter 4 for the description of the source data used. It is sufficient to select a characteristic subset of training video frames; typically, 30–100 samples are used when constructing the statistical models.

Fig. 5.7 shows a series of frames from a simulation sequence where hard constraints were applied to the points on the boundary of the face and varied over time (quickly rotating and moving) to simulate how the rest of the face

FIGURE 5.7: Human head simulated with the proposed method (Top: side to side motion, Bottom: "nodding" motion)

FIGURE 5.8: Same side to side head simulation with Müller's method. Note the unnatural behaviour.

realistically responds. In Fig. 5.7 one can clearly see that the deformations in the soft tissues around the mouth and nose are realistically computed and rendered. For comparison, the proposed method was compared with a vanilla implementation of meshless simulation due to Müller *et al.* [187]. However, a direct comparisons is not possible as even the piecewise multi-cluster extension of Müller's approach does not offer enough controllability to plausibly simulate a human face. Figure 5.8 shows Müller's approach in a similar scenario, clearly the proposed approach is incomparably an improvement.

Figure 5.7 shows results from a similar experiment to above, except that now the head has "nod" constraints moves it up and down fairly vigorously.

Again comparison to Müller's approach shows a much more biomechanically realistic simulation.

In order to illustrate the advantages of texture resynthesis, consider a close up of the above simulation, Fig. 5.9. It is important to not only simulate the shape (Fig. 5.9(c)) but also to resynthesise the textures accordingly. Figure 5.9(d) shows the resynthesised texture. Visually important details, such as deep wrinkles in the human face, synthesised as part of the texture greatly increase the realism.

(a) (b)

(c) (d)

FIGURE 5.9: (a) Surface in rest state. (c) Surface after deformation, but using the same texture as (a). (d) Texture recovered to reflect the deformation. (b) Underlying mesh. Notice how the deep wrinkles are synthesised in the texture, they are not a result of shading in the mesh.

5.5.2 Artificial Hand Simulation

Artificially created models were also simulated. For this experiment, an animated model of a human hand from the Utah 3D Animation Repository [3] was obtained. Note that the texture is static in this model.

FIGURE 5.10: Artificial hand simulated with the proposed method.

FIGURE 5.11: Face slap simulation with the proposed method.

This simple sequence encompasses some basic individual finger movement as well as a simple clenching action between the thumb and forefingers. This experiment simulates the hand falling under gravity and colliding with a hard uneven surface, see Fig. 5.10. This involves some collision detection, the methods of which are beyond the scope of this chapter. A basic collision detection procedure was implemented. Here, the hard constraints come from the collision with the world. Even such a simple model yields a realistic simulation: when points of the hand collide with the surface the rest of the hand responds naturally.

5.5.3 Face Slap Simulation

In this experiment the face and hand model from above were used to simulate the hand slapping the face. Figure 5.11 shows the hand coming into contact with the face and simulating a realistic face slap. As above, the objects objects respond faithfully to collisions.

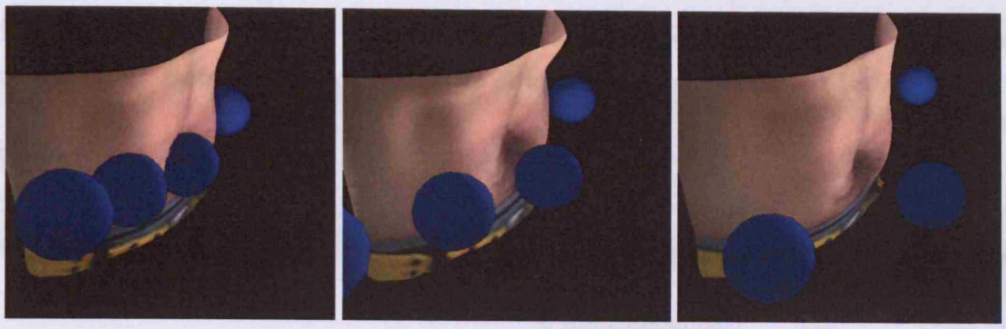
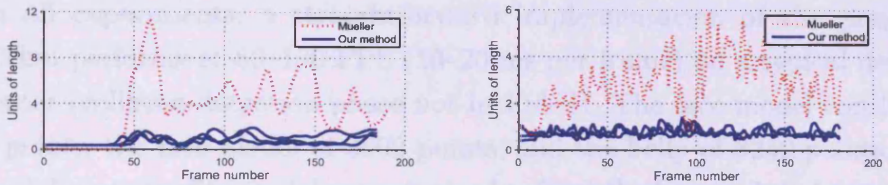


FIGURE 5.12: Balls hitting abdomen simulation with the proposed method.



(a) Rotating head experiment. (b) Falling hand experiment.

FIGURE 5.13: RMS difference from reference set.

5.5.4 Multiple Balls Hitting a Human Abdomen Simulation

Finally, consider an example where multiple balls hit a human abdomen. A person "wobbling his belly" was captured with a 3D video scanner and a statistical model of the abdomen deformations was built. The simulation in this experiment consisted of firing multiple balls at the abdomen, again observing realistic results (Fig. 5.12).

Finally, in Fig. 5.13, a basic empirical comparison with Müller's original method is shown. Two similar sets were constructed for this experiment: a training set and a reference set. The training set consists of all even frames from the original data sequence, with which the statistical model was built. The reference set (constructed by taking all odd samples from the original data sequence) was used to quantify the plausibility of the simulation: at every time step of a simulation the difference between the current state of the object and the nearest rigidly matching example from the reference set was computed. This difference measure is plotted in Fig. 5.13(a) and (b) for the above mentioned rotating head and falling hand examples. As expected, the plots show that the proposed method is closer to the "ground truth" reference

data than Müller's due to the fact that the simulation uses a statistical model that is built from similar data and is efficiently constrained.

5.5.5 Note on performance

> Last night the Chernobyl Nuclear Powerplant fulfilled the Five Year Plan of heat energy generation in just under 4 microseconds.
>
> "Pravda", Soviet newspaper

In all experiments, a straightforward implementation of the proposed algorithm performs at 50–100 FPS (10–20 ms per frame) on a typical desktop computer (collision detection phase not included). The face model consists of 8390 points, the face model of 3502 points, and the belly of 3249 points. The reduced dimensionality models were trained to keep the largest 3–4 eigenvectors, in these examples capturing more than 90% of total variation. This means, for example, that the proposed texture resynthesis method has a memory footprint only 3–4 times bigger than would have been needed for a straightforward texture mapping. Measurements show that the most expensive part of the simulation is the shape matching Alg. 5.2. However, in all experiments ($D = 3$–4) it always converged in between 1–4 iterations, depending on the magnitude of the "current" deformation. The maximum number of permitted iterations $k_{\max}$ can be capped to a smaller value, to further increase the speed by slightly sacrificing the accuracy. Same applies to tolerances ϵ_t, ϵ_b and ϵ_R. This works because even if at frame n the matching procedure is stopped slightly too early, before attaining the optimal target configuration of points, the points are then moved towards this approximately optimal configuration, and at the next, $n+1$'st, iteration the matching procedure begins with the new improved initial state and so on. In other words it is sufficient that the matching algorithm converges each time to a solution that is "ahead of the game".

Further improvement in speed can be achieved by simulating only a subset of points, interpolating the results to the rest of the object.

5.6 Future Work

Future work includes enhancing the proposed approach to accommodate for multiple models per object (akin to piecewise clusters in Müller *et al.* [187]). This would be relatively simple to achieve and would lead to even more flexibility when rendering complex articulated objects. In particular, for objects consisting of multiple independently moving parts, this would help overcome the exponential explosion of the model size, by creating a separate simple model for each of the parts (with an overlap as in Müller *et al.* [187]) and simulating them independently. This will also help overcome some of the limitations of the linear nature of the shape models. In the proposed method, constraints on the shape parameters enforce the non-linear plausible behaviour, but in case of *e.g.* large rotational deformations or deformations extrapolating way beyond the training data even they might not be appropriate.

Another direction involves incorporation of more complex dynamic constraints such as HMMs. It would be interesting to apply the proposed method and its future extensions to full human body motion, skin modelling, medical applications (including virtual surgery), as well exploring further applications in human facial animation.

While excellent visual realism was demonstrated qualitatively, admittedly, more experiments are required to better evaluate the proposed method quantitatively. This is a difficult task since visual realism is not easy to quantify. Additionally, since this is a niche approach, a direct comparison with some of the other mechanical simulation methods (other than geometry-based) is non-trivial.

5.7 Conclusions

A new way to use statistical models to achieve a high degree of realism in addition to the advantages of existing meshless methods when simulating deformable objects in real-time was presented. In particular, a system was built with a fully automated pipeline to construct customised models capturing idiosyncratic object deformations, encode these deformations into a learnt statistical model, and finally render a faithful simulation. Thus, this chapter offers a plug-and-play computationally cheap replacement for the method of

Müller *et al.* [187], at no extra cost in terms of preparation, which also features a significant improvement in quality, provided with a few training samples from readily available surface scanners, and makes no further assumptions (material properties are not known, nor are the dynamic properties of the object). Realism is improved by capturing important nuances of an object's deformation as well as incorporating texture resynthesis as an object undergoes deformation. The benefits of the proposed system were demonstrated with a few simulations of biomechanical objects (which are notoriously hard to simulate). The experiments demonstrate great potential and applicability of the proposed real-time approach.

CHAPTER

6 Conclusions and Future Work

> Life has improved, comrades. Life has become more joyous.
>
> Joseph Stalin

THIS thesis primarily focuses on methods to enable automatic construction of models of craniofacial appearance, for purposes of analysis, synthesis, and simulation. This research was in part motivated by the needs of orthodontics to automate the analysis and modelling of patients' craniofacial complex using 2D and 3D imagery. However, the methods developed and presented in this work are not limited to craniofacial imagery, but are generic and have much wider applicability in computer vision and graphics.

Chapters 3 and 4 concentrate on the problem of groupwise non-rigid registration itself and consider the cases of 2D images and 3D textured surfaces respectively. Chapter 3 offers a novel, fast, reliable, and fully automatic approach to groupwise non-rigid image registration. The efficiency is achieved through implicitly reducing the dimensionality of the search space by representing increasingly complex deformations as a superposition of simpler ones. At the heart of the optimisation framework is a stochastic optimiser, an adaptation of SPSA, intimately integrated into the groupwise registration framework.

To leverage the new source of data, video-rate 3D surface imagery, which is becoming an important medical imaging tool as well as proving valuable in

computer graphics and vision, Chapter 4 offers a generalisation of the approach presented in Chapter 3 to register ensembles of textured 3D surfaces.

The proposed registration approaches, are formulated in a way that is amenable for GPU implementation, allowing one to leverage the power of modern graphics hardware.

In all experiments, both methods demonstrate high robustness and success rate, accuracy, as well as fast convergence on various types of test data. This includes the notoriously difficult case of inter-subject registration. At the time of publishing the CVPR '09 paper (Sidorov *et al.* [243]), this was the first time that the groupwise registration of data possessing such variety (faces of multiple people) had been reported.

Experiments in Chapter 3 also show considerable improvement in terms of accuracy of solution and speed compared to existing methods. The usefulness of the proposed registration algorithms for appearance model building is further illustrated by examples of automatically constructing both 2D and 3D models of appearance from raw data.

Chapter 5 illustrates the usefulness of the ideas presented in the previous chapters by offering a novel application of statistical appearance modelling: statistically driven simulation. Chapter 5 presents a new way to use statistical models to achieve a high degree of realism in addition to the advantages of existing geometrically motivated methods when simulating deformable objects in real-time. This is illustrated by a system which was built comprising a fully automated pipeline to construct customised models capturing idiosyncratic object deformations, encode these deformations into a learnt statistical model, and finally render a faithful simulation.

Essentially, the proposed simulation approach is a plug-and-play computationally cheap replacement for the state of the art geometrically driven methods, at no extra cost in terms of preparation. However, the proposed approach features a significant improvement in quality, provided with a few training samples from readily available surface scanners.

In summary, the main contributions of this thesis, with respect to their appearance in the text, are:

- A novel efficient stochastic algorithm for groupwise non-rigid registration of images. The proposed algorithm is shown to register sizeable image

ensembles quickly and more accurately than state of the art methods (Chapter 3). Experiments demonstrate the reliability of the proposed approach on data with very high variability, in particular pioneering the notoriously difficult case of See also Sidorov *et al.* [243]. inter-subject registration.

- A generalisation of the above algorithm to the case of textured 3D surfaces (Chapter 4). The proposed 3D registration algorithm retains all the desirable properties of the above 2D algorithm and allows for groupwise non-rigid registration of 3D surfaces in a principled way. This opens new research prospects by allowing a new valuable source of data to be leveraged: textured 3D surfaces produced by video-rate surface scanners which have recently gained popularity. See also Sidorov *et al.* [241, 242, 244].
- To show the usefulness of the proposed registration framework in appearance model building, a novel application of statistical appearance modelling is presented: a faster that real-time quasi-mechanical simulator of deformable objects using statistical constraints (Chapter 5). Experiments demonstrate the entire pipeline from acquisition, registration and model building, to physically realistic real-time simulation of deformable objects.

While it is unlikely that a perfect general solution to the problem of groupwise image registration will be found in the nearest future, good practical solutions are perfectly possible, as demonstrated in this thesis and is constantly demonstrated by the human visual subsystem.

The work presented in this thesis has a lot of potential and opens new directions of future research. The possible avenues for future work were outlined at the end of each chapter and are now briefly summarised.

The research in Chapter 3 can be continued with an investigation of techniques to avoid using a single model of texture in the groupwise registration framework, and so avoid the problem of using the per-pixel statistics from initially poorly registered ensembles. The related question is that of investigating the pathways of information propagation between images, in the general sense. The problem of automatic construction of multilinear models is also

a possible extension of the research in Chapter 3, as well as modelling of non-diffeomorphic deformations, with *e.g.* layered appearance models.

The research in Chapter 4 can be followed up with an investigation of whether "bending invariants" of images, by analogy with those of surfaces, can be leveraged for image registration. It is also important to research whether the framework proposed in Chapter 4 can be modified to avoid embedding of surfaces into a reference plane, for example by using GMDS.

The simulation framework presented in Chapter 5 will benefit from the ability to simulate composite objects, *i.e.* simulating several interacting models simultaneously. This would allow to further increase the usefulness of the algorithm in applications like full human body motion, skin modelling, medical virtual surgery, as well further applications in human facial animation.

List of Acronyms

AAM	Active Appearance Model
ASM	Active Shape Model
BAAM	Bilinear Active Appearance Model
BEM	Boundary Element Method
BP	Belief Propagation
CAT	computed axial tomography
CC	correlation coefficient
CG	computer graphics
CPS	Clamped-Plate Spline
DCT	Discrete Cosine Transform
EM	Expectation-Maximisation
EVD	eigenvalue decomposition
FE	finite element
FFD	free-form deformation
FFT	Fast Fourier Transform
FLD	Fisher's Linear Discriminant
FLSM	Fast Lattice Shape Matching
FM	Fast Marching
FMM	Fast Marching Method
FPS	Farthest Point Sampling
FV	finite volume
GA	genetic algorithm
GD	gradient descent
GMDS	Generalised Multidimensional Scaling
GMM	Gaussian Mixture Model
GPA	Generalised Procrustes Analysis
GPLVM	Gaussian Process Latent Variable Model

GPU	graphics processing unit
HMM	Hidden Markov Model
KE	Kernel Eigenfaces
KF	Kernel Fisherfaces
KFLD	Kernel Fisher's Linear Discriminant
KLT	Karhunen-Loève transform
KPCA	Kernel Principal Component Analysis
LDA	Linear Discriminant Analysis
MAD	mean absolute difference
MDL	Minimum Description Length
MDS	Multidimensional Scaling
MFFD	Multi-level Free-form Deformation
MI	mutual information
MM	Morphable Model
MPEG	Moving Picture Experts Group
MRF	Markov Random Field
MRI	magnetic resonance imaging
MSE	mean square error
NCC	normalised cross correlation
NMI	normalised mutual information
ODE	ordinary differential equation
PCA	Principal Component Analysis
PDE	partial differential equation
PDF	probability density function
PDM	Point Distribution Model
PET	positron emission tomography
PSNR	peak signal to noise ratio
PSO	Particle Swarm Optimisation
RBF	radial basis function
RC	Residual Complexity
RoI	region of interest
SA	Simulated Annealing
SAD	sum of absolute differences
SD	standard deviation
SIFT	Scale-Invariant Feature Transform
SPSA	Simultaneous Perturbation Stochastic Approximation
SSD	sum of squared differences
TPS	Thin Plate Spline

Bibliography

[1] Middlebury Stereo Evaluation. http://vision.middlebury.edu/stereo/eval.

[2] QSlim mesh decimation toolbox. http://mgarland.org/software/qslim.html.

[3] Utah 3D animation repository. www.sci.utah.edu/~wald/animrep/hand/hand.obj.tgz.

[4] XM2VTS data set. http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb.

[5] XM2VTS manual annotation. http://www-prima.inrialpes.fr/FGnet/data/07-XM2VTS/xm2vts_markup.html.

[6] 3DMD. 4D capture system. http://www.3dmd.com.

[7] B. ALLEN, B. CURLESS, Z. POPOVIĆ, ET AL. Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 147–156. Eurographics Association, Aire-la-Ville, Switzerland, 2006.

[8] I. AMIDROR. Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, volume 11(2):pages 157–176, 2002.

[9] P. ANDRESEN, F. BOOKSTEIN, K. CONRADSEN, ET AL. Surface-bounded growth modeling applied to human mandibles. *IEEE Transactions on Medical Imaging*, volume 19(11):pages 1053–1063, November 2000. ISSN 0278-0062.

[10] E. ANGELINI, O. CLATZ, E. MANDONNET, ET AL. Glioma dynamics and computational models: a review of segmentation, registration, and in silico growth algorithms and their clinical applications. *Current Medical Imaging Reviews*, volume 3(4):pages 262–276, November 2007.

[11] N. ARCHIP, O. CLATZ, S. WHALEN, ET AL. Non-rigid alignment of pre-operative MRI, fMRI, and DT-MRI with intra-operative MRI for enhanced visualization and navigation in image-guided neurosurgery. *Neuroimage*, volume 35(2):pages 609–624, April 2007.

[12] V. I. ARNOLD. *Lectures on Partial Differential Equations.* Springer, 1st edition, January 2004. ISBN 3540404481.

[13] A. J. AUBREY. *Exploiting The Bimodality Of Speech In The Cocktail Party Problem.* PhD thesis, Centre of Digital Signal Processing, Cardiff University, 2008.

[14] S. BAKER, I. MATTHEWS, J. SCHNEIDER. Automatic construction of active appearance models as an image coding problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 26(10):pages 1380–1384, October 2004. ISSN 0162-8828.

[15] S. BALCI, P. GOLLAND, M. SHENTON, ET AL. Free-form B-spline deformation model for groupwise registration. In *10th International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 10, pages 23–30. October 2007.

[16] S. BALCI, P. GOLLAND, W. M. WELLS, III. Non-rigid groupwise registration using B-spline deformation model. In *10th International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 10, pages 105–121. 10 2007.

[17] D. BARAFF, A. WITKIN. Large steps in cloth simulation. In *Proc. of SIGGRAPH '98*, pages 43–54. ACM, New York, NY, USA, 1998. ISBN 0-89791-999-8.

[18] L. BELDIE, B. WALKER, Y. LU, ET AL. Finite element modelling of maxillofacial surgery and facial expressions — a preliminary study. *The International Journal of Medical Robotics and Computer Assisted Surgery (to appear)*, 2010.

[19] P. N. BELHUMEUR, J. P. HESPANHA, D. J. KRIEGMAN. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19(7):pages 711–720, August 1997. ISSN 0162-8828.

[20] L. BENEDIKT. *Using 3D Facial Motion for Biometric Identification.* PhD thesis, School of Computer Science, Cardiff University, August 2009.

[21] M. BERG, O. CHEONG, M. KREVELD, ET AL. *Computational Geometry: Algorithms and Applications.* Springer, 3rd edition, April 2008. ISBN 3540779736.

[22] F. BETTINGER, T.F.COOTES. A model of facial behaviour. *Proc. Int.Conf on Face and Gesture Recognition*, pages 123–128, 2004.

[23] K. K. BHATIA, J. HAJNAL, A. HAMMERS, ET AL. Similarity metrics for groupwise non-rigid registration. In *MICCAI'07: Proceedings of the 10th international conference on Medical image computing and computer-assisted intervention*, pages 544–552. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN 3-540-75758-9, 978-3-540-75758-0.

[24] M. J. BLACK, D. J. FLEET, Y. YACOOB. Robustly estimating changes in image appearance. *Computer Vision and Image Understanding*, volume 78:pages 8–31, 2000.

[25] M. J. BLACK, A. RANGARAJAN. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vision*, volume 19(1):pages 57–91, 1996. ISSN 0920-5691.

[26] V. BLANZ, K. SCHERBAUM, H.-P. SEIDEL. Fitting a morphable model to 3d scans of faces. In *IEEE 11th International Conference on Computer Vision (ICCV 2007)*, pages 1–8. October 2007. ISSN 1550-5499.

[27] V. BLANZ, K. SCHERBAUM, T. VETTER, ET AL. Exchanging faces in images. In M.-P. CANI, M. SLATER (editors), *The European Association for Computer Graphics 25th Annual Conference EUROGRAPHICS 2004*, volume 23(3) of *Computer Graphics Forum*, pages 669–676. Blackwell, Grenoble, France, 2004. ISBN 0167-7055.

[28] V. BLANZ, T. VETTER. A morphable model for the synthesis of 3D faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1999. ISBN 0-201-48560-5.

[29] D. BLEZEK, J. MILLER. Atlas stratification. In R. LARSEN, M. NIELSEN, J. SPORRING (editors), *Proceedings of 9th International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 11, pages 443–457. Med Image Anal, Copenhagen, Denmark, October 2006. ISBN 3-540-44707-5.

[30] T. BOGGIO. Sulle funzioni di Green d'ordine *m*. *Rendiconti – Circolo Matematico di Palermo*, volume 20:pages 97–135, 1905.

[31] D. BOMMES, L. KOBBELT. Accurate computation of geodesic distance fields for polygonal curves on triangle meshes. In H. P. A. LENSCH, B. ROSENHAHN, H.-P. SEIDEL, ET AL. (editors), *VMV*, pages 151–160. Aka GmbH, 2007.

[32] F. BOOKSTEIN. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 11(6):pages 567–585, June 1989. ISSN 0162-8828.

[33] F. L. BOOKSTEIN. *Morphometric tools for landmark data: geometry and biology*. Cambridge University Press, June 1997. ISBN 0521585988.

[34] I. Borg, P. Groenen. *Modern Multidimensional Scaling: Theory and Applications.* Springer, 2005.

[35] S. Bougleux, G. Peyré and, L. Cohen. Image compression with anisotropic triangulations. In *IEEE 12th International Conference on Computer Vision 2009*, pages 2343–2348. September 2009. ISSN 1550-5499.

[36] R. Bowden, T. A. Mitchell, M. Sarhadi. Non-linear statistical models for the 3D reconstruction of human pose and motion from monocular image sequences. *Image Vision Computing*, volume 18(9):pages 729–737, 2000.

[37] K. W. Bowyer, K. Chang, P. Flynn. A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition. *Comput. Vis. Image Underst.*, volume 101:pages 1–15, January 2006. ISSN 1077-3142.

[38] M. Brand, A. Hertzmann. Style machines. In *Proc. of SIGGRAPH '00*, pages 183–192. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000. ISBN 1-58113-208-5.

[39] K. Briechle, U. D. Hanebeck. Template matching using fast normalized cross correlation. In *Proceedings of SPIE: Optical Pattern Recognition XII*, volume 4387, pages 95–102. March 2001.

[40] M. Bro-Nielsen, C. Gramkow. Fast fluid registration of medical images. In *VBC '96: Proceedings of the 4th International Conference on Visualization in Biomedical Computing*, pages 267–276. Springer-Verlag, London, UK, 1996. ISBN 3-540-61649-7.

[41] A. Bronstein. Toolbox for surface comparison and analysis (TOSCA). http://tosca.cs.technion.ac.il, 2007.

[42] A. M. Bronstein, M. M. Bronstein, E. Gordon, et al. Fusion of 2D and 3D data in three-dimensional face recognition. In *ICIP*, pages 87–90. 2004.

[43] A. M. Bronstein, M. M. Bronstein, R. Kimmel. Expression-invariant 3D face recognition. In *AVBPA'03: Proceedings of the 4th international conference on Audio- and video-based biometric person authentication*, pages 62–70. Springer-Verlag, Berlin, Heidelberg, 2003. ISBN 3-540-40302-7.

[44] A. M. Bronstein, M. M. Bronstein, R. Kimmel. Expression-invariant face recognition via spherical embedding. In *ICIP (3)*, pages 756–759. 2005.

[45] A. M. Bronstein, M. M. Bronstein, R. Kimmel. Isometric embedding of facial surfaces in $\mathbb{S}^3$. In *In Proc. Int'l Conf. Scale Space and PDE Methods in Computer Vision, number 3459 in Lecture Notes on Computer Science*, pages 622–631. Springer-Verlag, 2005.

[46] A. M. Bronstein, M. M. Bronstein, R. Kimmel. Three-dimensional face recognition. *International Journal of Computer Vision*, volume 64(1):pages 5–30, 2005. ISSN 0920-5691.

[47] A. M. BRONSTEIN, M. M. BRONSTEIN, R. KIMMEL. Face2Face: An isometric model for facial animation. In F. J. P. LÓPEZ, R. B. FISHER (editors), *AMDO*, volume 4069 of *Lecture Notes in Computer Science*, pages 38–47. Springer, 2006. ISBN 3-540-36031-X.

[48] A. M. BRONSTEIN, M. M. BRONSTEIN, R. KIMMEL. Robust expression-invariant face recognition from partially missing data. In A. LEONARDIS, H. BISCHOF, A. PINZ (editors), *ECCV (3)*, volume 3953 of *Lecture Notes in Computer Science*, pages 396–408. Springer, 2006. ISBN 3-540-33836-5.

[49] A. M. BRONSTEIN, M. M. BRONSTEIN, R. KIMMEL. Expression-invariant representations of faces. *IEEE Transactions on Image Processing*, volume 16(1):pages 188–197, 2007.

[50] E. M. BRONSTEIN, M. M. BRONSTEIN, R. KIMMEL. Three-dimensional face recognition. *International Journal of Computer Vision*, volume 64:pages 5–30, 2005.

[51] E. M. BRONSTEIN, M. M. BRONSTEIN, A. SPIRA, ET AL. Face recognition from facial surface metric. In *in Proc. ECCV*, pages 225–237. Springer, 2004.

[52] K. CHANG, K. BOWYER, P. FLYNN. An evaluation of multimodal 2d+3d face biometrics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, volume 27(4):pages 619–624, 2005. ISSN 0162-8828.

[53] K. I. CHANG, K. W. BOWYER, P. J. FLYNN. Face recognition using 2d and 3d facial data. In *ACM Workshop on Multimodal User Authentication*, pages 25–32. 2003.

[54] K. I. CHANG, K. W. BOWYER, P. J. FLYNN. Multiple nose region matching for 3d face recognition under varying facial expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28:pages 1695–1700, 2006. ISSN 0162-8828.

[55] J. CHEN, Y. HAN. Shortest paths on a polyhedron. In *SCG '90: Proceedings of the sixth annual symposium on Computational geometry*, pages 360–369. ACM, New York, NY, USA, 1990. ISBN 0-89791-362-0.

[56] S. CHENG, V. STANKOVIC, L. STANKOVIC. Improved SIFT-based image registration using belief propagation. *ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2909–2912, 2009.

[57] G. CHRISTENSEN, R. RABBITT, M. MILLER. Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing*, volume 5(10):pages 1435–1447, October 1996. ISSN 1057-7149.

[58] H. CHUI. *Non-Rigid Point Matching: Algorithms, Extensions and Applications*. PhD thesis, Yale University, May 2001.

[59] T. COOTES, C. TAYLOR. Statistical models of appearance for computer vision, URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.1455, 2004.

[60] T. F. COOTES, G. J. EDWARDS, C. J. TAYLOR. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, pages 484–498. 1998.

[61] T. F. COOTES, G. J. EDWARDS, C. J. TAYLOR. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 23(6):pages 681–685, 2001. ISSN 0162-8828.

[62] T. F. COOTES, S. MARSLAND, C. J. TWINING, ET AL. Groupwise diffeomorphic non-rigid registration for automatic model building. In *Proceedings of the European Conference on Computer Vision*, pages 316–327. Springer, 2004.

[63] T. F. COOTES, C. J. TAYLOR. Combining point distribution models with shape models based on finite element analysis. In *Proc. of British Machine Vision Conf. (BMVC '94) (vol. 2)*, pages 419–428. BMVA Press, Surrey, UK, UK, 1994. ISBN 952-1898-1-X.

[64] T. F. COOTES, C. J. TAYLOR. An algorithm for tuning an active appearance model to new data. In *British Machine Vision Conference*, pages 919–928. 2006.

[65] T. F. COOTES, C. J. TAYLOR, D. H. COOPER, ET AL. Active shape models — their training and application. *Computer Vision and Image Understanding*, volume 61(1):pages 38–59, January 1995.

[66] T. F. COOTES, C. J. TWINING, V. PETROVIC, ET AL. Groupwise construction of appearance models using piece-wise affine deformations. In *Proceedings of BMVC05*, pages 879–888. 2005.

[67] T. F. COOTES, C. J. TWINING, V. S. PETROVIC, ET AL. Computing accurate correspondences across groups of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 32(11):pages 1994–2005, November 2010. ISSN 0162-8828.

[68] D. COSKER. *Animation of a Hierarchical Appearance Based Facial Model and Perceptual Analysis of Visual Speech.* PhD thesis, School of Computer Science, Cardiff University, July 2006.

[69] D. COSKER, E. KRUMHUBER, K. SIDOROV, ET AL. Discovering realistic facial dynamics for animation. In *3rd European Conference on Visual Media Production*, pages 173–173. November 2006. ISSN 0537-9989.

[70] S. COTIN, H. DELINGETTE, N. AYACHE. Real time volumetric deformable models for surgery simulation. In *Proceedings of the 4th International Conference on Visualization in Biomedical Computing*, pages 535–540. Springer-Verlag, London, UK, 1996. ISBN 3-540-61649-7.

[71] T. M. COVER, J. A. THOMAS. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 1991. ISBN 0471062596.

[72] T. F. COX, M. A. A. COX, T. F. COX. *Multidimensional Scaling, Second Edition.* Chapman & Hall/CRC, September 2000. ISBN 1584880945.

[73] H. S. M. COXETER. *Introduction to geometry.* Wiley, New York, 1963.

[74] D. CRISTINACCE, N. BUTCHER, T. COOTES. Facial motion analysis using clustered shortest path tree registration. In 1^{st} *International Workshop on Machine Learning for Vision-based Motion Analysis, part of* 10^{th} *European Conference on Computer Vision.* Marseille, France, 2008.

[75] B. CURLESS, M. LEVOY. A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on computer graphics and interactive techniques*, pages 303–312. ACM, New York, NY, USA, 1996. ISBN 0-89791-746-4.

[76] R. DAVIES, C. TWINING, C. TAYLOR. *Statistical Models of Shape: Optimisation and Evaluation.* Springer Publishing Company, Incorporated, 2008. ISBN 1848001371.

[77] R. H. DAVIES, C. J. TWINING, T. F. COOTES, ET AL. 3d statistical shape models using direct optimisation of description length. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 3–20. Springer-Verlag, London, UK, 2002. ISBN 3-540-43746-0.

[78] R. H. DAVIES, C. J. TWINING, T. F. COOTES, ET AL. A minimum description length approach to statistical shape modeling. *IEEE Transactions on Medical Imaging*, volume 21(5):pages 525–537, May 2002. ISSN 0278-0062.

[79] R. H. DAVIES, C. J. TWINING, T. F. COOTES, ET AL. Automatic construction of optimal statistical shape models. In *Proceedings of Australia-Japan Advanced Workshop on Computer Vision.* 2003.

[80] R. H. DAVIES, C. J. TWINING, C. TAYLOR. Groupwise surface correspondence by optimization: representation and regularization. *Medical Image Analysis*, volume 12(6):pages 787–796, 2008.

[81] R. H. DAVIES, C. J. TWINING, T. G. WILLIAMS, ET AL. Group-wise correspondence of surfaces using non-parametric regularisation and shape images. In *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro 2007*, pages 1208–1211. 2007.

[82] V. DE SILVA, J. B. TENENBAUM. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems*, volume 15, pages 705–712. 2003.

[83] G. DEBUNNE, M. DESBRUN, M.-P. CANI, ET AL. Dynamic real-time deformations using space & time adaptive sampling. In *Proc. of SIGGRAPH '01*, pages 31–36. ACM, New York, NY, USA, 2001. ISBN 1-58113-374-X.

[84] M. DESBRUN, M. GASCUEL. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Proc. of EG Workshop on Computer Animation and Simulation '96*, pages 61–76. Springer-Verlag, 1996.

[85] M. DESBRUN, M. MEYER, P. ALLIEZ. Intrinsic parameterizations of surface meshes. In *Eurographics 2002 Conference Proceedings*, volume 21, pages 209–218. 2002.

[86] M. DESBRUN, P. SCHRODER, D. P. SCHRODER, ET AL. Interactive animation of structured deformable objects. In *In Graphics Interface*, pages 1–8. 1999.

[87] E. W. DIJKSTRA. A note on two problems in connexion with graphs. *Numerische Mathematik*, volume 1(1):pages 269–271, December 1959. ISSN 0029-599X.

[88] M. P. DO-CARMO. *Differential Geometry of Curves and Surfaces.* Prentice Hall, 1st edition, February 1976. ISBN 0132125897.

[89] I. DRYDEN, K. MARDIA. *Statistical Shape Analysis.* John Wiley & Sons, 1998.

[90] J. DUCHON. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *Revue d'Automatique, d'Informatique et de Recherche Opérationnelle — Analyse numérique*, volume 10:pages 5–12, 1976.

[91] R. O. DUDA, P. E. HART, D. G. STORK. *Pattern Classification.* Wiley-Interscience, 2nd edition, November 2000. ISBN 0471056693.

[92] G. J. EDWARDS, T. F. COOTES, C. J. TAYLOR. Face recognition using Active Appearance Models. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision*, volume 2, pages 581–595. Springer-Verlag, London, UK, 1998. ISBN 3-540-64613-2.

[93] A. ELAD, R. KIMMEL. Bending invariant representations for surfaces. volume 1, pages I–168 – I–174 vol.1. 2001. ISSN 1063-6919.

[94] A. ELAD, R. KIMMEL. On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 25(10):pages 1285–1295, oct. 2003. ISSN 0162-8828.

[95] Y. ELDAR, M. LINDENBAUM, M. PORAT, ET AL. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, volume 6(9):pages 1305–1315, Sept 1997. ISSN 10577149.

[96] C. FALOUTSOS, K.-I. LIN. FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 163–174. ACM, New York, NY, USA, 1995. ISBN 0-89791-731-6.

[97] FGNET. Face and gesture network: Talking face database. http://www-prima.inrialpes.fr/FGnet/data /01-TalkingFace/talking_face.html.

[98] B. FISCHER, J. MODERSITZKI. Ill-posed medicine — an introduction to image registration. *Inverse Problems*, volume 24(3):page 034008, June 2008. ISSN 0266-5611.

[99] M. S. FLOATER. Parametrization and smooth approximation of surface triangulations. *Comput. Aided Geom. Des.*, volume 14(3):pages 231–250, 1997. ISSN 0167-8396.

[100] M. S. FLOATER, K. HORMANN, M. REIMERS. Parameterization of manifold triangulations. In C. K. CHUI, L. L. SCHUMAKER, J. STOCKLER (editors), *Approximation Theory X: Abstract and Classical Analysis, Innovations in Applied Mathematics*, pages 197–209. Vanderbilt University Press, Nashville, 2002.

[101] R. GAL, D. COHEN-OR. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.*, volume 25(1):pages 130–150, 2006. ISSN 0730-0301.

[102] X. GAO, Y. SU, X. LI, ET AL. A review of Active Appearance Models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, volume 40(2):pages 145–158, March 2010. ISSN 1094-6977.

[103] M. GARLAND. *Quadric-Based Polygonal Surface Simplification.* PhD thesis, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3891, May 1999.

[104] N. GELFAND, N. J. MITRA, L. J. GUIBAS, ET AL. Robust global registration. In *Symposium on Geometry Processing.* 2005.

[105] J. GIARD, B. MACQ. From mesh parameterization to geodesic distance estimation. In *25th European Workshop on Computational Geometry (EuroCG 2009).* Brussels, Belgium, 2009.

[106] S. F. F. GIBSON, B. MIRTICH. A Survey of Deformable Modeling in Computer Graphics. Technical report, 1997.

[107] M. GISSLER. Simulation and visualization of topolgy-changing plastic material. Technical report, CESCG, 2007.

[108] B. GLOCKER, N. KOMODAKIS, N. PARAGIOS, ET AL. Inter and intra-modal deformable registration: continuous deformations meet efficient optimal linear programming. In *IPMI'07: Proceedings of the 20th international conference on Information processing in medical imaging*, pages 408–420. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN 978-3-540-73272-3.

[109] D. E. GOLDBERG. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Professional, 1st edition, January 1989. ISBN 0201157675.

[110] J. GONZALEZ-MORA, F. DE LA TORRE, R. MURTHI, ET AL. Bilinear Active Appearance Models. In *IEEE 11th International Conference on Computer Vision (ICCV '2007)*, pages 1–8. October 2007. ISSN 1550-5499.

[111] A. GRAY, A. MOORE. Very fast multivariate kernel density estimation via computational geometry. In *Proc. of the Joint Statistical Meeting, San Francisco, CA*. 2003.

[112] H. GRAY, C. M. GOSS. *Anatomy of the human body.* Lea & Febiger, Philadelphia, 29th edition, 1973. ISBN 0812103777.

[113] K. GROCHOW, S. L. MARTIN, A. HERTZMANN, ET AL. Style-based inverse kinematics. *ACM Trans. Graph.*, volume 23(3):pages 522–531, 2004. ISSN 0730-0301.

[114] R. GROSS, I. MATTHEWS, S. BAKER. Constructing and fitting Active Appearance Models with occlusion. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshop on Face Processing in Video (FPIV '04)*, pages 72–72. June 2004.

[115] R. GROSS, I. MATTHEWS, S. BAKER. Generic vs. person specific Active Appearance Models. *Image and Vision Computing*, volume 23:pages 1080–1093, 2004.

[116] R. GROSS, I. MATTHEWS, S. BAKER. Active Appearance Models with occlusion. *Image and Vision Computing*, volume 24(1):pages 593–604, 2006.

[117] R. GROSSMANN, N. KIRYATI, R. KIMMEL. Computational surface flattening: a voxel-based approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, volume 24(4):pages 433–441, April 2002. ISSN 0162-8828.

[118] P. GROTHER. NIST special database 19 handprinted forms and characters database. Technical report, National Institute of Standards and Technology, March 1995.

[119] A. GUIMOND, J. MEUNIER, J.-P. THIRION. Automatic computation of average brain models. In *MICCAI '98: Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 631–640. Springer-Verlag, London, UK, 1998. ISBN 3-540-65136-5.

[120] X. GUO, H. QIN. Real-time meshless deformation: Collision detection and deformable objects. *Comput. Animat. Virtual Worlds*, volume 16(3-4):pages 189–200, 2005. ISSN 1546-4261.

[121] P. HALL, D. MARSHALL, R. MARTIN. Merging and splitting eigenspace models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 22(9):pages 1042–1049, 2000.

[122] P. HALL, D. MARSHALL, R. MARTIN. Adding and subtracting eigenspaces with EVD and SVD. In *Image and Vision Computing*, volume 20, pages 1009–1016. 2002.

[123] P. M. HALL, A. D. MARSHALL, R. R. MARTIN. Adding and subtracting eigenspaces. In T. P. PRIDMORE, D. ELLIMAN (editors), *Proceedings of the British Machine Vision Conference 1999*, pages 453–462. British Machine Vision Association, Nottingham, September 1999. ISBN 1-901725-09-X.

[124] R. HARTLEY, A. ZISSERMAN. *Multiple view geometry in computer vision.* Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-62304-9.

[125] A. HENRIQUES, B. WÜNSCHE, S. MARKS. An investigation of meshless deformation for fast soft tissue simulation in virtual surgery applications. *International Journal of Computer Assissted Radiology and Surgery*, volume 2:pages 69–171, 2008.

[126] Y. HICKS. *Modelling and Tracking of Articulated Human Motion.* Ph.D. thesis, School of Computer Science, Cardiff University, September 2003.

[127] N. J. HIGHAM. Computing the polar decomposition with applications. *SIAM Journal on Scientific and Statistical Computing*, volume 7(4):pages 1160–1174, 1986. ISSN 0196-5204.

[128] W. A. HOFF, K. NGUYEN, T. LYON. Computer vision-based registration techniques for augmented reality. In *Intelligent Robots and Computer Vision XV*, pages 538–548. 1996.

[129] A. HOSNI, M. BLEYER, M. GELAUTZ, ET AL. Local stereo matching using geodesic support weights. In *ICIP'09: Proceedings of the 16th IEEE international conference on Image processing*, pages 2069–2072. IEEE Press, Piscataway, NJ, USA, 2009. ISBN 978-1-4244-5653-6.

[130] Q.-X. HUANG, H. POTTMANN. Automatic and robust multi-view registration. Technical Report 152, Geometry Preprint Series, Vienna Univ.of Technology, December 2005.

[131] G. IRVING, J. TERAN, R. FEDKIW. Invertible finite elements for robust simulation of large deformation. In *SCA '04: Proc. of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 131–140. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2004. ISBN 3-905673-14-2.

[132] D. L. JAMES, D. K. PAI. ArtDefo: accurate real time deformable objects. In *Proc. of SIGGRAPH '99*, pages 65–72. ACM Press/Addison-Wesley, New York, NY, USA, 1999. ISBN 0-201-48560-5.

[133] H. JIANG, S. YU. Linear solution to scale and rotation invariant object matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2009*, pages 2474–2481. June 2009. ISSN 1063-6919.

[134] A. JOHNSON, M. HEBERT. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, volume 21(5):pages 433–449, May 1999. ISSN 0162-8828.

[135] H. JOHNSON, G. CHRISTENSEN. Consistent landmark and intensity-based image registration. *IEEE Transactions on Medical Imaging*, volume 21(5):pages 450–461, May 2002. ISSN 0278-0062.

[136] D. R. JONES, C. D. PERTTUNEN, B. E. STUCKMAN. Lipschitzian optimization without the lipschitz constant. *J. Optim. Theory Appl.*, volume 79(1):pages 157–181, 1993. ISSN 0022-3239.

[137] E. JONES, S. SOATTO. Layered active appearance models. In *10th IEEE International Conference on Computer Vision 2005*, volume 2, pages 1097–1102. October 2005. ISSN 1550-5499.

[138] K. KÄHLER, J. HABER, H. YAMAUCHI, ET AL. Head shop: generating animated head models with anatomical structure. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 55–63. ACM Press, New York, NY, USA, 2002. ISBN 1-58113-573-4.

[139] G. J. KATZ, J. T. KIDER, JR. All-pairs shortest-paths for large graphs on the GPU. In *GH '08: Proceedings of the 23rd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, pages 47–55. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2008. ISBN 978-3-905674-09-5.

[140] C. H. KAU, S. RICHMOND, A. ZHUROV, ET AL. Use of 3-dimensional surface acquisition to study facial morphology in 5 populations. *American Journal of Orthodontics & Dentofacial Orthopedics*, volume 137(4):pages S56–S57, April 2010.

[141] M. KAUS, V. PEKAR, C. LORENZ, ET AL. Automated 3D PDM construction from segmented images using deformable models. *IEEE Transactions on Medical Imaging*, volume 22(8):pages 1005–1013, 2003.

[142] R. KEISER, M. MÜLLER, B. HEIDELBERGER, ET AL. Contact handling for deformable point-based objects. In *Proc. Vision, Modeling, Visualization VMV'04*, pages 315–322. Stanford, California, 2004.

[143] J. R. KENT, W. E. CARLSON, R. E. PARENT. Shape transformation for polyhedral objects. *SIGGRAPH Comput. Graph.*, volume 26(2):pages 47–54, 1992. ISSN 0097-8930.

[144] R. KIMMEL, J. A. SETHIAN. Computing geodesic paths on manifolds. In *Proc. Natl. Acad. Sci. USA*, volume 95, pages 8431–8435. 1998.

[145] S. KIRKPATRICK, C. D. GELATT, M. P. VECCHI. Optimization by simulated annealing. *Science*, volume 220(4598):pages 671–680, 1983. ISSN 00368075.

[146] D. KIRSANOV. Exact geodesic for triangular meshes. http://www.mathworks.com/matlabcentral/fileexchange/18168-exact-geodesic-for-triangular-meshes, March 2008.

[147] D. KIRSANOV. Multiple source/target exact geodesic (shortest path) algorithm for triangular mesh (triangulated 2D surface in 3D). http://code.google.com/p/geodesic/, 2008.

[148] A. KLAUS, M. SORMANN, K. KARNER. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 15–18. IEEE Computer Society, Washington, DC, USA, 2006. ISBN 0-7695-2521-0.

[149] R. M. KOCH, M. H. GROSS, A. A. BOSSHARD. Emotion editing using finite elements. *Computer Graphics Forum*, volume 17(3):pages C295–C302, 1998.

[150] I. KOKKINOS, A. YUILLE. Unsupervised learning of object deformation models. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. 14-21 2007. ISSN 1550-5499.

[151] V. KOLMOGOROV, R. ZABIN. Computing visual correspondence with occlusions via graph cuts. In *International Conference on Computer Vision*, pages 508–515. 2001.

[152] V. KOLMOGOROV, R. ZABIN. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, volume 26(2):pages 147–159, feb. 2004. ISSN 0162-8828.

[153] Y. LAN, B. THEOBALD, R. HARVEY, ET AL. Improving visual features for lip-reading. In *Proceedings of the International Conference on Auditory-Visual Speech Processing*, pages 142–147. 2010.

[154] J. LASSETER. Principles of traditional animation applied to 3d computer animation. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 35–44. ACM, New York, NY, USA, 1987. ISBN 0-89791-227-6.

[155] N. D. LAWRENCE. Gaussian process latent variable models for visualisation of high dimensional data. In S. THRUN, L. SAUL, B. SCHÖLKOPF (editors), *Advances in Neural Information Processing Systems 16*, pages 329–336. MIT Press, Cambridge, MA, 2004.

[156] E. G. LEARNED-MILLER. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28:pages 236–250, 2006.

[157] S. LEE, G. WOLBERG, S. SHIN. Scattered data interpolation with multilevel B-splines. *IEEE Transactions on Visualization and Computer Graphics*, volume 3(3):pages 228–244, July–September 1997. ISSN 1077-2626.

[158] S.-Y. LEE, K.-Y. CHWA, S. Y. SHIN. Image metamorphosis using snakes and free-form deformations. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 439–448. ACM, New York, NY, USA, 1995. ISBN 0-89791-701-4.

[159] P. A. LEGG. *Multimodal Retinal Imaging: Improving Accuracy and Efficiency of Image Registration Using Mutual Information.* PhD thesis, School of Computer Science, Cardi University, Cardiff, UK, March 2010.

[160] A. D. D. LEOW, A. D. D. KLUNDER, C. R. R. JACK, ET AL. Longitudinal stability of MRI for mapping brain change using tensor-based morphometry. *Neuroimage*, volume 31(2):pages 627–640, February 2006. ISSN 1053-8119.

[161] M. LEVOY, K. PULLI, B. CURLESS, ET AL. The digital Michelangelo project: 3D scanning of large statues. In *SIGGRAPH '00: Proceedings of the 27th annual conference on computer graphics and interactive techniques*, pages 131–144. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000. ISBN 1-58113-208-5.

[162] J. P. LEWIS. Fast normalized cross-correlation. In *Vision Interface*, pages 120–123. Canadian Image Processing and Pattern Recognition Society, 1995.

[163] T. LEYVAND. Advanced topics in computer graphics. http://leyvand.com/research/adv-graphics/advgraphics-ex1.ppt, 2005.

[164] H. LI, T. SHEN, X. HUANG. Global optimization for alignment of generalized shapes. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2009*, pages 856–863. June 2009. ISSN 1063-6919.

[165] X. LI, I. GUSKOV. Multi-scale features for approximate alignment of point-based surfaces. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, pages 217–228. Eurographics Association, Aire-la-Ville, Switzerland, 2005. ISBN 3-905673-24-X.

[166] N. LINIAL, E. LONDON, Y. RABINOVICH. The geometry of graphs and some of its algorithmic applications. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 577–591. November 1994.

[167] G. R. LIU. *Mesh Free Methods: Moving Beyond the Finite Element Method.* CRC, 1st edition, July 2002. ISBN 0849312388.

[168] A. LORUSSO, D. W. EGGERT, R. B. FISHER. A comparison of four algorithms for estimating 3-D rigid transformations. In *Proc. of British Machine Vision Conf. (BMVC '95) (Vol. 1)*, pages 237–246. BMVA Press, Surrey, UK, UK, 1995. ISBN 0-9521898-2-8.

[169] J. LÖTJÖNEN, T. MÄKELÄ. Elastic matching using a deformation sphere. In W. J. NIESSEN, M. A. VIERGEVER (editors), *MICCAI*, volume 2208 of *Lecture Notes in Computer Science*, pages 541–548. Springer, 2001. ISBN 3-540-42697-3.

[170] L. J. P. VAN DER MAATEN, E. O. POSTMA, H. J. VAN DEN HERIK. Dimensionality reduction: a comparative review, URL: http://www.cs.unimaas.nl/l.vandermaaten/dr/DR_draft.pdf. Published online, 2007.

[171] J. MacCormick, M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 3–19. Springer-Verlag, London, UK, 2000. ISBN 3-540-67686-4.

[172] F. Maes, A. Collignon, D. V, et al. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, volume 16:pages 187–198, 1997.

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

[174] S. Marsland, C. Twining. Constructing diffeomorphic representations for the groupwise analysis of nonrigid registrations of medical images. *IEEE Transactions on Medical Imaging*, volume 23(8):pages 1006–1020, August 2004. ISSN 0278-0062.

[175] S. Marsland, C. J. Twining, C. J. Taylor. Groupwise non-rigid registration using polyharmonic clamped-plate splines. In R. E. Ellis, T. M. Peters (editors), *MICCAI (2)*, volume 2879 of *Lecture Notes in Computer Science*, pages 771–779. Springer, 2003. ISBN 3-540-20464-4.

[176] S. Marsland, C. J. Twining, C. J. Taylor. A minimum description length objective function for groupwise non-rigid image registration. *Image and Vision Computing*, volume 26(3):pages 333–346, 2008. ISSN 0262-8856.

[177] J. Martin, A. Pentland, R. Kikinis. Shape analysis of brain structures using physical and experimental modes. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1994*, pages 752–755. June 1994.

[178] J. L. Maryak, D. C. Chin. Global random optimization by simultaneous perturbation stochastic approximation. *IEEE Transacrions on Automatic Control*, volume 53(3):pages 780–783, 2008.

[179] W. S. Massey. *A Basic Course in Algebraic Topology*. Springer-Verlag, New York, 1997. ISBN 038797430X.

[180] I. Matthews, T. Cootes, J. Bangham, et al. Extraction of visual features for lipreading. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, volume 24(2):pages 198–213, February 2002. ISSN 0162-8828.

[181] G. Meisters, C. Olech. Locally one-to-one mappings and a classical theorem on Schlicht functions. *Duke Mathematical Journal*, volume 30:pages 63–80, 1963.

[182] E. Miller, N. Matsakis, P. Viola. Learning from one example through shared densities on transforms. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2000*, volume 1:pages 464–471, 2000.

[183] J. S. B. MITCHELL, D. M. MOUNT, C. H. PAPADIMITRIOU. The discrete geodesic problem. *SIAM J. Comput.*, volume 16(4):pages 647–668, August 1987. ISSN 0097-5397.

[184] P. MOON, D. E. SPENCER. *Field Theory Handbook: Including Coordinate Systems, Differential Equations and Their Solutions.* Springer-Verlag, Berlin, 2nd edition, 1988.

[185] K. MORAVEC, R. HARVEY, J. BANGHARN. Scale trees for stereo vision. *Vision, Image and Signal Processing, IEE Proceedings*, volume 147(4):pages 363–370, August 2000. ISSN 1350-245X.

[186] M. MÜLLER, M. GROSS. Interactive virtual materials. In *GI '04: Proc. of Graphics Interface 2004*, pages 239–246. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. ISBN 1-56881-227-2.

[187] M. MÜLLER, B. HEIDELBERGER, M. TESCHNER, ET AL. Meshless deformations based on shape matching. *ACM Trans. Graph.*, volume 24(3):pages 471–478, 2005. ISSN 0730-0301.

[188] M. MÜLLER, R. KEISER, A. NEALEN, ET AL. Point based animation of elastic, plastic and melting objects. In *SCA '04: Proc. of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 141–151. Eurographics Association, Aire-la-Ville, Switzerland, 2004. ISBN 3-905673-14-2.

[189] A. MYRONENKO, X. B. SONG. Image registration by minimization of residual complexity. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2009*, pages 49–56. IEEE, 2009. ISBN 978-1-4244-3992-8.

[190] A. NEALEN, M. MUELLER, R. KEISER, ET AL. Physically based deformable models in computer graphics. *Computer Graphics Forum*, volume 25(4):pages 809–836, December 2006. ISSN 0167-7055.

[191] A. NETRAVALI, J. ROBBINS. Motion-compensated television coding: some new results. *Bell System Tech.*, volume 59(11):pages 1735–1745, November 1980.

[192] E. NOIRFALISE, J. LAPREST, F. JURIE, ET AL. Real-time registration for image mosaicing. In *Proceedings of The 13th British Mashine Vision Conference*, pages 617–625. 2002.

[193] M. M. NORDSTRØM, M. LARSEN, J. SIERAKOWSKI, ET AL. The IMM face database — an annotated dataset of 240 face images. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, May 2004.

[194] T. O'Donnell, S. Aharon, A. Gupta, et al. Multi-modality model-based registration in the cardiac domain. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2000*, volume 2:pages 790–791, 2000. ISSN 1063-6919.

[195] A. S. Ogale, Y. Aloimonos. Shape and the stereo correspondence problem. *International Journal of Computer Vision*, volume 65(3):pages 147–162, 2005. ISSN 0920-5691.

[196] B. O'Neil. *Elementary Differential Geometry.* Academic Press, New York, NY, USA, 1966.

[197] M. Pantic. *Encyclopedia of Multimedia Technology and Networking*, volume 1, chapter "Face for Interface", pages 308–314. IGI Publishing, Hershey, PA, May 2005. ISBN 9781605660141.

[198] H. J. Park, M. Kubicki, M. E. Shenton, et al. Spatial normalization of diffusion tensor MRI using multiple channels. *Neuroimage*, volume 20(4):pages 1995–2009, 2003.

[199] S. I. Park, J. K. Hodgins. Capturing and animating skin deformation in human motion. *ACM Trans. Graph.*, volume 25(3):pages 881–889, 2006. ISSN 0730-0301.

[200] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, volume 2(6):pages 559–572, 1901.

[201] M. Péchaud, I. Vanzetta, T. Deneux, et al. SIFT-based sequence registration and flow-based cortical vessel segmentation applied to high resolution optical imaging data. In *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro 2008*, pages 720–723. May 2008.

[202] A. Pentland, S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 13(7):pages 715–729, 1991. ISSN 0162-8828.

[203] A. Pentland, J. Williams. Good vibrations: modal dynamics for graphics and animation. *SIGGRAPH Comput. Graph.*, volume 23(3):pages 207–214, 1989. ISSN 0097-8930.

[204] P. Pérez, M. Gangnet, A. Blake. Poisson image editing. *ACM Transactions on Graphics (SIGGRAPH '03)*, volume 22(3):pages 313–318, 2003. ISSN 0730-0301.

[205] V. Petrovic, T. Cootes, A. Mills, et al. Automated analysis of deformable structure in groups of images. In *Proc. of British Machine Vision Conf. (BMVC '07)*, pages 302–311. 2007.

[206] G. Peyré. Toolbox Fast Marching. http://www.mathworks.com/matlabcentral/fileexchange/6110, 2009.

[207] G. PEYRÉ, L. COHEN. Geodesic remeshing using front propagation. In *International Journal of Computer Vision*, pages 33–40. 2003.

[208] G. PEYRÉ, L. COHEN. Geodesic computations for fast and accurate surface flattening. In *Eurographics*, pages 1–10. 2004.

[209] G. PEYRÉ, L. D. COHEN. Geodesic remeshing using front propagation. *International Journal of Computer Vision*, volume 69(1):pages 145–156, 2006. ISSN 0920-5691.

[210] G. PICINBONO, H. DELINGETTE, N. AYACHE. Non-Linear Anisotropic Elasticity for Real-Time Surgery Simulation. Research Report RR-4028, INRIA, 2000.

[211] G. PICINBONO, H. DELINGETTE, N. AYACHE. Non-linear and anisotropic elastic soft tissue models for medical simulation. In *ICRA2001: IEEE International Conference Robotics and Automation*. Seoul, Korea, 2001.

[212] J. C. PLATT. Fastmap, metricmap, and landmark mds are all nystrom algorithms. In *In Proceedings of 10th International Workshop on Artificial Intelligence and Statistics*, pages 261–268. 2005.

[213] J. P. W. PLUIM, J. B. A. MAINTZ, M. A. VIERGEVER. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, volume 22(8):pages 986–1004, July 2003.

[214] C. POHL, J. L. VAN GENDEREN. Multisensor image fusion in remote sensing: concepts, methods and applications. *International Journal of Remote Sensing*, volume 19(5):pages 823–854, March 1998.

[215] H. POPAT, E. HENLEY, S. RICHMOND, ET AL. A comparison of the reproducibility of verbal and nonverbal facial gestures using three-dimensional motion analysis. *Otolaryngolohgy —Head and Neck Surgery*, volume 142(6):pages 867–872, June 2010.

[216] H. POPAT, S. RICHMOND. New developments in: three-dimensional planning for orthognathic surgery. *Journal of Orthodontics*, volume 37(1):pages 62–71, March 2010.

[217] E. PRAUN, A. FINKELSTEIN, H. HOPPE. Lapped textures. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 465–470. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000. ISBN 1-58113-208-5.

[218] E. PRAUN, H. HOPPE, A. FINKELSTEIN. Robust mesh watermarking. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 49–56. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1999. ISBN 0-201-48560-5.

[219] F. P. PREPARATA, S. J. HONG. Convex hulls of finite sets of points in two and three dimensions. *Commun. ACM*, volume 20(2):pages 87–93, 1977. ISSN 0001-0782.

[220] W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, ET AL. *Numerical Recipes 3rd Edition: The Art of Scientific Computing.* Cambridge University Press, New York, NY, USA, 2007. ISBN 0521880688.

[221] R. J. RENKA. Multivariate interpolation of large sets of scattered data. *ACM Trans. Math. Softw.*, volume 14(2):pages 139–148, 1988. ISSN 0098-3500.

[222] J. D. M. RENNIE. Derivation of the F-measure. http://people.csail.mit.edu/jrennie/writing+, February 2004.

[223] J. RISSANEN. *Stochastic Complexity in Statistical Inquiry Theory.* World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1989. ISBN 9971508591.

[224] A. R. RIVERS, D. L. JAMES. FastLSM: fast lattice shape matching for robust real-time deformation. *ACM Trans. Graph.*, volume 26(3):pages 82:1–82:6, 2007.

[225] S. M. ROSS. *Introduction to Probability and Statistics for Engineers and Scientists.* John Wiley & Sons, June 1987. ISBN 047181752X.

[226] S. ROWEIS. EM algorithms for PCA and SPCA. In *in Advances in Neural Information Processing Systems*, pages 626–632. MIT Press, 1998.

[227] D. RUECKERT, L. SONODA, C. HAYES, ET AL. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging*, volume 18(8):pages 712–721, August 1999. ISSN 0278-0062.

[228] O. SAMKO. *Low Dimension Hierarchical Subspace Modelling of High Dimensional Data.* PhD thesis, School of Computer Science, Cardiff University, Cardiff, UK, June 2009.

[229] P. V. SANDER, J. SNYDER, S. J. GORTLER, ET AL. Texture mapping progressive meshes. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 409–416. ACM, New York, NY, USA, 2001. ISBN 1-58113-374-X.

[230] D. SCHARSTEIN, R. SZELISKI. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, volume 47(1-3):pages 7–42, 2002. ISSN 0920-5691.

[231] K. SCHERBAUM. *Learning Based Prediction and 3D-Visualisation of ChildrenTs Facial Growth.* Master's thesis, Hochschule der Medien, Stuttgart, 2005.

[232] B. SCHÖLKOPF. The kernel trick for distances. In T. K. LEEN, T. G. DIETTERICH, V. TRESP (editors), *Advances in Neural Information Processing Systems 13*, pages 301–307. MIT Press, Cambridge, MA, USA, 2000.

[233] E. SCHWARTZ, A. SHAW, E. WOLFSON. A numerical solution to the generalized mapmaker's problem: flattening nonconvex polyhedral surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, volume 11(9):pages 1005–1008, September 1989. ISSN 0162-8828.

[234] N. SEBE, M. LEW, D. HUIJSMANS. Toward improved ranking metrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 22(10):pages 1132–1143, October 2000. ISSN 0162-8828.

[235] G. A. F. SEBER. *Multivariate Observations*. John Wiley and Sons, 2004. ISBN 0471691216.

[236] J. A. SETHIAN. A fast marching level set method for monotonically advancing fronts. In *Proc. Nat. Acad. Sci*, pages 1591–1595. 1995.

[237] J. A. SETHIAN. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, volume 93(4):pages 1591–1595, 1996.

[238] J. A. SETHIAN. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, June 1999. ISBN 0521642043.

[239] C. E. SHANNON. A mathematical theory of communication. *Bell system technical journal*, volume 27, 1948.

[240] G. SHARP, S. LEE, D. WEHE. Icp registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24(1):pages 90–102, January 2002. ISSN 0162-8828.

[241] K. SIDOROV, D. MARSHALL, S. RICHMOND. *Three-Dimensional Imaging for Orthodontics and Maxillofacial Surgery*, chapter 18: Nonrigid Image Registration Using Groupwise Methods, pages 290–304. Blackwell Publishing Ltd, 2010. ISBN 9781405162401.

[242] K. A. SIDOROV, A. D. MARSHALL, P. L. ROSIN, ET AL. Towards efficient 3D facial appearance models. In D. METAX, J. POPOVIC (editors), *ACM SIGGRAPH Symposium on Computer Animation*. 2007.

[243] K. A. SIDOROV, S. RICHMOND, D. MARSHALL. An efficient stochastic approach to groupwise non-rigid image registration. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 2208–2213. IEEE Computer Society, Los Alamitos, CA, USA, 2009. ISBN 978-1-4244-3992-8.

[244] K. A. SIDOROV, S. RICHMOND, D. MARSHALL. Efficient groupwise non-rigid registration of textured surfaces. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2011)*. IEEE Computer Society, Los Alamitos, CA, USA, 2011 (accepted, to appear).

[245] E. SIFAKIS, R. FEDKIW. Facial muscle activations from motion capture. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, page 1195. IEEE Computer Society, Washington, DC, USA, 2005. ISBN 0-7695-2372-2.

[246] E. SIFAKIS, A. SELLE, A. ROBINSON-MOSHER, ET AL. Simulating speech with a physics-based facial muscle model. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 261–270. Eurographics Association, Aire-la-Ville, Switzerland, 2006. ISBN 3-905673-34-7.

[247] L. SIROVICH, M. KIRBY. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, volume 4(3):pages 519–524, March 1987.

[248] J. SPALL, S. HILL, D. STARK. Theoretical framework for comparing several stochastic optimization approaches. In *Probabilistic and Randomized Methods for Design under Uncertainty*, pages 99–117. Springer, 2006.

[249] J. C. SPALL. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Trans. on Aerospace Electronic Systems*, volume 34:pages 817–823, July 1998.

[250] J. C. SPALL. Overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, volume 19(4):pages 482–492, 1998.

[251] H. SPÄTH. *Two Dimensional Spline Interpolation Algorithms*. A. K. Peters, Ltd., Natick, MA, USA, 1995. ISBN 1-56881-017-2.

[252] R. SPRENGEL, K. ROHR, H. STIEHL. Thin-plate spline approximation for image registration. volume 3, pages 1190–1191. October 1996.

[253] D. STEINEMANN, M. A. OTADUY, M. GROSS. Fast adaptive shape matching deformations. In *SCA '08: Proc. of the ACM SIGGRAPH/EG Symp. on Computer Animation '08*, pages 87–94. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2008. ISBN 978-3-905674-10-1.

[254] C. STUDHOLME, D. L. G. HILL, D. J. HAWKES. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, volume 32(1):pages 71–86, 1999.

[255] J. SÜBAMUTH, M. WINTER, G. GREINER. Reconstructing animated meshes from time-varying point clouds. *Computer Graphics Forum*, volume 27(5):pages 1469–1476, 2008.

[256] R. W. SUMNER, M. ZWICKER, C. GOTSMAN, ET AL. Mesh-based inverse kinematics. *ACM Trans. Graph.*, volume 24(3):pages 488–495, 2005. ISSN 0730-0301.

[257] J. SUN, N.-N. ZHENG, H.-Y. SHUM. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 25:pages 787–800, 2003. ISSN 0162-8828.

[258] V. SURAZHSKY, T. SURAZHSKY, D. KIRSANOV, ET AL. Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.*, volume 24(3):pages 553–560, 2005. ISSN 0730-0301.

[259] R. SZELISKI. *Handbook of Mathematical Models in Computer Vision*, chapter "Image alignment and stitching", pages 273–292. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387263713.

[260] J. O. TALTON. A short survey of mesh simplification algorithms. Technical report, University of Illinois at Urbana-Champaign, 2004.

[261] G. K. L. TAM, Z. Q. CHENG, Y. LAI, ET AL. From rigid to non-rigid registration of 3D point clouds and meshes. In *Eurographics*. 2010.

[262] J. TERAN, S. BLEMKER, V. N. T. HING, ET AL. Finite volume methods for the simulation of skeletal muscle. In *SCA '03: Proc. of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 68–74. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2003. ISBN 1-58113-659-5.

[263] J. TERAN, E. SIFAKIS, G. IRVING, ET AL. Robust quasistatic finite elements and flesh simulation. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 181–190. ACM Press, New York, NY, USA, 2005. ISBN 1-7695-2270-X.

[264] D. TERZOPOULOS, J. PLATT, A. BARR, ET AL. Elastically deformable models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 205–214. ACM, New York, NY, USA, 1987. ISBN 0-89791-227-6.

[265] D. TERZOPOULOS, J. PLATT, A. BARR, ET AL. Elastically deformable models. In *Proc. of SIGGRAPH '87*, pages 205–214. ACM, New York, NY, USA, 1987. ISBN 0-89791-227-6.

[266] M. TESCHNER, S. GIROD, B. GIROD. Optimization approaches for soft-tissue prediction in craniofacial surgery simulation. In *Medical Image Computing and Computer-Assisted Intervention MICCAI'99*, pages 1183–1190. 1999.

[267] M. TESCHNER, S. GIROD, B. GIROD. Direct computation of nonlinear soft-tissue deformation. In *VMV'00*, pages 383–390. 2000.

[268] M. TESCHNER, B. HEIDELBERGER, M. MULLER, ET AL. A versatile and robust model for geometrically complex deformable solids. In *CGI '04: Proceedings of the Computer Graphics International (CGI'04)*, pages 312–319. IEEE Computer Society, Washington, DC, USA, 2004. ISBN 0-7695-2171-1.

[269] M. TESCHNER, S. KIMMERLE, B. HEIDELBERGER, ET AL. Collision detection for deformable objects. *Computer Graphics Forum*, volume 24:pages 119–140, 2005.

[270] B. TIDDEMAN, N. DUFFY, G. RABEY. A general method for overlap control in image warping. *Computers & Graphics*, volume 25(1):pages 59–66, 2001.

[271] S. TOELG, T. POGGIO. Towards an example-based image compression architecture for video-conferencing. Technical report, MIT Artificial Intelligence Laboratory, Cambridge, MA, USA, 1994.

[272] D. L. TONNESEN. *Dynamically coupled particle systems for geometric modeling, reconstruction, and animation.* Ph.D. thesis, Department of Computer Science, University of Toronto, Toronto, Ont., Canada, Canada, 1998. Adviser-Terzopoulos, Demetri.

[273] J. TSITSIKLIS. Efficient algorithms for globally optimal trajectories. *Automatic Control, IEEE Transactions on*, volume 40(9):pages 1528–1538, September 1995. ISSN 0018-9286.

[274] M. TURK, A. PENTLAND. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, volume 3(1):pages 71–86, 1991. ISSN 0898-929X.

[275] M. TURK, A. PENTLAND. Face recognition using eigenfaces. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1991*, pages 586–591. June 1991.

[276] C. TWINING, S. MARSLAND, C. TAYLOR. Measuring geodesic distances on the space of bounded diffeomorphisms. In *British Machine Vision Conference (BMVC 2002)*, pages 847–856. BMVA Press, 2002.

[277] C. TWINING, S. MARSLAND, C. TAYLOR. Groupwise non-rigid registration of medical images: the Minimum Description Length approach. In *Medical Image Analysis and Understanding*, pages 81–84. 2004.

[278] C. J. TWINING, T. COOTES, S. MARSL, ET AL. A unified information-theoretic approach to groupwise non-rigid registration and model building. In *In Proceedings of the International Conference on Pattern Recognition (ICPR*, pages 1–14. Springer, 2005.

[279] C. J. TWINING, S. MARSLAND. Groupwise non-rigid registration: the Minimum Description Length approach. In *Proceedings of British Machine Vision Conference (BMVC '04)*, pages 417–426. 2004.

[280] J. ULYSSES, A. CONCI. Measuring similarity in medical registration. In F. R. LETA, A. CONCI (editors), *Proceedings of IWSSIP 2010*, pages 558–561. EdUFF, June 2010. ISBN 978-85-228-0565-5.

[281] O. VASICEK. A test for normality based on sample entropy. *Journal of the Royal Statistical Society B*, volume 38:pages 54–59, 1976.

[282] M. VASILESCU, D. TERZOPOULOS. Multilinear image analysis for facial recognition. In *Proceedings of 16th International Conference on Pattern Recognition 2002*, volume 2, pages 511–514. 2002. ISSN 1051-4651.

[283] M. VASILESCU, D. TERZOPOULOS. Multilinear subspace analysis of image ensembles. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003*, volume 2:pages 93–99, June 2003. ISSN 1063-6919.

[284] M. VASILESCU, D. TERZOPOULOS. Multilinear independent components analysis. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2005*, volume 1, pages 547–553. June 2005. ISSN 1063-6919.

[285] M. VASILESCU, D. TERZOPOULOS. Multilinear projection for appearance-based recognition in the tensor framework. In *IEEE 11th International Conference on Computer Vision 2007*, pages 1–8. October 2007. ISSN 1550-5499.

[286] M. A. O. VASILESCU, D. TERZOPOULOS. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the European Conference on Computer Vision*, pages 447–460. 2002.

[287] P. VIOLA, W. M. WELLS, III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, volume 24(2):pages 137–154, 1997. ISSN 0920-5691.

[288] K. WALLI, H. RHODY. Automated image registration to 3-D scene models. In *37th IEEE Applied Imagery Pattern Recognition Workshop 2008*, pages 1–8. October 2008. ISSN 1550-5219.

[289] M. WAND, B. ADAMS, M. OVSJANIKOV, ET AL. Efficient reconstruction of non-rigid shape and motion from real-time 3D scanner data. *ACM Trans. on Graphics*, volume 28(2):pages 15:1–15, 2009.

[290] M. WAND, P. JENKE, Q. HUANG, ET AL. Reconstruction of deforming geometry from time-varying point clouds. In A. G. BELYAEV, M. GARLAND (editors), *Symposium on Geometry Processing*, volume 257 of *ACM International Conference Proceeding Series*, pages 49–58. Eurographics Association, 2007. ISBN 978-3-905673-46-3.

[291] J. T. WANG, X. WANG, K. IP LIN, ET AL. Evaluating a class of distance-mapping algorithms for data mining and clustering. In *Knowledge Discovery and Data Mining*, pages 307–311. ACM Press, 1999.

[292] K. WATERS. A muscle model for animation three-dimensional facial expression. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 17–24. ACM, New York, NY, USA, 1987. ISBN 0-89791-227-6.

[293] H. WENDLAND. *Scattered Data Approximation.* Number 17 in Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, UK, 2004. ISBN 0521843359.

[294] C. WILKINSON, C. RYNN, H. PETERS, ET AL. A blind accuracy assessment of computer-modeled forensic facial reconstruction using computed tomography data from live subjects. *Forensic Science, Medicine, and Pathology*, volume 2:pages 179–187, 2006. ISSN 1547-769X.

[295] D. WOLPERT, W. MACREADY. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, volume 1(1):pages 67–82, April 1997. ISSN 1089-778X.

[296] M. H. YANG. Kernel Eigenfaces vs. Kernel Fisherfaces: face recognition using kernel methods. In *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 215–220. IEEE Computer Society, Washington, DC, USA, May 2002. ISBN 0-7695-1602-5.

[297] Q. YANG, L. WANG, R. YANG, ET AL. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 31(3):pages 492–504, 2009. ISSN 0162-8828.

[298] S. YOSHIZAWA, A. BELYAEV, H.-P. SEIDEL. A fast and simple stretch-minimizing mesh parameterization. In *SMI '04: Proceedings of the Shape Modeling International 2004*, pages 200–208. IEEE Computer Society, Washington, DC, USA, 2004. ISBN 0-7695-2075-8.

[299] G. YOUNG, A. HOUSEHOLDER. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, volume 3(1):pages 19–22, March 1938.

[300] Q. ZHANG. Geometry-driven photorealistic facial expression synthesis. *IEEE Transactions on Visualization and Computer Graphics*, volume 12(1):pages 48–60, 2006. ISSN 1077-2626.

[301] W. ZHAO, S. GAO, H. LIN. A robust hole-filling algorithm for triangular mesh. *Vis. Comput.*, volume 23(12):pages 987–997, 2007. ISSN 0178-2789.

[302] G. ZIGELMAN, R. KIMMEL, N. KIRYATI. Texture mapping using surface flattening via multidimensional scaling. *IEEE Trans. on Visualization and Computer Graphics*, volume 8(2):pages 198–207, 2002. ISSN 1077-2626.

[303] B. ZITOVA, J. FLUSSER. Image registration methods: a survey. *Image and Vision Computing*, volume 21(11):pages 977–1000, October 2003.

[304] Б. Н. Делоне. Sur la sphère vide. *Известия Академии Наук СССР. Отделение математических и естественных наук*, volume 7:pages 793–800, 1934.

[305] В. Н. Кублановская. О некоторых алгорифмах для решения полной проблемы собственных значений. *Журнал вычислительной математики и математической физики*, volume 1(4):pages 555–570, 1961.

[306] А. В. Скворцов. *Триангуляция Делоне и её применение.* Томск: Издательство Томского университета, Томск, 2002. ISBN 5-7511-1501-5.

Colophon

This thesis was typeset using the LaTeX typesetting system created by Leslie Lamport, using the `memoir` class. The text was edited entirely with `Vim 7` using the `Vim-LaTeX` suite. Figures exported from MATLAB were prepared using `exportfig`, `laprint`, and `matlabfrag` packages. Manually drawn figures were created with `ipe`.