

3D TOOTH SURFACE RECONSTRUCTION

STÉPHANIE ISABELLE BUCHAILLARD

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgment

I would like to express my sincere appreciation and gratitude to my supervisors, Assoc. Prof. Ong Sim Heng, as well as Assoc. Prof. Kelvin W. C. Foong, for their advice, guidance and assistance throughout the course of this project.

I am especially grateful to Dr. Yohan Payan from TIMC-GMCAO laboratory for his kind assistance and for providing ideas for this project.

I would also like to thank Mr. Francis Hoon from the Vision and Image Processing laboratory for his assistance during the entire course of this research.

Last but not least, I would like to extend my appreciation to all those who have helped me one way or another during this project.

Stéphanie I. Buchaillard

Table of Contents

Acknowledgment	i
Table of Contents	ii
Summary	v
Glossary	vii
List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Motivation	1
1.2 Aim of the thesis	4
1.3 Organization of the thesis	6
2 Literature survey	7
2.1 Definitions of important terms	7
2.1.1 Rigid-body transformation	7
2.1.2 Nonrigid-body transformation	8
2.2 2D/3D Registration Schemes	9
2.3 Common Deformable Models	10
2.4 A statistical approach	12
3 Preparatory work: Data collection	14
3.1 Teeth collection	14
3.1.1 The human dentition	14
3.1.2 Scanning equipment	15
3.2 Point to point correspondence	17
3.3 Extraction of the crown/root of a tooth	20

4	Deformable Models	21
4.1	Theoretical aspects	21
4.1.1	Introduction to principal component analysis	21
4.1.2	Statistical model of Cootes <i>et al.</i>	24
4.2	Properties of the defined model	26
4.2.1	Creation of the statistical model	26
4.2.2	Influence of the modes	29
4.3	Definition of crown/root parameters	33
5	Registration process	34
5.1	Rigid Registration	34
5.2	Elastic Registration: Optimizing the Modes' Weights	36
5.2.1	Choice of the optimization scheme	38
5.2.2	Optimization using a distance map	40
5.2.3	Optimization without distance map	41
5.3	Generation of a 3D mesh	42
5.4	Framework	42
6	Experiments and Results	44
6.1	Experiments	44
6.1.1	Tests performed	44
6.1.2	Error measure	46
6.2	Influence of the number of modes	47
6.3	Leave-one-out tests	48
6.3.1	Tooth reconstruction based on the crown only	49
6.3.2	Tooth reconstruction based on the root only	52
6.3.3	Discussion	55
6.4	Reconstruction using patient's data	58
6.5	Reconstruction using feature points	60
6.6	Influence of the number of specimens used to define the statistical model	61
6.7	Computation time	62
6.8	Effect of the target's density	64
6.9	Summary	65
7	Conclusion	67

A The Levenberg-Marquardt algorithm	69
A.1 Function approximation	69
A.2 Gradient and Hessian of χ^2	70
A.3 Levenberg-Marquardt strategy	71
B Computing a distance map	73
B.1 The octree-spline decomposition	73
B.1.1 The octree decomposition	73
B.1.2 Computation of the distance map	74
B.2 Finding the neighbors of an octant	76
C Kd-Tree Decomposition	78
C.1 Structure and building of a Kd-tree	78
C.2 Nearest Neighbor Search	79
D Leave-one-out tests: numerical results	82
References	86

Summary

Many maxillofacial surgery applications as well as forensic medicine often require an accurate knowledge of the teeth 3D shape. However, this information may not be available if the tooth is broken or when we are working with a dental cast. Furthermore, current methods of generating 3D images, such as computer tomography (CT) imaging, are radiologically invasive and not always justified. In these cases, an alternative method is necessary to approximate the shape.

Herein, we describe a method to generate the 3D representation of a tooth using partial information about its shape (e.g., the crown or the root only). The information required consists of a cloud of points representing the available part. Data information can be obtained, for instance, by segmenting a dental cast. The shape is then defined using a statistical model that is constituted of a mean shape and a set of modes. A statistical model is necessary for each kind of tooth: this study was conducted using upper right second premolars (single rooted teeth). To that end, a set of 22 teeth was digitized using a micro-CT scanner and used throughout this research.

The reconstruction was then performed defining the optimal registration between the mean shape and the specimens to reconstruct and optimizing the statistical model parameters. This method allows us to generate the global shape fast, thanks to the small number of parameters to adjust, and requires little or no interaction from the user. Furthermore, by constraining the possible deformations of the statistical model, we can prevent the final shape instance to vary too much from the typical shape of the reconstructed tooth.

Different experiments were conducted to investigate the validity of the method. Leave-one-

out tests were performed to test the capacity of this approach to reconstruct a tooth shape given partial information. Other tests were realized using patient's data, adding feature points or investigating the influence of parameters such as the number of modes or the density of the tooth to reconstruct.

During these experiments, we found that the shape reconstruction process produced satisfactory results when we generated the tooth using crown information: the shape of the teeth (original tooth and its reconstructed version) were similar, as well as their height. The method proposed gave only a coarse approximation of the tooth shape when dealing with root information, but could be slightly improved by the introduction of feature points.

Glossary

Buccal Lateral surfaces of side teeth, opposite to the tongue.

Distal Surface of a tooth in contact with another one, closer to the back of the mouth.

Endodontics A dental specialty concerned with the maintenance of the dental pulp in a state of health and the treatment of the pulp cavity (pulp chamber and pulp canal).

ICP Iterative Closest Point Algorithm (Chapter 5).

Malocclusion Poor positioning or inappropriate contact between the teeth on closure.

Mesial Surface of a tooth in contact with another one, closer to the center of the mouth.

Occlusal The chewing or grinding surface of the premolar and molar teeth.

Orthodontics The dental specialty and practice of preventing and correcting irregularities of the teeth, as by the use of braces.

Pantomogram A panoramic radiographic record of the maxillary and mandibular dental arches and their associated structures, obtained by a pantomograph.

PCA Principal Component Analysis (Chapter 4).

PDM Point Distribution Model (Chapter 4).

Pose estimation Process of determining the position and orientation of an object with respect to a coordinate system.

Registration Process of defining the correct alignment between two elements of identical or different modalities (Chapter 2).

Root canal Channel used by the blood vessels and nerves to reach the pulp cavity.

List of Tables

1.1	Most common imaging modalities in the medical field.	3
3.1	Specifications of SkyScan-1076 micro-CT system.	16
3.2	Precision of the point to point correspondence.	19
4.1	Eigenvalues of the training set's autocorrelation matrix.	27
4.2	Maximum and minimum values of the modes weights for the training set.	30
6.1	Number of tests realized given the number of specimens N	62
6.2	Average complexity of the elastic registration for the two algorithms.	63
D.1	Leave-one-out test on the specimens' set using the crowns only (method KD) . . .	82
D.2	Leave-one-out test on the specimens' set using the crowns only (method OD) . . .	83
D.3	Leave-one-out test on the specimens' set using the roots only (method KD) . . .	84
D.4	Leave-one-out test on the specimens' set using the roots only (method OD) . . .	85

List of Figures

1.1	Dental cast.	2
1.2	Fitting a tooth on a dental cast.	5
3.1	Adult dental set: the four quadrants.	15
3.2	SkyScan-1076 high-resolution in-vivo micro-CT system.	16
3.3	Micro-CT slices and the corresponding tooth after reconstruction.	17
3.4	Generic model and elastic registration onto a tooth exemplar.	19
3.5	Interface used to define the crown or the root of a tooth.	20
3.6	A 3D image of a tooth, its crown and its root.	20
4.1	Example of dimensionality reduction using a PCA.	24
4.2	Eigenvalues of the training set's autocorrelation matrix.	28
4.3	Percentage of variability for the first 8 modes of deformation.	28
4.4	Cumulative percentage of variability given the number of modes.	29
4.5	Influence of the first mode of decomposition on the mean shape.	31
4.6	Influence of the second mode of decomposition on the mean shape.	31
4.7	Influence of the third mode of decomposition on the mean shape.	32
4.8	Influence of the fourth mode of decomposition on the mean shape.	32
5.1	Automatic rigid registration using the ICP algorithm.	37
5.2	Global tooth shape recovery process.	43
6.1	Framework for the pose verification.	48
6.2	Influence of the number of modes on the reconstruction using the crown.	49

6.3	Influence of the number of modes on the reconstruction using the root.	49
6.4	Results of the leave-one-out test using crown information (method KD).	50
6.5	Examples of teeth reconstructed using crown information only (method KD). . .	50
6.6	Results of the leave-one-out test using crown information (method OD).	51
6.7	Examples of teeth reconstructed using crown information only (method OD) . . .	51
6.8	Results of the leave-one-out test using root information (method KD).	53
6.9	Examples of teeth reconstructed using root information only (method KD). . . .	53
6.10	Results of the leave-one-out test using root information (method OD).	54
6.11	Examples of teeth reconstructed using root information only (method OD). . . .	54
6.12	Distribution of the HD (reconstruction based on crown data)	55
6.13	Distribution of the HD (reconstruction based on root data).	57
6.14	Process of matching two roots	57
6.15	The segmentation of teeth from a dental cast affects the crown shape.	58
6.16	Process of fitting a tooth on a dental cast.	59
6.17	Location of the feature points for a reconstruction using root information.	60
6.18	Effect of adding feature points on the reconstruction.	61
6.19	Influence of the number of specimens used to build the statistical model.	62
6.20	Computation time of the two methods.	64
6.21	Influence of the density on the reconstruction accuracy.	64
B.1	Tree Representation.	74
B.2	Example of octree decomposition applied on a tooth root.	74
C.1	An example of space subdivision in dimension 2 using a Kd-Tree ($d = 2$).	79
C.2	Example of a Kd-Tree decomposition ($d = 2$).	79

Chapter 1

Introduction

1.1 Motivation

Having an accurate knowledge of the 3D shape of a tooth and the position of the tooth root is very important in most maxillofacial surgery applications, endodontic procedures, malocclusion problems, and treatment simulations. Currently, the shape of a tooth in the mouth is represented in two-dimension by an x-ray film. As teeth are 3D structures with complex shapes, an accurate 3D representation of teeth shape is vital in facilitating clinical treatment. For example, orthodontists try to reposition the teeth of a patient in case of malocclusion using usually brackets¹ attached to them and connected by an arch wire. In this case, the exact location, orientation and shape of the teeth are necessary to plan appropriate movements during the treatment. Having a good knowledge of the shape of a given tooth is also extremely helpful to create implants of good quality. Dental implants are metal anchors that act as tooth root substitutes. They can be inserted into the jawbone when teeth are missing and then fused with the bone. They have been used very successfully for the past 20 years and dentistry has come to rely on them for solving problems that were formerly insolvable. Their utility is not only aesthetic. Indeed, without the root, the bone around the missing tooth would gradually recede. Currently, the shape of an implant is usually

¹Small attachments that are bonded directly to the tooth surface using a special adhesive.

defined using an x-ray of the missing tooth area to get an approximate shape of the implant using the neighboring teeth.

3D information about the tooth shape is unfortunately not always available. Table 1.1 summarizes the most common medical imaging modalities. Computer tomography (CT) is an efficient way of generating 3D objects than can be used in some of the above-mentioned cases. However, CT imaging of dental patients is not always possible and is radiologically invasive. Rather than working with CT data, orthodontists regularly work with dental casts, also known as study models (Fig 1.1). Plaster-like materials are used to impress both upper and lower arches to create a replica of the patient's set of teeth. The mould that is obtained is then used to create the dental cast. These models, whose creation is totally safe for the patient, are used to prepare treatment plans and to make accurate measurements. However, they only provide information about the crowns of the teeth, but no information about the roots, which are hidden under the gum.



Figure 1.1: Dental cast.

Table 1.1: Most common imaging modalities in the medical field.

Modality	X-ray ^a	Ultrasound (US)	Computed Tomography (CT)	Magnetic Resonance Imaging (MRI)	Nuclear Medicine Imaging ^b (NMI)	Video
Dimension	2D	2D / 2.5D	2.5D	2.5D	2D	2D (intern use) 2D/2.5D (extern use)
Radiations	Low	No	Yes (1 to 10 mSv)	No	$\leq \frac{1}{2}$ the radiation dose usually used in CT.	No
Resolution	Very good resolution.	Low penetration, poor resolution, distortion, noise.	Excellent resolution.	Similar to CT.	Poor spatial resolution.	Intern: Poor resolution, invasive and difficult to manipulate. Extern: good resolution, easy to use.
Cost	Traditional, cheap, widely available.	Cheap, easy to use.	Expensive, slow.	Very expensive, complex, slow.	Very expensive and time-consuming.	Cheap.
Use	Excellent for bones and metal. Fluoroscopy for intraoperative navigation.	Obstetric use, scanning of organs as liver, kidneys or thyroid gland.	Image a wide variety of body structures and internal organs (sinus and brain studies, abdominal organs).	Excellent resolution for soft tissue, skeleton, joint, spine, brain and head.	Used for anatomic and functional information.	Intern: laparoscopy, endoscopy. Extern: e.g., tooth crown or tongue.

^aFilm x-ray, digital x-ray, fluoroscopy, digital subtraction angiography (DSA)

^bPositron emission tomography (PET), single photon emission tomography (SPECT)

Information concerning teeth shapes are also required in forensic medicine, e.g., for identification purposes [1]. Indeed, teeth are the most durable portion of the body and have the ability to resist erosion, deterioration and fire. They must be exposed to a temperature of over 500°C to be reduced to ashes and demonstrate a variety of form and varied conditions of wear, trauma, disease, and professional manipulation. However, only fragments of teeth are sometimes available,

and the missing part needs to be obtained by guesswork.

Therefore, to overcome the lack of original 3D data, alternative methods are necessary to define the tooth shape when the information is simply missing, or to avoid using radiologically invasive methods that are not absolutely justified.

1.2 Aim of the thesis

The different cases presented in section 1.1 require a good knowledge of the 3D representation of teeth. The aim of this research is to develop a method that allows orthodontists to define the 3D shape of a tooth using only partial information (e.g., points corresponding to the crown area), and without the use of x-rays, CT or MRI (magnetic resonance imaging). Here are some possible applications:

- Implant creation: Using the crown of the corresponding mirror tooth could help us determine the shape of the missing tooth and thus create an implant of higher quality.
- Forensic medicine: We may only have fragments of a tooth at our disposal (person identification using for example fragments found at a crash site, in criminal investigations or after an attack). Given the crown or root only, we could try defining the tooth shape using all the a priori knowledge for the tooth to be reconstructed.
- Tooth fitting on a dental cast: we could try to fit a model onto the dental cast teeth to get root information (Fig 1.2).

Different methods allow orthodontists to get highly accurate crown models: the more widespread method consists in a surface scanning of the dental casts. Another possibility lies in direct imaging methods using an intra-oral camera [2] (OraScanner from OraMetrix GmbH, ALVC1600 from Digital Doc). The major advantages of intra-oral camera are that they are non-radiative and highly accurate. Once a three-dimensional crown model has been built using any one of these

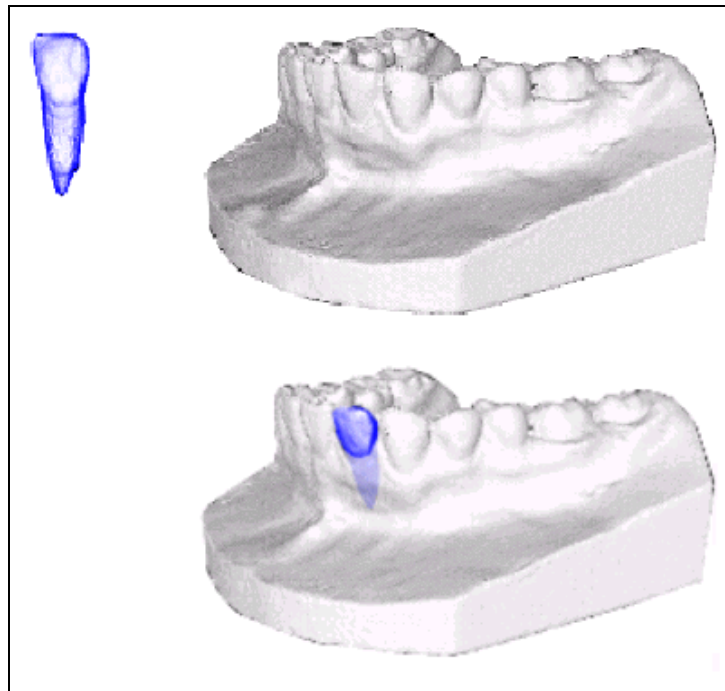


Figure 1.2: The blue tooth (top left hand corner) is fitted on the dental cast in order to define the position of the root.

possibilities, we still need a method to reconstruct the missing part of the tooth.

The approach presented here is based on the use of a statistical model (a special form of a deformable model). For a given kind of tooth, we need to create this model using a set of teeth belonging to the same family. Its possible deformations (called modes of deformations) have to represent the possible deformations of all the teeth belonging to the same family. After aligning the patient's tooth with the model (using translations and/or rotations) and adjusting the different parameters of this model, we hope to determine the shape of the tooth.

This method will allow us to generate the tooth rapidly due to the small number of parameters, and requires little or no interaction from the user. Furthermore, by constraining the possible deformations of the statistical model, we can prevent the final shape instance to vary too much from the typical shape of the reconstructed tooth.

1.3 Organization of the thesis

The outline of this thesis is as follows:

Chapter 2 - After an introduction on the notion of registration, we present a short review to highlight the different methods currently used in the registration area and underline their advantages and drawbacks with respect to the method proposed here.

Chapter 3 - We briefly describe the equipment and software used to construct the specimens, and the method of extracting the different elements for the reconstruction.

Chapter 4 - We define the notion of deformable model that is the central element of this research: we focus on the theoretical aspects as well as the properties of the model created.

Chapter 5 - We describe the algorithm to retrieve the shape of a tooth: two different approaches are introduced here to optimize the parameters of our model.

Chapter 6 - We present the different experiments that were conducted. A criterion is first defined to judge of the quality of this method, and the results are summarized (reconstruction using either the root or the crown only). The results are then discussed.

Chapter 7 - We present concluding remarks, their implications as well as suggestions for future work.

Appendices - In the appendices, the reader will find detailed information about the algorithms used in the implementation of our method: Levenberg-Marquardt algorithm (Appendix A), octree-spline decomposition (Appendix B) and Kd-Tree decomposition (Appendix C), as well as numerical results of the experiments conducted (Appendix D).

Chapter 2

Literature survey

2.1 Definitions of important terms

In this research, we aim at deforming a model, called the mean shape, into a target model (the patient's data). The nature of the transformation will be explained in Chapter 4. At this point, we first introduce the notion of registration as well as its major characteristics. Given two coordinate systems, Ref_A and Ref_B and sets of points $X_A = \{\mathbf{x}_{A_1}, \dots, \mathbf{x}_{A_N}\}$ and $X_B = \{\mathbf{x}_{B_1}, \dots, \mathbf{x}_{B_N}\}$ associated with homologous features in the two coordinate systems, the general goal of registration is to determine the geometrical transformation T such that $X_A = X_B$. In other words, we wish to align points in one view of an object with the corresponding points in another view of another object [3]. Since the two features are usually not perfectly identical, we try to obtain a transformation T that minimizes the distance between the two features; the distance function can be the maximum of minimum distances between the points, the root mean squared sum of minimum distances or any other appropriate metric.

2.1.1 Rigid-body transformation

An image coordinate transformation is called *rigid* when only translations and rotations are allowed. A general rigid-body transformation can be expressed as combination of two transforma-

tions, a rotation \mathbf{R}_{AB} and a translation \mathbf{t}_{AB} , i.e.,

$$\mathbf{x}_{B_i} = \mathbf{R}_{AB}\mathbf{x}_{A_i} + \mathbf{t}_{AB} \quad (2.1)$$

Consequently, rigid-body registration typically seeks the values of \mathbf{R}_{AB} and \mathbf{t}_{AB} that minimize

$$\min_{\mathbf{R}_{AB}, \mathbf{t}_{AB}} \sum_{i=1}^N \|\mathbf{x}_{B_i} - (\mathbf{R}_{AB}\mathbf{x}_{A_i} + \mathbf{t}_{AB})\|^2 \quad (2.2)$$

given 3D correspondences \mathbf{x}_{A_i} and \mathbf{x}_{B_i} .

A possibility is to use an orthonormal matrix representation, which can be viewed as a mapping reference from frame A to frame B once the translation between their origins has been compensated for. To represent rotations, quaternions¹ and dual quaternions² are also widespread. Contrary to methods that employ orthonormal matrices and quaternions, which first determine the optimal orientation and then use this solution to obtain the translation, the dual quaternion technique solves for both relative orientation and position by minimizing a cost function.

2.1.2 Nonrigid-body transformation

When the deformation between the two objects is noticeable (on account of factors, other than noise or distortion), we need to apply a *nonrigid-body* transformation. The first and simplest class of functions corresponds to affine³ transformation. Contrary to Eq. 2.1, the affine transformation does not respect angles and lengths. Another formulation consists of mapping surface A to surface B thanks to a global polynomial function. A transformation is called global if it applies to the entire image and local if subsections of the image each have their own transformations.

Generally, the order of these functions does not exceed 5.

¹A quaternion can be viewed as a number composed of a real part and three imaginary parts, or as a 3D-vector in \mathbb{R}^3 and a scalar in \mathbb{R}

²A dual quaternion is simply a pair of quaternions.

³If a transformation maps parallel lines onto parallel lines, it is called *affine*.

A Cartesian formulation of the coefficients of a global polynomial transformation is:

$$\left\{ \begin{array}{l} x_A = \sum_{i,j,k} a_{ijk} x_A^i y_A^j z_A^k \\ y_A = \sum_{i,j,k} b_{ijk} x_A^i y_A^j z_A^k \\ z_A = \sum_{i,j,k} c_{ijk} x_A^i y_A^j z_A^k \end{array} \right. \quad (2.3)$$

where a_{ijk} , b_{ijk} and c_{ijk} are the coefficients to be determined.

To realize a more general mapping, another possibility consists of using piecewise polynomial functions, or splines, to interpolate or approximate the motion between the two images. The approximation can be highly accurate when the number of points is very high and/or when the noise is significant. If this is not the case, we can interpolate (the transformation has to match exactly for the control points). Two widespread interpolation schemes are the thin-plate spline⁴ and the B-spline⁵. The first is useful when points are sparse and give better results for deformation over a 2D rather than over a 3D domain. The second is very useful in the context of smoothing and least-squares spline approximation.

2.2 2D/3D Registration Schemes

Registration processes are not only limited to 2D/2D or 3D/3D correspondences, but can also be used with objects of different modalities.

Enciso *et al.* [4] propose a 3D reconstruction based on 2D patient radiographs. Using thin-plate splines, they aim at producing a “best fit” patient-specific 3D geometric polygonal mesh of a tooth. The user needs to define manually a series of landmarks both on the patient’s x-ray and on a 2D view of the deformable model (the landmarks need to be located on the contour of both images). Thin-plate splines are then used to minimize a bending energy. Lapeer and Prager [5]

⁴The thin-plate spline is the two-dimensional analog of the cubic spline in one dimension. Thin-plate splines are used for surface interpolation over scattered data. They can be interpreted as a linear combination of a plane with a smooth surface.

⁵A B-spline is a generalization of the Bézier curve.

propose a similar scheme using not one but two or more orthogonal atlas images to reconstruct a newborn skull. They aim at warping an adult skull into the newborn skull by defining landmarks on the two skulls and then minimizing a bending energy based on a thin-plate spline function.

These methods are simple and easy to use and allow retrieving the third dimension that is missing. However, they only exploit 2D data. Although interesting for specimens in good condition, its extension to broken teeth does not seem possible. Indeed, for the method to work, the user needs to place the landmarks such that they represent globally the entire shape of the specimen, and not only a small part of it (e.g., crown or root). Furthermore, these methods do not exploit the possible shape variations of a given specimen, and a 2D correspondence (sufficient for some applications) does not necessarily imply a 3D one.

2.3 Common Deformable Models

A popular ND/ND registration scheme ($N = 2$ or 3) consists in warping a first volume (called a deformable model) into a second one. We present below different methods currently used to realize this kind of warping.

The notion of *active contour model* (also called *snake*) was introduced by Kass, Witkin, and Terzopoulos in [6]. It is an energy minimizing spline; the snake's energy depends on both its shape and its location within an image (2D or 3D). The energy function is a weighted combination of internal and external forces:

$$E_{snake} = \int_0^1 (E_{intern} \mathbf{v}(s) + E_{image} \mathbf{v}(s) + E_{constraint} \mathbf{v}(s)) ds \quad (2.4)$$

where $E_{intern} \equiv$ internal spline energy, E_{image} measures the attraction of the snake toward the image features, $E_{constraint}$ measures the external constraints and $\mathbf{v}(s)$ represents the parametric form of the target. Snakes are very powerful tools for image segmentation. However, since they can take any shape, they could lead to objects far from those expected. In the case of partial

forms (broken teeth, partial knowledge about the shape), they would be of little use and could lead to unexpected forms.

Another popular deformable scheme is based on *finite elements* (FE). An extension to 3D images was realized by Cohen and Cohen [7]. Similar to snakes, this method is an energy minimization procedure. The finite element method is applied to elements (usually triangular) that minimize a surface energy function. In their article, Cohen *et al.* propose a way of minimizing the energy of 3D balloons. 3D balloons are surface FE models. Applying the Euler-Lagrange equation to the surface and applying finite differences in space leads to an equation of the type

$$\mathbf{AV} = \mathbf{F} \quad (2.5)$$

where \mathbf{A} is the stiffness matrix, and \mathbf{V} and \mathbf{F} are respectively the vectors of positions and forces at the nodes of the 3D mesh. The different FE matching processes differ essentially in the energy function used to compare the different bodies as well as the matching criterion selected (for example minimization of the sum of the squared differences).

Free-form deformation is considered as a volumic deformation. First introduced by Sederberg and Parry [8], this technique involves the trivariate tensor product Bernstein polynomial. The model to deform is first enclosed in a bounding box. A local coordinate system $(\mathbf{s}, \mathbf{t}, \mathbf{u})$ centered at \mathbf{X}_0 is then superimposed on this bounding box and a grid of control points \mathbf{P}_{ijk} defined (with $0 \leq i \leq l, 0 \leq j \leq m, 0 \leq k \leq n$) by

$$\mathbf{P}_{ijk} = \mathbf{X}_0 + \frac{i}{l} \mathbf{s} + \frac{j}{m} \mathbf{t} + \frac{k}{n} \mathbf{u} \quad (2.6)$$

The deformations applied to the model enclosed are linked to the displacement of the points \mathbf{P}_{ijk} . Moving the points deforms the bounding box and consequently the model. Any point \mathbf{X} belonging to the box is transformed to \mathbf{X}_{ffd} after computing its local coordinates in the system $(\mathbf{s}, \mathbf{t}, \mathbf{u})$ and applying the deformation function (trivariate tensor product Bernstein polynomial).

The coefficients of the Bernstein polynomial are the control points \mathbf{P}_{ijk} . This method has the advantage of being applicable either globally or locally (respecting some continuity conditions) and to produce smooth surfaces, though it can lead rapidly to a high number of coefficients $((l + 1)(m + 1)(n + 1))$.

2.4 A statistical approach

The methods described in Sections 2.2 and 2.3 have proven to be powerful tools to reconstruct 3D shapes using sparse data. However, they all require a good knowledge of the target global shape. The major advantage of the technique we propose to use (method introduced by Fleute *et al.* [9]) is to offer the possibility to reconstruct an object given only partial information about its shape.

This method allows us to exploit the possible deformations of the target, contrary to the above mentioned methods. We use a *statistical model* (presented in Chapter 4) whose variations are based on the variations observed on a set of teeth belonging to the same family. The global idea of this method is to first align the two models using a manual or automatic rigid registration. Once the volumes are aligned, we can vary the statistical model parameters to warp the mean shape into the target. The parameters are determined by minimizing the sum of squared distances between the statistical model and the target.

The efficiency of this algorithm is due partly to the small number of parameters to optimize, as explained in Chapter 4. Furthermore, by constraining the parameters of the statistical model, we can prevent the reconstructed specimen to vary too much from the average shape, that would be impossible using one of the deformable schemes from Section 2.3. The main drawback is the amount of data necessary as well as the time required to construct such a model. However, this model needs to be built only once.

Fleute *et al.*'s method intends to reconstruct the upper part of a tibia during anterior cruciate

ligament reconstruction (ACL) surgery. The extension of this method to teeth can be improved by bringing simple modifications during the reconstruction:

- The data used during this research had a much higher resolution than those used by Fleute *et al.*. This difference had major implications on the definition of the modes of deformation as well as the evaluation of the function to minimize.
- The original objective function to minimize was also modified to allow us to use more powerful optimization tools for the matching and to better avoid local minima during the minimization process.
- Sub-models of the statistical model (crown or root) were also created to speed up the computation and improve the quality of the results.
- Finally, feature points were introduced for a better control of the reconstruction during shape recovery using root information.

Other minor modifications were added to get a reconstruction as fast as possible, despite the higher resolution of the models to reconstruct.

Chapter 3

Preparatory work: Data collection

3.1 Teeth collection

3.1.1 The human dentition

In order to create a statistical model, we need a set of teeth of the same kind. Due to the time required to build 3D models and the large number of specimens necessary, it was decided to test the method on a single type of tooth. Upper right second premolars were chosen (see Figure 3.1). Canines, incisors and premolars are single-rooted teeth; consequently, the matching process is easier than for molars, which can have two, three or four roots.

The selected teeth had to be complete (i.e., they should include the root and have no cavities) and present little or no sign of decay. Furthermore, the set of teeth should account for, as far as possible, as much of the variability we can have among upper right second premolars. Twenty-two teeth were collected, with heights varying from 18 mm to 24 mm.

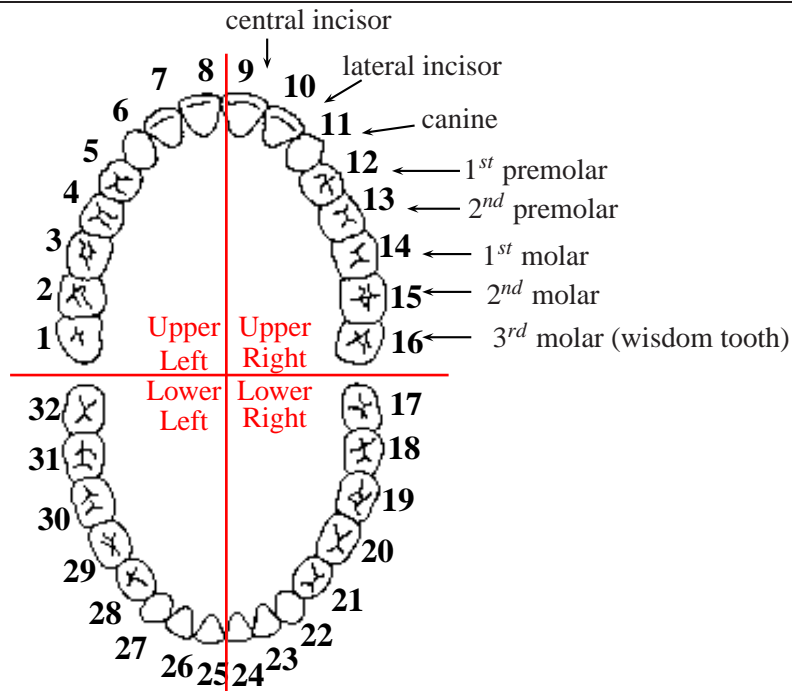


Figure 3.1: Adult dental set: the entire dentition is divided into into four quadrants, each quadrant containing 8 teeth.

3.1.2 Scanning equipment

Each tooth of the set was scanned at a resolution of $35\ \mu\text{m}$ with a SkyScan-1076 (Figure 3.2).

The specifications of the SkyScan-1076 micro-CT system are listed in Table 3.1.

The contour of every tooth was obtained manually on each CT slice using a commercial software (Analyze®). The construction of the deformable model requires to have the outer surface of the tooth only. An automated boundary detection would be less accurate and would let appear the root canal. For each slice, the boundaries are detected using a region growing method controlled by the user, who can choose the number of boundaries per slice and set the gray level limit separating the inside from the outside of the volume. Figure 3.3 shows, on the left, 3 slices obtained from micro-CT with the contours extracted in red. These contours are used to reconstruct the 3D surface (represented on the right). The blue lines indicate the position of the slices on the 3D surface.

The reconstructed tooth models had up to 72,000 points and 150,000 triangles. To simplify



Figure 3.2: SkyScan-1076 high-resolution in-vivo micro-CT system.

Table 3.1: Specifications of SkyScan-1076 micro-CT system.

SYSTEM	SkyScan 1076 In-vivo scanner
Maximum object size	68 mm or 35 mm diameter, 200 mm length
X-ray source	20-100 kV, 10 W, < 5 μm spot size (@4W), air cooled sealed type
X-ray detector	10 Megapixel (4000 x 2300 x 12 bit) cooled digital x-ray camera with fibre-optic coupling to scintillator
Reconstruction arrays	1000 x 1000 to 8000 x 8000 cross-section format (isotropic grid), 9 μm / 18 μm / 35 μm pixel size in any place of the scanning area.
Detail detectability	9-14 μm (contrast dependent)
Scanning system	Source-detector pair rotation with 0.02 deg. min. step size, 50 μm object positioning accuracy with 400 mm travel, 50 mm camera positioning with 1 μm accuracy, <10 microns overall stability during scanning

subsequent processing, a decimation process using VTK was applied to reduce the resolution to 105 μm ¹.

¹The reader can find useful information about decimation using VTK in “The Mesh Decimation Using VTK” from Michael Knapp, http://www.cg.tuwien.ac.at/courses/Seminar/SS2002/Knapp_paper.pdf and [10] or on the VTK website <http://public.kitware.com/VTK/>.

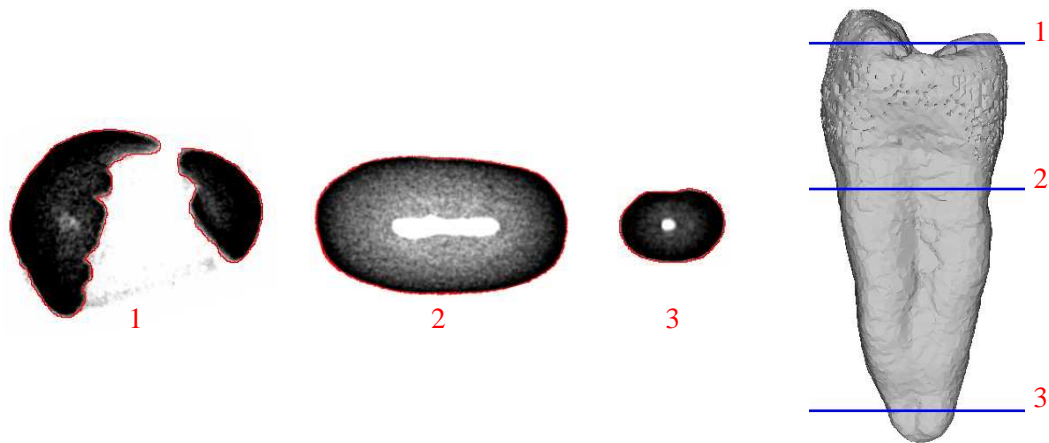


Figure 3.3: Micro-CT slices and the corresponding tooth after reconstruction.

3.2 Point to point correspondence

The creation of the deformable model requires a one to one correspondence between the digitized teeth. After reconstruction, each specimen is in the form of an unorganized cloud of points associated with a triangular mesh. To be able to construct a mean shape, one needs to know the relation between the points of the different specimens. A tooth that is representative of the kind of teeth used in this study is defined as the generic model. The digitized model of this tooth (a 3D triangular mesh of M points) is then deformed to match every other tooth collected. It means the i^{th} point of a model A can now be associated with the i^{th} point of a second model B (for example the points representing the extremity of the root).

This approach [11]² requires a manual rigid registration process to align each training tooth with the generic one. In addition to the rotation and translation of the target, a scaling can be performed if necessary. Then, the two volumes are enclosed in a bounding box and the generic model is used as a deformable model. A non-rigid registration using free-form deformation (FFD) (as described in Section 2.3) and splines is performed to deform the generic model to warp each specimen. Figure 3.4a shows the generic tooth model. Figure 3.4b shows one exemple of the 3D

²The software used was made available by the TIMC-GMCAO laboratory (“Techniques de l’Imagerie, de la Modélisation et de la Cognition - Gestes Médico-Chirurgicaux Assistés par ordinateur”), 38706 La Tronche, France, <http://www-timc.imag.fr/gmcao/>.

tooth models reconstructed from microtomography and Figure 3.4c represents the same tooth before and after the non-rigid registration. Iterating the process 2 or 3 times can improve the results (the values of the FFD's control points are stored after each iteration and used as initial values at the beginning of the next iteration). Each final model (i.e., after matching) is composed of M points ($M \approx 16,000$) and can be represented by a vector $\mathbf{m} = (x_0, y_0, z_0, \dots, x_{M-1}, y_{M-1}, z_{M-1})$. For the different specimens, the mean distance and the maximum distance between the surface of the 3D tooth models reconstructed from microtomography and the surface of the generic model after matching vary as summarized in Table 3.2. On average, the mean distance equals 0.12 mm and the maximum mean distance 1.07 mm (0.24 mm and 1.71 mm for the specimen in Fig. 3.4). Table 3.2 shows that the matching process leads globally to satisfactory results. However, when the form of the generic model is far from those of the target (e.g., the root is highly curved or the tooth is much bigger), some discrepancies can appear.

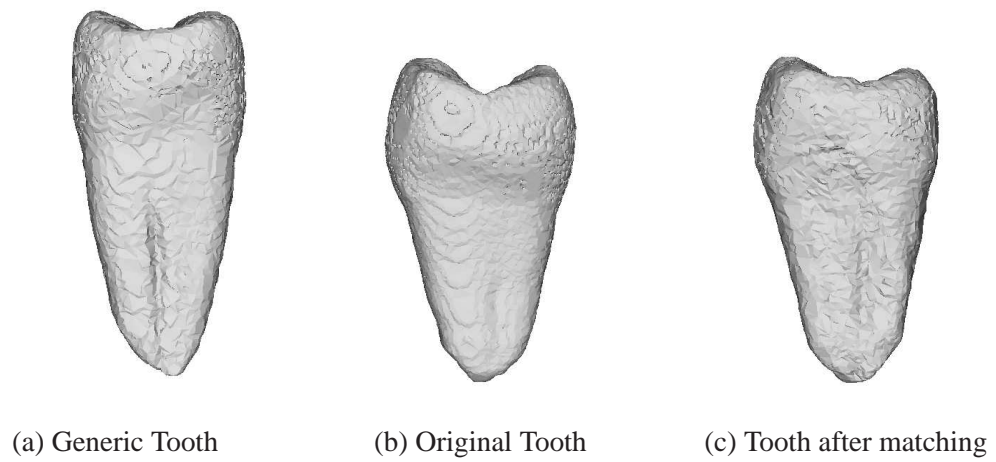


Figure 3.4: Generic model and elastic registration onto a tooth exemplar.

Table 3.2: Precision of the point to point correspondence.

Specimen	Mean dist. (mm)	Max dist. (mm)
1	0.14	1.21
2	0.15	1.18
3	0.09	0.63
4	0.09	1.07
5	0.22	2.61
6	0.24	1.71
7	0.15	1.22
8	0.07	0.55
9	0.13	0.98
10	0.12	0.81
11	0.14	0.58
12	0.08	0.91
13	0.08	0.75
14	0.06	0.55
15	0.08	0.82
16	0.08	0.96
17	0.08	0.71
18	0.10	0.67
19	0.07	1.53
20	0.07	1.23
21	0.10	1.02
22	0.26	1.86

3.3 Extraction of the crown/root of a tooth

In order to validate the methods, different reconstructions were made using either the root or crown of the specimens collected. The crown and root of the mean shapes need also to be defined to create sub-statistical models. To specify the crown or the root region of a tooth, the user needs to translate and rotate a plane that represents the limit between the crown and the root areas. Figure 3.5 shows the GUI designed to realize the extraction and Figure 3.6 the results obtained for a given tooth.

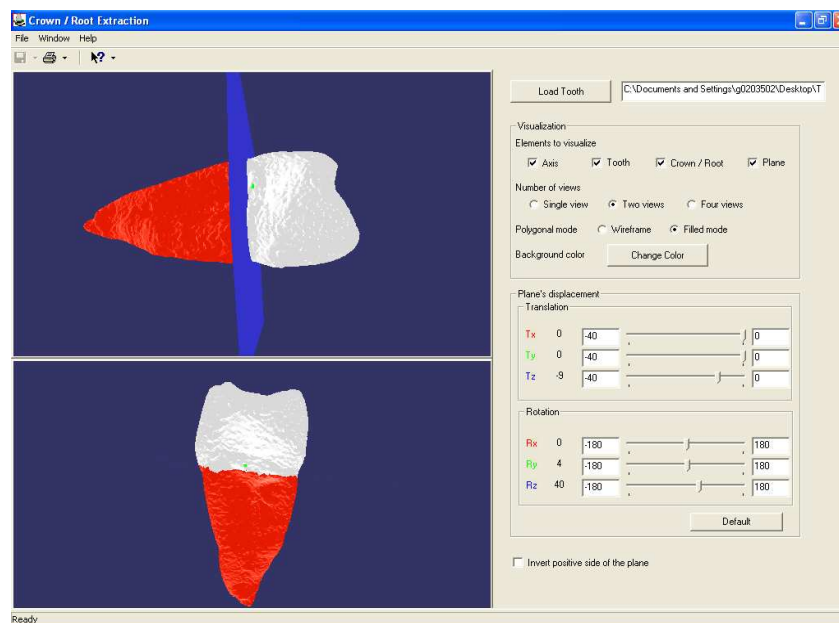


Figure 3.5: Interface used to define the crown or the root of a tooth. Controls on the right side of the GUI allow the user to translate and/or rotate a plane, invert its positive side and choose the elements to visualize as well as the number of views.



Figure 3.6: From the left to the right: a 3D image of a tooth, its crown and its root as defined using the program described above.

Chapter 4

Deformable Models

4.1 Theoretical aspects

Different types of deformable models were presented in Chapter 2. However, they do not exploit the possible deformations of the model studied. The model used in this research was introduced by Cootes *et al.* [12]. Known as the Point Distribution Model (PDM), it describes the average shape and the characteristic shape variations of a set of examples. Its definition derives directly from principal component analysis (PCA).

4.1.1 Introduction to principal component analysis

PCA [13], also known as the Hotelling transform and empirical orthogonal functions, is a widespread method used to reduce the dimensionality of a set of features or to identify new meaningful underlying variables. Originally introduced by Pearson [14], this method is nowadays used in a wide range of areas such as climatology, psychometrics or computer science. The central idea of PCA is to describe the variations of a set of multivariate data in terms of uncorrelated variables, each of which being a particular linear combination of the original variables. These new variables are derived in decreasing order of importance such that the first principal component accounts for as much as possible of the variation in the original data. When the first components account

for most of the variation, they can be used to represent the original data with minor loss of information, thus providing a reduction in the dimensionality of the data and facilitating further processing.

Consider a cloud of points $X = \{\mathbf{x}_0, \dots, \mathbf{x}_N\}$ where $\mathbf{x}_i \in \mathbb{R}^p$ ($0 \leq i \leq N$). In \mathbb{R}^p , we are looking for an affine subspace S_a of dimension $q < p$ that minimizes the inertia I of the cloud of points X (using the usual norm):

$$I = \sum_{i=0}^{N-1} \|\mathbf{x}_i - \hat{\mathbf{x}}\|^2 \quad (4.1)$$

Geometrically, we wish to find the best q -dimensional projection of the data. In this subspace, we are looking for an origin \mathbf{m}_{or} as well as an orthonormal canonical basis $\mathbf{e}_1, \dots, \mathbf{e}_q$. \mathbf{m}_{or} is chosen such that it corresponds to the centroid of the cloud of points, i.e.,

$$\mathbf{m}_{or} = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{x}_i = \mathbb{E}(\mathbf{x}) \quad (4.2)$$

For a given specimen \mathbf{x} , the first and second components y_1 and y_2 are a linear combination of the original variables, i.e.,

$$y_1 = \sum_{i=1}^p e_1^i \mathbf{x}_i = \mathbf{e}'_1 \mathbf{x} \quad (4.3)$$

$$y_2 = \sum_{i=1}^p e_2^i \mathbf{x}_i = \mathbf{e}'_2 \mathbf{x} \quad (4.4)$$

Since we are looking for directions of projection, the vectors \mathbf{e}_1 and \mathbf{e}_2 are considered as unit vectors. \mathbf{e}_1 has to be chosen such that y_1 accounts for the maximum variance. \mathbf{e}_2 should have the greatest variance, subject to the fact that \mathbf{e}_1 and \mathbf{e}_2 are uncorrelated, i.e.,

$$\begin{cases} \mathbf{e}'_2 \mathbf{e}_2 = 1 \\ \mathbf{e}'_2 \mathbf{e}_1 = 0 \end{cases} \quad (4.5)$$

Similarly, the j^{th} principal component, defined by

$$y_j = \mathbf{e}'_j \mathbf{x} \quad (4.6)$$

has to have the greatest variance, subject to

$$\begin{cases} \mathbf{e}'_j \mathbf{e}_j = 1 \\ \mathbf{e}'_j \mathbf{e}_i = 0 \quad (\forall 0 < i < j) \end{cases} \quad (4.7)$$

Consequently, to find the coefficients defining the first principal component, the elements of \mathbf{e}_1 must be chosen to maximize the variance of y_1 , subject to the normalization constraint $\mathbf{e}'_1 \mathbf{e}_1 = 1$.

$$\begin{aligned} \text{Var}(y_1) &= \text{Var}(\mathbf{e}'_1 \mathbf{x}) \\ &= E[(\mathbf{e}'_1 (\mathbf{x} - E(\mathbf{x})))^2] \\ &= \mathbf{e}'_1 E[(\mathbf{x} - E(\mathbf{x}))(\mathbf{x} - E(\mathbf{x}))^T] \mathbf{e}_1 \\ &= \mathbf{e}'_1 \mathbf{S} \mathbf{e}_1 \end{aligned} \quad (4.8)$$

with \mathbf{S} the covariance matrix of the set X . Introducing the Lagrange multiplier l_1 , maximizing Eq. 4.8 subject to $\mathbf{e}'_1 \mathbf{e}_1 = 1$ is equivalent to finding the maximum of

$$\begin{aligned} L(\mathbf{e}_1, l_1) &= \text{Var}(y_1) + l_1(1 - \mathbf{e}'_1 \mathbf{e}_1) \\ &= \mathbf{e}'_1 \mathbf{S} \mathbf{e}_1 + l_1(1 - \mathbf{e}'_1 \mathbf{e}_1) \end{aligned} \quad (4.9)$$

Taking the derivative with respect to \mathbf{e}_1 and equalizing to zero, we get $\mathbf{S} \mathbf{e}_1 = l_1 \mathbf{e}_1$, thus (l_1, \mathbf{e}_1) is a couple (eigenvalue, eigenvector) of \mathbf{S} .

$$\begin{aligned} \text{Var}(y_1) &= \mathbf{e}'_1 \mathbf{S} \mathbf{e}_1 \\ &= l_1 \mathbf{e}'_1 \mathbf{e}_1 \\ &= l_1 \end{aligned} \quad (4.10)$$

If the eigenvalues of \mathbf{S} are $\lambda_1, \dots, \lambda_p$ in decreasing order, l_1 correspond to λ_1 . Similarly, it can be shown that the i^{th} components correspond to the i^{th} eigenvector. The total variance of the p principal components will equal the total variance of the original variables so that

$$\sum_{i=1}^p \lambda_i = \text{trace}(\mathbf{S}) \quad (4.11)$$

and the j^{th} component accounts for a proportion

$$t = \frac{\lambda_i}{\text{trace}(\mathbf{S})} \quad (4.12)$$

Rather than computing the principal components using the variance-covariance matrix, they are usually extracted using the autocorrelation matrix \mathbf{R}_{ac} given by

$$\mathbf{R}_{ac} = \frac{1}{N-1} \sum_{i=0}^{N-1} (\mathbf{x}_i - \mathbf{m}_{or})(\mathbf{x}_i - \mathbf{m}_{or})^T \quad (4.13)$$

This is equivalent to extracting the eigenvectors after standardising every variable to have unit variance and allows to work with data of different nature and different range of variation.

Figure 4.1 illustrates a case where $p = 2$. The red dots represent the set X and the green dot the centroid \mathbf{m}_{or} . To reduce the dimensionality from 2 to 1, the optimum projection corresponds to the orthogonal projection onto the blue line, with direction given by \mathbf{e}_1 .

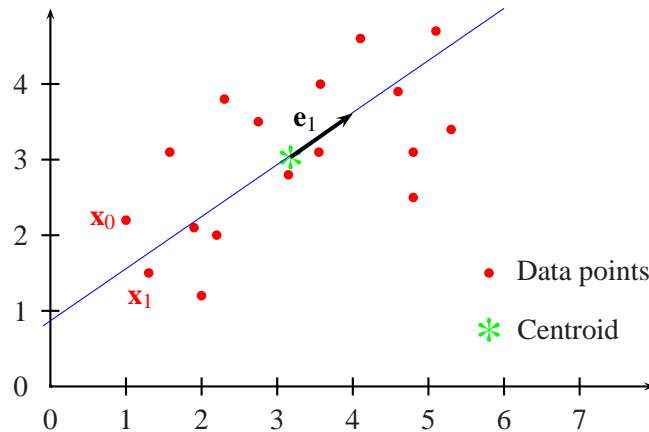


Figure 4.1: Example of dimensionality reduction using a PCA.

4.1.2 Statistical model of Cootes *et al.*

In Cootes' approach, the shape and deformations of a generic object are expressed statistically by formulating the shape as a vector representing a set of points that describe the object. The different specimens are considered as vectors of dimension M and their points as M variables. This shape and its deformations (expressed with a training set, indicative of the object deformations) are learnt through statistical analysis. The PCA is used to detect relationships between the

variables. The first element of the statistical model corresponds to the mean shape $\bar{\mathbf{m}}$ of the set of teeth, defined by $\bar{\mathbf{m}} = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{m}_i$, with N the number of specimens and \mathbf{m}_i the specimens of the training set. The second element corresponds to the different modes of variations. These modes symbolize the ways in which the points tend to move together. They are the eigenvectors \mathbf{e}_i of the covariance matrix \mathbf{R}_{ac} as defined above.

Given the high resolution of the 3D models used, a direct estimation of the eigenvectors of \mathbf{R}_{ac} is totally impossible. Instead of working in the variable (tooth points) space $\mathbb{R}^{3 \times M}$ (dimension $3 \times M$), the eigenvectors are defined in the training sample space \mathbb{R}^N (dimension $N \ll 3 \times M$). Simple mathematical considerations [15] give us an immediate correspondence between the eigenvectors and eigenvalues in these two spaces. The eigenvalues in $\mathbb{R}^{3 \times M}$ and \mathbb{R}^N are identical, and if $(\lambda_\alpha, \mathbf{u}_\alpha)$ is a couple (eigenvector, associated eigenvalue) in $\mathbb{R}^{3 \times M}$, then $(\lambda_\alpha, \mathbf{v}_\alpha)$ is also a couple (eigenvector, associated eigenvalue) in $\mathbb{R}^{3 \times N}$ where

$$\mathbf{v}_\alpha = \frac{1}{\sqrt{\lambda_\alpha}} \mathbf{X} \mathbf{u}_\alpha \quad (4.14)$$

and the rows of the matrix \mathbf{X} correspond to the tooth points ($N \times 3M$ matrix).

The proportion of the total variance explained by each vector is proportional to the corresponding eigenvalues (Eq. 4.12). Consequently, the eigenvectors \mathbf{e}_i associated with the maximum eigenvalues λ_i correspond to the major deformation modes.

Any shape belonging to the training set can be approximated as a sum of the mean model and a linear combination of the first N_{pc} modes, i.e.,

$$\mathbf{m} = \bar{\mathbf{m}} + \sum_{i=1}^{N_{pc}} \omega_i \mathbf{e}_i \quad (4.15)$$

where ω_i are the weights associated with the eigenvectors \mathbf{e}_i . By constraining every ω_i such that

$$-K_{\min,i} \times \lambda_i \leq \omega_i \leq K_{\max,i} \times \lambda_i \quad (4.16)$$

with $K_{\min,i}$ and $K_{\max,i}$ some constants determined from the variations observed among the sample,

we can limit the deviations from the mean model.

4.2 Properties of the defined model

4.2.1 Creation of the statistical model

A deformable model representative of an upper right second premolar was built using the 22 teeth of the training set. Table 4.1 summarizes the results obtained for the different modes of deformation.

- Figure 4.2 represents the eigenvalues of the autocorrelation matrix in decreasing order. The first eigenvalue is associated with the first mode, the second eigenvalue with the second and so on.
- Figure 4.3 gives the percentage of variability explained by each of the first eight modes independently. The first mode alone explains nearly 50% of the variability observed in the test set, while the modes of degree equal or higher to 9 represent no more than 1% of the variability.
- Figure 4.4 shows the cumulative percentage of variability given the number of components. Seven components can explain 96% of the variability, whereas thirteen components explain more than 99% of the variability between the different samples.

Three different approaches are commonly used to determine the number of components to select:

1. The first rule consists in fixing the percentage of variance the modes have to explain, e.g., 95% or 99%. This would represent a number of components equal to 7 or 13 respectively.
2. The second rule excludes the components that account for a percentage of the variance smaller than a given limit, e.g., 1%. This would exclude the modes higher than 9.

3. The last method consists in plotting the eigenvalues λ_i as a function of i (Figure 4.2) and selecting the number of components as the value corresponding to an “elbow” in the representation. In this later case, the number of components could be fixed equal to 4 (first “elbow”) or 10 (second “elbow”).

These different approaches would lead to a number of components quite different. Consequently, rather than fixing definitely the number of components, we decided to realize the reconstruction tests with a number of components lying between 7 and 13. Increasing the number of modes above 13 would introduce noise in the reconstruction, but hardly any useful information about tooth shape.

Table 4.1: Eigenvalues of the training set’s autocorrelation matrix and the corresponding amount of variance accounted for.

Mode	Eigenvalues	% total variance	Cumulative eigenvalues	Cumulative % of variance
1	1.831e+08	48.59	1.831e+08	48.59
2	8.053e+07	21.38	2.636e+08	69.97
3	4.490e+07	11.92	3.085e+08	81.89
4	1.981e+07	5.26	3.283e+08	87.15
5	1.552e+07	4.12	3.438e+08	91.27
6	9.590e+06	2.55	3.534e+08	93.81
7	7.134e+06	1.89	3.606e+08	95.71
8	4.210e+06	1.12	3.648e+08	96.82
9	3.202e+06	.85	3.680e+08	97.67
10	1.917e+06	.51	3.699e+08	98.18
11	1.287e+06	.34	3.712e+08	98.52
12	1.150e+06	.31	3.723e+08	98.83
13	1.036e+06	.27	3.734e+08	99.10
14	6.747e+05	.18	3.740e+08	99.28
15	6.628e+05	.18	3.747e+08	99.46
16	5.323e+05	.14	3.752e+08	99.60
17	4.690e+05	.12	3.757e+08	99.73
18	3.861e+05	.10	3.761e+08	99.83
19	2.584e+05	.07	3.763e+08	99.90
20	2.326e+05	.06	3.766e+08	99.96
21	1.563e+05	.04	3.767e+08	100.00
22	7.250e-09	.00	3.767e+08	100.00

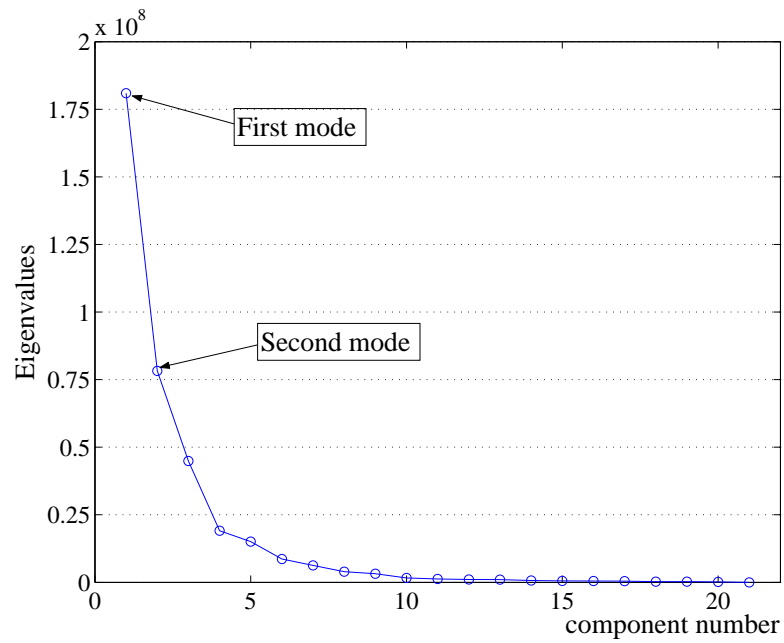


Figure 4.2: Eigenvalues of the training set's autocorrelation matrix.

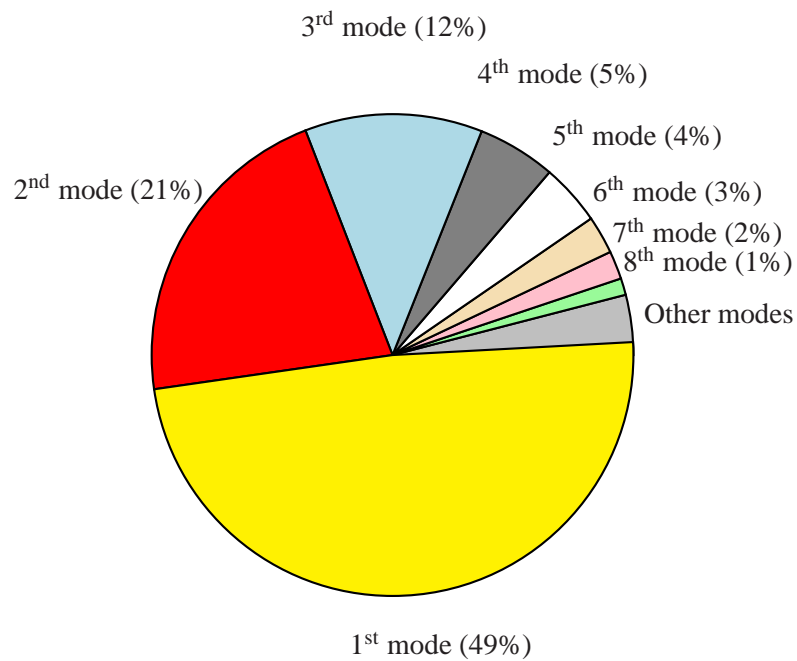


Figure 4.3: Percentage of variability for the first 8 modes of deformation.

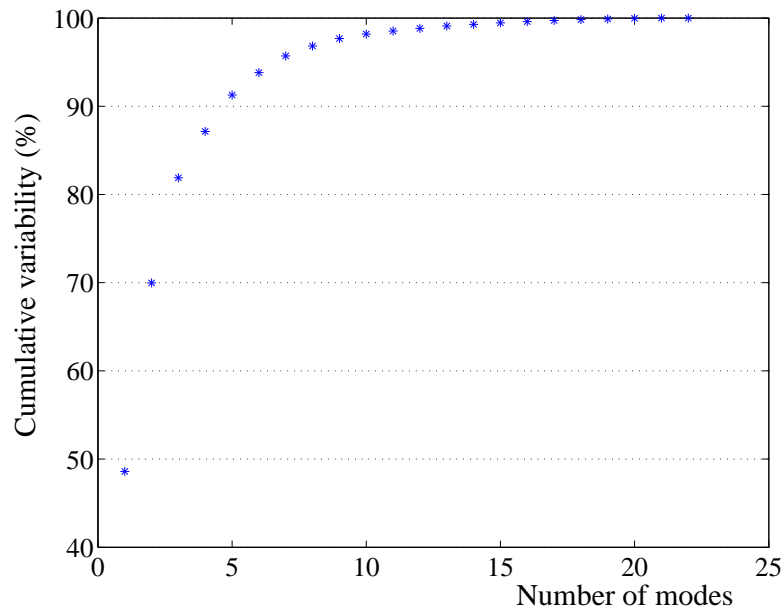


Figure 4.4: Cumulative percentage of variability given the number of modes.

4.2.2 Influence of the modes

To constrain the statistical model deformations, we need to define the value of the constants in Equation 4.16. For each specimen k , the coefficients y_i^k ($1 \leq i \leq 13$) of the decomposition were computed (see Eq. 4.6). Table 4.2 indicates, for each mode i , the minimum and the maximum value of $\frac{y_i^k}{\lambda_i}$. Based on this table, the constant $K_{\min,1}$ was set to 0.3 and $K_{\max,1}$ to 0.8. For the modes superior or equal to 2, the constant $K_{\min,i}$ and $K_{\max,i}$ ($2 \leq i \leq 13$) were both set equal to 0.55.

Figures 4.5 to 4.8 show the influence on the first four modes of the decomposition. In each figure, the mesial, buccal and occlusal views are represented. The tooth in the middle represents the mean shape. Teeth on both sides represent the extreme variations than can be induced by a given mode. The tooth on the left shows the new shape when the i^{th} mode is added weighted by a coefficient $-K_{\min,i} \times \lambda_i$ ($1 \leq i \leq 4$), whereas the tooth on the right shows the new shape when the i^{th} mode is added weighted by a coefficient $K_{\max,i} \times \lambda_i$.

The first mode has a huge influence on the size of the teeth and also the global shape of the root. The second mode has a much smaller influence on the tooth height. However, this mode will

Table 4.2: Maximum and minimum values of the modes weights for the training set's specimen.

Mode	Minimum	Maximum
1	-0.6304	0.2394
2	-0.4914	0.3850
3	-0.3719	0.4139
4	-0.4669	0.3591
5	-0.3804	0.3715
6	-0.4995	0.3606
7	-0.3494	0.5171
8	-0.3232	0.3801
9	-0.4454	0.3633
10	-0.3719	0.4139
11	-0.4669	0.3591
12	-0.3804	0.3715
13	-0.4995	0.3606

have a great impact on the width of the root. The other modes have different influences, more or less similar, that cannot always be simply described. These constraints will help avoiding getting shapes that do not respect the statistical shape variations during the reconstruction.

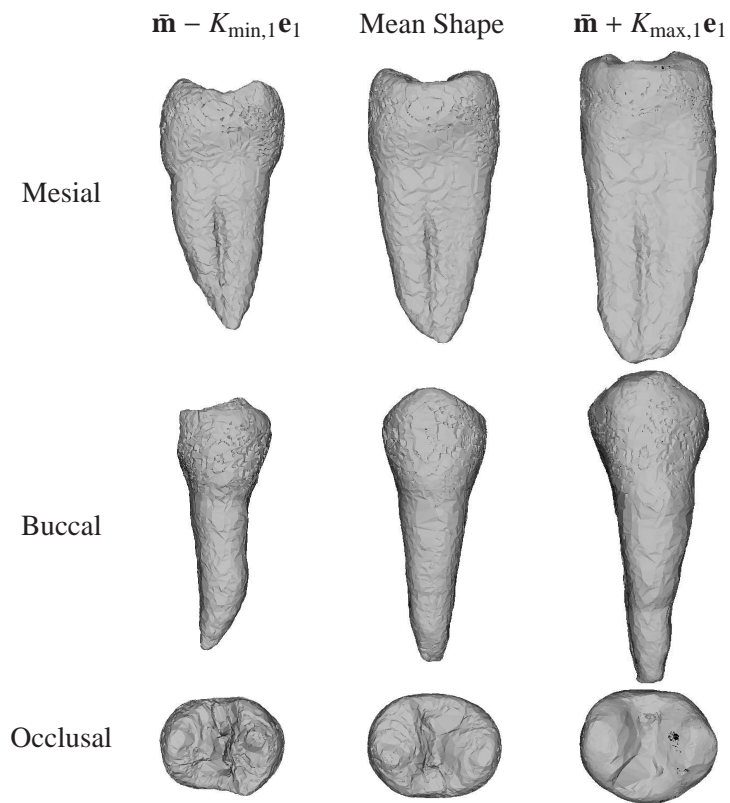


Figure 4.5: Influence of the first mode of decomposition on the mean shape.

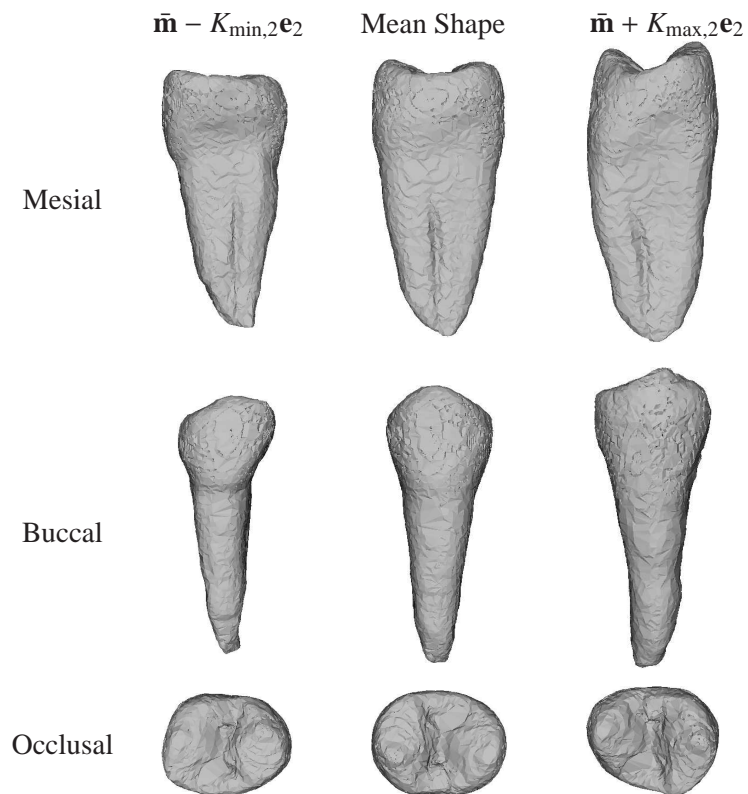


Figure 4.6: Influence of the second mode of decomposition on the mean shape.

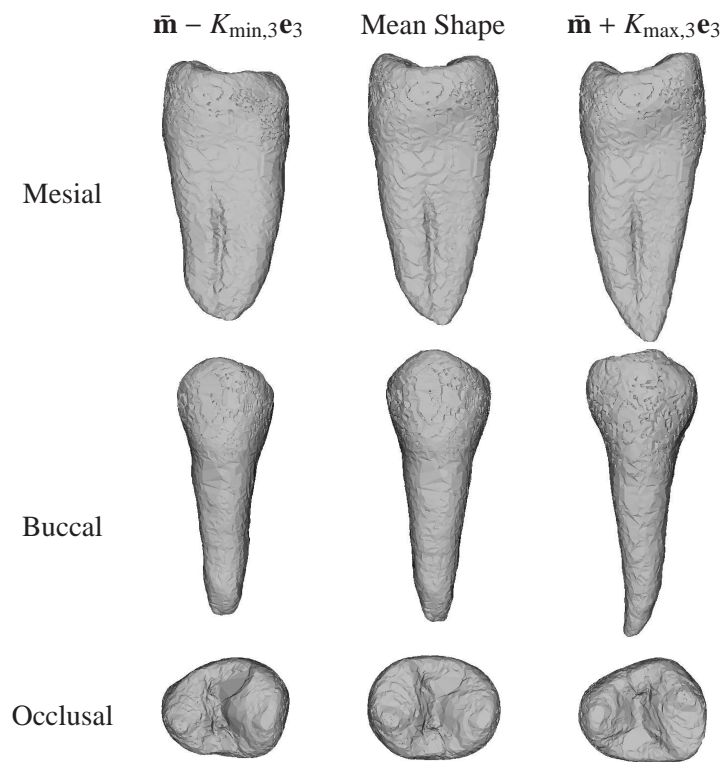


Figure 4.7: Influence of the third mode of decomposition on the mean shape.

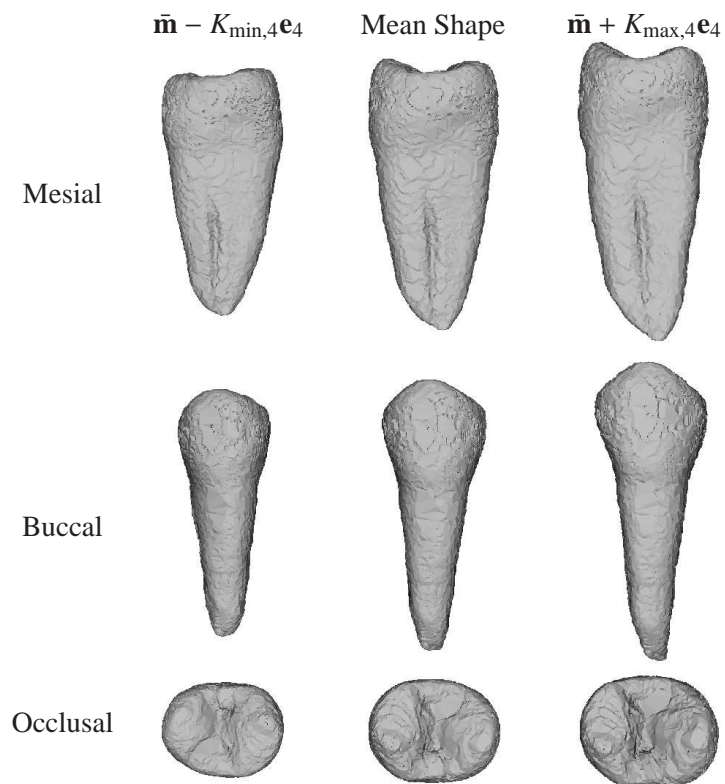


Figure 4.8: Influence of the fourth mode of decomposition on the mean shape.

4.3 Definition of crown/root parameters

To reconstruct a tooth, one could try to deform the entire model to match the shape. However, the computation time would be quite high. If the fragments available correspond neither to the entire (or nearly entire) crown nor to the entire root, deforming the global model is the only possibility to recover the tooth shape. Except for forensic medicine, the specimen to reconstruct is usually of one of these two forms: either a root only or a crown only (e.g., dental cast or patient's tooth digitized using an intra-oral camera). That is why after defining the statistical model, two other statistical models were derived, one for the root, the other for the crown, in order to exploit this complementary information.

- The crown (or the root) of the mean shape was extracted using the method introduced Section 3.3.
- New modes of deformation (vectors \mathbf{e}_i) were also determined using the points of the mean shape belonging to the crown (or the root).

Chapter 5

Registration process

Once the statistical model has been defined, the patient's tooth shape is inferred by computing the optimal rigid and elastic transformations of the mean model, as described by Fleute *et al.* [9]. We aim at matching the patient's crown (or root) with the crown (or root) of the mean shape by determining the contribution of the different modes of the statistical model. The crown (or root) shape variations with respect to the mean model are then used to infer the root shape.

5.1 Rigid Registration

In order to reconstruct the target tooth, an intermediary step is included before the final deformation. A rigid registration (see Section 2.1.1) is performed to align the mean shape with the target (the target is considered as fixed). Although not compulsory, this step can greatly decrease the running time of the reconstruction and improve the final results.

A manual rigid registration (optional) can be performed (the user can rotate and/or translate the mean shape through an interface). If the poses of the two teeth are radically different, this step becomes crucial. The Iterative Closest Point (ICP) algorithm is then used to perform the best possible alignment between the two models. This algorithm was introduced by Chen and Medioni[16] and Besl and McKay[17]. This method has the advantage of being very fast as long

as the process to find the nearest points from one volume to the other is effective. However, local minima can be an issue, which makes the manual rigid registration compulsory when the orientations of the two volumes are highly different.

Principle of the ICP algorithm:

Given two sets of 3D points, $V_M = \{\mathbf{m}_1, \dots, \mathbf{m}_M\}$ considered as fixed and $V_D = \{\mathbf{d}_1, \dots, \mathbf{d}_D\}$, we want to find the optimal rigid registration consisting of a rotation \mathbf{R}_{ICP} and a translation \mathbf{T}_{ICP} which minimizes the following cost function:

$$E(\mathbf{R}_{ICP}, \mathbf{T}_{ICP}) = \sum_{i=1}^M \sum_{j=1}^D w_{ij} \|\mathbf{m}_i - (\mathbf{R}_{ICP} \mathbf{d}_j + \mathbf{T}_{ICP})\|^2 \quad (5.1)$$

where $w_{ij} = 1$ if the i^{th} point of V_M and the j^{th} point of V_D are corresponding points, and 0 otherwise.

The ICP algorithm calculates the optimal rotation and translation iteratively. At each step k of the algorithm, the set of closest points is computed and then \mathbf{R}_{ICP}^k and \mathbf{T}_{ICP}^k are determined. That leads to a new set V_D called V_{D_k} after each transformation. A quaternion based-algorithm is used to perform the solid registration. The rotation is represented using a quaternion $\mathbf{q}_R = [q_0, q_1, q_2, q_3]^t$, where $q_0 \geq 0$ and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. The corresponding 3×3 rotation matrix \mathbf{R}_{ICP} is generated according to the following scheme:

$$\mathbf{R}_{ICP} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix} \quad (5.2)$$

The translation is represented by a vector $\mathbf{q}_T = [q_4, q_5, q_6]$. Both the rotation and the translation can be summarized by a single vector $\mathbf{q} = [\mathbf{q}_R | \mathbf{q}_T]$. To determine Y_k , the set of nearest neighbors of every point of V_{D_k} to V_M at step k , we opted for the use of a Kd-tree (described in Appendix C) with $d = 3$. A Kd-tree is a data structure commonly used for searching in a space of dimension d . Its particular hierarchy gives a fast access to the nearest neighbor of every point of V_{D_k} to V_M .

To compute \mathbf{q}_k , the quaternion at step k , we first define $\mathbf{Q}(\boldsymbol{\Sigma}_{DM})$:

$$\mathbf{Q}(\boldsymbol{\Sigma}_{DM}) = \begin{bmatrix} \text{trace}(\boldsymbol{\Sigma}_{DM}) & \boldsymbol{\Delta}^t \\ \boldsymbol{\Delta} & \boldsymbol{\Sigma}_{DM} + \boldsymbol{\Sigma}_{DM}^t - \text{trace}(\boldsymbol{\Sigma}_{DM})\mathbf{I}_3 \end{bmatrix} \quad (5.3)$$

with μ_D and μ_M the ‘‘centers of mass’’ of the sets V_{D_k} and V_M ,

$\boldsymbol{\Sigma}_{DM}$ the cross-covariance matrix is given by

$$\boldsymbol{\Sigma}_{DM} = \frac{1}{D} \sum_{i=1}^D [(\mathbf{d}_i - \mu_D)(\mathbf{m}_i - \mu_M)^t] \quad (5.4)$$

$\mathbf{A} = \boldsymbol{\Sigma}_{DM} - \boldsymbol{\Sigma}_{DM}^t$ and $\boldsymbol{\Delta} = [A_{23}, A_{31}, A_{12}]^t$.

$\mathbf{q}_R^k = [q_0^k, q_1^k, q_2^k, q_3^k]^t$ is the unit eigenvector corresponding to the maximum eigenvalue of $\mathbf{Q}(\boldsymbol{\Sigma}_{DM})$ and $\mathbf{q}_T^k = \mu_M - \mathbf{R}_{ICP}^k \mu_D$. The algorithm terminates when the change in mean-square error (Eq. 5.1) falls below a preset threshold. The assumption is that the correspondence is correct during the last iteration.

Once \mathbf{R}_{ICP} and \mathbf{T}_{ICP} have been calculated, the transformation is inverted in order to move the deformable model toward the target¹.

Figure 5.1 represents the first four steps during the rigid registration of a tooth root (red cloud of points) with the mean shape (green and white 3D meshes, where the white mesh represents the initial position of the mean shape and the green one represents the mean shape after rotation and translation). The top left image shows the first iteration of the ICP algorithm and the bottom right image the fourth one (last iteration).

5.2 Elastic Registration: Optimizing the Modes’ Weights

Once the two volumes have a similar position and a similar orientation, the statistical model is used to reconstruct the tooth. The deformable model used here corresponds either to the global model or to one of the two partial models (crown or root) if applicable. During the elastic

¹Moving the mean shape directly toward the target would also be possible. However, the density of the mean shape is (normally) much higher than those of the target and this would slow down the registration process. Both registration schemes lead to similar results

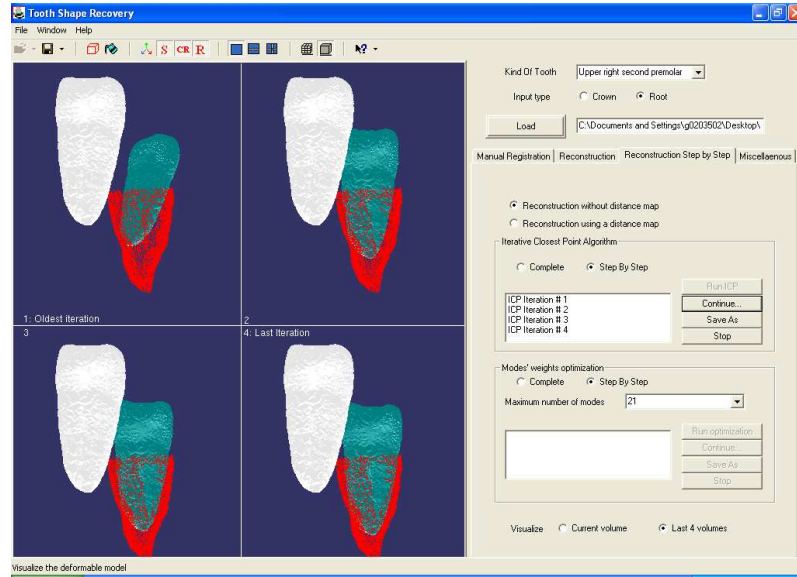


Figure 5.1: Automatic rigid registration using the ICP algorithm.

registration, the values of the different modes as well as the parameters of the rigid registration are optimized in order to warp the mean shape onto the target.

The optimum parameters can be computed through the minimization of a merit function E to measure the goodness-of-fit. The merit function used here (Equation 5.5) is the simple mean-squared distance² between the crowns (or roots) of the two volumes:

$$E(\mathbf{p}) = \sum_{i=0}^{M_{pdm}-1} \min(\|\mathbf{d}_j - \mathbf{m}_i\|^2)_{1 \leq j \leq K} \quad (5.5)$$

$$\text{with } \mathbf{m} = \mathbf{R}(\bar{\mathbf{m}} + \sum_{l=1}^{N_{pc}} \omega_l \mathbf{e}_l) + \mathbf{T}$$

where \mathbf{p} is a vector representing the different parameters to adjust, M_{pdm} the number of points of the PDM, K the number of points of the target tooth, \mathbf{d} the vector representing the target (dimension $3 \times K$), \mathbf{T} a translation vector, \mathbf{R} a rotation matrix, and \mathbf{e}_l the principal components obtained in Section 4.2.1 (N_{pc} corresponds to the number of principal components selected).

We need to estimate the six components that define the rigid-body transformation between the two volumes (3 parameters for \mathbf{T} and 3 parameters for \mathbf{R}) as well as the optimum weights

²Euclidean distance

for the N_{pc} principal components, i.e., we have to solve an optimization problem having only $6 + N_{pc}$ parameters. As mentioned in Section 4.2.2, the number of modes to consider belongs to the range 7 to 13, leading to a number of parameters to optimize between 13 and 20.

5.2.1 Choice of the optimization scheme

The problem at hand corresponds to the constraint minimization of a non-linear function. Different schemes were envisaged to realize the optimization [18].

Descent algorithms Steepest descent algorithms (SDA) are extremely simple to implement.

However, they are both unstable and inefficient, due to the slow rate of convergence. Consequently, they are never used alone for complex optimization problems.

Simulated annealing algorithms (SAA) SDA as well as the other methods described below

only accept the direction of minimization if it leads to a decrease of the objective function E . On the contrary, SAA accept changes leading to a decrease with a probability $P = \exp(-\frac{E_{k+1}-E_k}{T})$. The parameter T is called temperature, by analogy with the annealing process of metals and glass. Its value is decreased progressively. A high temperature will lead to higher chance of moving to an uphill position. The higher the number of iterations, the smaller the probability to choose a direction increasing the objective function. Consequently, SAA's major advantage is to avoid deep minima. However, this method can become quite slow or inefficient if the initial value of T is not chosen properly. The search process varies between the different SAA, and can take the form of a random walk. This method is usually used when we have little information about the function's properties such as continuity and derivability.

Newton's method Newton's method is based on a second order Taylor series expansion of the function to minimize about the current point \mathbf{x}_k . This leads to a parabolic approximation of the function at \mathbf{x}_k . Finding the minimum of this parabola is then straightforward. The

process is repeated upon convergence. Although very powerful, this method requires the Hessian matrix to be positive definite. Furthermore, the computation and the inversion of the Hessian matrix is extremely time-consuming. The Gauss-Newton method is an extension of Newton's method that approximates the Hessian to speed up the algorithm. These algorithms work fast near the optimal solution, but poorly far from it or when the Hessian is ill-conditioned, and can lead to local minima.

Levenberg-Marquardt algorithm The Levenberg-Marquardt method (described in Appendix A) can be considered as a cross between the Gauss-Newton and the steepest descent algorithms. The continuous switch between these two schemes leads to an algorithm that is both robust and efficient, and particularly suited for least mean squares minimization.

Contrary to Fleute *et al.* [9], we chose to perform the optimization of $E(\mathbf{p})$ using the Levenberg-Marquardt algorithm. Similar to other gradient methods, e.g., conjugate gradient algorithm and Newton-Gauss algorithm, the Levenberg-Marquardt algorithm can lead to local minima. To get round this difficulty, the modes are progressively added. Since the original pose is close to the optimal one and no prior information exists about the weight values, all the parameters are set to zero, and the first optimization is done. After each adding, the optimizing process is iterated, keeping the values computed before as the initial parameters. To compute the minimum distances (and consequently the gradient), two different options were tested, with and without the use of a precomputed distance map.

As mentioned before, the modes coefficient ω_i are all constrained in the range $[-K_{\min,i} \times \lambda_i, +K_{\max,i} \times \lambda_i]$. No constraint is added for the translation vector components. The rotation angles around every axis of the coordinate system are limited to $[-3^\circ, +3^\circ]$.

5.2.2 Optimization using a distance map

Computation of the distances

Computing the minimum distances is the most computationally expensive part of the minimization. Here, distances were approximated using an octree-spline distance map (described in Appendix B). The distance map is computed using the points of the target. Since the mean shape is deformed after each deformation, computing the distances on the mean shape is impossible. Our objective function is slightly different from those introduced in Equation 5.5. Indeed, the summation is not realized over the target's points, but over the mean shape. The octree comprises 6 levels of decomposition. The size of the terminal octants depends on the size of the target, those of the mean shape as well as the relative position of the two volumes (since the final octants' size corresponds to $\frac{1}{64}$ times the size of the original bounding box enclosing the two volumes). Since the pose is nearly optimal at the beginning of the minimization, each terminal octant has a size inferior to $400 \mu\text{m}$. Due to the resolution of the initial data (after decimation, the teeth of the training set have a resolution of $105\mu\text{m}$), this resolution seems to be the most appropriate.

Computation of the gradient

To compute the partial derivatives with respect to every component, we use Ridder's method of polynomial extrapolation [19]. Let us consider the derivative of a function f at x (x scalar).

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (5.6)$$

Taking an h small enough to compute this derivative could lead to an inaccurate result due to the fact that a computer cannot store most numbers with total accuracy and inevitably introduced an approximation. Instead of taking a fixed value for h , Ridder tries to extrapolate $f'(x)$ when $h \rightarrow 0$. Polynomial extrapolations of higher and higher order are produced, with a value of h smaller and smaller. The termination criterion is reached when either the estimated error is less than a preset

error tolerance, the error estimate is significantly worse than the one previously obtained or the maximum order is reached. Ridder's method derives directly from the Richardson extrapolation (the general idea of this method was first published by Bulirsch in [20]). Though this method leads to excellent results, its main drawback is the large number of function evaluations.

5.2.3 Optimization without distance map

Computation of the distances

For each point of the target, its closest point on the mean shape after modification by the actual parameters is determined, giving immediately the minimum distance. The set of closest points is computed using Kd-trees (Appendix C), a method that was already used during the ICP-based rigid registration. It was decided to look for the closest points of the target, since its density is supposed to be much lower than those of the mean shape. Consequently, this second method requires constructing a new Kd-tree³ after each iteration of the Levenberg-Marquardt algorithm.

Computation of the gradient

Contrary to the previous case, a closed-form expression for the gradient of $E(\mathbf{p})$ becomes available. We first introduce $E_i(\mathbf{p})$:

$$E_i(\mathbf{p}) = \min(\|\mathbf{d}_j - \mathbf{m}_i\|)_{1 \leq j \leq K} \quad (5.7)$$

Then $E(\mathbf{p}) = \sum_{i=0}^{M_{pdm}-1} E_i(\mathbf{p})$ and $\nabla E(\mathbf{p}) = \sum_{i=0}^{M_{pdm}-1} \nabla E_i(\mathbf{p})$. We also introduce $\mathbf{T} = \{T_x, T_y, T_z\}$ and $\theta_x, \theta_y, \theta_z$ the rotation angles around $\vec{\mathbf{x}}, \vec{\mathbf{y}}, \vec{\mathbf{z}}$ respectively⁴. Finally, we define the set I such that $I(j)$ is the closest point's index of the i^{th} target point.

Derivatives with respect to the modes weights:

$$\frac{\delta E_i}{\delta \omega_s} = 2 \left(\left(\vec{\mathbf{m}} + \sum_{k=1}^{N_{pc}} \omega_k \mathbf{e}_{k_{I(i)}} \right)^T + (\mathbf{T} - \mathbf{d}_i)^T \mathbf{R} \right) \cdot \mathbf{e}_{k_{I(i)}} \cdot \mathbf{d}_i \quad (5.8)$$

³Kd-Tree: construction time: $O(dM_{pdm} \ln M_{pdm})$, with d the space dimension, here $d = 3$.

⁴The rotation matrix corresponds to the product $\mathbf{R}_z \mathbf{R}_y \mathbf{R}_x$, where \mathbf{R}_i is the rotation around $\vec{\mathbf{i}}$.

Derivatives with respect to the translation vector's components:

$$\begin{aligned} \frac{\delta E_i}{\delta T_x} &= (A_1, 0, 0)^T \mathbf{d}_i, & \frac{\delta E_i}{\delta T_y} &= (0, A_2, 0)^T \mathbf{d}_i & \text{and} & & \frac{\delta E_i}{\delta T_z} &= (0, 0, A_3)^T \mathbf{d}_i \\ \text{with } \mathbf{A} &= 2 \left(\mathbf{T} + \mathbf{R} \left(\bar{\mathbf{m}} + \sum_{k=1}^{N_{pc}} \omega_k \mathbf{e}_{k_{I(i)}} \right) - \mathbf{d}_i \right) \end{aligned} \quad (5.9)$$

Derivatives with respect to the rotation angles:

$$\begin{aligned} \frac{\delta E_i}{\delta \theta_a} &= \frac{\delta g}{\delta \theta_a} \\ \text{with } g &= 2 (\mathbf{T} - \mathbf{d}_i)^T \mathbf{R} \left(\bar{\mathbf{m}} + \sum_{k=1}^{N_{pc}} \omega_k \mathbf{e}_{k_{I(i)}} \right) \quad \text{and} \quad a \in \{x, y, z\} \end{aligned} \quad (5.10)$$

Deriving a closed-form expression for the $\frac{\delta E_i}{\delta \theta_a}$ is then straightforward.

Equations 5.8, 5.9 and 5.10 possess common factors. Consequently, the computation of the partial derivatives with respect to all the parameters is extremely fast.

5.3 Generation of a 3D mesh

As mentioned before, the target is a simple cloud of points and the model a 3D mesh (obtained using a marching cube algorithm [21]). However, contrary to Fleute *et al.* [9], we do not use this information during the optimization process (neither during the rigid registration nor during the solid one). Due to the high density of the mean shape, the distance from a point of the target to the closest point on the mean shape gives an excellent approximation of the distance from a point of the target to the surface of the mean shape. Once the registration process is ended, the output consists in a 3D mesh of high density (those of the deformable model, i.e., approximately 16,000 points).

5.4 Framework

Figure 5.2 summarizes the entire process. The outline phase of the algorithm (construction of the statistical model) was implemented in Matlab, except for extraction of the crowns or roots,

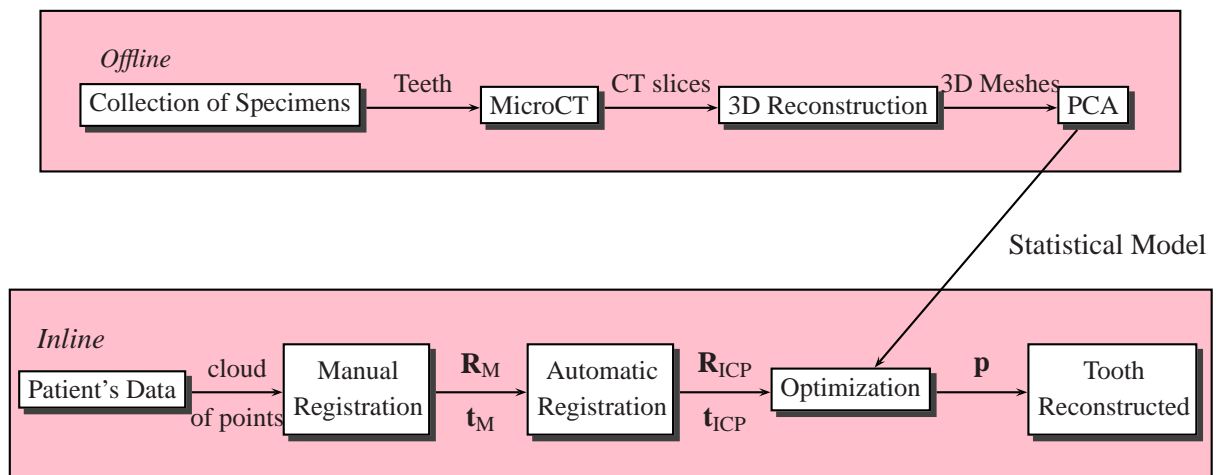


Figure 5.2: Global tooth shape recovery process.

which was implemented in C++ (Figure 3.5). The GUI shown in Figure 5.1 was developed in C++ in order to implement the inline phase of the reconstruction. It allows us to:

- Realize the registration of a specimen with the mean shape manually (rotation, translation) and possibly scaling.
- Launch the reconstruction using either one of the two methods described above. The user has the possibility to visualize the results step by step in order to observe every iteration of the ICP algorithm or evaluate the effect of every component on the tooth shape, or to launch the entire process.
- Manipulate the different objects of the scene (e.g., translation, rotation or zoom in and out).

Chapter 6

Experiments and Results

6.1 Experiments

6.1.1 Tests performed

Different experiments were conducted in order to investigate the performance and capabilities of the method implemented. In this chapter, the two optimization processes defined in Chapter 5 will be referred as methods KD and OD (the first method refers to the optimization using a kd-tree decomposition (KD) and without distance map, the second method to the optimization using an octree-spline (OD) distance map).

Influence of the number of modes: This test aims at choosing the optimal number of modes for the reconstructions. As mentioned earlier, the number of modes can play an important part in the quality of the reconstruction. An insufficient number of modes would limit the deformations of our mean model and prevent it taking on the exact shape of the target, whereas a high number of deformation modes would slow down the reconstruction and above all introduce a damageable noise.

Leave-one-out test on the training set: The aim of this test is to validate the capabilities of the method to reconstruct a tooth given its root or its crown. For each of the N teeth belonging to the

training set, the following test is realized:

1. The PCA is first performed on $N - 1$ teeth (excluding tooth T_i) and a new statistical model is defined (mean shape, modes for the entire tooth as well as partial statistical models for the root and the crown).
2. T_i 's crown and root are extracted.
3. The new modes defined are used to reconstruct T_i given its crown (new model TC_i) and its root (new model TR_i).
4. The distance between the tooth T_i before (entire shape) and after the reconstruction (TC_i or TR_i) is then calculated.

The statistical model S_M defined using the entire set cannot be used to reconstruct the teeth in the leave-one-out test. Indeed, we aim at testing the generalization capacity of this method, i.e., its capacity to perform well on unseen targets. Consequently, testing the system with S_M would lead to overoptimistic results. A good generalization can occur only if the data set is sufficiently rich. This test should allow us to determine the validity of the method and whether or not the specimens collected present sufficient variability in their size and shape.

Test using patient's data: In this experiment, we investigate the potential of this method when dealing with patient's data.

Use of feature points: We then investigate the use of feature points to improve the results of the reconstruction when we only have root information.

Influence of the target resolution on the results: The results presented in the leave-out-one test section were obtained using a tooth whose resolution is similar to the resolution of the mean shape. In this part, the effect of a decrease in the target resolution is studied.

Computation time: In this experiment, we present the differences observed in the computation time between the two methods used (with or without distance map).

Number of specimens necessary to build the statistical model: 22 specimens were used during this study. In this experiment, we evaluate experimentally the number of teeth necessary to get a good knowledge of the variability among second upper right premolars.

6.1.2 Error measure

In order to check the validity of the reconstruction, we need to define an estimator to compare the teeth with their reconstructed shape $TRec_i$ ($= TC_i$ or TR_i). Though the Euclidean distance would give a good approximation of the distance between the two volumes due to the high density of $TRec_i$, it was decided to use the Hausdorff distance as described by Aspert *et al.* in [22]. The gain in accuracy should be minor, but this metric is more appropriate for a comparison of 3D meshes. Its advantages are to account for the shape of the volumes as well as their relative position.

The Hausdorff distance (HD) $d(\mathcal{S}, \mathcal{S}')$ between two surfaces \mathcal{S} and \mathcal{S}' is given by

$$d(\mathcal{S}, \mathcal{S}') = \max_{p \in \mathcal{S}} d(p, \mathcal{S}'), \quad (6.1)$$

$$\text{with } d(p, \mathcal{S}') = \min_{p' \in \mathcal{S}'} \|p - p'\|_2, \quad p' \in \mathcal{S}'$$

where $\|\cdot\|_2$ denotes the Euclidean norm¹ and p and p' are points in \mathbb{R}^3 .

Since maximin functions are asymmetric, the symmetrical Hausdorff distance d_s is used instead of the simple HD.

$$d_s(\mathcal{S}, \mathcal{S}') = \max [d(\mathcal{S}, \mathcal{S}'), d(\mathcal{S}', \mathcal{S})] \quad (6.2)$$

We also introduce the mean distance d_m and the root mean square error (RMS) d_{rms} between two surfaces \mathcal{S} and \mathcal{S}' . The corresponding symmetric distances will be used later to compare the teeth T_i with $TRec_i$.

$$d_m(\mathcal{S}, \mathcal{S}') = \frac{1}{|\mathcal{S}|} \iint_{p \in \mathcal{S}} d(p, \mathcal{S}') d\mathcal{S} \quad (6.3)$$

$$d_{rms}(\mathcal{S}, \mathcal{S}') = \sqrt{\frac{1}{|\mathcal{S}|} \iint_{p \in \mathcal{S}} d(p, \mathcal{S}')^2 d\mathcal{S}} \quad (6.4)$$

¹It could be any other metric.

To test if the respective position of the two models is optimal, an iterative scheme from coarse to fine resolution was used. The translation minimizing the distance between the two models was first computed and then the optimal rotation. The coordinates of the translation vector are represented as a point in a 3D grid whose points regularly spaced. The procedure is as follows:

1. The translation is applied to the reconstructed tooth.
2. The mean HD between the two volumes is computed. If this distance is smaller than the previous one, the new value is stored.
3. Once all the points of the grid have been tested, the process is iterated using a new grid centered on the optimal translation just defined.

Once the translation has been computed, the optimal rotation is then determined using the same scheme. Figure 6.1 illustrates the method's framework (Nb_It represents the number of iterations). The tests did not lead to a significant difference in the pose of the original models and the reconstructed ones. Consequently, all the distances mentioned below correspond to a Hausdorff distance estimation without supplementary solid registration.

6.2 Influence of the number of modes

The number of modes necessary for the reconstruction was determined by choosing at random 3 teeth from the training test and reconstructing them using either the root or the crown. Then, the symmetric Hausdorff distance was estimated. The reconstruction was done using method KD. The method used to define the statistical models is similar to the leave-one-out method presented in Section 6.1.1.

As shown by Figure 6.2 and 6.3, choosing the optimal number of modes is not obvious. For the reconstruction using the crown, increasing the number of modes can either improve (tooth c), deteriorate (tooth a) or even have no effect on the reconstruction (tooth b). The results obtained

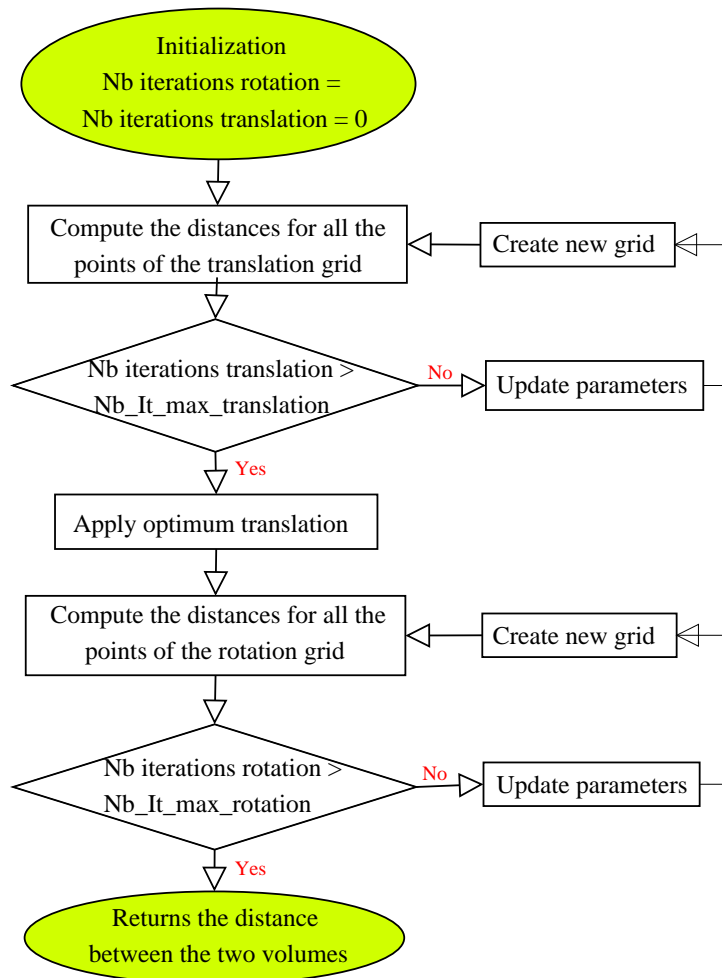


Figure 6.1: Framework for the pose verification.

using the roots present different variations, but no global tendency either.

It was decided to set the number of modes equal to 7. Though this may not be the optimal choice for all the teeth, it can prevent the dramatic effects that are occasionally observed on the tooth shape when it is greater than 8 (if the reconstruction is made using the root).

From this point on, all the reconstructions mentioned were performed using seven modes of deformation, unless otherwise mentioned.

6.3 Leave-one-out tests

The results presented here were obtained following the methodology introduced in Section 6.1.1.

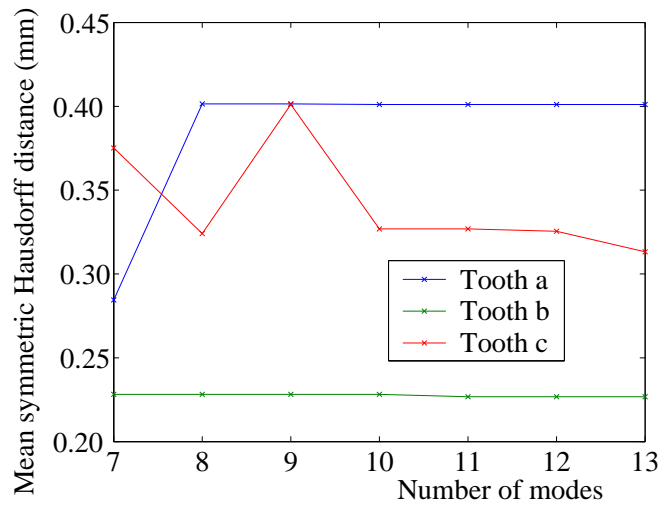


Figure 6.2: Influence of the number of modes on the reconstruction using the crown only.

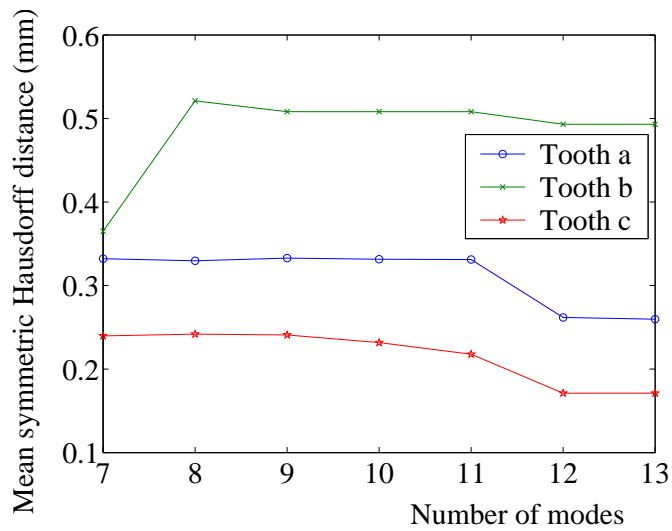


Figure 6.3: Influence of the number of modes on the reconstruction using the root only.

6.3.1 Tooth reconstruction based on the crown only

Tables D.3 and D.4 present the results obtained for the leave-one-out tests using the crown only. For each specimen, the tables indicate the minimum distance to the tooth after reconstruction, the maximum distance, the mean distance as well as the mean square error (RMS). Major results are summarized in Figures 6.4 and 6.6. Figures 6.5 and 6.7 show, from the top to the bottom, the results obtained in the best case, the worst case and the general case. The gray mesh represents the tooth after reconstruction and the red solid the original volume.

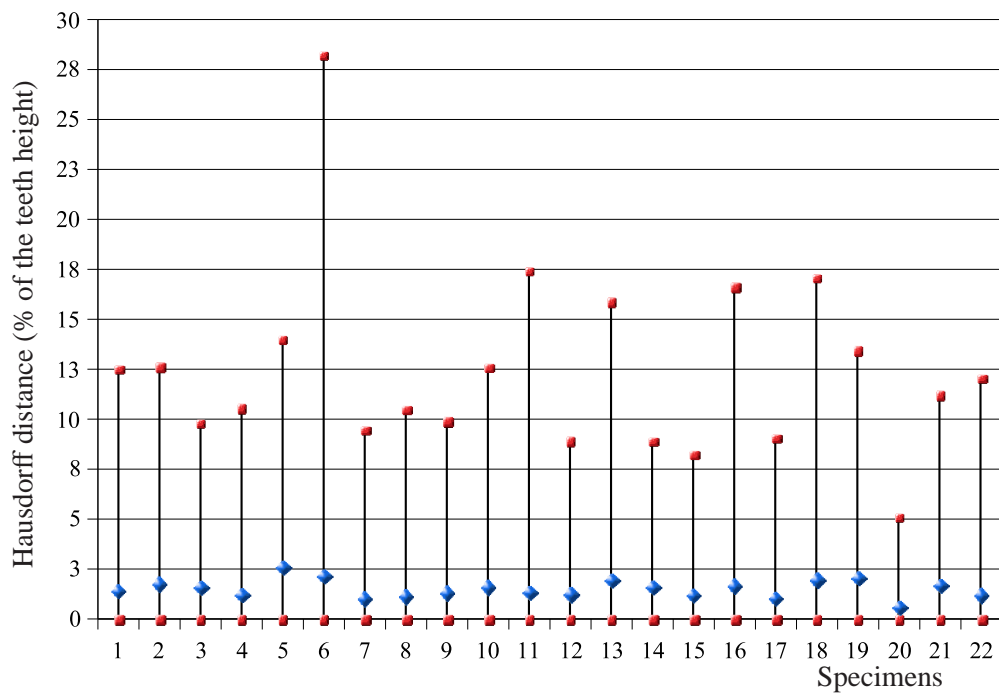


Figure 6.4: Minimum, mean, and maximum HD for each of the 22 specimens, during the leave-one-out test based on crown information (method KD). Extrema are represented by the red dots and mean values by the blue dots.

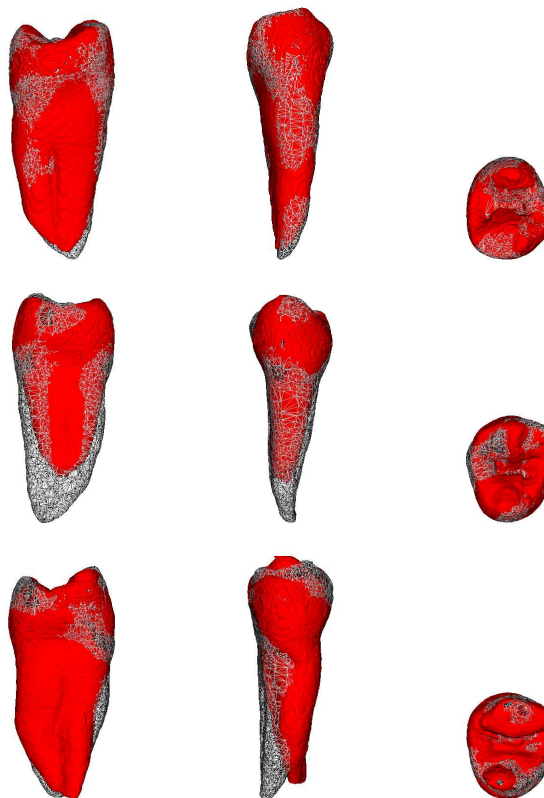


Figure 6.5: Examples of teeth reconstructed using crown information only (method KD). From the top to the bottom, the teeth correspond to the specimens 20, 6 and 2.

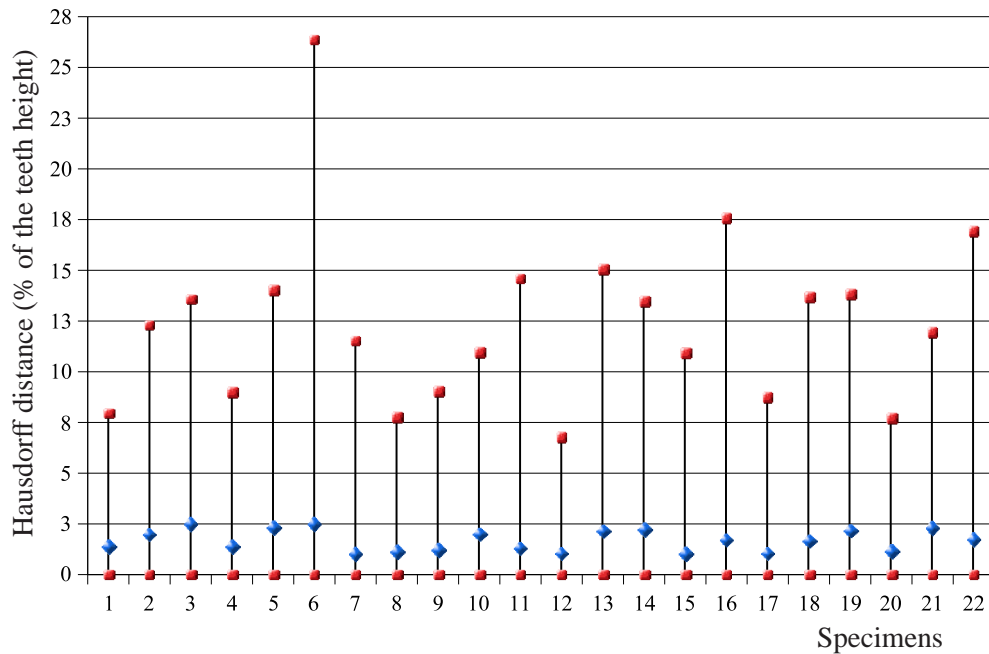


Figure 6.6: Minimum, mean, and maximum HD for each of the 22 specimens, during the leave-one-out test based on crown information (method OD). Extrema are represented by the red dots and mean values by the blue dots.

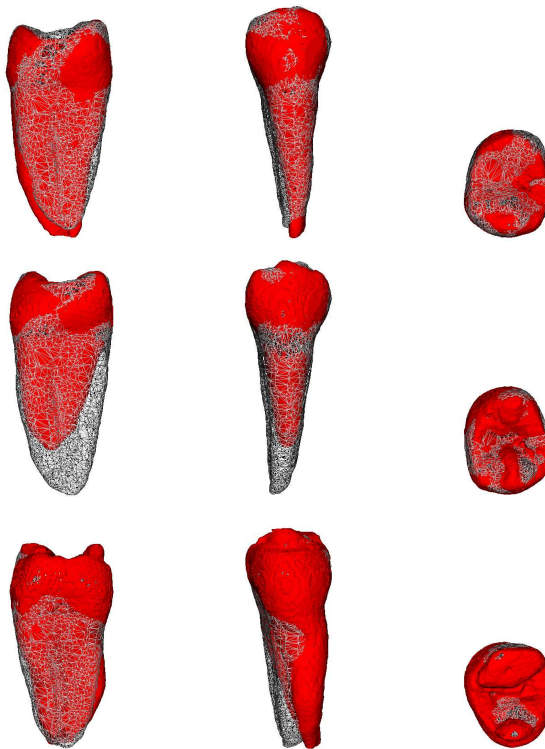


Figure 6.7: Examples of teeth reconstructed using crown information only (method OD). From the top to the bottom, the teeth correspond to the specimens 12, 6 and 2.

6.3.2 Tooth reconstruction based on the root only

Figures 6.8 and 6.10 present the results obtained for the leave-one-out tests using the root only (numerical results can be found in Tables D.3 and D.4). Figure 6.9 and Figure 6.11 show, from the top to the bottom, the results obtained in the best case, the worst case and the general case. The gray mesh represents the tooth after reconstruction and the red solid the original volume.

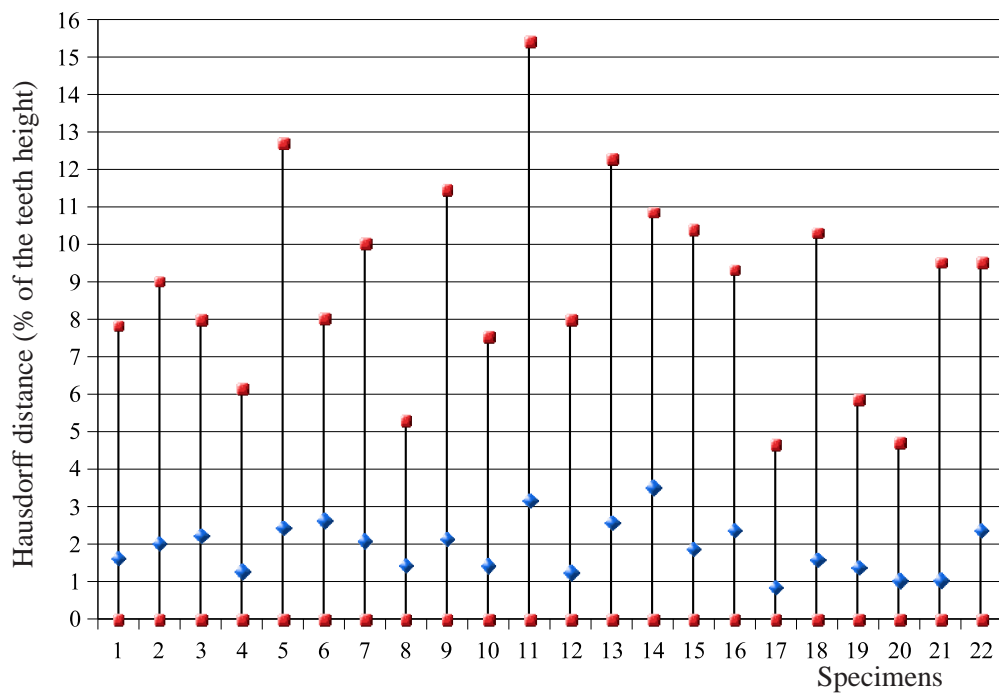


Figure 6.8: Minimum, mean, and maximum HD for each of the 22 specimens, during the leave-one-out test based on root information (method KD). Extrema are represented by the red dots and mean values by the blue dots.

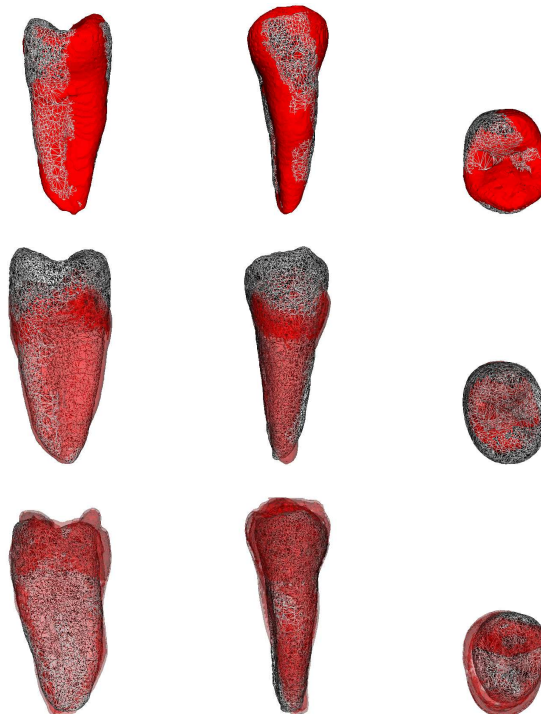


Figure 6.9: Examples of teeth reconstructed using root information only (method KD). From the top to the bottom, the teeth correspond to the specimens 17, 11 and 2.

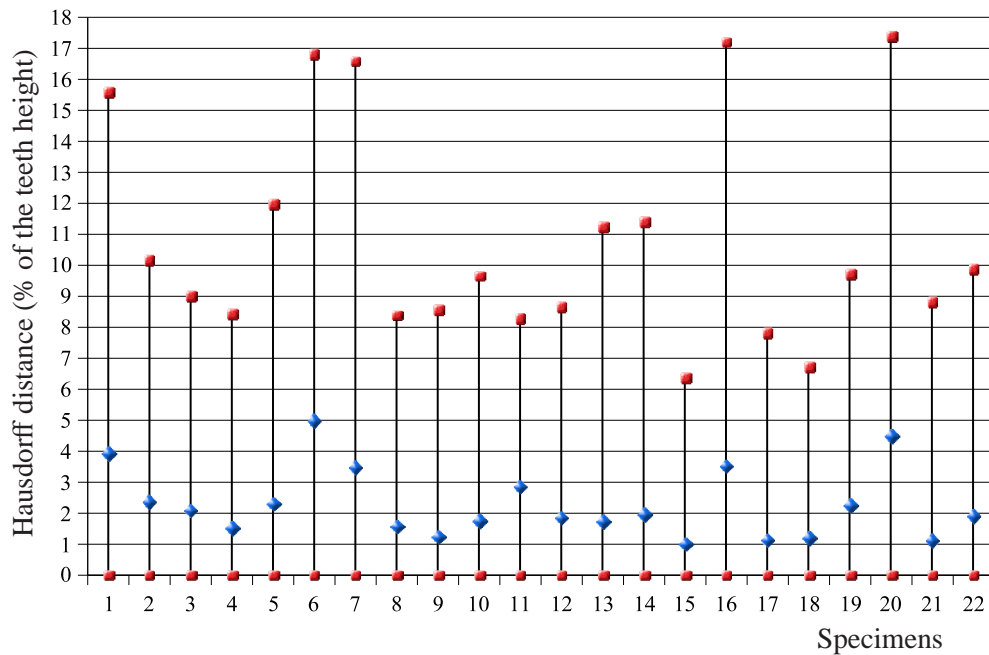


Figure 6.10: Minimum, mean, and maximum HD for each of the 22 specimens, during the leave-one-out test based on root information (method OD). Extrema are represented by the red dots and mean values by the blue dots.

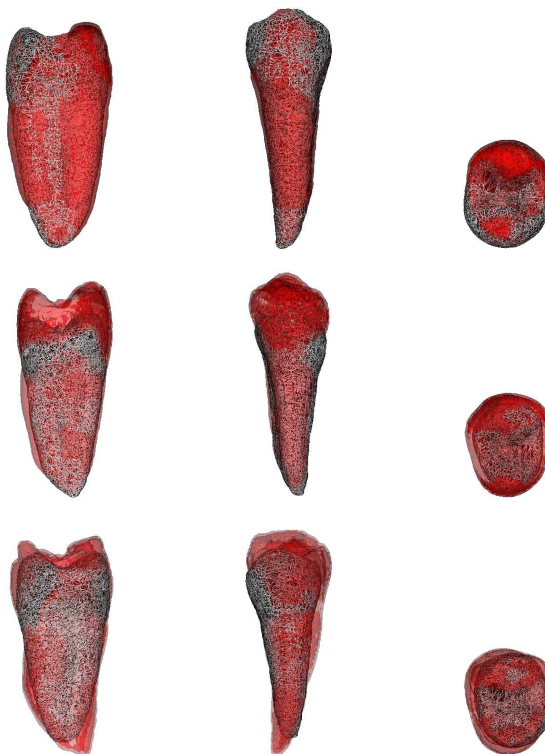


Figure 6.11: Examples of teeth reconstructed using root information only (method OD). From the top to the bottom, the teeth correspond to the specimens 5, 20 and 2.

6.3.3 Discussion

Reconstruction using the crown

As shown by Tables D.1 and D.2, and Figures 6.4 to 6.7, the results are globally satisfactory. The two methods lead globally to results of a similar quality, though the minimum, mean, maximum and RMS Hausdorff distances between a given original specimen and its reconstructed shape can vary according to the method used. This is due to the multiple approximations realized when using method OD.

For all the teeth in the test set, we have a perfect match between the crown of the original teeth T_i and TC_i 's crown. Figure 6.12 shows the Hausdorff distance between T_i and TC_i for the teeth of Figure 6.5, represented on TC_i . For the different teeth, the higher distances will usually appear at the extremity of the root, and never near the crown. The small distances observed in the crown area prove that the degrees of freedom of the reconstruction algorithm are sufficient for the crowns of the mean shape and those of T_i to match and that no problem of local minimum is encountered. On the contrary, the root of the reconstructed tooth is only an approximation of the original root in agreement with the statistics built. Therefore, the results are limited by the existence of a strong correlation between the different parameters describing the shape of a tooth and in particular by the statistical possibility of retrieving the root parameters given only crown information.

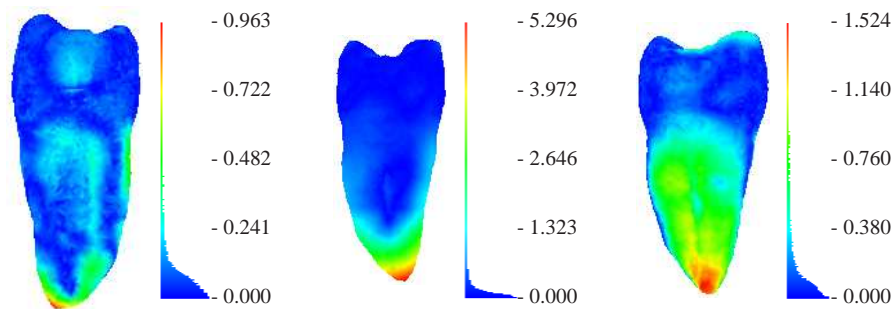


Figure 6.12: Distribution of the Hausdorff distance on the reconstructed shapes (reconstruction based on crown data). Distances are given in mm.

Globally, this method gives a good approximation of the tooth height and those of the tooth width. This proves that the model built gives a good knowledge about the variability among upper right second premolars, and that crown information contains statistically root information. However, one particular tooth in the training set lead to results far from those expected using anyone of the two methods. Indeed, the reconstruction is based on a statistical representation. Consequently, outliers will lead to erroneous results. For most teeth, there exists a correlation between the height and width of the tooth, and the size of the crown (parameters such as shape or width). When this correlation is not respected, the method fails. In the case of specimen 6, the crown is extremely large compared to its height: this leads to a reconstructed model that is much higher than the original one (Figures 6.5 and 6.7, middle tooth). The only way to be totally sure that this correlation is respected would be to compare the output of the reconstruction with a 2D view of the tooth as shown in Section 6.4. If any consideration, such as the external aspect of the tooth or the comparison with an x-ray, makes us think that this correlation does not exist, we would have to use a method where the length and width of the tooth could be introduced as criteria. These criteria would be considered as constraints during the optimization process but they might considerably slow down the reconstruction.

Reconstruction using the root

As shown by Tables D.3 and D.4, and Figures 6.8 to 6.11, the results are far from being reliable. The Hausdorff distances between the original teeth and their reconstructed versions using root information are smaller than those obtained during the reconstructions using crown information, but are higher with respect to the part to infer, i.e., crown in the first case and root in the second case.

Figure 6.13 shows the Hausdorff distance between T_i and TR_i , represented on the reconstructed tooth, for two different specimens. We notice that even the roots of the target and the mean shape do not always match correctly (cf. tooth on the left).

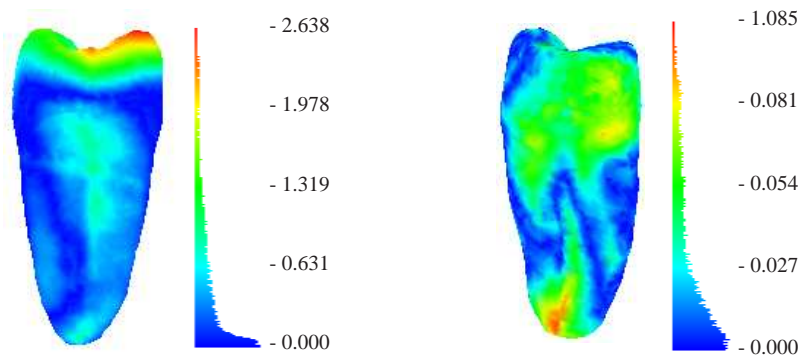


Figure 6.13: Distribution of the Hausdorff distance on the reconstructed shapes (reconstruction based on root data). Distances are given in mm.

The major problems comes from the root shape. Its simplicity makes the matching process more difficult. Indeed, as shown in Figure 6.14, when one tries to minimize the distance between the crown and the mean shape by summing over the root's points, the optimization process may lead to satisfactory results if the target (root only) is bigger than the mean shape (root only). In this case, *A* will match *C* and *B* will match *D*. On the contrary, if the root is smaller, the matching will not work: *C* will match *E* instead of *A*, and *D* will match *F* instead of *B*. Consequently, the root of the reconstructed tooth will be much higher than the original model. One possibility to avoid this problem is the use of feature points on both the target and the mean shape.

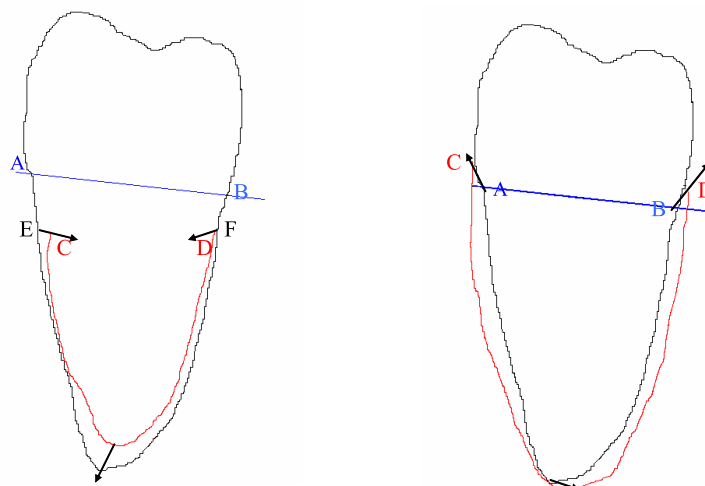


Figure 6.14: Process of matching two roots. The black tooth represents the mean shape and the red tooth the target.

The root brings less information about the tooth global shape than the crown. This lack of information prevents us from inferring the crown shape correctly. Increasing the number of deformation modes usually leads to a higher accuracy, but can also bring too much variability into the reconstruction.

6.4 Reconstruction using patient's data

Another test was then realized using real data. Though the test realized above is realistic in forensic medicine, data used by orthodontists usually present losses at the interstices as shown by Figure 6.15, whatever the segmentation method used to extract the teeth crown.

A study model was digitized using a laser scanner Cyberware Rapid 3D Digitizer Model 3030R-HIREZ. The upper right second premolar was then extracted using Kondo *et al.*'s method of segmentation [23] (Figure 6.16a) and the 3D shape of the tooth obtained (Figure 6.16b). Despite the important loss of information introduced by the segmentation, the reconstruction leads to an excellent match between the original crown and those of the computed tooth. The reconstructed tooth was then combined with the orthopantomogram of the same patient (Figure 6.16c) to test the validity of the shape reconstruction. As shown in Figure 6.16d, the method proposed leads to a very good estimation of the tooth shape and size.

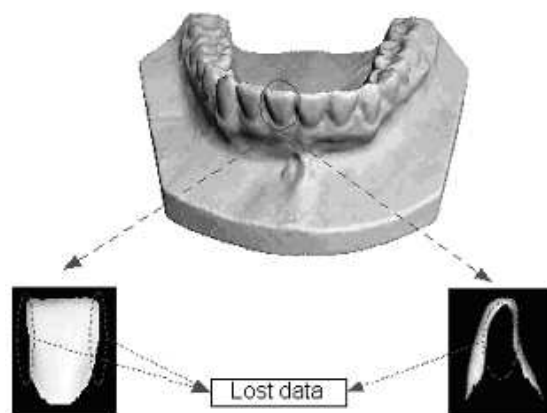


Figure 6.15: The segmentation of teeth from a dental cast affects the crown shape: some information is lost on the buccal view (left image) and distal view (right image) of the crown extracted.

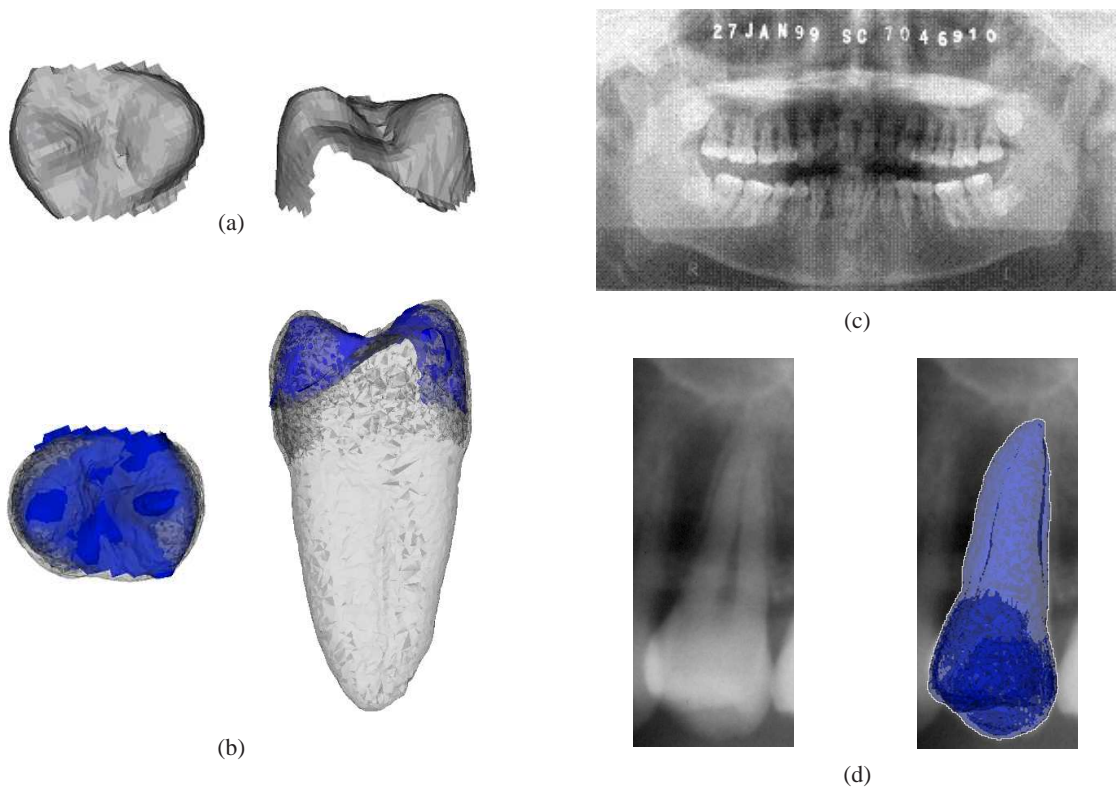


Figure 6.16: Process of fitting a tooth on a dental cast. (a) Crown of the tooth after segmentation. (b) Tooth after reconstruction (gray surface) and original crown (blue surface). (c) Patient's OPG. (d) X-ray of the patient's tooth matched with the tooth reconstructed (blue).

6.5 Reconstruction using feature points

In order to improve the reconstruction results using root information, a test was performed using feature points on both the mean shape and the target. Figure 6.17 shows their location. The left image, which corresponds to the projection of the mesial view of a root, represents the position on the 3 feature points F_1 , F_2 and F_3 for a given tooth. The right image represents a cross-section of the root parallel to the occlusal view and the exact location of F_2 and F_3 .

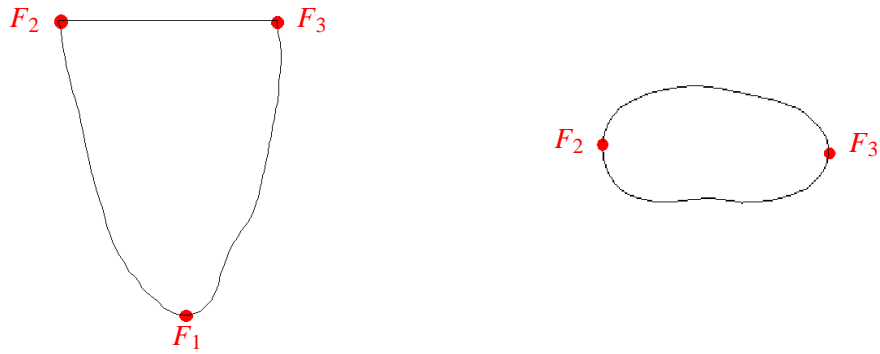


Figure 6.17: Location of the feature points for a reconstruction using root information.

These feature points are automatically defined for the two teeth (F_{1t} , F_{2t} and F_{3t} for the target and F_{1m} , F_{2m} and F_{3m} for the mean shape after deformation). In order to enforce the corresponding feature points to match, a penalty term is added to the objective function. Equation 5.5 is modified such that it becomes

$$E(\mathbf{p}) = \sum_{i=0}^{M_{pdm}-1} \min(\|\mathbf{d}_j - \mathbf{m}_i\|^2)_{1 \leq j \leq K} + W \sum_{j=1}^3 \|F_{jt} - F_{jm}(\mathbf{p})\|^2 \quad (6.5)$$

The weight W in Equation 6.5 has to be chosen such that it penalizes the objective function when the feature points are far apart without minimizing too much the influence of the first term (i.e., the influence of the other points of the target).

Figure 6.18 shows the effect of adding feature points on the reconstruction. The first three teeth images show the result of the reconstruction without the use of feature points. The three images below represent the result of the reconstruction using feature points in the same conditions as previously (same number of modes, same algorithm) with a weight $W = 150$.

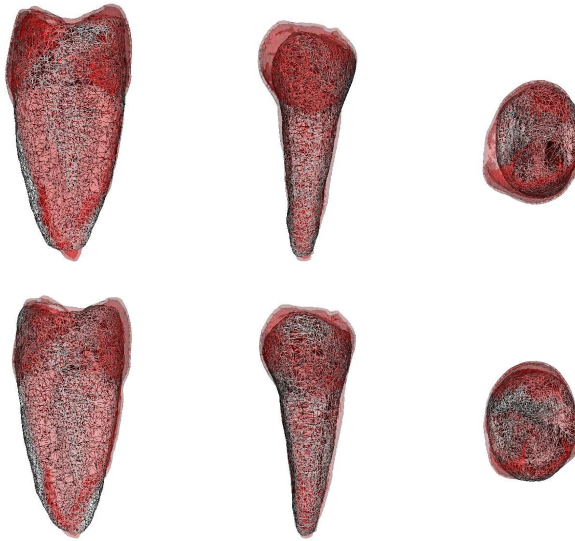


Figure 6.18: Effect of adding feature points on the reconstruction.

The use of features points ensures a better correspondence between the roots of the two teeth (original tooth and reconstructed one). This leads to a better determination of the crown shape. The different experiments carried out showed a slight improvement in the shape determination (the influence is more or less important among the specimens). However, using feature points is not always sufficient to get a good approximation of the tooth shape. As mentioned earlier, the root brings little information about the tooth shape, and the reconstructed tooth may be a poor approximation of the original tooth. We can also notice that the use of feature points has nearly no impact on the reconstruction time.

6.6 Influence of the number of specimens used to define the statistical model

As mentioned previously, 22 specimens were used to build the statistical model. In this experiment, we evaluate the influence of the number of specimens used on the reconstruction.

Different leave-one-out tests (using crown information) were made using a number N of specimens between 2 and 22. For each value of N , p groups were constituted randomly (the value of p as a function of N can be found in Table 6.1). The mean of the mean Hausdorff distance

was then evaluated for each value of N and can be visualized in Figure 6.19. All the experiments were made using the same algorithm (method KD) and 7 modes of deformations except for the tests involving a number of teeth N strictly inferior to 8, where the maximum number of modes available was used². More than 10 teeth seem necessary to get a good precision during the reconstruction. In other words, 10 specimens give a good knowledge about the variability among the kind of tooth tested. However, the variance observed among the results decreases slowly when N increases, leading to more predictable results for a higher value of N . Combining these two informations, 12 teeth would appear to be sufficient.

Table 6.1: Number of tests realized given the number of specimens N .

Number or teeth per group (N)	2	4	6	8	10	12	14	16	18	20	22
Number of groups (p)	12	10	8	6	5	4	3	3	3	2	1

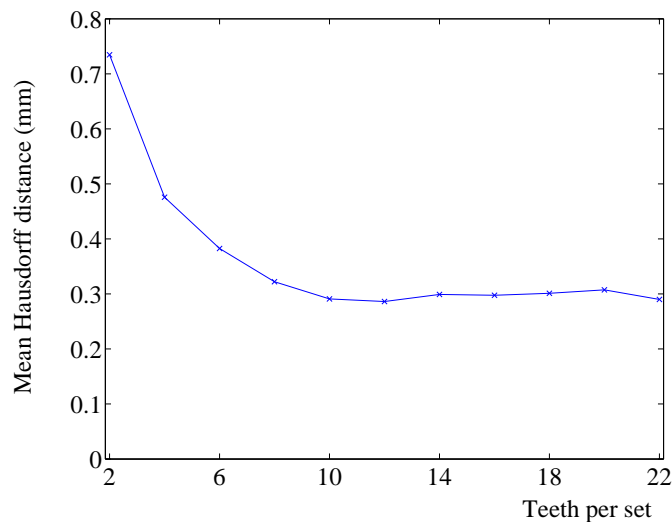


Figure 6.19: Influence of the number of specimens used to build the statistical model.

6.7 Computation time

The computation time for the two methods is now compared (using a Pentium IV 2.4 GHz). It includes the entire registration process, i.e., both the rigid and the elastic registrations. Figure 6.20

²Given a set of N specimens, each statistical model is built using a set of $N - 1$ specimens during the leave-one-out method. The modes of deformation correspond to the eigenvectors of the $(N - 1) \times (N - 1)$ set's autocorrelation matrix. Consequently, only $N - 1$ modes can be defined.

shows the computation time measured during the leave-one-out test. For the reconstruction using the crown or the root, method KD is much faster than method OD. The reconstruction involving the crown takes on average only 26.3 seconds for method KD, against 259 seconds for method OD, and 9.6 seconds when dealing with root information against 51.8 seconds for method OD. Though the distance map allows accessing the minimum distances very quickly, method OD is greatly penalized by the computation time of the gradient. Furthermore, method OD is more affected by an increase in the density of the statistical model than method KD.

Table 6.2 gives the average complexity of the elastic registration for the two algorithms. The average complexities were either obtained by direct estimation or found in the literature [24][25]. However, comparing the computation time for the two methods based on the complexity only is impossible due to the high number of iterations introduced during some steps, e.g., the computation of the gradient for method OD. When using method KD, the computations related to the Levenberg-Marquardt algorithm represent a high proportion of the computation time. On the contrary, the slower steps in method OD correspond to the computation of the distance map and those of the gradient, and the computations related to the Levenberg-Marquardt algorithm represent only a minor percentage of the total computation time. This makes method OD more sensible to an increase in the number of points of the statistical model and those of the target.

Table 6.2: Average complexity of the elastic registration for the two algorithms.

	Method KD	Method OD
Prior to the optimization process.		
Precomputed distance map		$O(M_{pdm} \log M_{pdm})$
For every iteration of the iteration process.		
Kd-tree decomposition	$O(K \log K)$	
Gradient computations	from $O(M_{pdm} \log K)$ to $O(M_{pdm} K)$	$O(K \log M_{pdm})$
Distance estimations	from $O(M_{pdm} \log K)$ to $O(M_{pdm} K)$	$O(K \log M_{pdm})$
Levenberg-Marquardt algorithm (other computations)	$O(1)$	$O(1)$
Application of the transformation	$O(K)$	$O(K)$

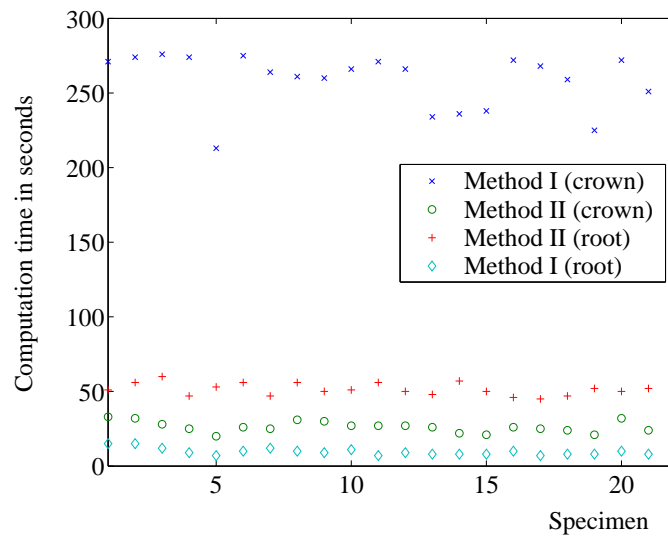


Figure 6.20: Computation time of the two methods.

6.8 Effect of the target's density on the reconstruction

All the results presented above involve specimens whose resolution is close to the resolution of the statistical model. However, such a precision is not always available. In order to study the influence of the density, the crown of a specimen was decimated leading to new specimens S_0, \dots, S_5 having a number of points varying from 2,500 to 16,000. Each of these specimens was used to reconstruct the original tooth with method KD. Figure 6.21 shows the Hausdorff distance between the original tooth and its reconstructions based on S_0, \dots, S_5 .

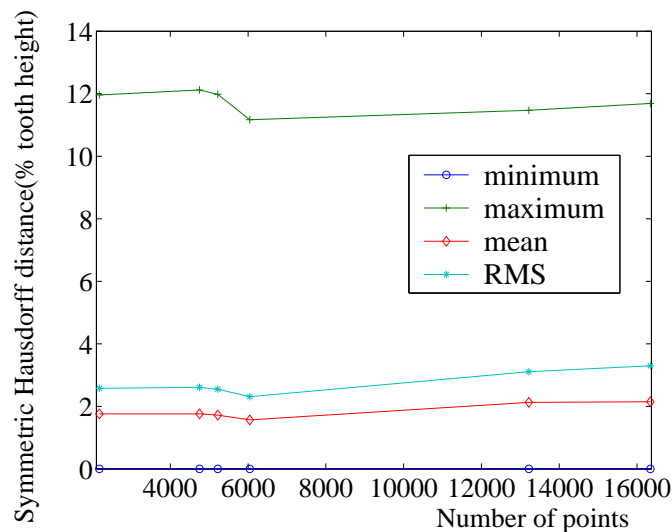


Figure 6.21: Influence of the density on the reconstruction accuracy.

Decreasing the density of the target does not seem to have a real influence on the results in the range studied. As long as the cloud of points gives a good information about the global form of the crown, the quality of the output is not really affected. Similar results were observed using other specimens.

6.9 Summary

The different tests realized helped us determining the optimal number of modes necessary to perform the reconstructions (Section 6.2) as well as the minimum number of specimens required to build a statistical model for the kind of tooth tested (Section 6.6). 12 specimens would appear to be sufficient and increasing the number of specimens further may not improve the results.

Leave-one-out tests (Section 6.3) proved that the statistical model built gives us a good knowledge about the variability among upper right second premolars and that both methods investigated lead globally to similar results, when using either crown or root information.

The approach proposed proved to be efficient to reconstruct a tooth given crown information (Section 6.3.1), even if information about the crown shape is lost due to the segmentation process when using patients' dental casts (Section 6.4) or when the density of the patients' teeth are much lower than those of the specimens used (Section 6.8).

When using root information, the system failed to retrieve the teeth correct shape (Section 6.3.2). The variability observed among the results obtained makes this method unsuitable for shape recovery. However, the introduction of feature points (Section 6.5) can slightly improve the reconstruction. The performance of this method is still limited by the poor amount of information given by the root.

Finally, Section 6.7 showed the superiority of method KD over method OD. Though the two methods lead to results of a similar quality, method KD is much faster than method OD. Though using a distance map decreases the time necessary to compute the smallest distances from a point

of a volume A to the closest point of a volume B, method OD is greatly penalized by the amount of time necessary to compute the gradients.

Chapter 7

Conclusion

We have presented a method to reconstruct a tooth using partial information about its shape, and without the use of x-rays, CT or MRI. The only necessary information consists in a simple cloud of points. The method consists in two major phases: we first need to define a statistical model (mean shape + modes of deformation) to represent the kind of tooth we want to reconstruct (this model is built only once using a set of specimens). During the second phase, the mean shape is first aligned with the target using an Iterative Closest Point Algorithm, and the modes of the deformable model are optimized to match the target. The aim is to exploit the whole prior information available for a given tooth and minimize the interaction with the user.

The different experiments conducted showed us that this method was suitable for tooth shape recovery when we have only crown information at our disposal. The reconstructed tooth gives us a good approximation of the width and height of the target as long as its shape respects the statistical variations. However, with only root information, the system fails to retrieve the correct shape. This problem is due to the particular topology of the root and to the shortage of information provided by the tooth. Using feature points on the root of the mean shape and those of the target can slightly improve the reconstruction process. The two methods that were tested (with and without the use of a distance map to compute the distances) lead to a similar precision but

the first one proved to be much faster. To finish with, the density of the input did not seem to have a major impact on the reconstruction.

A few improvements could be brought to this method. The quality of the method depends highly on the PDM's definition. To define its modes, we first need to find a point to point correspondence between a tooth considered as a generic model and all the other teeth of the set (T_0, \dots, T_N) . However, the method lets appear some discrepancies when the form of the generic model is far from those of T_i . In this case, we lose some information about T_i 's shape. A more accurate method could give a better knowledge of the statistical variations among the teeth of a given type, leading to more accurate modes of deformation.

The algorithm could be accelerated by using a more powerful algorithm when realizing the automatic rigid registration [26]. Improved Kd-trees could also speed up the elastic registration process [27] when using method KD.

The method describes above supposed that only information about either the crown or root is known. When supplementary information is available, such as x-ray of the tooth to reconstruct when dealing with patient's data, we could try to integrate this information in order to tune the parameters of the reconstructed model more precisely.

So far, the method described was developed using only second upper-right premolars. However, it could easily be extended to all single-rooted teeth. Furthermore, generating a new PDM per tooth is not absolutely necessary: mirror teeth can be reconstructed using the same training set, since the mirror PDM could be created using the mirror view of the mean shape and those of the different modes. Teeth whose shapes differ only in size (e.g., first and second premolar) could also use the same data base (simple scaling of the mean shape and different modes of the PDM).

Appendix A

The Levenberg-Marquardt algorithm

Suppose that we are fitting $3 \times N$ data points $(\vec{\mathbf{x}}_i, \vec{\mathbf{y}}_i)$ (N represents the number of points in a 3D dimensional space). The relation between the two sets $X = \{\vec{\mathbf{x}}_0, \dots, \vec{\mathbf{x}}_N\}$ and $Y = \{\vec{\mathbf{y}}_0, \dots, \vec{\mathbf{y}}_N\}$ is given by a function y such that $y(\vec{\mathbf{x}}) = y(\vec{\mathbf{x}}; \vec{\mathbf{a}})$.

A.1 Function approximation

Given a function f , scalar, that depends on different real variables represented by the vector $\vec{\mathbf{a}}$, in a coordinate system centered at $\vec{\mathbf{x}}_0$, we can approximate f at $\vec{\mathbf{x}}$ near $\vec{\mathbf{x}}_0$ using the Taylor series expansion about $\vec{\mathbf{x}}$:

$$\begin{aligned} f(\vec{\mathbf{x}}) &= f(\vec{\mathbf{x}}_0) + \sum_i \frac{\partial f}{\partial x_i}(\vec{\mathbf{x}}_0) x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j}(\vec{\mathbf{x}}_0) x_i x_j + \dots \\ &\approx c - \vec{\mathbf{b}} \cdot \vec{\mathbf{x}} + \frac{1}{2} \vec{\mathbf{x}} \mathbf{H} \vec{\mathbf{x}} \end{aligned} \tag{A.1}$$

where $c \equiv f(\vec{\mathbf{x}}_0)$, $\vec{\mathbf{b}} \equiv -\nabla f(\vec{\mathbf{x}}_0)$, and $H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(\vec{\mathbf{x}}_0)$ are the components of the Hessian matrix \mathbf{H} of f at $\vec{\mathbf{x}}_0$.

If the approximation in Eq. A.1 is exact¹, we can compute the gradient of f directly:

$$\nabla f = \mathbf{H} \vec{\mathbf{x}} - \vec{\mathbf{b}} \tag{A.2}$$

¹in this case, the function f is said to be quadratic

and the gradient will be zero (extremum of the function) when \vec{x} satisfies: $\mathbf{H} \vec{x} = b$.

If Eq. A.1 is only an approximation of f , (A.2) gives a direct formula to converge toward a minimum of f :

$$\vec{x}_{\min} = \vec{x}_0 - \mathbf{H}^{-1} \nabla f(\vec{x}_0) \quad (\text{A.3})$$

However, (A.1) can be a bad approximation. In this case, we can move toward the minimum iteratively (steepest descent minimization):

$$\vec{x}_1 = \vec{x}_0 - \text{Constant} \times \nabla f(\vec{x}_0) \quad (\text{A.4})$$

A.2 Gradient and Hessian of χ^2

Suppose the function model to be fitted is given by $y = y(\vec{x}, \vec{a})$ (we want to adjust the parameters $\vec{a} = \{a_1, a_2, \dots, a_M\}$ of the function). The χ^2 merit function is just the sum of the distances between the points of the two sets X and Y :

$$\chi^2(\vec{a}) = \sum_{i=1}^N [y_i - y(x_i, \vec{a})]^2 \quad (\text{A.5})$$

χ^2 is a form non necessarily quadratic, defined on the vectorial space of the parameters to adjust (space of dimension M).

Deriving (A.5) with respect to the components of \vec{a} , we can easily express the gradient components:

$$\frac{\partial \chi^2}{\partial a_k} = -2 \sum_{i=1}^N \left[(y_i - y(x_i, \vec{a})) \frac{\partial y(x_i, \vec{a})}{\partial a_k} \right] \quad (\text{A.6})$$

Taking an additional partial derivatives, we get the components of the Hessian matrix:

$$\frac{\partial^2 \chi^2}{\partial a_k \partial a_l} = 2 \sum_{i=1}^N \left[\frac{\partial y(x_i, \vec{a})}{\partial a_k} \frac{\partial y(x_i, \vec{a})}{\partial a_l} \right] \quad (\text{A.7})$$

It is conventional to use the following notations in order to simplify the equations: $\beta_k \equiv -\frac{1}{2} \frac{\partial \chi^2}{\partial a_k}$ and $\alpha_{kl} \equiv \frac{1}{2} \frac{\partial^2 \chi^2}{\partial a_k \partial a_l}$ (this is equivalent to: $\vec{\beta} = -\frac{1}{2} \nabla \chi^2$ and $[\alpha] = \frac{1}{2} \mathbf{H}$). With these notations and taking $\delta \vec{a} = \vec{a}_{\min} - \vec{a}_0$ ($\delta \vec{a}$ is the incrementation to realize on the initial set of parameters \vec{a}_0 in

the initial convergence scheme (A.3)), the minimization of a function considered as quadratic is equivalent to the resolution a linear system:

$$[\alpha] \delta \vec{a} = \vec{\beta} \quad (\text{A.8})$$

However, when we are far from the quadratic form, the steepest descent is simply:

$$\delta \vec{a} = \text{Constant} \times \vec{\beta} \quad (\text{A.9})$$

A.3 Levenberg-Marquardt strategy

Levenberg and Marquardt proposed an efficient and ingenious method to switch continuously between the inverse-Hessian method and the steepest descent scheme. This second method can be used at the beginning of the minimization (when we are still far from the solution) and is progressively replaced by the inverse-Hessian method when we come closer to the minimum.

We can briefly summarize the Levenberg-Marquardt method as a strategy to find the minimum of χ^2 uniting the two schemes (A.8) and (A.9) at best, thanks to two major ideas. The first idea consists in modifying the steepest descent algorithm (A.9) by replacing the stepsize (Constant) by a vector whose components are judiciously chosen. We can interpret this choice as a scaling, for every parameter, of the stepsize taken in the direction of χ^2 minimum. We realize this choice by noticing that the constant of proportionality between a partial derivative with respect to a_k and a finite difference in a_k has naturally the dimension of a_k^2 . Furthermore, we postulate that a magnitude order of this constant can be given by a component of the matrix $[\alpha]$. Since the only component of $[\alpha]$ dependent of a_k that has the required dimension is $\frac{1}{\alpha_{kk}}$, (A.9) has to be replaced by:

$$\delta a_l = \frac{1}{\lambda \alpha_{ll}} \beta_l \quad (\text{A.10})$$

where λ is a factor > 1 allowing to reduct globally (and not component by component) the stepsize when it becomes too big (as in the steepest descent method). The second idea consists in setting $[\alpha]' = [\alpha] + \lambda \mathbf{I}_d$, with \mathbf{I}_d the identity matrix $M \times M$. The two schemes (A.8) and (A.10)

Appendix A - The Levenberg-Marquardt algo.

can be replaced by a single formula: we need to find the stepsize $\delta\vec{a}$ such that:

$$[\alpha]' \delta\vec{a} = \vec{\beta} \tag{A.11}$$

Note: $(\lambda \rightarrow \infty) \Rightarrow (\text{Eq A.11} \Leftrightarrow \text{Eq A.10})$ $(\lambda \rightarrow 0) \Rightarrow (\text{Eq A.11} \Leftrightarrow \text{Eq A.8})$

For more information about the implementation of the Levenberg-Marquardt algorithm, the reader can refer to : “Numerical Recipes in C++”[19].

Appendix B

Computing a distance map

As mentioned in Chapter 4, we aim at deforming a mean model to match a target. However, to realize the minimization process, we need, for each point of the target, to be able to find the closest point on the deformable model. To do so, one method consists in using a distance map (based on an octree decomposition) as defined by Szeliski and Lavallée in [28].

B.1 The octree-spline decomposition

B.1.1 The octree decomposition

The two volumes (target and deformable model) are enclosed in a cubic bounding box and a classical octree decomposition is realized, based on the points of the patient's tooth, denoted as \mathcal{S} .

The bounding box is then iteratively decomposed into smaller cubes (octants) until the maximum number of levels is reached (inversely proportional to the resolution of the decomposition). Each octant can be either white, gray or black. It is considered as white if no point of the volume is enclosed in it, gray if at least one point of the volume is enclosed in the octant but its level is strictly inferior to the maximum level, and black in the last case (octant non empty but the maximum number of levels has been reached). If an octant is either white or black, no further

subdivision is done, as shown in Figure B.1. Otherwise, it is subdivided into 8 children of length half those of the current octant. Figure B.2 shows the example of an octree decomposition realized on a tooth root using a resolution equal to 3.

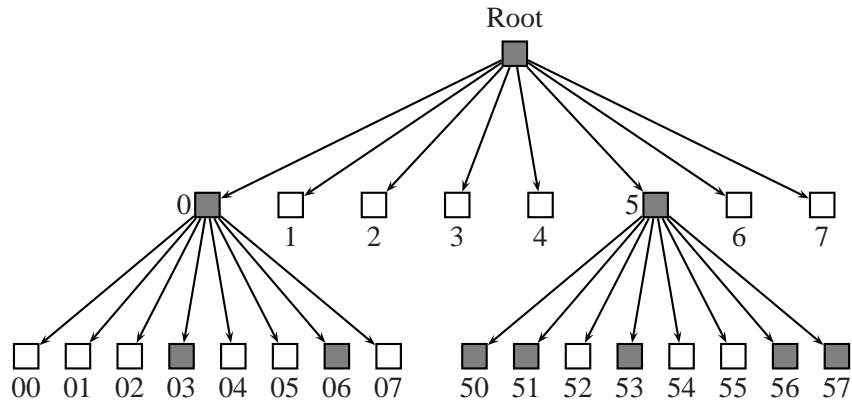


Figure B.1: Tree Representation.

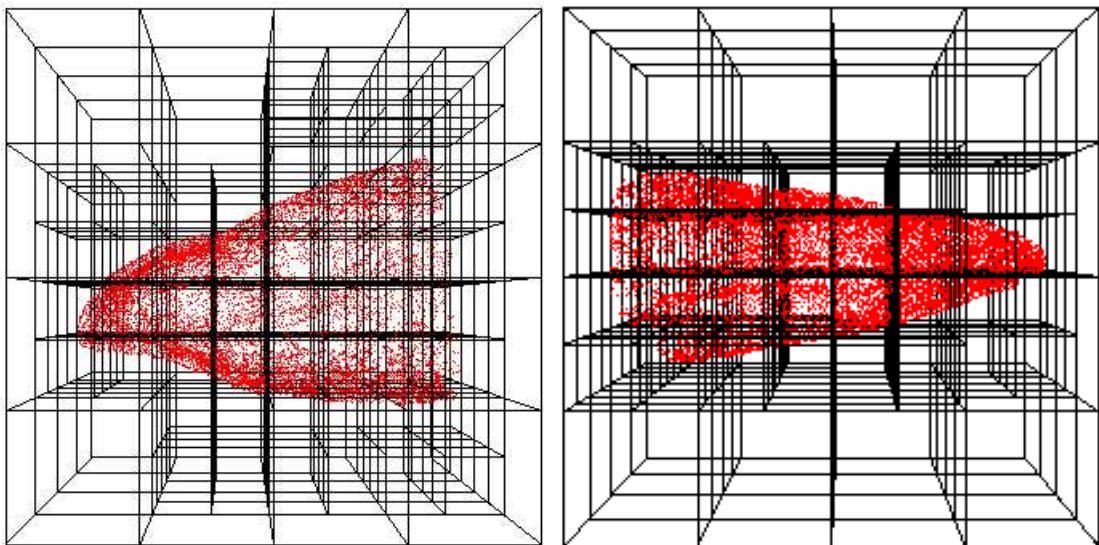


Figure B.2: Example of octree decomposition applied on a tooth root (red cloud of points) using 3 levels of decomposition. The two figures show the same decomposition from a different viewpoint.

B.1.2 Computation of the distance map

The simplest distance map consists of a uniform division of the space. However, this would lead to a huge amount of distances to compute and store. Furthermore, a higher accuracy is usually required near the surface rather than far from it. The octree decomposition is an efficient and

easy way to satisfy this last criterion while minimizing the amount of memory necessary.

1. The first step of the distance map computation corresponds to the octree decomposition defined in Section B.1.1.
2. Octants lying near the surface of the target are then refined when necessary. Indeed, due to the nature of the decomposition, a huge number of empty nodes can be present near the surface, which is in contradiction with the aim of the octree decomposition (having a higher accuracy near the surface). To realize this refinement, the octree is visited using a post-order traversal and the neighbors of every node are computed using Bhattacharya's method (see Section B.2).
3. For each corner of the terminal octants, the minimum distance to the patient's tooth is computed and stored. The octree is traversed in a bottom-up fashion, and the minimum distances computed for every terminal nodes. Simple geometrical considerations are made to avoid useless computations. After determination of the distances for a given octant, these distances are reported to the octant's neighbors. Once all the distances have been computed, they are reported to the terminal nodes parents.
4. At the end, the distance between a new point and the enclosed volume will be computed using a trilinear interpolation (Eq. B.1). Without further modifications, discontinuities (cracks) could appear on the contact surfaces between two terminal octants (either edge or face). Though these discontinuities do not represent a real problem if we only need the distances, they could have a negative impact when trying to compute the gradient. To eliminate these cracks, the octree is traversed using in a top-down breadth-first fashion. For every octant traversed of size s_1 , we defined its edge and face neighbors. If one of its corner c lies on the edge e or the face f of an octant of size s_2 such that $s_2 > s_1$, the distance stored for c is replaced by the linear (edge neighbor) or bilinear (face neighbor)

interpolation on e or f . Tables defining the edges and faces to check for a given octant were defined to minimize the number of tests to realize.

5. Given a new point \mathbf{P} , calculating the minimum distance $d(\mathbf{P})$ from \mathbf{P} to \mathcal{S} only requires to find the octant \mathcal{O} the points belongs to (using a simple top-down traversal). The local normalized coordinates (u, v, w) of \mathbf{P} with respect to the octant are used to realize a trilinear interpolation over the 8 corners of this octant.

$$d(\mathbf{P}) = \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 b_i(u) b_j(v) b_k(w) d_{ijk} \quad (\text{B.1})$$

where d_{ijk} are the distances stored for \mathcal{O} 's 8 corners and $b_l(t) = \delta_{lt} + (1 - \delta_{lt})$ (δ being the Kronecker symbol).

B.2 Finding the neighbors of an octant

This method allows us to obtain a good approximation of the required distances. The major difficulty lies in the neighbor finding, required during steps 2, 3 and 4. An efficient method is crucial since its represents a major part of the computation time. One way of doing so is described by Bhattacharya in [29]. During the octree's construction, all the octants are given a specific index $a_1 a_2 \dots a_m$ where m represents the level, and $a_i \in \{000, 001, 010, 011, 101, 110, 111\}$, $1 \leq i \leq m$ (the value of a_i depends on the octant position with respect to its parent).

Each octant \mathcal{O} has at most 6 face neighbors, 12 edge neighbors and 8 vertex neighbors. To retrieve the neighbors, we just need to retrieve their index. Two tables are defined for each case (face, edge or vertex neighbor):

- the first one allows us determining if the neighbor for a chosen direction belongs or not to the same parent.
- the second one gives us the last value a_m of the neighbor index.

If the neighbor belongs to the same node, we just need to modify a_m to get the neighbor's index.

Otherwise, we have to change the last three bits and iterate the process with O 's parent (octant of index $a_1 a_2 \dots a_{m-1}$).

Appendix C

Kd-Tree Decomposition

To speed up the determination of the nearest neighbors, we can prestructure the data by creating a *Kd-Tree*. This tree is built only once. Then we can determine the nearest neighbors using the special structure of the tree to limit the number of distance computations.

C.1 Structure and building of a Kd-tree

A Kd-tree is a binary tree used to represent data of dimension d (in our example $d = 3$). Each node of the binary tree represents a subset of the data records $S = \{\mathbf{x}_i \mid i = 1, \dots, n\}$ and a partitioning of that subset. Each non-terminal node has two children that represent the two subsets defined by the partitioning. The terminal nodes represent mutually exclusive small subsets of that records. A Kd-tree divides the space into a collection of hyperrectangles that correspond to the terminal nodes (Fig. C.1 and Fig. C.2 present a 2D space subdivision as well as the possible structure of the corresponding Kd-Tree).

To build a Kd-Tree, we consider that all the points belongs to a hyperrectangle of infinite dimension. At each step of the construction (for every node), we need to define a direction d_i along which to split the space as well as a pivot (a point belonging to our subset). The hyperrectangle will be divided into a left hyperrectangle and a right hyperrectangle. Points of the node subset

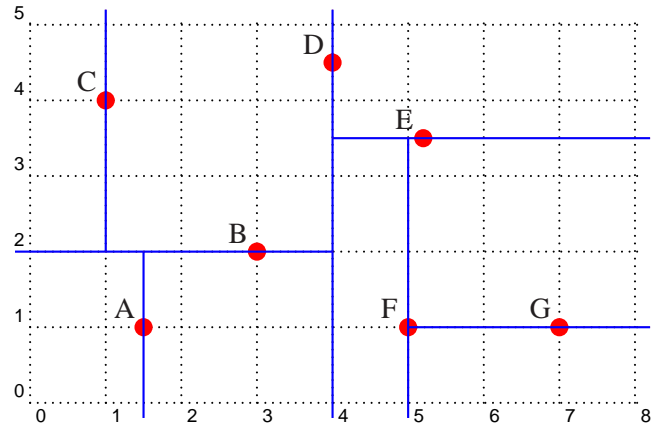


Figure C.1: An example of space subdivision in dimension 2 using a Kd-Tree ($d = 2$): the red dots represent points in a 2D space and the blue lines delimit the borders between the different subspaces.

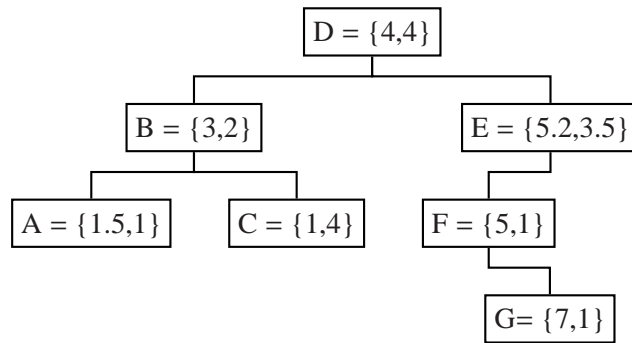


Figure C.2: Example of a Kd-Tree decomposition ($d = 2$). The nodes refer to Figure C.1.

(pivot excluded) whose i^{th} coordinate is smaller than the i^{th} coordinate of the pivot are classified as belonging to the right hyperrectangle and the others to the left one. A question remains: how can we determine the splitting direction as well as the pivot? Different possibilities exist and we chose the following that gives more square regions (increasing the speed of the nearest neighbor search) and gives a tree reasonably balanced:

- The direction is chosen such that it maximizes the range of the \mathbf{x}_i 's.
- The pivot is chosen such that it is in the middle of the most spread dimension.

More information about Kd-trees can be found in Bentley's article [30].

C.2 Nearest Neighbor Search

The algorithm used is described by Moore in [31]. We first need to determine to which leaf the point \mathbf{x} to classify belongs. To do so, we need to travel down the tree using its particular structure.

Once the leaf node has been determined, we can compute the distance between \mathbf{x} and the leaf node point \mathbf{l}_n . However, \mathbf{l}_n is not necessarily the nearest neighbor, even if it may give a good approximation. If a closer point exists, it must lie in a hypersphere of radius $\|\mathbf{l}_n - \mathbf{x}\|^1$ centered at \mathbf{x} . We need to go up to visit the parent's node and check if it is possible or not to have a closer solution in the parent's other child. If no closer neighbor exists in the other child, we can move up a further level. In the other case, we need to explore recursively the other child.

To check if a better solution exists, we need to have a method to determine if a hypersphere hs centered at \mathbf{x} with radius r intersects or not a hyperrectangle hr . To do so, we represent hr by two arrays: one for the maximum coordinates, the other one for the minimum coordinates. We define the vertex $\mathbf{p} = (p_1, p_2, \dots, p_d)$ of hr that is closer to \mathbf{x} as:

$$p_i = \begin{cases} hr_i^{min} & \text{if } t_i \leq hr_i^{min} \\ t_i & \text{if } hr_i^{min} < t_i < hr_i^{max} \\ hr_i^{max} & \text{if } t_i \geq hr_i^{max} \end{cases}$$

hs intersects hr if and only if $\|\mathbf{p} - \mathbf{x}\| \leq r$

The algorithm is summarized below: **nearest** \equiv nearest point to the target \mathbf{x} and $dist \equiv$ the distance between \mathbf{x} and **nearest**.

NearestNeighbor(Node kd , Target \mathbf{x} , Hyperrectangle hr , float max_dist)

If kd is empty, $dist = \infty$ and exit.

$s :=$ splitting direction and **pivot** :=node's representative point.

Cut hr into two hyperrectangles hr_left and hr_right .

If ($\mathbf{x} \in hr_left$)

$nearer_kd := kd_left_child$ and $nearer_hr := hr_left$

$further_kd := kd_right_child$ and $further_hr := hr_right$

else

¹denotes the Euclidean distance

nearer_kd := *kd_right_child* and *nearer_hr* := *hr_right*

further_kd := *kd_left_child* and *further_hr* := *hr_left*

Recursively, call `NearestNeighbor(nearer_kd, x, nearer_hr, max_dist)` and
nearest and *dist*

max_dist := `min(max_dist, dist)`

If the hypersphere of radius *max_dist* centered at **x** intersects *further_kd*

If $\|\mathbf{pivot} - \mathbf{x}\| < \mathit{dist}$

nearest := value and class of the node's representative point.

dist := $\|\mathbf{pivot} - \mathbf{x}\|$

max_dist := *dist*

Recursively, call `NearestNeighbor(further_kd, x, further_hr, max_dist)`

and store the results in **temp_nearest** and *temp_dist*.

If $\mathit{temp_dist} < \mathit{dist}$, **nearest** := **temp_nearest** and *dist* := *temp_dist*.

Appendix D

Leave-one-out tests: numerical results

Table D.1: Leave-one-out test on the specimens' set using the crowns only (method KD).

Specimen	Hausdorff distances							
	Distances				Percentage of the tooth height			
	Min	Max	Mean	RMS	Min	Max	Mean	RMS
1	1.3e-5	2.56	0.28	0.49	6.1e-5	12.48	0.39	2.39
2	2.4e-6	2.63	0.37	0.54	1.1e-5	12.65	1.78	2.60
3	1.8e-5	1.91	0.30	0.45	5.6e-5	9.77	1.56	2.30
4	1.1e-4	2.05	0.23	0.38	5.6e-4	10.57	1.19	1.98
5	6.1e-6	2.80	0.51	0.77	3.0e-5	13.96	2.56	3.85
6	3.5e-6	5.30	0.40	0.86	1.9e-5	28.19	2.13	4.49
7	7.0e-6	1.90	0.21	0.34	3.5e-5	9.43	1.04	1.71
8	1.1e-5	2.07	0.23	0.40	5.3e-5	10.46	1.15	2.00
9	2.2e-5	1.70	0.23	0.33	1.3e-4	9.91	1.33	1.95
10	2.1e-5	2.24	0.29	0.46	1.2e-4	12.57	1.62	2.57
11	1.8e-5	2.97	0.22	0.43	1.0e-4	17.40	1.30	2.49
12	1.1e-5	1.84	0.26	0.40	5.1e-5	8.93	1.25	1.95
13	1.1e-5	3.25	0.39	0.70	5.1e-5	15.90	1.92	3.42
14	.1e-5	1.73	0.31	0.49	1.1e-4	8.87	1.57	2.53
15	1.1e-5	1.51	0.21	0.31	5.7e-5	8.21	1.17	1.71
16	2.6e-6	3.52	0.36	0.66	1.2e-5	16.64	1.68	3.12
17	3.1e-6	1.81	0.20	0.33	1.5e-5	9.03	1.01	1.65
18	3.5e-6	3.05	0.35	0.61	2.0e-5	17.05	1.98	3.41
19	7.0e-6	2.61	0.39	0.63	3.6e-5	13.46	2.03	3.26
20	1.1e-5	0.96	0.11	0.16	5.7e-5	5.07	0.56	0.84
21	2.6e-4	2.55	0.38	0.61	1.1e-3	11.23	1.65	2.67
22	1.9e-6	2.31	0.23	0.38	9.6e-6	12.02	1.22	1.96
Min	1.9e-6	0.96	0.11	0.16	9.6e-6	5.07	0.56	0.84
Max	2.6e-4	5.30	0.51	0.86	1.1e-3	28.19	2.56	4.59
Mean	2.6e-5	2.42	0.29	0.49	1.3e-4	12.44	1.50	2.50

Table D.2: Leave-one-out test on the specimens' set using the crowns only (method OD).

Specimen	Hausdorff distances							
	Distances				Percentage of the tooth height			
	Min	Max	Mean	RMS	Min	Max	Mean	RMS
1	9.6e-6	1.63	0.29	0.42	4.7e-5	7.95	1.42	2.05
2	3.2e-5	2.55	0.41	0.57	1.5e-4	12.30	1.97	2.74
3	3.5e-5	2.65	0.49	0.74	1.8e-4	13.57	2.53	3.78
4	1.9e-6	1.76	0.27	0.41	1.0e-5	9.03	1.41	2.12
5	6.3e-8	2.83	0.47	0.72	3.1e-7	14.07	2.35	3.61
6	5.4e-6	4.95	0.48	0.97	2.9e-5	26.37	2.53	5.14
7	4.2e-6	2.32	0.21	0.36	2.1e-5	11.53	1.05	1.81
8	2.4e-5	1.55	0.23	0.34	1.2e-4	7.80	1.16	1.73
9	8.8e-6	1.55	0.21	0.31	5.1e-5	9.07	1.25	1.82
10	3.3e-5	1.96	0.35	0.54	1.9e-4	11.00	1.99	3.03
11	3.0e-5	2.49	0.22	0.40	1.7e-4	14.59	1.29	2.35
12	6.0e-6	1.41	0.21	0.32	2.9e-5	6.82	1.04	1.54
13	6.7e-6	3.09	0.44	0.74	3.3e-5	15.09	2.14	3.61
14	9.9e-6	2.64	0.44	0.73	5.1e-5	13.51	2.25	3.74
15	2.2e-5	2.02	0.20	0.46	1.2e-4	10.97	1.06	2.48
16	3.1e-6	3.73	0.36	0.66	1.4e-5	17.62	1.70	3.13
17	4.9e-6	1.76	0.21	0.34	2.5e-5	8.78	1.04	1.71
18	8.0e-6	2.46	0.30	0.47	4.5e-5	13.72	1.65	2.60
19	7.9e-6	2.69	0.42	0.69	4.1e-5	13.87	2.16	3.56
20	9.8e-7	1.47	0.23	0.34	5.1e-6	7.75	1.19	1.80
21	2.4e-5	2.72	0.53	0.76	1.1e-4	11.98	2.33	3.33
22	3.1e-5	3.26	0.34	0.60	1.6e-4	16.96	1.77	3.11
Min	6.3e-8	1.41	0.20	0.31	3.1e-7	6.82	1.04	1.54
Max	3.5e-5	4.95	0.53	0.97	1.9e-4	26.37	2.53	5.14
Mean	1.4e-5	2.43	0.33	0.54	7.3e-5	12.47	1.70	2.76

Table D.3: Leave-one-out test on the specimens' set using the roots only (method KD).

Specimen	Hausdorff distances							
	Distances				Percentage of the tooth height			
	Min	Max	Mean	RMS	Min	Max	Mean	RMS
1	3.4e-5	1.60	0.33	0.44	1.7e-4	7.83	1.62	2.15
2	2.2e-5	1.87	0.42	0.55	1.1e-4	9.01	2.01	2.63
3	1.4e-5	1.56	0.43	0.55	7.0e-5	8.01	2.22	2.84
4	7.2e-6	1.20	0.25	0.34	3.7e-5	6.17	1.29	1.75
5	4.4e-6	2.56	0.49	0.65	2.2e-5	12.72	2.43	3.24
6	1.8e-5	1.51	0.50	0.61	9.7e-5	8.04	2.65	3.22
7	3.9e-5	2.02	0.42	0.57	2.0e-4	10.05	2.10	2.83
8	1.2e-5	1.05	0.28	0.36	5.8e-5	5.32	1.43	1.79
9	3.3e-5	1.96	0.36	0.53	1.9e-4	11.47	2.13	3.11
10	5.6e-5	1.34	0.26	0.34	3.2e-4	7.55	1.44	1.92
11	7.0e-6	2.64	0.54	0.78	4.1e-5	15.44	3.16	4.58
12	2.5e-5	1.65	0.26	0.38	1.2e-4	8.01	1.26	1.85
13	4.7e-5	2.52	0.53	0.74	2.3e-4	12.30	2.57	3.60
14	7.7e-5	2.12	0.69	0.87	3.9e-4	10.85	3.53	4.46
15	6.3e-6	1.92	0.34	0.49	3.4e-5	10.42	1.87	2.68
16	4.5e-5	1.97	0.50	0.63	2.1e-4	9.32	2.37	2.99
17	6.3e-5	0.94	0.17	0.22	3.1e-4	4.67	0.84	1.11
18	3.4e-5	1.84	0.28	0.37	1.9e-4	10.30	1.58	2.09
19	1.6e-5	1.14	0.27	0.34	8.3e-5	5.88	1.37	1.75
20	2.1e-5	0.90	0.20	0.26	1.1e-4	4.73	1.04	1.37
21	6.5e-6	2.16	0.24	0.44	2.9e-5	9.52	1.05	1.94
22	3.5e-5	1.84	0.45	0.60	1.8e-4	9.54	2.36	3.11
Min	4.4e-6	0.90	0.17	0.22	2.2e-5	4.67	0.84	1.11
Max	7.7e-5	2.64	0.69	0.87	3.9e-4	15.44	3.53	4.58
Mean	2.8e-5	1.74	0.37	0.50	1.5e-4	8.96	1.92	2.59

Table D.4: Leave-one-out test on the specimens' set using the roots only (method OD).

Specimen	Hausdorff distances							
	Distances				Percentage of the tooth height			
	Min	Max	Mean	RMS	Min	Max	Mean	RMS
1	4.8e-5	3.20	0.81	1.08	2.3e-4	15.61	3.95	5.28
2	3.0e-5	2.12	0.50	0.65	1.5e-4	10.19	2.39	3.12
3	3.1e-5	1.76	0.41	0.54	1.6e-4	9.03	2.08	2.77
4	4.6e-6	1.64	0.30	0.44	2.4e-5	8.45	1.54	2.27
5	5.7e-5	2.41	0.46	0.64	2.8e-4	11.99	2.30	3.19
6	6.6e-6	3.16	0.94	1.18	3.5e-5	16.83	5.01	6.27
7	7.7e-6	3.34	0.70	1.06	3.8e-5	16.58	3.50	5.28
8	1.6e-5	1.66	0.31	0.46	7.8e-5	8.38	1.57	2.31
9	2.5e-5	1.47	0.21	0.29	1.5e-4	8.59	1.24	1.69
10	1.8e-5	1.72	0.32	0.46	1.0e-4	9.66	1.77	2.61
11	9.5e-6	1.42	0.49	0.59	5.6e-5	8.30	2.85	3.47
12	4.3e-5	1.79	0.38	0.54	2.1e-4	8.68	1.85	2.60
13	4.9e-5	2.31	0.36	0.55	2.4e-4	11.26	1.75	2.69
14	9.0e-6	2.23	0.39	0.59	4.6e-5	11.43	1.98	3.01
15	7.4e-6	1.18	0.18	0.26	4.0e-5	6.39	1.00	1.39
16	5.9e-5	3.64	0.74	1.11	2.8e-4	17.20	3.51	5.25
17	3.2e-5	1.57	0.23	0.35	1.6e-4	7.83	1.13	1.77
18	2.2e-5	1.21	0.22	0.29	1.2e-4	6.74	1.22	1.64
19	4.2e-5	1.89	0.44	0.62	2.2e-4	9.74	2.27	3.19
20	9.1e-6	3.31	0.86	1.21	4.8e-5	17.41	4.51	6.39
21	9.9e-6	2.01	0.26	0.44	4.4e-5	8.84	1.14	1.93
22	2.7e-5	1.90	0.37	0.51	1.4e-4	9.90	1.93	2.64
Min	4.6e-6	1.18	0.18	0.26	2.4e-5	6.39	1.00	1.39
Max	5.9e-5	3.64	0.94	1.21	2.8e-4	17.41	5.01	6.39
Mean	2.6e-5	2.13	0.45	0.63	1.3e-4	10.87	2.30	3.22

References

- [1] S. Keiser-Nielsen, *Person identification by means of the teeth*, John Wright, Bristol, 1980.
- [2] S. M. Yamany, A. A. Farag, D. Tasman, and A. G. Farman, "A 3-D reconstruction system for the human jaw using a sequence of optical images," *IEEE Trans. Med. Imag.*, vol. 19, no. 5, pp. 538–547, May 2000.
- [3] Milan Sonka and J. Michael Fitzpatrick, Eds., *Medical Imaging, Volume 2: Medical Image Processing and Analysis*, SPIE Press, 2000.
- [4] R. Enciso, John P. Lewis, U. Neumann, and J. Mah, "3D tooth shape from radiographs using thin-plate splines," in *"MMVR11 - NextMed: Health Horizon", The 11th Annual Medicine Meets Virtual Reality Conference*, Jan. 2003, pp. 62–64.
- [5] R.J.A. Lapeer and R.W. Prager, "3D shape recovery of a newborn skull using thin-plate splines," *Computerized Medical Imaging and Graphics*, vol. 3, no. 24, pp. 193–204, May–June 2000.
- [6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal on Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [7] Laurent D. Cohen and Isaac Cohen, "Finite element methods for active contour models and balloons for 2D and 3D images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 11, pp. 1131–1147, Nov. 1993.

-
- [8] T. W. Sederberg and S. R. Parry, “Free-form deformation of solid geometric models,” in *ACM Siggraph Conference Proceedings*, 1986, vol. 20(4), pp. 151–160.
- [9] Markus Fleute, Stéphane Lavallée, and Rémi Julliard, “Incorporating a statistically based shape model into a system for computer-assisted anterior cruciate ligament surgery,” *Medical Image Analysis*, vol. 3, no. 3, pp. 209–222, 1999.
- [10] *The VTK User’s Guide, VTK 4.2*, Kitware, Inc., 2003.
- [11] S. Lavallée, R. Szeliski, and L. Brunie, “Anatomy-based registration of three-dimensional medical images, range images, x-ray, projections, and three-dimensional models using octree-splines,” in *Computer-Integrated Surgery: Technology and Clinical Applications*, R. H. Taylor et al., Eds. 1996, pp. 115–143, MIT Press, Cambridge, Massachusetts.
- [12] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Training models of shapes from sets of examples,” in *Proceedings of the British Machine Vision Conference*, 1992.
- [13] B. S. Everitt and G. Dunn, *Applied Multivariate Data Analysis*, Hodder Arnold, 1991.
- [14] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *The London, Edinburgh and Dublin philosophical magazine and journal of science*, vol. 2 (Series 6), pp. 559–572, 1901.
- [15] Ludovic Lebart, Alain Morineau, and Marie Piron, *Statistique Exploratoire Multidimensionnelle*, Dunod, third edition, 2000, French.
- [16] Y. Chen and G. Medioni, “Object modeling by registration of multiple range images,” *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 2724–2728, 1991.
- [17] Paul J. Besl and Neil D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

-
- [18] Dimitri P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, second edition, 1999.
- [19] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes in C++. The Art of Scientific Computing*, Cambridge University Press, Cambridge, second edition, 2002.
- [20] R. Bulirsch, “Bemerkungen zur Romberg-Integration,” *Numerische Mathematik*, vol. 6, pp. 6–16, 1964, German.
- [21] W. Lorensen and H. Cline, “Marching cubes: A high resolution 3-D surface construction algorithm,” *Computer Graphics*, vol. 21, pp. 163–169, 1987.
- [22] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, “MESH: Measuring errors between surfaces using the Hausdorff distance,” in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2002, vol. 1, pp. 705 – 708, <http://mesh.epfl.ch>.
- [23] Toshiaki Kondo, S. H. Ong, and Kelvin W. C. Foong, “Tooth segmentation of dental study models using range images,” *IEEE Trans. Med. Imag.*, vol. 23, no. 3, pp. 350–362, Mar. 2004.
- [24] M. D. Dikaiakos and J. Stadel, “A performance study of cosmological simulations on message-passing and shared-memory multiprocessors,” in *Proceedings of the International Conference on Supercomputing*, 1996, pp. 94–101.
- [25] L. Szirmay-Kalos and G. Marton, “Worst-case versus average case complexity of ray-shooting,” *Computing*, vol. 61, no. 2, pp. 103–131, 1998.
- [26] T. Jost and H. Hügli, “A multi-resolution scheme ICP algorithm for fast shape registration,” in *Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission*, June 2002, pp. 540–543.

- [27] D. Talbert and D. Fisher, “Exploiting sample-data distributions to reduce the cost of nearest-neighbor searches with kd-trees,” in *Proceedings of IDA 99 - 3rd International Symposium on Intelligent Data Analysis*, Aug. 1999, pp. 407–414.
- [28] Richard Szeliski and Stéphane Lavallée, “Matching 3D anatomical surfaces with non-rigid deformations using octree-splines,” *International Journal on Computer Vision*, vol. 18, no. 2, pp. 171–186, 1996.
- [29] Parthajit Bhattacharya, “Efficient neighbor finding algorithms in quadtree and octree,” M.S. thesis, Indian Institute of Technology, Kanpur, Nov. 2001.
- [30] Jon Louis Bentley, “Multidimensional binary search trees used for associative searching,” *Communications of the ACM*, vol. 18, pp. 509–517, 1975.
- [31] Andrew Moore, “Efficient memory-based learning for robot control,” Tech. Rep. 209, Computer Laboratory, University of Cambridge, Cambridge, UK, Oct. 1990.