Fast Free Vibration Modal Analysis of 2D Physics-Based Deformable Objects

Stelios Krinidis† and Ioannis Pitas†

Department of Informatics
Aristotle University of Thessaloniki
Box 451, 54124 Thessaloniki, Greece
e-mail: pitas@zeus.csd.auth.gr

This work has been supported by the IST Network of Excellence SIMILAR, funded by the Commission of the European Communities.

Abstract

This paper presents an accurate, very fast approach for the deformations of 2D physically based shape models representing open and closed curves. The introduced models are much faster than other deformable models (e.g. finite elements methods). The approach relies on the determination of explicit deformation governing equations, that involve neither eigenvalue decomposition, nor any other computationally intensive numerical operation. The approach was evaluated and compared with another fast and accurate physics-based deformable shape model, both in terms of deformation accuracy and computation time. The conclusion is that the introduced model is completely accurate and is deformed very fast on current personal computers (Pentium III), achieving more than 380 contour deformations per second.

Index Terms

Deformable model, modal analysis, real-time deformations, finite element method, eigenvalue decomposition, deformable curves, governing equation.

I. INTRODUCTION

A key problem in machine vision is how to describe features, contours, surfaces, and volumes, so that they can be segmented, recognized, matched, or any other similar underlying process. The primary difficulties can be summarized as follows:

- object descriptions are sensitive to noise
- objects can be nonrigid
- the shape of the 2D object projection varies with the viewing geometry.

These problems have motivated the use of deformable models [1]-[11] to interpolate, smooth, and warp raw data, since these models provide reliable shape reconstruction tools that are both robust and generic. Deformable models cover a very wide range of applications and have been used in pattern recognition [12], [13], computer animation [14], geometric modelling [15], surgery simulation [16], medical imaging [17], tracking [18], image segmentation [19], etc.

The class of deformable shape models originates with the method of active contours (“snakes”) introduced by Terzopoulos *et al.* [3], that are used to locate smooth curves in 2-D images. Since then, deformable models have been used for a number of applications in 2-D and 3-D by Terzopoulos, Witkin and Kass [20], where the deformable surface is constrained to encourage axial symmetry and is evolving under the forces determined from

a 2-D image or a pair of 2-D images. Deformable surfaces employ data information about forces from true 3-D data sets [21].

There have been two basic classes of deformable models: those based on parametric solid modelling primitives, such as those employed by Pentland *et al.* [5] and those based on mesh-like surface models, such as those in the work of Terzopoulos *et al.* [20]. In the case of parametric solid modelling, fitting has been performed using the "inside-outside" function of the modelling primitive [22], [23], [24], rendering this formulation to a very powerful, effective and useful method of representing shapes as canonical deformations of the prototype object [5], [22], [25]. This description is advantageous in the sense that it can produce a unique, compact description that is well suited for a number of computer vision applications (such as model-based segmentation, shape recognition, shape matching, and search). On the other hand, a drawback of this modelling method is that it may not have enough degrees of freedom to account for fine surface details. On the contrary, in the case of mesh-like models, which have been extensively used, fitting is performed using a physically motivated energy function [20], [26]. This formulation provides an accurate and very detailed shape description. Unfortunately, this description is neither unique, nor compact. Consequently, it cannot be used in some computer vision applications (such as shape recognition and search), without additional processing layers.

Some of the most significant mesh-like model subcategories, are the well known surface and contour models based on modal analysis [17]. They have been extensively used in the literature [17], [25], [27], [28] and will be the main focus of the remainder of this paper. Modal analysis is an alternative approach to the direct integration of the system of differential equations that result from any finite element method. It is based on *mode superposition*, i.e. expressing the time-dependent solution as the superposition of the natural modes of the system. The solution of an eigenvalue problem is necessary to find these modes. Thus, modal analysis involves the eigen-decomposition of the surface or contour deformation model of the object at hand that produces a robust, frequency-ordered shape description of the object in terms of the eigenvectors of the stiffness matrix of the model [17]. A more detailed presentation of modal analysis is presented in Section II.

All deformable models presented above are quite slow requiring dozens of iterations in the case of parametric formulation, and up to hundreds iterations in the case of physically-based mesh deformations that are based on modal analysis [25], [28]. The same stands for the well-known and popular in computer vision applications, level sets-based deformable models

[29]. Hence, all deformation-based shape modelling approaches are computationally intensive, since each iteration turns out to be very time-consuming. Thus, although deformable models are a powerful tool in computer vision applications, their scope is somewhat limited by their heavy computational burden, especially when real time applications are concerned. Although there are papers in the literature [25], [30] that claim real time deformations, e.g. the one proposed by Basdogan [30], they still have several disadvantages, since they require computationally intensive procedures, such as the eigenvalue decomposition, which is necessary for initialization. Additionally, all these methods share another disadvantage, the fact that their deformations do not have an analytic expression; they have no explicit formulation that can be used for their description and implementation. Thus, the implementation of these types of deformable models is quite difficult. Finally, the process of studying the model in order to infer its behavior, is quite painful and, in some cases, impossible.

We address all these problems by introducing a 2D physics-based deformable approach representing open and closed curves, based on modal analysis, using explicit functions. This leads to a very fast model deformation approach, involving only the calculation of an explicit function for the description of the deformations of the object under examination, as opposed to the eigenvalue decomposition that is necessary in classical shape modal analysis.

Our approach was motivated by the technique presented in [17], [27], which consists of analyzing non-rigid motion, with application to medical images. Nastar *et al.* [17] approximated the dynamic deformations of the object contour, using a physically based deformable curve/surface. In order to reduce the number of parameters describing the deformation, modal analysis was exploited, thus providing a spatial smoothing of the curve/surface. On the same basis, we employ modal analysis and, after simplifying the deformation governing equations, we prove that the deformations of open and closed 2D curves have explicit governing equations, involving neither eigenvalue decomposition, nor any other computationally intensive operation (EFMA, Section III). Moreover, a new methodology (TEFMA, Section III), which further reduces the computational complexity of the introduced algorithm (EFMA) is presented. This methodology was motivated by the propagation and the interaction of the deformations of the model. It is capable of achieving more than 380 contour deformations per second on a typical image of size 512×512.

The remainder of the paper is organized as follows. The 2D physics-based deformable shape models [17] are presented in Section II. In Section III the proof and the properties of the explicit formulation of the deformable physics-based governing equations are introduced.

A comparison between the initial deformation models [17] and the novel ones, are presented in Section IV. A simple application demonstrating the effectiveness and the usage of the new approach is also illustrated in the same Section and final conclusions are drawn in Section V.

II. 2D Physics-Based Deformable Shape Modelling Based On Modal Analysis

In this Section, physically based deformable shape models [17], [27] exploiting modal analysis are introduced. We consider both the surface and volume properties of the objects at hand. We restrict ourselves to elastic deformations, i.e. we assume that the object recovers its original configuration as soon as all applied forces causing the deformation are removed.

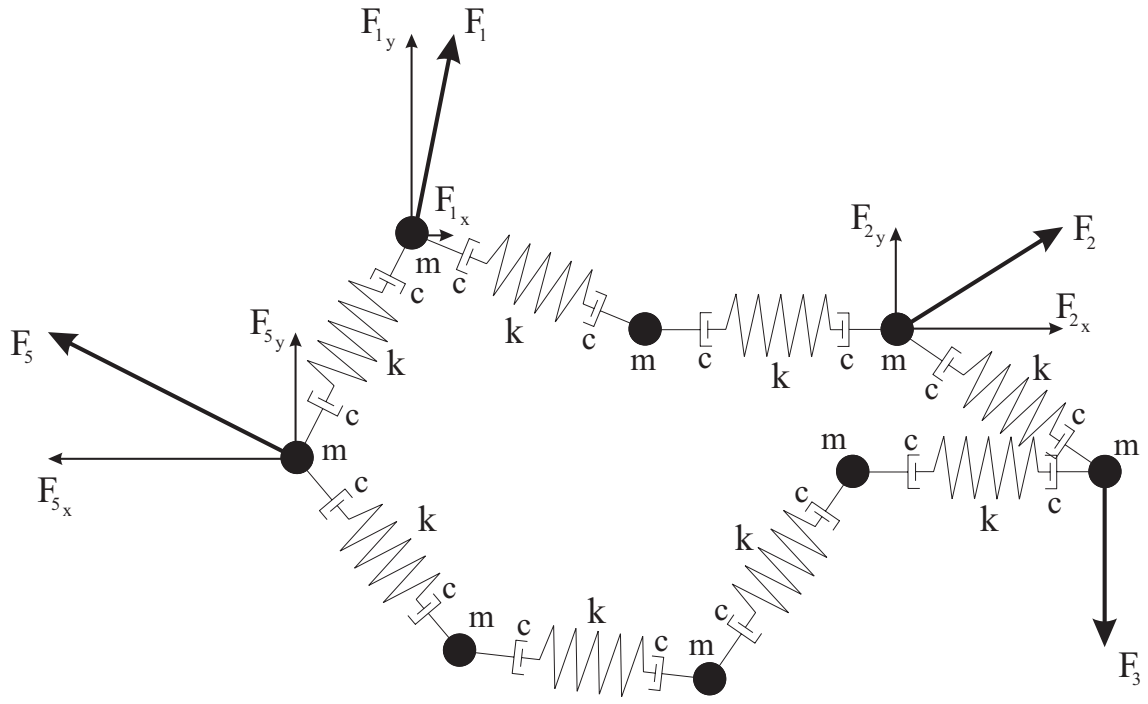

Fig. 1. *2D closed model example of 8 nodes of mass m connected with identical springs of stiffness k. Four forces are acting on the model, that produce model deformation.*

Modelling an elastic 2D boundary can be achieved by an open or closed chain topology of N virtual masses on the contour, shown in Figure 1. Each model node has a mass m and is connected to its two neighbors with identical springs of stiffness k. The ratio $a = \frac{k}{m}$ constitutes the so-called *characteristic value* of the model, which is a constant value that describes its physical characteristics and determines its physical behavior. When a increases,

the object tends to behave as a rigid one, which means in practice, that the model can be spatially moved without any deformation. On the other hand, when a decreases, model tends to be treated as a fully deformable one, which means that each force affects only the node (mass) it is applied to. The node coordinates of the model under examination are stacked in vector:

$$\mathbf{X}_t = \left[x_1^t, y_1^t, \ldots, x_N^t, y_N^t\right]^T \tag{1}$$

where N is the number of vertices (masses) of the chain and t denotes the t-th deformation time instance. The model under study, is a physics-based system governed by the fundamental equation of dynamics:

$$\boldsymbol{f}_e(x_i^t) + \boldsymbol{f}_d(x_i^t) + \boldsymbol{f}_{ext}(x_i^t) = m_i \ddot{x}_i^t \tag{2}$$

where m_i is the mass of the point under study and $\ddot{x}_i^t$ its acceleration under total load of forces. $\boldsymbol{f}_d(\cdot)$ is a damping force, $\boldsymbol{f}_{ext}(\cdot)$ the external load on node under study, and $\boldsymbol{f}_e(\cdot)$ is the elastic force due to node neighbors. The above governing equation is expressed for all model nodes, leading to a nonlinear system of coupled differential equations, since the displacement of a node depends on the displacement of its neighbors.

In order to solve this system of equations, we propose to set the natural length l_0 of the springs equal to zero. This assumption does not import any restriction to the initial configuration of the model, if we add an equilibrium force $\boldsymbol{f}_{eq}(x_i^t) = -\boldsymbol{f}_e(x_i^t)$. This force keeps the model inflated, so that it does not shrink to a point. We assume that, at any future time, this equilibrium force is constant. Hence, the natural state of the model is its initial configuration. This assumption has a main advantage, that the model can be considered within the framework of linear elasticity. As a consequence, our solution lies in a set of linear differential equations with node displacements decoupled in each coordinate, regardless of the magnitude of the displacements. To enforce the assumption of constant equilibrium force $\boldsymbol{f}_{eq}(\cdot)$, the angular variations of the springs orientation should be sufficient small ($< 15^o$), in which case the aforementioned approximation is valid [5], [31].

The governing equation can now be written in a matrix form [32]:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{F}_t \tag{3}$$

where $\mathbf{U} = \boldsymbol{X}_\mathbf{t} - \boldsymbol{X}_{\mathbf{t_0}}$ is the nodal displacements vector. $\mathbf{M}, \mathbf{C}$, and $\mathbf{K}$ [33] are the mass, damping, and stiffness matrices of the model, respectively, and $\mathbf{F_t}$ is the external force vector, usually resulting from the attraction of the model by the object contour (sometimes based

on the Euclidean distance between the object contour and the node coordinates [34], [35]). Note, that the above formulation provides a simplification between forces $\boldsymbol{f}_{eq}(\cdot)$ and $\boldsymbol{f}_{e}(\cdot)$, so that do not appear in the governing equation. These forces can be viewed as *internal* forces which do not need to be computed. Furthermore, equation (3) is a finite element formulation of the deformation process.

Instead of solving directly the equilibrium equation (3), one can transform it by a change of basis:

$$\mathbf{U} = \mathbf{\Psi}\tilde{\mathbf{U}} \tag{4}$$

where $\mathbf{\Psi}$ is the square nonsingular transformation matrix of order N to be determined, and $\tilde{\mathbf{U}}$ is referred to as the *generalized displacement* vector. One effective way of choosing $\mathbf{\Psi}$ is setting it equal to $\mathbf{\Phi}$, a matrix whose entries are the eigenvectors of the generalized eigenproblem:

$$\mathbf{K}\boldsymbol{\phi_i} = \omega_i^2\mathbf{M}\boldsymbol{\phi_i} \tag{5}$$

$$\mathbf{U} = \mathbf{\Phi}\tilde{\mathbf{U}} = \sum_{i=1}^{N} \tilde{u}_i\boldsymbol{\phi_i} \tag{6}$$

Equation (6) is referred to as the modal superposition equation. The i-th eigenvector, i.e. the i-th column of $\mathbf{\Phi}$, denoted by ϕ_i, is also called the i-th *vibration mode*, $\tilde{u}_i$ (the i-th scalar component of $\tilde{\mathbf{U}}$) is its amplitude, and ω_i is the corresponding eigenvalue (also called *frequency*). If the matrix $\tilde{\boldsymbol{C}} = \mathbf{\Phi}^T\boldsymbol{C}\mathbf{\Phi}$ is diagonal (standard Rayleigh hypothesis [17]), then in the modal space the governing matrix-form equations decoupled into N scalar equations:

$$\ddot{\tilde{u}}_i^t + \tilde{c}_i\dot{\tilde{u}}_i^t + \omega_i^2\tilde{u}_i^t = \tilde{f}_i^t, \qquad\qquad i = 1, \ldots, N. \tag{7}$$

Solving these equations at time t leads to $(\tilde{u}_i^t)_{i=1,\ldots,N}$, and the displacement $\boldsymbol{U}$ of the model nodes is obtained by the modal superposition equation (6).

In practice, we wish to approximate nodal displacements $\mathbf{U}$ by $\hat{\mathbf{U}}$, which is the truncated sum of the N' low-frequency vibration modes, where $N' \ll N$:

$$\mathbf{U} \approx \hat{\mathbf{U}} = \sum_{i=1}^{N'} \tilde{u}_i\boldsymbol{\phi_i} \tag{8}$$

Eigenvectors $(\phi_i)_{i=1,\ldots,N'}$ form the *reduced modal basis* of the system. This is the major advantage of modal analysis: it is solved in a subspace corresponding to the N' truncated low-frequency vibration modes of the deformable structure [5], [17], [27]. The number of vibration modes retained in the object description, is chosen so as to obtain a compact but

adequately accurate representation. A typical *a priori* value for N', covering many types of standard deformations is equal to one quarter of the number of the vibration modes.

An important advantage of the formulations described so far, in the full as well as the truncated modal space, is that the vibration modes ϕ_i and the frequencies ω_i of an open or closed chain topology have an explicit expression [17] and they do not have to be computed using eigen-decomposition techniques. The frequencies and the vibration modes of the closed chain are given by:

$$\omega_i^2 = 4a \sin^2\left(\frac{\pi i}{N}\right) \tag{9}$$

$$\boldsymbol{\phi}_i = \left[\ldots, \cos\frac{2\pi ij}{N}, \ldots\right]^T \tag{10}$$

where $i \in \{1, 2, \ldots, N\}$ and $j \in \mathcal{B}(N)$. $\mathcal{B}(N)$ is the first Brillouin zone [17] and is equal to $\left\{-\frac{N}{2}+1, \ldots, \frac{N}{2}\right\}$ for N even, and $\left\{-\frac{N-1}{2}, \ldots, \frac{N-1}{2}\right\}$ for N odd. Furthermore, the case of an open chain topology is very similar, where the frequencies and the corresponding vibration modes are obtained by:

$$\omega_i^2 = 4a \sin^2\left(\frac{\pi i}{2N}\right) \tag{11}$$

$$\phi_i = \left[\ldots, \cos\frac{2\pi(2i-1)j}{2N}, \ldots\right]^T \tag{12}$$

where $i \in \{1, 2, \ldots, N\}$ and $j \in [0, \ldots, N-1]$.

Thus, the deformations of the described *Deformable Model based on Modal Analysis* (abbreviated here as DMMA), for a closed chain as well as an open one, can be given by:

$$u_i^t = \sum_j^N \frac{\sum_{n=1}^N [f_n \phi_n(j)]}{(1+\omega_j^2)\sum_{n=1}^N \phi_n^2(j)} \phi_i(j) \tag{13}$$

whereas the deformations of the *Truncated Deformable Model based on Modal Analysis* (abbreviated here as TDMMA), for both open and closed chains, can be described by:

$$\hat{u}_i^t = \sum_j^{N'} \frac{\sum_{n=1}^N [f_n \phi_n(j)]}{(1+\omega_j^2)\sum_{n=1}^N \phi_n^2(j)} \phi_i(j) \tag{14}$$

The above equations are a brief description of the aforementioned methodology of deformable models that are based on modal analysis. As you can see, displacements are directly related to the eigenvalues ω_j^2 and eigenvectors $\phi_n(j)$ of the model.

Figure 2 illustrates in a clear and comprehensive way the vibration modes based contour modelling of a 2D slice of a 3D CT scanned tooth germ volume. This example is primarily used to illustrate graphically the operation of the above mentioned deformable models.

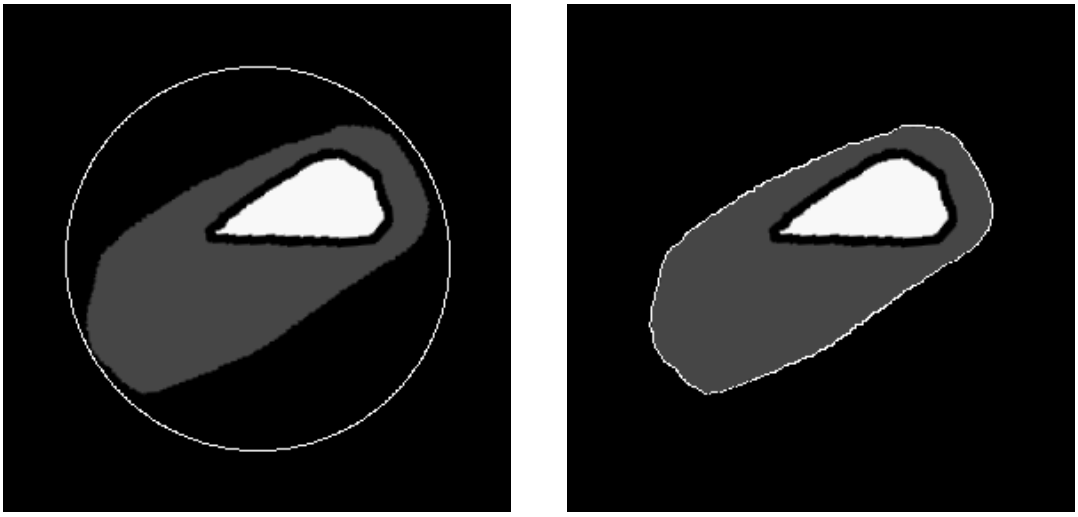

Fig. 2. *Contour modelling of a 2D slice of a 3D tooth germ volume.* **(a)** *The initial 2D closed deformable model initialized around the tooth germ.* **(b)** *The deformable model at equilibrium.*

The 25% lowest frequency modes were retained for this representation, as this truncated description provides a satisfactory compromise between accuracy and complexity of the representation. The circular chain is (Fig. 2a) and the vibration amplitudes are explicitly computed by equation (14). Rigid body modes ($\omega_i = 0$) are discarded and the nodal displacement can be recovered using equation (8). We can calculate the model driving forces firstly by applying a contour detection algorithm on the image and by calculating subsequently the distance field.

The deformable model is manually initialized around the object under consideration. The contour model nodes are separated into two categories, the landmark and the non-landmark nodes. Landmark nodes are those with artificial or anatomical importance, such as high curvature points, whereas non-landmarks are the rest. A corner detection algorithm [36] is exploited to specify the external force for landmarks nodes at each frame or better at time instant t:

$$f_{ext}(x_i^t) = \frac{1}{2} w \|dist\|^2 \tag{15}$$

where $dist$ is the distance between the landmark and its new position as it is estimated by the corner detection algorithm, and w is a constant. The rest external force vector, i.e. for non-landmarks nodes, is completed by:

$$f_{ext}(x_i^t) = \frac{1}{2} w_{nl} \|dist_{nl}\|^2 \tag{16}$$

where $dist_{nl}$ is the distance between the non-landmark node and its closest boundary point, and $w_{nl} \ll w$ is a constant. This force vector is used in order to pull the model to track the object of interest. The method is an iterative one, thus its final position is achieved after a substantial number of iterations (number of frames). The corner detection algorithm, at each iteration, determines the driving forces which deform the model, using one of the above mentioned methods. Thus, the model is deformed after each iteration (frame). We assume that the deformation at each iteration is performed in one step.

III. A Closed-Form Representation of the 2D Physics-Based Deformable Models

It is obvious that the deformations described in the previous Section are still computationally intensive, since they require the calculation of a large number of summations in (13) and (14). In this Section, our goal is to simplify the deformation process (Section II). This is achieved by using the properties of trigonometric functions series, as can be seen in Appendix. Thus, we can introduce a new way of calculating the deformations, achieving very fast deformation computation, without loss of accuracy.

As already mentioned, the deformations $\boldsymbol{U}$ of the DMMA applied to a closed chain are given by equation (13). (13) is applied to the x and y coordinates independently. f_i denotes the x and y components of the force acting on node i:

$$\boldsymbol{F} = (f_{1_x}, f_{1_y}, f_{2_x}, f_{2_y}, \ldots, f_{N_x}, f_{N_y})^T \tag{17}$$

and finally u_i denotes the x and y components of the displacement of the node i:

$$\boldsymbol{U} = (u_{1_x}, u_{1_y}, u_{2_x}, u_{2_y}, \ldots, u_{N_x}, u_{N_y})^T \tag{18}$$

As we have proved in the Appendix, the deformations of the DMMA are equal, for closed curves, to:

$$u_i = \frac{1}{\sqrt{1+4a}} \sum_{j=1}^{N} f_j a^d \frac{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N-2d} + \left(\frac{\sqrt{1+4a}-1}{2}\right)^{N-2d}}{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N} - \left(\frac{\sqrt{1+4a}-1}{2}\right)^{N}} \tag{19}$$

We call formulation (19) *Explicit Formulation of Modal Analysis* (EFMA) model. The deformation governing equation for open chain physics-based deformable models can be proved, in a similar way as in the Appendix, to be:

$$u_i = \frac{1}{\sqrt{1+4a}} \sum_{j=1}^{N} f_j \frac{S_{i,j}}{\mu^{2N} - \lambda^{2N}} \tag{20}$$

$$S_{i,j} = a^{|i-j|}\left[\mu^{2N-2|i-j|} + \lambda^{2N-2|i-j|}\right] + a^{i+j-1}\left[\mu^{2N-2(i+j-1)} + \lambda^{2N-2(i+j-1)}\right]$$

where μ and λ are two constant values equal to $\frac{\sqrt{1+4a}+1}{2}$ and $\frac{\sqrt{1+4a}-1}{2}$ respectively, a is the model's characteristic value, N is the total number of model nodes, and d is the distance (in number of nodes) between the node under examination and the node, where force f_i is applied. Hence, in practice, d is equal to $min(|i-j|, |N-|i-j||)$ for the closed case models.

Due to the fact that the model exhibits linear elasticity and that neighboring nodes cause the forces $\boldsymbol{f}_e(\cdot)$, we are able to approximate nodal displacements $\mathbf{U}$ (19) without any substantial loss of accuracy, by the truncated contribution of the N'' adjacent nodes, where $N'' \ll N$:

$$u_i \approx \frac{1}{\sqrt{1+4a}} \sum_{j=i-N''}^{i+N''} f_j a^d \frac{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N-2d} + \left(\frac{\sqrt{1+4a}-1}{2}\right)^{N-2d}}{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N} - \left(\frac{\sqrt{1+4a}-1}{2}\right)^{N}} \tag{22}$$

The nodes i-N'' to i+N'', for each node i, form the *reduced nodal basis* of the system. This is the major advantage of the new form of the deformation governing equation: it is solved in a *nodal subspace* corresponding to the N'' adjacent model nodes of the deformable structure [5], [17], [27]. From now on, the deformation calculation in a reduced nodal space (22) will be called *Truncated Explicit Formulation of Modal Analysis* (TEFMA). The number of nodes retained in (22) is chosen so as to produce a compact but adequately accurate shape representation. After performing a sequence of experiments, it has been observed that a typical *a priori* value for N'', covering many types of standard deformations, is equal to $\frac{\sqrt{1+4a}+1}{2} lnN$. Thus, only a very small number of adjacent nodes is taken into account in the calculations. For instance, if we consider a 2D closed deformable model of 1,000 nodes having a equal to 1, equation (19) requires $2 \times 1,000 \times 1,000 = 2,000,000$ elementary calculations, each corresponding to the calculation of one summation term. It is clear that equation (22) in the subspace $N'' = \frac{\sqrt{5}+1}{2} lnN$ of the adjacent (neighboring) nodes would noticeably reduce the computation time, as the number of the elementary calculations would be reduced to $2 \times 1,000 \times \left\lceil \frac{\sqrt{5}+1}{2} ln(1,000) \right\rceil = 24,000$.

The characteristic value a influences object rigidity as follows. When a takes very large values and reaches infinity, which, in practice, means that the deformable object has springs characterized by very large stiffness with respect to the node mass, the limit of the governing equation (19) is:

$$\lim_{a\to\infty}\left[\frac{a^d}{\sqrt{1+4a}}\frac{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N-2d}+\left(\frac{\sqrt{1+4a}-1}{2}\right)^{N-2d}}{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N}-\left(\frac{\sqrt{1+4a}-1}{2}\right)^{N}}\right]=\frac{1}{N} \tag{23}$$

Thus, the deformations described by equation (19), when a takes very large values, are given by $u_i=\left[u_{i_x},u_{i_y}\right]=\frac{1}{N}\left[\sum_{j=1}^{N}f_{j_x},\sum_{j=1}^{N}f_{j_y}\right]$. This means that all model nodes exhibit the same displacements. This is a rigid object motion with displacements equal to the normalized summation of the respective forces. On the other hand, when the characteristic value a takes very small values (close to zero), which, in practice, means that the deformable object has very flexible springs (springs with very low stiffness) with respect to the node mass, the limit of the governing equation (19) is given by:

$$u_{n+d}=\lim_{a\to 0}\left[\sum_{j=1}^{N}f_j\frac{a^d}{\sqrt{1+4a}}\frac{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N-2d}+\left(\frac{\sqrt{1+4a}-1}{2}\right)^{N-2d}}{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N}-\left(\frac{\sqrt{1+4a}-1}{2}\right)^{N}}\right]=\begin{cases} f_n, & d=0\\ 0, & d\neq 0\end{cases} \tag{24}$$

where d is the distance (in number of nodes) between the node n under examination and the node where the force is applied. This means that each force applied to a model node affects only this node, and its displacement is proportional to the force value. Thus, the object under consideration is a fully deformable object without correlation between node displacements. When $a=1$, i.e. when the object node mass and the spring stiffness are equal, $k=m$, equation (19) is transformed to:

$$U_i=\frac{1}{\sqrt{5}}\sum_{j=1}^{N}f_j\frac{\left(\frac{\sqrt{5}+1}{2}\right)^{N-2d}+\left(\frac{\sqrt{5}-1}{2}\right)^{N-2d}}{\left(\frac{\sqrt{5}+1}{2}\right)^{N}-\left(\frac{\sqrt{5}-1}{2}\right)^{N}} \tag{25}$$

In that case, it is observed that the displacement propagation to the adjacent nodes is relatively small and therefore, the nodal subspace is low. One can notice that equation (25) is similar to the Fibonacci sequence, as the numerator and the denominator of the fraction are Fibonacci sequences. In this case, the nodal subspace is small. Another case is when $a>1$, i.e. the deformable object has spring stiffness greater than the node mass. Hence, a force applied to a node, is expected to propagate to its adjacent nodes, depending on the value of a. Greater values of a correspond to greater levels of deformation propagation. Thus, each force affects a substantial number of the object nodes. If a is less than 1, which means that the spring stiffness of the object under consideration is less than the corresponding node mass, we have

decreasing levels of deformation propagation to its adjacent nodes, restricting its influence to a small local node area, depending on the corresponding value of a. Smaller values of a correspond to decreasing levels of deformation propagation. That means that each force affects increasingly fewer object nodes.

IV. Performance Analysis And Simulations

In this Section, a comparison between the DMMA and TDMMA described in Section II and the EFMA and TEFMA introduced in Section III is presented. The comparison is performed in terms of the displacement estimation error and the required computation time. The relative displacement error is defined as:

$$\left\{ \begin{array}{c} E_{TDMMA} \\ E_{TEFMA} \end{array} \right\} = \frac{\frac{1}{N}\sum_i \|U_i' - U_i\|}{\max_i \|U_i' - U_i\|} \tag{26}$$

where U_i and U_i' are the displacements calculated using DMMA (13) and the deformable method under comparison (TDMMA and TEFMA) respectively, i.e., DMMA is the reference deformable method. The displacement error E_{EFMA} (26) of the EFMA and the DMMA is zero, as proven in the Appendix. All experiments are performed on a Pentium III (700 MHz) workstation under Windows 2000 Professional without any particular code optimization.

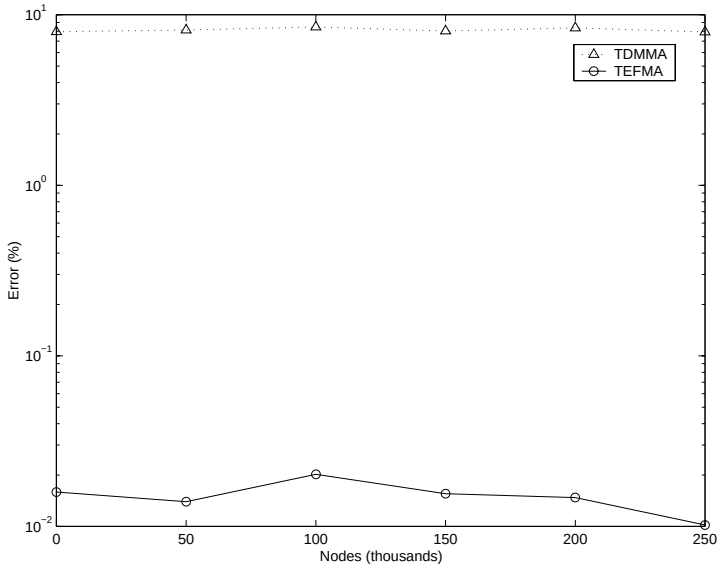

Fig. 3. ***Error comparison*** *of TDMMA and TEFMA methods.*

The displacement errors E_{TDMMA} and E_{TEFMA} (26) comparison versus node number N are illustrated in Figure 3. All errors in the Figure, are in a percentage form, due to the

fact that the absolute error is not that representative, since it depends on the input forces applied to the model. Note that error axis Y is a logarithmic one. In the case of TDMMA, where only 25% of the eigenvalues are taken into account, the relative average error is only 8.04721%, which is not so important. Thus, as Nastar *et al.* [17] claim, one can use the TDMMA without any particular loss of accuracy. In the case of TEFMA only $\frac{\sqrt{1+4a}+1}{2} lnN$ adjacent (neighboring) nodes are considered. The relative average error is only 0.01553%, which is negligible with respect to the global deformation of the model, and, consequently, deformations can be computed using TEFMA without any particular loss of accuracy. Besides, it is clearly depicted in Figure 3 that deformations extracted from the TEFMA have, in average, a relative error 500 times less than the corresponding deformation computed by TDMMA. It must be noted that the errors do not change with the number of nodes N.

Furthermore, Figures 4a and 4b show the errors (E_{TDMMA} and E_{TEFMA}) of the deformations computed using TDMMA and TEFMA versus the used modal and nodal subspace. The experiment has been performed for various values of a varying from 0.01 to 100. In both figures, the Y axis denotes the error E_{TDMMA} and E_{TEFMA} (26) of each method (TDMMA and TEFMA) under comparison. In Figure 4a, X axis depicts the percentage of modal vibration modes exploited by TDMMA, while in Figure 4b, X axis denotes the percentage $\left(\frac{N''}{N}100\%\right)$ of adjacent nodes utilized by TEFMA. Those two figures illustrate the relation between deformation error and the modal or nodal subspace used for their calculations. First of all, let us see the relation between the error, modal or nodal subspace, and the characteristic value a of the model. As a increases fewer modal vibration modes are needed when performing TDMMA deformations to achieve the same error (Figure 4a). The exactly opposite statement stands for the TEFMA deformations, i.e. as a increases, an increasing number of adjacent nodes are exploited in their calculations (Figure 4b). Furthermore, one can notice in Figure 4a that E_{TDMMA}, in most cases, falls under an acceptable relative error threshold (i.e. 10%), using more than 40% of the modal vibration modes, while TEFMA needs only a small nodal neighborhood (9%) to achieve the same error. Thus, TEFMA formulation is preferred, since it can achieve an acceptable error using a very small nodal subspace for its calculations. This claim is enforced by the error comparison shown in Figures 4c and 4d, where Y axis denotes the deformation execution time. Thus, TEFMA is proved to be faster than TDMMA under every circumstances. For instance, for a relative error threshold of 10%, TDMMA requires 6.1998%, 15.2746% and 80.5342% of the vibration modes for a model with a physical characteristic value of 100, 1 and 0.001 respectively. The corresponding

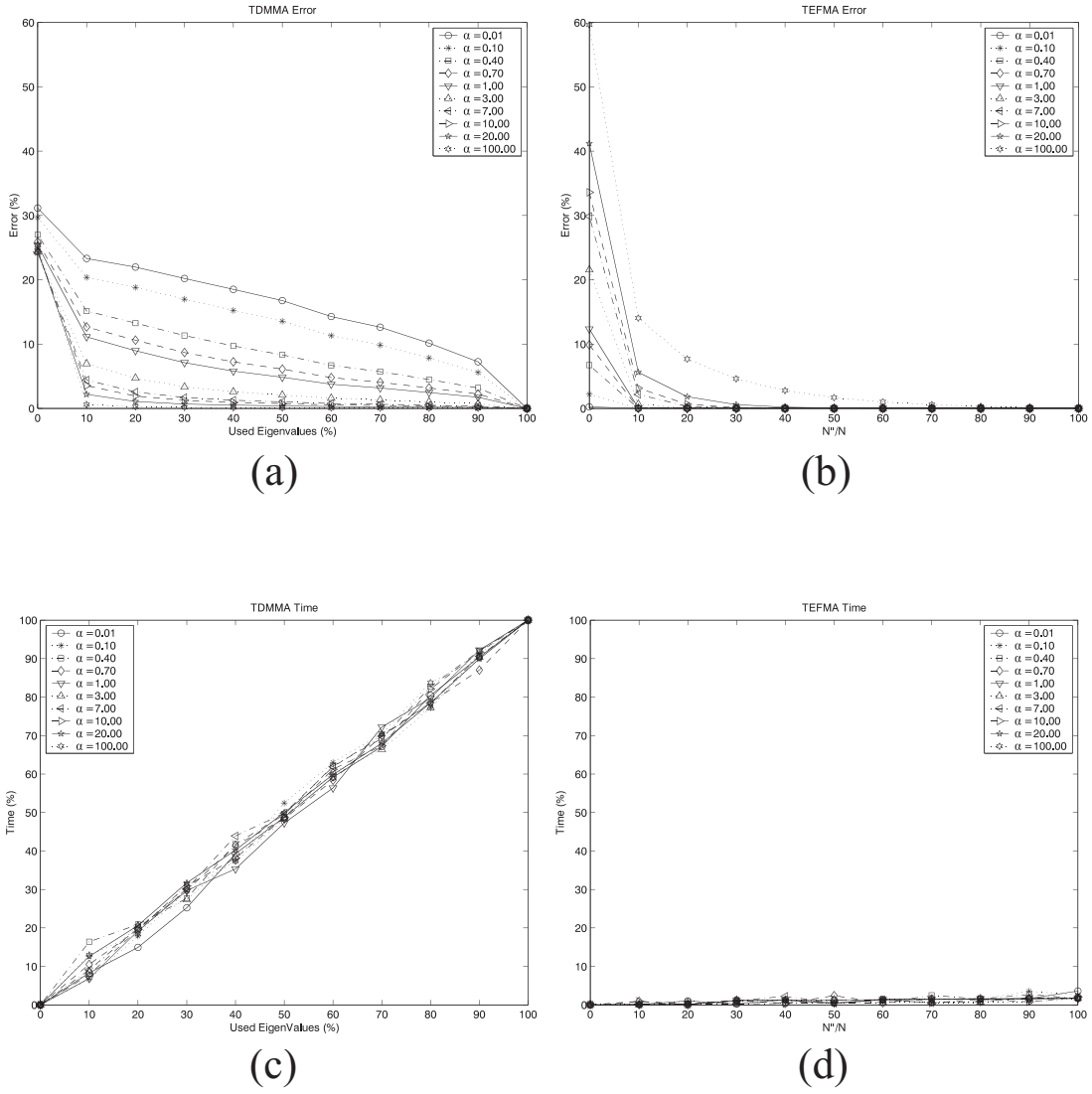

Fig. 4. ***Error and Time comparison*** *versus nodal and nodal subspace used for the deformations. Different plots corresponds to different values of the characteristic value a of the model.*

relative execution time of these deformations is 4.7270%, 16.0057% and 81.1725%. On the other hand, the deformations of the aforementioned models (having the same characteristic value a), using TEFMA, require only a very small set of adjacent nodes (16.3676%, 1.9106% and 0.1025% respectively), while the execution time is proved to be very fast, i.e. 0.6461%, 0.0354% and 0.0523% respectively. One can notice that these two deformation methods can achieve the same accuracy, but TEFMA is much faster under all circumstances.

Additionally, we have performed a computational complexity analysis of the methods

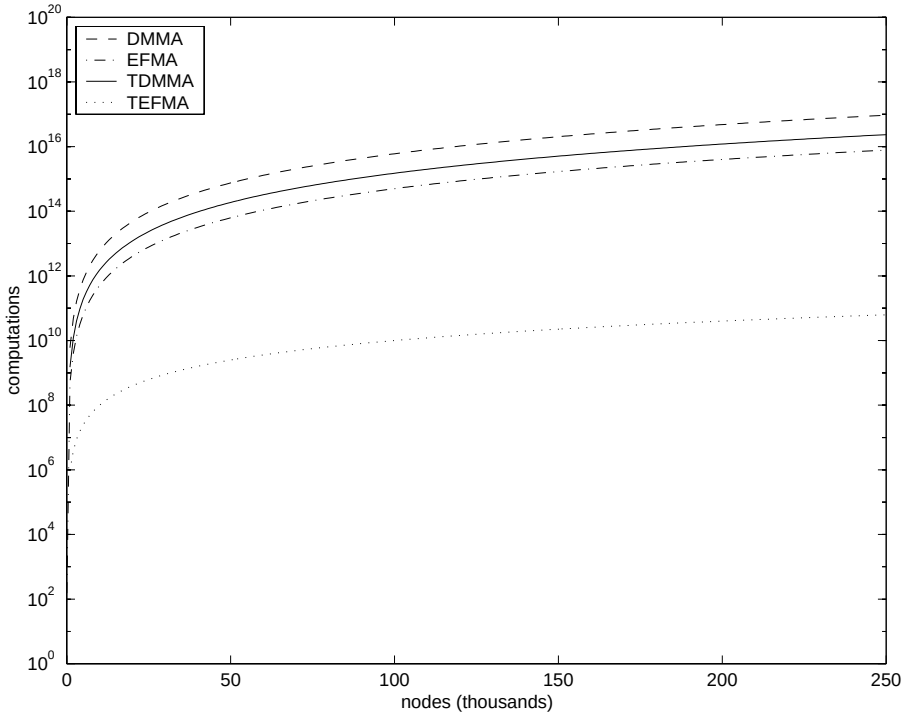

Fig. 5. ***Complexity analysis.*** *Computations of the DMMA, TDMMA (25% of eigenvalues), EFMA and TEFMA versus the number of model nodes.*

under comparison. We assume that the four elementary mathematical operations – additions, abstractions, multiplications and divisions – have the same computation time. Furthermore, sine, cosine and square root calculations are assumed to be computed by a Taylor series expansion [37]. Then it can be proven that DMMA (13) requires approximately $6N^3+10N^2+\frac{189}{2}N$ elementary computations, where N is the number of model nodes. EFMA (19) requires only $\frac{1}{2}N^3+N^2+85$ computations, while TDMMA and TEFMA require $\frac{3}{2}N^3+\frac{11}{4}N^2+\frac{477}{8}N$ and $N^2-NlnN(1+lnN)+85$ computations respectively. These curves are plotted in Figure 5. It is worth noting that TEFMA has computational complexity of order $O(N^2)$, whereas even TDMMA has computational complexity $O(N^3)$.

Furthermore, we have performed computation time benchmarking experimentally. Figures 4c, 4d and 6 show the computation time for all methods under comparison. It must be mentioned that, in Figure 6, the computation times are plotted in a logarithmic axis for better visualization. EFMA and TDMMA require approximately the same computation time. Both are 3 times faster than DMMA. The computation times of these two cases are quite satisfactory, but they are still far slower than TEFMA. TEFMA computes the deformations very fast. The computation time of TEFMA is more than 1 second only when the model

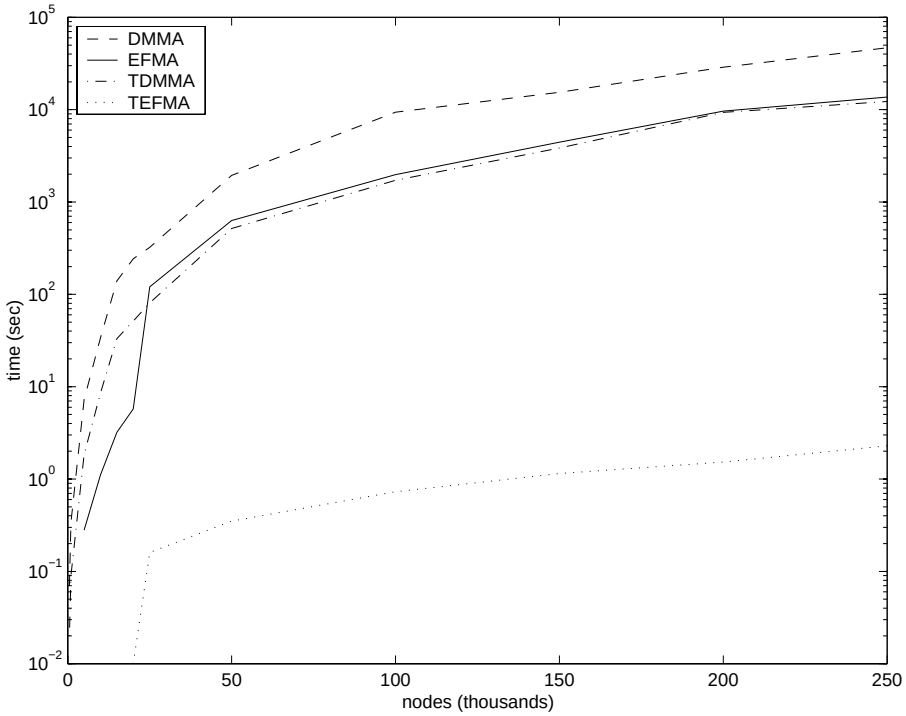

Fig. 6. ***Time comparison*** *of DMMA, TDMMA, EFMA and TEFMA methods versus the number of model nodes.*

has more than 150.000 nodes. Therefore, TEFMA can be deformed 100 times per second for contours having 20.000 nodes and $6,25$ times per second for contours of 25.000 nodes. Hence, it can be safely used in many real time applications. TEFMA can achieve a speedup of 4-5 orders of magnitude versus DMMA for almost any contour length. Furthermore, computation time comparison for TDMMA and TEFMA can also be seen in Figure 4. In all our experiments until now motivation (external) forces are assumed to be random, i.e. a random force vector has been chosen to deform our models.

The comparison between the initial (DMMA and TDMMA) and the introduced (EFMA and TEFMA) models is extended to a real contour tracking problem and is focused mainly on increasing computation speed. Contour tracking is a characteristic computer vision problem, where lip tracking is a special case. This is the reason that it has been selected to demonstrate the effectiveness of the proposed methodology. The main goal of this demonstration is to contrast the proposed and the original deformable models, with respect to speed and accuracy.

The model used in our experiments consists of 1472 nodes and has been applied to images illustrating human lips from a neutral position to a fully expressive one, e.g. smile (Figure 7a). The video stream consists of 17 frames, some of which are shown in Figure 7a. Thus,

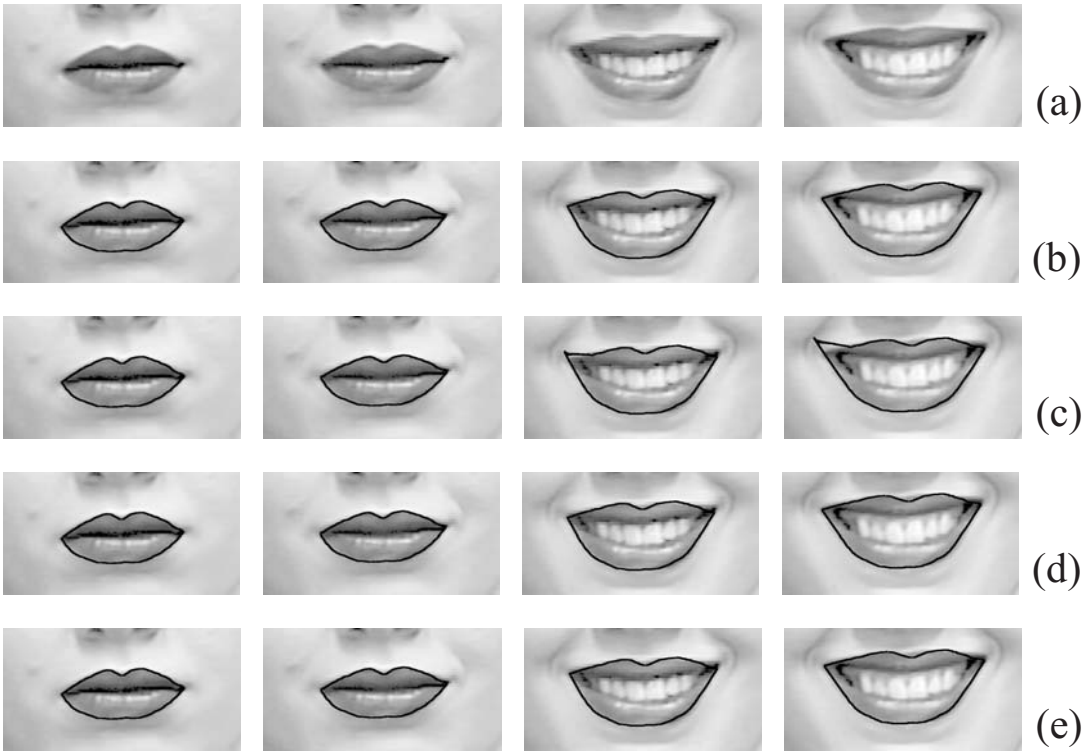

Fig. 7. ***Lip tracking*** *in video stream (a), using (b) DMMA, (c) TDMMA (25% of the eigenvalues are kept), (d) EFMA, and (e) TEFMA.*

TABLE I

Time comparison. *The EFMA and the TEFMA compared with the DMMA and the TDMMA respectively. All times are measured in seconds and the number of nodes of the snake (deformable model) under consideration is 1472.*

	DMMA	TDMMA	EFMA	TEFMA	Rest
time (sec)	10,523	2,793	0,476	0,045	1,510
total time (sec)	12,033	4,303	1,987	1,555	
average time (sec)	0,6190	0,1643	0,0280	0,0026	0,0888
total average time (sec)	0,7078	0,2531	0,1169	0,0915	
$100\frac{time}{time_{DMMA}}$	100,000%	26,541%	4,525%	0,426%	14.353%
$100\frac{time}{time_{Rest}}$	696,700%	184,910%	31,529%	2,968%	100,000%
deformations per seconds	1,616	6,086	35,714	384,615	
algorithm iterations per seconds	1,413	3,951	8,554	10,929	

the iterations of the algorithm in our experiments are limited to 17, and as a consequence the deformations as well. The computational time of the model deformations are illustrated in Table I. It is obvious that EFMA and specially TEFMA are much faster than either DMMA

or TDMMA. Moreover, we can see in Table I, that DMMA, TDMMA, EFMA and TEFMA require in average 0.6191, 0.1643, 0.0280 and 0.0026 seconds per iteration respectively, while the rest of our tracking algorithm needs 0.0888 seconds. Thus, TEFMA can achieve a rate of over 380 model deformations per second, and the whole algorithm can perform about 11 iterations per second, a fact that establishes it as a potential real time application, something that is almost prohibiting when implementing deformable based applications. In all of the above mentioned experiments, we assumed that the physics-based deformable model characteristic value a is equal to 1.

The results of each deformable model case are depicted in Figure 7. Figure 7 illustrates the model adaptation on human lips over a video stream from a neutral position to a fully smiled one (Figure 7a), exploiting DMMA (Figure 7b), TDMMA (Figure 7c), EFMA (Figure 7d) and TEFMA (Figure 7e). The above mentioned experiment was performed using models with physical characteristics equal to 1 (a=1). It is worth noticing that there is no visible difference between the results of the DMMA, EFMA and TEFMA, while TDMMA produces small errors in deformation accuracy (Figure 7c). In this example, it is obvious that TEFMA has the best computational time and deformation accuracy.

V. Conclusion

In this paper, we have presented a closed-form solution for 2D physics-based shape model deformation along with its properties. The deformation equations [17] were simplified and analyzed. As a result, a closed-form solution can be reached for the objects deformation. The presented 2D physics-based deformable model drastically reduces the computation time needed to perform the deformation. This solution is very useful in analyzing the deformation behavior of the contour at hand. The extremely low computational time and the low deformation error with respect to all the other available techniques in the literature, makes the introduced physics-based deformable model a very promising tool for various image analysis and computer vision applications.

Appendix

A. Derivation of Closed-Form Governing Equations

The proof of the explicit governing equations is going to be restricted to closed curves. Hence, the physics-based deformable model representing closed curves (Section II), can be

explicitly described by:

$$\boldsymbol{U} = \sum_{i=1}^{N} \sum_{j} \frac{\sum_{n=1}^{N} [F_n \phi_n(j)]}{(1+\omega_j^2) \sum_{n=1}^{N} \phi_n^2(j)} \phi_i(j), \qquad j \in \mathcal{B}(N), \tag{A.1}$$

where ω_i^2 is given by equation (9), $\phi_i(j)$ by equation (10), $\mathcal{B}(N)$ is the first Brillouin zone, and finally F_i denotes the x and y components of the force acting on node i.

$\mathcal{B}(N)$ is equal to $\left\{-\frac{N}{2}+1, \ldots, \frac{N}{2}\right\}$ for N even, and $\left\{-\frac{N-1}{2}, \ldots, \frac{N-1}{2}\right\}$ for N odd. Furthermore, for equal eigenvalues, ω_i^2 (9), the corresponding eigenvectors (10) are modified to consist of sines instead of cosines. Thus, the first eigenvector of the set of equal eigenvalues consists of cosines (as shown in equation 9), the next one of sines $\left(sin\frac{2\pi nj}{N}\right)$, the next of cosines, and so on.

The normalization factor ($\sum_{n=1}^{N} \phi_n^2(i)$) in equation (A.1) can be analyzed [38] to be given by:

$$\mathcal{H} = \sum_{n=1}^{N} \phi_n^2(i) = \begin{cases} \frac{N}{2}, & i \neq 0, \frac{N}{2} \\ N, & i = 0, \frac{N}{2} \end{cases}, \qquad i \in \mathcal{B}(N), \tag{A.2}$$

for both the cosine and sine cases.

Furthermore, it is assumed that the vector $\mathbf{F}$, a $2N \times 1$ vector describing the x and y components of the forces acting on the model nodes, is a vector that consists of zero elements except one, the F_n element. Therefore, only one force, F_n is acting on the model, and more specifically on node n. Thus, equation A.1 can now be simplified into:

$$\boldsymbol{U} = \sum_{i=1}^{N} \sum_{j} \frac{F_n \phi_n(j)}{(1+\omega_j^2)\mathcal{H}} \phi_i(j), \qquad j \in \mathcal{B}(N), \tag{A.3}$$

and the deformation U_i of node i – assuming that the only force acting on the model is F_n – is described by:

$$U_i = \sum_{j} F_n \frac{\phi_n(j)\phi_i(j)}{(1+\omega_j^2)\mathcal{H}}, \qquad j \in \mathcal{B}(N). \tag{A.4}$$

Due to the fact that $j \in \mathcal{B}(N)$, there exist two cases, one for N even, and one for N odd. Consequently, our proof is separated into two distinct cases.

1. **N is even.**

   Since N is even, j lies in the interval $\left\{-\frac{N}{2}+1, \ldots, \frac{N}{2}\right\}$, or even better in $\left\{0, \frac{N}{2}, \pm\left(\frac{N}{2}-1\right), \pm\left(\frac{N}{2}-2\right), \ldots\right\}$. Thus, equation (A.4), becomes:

$$U_i = F_n \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \left[\frac{1}{\mathcal{H}} \frac{\phi_n(j)\phi_i(j)}{(1+4asin^2\frac{j\pi}{N})} \right], \tag{A.5}$$

where a is the characteristic value of the model. Furthermore, equation (A.5) can be rewritten as:

$$\begin{aligned} U_i &= F_n \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} \left[\frac{1}{\mathcal{H}} \frac{\phi_n(j)\phi_i(j)}{(1+4asin^2\frac{\pi j}{N})} \right] \\ &= F_n \left[\frac{1}{N} \frac{\phi_n(0)\phi_i(0)}{1+4asin^2\frac{\pi*0}{N}} + \frac{1}{N} \frac{\phi_n(\frac{N}{2})\phi_i(\frac{N}{2})}{1+4asin^2\frac{\pi\frac{N}{2}}{N}} + \sum_{\substack{j=-\frac{N}{2}+1 \\ j\neq 0,\frac{N}{2}}}^{\frac{N}{2}} \frac{2}{N} \frac{\phi_n(j)\phi_i(j)}{1+4asin^2\frac{\pi j}{N}} \right] \\ &= F_n \left[\frac{1}{N} + \frac{(-1)^{n-i}}{N(1+4a)} + \frac{2}{N} \sum_{j=1}^{\frac{N}{2}-1} \left(\frac{cos\frac{2\pi nj}{N} cos\frac{2\pi ij}{N}}{1+4asin^2\frac{\pi j}{N}} + \frac{sin\frac{2\pi nj}{N} sin\frac{2\pi ij}{N}}{1+4asin^2\frac{\pi j}{N}} \right) \right] \\ &= F_n \left[\frac{1}{N} + \frac{(-1)^{n-i}}{N(1+4a)} + \frac{2}{N} \sum_{j=1}^{\frac{N}{2}-1} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}} \right]. \end{aligned} \tag{A.6}$$

Since, the formula $\frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}}$ is symmetrical with respect to j factor, the formula produces the same result for j and for N-j. Thus, we can write:

$$\begin{aligned} U_i &= F_n \left[\frac{1}{N} + \frac{(-1)^{n-i}}{N(1+4a)} + \frac{1}{N} \sum_{j=1}^{\frac{N}{2}-1} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}} + \frac{1}{N} \sum_{j=\frac{N}{2}+1}^{N-1} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}} \right] \\ &= F_n \left[\frac{1}{N} \frac{cos\frac{2\pi(n-i)*0}{N}}{1+4asin^2\frac{\pi*0}{N}} + \frac{1}{N} \frac{cos\frac{2\pi(n-i)\frac{N}{2}}{N}}{1+4asin^2\frac{\pi\frac{N}{2}}{N}} + \frac{1}{N} \sum_{j=1}^{\frac{N}{2}-1} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}} + \frac{1}{N} \sum_{j=\frac{N}{2}+1}^{N-1} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}} \right] \\ &= F_n \frac{1}{N} \sum_{j=0}^{N-1} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}}. \end{aligned} \tag{A.7}$$

Therefore, for N even, deformations of the physics-based model, provided that only one force, F_n, is applied to it (to node n), are now described by equation (A.7).

2. N is odd.

For N odd, j lies in the interval $\left\{-\frac{N-1}{2}, \ldots, \frac{N-1}{2}\right\}$, or even better in $\left\{0, \pm\left(\frac{N-1}{2}\right), \pm\left(\frac{N-1}{2}-1\right), \ldots\right\}$. Thus, equation (A.4), for N odd becomes:

$$U_i = F_n \sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} \left[\frac{1}{\mathcal{H}} \frac{\phi_n(j)\phi_i(j)}{(1+4asin^2\frac{\pi j}{N})} \right], \tag{A.8}$$

where a is the model characteristic value. Hence, equation (A.8) can be written as:

$$\begin{aligned} U_i &= F_n \sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} \left[\frac{1}{\mathcal{H}} \frac{\phi_n(j)\phi_i(j)}{(1+4asin^2\frac{\pi j}{N})} \right] \\ &= F_n \left[\frac{1}{N} \frac{\phi_n(0)\phi_i(0)}{1+4asin^2\frac{\pi*0}{N}} + \sum_{\substack{j=-\frac{N-1}{2} \\ j\neq 0}}^{\frac{N-1}{2}} \frac{2}{N} \frac{\phi_n(j)\phi_i(j)}{1+4asin^2\frac{\pi j}{N}} \right] \\ &= F_n \left[\frac{1}{N} + \frac{2}{N} \sum_{j=1}^{\frac{N-1}{2}} \left(\frac{cos\frac{2\pi nj}{N} cos\frac{2\pi ij}{N}}{1+4asin^2\frac{\pi j}{N}} + \frac{sin\frac{2\pi nj}{N} sin\frac{2\pi ij}{N}}{1+4asin^2\frac{\pi j}{N}} \right) \right] \\ &= F_n \left[\frac{1}{N} + \frac{2}{N} \sum_{j=1}^{\frac{N-1}{2}} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}} \right]. \end{aligned} \tag{A.9}$$

Furthermore, as previously mentioned, the formula $\frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}}$ is symmetrical with respect to factor j. That is, the formula produces the same results for j and for N-j.

Thus, we can write:

$$\begin{aligned} U_i &= F_n \left[\frac{1}{N} + \frac{1}{N} \sum_{j=1}^{\frac{N-1}{2}} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}} + \frac{1}{N} \sum_{j=\frac{N+1}{2}}^{N-1} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}} \right] \\ &= F_n \left[\frac{1}{N} \frac{cos\frac{2\pi(n-i)*0}{N}}{1+4asin^2\frac{\pi*0}{N}} + \frac{1}{N} \sum_{j=1}^{N-1} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}} \right] \\ &= F_n \frac{1}{N} \sum_{j=0}^{N-1} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}}. \end{aligned} \tag{A.10}$$

Therefore, for N odd, deformations of the physics-based model, provided that only one force F_n is applied to it (to node n), are now described by equation (A.10)

Thus, for both cases, deformations of the model (eq. A.7 and A.10), are given by:

$$U_i = F_n \frac{1}{N} \sum_{j=0}^{N-1} \frac{cos\frac{2\pi(n-i)j}{N}}{1+4asin^2\frac{\pi j}{N}}. \tag{A.11}$$

Now, if we associate variable i (the node under examination) and n (the node where the force F_n is applied to), the model can be fatherly simplified. Then, we assume that node i is related to node n by $i = n + d$, where d is the distance (in nodes) between nodes i and n. For example, if d is equal to zero, the node under examination is the same with the one that force F_n is applied to. Hence, the above equation (A.11) can be rewritten:

$$U_{n+d} = F_n \frac{1}{N} \sum_{j=0}^{N-1} \frac{cos\frac{2\pi dj}{N}}{1+4asin^2\frac{\pi j}{N}}. \tag{A.12}$$

Analyzing the above equation (Section B), we conclude that it is equivalent to:

$$U_{n+d} = F_n \frac{a^d}{\sqrt{1+4a}} \frac{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N-2d} + \left(\frac{\sqrt{1+4a}-1}{2}\right)^{N-2d}}{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N} - \left(\frac{\sqrt{1+4a}-1}{2}\right)^{N}}. \tag{A.13}$$

Therefore, the physics-based deformable model with respect to closed curves (eq. A.1), can be fully described by:

$$\boldsymbol{U} = \frac{1}{\sqrt{1+4a}} \sum_{i=1}^{N} \sum_{j=1}^{N} F_j a^d \frac{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N-2d} + \left(\frac{\sqrt{1+4a}-1}{2}\right)^{N-2d}}{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N} - \left(\frac{\sqrt{1+4a}-1}{2}\right)^{N}}, \tag{A.14}$$

where a is the model characteristic value, N is the total number of model nodes, and finally, d is the distance (in nodes) between the node under examination and the node that force F_i is applied to. Hence, in practice, d is equal to $min(|i-j|, |N-|i-j||)$.

B. Summation Reduction of the Basic Deformation Function

In this Section, our goal is to simplify equation (A.12), into an equivalent, compact equation, such that the calculation involves no summations or any other computationally intensive operations.

First of all, suppose that there exists:

$$\begin{aligned}\mathcal{F}(N,a,d) &= \tfrac{1}{N}\sum_{j=0}^{N-1}\tfrac{cos\frac{2\pi dj}{N}}{1+4asin^2\frac{\pi j}{N}}\\ &= \tfrac{1}{N}\sum_{j=0}^{N-1}\tfrac{cos\left(\frac{2\pi j}{N}+\frac{2\pi(d-1)j}{N}\right)}{1+2a-2acos\frac{2\pi j}{N}}\\ &= \underbrace{\frac{1}{N}\sum_{j=0}^{N-1}\frac{cos\frac{2\pi j}{N}cos\frac{2\pi(d-1)j}{N}}{1+2a-2acos\frac{2\pi j}{N}}}_{G} - \underbrace{\frac{1}{N}\sum_{j=0}^{N-1}\frac{sin\frac{2\pi j}{N}sin\frac{2\pi(d-1)j}{N}}{1+2a-2acos\frac{2\pi j}{N}}}_{Q}.\end{aligned} \tag{B.1}$$

More specifically, the above function (B.1) separated into two parts, G and Q. Hence, part G can be analyzed to:

$$\begin{aligned}G &= \tfrac{1}{N}\sum_{j=0}^{N-1}\tfrac{cos\frac{2\pi j}{N}cos\frac{2\pi(d-1)j}{N}}{1+2a-2acos\frac{2\pi j}{N}}\\ &= \tfrac{1}{N}\sum_{j=0}^{N-1}\tfrac{-2acos\frac{2\pi j}{N}cos\frac{2\pi(d-1)j}{N}}{-2a\left(1+2a-2acos\frac{2\pi j}{N}\right)}\\ &= \tfrac{1}{-2aN}\sum_{j=0}^{N-1}\tfrac{-2acos\frac{2\pi j}{N}cos\frac{2\pi(d-1)j}{N}+[1+2a-(1+2a)]cos\frac{2\pi(d-1)j}{N}}{1+2a-2acos\frac{2\pi j}{N}}\\ &= \tfrac{1}{-2aN}\sum_{j=0}^{N-1}cos\tfrac{2\pi(d-1)j}{N} + \tfrac{1+2a}{2aN}\sum_{j=0}^{N-1}\tfrac{cos\frac{2\pi(d-1)j}{N}}{1+2a-2acos\frac{2\pi j}{N}}\\ &= \tfrac{1}{-2aN}\sum_{j=0}^{N-1}cos\tfrac{2\pi(d-1)j}{N} + \tfrac{1+2a}{2a}\mathcal{F}(N,a,d-1).\end{aligned} \tag{B.2}$$

Moreover, it is known [38] that $\sum_{j=0}^{N-1}cos\frac{2\pi(d-1)j}{N}=0$ for every d, and as a consequence, equation (B.2) can be fatherly simplified to:

$$G=\frac{1+2a}{2a}\mathcal{F}(N,a,d-1). \tag{B.3}$$

In a similar manner, part Q is analyzed to:

$$Q=\frac{1}{2}\mathcal{F}(N,a,d-2)-\frac{1}{2}\mathcal{F}(N,a,d). \tag{B.4}$$

Therefore, equation (B.1) is now described by the recursive equation:

$$\mathcal{F}(N,a,d)=\frac{1+2a}{a}\mathcal{F}(N,a,d-1)-\mathcal{F}(N,a,d-2). \tag{B.5}$$

After solving the above homogeneous recurrence equation [37] and determining its characteristic polynomial and its roots, equation (B.5), can be rewritten in a more simple form, involving no recurrence:

$$\mathcal{F}(N,a,d)=c_1\frac{1}{a^d}\mu^{2d}+c_2\frac{1}{a^d}\lambda^{2d}, \tag{B.6}$$

where μ and λ are two constant values equal to $\frac{\sqrt{1+4a}+1}{2}$ and $\frac{\sqrt{1+4a}-1}{2}$ respectively, and c_1 and c_2 are two parameters related to the initial values of the function under examination. The parameters c_1 and c_2 can be easily estimated by solving the following system of equations:

$$\left.\begin{array}{l} \mathcal{F}(N,a,0) = \mathcal{F}_0 = c_1 + c_2 \\ \mathcal{F}(N,a,1) = \mathcal{F}_1 = c_1\frac{1}{a}\mu^2 + c_2\frac{1}{a}\lambda^2 \end{array}\right\} \begin{array}{l} c_1 = \frac{a\mathcal{F}_1 - \lambda^2\mathcal{F}_0}{\mu^2-\lambda^2} \\ c_2 = -\frac{a\mathcal{F}_1 - \mu^2\mathcal{F}_0}{\mu^2-\lambda^2}. \end{array} \tag{B.7}$$

Therefore, equation (B.6) can be rewritten in the form:

$$\mathcal{F}(N,a,d) = \frac{1}{a^{d-1}\sqrt{1+4a}}\left[\left(\mathcal{F}_1 - \frac{a\mathcal{F}_0}{\mu^2}\right)\mu^{2d} - \left(\mathcal{F}_1 - \frac{a\mathcal{F}_0}{\lambda^2}\right)\lambda^{2d}\right]. \tag{B.8}$$

Furthermore, $\mathcal{F}_1 - \frac{a\mathcal{F}_0}{\mu^2}$ and $\mathcal{F}_1 - \frac{a\mathcal{F}_0}{\lambda^2}$ can be explicitly calculated [39], and they are equal to $-\frac{1}{\lambda}\frac{\mu^{N-1}}{\mu^N-\lambda^N}$ and $\frac{1}{\mu}\frac{\lambda^{N-1}}{\mu^N-\lambda^N}$ respectively. Thus, our initial function (eq. B.1) can be transformed to in its final compact form, accomplishing our goal:

$$\mathcal{F}(N,a,d) = \frac{a^d}{\sqrt{1+4a}}\frac{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N-2d} + \left(\frac{\sqrt{1+4a}-1}{2}\right)^{N-2d}}{\left(\frac{\sqrt{1+4a}+1}{2}\right)^{N} - \left(\frac{\sqrt{1+4a}-1}{2}\right)^{N}}. \tag{B.9}$$

REFERENCES

[1] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE trans. Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

[2] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects. In *Proceedings of the European Conference on Computer Vision.*, Santa Margherita Ligure, Italy, May 1992.

[3] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[4] A. Pentland. Perceptual organization and representation of natural form. *Artificial Intelligence*, 28(3):293–331, 1986.

[5] A. Pentland and S. Sclaroff. Closed-form solutions for physically-based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):715–729, July 1991.

[6] L. Staib and J. Duncan. Parametrically deformable contour models. In *Proceedings of Computer Vision and Pattern Recognition*, pages 98–103, 1989.

[7] D. Terzopoulos and D. Metaxas. Dynamic 3-D models with local and global deformations: Deformable superquadrics. *IEEE trans. Pettern Analysis and Machine Intelligence*, 13(7):703–714, July 1991.

[8] D. Terzopoulos, A. Witkin, and M. Kass. Constrains on deformable models: Recovering 3D shape and mon-rigid motion. *Artificial Intelligence*, 36:91–123, 1988.

[9] G. Gubsol, J. P. Thirion, and N. Ayache. A scheme for automatically building 3D morphometric anatomical atlases: Application to skull atlas. *Medical Image Analysis*, 2(1):37–60, 1998.

[10] J. P. Thirion. Image matching as a diffusion process: An analogy with Maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[11] S. Sclaroff and A. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):545–561, June 1995.

[12] C. Dorai and A. Jain. COSMOS - A representation scheme for 3D free-form objects. *IEEE trans. Pattern Analysis and Machine Intelligence*, 19(10):1115–1130, 1997.

[13] A. Aguado, M. Nixon, and M. Montiel. Parameterizing arbitrary shapes via fourier descriptors for evidence-gathering extraction. *Computer Vision and Image Understanding*, 69(2):202–221, Feb 1998.

[14] D. Terzopoulos, J. Platt, and K. Fleischer. Elastically deformable models. *ACM Computer Graphics (SIGGRAPH'87)*, 21(4):205–214, 1987.

[15] J. Mallet. Discrete smooth interpolating in geometric modeling. *Computer Aided Design (CAD)*, 24(4):178–192, 1992.

[16] H. Delingette, G. Subsol, and S. Cotin. A craniofacial surgery testbed. Technical Report 2199, INRIA Sophia Antipolis, France, 1992.

[17] C. Nastar and N. Ayache. Frequency-based nonrigid motion analysis: Application to four dimensional medical images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1069–1079, 1996.

[18] A. Blake and I. Micheal. 3D position, attitude and shape input using video tracking of hands and lips. In *ACM Computer Graphics (SIGGRAPHS'94)*, pages 185–192, Orlando, USA, 1994.

[19] A. Kelemen, G. Szekely, and G. Gerig. 3D model-based segmentation of brain mri. In *Workshop on Biomedical Image Analysis (WBIA'98)*, pages 4–13, Santa Barbara, USA, June 1998.

[20] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models for a 3-D object reconstruction. *International Journal of Computer Vision*, 1(3):211–221, Oct. 1987.

[21] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for a 2-D and 3-D images. *IEEE trans. Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, Nov 1993.

[22] A. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4(2):107–126, March 1990.

[23] F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):131–147, 1990.

[24] T. Boult and A. Gross. Recovery of superquadrics from depth information. In *Proceedings of the AAAI Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pages 128–137, Oct. 1987.

[25] F. Leymarie and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):617–634, June 1993.

[26] K.Sobottka and I.Pitas. A novel method for automatic face segmentation, facial feature extraction and tracking. *Image Communication, Elsevier*, 12(3):263–281, June 1998.

[27] C. Nikou, G. Bueno, F. Heitz, and J. P. Armspach. A joint physics-based statistical deformable model for multimodal brain image analysis. *IEEE Transactions on Medical Imaging*, 20(10):1026–1037, 2001.

[28] C. Nastar and N. Ayache. Fast segmentation, tracking, and analysis of deformable objects. In *Proceedings of the Fourth International Conference on Computer Vision (ICCV'93)*, pages 11–14, Berlin, Germany, May 1993.

[29] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, February 1995.

[30] C. Basdogan. Real-time simulation of dynamically deformable finite element models using modal analysis and spectral lanczos decomposition methods. In *Proceedings of the Medicine Meets Virtual Reality (MMVR'01)*, pages 46–52, January 2001.

[31] A. Pentland and B. Horowitz. Recovery of non-rigid motion and structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):730–742, July 1991.

[32] K. J. Bathe. *Finite Element Procedure*. Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[33] C. Nastar. *Models physiques deformables et modes vibratoires pour l'analyse du mouvement non-rigide dans les images multidimensionnelles*. Ph.D. Thesis, Ecole Nationale des Ponts et Chaussees, 1994.

[34] G. Borgefors. On digital distance transforms in three dimensions. *Computer Vision and Image Understanding*, 64(3):368–376, 1996.

[35] Per-Erik Danielsson. Euclidean distance transform. *Computer Graphics and Image Processing*, 14:227–28, 1980.

[36] J. Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[37] G. Brassard and P. Bratley. *Funtamentals of Algorithm*. Englewood Cliffs, New Jersey, 1996.

[38] M. R. Spiegel. *Mathematical Formulas*. McGraw-Hill, New York, 1996.

[39] L. B. W. Jolley. *Summation of Series*. Dover Publications, New York, 1961.

Stelios Krinidis was born in Kavala, Greece, in 1978. He received the Diploma in Informatics from the Aristotle University of Thessaloniki, Greece, in 1999. In June 2000, he joined the Artificial Intelligence and Information Analysis group at the same university, where he is currently a researcher and teaching assistant and studies towards his Ph.D. degree on 3D image processing. His research interests include 3D image processing and analysis and computer vision.

Ioannis Pitas (SM'94) received the Diploma of Electrical Engineering in 1980 and the PhD degree in Electrical Engineering in 1985 both from the University of Thessaloniki, Greece. Since 1994, he has been a Professor at the Department of Informatics, University of Thessaloniki. From 1980 to 1993 he served as Scientific Assistant, Lecturer, Assistant Professor, and Associate Professor in the Department of Electrical and Computer Engineering at the same University. He served as a Visiting Research Associate at the University of Toronto, Canada, University of Erlangen-Nuernberg, Germany, Tampere University of Technology, Finland, as Visiting Assistant Professor at the University of Toronto and as Visiting Professor at the University of British Columbia, Vancouver, Canada. He was lecturer in short courses for continuing education. His current interests are in the areas of digital image processing, multidimensional signal processing, watermarking and computer vision. He has published over 380 papers and contributed in 13 books in his areas of interest. He is the co-author of the books "Nonlinear Digital Filters: Principles and Applications" (Kluwer, 1990), "3-D Image Processing Algorithms" (J. Wiley, 2000), Nonlinear Model-Based Image/Video Processing and Analysis (J. Wiley, 2001) and author of "Digital Image Processing Algorithms and Applications" (J. Wiley, 2000). He is the editor of the book "Parallel Algorithms and Architectures for Digital Image Processing, Computer Vision and Neural Networks" (Wiley, 1993). Dr Pitas has been member of the European Community ESPRIT Parallel Action Committee. He has also been an invited speaker and/or member of the program committee of several scientific conferences and workshops. He was Associate Editor of the IEEE Transactions on Circuits and Systems, Associate Editor of the IEEE Transactions on Neural Networks, co-editor of Multidimensional Systems and Signal Processing and he is currently an Associate Editor of the IEEE Transactions on Image Processing. He was general chair of the 1995 IEEE Workshop on Nonlinear Signal and Image Processing (NSIP95), technical chair of the 1998 European Signal Processing Conference and general chair of IEEE ICIP2001.