TR - H - 127

Applying Energy-Minimization Splines To X-Ray Vocal Tract Images

Frédérique Garcia
Mark K. Tiede
Kevin G. Munhall
Eric Vatikiotis-Bateson

1995. 2. 7

ATR人間情報通信研究所

〒619-02　京都府相楽郡精華町光台2-2　☎ 0774-95-1011

ATR Human Information Processing Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan
Telephone: +81-774-95-1011
Facsimile: +81-774-95-1008

Applying Energy-Minimization Splines To X-Ray Vocal Tract Images

Frédérique GARCIA
Kevin G. Munhall
Mark K. Tiede
Eric Vatikiotis-Bateson

ATR Human Information Processing Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun
Kyoto 619-02 JAPAN

ABSTRACT

Snakes are energy-minimization splines, which have been applied successfully to a variety of visual image recognition tasks. We have implemented a version of this physical model for the analysis of sequences of X-ray ciné images of the vocal tract and video images of the face recorded during speech. After introducing the basic theory of the snake, and describing its implementation, the software interface developed for the Apple Macintosh™ platform is discussed. Then, some results achieved with these two types of images are demonstrated. It is shown that the snake can follow the tongue surface automatically through a sequence of images even though image quality is quite poor, and that it can follow the complex profile of the lips and chin in the video images. Finally, we discuss several limitations of the method, both temporary and inherent, as well as the sort of improvements we expect to implement in the future.

1. INTRODUCTION

Two separate problems face speech researchers trying to measure the vocal tract. First, the vocal tract is difficult to image. Often, invasive techniques are required and no current technology can provide accurate measures of the moving vocal tract in 3-D. Second, it is difficult to extract measures from the available vocal tract images, because the surfaces of the articulators are often not clear and there are frequent occlusions of surface boundaries. In this report we address the second problem of measuring vocal tract images by describing preliminary efforts to adapt an energy minimizing spline technique (SNAKES; Kass, Witkin, & Terzopoulos, 1988) to digitally processed speech images. In particular, we demonstrate the use of snakes on sequences of midsagittal X-ray images of the vocal tract and to video images of the face during speech.

We begin with a brief overview of vocal tract motion and a discussion of the importance of articulatory information to various applications. Next, we discuss the unique measurement problems that face analysis of vocal tract images. Then, we give an overview of the energy minimizing SNAKE procedures. Finally, we describe the software interface that we have developed to analyze X-ray and video images, providing several examples of the application of the technique.

1.1 Vocal Tract Motion During Speech

The human vocal tract consists of rigid (hard-palate and maxilla), semi-rigid (rear pharyngeal wall), and deformable (soft-palate, tongue, and lips) structures. It is a biologically unique structure whose integrity as a tube resonator is maintained while undergoing a wide range of changes in shape and length. The large repertoire of vocal tract shapes used for speech production, ranges from the fairly uniform tube (e.g., the vowel *schwa*) to complex shapes with multiple constrictions (e.g., labiovelar consonants such as /w/). In general, vowels and consonants are categorically distinguishable based on whether they exploit the resonance properties of the open tube (vowels) or constrictions along its length (consonants).

Although the soft-palate, epiglottis, and rear pharyngeal wall can affect the cross-sectional width of the vocal tract and the lips can change the effective length of the tract or close it off, by the far the most interesting and variable articulator is the tongue. As shown in Figure 1 below, the tongue surface extends from the lower pharynx to the tongue tip, thus constituting almost the entire ventral side of the vocal tract, about 13-15 cm. Because of its length and inverted-"L" (Γ) shape, deformation of the tongue can open one portion of the tract, while closing another. In the figure, the midsagittal cross-sections of the vocal tract and their associated spectra are shown for the three 'point' vowels, /a, i, u/. When the tongue is lowered and retracted for /a/, a large front cavity is formed; alternately, when the tongue is raised and fronted for /i/, a large pharyngeal cavity results. These different configurations of the vocal tract produce different acoustic spectra such as those shown on the right.

While static images of exaggerated postures for specific sounds reveal high quality volume and shape information, it is the vocal tract's dynamic characteristics that need to be known in order to understand either speech motor control or the articulator-acoustic transform. Attempts to the study the tongue's role in speech production have been hampered by its complex and largely inaccessible behavior and physiology. Very little is known about the intrinsic musculature responsible for controlling tongue shape. Recordings of muscular activity have been restricted

largely to extrinsic muscles responsible for moving the tongue relative to other structures such as the hyoid bone, larynx, and jaw (e.g., Baer, Alfonso & Honda 1988). Kinematic studies of tongue behavior have been hampered by the difficulty in making measurable observations of the relevant structures at the required speed. There are various techniques (e.g., X-ray microbeam and electromagnetometer) available for making high-speed (i.e., > 100 Hz) flesh-point measures, but these are typically restricted to the anterior tongue, whose correlation with the posterior (dorsal and pharyngeal) tongue is not established. Probably the best means of recording tongue and, for that matter, vocal tract behavior has been x-ray ciné. However, practical analytic tools for image processing and analysis have become available only recently.

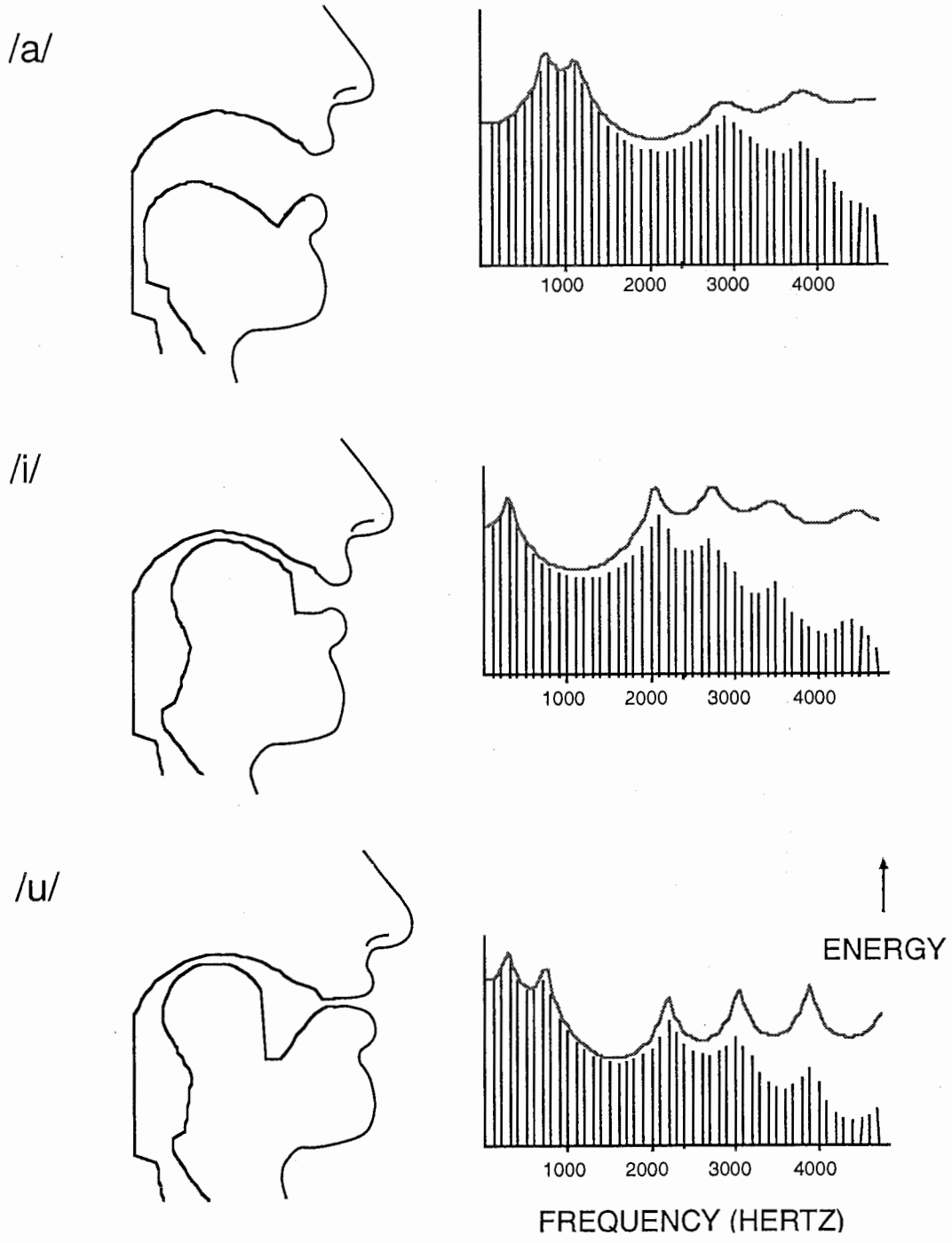


Figure 1. Vocal tract configurations for the three English 'corner' vowels /a, i, u/ are shown on the left. The frequency and energy amplitude of the associated vocal tract filter (solid line) and output spectra (vertical lines) are shown on the right. This figure is from Rubin and Vatikiotis-Bateson (in press; used by permission).

Accurate descriptions of vocal tract shape and motion are useful in a number of basic and applied research areas. Information about the basic form of the oral cavities and movement characteristics is essential for progress in oral anatomy and oral physiology. Clinical disciplines such as dentistry and speech pathology also rely on this information to develop therapeutic intervention strategies. A more recent use for information about articulation is in techniques for automatic recognition of speech and articulatory synthesis. When recognition systems are augmented with information about the motion of speech articulators, there are significant improvements in performance (Rose, Schroeter, Sondhi, & Ghitza, 1994). In speech synthesis high quality synthesis can be achieved by manipulating a computer simulation of the vocal tract in motion (e.g., Hirayama, Vatikiotis-Bateson & Kawato, 1993). Progress in each of these fields depends on the accurate measurement of a large body of vocal tract images.

1.2 Image Processing In Speech Research

In the past, most measurements of X-ray images (e.g., Perkell, 1969) and facial images (e.g., Fujimura, 1961) in speech were carried out manually, frame-by-frame. This is extremely time-consuming and the amount of labor required to measure even a few minutes of speech severely limits the data that can be processed. Manual processing of each image separately has an additional limitation. When a sequences of images is animated our visual systems take advantage of the coherence across the images to detect edges and object boundaries. This use of 'structure from motion' is lost when image sequences are viewed one image at a time. As a result, the tracing of some surfaces in still images becomes a much more difficult and potentially more error prone task. Our long-term goal, then, is automatic measurement of image sequences. In addition we would like to be able to automatically estimate the time-varying parameters that describe the shape of the vocal tract. There are a number of difficulties in achieving this goal and this project is only a preliminary step.

1.2.1 X-ray images —Midsagittal X-ray images still offer the best overall view of the moving vocal tract, but new films of normal subjects are seldom filmed anymore because of concerns about biohazards due to the high X-ray dosage. There exists, however, a large body of films that were recorded over the past 40 years (Dart, 1987; Munhall, Vatikiotis-Bateson, & Tohkura, 1994; submitted). Although these films are an invaluable research resource, digitally processing them presents a number of difficulties. The images on single frames are often noisy and image boundaries for a given articulator can be obscured by superimposed structures. For example, the mandible and radio-opaque fillings in the teeth can occlude portions of the tongue surface. Noise due to scintillation adds a random noise to the images. In addition, soft tissue structures like the tongue, velum and lips can be only poorly imaged with very low contrast boundaries because of saturation of the film emulsion.

The edges of the vocal tract can be indistinct for other reasons as well. As indicated above, the vocal tract is a three-dimensional acoustic tube. Unlike magnetic resonance imaging (MRI), which provides discrete and non-overlapping slices of the body, midsagittal X-ray imaging projects views of the whole head and neck area onto a single plane. The tongue grooves considerably during the production of some vowels (Stone, 1990) and when this behavior is viewed in a midsagittal X-ray, the grooving can produce multiple visible edges for the tongue surface. The superimposition of transparent soft tissue structures such as the tongue and more opaque bony structures such as the jaw, thus can produce a series of overlapping and interrupting edges.

The motion of the vocal tract within the images is equally complicated. Most vocal tract motion is non rigid because the articulators deform as well as change position in the image plane. Thus, for each image frame the shape of many of the articulators as well as their positions must be determined anew. Because of this and the general image problems described above, it will be necessary to augment edge detection approaches with some type of model-based image processing or smoothness constraint. Similar conclusions have been reached in other work on non ridgid motion (e.g., Staib & Duncan, 1990; Pentand & Horowitz, 1991). The work described here applies external smoothness constraints to the image processing and by doing so overcomes some of the image processing problems outlined above.

1.2.2 Facial Images — The analysis of facial images is an active area of speech research (Benoît, Lallouache, Mohamadi, & Abry, 1992), the study of emotion (Ekman & Friesen, 1975) as well as in applied animation (Waters & Terzopoulos, 1992) and video teleconferencing research (Toelg & Poggio, 1994). Facial image processing shares many of the problems of X-ray image processing. Edges can be difficult to measure because of viewing angle, lighting and shading problems, specularities, and the complexity of the geometric projection. Image boundaries can be indistinct because the features of the face are three-dimensional and the boundaries in their 2-D projections can reflect curved surfaces in the third dimension. Measurements of the oral aperture also have the problem that the teeth and tongue surface may be visible in the image and they can match the lip surface in brightness.

As in the interior vocal tract, the motion is generally non rigid. Thus, the shape and position of the structures must be determined frame by frame. These problems also suggest that simple edge detection procedures are unlikely to be successful without some external constraint (see also Yuille, Cohen, & Halliman, 1989). As in the x-ray images we will apply smoothness constraints to the task of identifying facial features.

2. SNAKES

In this section, we give a brief functional overview of the energy minimizing spline or 'snake' algorithm developed by Kass *et al.*, (1988). The main purpose here is to orient the reader to the basic dynamic characteristics of the algorithm and to preface the discussion of its implementation given in Section 3. Parts of the discussions in Sections 2 and 3 are paraphrased from Kass *et al.*, (1988) and Begin (1993).

2.1 The Snake Algorithm

Kass *et al.*, (1988) demonstrated the usefulness of the snake for interactive detection of image contours. Later, they developed 3D deformable object models (Terzopoulos & Fleischer, 1988) incorporating physical characteristics that made them capable of both elastic and inelastic behaviors such as viscoelasticity, plasticity and fracture. These models are governed by the mechanical laws of continuous bodies, being acted upon by both internal forces that afford resistance to deformation, and external forces that move and deform them from their original shape, thereby yielding realistic dynamics. The basic snake is a deformable curve under the influence of internal forces due to its elasticity, and external forces set by edges in the image and by user-defined springs. Despite its elasticity, the snake can

still exhibit a small amount of viscoelastic behavior. The external forces are computed from the image intensity levels (Kass *et al* , 1988).

The snake is represented parametrically by the equation:

$$\mathrm{v}(s) = (\mathrm{x}(s), \mathrm{y}(s)), \tag{1}$$

where s is the intrinsic coordinate of the snake in its own domain $\Omega = [0,1]$. Coefficients x and y relate the snake coordinates to an inertial reference, ϕ, in two dimensional Euclidean space, in this case the image coordinates (in pixels). Thus, $\mathrm{v}(s)$ is a vector-valued function of material coordinates.

The energy of the snake is

$$\mathrm{E} = \int_{\Omega} \mathrm{E}_{int}(\mathrm{v}(s)) + \mathrm{E}_{ext}(\mathrm{v}(s))ds \tag{2}$$

where E_{int} is the internal energy of the snake, and E_{ext} the external energy, due to the image and interactive constraints (springs). The snake finds an equilibrium position by minimizing its total energy.

We see in the following sections how the internal and external energies are defined, and we show in Section 3.3 how this model is implemented after discretization of the snake.

2.2 The Internal Force

The internal energy is given by :

$$\mathrm{E}_{int} = (\omega_1(s)\left|\frac{\partial \mathrm{v}(s)}{\partial s}\right|^2 + \omega_2(s)\left|\frac{\partial^2 \mathrm{v}(s)}{\partial s^2}\right|^2)/2 \tag{3}$$

where ω_1 and ω_2 are functions of the snake's intrinsic coordinates, and control the importance of each term (first-order and second-order) in the internal constraints.

2.3 External Forces

The snake moves in the image by minimizing its energy. In this section we see how to define, from the image, an energy function which attracts the snake to salient features in images.

2.3.1 Image forces — Kass *et al.* (1988) propose three different functions for attracting a snake to lines, edges or terminations. For our purposes, it is sufficient for the snake to be attracted by edges, though it is possible to use a weighted combination of the three functions:

$$\mathrm{E}_{image} = W_{line}E_{line} + W_{edge}E_{edge} + W_{term}E_{term} \tag{4}$$

Given an intensity function, $\mathbf{I}(x, y)$, we compute the energy function for the image, whose local minima correspond to edges in the image. The local minima can be used to attract the snake at a distance. This is due in part to the elasticity of the snake; if part of the snake finds a local minimum in the energy function, it

attracts the rest of the snake, thanks to its internal energy. But this presupposes that the original snake was set very close to the edge, and was specified with high elasticity and stiffness values. This leads to a problem when working with winding edge contours.

For a smooth edge, $-|\nabla \mathbf{I}(x,y)|^2$ is a good energy function, in which ∇ is the gradient, and $\mathbf{I}(x, y)$ is the intensity level of the pixel at coordinates x, y. But if the change of intensity is very rapid, then the force field will be narrow, so the snake has to be set very close to the edge to remain in the force field. Therefore, the input image should be smoothed in order to enlarge the force field corresponding to edges.

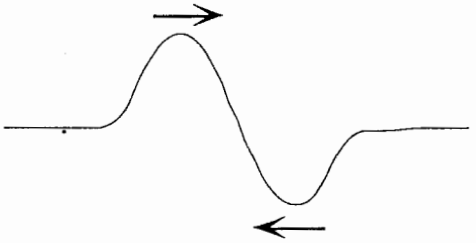
A good way to smooth the signal is to convolve it with a Gaussian:

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \tag{5}$$

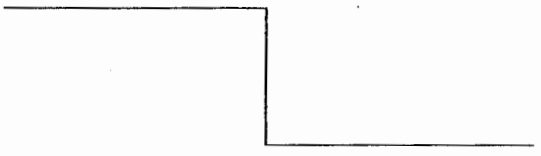
Let's suppose we have an edge defined by a rapid change in intensity level, as shown below on the left, and we convolve it by a Gaussian; we obtain the smoothed signal shown on the right.

Next, we compute the gradient, depicted by the first derivative:

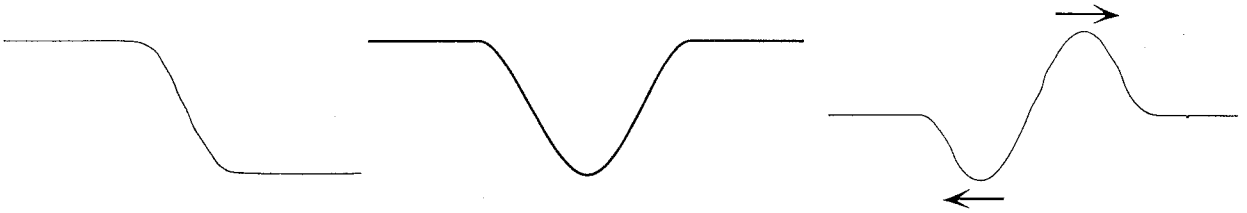
A force field is needed, which is positive to the left of the edge and negative to the right, to push the snake to the edge. We obtain this result by computing the second derivative (arrows show the direction of the forces):



But suppose we have this edge :



We will obtain a second derivative having the wrong order of signs for the force:



From left to right, the derivation of the smoothed edge and the velocity results in an acceleration profile, whose forces will push the snake away from the edge.

We just have to take the absolute value of the first derivative to take care of this problem. Then the force to compute is :

$$F = \nabla(|\nabla(G * I)|) \tag{6}$$

The way this function is implemented and the influence of the Gaussian variance on the result is discussed in Section 3.2.

2.3.2 Springs — Selective external forces may be exerted on the nodes of the snake by attaching springs. One end of a spring is connected to a node of the snake, and the other end is anchored to a fixed point in the image. The external force exerted on the node is:

$$-k(x_1 - x_2)^2$$

where x_1 and x_2 are the positions of the two ends of the spring.

3. IMPLEMENTATION

3.1 Overview

The software described here was written in C (Metrowerks CodeWarrior ™) on a Power Macintosh. The version of the snake code implemented in the software was provided by Dimitri Terzopoulos.

3.2 Image Forces Computation

3.2.1 Smoothing — The first step in computing the image force is to average the image by a Gaussian convolution. In the original source code, this binomial convolution was done by applying the following mask :

$$\begin{bmatrix} 2 & 2 & 2 \\ 2 & 4 & 2 \\ 2 & 2 & 2 \end{bmatrix} /8$$

But, as this computation must be done many times, the resulting computation time was very long. So, this part of the code was altered by substituting the mask:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 8 & 2 & 0 \\ 1 & 8 & 20 & 8 & 1 \\ 0 & 2 & 8 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} / 64$$

A similar result is realized by applying this mask half the number of times. Although the complexity of the computation remains $O(n^2)$, the time saved convolving large images is noticeable.

3.2.2 First gradient — For all points of the image, we compute the magnitude of the first gradient:

$$\mathrm{Grad}(x,y) = \sqrt{\left(\frac{\mathrm{I}(x+1,y)-\mathrm{I}(x-1,y)}{2}\right)^2 + \left(\frac{\mathrm{I}(x,y+1)-\mathrm{I}(x,y-1)}{2}\right)^2}$$

To have values between [0,255] we compute

$$\mathrm{GradM}(x,y) = \frac{\mathrm{Grad}(x,y) * 255}{\mathrm{Grad}_{max}}$$

These two steps, the averaging and the first gradient, are time-consuming computations, so the program offers the possibility of storing the result in a PICT file (see Section 4, below).

3.2.3 Second gradient — Next, we compute the second gradient:

$$\mathrm{F} = \nabla(|\nabla(G * \mathrm{I})|)$$

In order to decompose the equations of motion (explained in Section 3.3, below), the gradient components of the potential function, *df-dx* and *df-dy*, must be computed. These values do not need to be computed for all the points of the image. Rather, we compute the second derivative for each node of the snake, at each step. These values are computed using bilinear interpolation between the four neighboring pixels of each snake node.

3.3 Snake

3.3.1 Semi-discretization — We discretize the snake s by a distribution of $\mathbf{N}$ nodes. We assume that the interspace defined by $h = 1/(N+1)$ is constant. Nodes are indexed by integers n, with $0 \le n \le \mathbf{N}$.

The snake is represented by a vector of $\mathbf{N}$ elements:

$$\underline{v} = (v[1], v[2], \ldots, v[\mathbf{N}])$$

To solve Equation 3 (Section 2.2), we approximate the derivatives with finite differences. Using the discretization of the snake body, we can write the internal energy for the node i of the snake:

$$E_{\text{int}}(i) = \omega_1(i)\frac{|v_i - v_{i-1}|^2}{2h^2} + \omega_2(i)\frac{|v_{i-1} - 2v_i + v_{i+1}|^2}{2h^4} \tag{7}$$

Discretizing the energy functional $\mathrm{E} = \int_{\Omega} \mathrm{E}_{int}(\mathrm{v}(s)) + \mathrm{E}_{ext}(\mathrm{v}(s))ds$ leads to the equation:

$$\mathrm{E} = \sum_{i=1}^{\mathrm{N}} \mathrm{E}_{int}(i) + \mathrm{E}_{ext}(i) \tag{8}$$

We assume in the following that the coefficients ω_1 and ω_2 are constant within the snake (see discussion in Section 6). The minimization of this equation using finite difference, gives rise to:

$$\begin{aligned} &\mathrm{v}_i(2\omega_1 + 6\omega_2) - \mathrm{v}_{i-1}(\omega_1 + 4\omega_2) \\ &-\mathrm{v}_{i-1}(\omega_1 + 4\omega_2) + \mathrm{v}_{i-2}\,\omega_2 + \mathrm{v}_{i-2}\,\omega_2 + \mathrm{f}(i) = 0 \end{aligned} \tag{9}$$

where $\mathrm{f}(i) = \dfrac{\partial \mathrm{E}_{ext}}{\partial \mathrm{v}_i}$.

For the computation, two projections on the x and y axes are used. The vector $\underline{v}$ is therefore decomposed into two vectors, $\underline{x}$ and $\underline{y}$.

Equation (9) can be written in matrix form :

$$\begin{aligned} \mathbf{A}\underline{x} + \underline{\mathrm{f}}_x &= 0 \\ \mathbf{A}\underline{y} + \underline{\mathrm{f}}_y &= 0 \end{aligned} \tag{10}$$

where $\underline{\mathrm{f}}_x$ and $\underline{\mathrm{f}}_y$ are N dimensional vectors, and $\underline{\mathrm{f}}_{\mathrm{x}}(\mathrm{i})$ and $\underline{\mathrm{f}}_{\mathrm{y}}(\mathrm{i})$ are the Cartesian components of the external force exerted on node i of the snake. Furthermore,

$$\mathbf{A} = (2\omega_1 + 6\omega_2)\mathbf{I}_{\mathbf{N}} - (\omega_1 + 4\omega_2)\mathbf{J}_{\mathbf{N}} + \omega_2\mathbf{L}_{\mathrm{N}}. \tag{11}$$

$\mathbf{A}$ is the Stiffness matrix. $\mathbf{I}_{\mathbf{N}}$ is the N-dimensional unity matrix. $\mathbf{J}_{\mathbf{N}}$ is an N-dimensional square matrix, which computes v[n] from v[n+1] and v[n-1]:

$$\mathbf{J}_{\mathbf{N}} = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & 1 & \ddots & & 0 \\ 0 & 0 & 1 & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & & \ddots & \ddots & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$\mathbf{L}_{\mathbf{N}}$ is an N-dimensional square matrix, which computes v[n] from v[n+2] and v[n-2]):

$$\mathbf{L_N} = \begin{pmatrix} 0 & 0 & 1 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & & & 0 & 0 \\ 1 & 0 & 0 & 0 & \ddots & & 0 & 0 \\ 0 & 1 & 0 & 0 & \ddots & \ddots & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 1 & 0 \\ \vdots & & & \ddots & \ddots & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Equations 10 are equivalent to

$$\begin{aligned} \mathbf{A}\underline{x}_t + \underline{f}_x(\underline{x}_{t-1}, \underline{y}_{t-1}) &= -\gamma(\underline{x}_t - \underline{x}_{t-1}) \\ \mathbf{A}\underline{y}_t + \underline{f}_y(\underline{x}_{t-1}, \underline{y}_{t-1}) &= -\gamma(\underline{y}_t - \underline{y}_{t-1}) \end{aligned} \tag{12}$$

where γ is the temporal step size. Note: although this is the step size used by Kass *et al.*, (1988), damping is defined differently in Terzopoulos & Fleischer (1989).

These equations can be solved by matrix inversion using LDU (Lower Diagonal Upper) decomposition:

$$\begin{aligned} \underline{x}_t &= (\mathbf{A} + \gamma\mathbf{I})^{-1}(\gamma\underline{x}_{t-1} - \underline{f}_x(\underline{x}_{t-1}, \underline{y}_{t-1})) \\ \underline{y}_t &= (\mathbf{A} + \gamma\mathbf{I})^{-1}(\gamma\underline{y}_{t-1} - \underline{f}_y(\underline{x}_{t-1}, \underline{y}_{t-1})) \end{aligned} \tag{13}$$

In their work about deformable models, Terzopoulos *et al.* (1989) present a different model, using more physical characteristics, and based on the Lagrange Equation. These models use an internal energy which is a function of the deformation of the model from its original shape. In our model, the equilibrium shape (i.e. the shape corresponding to a null internal energy) is a straight line. We give details about this model in Chapter 6, because it could be usefully applied to our particular problem.

3.4 Forces Added To The Basic Model

We saw in the previous section how the internal and image forces are computed. Two additional components were added to the basic model: interactively controlled springs attached to the snake, and viscoelasticity, which allows the snake's compression behavior to be controlled.

When a spring is attached to a node of the snake, the external force exerted on that node is a weighted combination of the image force on the point of the image where the node is and the spring force. Spring force is controlled by the length and angle of the spring relative to the node.

For each cycle of computation, a simple metric force is applied to each node of the snake. This force attempts to maintain the distance between successive nodes defined by the set of arc-lengths stored in memory. If the viscoelasticity parameter is turned off, then only the original arc-lengths, computed at the creation of the snake, are used. However, the snake's viscoelastic properties can be controlled by setting the rate (number of computation cycles) at which arc-length is recomputed.

When a high rate is used, e.g., every cycle, internode distances will change very rapidly. Low rates will change more slowly, since the arc-lengths used by the internode distance constraint are not recomputed as often.

4. SOFTWARE INTERFACE

4.1 Overview

In this section, the software interface for running the Snake code on the Apple Macintosh is described. The interface allows manipulation of individual images and image sequences, setting of various parameters, defining the shape for the snake, and extraction of measurement data for further analysis.

4.2 Loading Images And Setting Forces

Image files must be in PICT format to be loaded by the program. The command **Open Image File** opens a PICT image, and draws it to the screen.

The external forces are computed from the image file and the first step is the averaging (smoothing) of the image. So, before opening the image file, the **Number of Averaging Steps** parameter must be set in the **Control/Config** menu. The more the image is averaged, the larger the force field, i.e. the snake is attracted to edges from larger distances. On the other hand, greater averaging leads to less precise fits. The choice of precision versus distance depends on the type and quality of image. For instance, in the examples in Section 5, the averaging parameter was set to 5 for the profile image, because precision was needed for the lips. But for the X-ray images, the parameter was set to 30 because the snake had to make fairly large jumps from one image in the sequence to the other.

Open & Compute opens an image file and computes the image forces. Computation of the forces is time consuming, therefore, it is advisable to crop the image to just that part containing the area of interest. This is done using **Define a Zone** and entails clicking with the mouse on two points in the image to define the opposite corners of a rectangle within which the forces will be computed. Alternatively, selecting **All Image** will compute forces for the entire image.

Another way to avoid excessive computation, especially if the same image will be analyzed more than once, is to save the image forces as a PICT image, using **Save Forces**. Also, batch computation of forces can be done by using **Open and Compute** to select multiple images for computation, and then let the program compute and save the forces automatically. The forces images are saved with the extension *forces* appended to the image file name (and later opened using **Open Forces File**). When **Open & Compute** is selected, a window appears with three buttons, **Choose File, OK** and **Cancel.** At present, multiple files are selected singly by using **Choose File** as many times as necessary. When all the desired files are selected, clicking on **OK** starts the computation. This procedure will be simplified at a later date.

Save Image and Snake save the original image, drawing the actual position of the snake on it. This is useful to print results.

4.3 Defining And Running The Snake

4.3.1 Parameter configuration — All snake parameters must be defined using the **Control/Config** menu before drawing a new snake or attaching springs. Guidelines

for configuring these parameters are given in the appendix. Use of the first parameter, **Number Averaging Steps**, was described in Section 4.2.

Strength Length, **Strength Image**, **Damping** and **Spring Strength** are parameters used in computing the total force exerted on the snake. They affect, respectively, the attraction between nodes, image force, damping and spring strength.

The parameter, **Viscoelastic Time**, controls the rate at which the distance between nodes (the set of interpolated points) of the snake are recomputed. Lower settings result in greater compressibility of the snake. A useful range is between 1 and 100. Viscoelasticity is turned off by setting the parameter to zero.

Parameters ω_0, ω_1 and ω_2 define the stiffness of the snake. In this version of the snake algorithm, ω_0 is not implemented, so it should be set to zero. The values of parameters ω_1 and ω_2 should be small, 0-1. They determine whether the snake behaves more as a membranous or thin-plate spline. Of the two parameters, ω_2 is more important; the bigger this parameter is, the more the snake "wants" to be a straight line.

The **Threshold for Snake Motion** parameter controls when the snake ceases computation. The threshold is the minimum number of pixels, by coordinate (x or y) and by node of the snake, of motion allowed to the snake. If the average motion for all the points is lower than this parameter value, then the snake has stopped.

There is a button for choosing between open (default) or closed snakes. The closed option, which creates a circular snake is useful for fitting whole objects, e.g., lip aperture in the frontal image plane.

4.3.2 Snake definition — The **Define Snake** button causes the active image to be displayed in another window. The snake is defined by pointing and clicking the mouse cursor on key points near the edge of interest. Line segments are drawn between the chosen points. Once the snake is defined, the program interpolates the curve by adding the previously determined number of nodes between each pair of chosen points. Remember: interpolation is controlled by setting the **Number of Nodes Between Two Points** parameter of the **Configuration** command. The selected value is important, because it affects snake stiffness. As the number, or density, of points is increased, the snake becomes less stiff. For good fitting of the lip profile then, a value of almost 200 points was used, but only 60 points were needed for the tongue, which is stiffer than the lips.

Springs can be added to exert external control on the snake using the **Add Springs** button. The image is redrawn with the snake. To define a spring, click first on a point of the snake, which will be pulled by the spring, and a second point on the image to define the other end. Length of the spring and its angle relative to the snake determine spring stiffness and direction of pull, respectively. The **Remove Springs** command removes all defined springs.

Both the chosen parameters and the snake can be saved in a file using the **Control/SaveConfig/To a File** command. These values can also save as defaults using **Control/Save/As Default**; they will be loaded automatically every time the program is run.

4.3.3 Running the snake — After defining a snake, use **Control/Continue** to begin the fit. If **Configuration/Pause** is on, the snake will make just a step. If not, it will run until it reaches the pre-set value of equilibrium (**Threshold for Snake Motion**

parameter in **Control/Config**). Clicking the mouse anywhere on the image will also stop the program.

4.4 Measurements

In order to extract parameter values for further analysis, the software is being modified to record snake coefficients, useful for estimating the dynamic characteristics of tongue behavior, as well as coordinate values of any structure of interest. At present, the command **Measurements/Positions** only displays the image and the snake in a new window, and allows the x-y coordinates of any point in the window to be determined by clicking the mouse.

5 RESULTS

The snake-fitting algorithm was evaluated using two kinds of images: video images showing the profile of the lower face (lips and jaw), and sequences of x-ray ciné. In this section, we demonstrate how the snake can be adapted to specific types of images by parameter tuning.

5.1 Video Images Of The Profile

As can be seen in Figure 2, the most interesting aspect of fitting the face profile with an energy minimization function is to capture adequately the sharp corner at which the two lips meet (Fig. 2a). Since the equilibrium shape of the snake is a straight line and in order for it to remain supple, the snake has to be defined with a small value for w_2, and must be composed of a large number of nodes. In Section 4.2, we describe the relation between the number of averaging steps applied to the image and the precision of the fit. For this kind of image, the number of averaging steps has to be small in order to achieve a precise fit, especially at the corner between the lips. Limited smoothing results in fairly sharp edges in the first derivative (Fig. 2b) and a narrow force field around the edge (Fig. 2c). Therefore, the snake has to be initialized very close to the edge (Fig. 2d). Figures 2e-g show intermediate steps of the fitting, Figure 2h shows the final result.

a)  .....b)  .....c) 

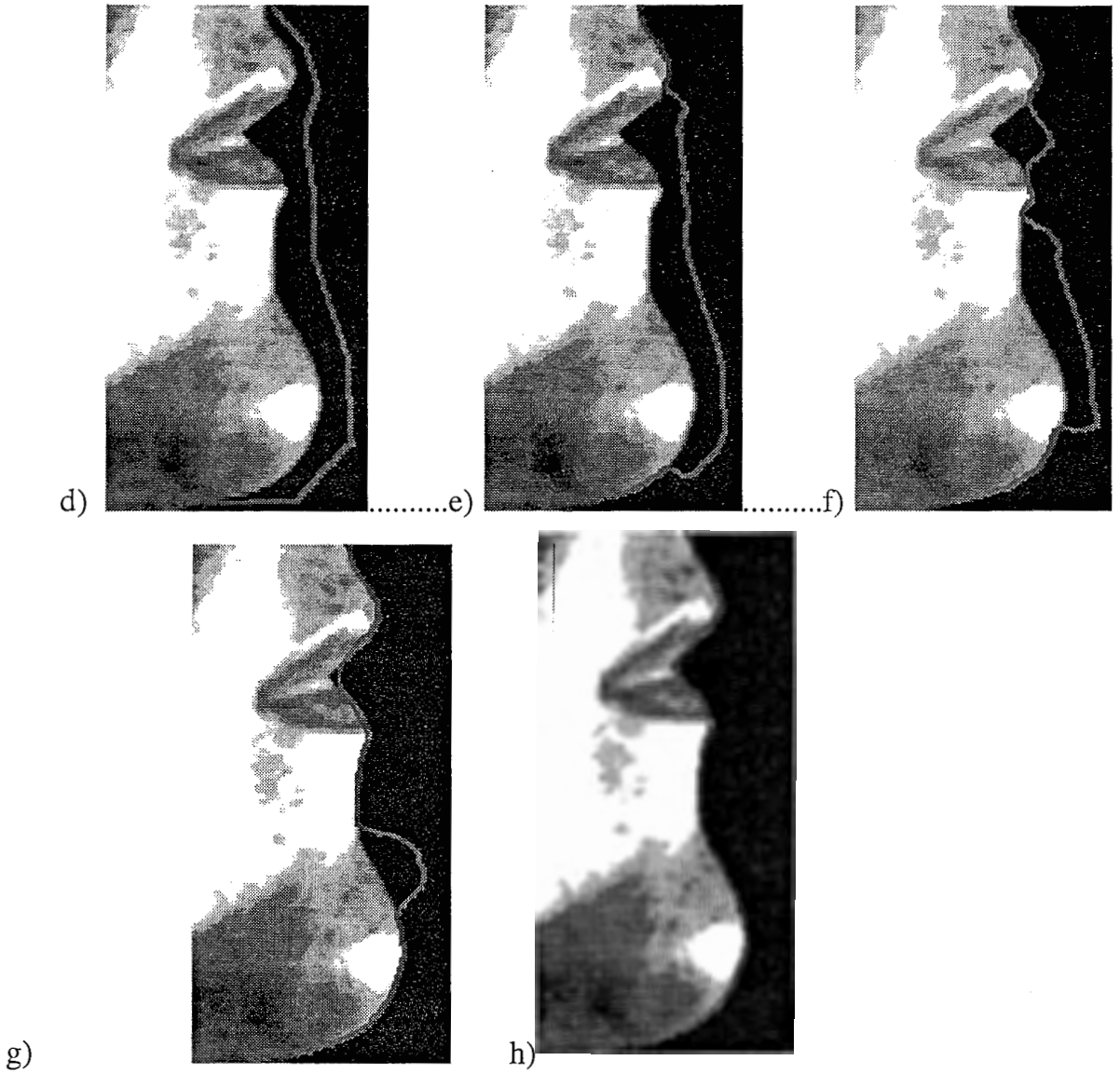


Figure 2. Various steps in the processing sequence of fitting a snake to a video image of the lip-chin profile. Shown are the original image (a), the first (b) and second (c) derivatives, snake initialization (d), several processing iterations (e-g), and the final fit (h).

5.2 X-Ray Images

A sample X-ray image is shown in Figure 3. Two complicating characteristics of such X-ray images should be noted: they are very noisy; and they represent in two dimensions the superposition of transparent (e.g., tongue) and more opaque (e.g., teeth, jaw) 3D structures.

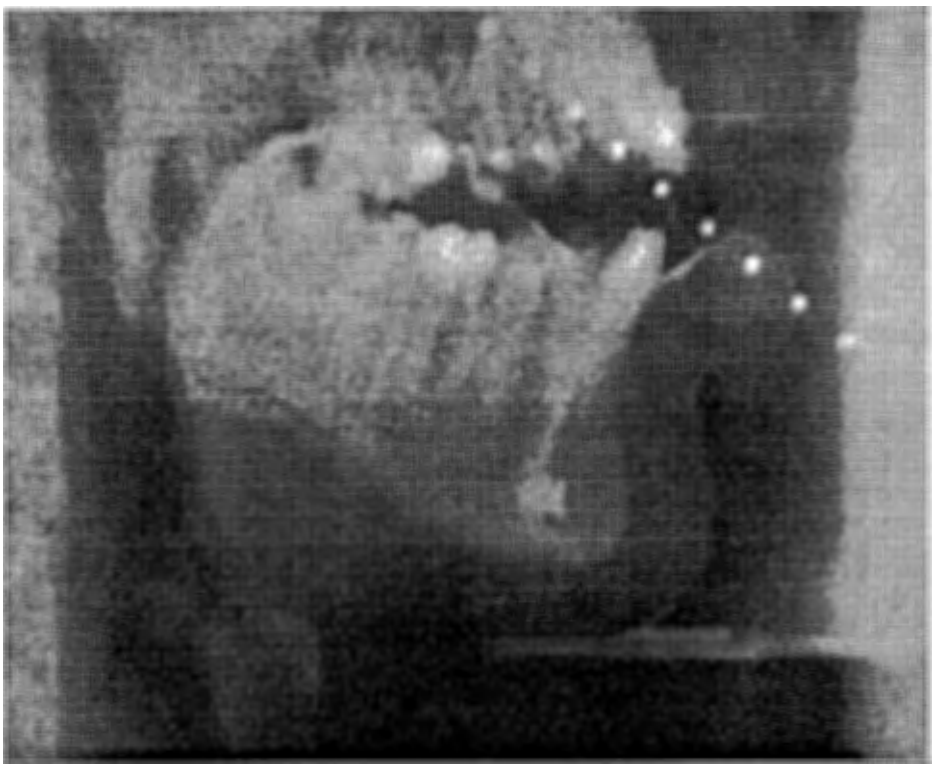
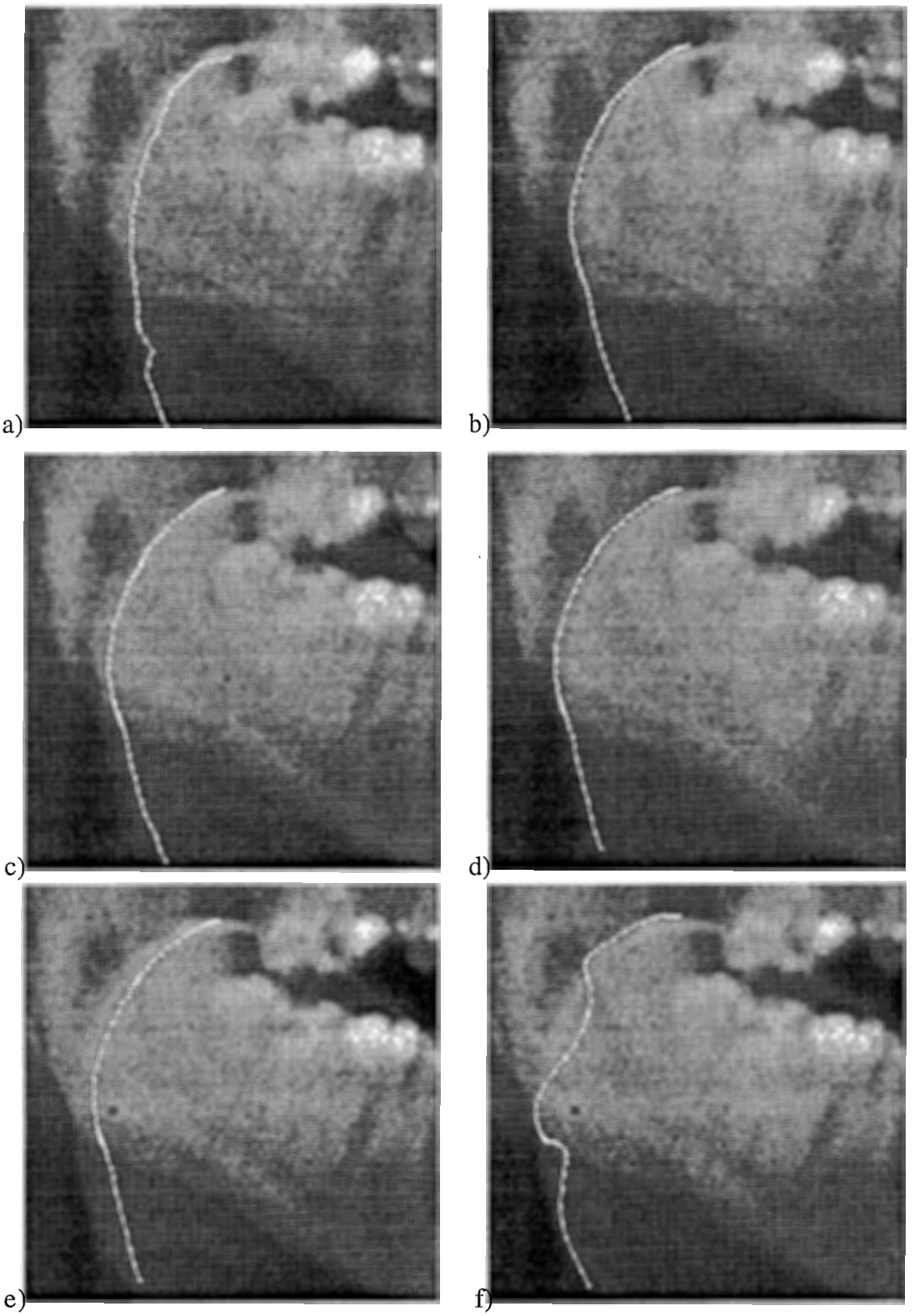


Figure 3. Midsagittal X-ray image of the vocal tract during vowel production shows clearly the lips (right), jaw and teeth, dorsal tongue surface, and pharyngeal cavity. Less clear are the superior borders of the oral cavity and the anterior tongue surface. Physical dimensions can be recovered from the string of radio-opaque beads, spaced 1 cm apart.

Figure 4 shows how the snake can follow the tongue surface through a sequence of poor quality images. In order to fit the snake to the tongue surface in these images, the value of the stiffness parameter must be quite high. Also, the images have to be smoothed with a large Gaussian variance to remove the spurious edges caused by the noise. This gives a large force field, which incidentally allows the snake to track the tongue surface from one image in the sequence to the next, even when the motion of the tongue is quite rapid. Figure 4a shows the first image where the snake is defined and initialized fairly close to the tongue surface. Running the snake gives the fit shown in Figure 4b. As shown in Figure 4c, the final fit for one image is used to initialize the snake for the next image. The resulting fit for the second image is shown in Figure 4d. This process is then repeated for the remaining images of the sequence. Figure 4(e-g) shows the initialization of the snake, an intermediate stage of computation, and the final fit for the third image. Figure 3(h-j) shows the same thing for the fourth image. The jump between the third and the fourth images is larger than the others and demonstrates the ability of the snake to find the moving tongue surface from one frame to the next.


a)
b)
c)
d)
e)
f)

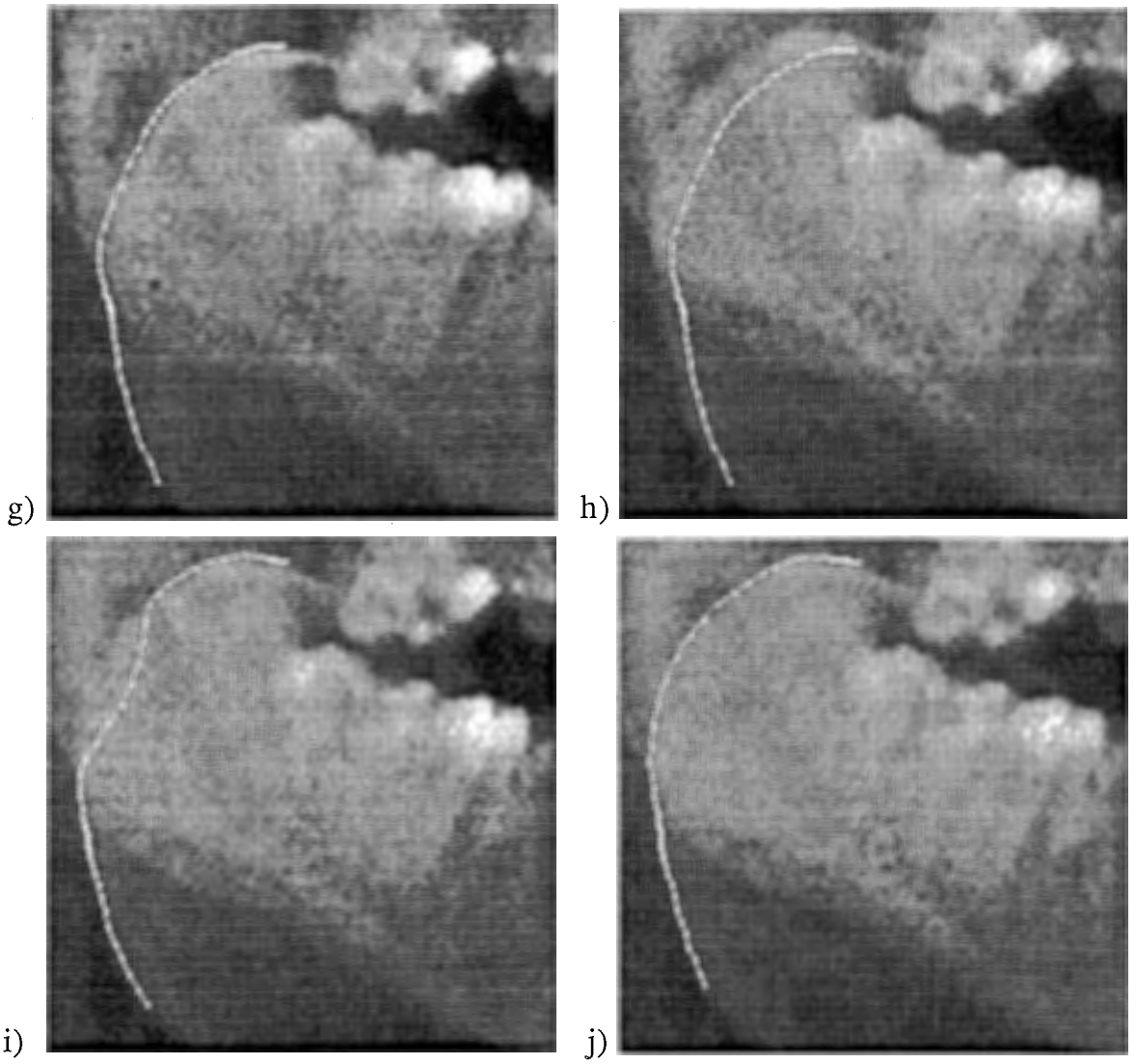


Figure 4. Steps in fitting the snake to tongue surfaces in a sequence of four X-ray ciné images. Shown are initialization and final fits of the snake for the first image (a-b); snake initialization of the second image using the final fit of the first image (c); the final fit for the second image (d); initial, intermediate, and final snake fits for the third (e-g) and fourth (h-j) images.

6. IMPROVEMENTS

In this section, we discuss several problems inherent either to the implementation of the snake algorithm or to X-ray images in particular, along with improvements we expect to implement in the future.

6.1 Tracking The Entire Tongue Surface

As shown in Section 5, the snake can track quite well the posterior tongue surface through sequences of X-Ray images. However, it cannot track the entire surface of the tongue, because of the superimposition of opaque structures such as the teeth and dental fillings onto the more transparent tongue. Since the intensity of the teeth in the image is much stronger than that of the tongue surface, the image forces for the tongue are largely canceled out by the energy outlining the edge of the teeth. Thus, in the current implementation, the snake is strongly attracted to the teeth in that part of the image. In this section, two improvements are discussed that could allow the snake to detect the tongue through the teeth. The two

methods entail modifying the snake's internal structure to give it a behavior closer to the real behavior of the tongue.

The tongue is a very elastic structure that moves and deforms a lot during speech. However, the deformation is not uniform but varies for different regions of the tongue. For example, the root (pharyngeal) of the tongue does not deform nearly as much as the dorsal or even the anterior portions. It might be useful to incorporate the same characteristics into the snake, and could be done defining region-specific stiffness parameters as functions of the intrinsic coordinates of the snake.

To this end, Terzopoulos & Fleischer (1988) present a deformable, "hybrid model" whose potential energy is a function of the difference between its actual shape and its original shape, i.e. the snake tries to restore its original shape. Being able to give the snake memory of its initial shape could prevent it from assuming unrealistic shapes, as happens when it erroneously tracks the teeth. Terzopoulos & Fleischer's model is based on a decomposition of the snake's node positions into two components, a rigid component and a deformation component. Since the internal energy is a function of the snake's deformation from its original shape, this model allows translation and rotation of the rigid component.

To achieve more realistic behavior, they assign a mass to the model whose physical behavior is represented by the Lagrange equation of motion. This leads to the equations :

$$\mathbf{M}\frac{\partial^2 \underline{x}}{\partial t^2} + \mathbf{C}\frac{\partial \underline{x}}{\partial t} + \mathbf{A}\underline{e}_x = \underline{f}_x$$

$$\mathbf{M}\frac{\partial^2 \underline{y}}{\partial t^2} + \mathbf{C}\frac{\partial \underline{y}}{\partial t} + \mathbf{A}\underline{e}_y = \underline{f}_y$$

where **M** and **C** are diagonal square matrices with the coefficients μ (the mass density which controls the inertial force) on the diagonal of **M** and γ (the damping density which controls the velocity damping) on the diagonal of **C**. $\underline{f}_x$ and $\underline{f}_y$ are N-dimensional vectors of the projection on the x and y axes of the external force for each node of the snake. Finally,

$$\mathbf{A} = (\omega_0 + \frac{2\omega_1}{h^2} + \frac{6\omega_2}{h^4})\mathbf{I_N} + (\frac{-\omega_1}{h^2} + \frac{-4\omega_2}{h^4})\mathbf{J_N} + \frac{\omega_2}{h^4}\mathbf{L_N},$$

where I_N, J_N and L_N are the matrixes defined in Section 3.3, ω_0, ω_1 and ω_2 are the stiffness parameters, and h is the distance between nodes. $\underline{e}_x$ and $\underline{e}_y$ are the x- and y-axis projections of the deformation component coordinates. With the rigid component defined at the center of mass of the nodes, the deformation component is the difference between the coordinates of the snake in a space oriented to the snake's attained center of mass and the coordinates of the initial snake's original center of mass.

6.2 Improving The Interface

Immediate improvements to the interface should be of two types. First, the program needs to be easier to use and more flexible. For example, at present, the user can choose a sequence of images, and then compute and save the different

forces files automatically; however, the user has no control over the naming and destination of the resultant force files (the program just adds the extension "forces" at the end of the image file name). Second, and perhaps more important to achieving the goal of using snakes to extract useful vocal tract parameters, is the ability to extract parameter values and physical measures from the snake. The program currently allows the user to read out pixel coordinates to the screen. In the future, it will be necessary to send such values to text files for subsequent analysis as well as to obtain estimates of curvature for local regions as well as the entire snake. Such measures could then be used to estimate the time-varying characteristics of the articulator structures being examined.

ACKNOWLEDGMENT

This work was completed while the first author was a student intern (ENST, Paris) at ATR Human Information Processing Laboratories. We are grateful to Dimitri Terzopoulos for kindly providing the snake code and for answering questions concerning its implementation.

BIBLIOGRAPHY

Baer, T., Alfonso, P.J. & Honda, K. (1988). Electromyography of the tongue muscles during vowels in /əpVp/ environment. *Annual Bulletin of R.I.L.P., University of Tokyo*, **7**, 7-18.

Begin, C. (1993). Physical models for edge finding : Snakes", *ATR Technical Report*, **TR-H-044**.

Benoit, C., Lallouache, Mohamadi, T., & Abry, C. (1992). A set of French *visemes* for visual speech synthesis. In Bailly, G. & Benoit, C. (Eds), *Talking machines: Theories, models, and designs*, pp: 485-504. Amsterdam: North-Holland.

Cohen, L.D. (1991). On active contour models and balloons. *Computer Vision, Graphics and Image Understanding*, **53**, 211-218.

Dart, S.N. (1987). A bibliography of X-ray studies of speech. *UCLA Working Papers in Phonetics*, **66**, 1-97.

Ekman, P. & Friesen, W. (1975). *Unmasking the human face.* Prentice Hall Inc. for Ekman (1982)

Fujimura, O. (1961) Bilabial stop and nasal consonants: A motion picture study and its acoustical implications. *Journal of Speech and Hearing Research*, **4**, 233-247.

Hirayama, M., Vatikiotis-Bateson, E., & Kawato, M. (1993). Physiologically based speech synthesis using neural networks. *IEICE Transactions*, **E76-A**, 1898-1910.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes : Active contour models. *International Journal of Computer Vision*, **3**,321-331.

Maeda, S. (1992). From EMG to sound patterns of vowels : Software. *ATR Technical Report*, **TR-H-005**.

Munhall, K.G., Vatikiotis-Bateson, E. & Tohkura, Y. (1994). X-ray film database for speech research. *ATR Technical Report*, **TR-H-116** (in press, *Journal of the Acoustical Society of America*).

Pavlidis, T. (1982). *Algorithms for graphics and image processing.* Computer Science Press.

Pentland, A. & Horowitz, B. (1991). Recovery of nonrigid motion and structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , **13**, 730-742.

Perkell, J.S. (1969). *Pysiology of speech production.* Cambridge, MA: MIT Press.

Ronfard, R. (1994). Region-based strategies for active contour models., *International Journal of Computer Vision,*, **13**, 229-251.

Rose, R.C., Schroeter, J., Sondhi, M.M. & Ghitza, O. (1994). Speech production models in automatic speech recognition — Forming a lasting marriage between speech science and speech technology, *Journal of the Acoustical Society of America*, **95**, 2848.

Rubin, P., & Vatikiotis-Bateson, E. (in press). Measuring and modeling speech production in humans. In S. L. Hopp & C. S. Evans (Eds.), *Animal Acoustic Communication: Recent Technical Advances* . Heidelberg: Springer-Verlag.

Staib, L. & Duncan, J. (1989). Parametrically deformable contour models. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, (pp. 98-103). San Diego, Ca.

Stone, M. (1990). A three-dimensional model of tongue movement based on ultrasound and x-ray microbeam data. *Journal of the Acoustical Society of America*, **87**, 2207-2217.

Strang, G. (1986). *Introduction to applied mathematics.* Wellesley-Cambridge Press.

Toelg, S. & Poggio, T. (1994). Towards an example-based image compression architecture for video-teleconferencing, *A.I. Memo (MIT Center for Biological and Computational Learning)*, 1494, 1-38.

Terzopoulos, D., Witkin, A. & Kass, M. (1987). Symmetry-seeking models and 3D object reconstruction. *International Journal of Computer Vision*, **1**, 211-221.

Terzopoulos, D. & Fleischer, K. (1988). Deformable models. *The Visual Computer*, **4**, 306-331.

Terzopoulos, D. & Witkin, A. (1988). Physically-based models with rigid and deformable components. *Graphics-Interface '88*, 146-154.

Tiede, M.K. & Vatikiotis-Bateson, E. (1994). Extracting movement parameters from a videodisc-based cineradiographic database",

Waters, K. & Terzopoulos, D. (1992). The computer synthesis of expressive faces. *Philosophical Transactions of the Royal Society of London B*, **335**, 87-93.

Williams, D.J. & Shah, M. (1992). A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics and Image Understanding*, **55**, 14-26.

Yuille, A., Cohen, D., & Hallinan, P. (1989). Feature extraction from faces using deformable templates. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (pp. 104-109). San Diego, CA.

APPENDIX

How To Use The Program 

This is a more detailed explanation of how to use the program in which the different steps and available options are described.

1) Image :

Use **File/Open Image File** and choose an input image file. This image must be in **PICT** format.

2) Forces :

Use **File/Open Force File**, to open a force file you saved previously, or choose the Number of Averaging Steps in **Control/Configure.**

Use **File/Open & Compute/Choose**, choose the file (or many files), and click on **OK** when finished. Choose **All Image** to have forces computed on the entire image, or **Define a Zone.** With the last command, a window appears, displaying the image : click on two points of the image to define a rectangle, then on the corner of the window to close it.

3) Parameters:

There are 3 ways to set the parameters :

- Use the default values, if some have been saved.
- Open a configuration file : **Control/Open Config File**
- Set the parameters by hand : **Control/Configure**

Example set of values for the parameters (used for the tongue) :

Num Averaging Steps	5
Strength Length Force	0.06
Strength Image Force	0.055
Damping	0.5
Viscoelastic Time	20
Spring Strength	0.8
Number of Nodes between 2 points	20
Threshold for snake motion	0.05
ω_0:	0
ω_1:	-0.01
ω_2:	100
Snake closed	No

4) Display:

By default, the image is displayed with the snake as a curve. Click on **Control/Show Forces** to display the first derivative. Click on **Control/Show**

Points of Snakes to display the snake as a set of nodes (note that the program runs slower in this mode).

5) **Snake:**

The snake can be defined in three ways: using the default configuration, if there is one, using a saved configuration, or it can be configured by hand. Note: Make sure the parameter values are set before defining the snake; an error will occur if the snake is defined with all the parameters set to zero. Define the snake with **Control/Configure/Define Snake**. Click on points in the image to set the snake position. The program will linearly interpolate the curve, adding the number of nodes between user specified snake nodes specified by **Number of nodes between 2 points**. Click on the window-closing button (upper left corner) to close the window, and then click on **OK** to confirm your snake choice (or **cancel** to start again).

6) **Springs:**

Snake behavior can be controlled locally by adding springs. For this, use **Control/Configure/AddSprings**. For each spring to be defined, click on two points. The first point chosen must be on the snake; the second point is chosen bearing in mind that the distance and angle relative to the surface of the snake through the first point determines the strength and direction of the force exerted on the snake.

7) **Run:**

Use **Control/Continue** to run the snake. If **Control/Pause** is set it will execute just one step. If not, the program will run until either the average motion is lower than the **Threshold for snake motion** or the mouse button is clicked.

8) **Measurements:**

Using **Measurements/Positions**, a window appears containing the input image and the snake. Click on any point, and its pixel coordinates are displayed at the lower left corner of the window. Click on the upper left corner to close the window.

9) **Save:**

Various stages of processing can be saved. The initialization parameter values for the snake can be stored as the default configuration (**Control/Save Config/As Default**) or as a separate configuration file. The input image along with the snake drawn in it can be saved as a PICT file, using **File/Save Image & Snake**. Also, the image forces can be saved in a PICT file, using **File/Save Forces**. Although the second derivative of the input image, which gives the force applied to the snake, is computed only for the two axes of each node of the snake, what the file force field looks like can be seen using **Control/Save 2nd Derivative**. This computes an approximation of the second derivative on the entire image, and saves it as a PICT file, which shows the magnitude but not the signs of the two component forces.