May 1994

Modeling and Animating the Human Tongue during Speech Production

Catherine Pelachaud
University of Pennsylvania

C. W. A. M. van Overveld
University of Technology, Eindhoven

Chin Seah
University of Pennsylvania

Copyright 1994 IEEE. Reprinted from *Proceedings Computer Animation '94*, May 1994, pages 40-49.
Publisher URL: http://dx.doi.org/10.1109/CA.1994.324008

Modeling and Animating the Human Tongue during Speech Production

Abstract
A geometric and kinematic model for describing the global shape and the predominant motions of the human tongue, to be applied in computer animation, is discussed. The model consists of a spatial configuration of moving points that form the vertices of a mesh of 9 3-D triangles. These triangles are interpreted as charge centres (the so-called skeleton) for a potential field, and the surface of the tongue is modelled as an equi-potential surface of this field. In turn, this surface is approximated by a triangular mesh prior to rendering. As to the motion of the skeleton, precautions are taken in order to achieve (approximate) volume conservation; the computation of the triangular mesh describing the surface of the tongue implements penetration avoidance with respect to the palate. Further, the motions of the skeleton derive from a formal speech model which also controls the motion of the lips to arrive at a visually plausible speech synchronous mouth model.

Comments
Copyright 1994 IEEE. Reprinted from *Proceedings Computer Animation '94*, May 1994, pages 40-49.
Publisher URL: http://dx.doi.org/10.1109/CA.1994.324008

Modeling and Animating the Human Tongue during Speech Production

Catherine Pelachaud

Dept of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

C.W.A.M. van Overveld

Dept of Mathematics and Computing Science
University of Technology
Eindhoven

Chin Seah

Dept of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

Abstract

A geometric and kinematic model for describing the global shape and the predominant motions of the human tongue, to be applied in computer animation, is discussed. The model consists of a spatial configuration of moving points that form the vertices of a mesh of 9 3-D triangles. These triangles are interpreted as charge centres (the so-called skeleton) for a potential field, and the surface of the tongue is modelled as an equi-potential surface of this field. In turn, this surface is approximated by a triangular mesh prior to rendering. As to the motion of the skeleton, precautions are taken in order to achieve (approximate) volume conservation; the computation of the triangular mesh describing the surface of the tongue implements penetration avoidance with respect to the palate. Further, the motions of the skeleton derive from a formal speech model which also controls the motion of the lips to arrive at a visually plausible speech synchronous mouth model.

1 Introduction

In this paper, a simple tool is discussed to model the shape of a human tongue. This tool also supports simulated tongue movements during speech production, useful in the context of computer animation. In real life, the tongue plays an important role in speech production. Some phonemic elements are not differentiated by their corresponding lip shapes; rather they are distinguished by the movement of the tongue (for example /d/, /k/...). Even though only a small portion of the tongue is visible during normal speech, taking the tongue shape into account will enhance the visual plausibility of a computer graphics facial animation system. A geometric tongue model, however, is far from trivial: indeed, the tongue is a complex and flexible organ with highly articulated and irregular motions. In most facial animation system, tongue movement is not considered, or if so it is over simplified. In most cases it is represented by a parallelepiped that can move inward, outward, upward, and downward [12], [3], [17], [14]. We propose to model the tongue based on the soft object technique of [27]. This technique assumes a so called skeleton, comprising of few geometric primitives (in our case 9 triangles) that serves as a charge distribution causing a spatial potential field. The modelled soft object is an equi-potential surface defined by this field. Modifying the skeleton will modify the equi-potential surface, i.e. the soft object. Since the skeleton has only few degrees of freedom, defining the behavior over time of the skeleton is a convenient way to define the behavior over time of the resulting complex shape. We propose an interactive tool to model the shape of the skeleton; moreover we propose an algorithm to compute the soft object which implements penetration avoidance such that the tongue stays within the palate at each frame while the volume is approximately constant.

The key problem discussed in this paper is to construct a visually plausible model of a moving tongue which is driven by speech. It turns out, however, that in order to solve this problem, techniques from some quite diverse origins had to be used (skeleton modelling; an algorithm for adaptive triangulating of a mesh; parameterizing a skeleton and skeleton inbetween; two-level penetration detection in combination with approximate volume conservation; avoiding bulging artifacts in potential surfaces by applying negatively charged primitives, and coarticulation modelling for speech). A combination of techniques of this kind may appear useful in other contexts as well; therefore this research may be seen as an exercise in combining geometric modelling and motion specification techniques rather than making a tongue model per se.

In the next section we explain our model and we discuss how the primitives for the model are built. We also summarize briefly our implementation of the soft object technique. The user is referred to [24] for more details on the construction of a triangle mesh to represent the equi-potential surface. In the subsequent section we describe our penetration avoidance algorithm. Finally we show how we compute tongue shapes during speech production. Lip shapes are also

0-8186-6240-9/94 $3.00

computed and coarticulation effects are taken into account to produce the final animation.

2 Tongue Definition

The human tongue plays a significant part in speech production. Sounds are differentiated, among other factors, by the position of the tongue related to the palate, and by the curvature and contraction of the tongue.

The tongue is a highly flexible organ. It comprises muscles, fat and connective tissue [21]. Longitudinal and transverse muscles interleave. Their contraction patterns determine the direction of the tongue deformation. The contraction of longitudinal muscle will shorten and draw back the tongue while the contraction of the other group of muscles will flatten and extend it [21]. Moreover the tongue can be bent, twisted and tensed [20].

3 Tongue Modeling

Our goal is to find a compromise between a highly flexible structure with very complex movements and a simple representation made up from few primitives, each with few parameters. In this respect, the soft object technique seems to be a promising approach. Few primitives (9 triangles) define the model. We present first how we form the skeleton for the soft object. Then we discuss a tool to help create different skeleton shapes. This tool relates the geometric parameters from the skeleton (i.e. the locations of the vertices of the 9 triangles) to what will be called shape parameters. Each shape parameter implements a meaningful shape attribute of the tongue; each shape parameter can be modified interactively. Finally we explain how the final shape of the tongue is computed from the skeleton; to this end, the soft object technique will be explained briefly.

3.1 3-D Model

In [21], Maureen Stone proposed a 3-D model of the tongue. She defined 5 segments in the coronal plane – one medial and two laterals (on each side of the median) – and 5 segments in the sagittal plane – root, posterior, dorsal, middle and anterior. Our model can be compared with Maureen Stone's 3-D model. On the one hand we want to be able to model an asymmetric tongue shape and on the other hand we want to keep the number of degrees of freedom possibly low. Therefore we retain only 3 segments in the sagittal plane and 3 segments in the coronal plane (see figure 1).

Referring to figure 1, we denote $vl[i]$ the points of the tongue skeleton. By moving points $vl[1]$ and $vl[2]$ along the median, they will represent respectively the anterior/middle and the dorsal/posterior degrees of freedom. In normal speech, the anterior and middle segments are never independent characteristics of a tongue shape simultaneously (similarly for dorsal and posterior), so these don't have to occur as independent shape parameters. Therefore, in our model the remaining segments in the coronal plane are one medial and one lateral segment. Bent, twisted, and curved shapes can still be represented with our model as well as asymmetric shapes, but the "groove" shape

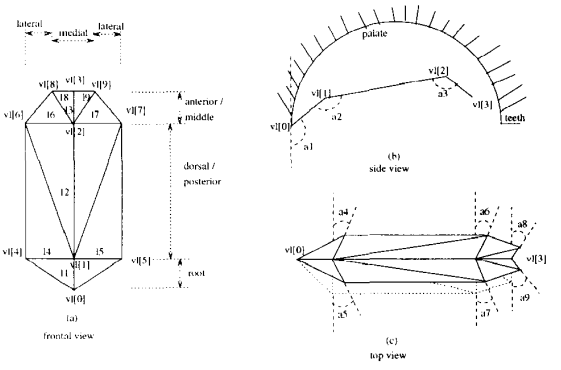

Figure 1: Skeleton of the tongue

can not be modelled independently anymore. Nevertheless, since we aim at modelling normal speech, where only a small portion of the tongue contributes to the visual appearance of the mouth, this approximation turns out to be sufficiently versatile for modelling tongue shape during speech production.

3.2 Geometric Representation

The tongue model consists of 9 triangles (see figure 1). The median is divided into three parts. The two middle points $vl[1]$ and $vl[2]$ can move along the median. A tool has been developed to modify interactively and independently each shape parameter of the model; these shape parameters are: the lengths of the edges l_i forming the median and the angles a_i between these edges. Each modification creates a new tongue shape (figure 6). By rotating the segments in the sagittal plane the tongue can be made to bend or roll. The external points $vl[i]$ can be moved by rotating the edges in the coronal plane: the tongue can be made to twist or take a U-shape.

Modifying the lengths of the edges will modify the tongue surface: the tongue can be made to compress, stretch, narrow, or flatten.

The relations between the points of the tongue skeleton (geometric parameters) and the shape parameters are (please also refer to figure 1 for the meaning of the shape parameters): vl are the points, l_i are the size of the segments and a_i are the angles):

$$
\begin{aligned}
vl[0].x &= x_0; \\
vl[0].y &= y_0; \\
vl[0].z &= z_0; \\
vl[1].x &= vl[0].x + (l_1 * \sin(a_1)); \\
vl[1].y &= vl[0].y - (l_1 * \cos(a_1)); \\
vl[1].z &= vl[0].z; \\
vl[2].x &= vl[1].x - (l_2 * \sin(a_2)); \\
vl[2].y &= vl[1].y + (l_2 * \cos(a_2)); \\
vl[2].z &= vl[1].z; \\
vl[3].x &= vl[2].x + (l_3 * \sin(a_3)); \\
vl[3].y &= vl[2].y - (l_3 * \cos(a_3));
\end{aligned}
$$

$$
\begin{aligned}
vl[3].z &= vl[2].z; \\
vl[4].z &= vl[1].z - (l_4 * \cos(a_1)); \\
vl[4].x &= vl[1].x + (l_4 * \sin(a_1)); \\
vl[4].y &= vl[1].y; \\
vl[5].x &= vl[1].x + (l_5 * \sin(a_2)); \\
vl[5].y &= vl[1].y; \\
vl[5].z &= vl[1].z + (l_5 * \cos(a_2)); \\
vl[6].x &= vl[2].x + (l_6 * \sin(a_3)); \\
vl[6].y &= vl[2].y; \\
vl[6].z &= vl[2].z - (l_6 * \cos(a_3)); \\
vl[7].x &= vl[2].x + (l_7 * \sin(a_4)); \\
vl[7].y &= vl[2].y; \\
vl[7].z &= vl[2].z + (l_7 * \cos(a_4)); \\
vl[8].x &= vl[3].x + (l_8 * \sin(a_5)); \\
vl[8].y &= vl[3].y; \\
vl[8].z &= vl[3].z - (l_8 * \cos(a_5)); \\
vl[9].x &= vl[3].x + (l_9 * \sin(a_6)); \\
vl[9].y &= vl[3].y; \\
vl[9].z &= vl[3].z + (l_9 * \cos(a_6));
\end{aligned}
$$

3.3 The Soft Object Technique

Equi-potential surfaces are a sub-class of implicit functions. Among other things, they can serve to model soft objects. Equi-potential surfaces are expensive to render directly (e.g. using ray tracing); rather, they should be converted into a triangle mesh prior to rendering. In [24], a method is proposed to convert an equi-potential surface into a triangle mesh in such a way that the triangle shapes adapt to the local curvature of the equi-potential surface: relatively flat areas give rise to large triangles whereas small triangles occur in strongly curved regions; moreover, isotropically curved surface regions translate into nearly equilateral triangles and highly anisotropically curved regions give very acute triangles.

In order to implement this, the notions of an *acceptable surface* and of *acceptable edges* are introduced. An acceptable surface is a surface where for each two points, a and b, the angle between the normals in a and b does not exceed a constant factor β times $|a-b|$. The intuition here is that a surface is only acceptable if the maximal variation of the normal vector orientation per unit distance does not exceed a given constant β. The value of β relates to the maximal curvature of the surface. For an edge ab to be acceptable, its length $|a-b|$ should not exceed a given threshold L_{max} and the angle between the normals in the points a and b should not exceed a given threshold value α. Indeed if an edge (with its end points on the surface) were long enough to allow the orientation of the normal vector of the underlying surface to make a full 90 degrees turn and back, the distance between the surface and this edge could become arbitrarily large. Given β, a safe value of L_{max} can be computed to avoid this. Similarly, the maximal angle α between the normal vectors on the surface in the extreme points of an edge allows us to put an upper-bound on the deviation between the surface and an acceptable edge. Given β, α, and L_{max} quantitative estimates for the maximal deviation of the surface and the triangular mesh approximation can be derived (see [24]).

Summarizing, the tessalation is characterized by:

- the surface is given by $f(r) = 0, r \in \Re^3$
- a cord (edge of a triangle) is a tuple (a, b, na, nb) where
$$f(a) = f(b) = 0 \quad \text{and}$$
$$na = \nabla f(a) \quad \text{and} \quad nb = \nabla f(b)$$
- the surface is called acceptable iff for every cord ab on the surface
$$\angle(na, nb) \leq \beta \mid a - b \mid$$
- a cord is called acceptable iff
$$\angle(na, nb) \leq \alpha \quad \text{and} \quad \mid a - b \mid \leq L_{max}$$
- a triangle consisting of three cords is acceptable iff all three cords are acceptable

In this case, $f(r)$ is a potential field, caused by a set of point charges. Each triangle contributes one point charge; in order to compute the potential in point r in space, the point charge is located in the point R within the triangle, closest to r. Such a point charge is represented by a tuple (R, ρ) where R is the in 3-D space and ρ is its charge. The potential due to this point charge in point r is

$$f(r) = \frac{\rho}{\mid r - R \mid}$$

If two triangles share an edge, the resulting potential is twice as high near this edge which results in unwanted bulging of the equi-potential field. This is remedied by adding line charges located at the common edges with negative contributions. In turn, this would cause over compensation near the vertices where common edges meet, so we also have to add positive point charges in the common vertices.

For all triangles, lines and vertices, the combined equi-potential surface is

$$S = \{r \in \Re^3 \mid \sum_i \frac{\rho_i}{\mid r - R_i \mid} = V_0\}$$

The algorithm discussed in detail in [24] guarantees that the vertices of the adaptive triangular mesh approximating S are on $f(r) = V_0$; that a closed surface results, and that the surface is tessalated by acceptable chords only.

To assure that vertices lay on the surface, initially the value of V_0 is set to a value close to 0. As a result, S will be very large and nearly spherical. A sphere-shaped surface is straightforwardly triangulated, and adaptiveness does not matter since the curvature is

the same everywhere. Next, V_0 is increased slightly. The surface S shrinks and may become slightly more involved. Since the vertices only have to move little, however, a linearization of the expression for $f(r)$ is sufficiently accurate to compute new locations of the vertices. The acceptability criterion is checked for all edges; if an edge is not acceptable it is split. In this manner, the value of V_0 is increased in several steps until it reaches the value for which the final shape of the equi-potential surface is defined. This stepwise approach assures that underway the vertices of the triangle mesh stay on the (shrinking) surface S and the repeated checking of the acceptability criterion guarantees that an adaptive triangulation results.

In order to achieve a closed surface, the starting polyhedron is chosen to be closed; e.g. one can take a tetrahedron which is the simplest closed triangular mesh.

4 Animation

Animating the skeleton over time, given an input text, is achieved by outputting for each phonemic item the values of all the shape parameters (the edge lengths of the median and the angles between these edges) defining the skeleton. Thus a tongue skeleton is obtained for every key-frame. Since during the construction of a soft object, the number of vertices in the triangular mesh is not necessarily the same for all shapes of the skeleton, interpolating the resulting meshes is in general not possible. Instead, the animation is obtained by interpolating between tongue skeletons; so to each frame corresponds a tongue skeleton and the soft object algorithm computes for every frame the final tongue shape.

4.1 Compressibility and Velocity

In real life, compressibility is an important feature of the tongue. The tongue does not extend or retract uniformly along its surface. Each segment can be compressed or extended independently. The pattern of compression and expansion varies over the tongue.

The middle segment has a tendency to be more compressed than the other segments. Consonants and vowels show different patterns of compression and retraction. During the production of consonants, the tongue tries to reach the palate and shows greater change of position. During the production of vowels, the tongue has the tendency to compress more in order to open the vocal tract. For vowels the degree of compressibility is mainly a function of tongue height.

Some differences between segments occur also in the timing of tongue movement. Some points arrive earlier followed by the other points: each segment has its own characteristic velocity [20].

Regardless of context, the vowel expansion and compression patterns vary roughly as a function of tongue height. The higher vowels, /i/ and /o/, cause the anterior segment to become compressed and retracted whereas the dorsal segment moves upward. For /a/, the middle and dorsal segments are compressed.

Finally, for vowels the tongue is more compressed than for consonants. This reflects the fact that the tongue displaces farther for the consonants to contact the palate and it retracts for the vowels to open the vocal tract.

5 Penetration Avoidance

When the tongue moves we have to check that it does not penetrate the palate and the teeth. The palate is modeled as a semi-sphere and the upper teeth are simulated by a finite strip-shaped plane (see figure 1). It is a close simplification of the real shape of a palate. Figure 5 shows a clipping view of the face with the tongue and the palate. Figure 7 shows a frontal view of the model. For the moment we do not deal with collision detection between the tongue and the lower mandible and lower teeth. They play a minor role in speech production. The penetration avoidance algorithm works in two passes. The first pass takes place at the skeleton level; the second pass takes place when the triangle mesh representing the soft object is computed. A possible case of penetration is detected more easily and faster during the first pass since here checking involves fewer points. This step assures the tongue skeleton to be within a given sphere-and-plane configuration, sufficiently small within the palate (to be called the virtual palate) to guarantee that the associated equi-potential surface does not penetrate the true palate; also, in the first pass the area of the triangles of the tongue skeleton is kept approximately constant, which implements approximate volume conservation of the tongue as a whole. The second pass corrects for possible penetrations of the resulting equi-potential surface with the true palate.

The algorithm works in essentially the same way in both passes. First, a possible penetration is checked, either with the skeleton and the virtual palate or the equi-potential surface and the true palate. If there is no penetration, the algorithm terminates. If there is a penetration, the penetrating points (of the skeleton or of the triangular mesh representing the equi-potential surface) are moved back inside the (virtual) palate. In order to preserve volume, in pass one the algorithm assures that

- the length of a segment with a penetrating extreme remains constant;
- if a segment of the sagittal plane is compressed (respectively expanded), the other segments of the coronal plane expand (respectively compress) to compensate.

5.1 First pass

Here we describe how to check for penetration of the virtual palate, modelled using a sphere and a planar strip, by the skeleton. Also, the criteria to be used for preserving volume of the tongue are discussed.

5.1.1 Penetration of a Sphere

The first pass takes place when computing the shapes of the skeleton over time. The program checks if, for each frame, each vertex of the tongue skeleton is within the virtual palate. The first check is within the

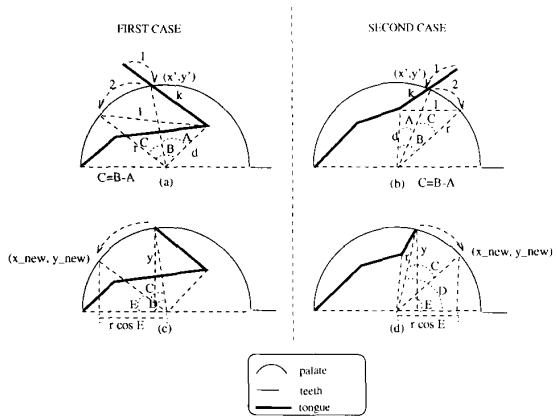

Figure 2: Collision with a sphere

sphere part of the virtual palate. The condition for penetrating the sphere is:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 > r^2,$$

where (x, y, z) are the coordinates of a vertex, (x_0, y_0, z_0) are the coordinates of the center of the sphere, and r is its radius.

If a vertex penetrates the sphere, its new position (x', y', z') is found by projecting onto the sphere:

$$x' = x_0 + ((\frac{r}{d}) * (x - x_0)),$$

where $d = |(x, y, z) - (x_0, y_0, z_0)|$ (see figure 2 (a) and (b)). Similar formulas hold for y and z coordinates.

Moving a vertex as described above would change the length l of the incident edges in the skeleton. Since these lengths are shape parameters that carry significance for the current tongue shape, however, they should be invariant. Therefore, instead of merely translating the single affected vertex, the algorithm rotates the incident segment instead. Firstly, we need to determine the angle to rotate which is equal to the angle between the vertex that moved and its neighboring vertex:

$$B = \arccos(\frac{r^2 + d^2 - l^2}{2 * r * d})$$

$$A = \arccos(\frac{r^2 + d^2 - k^2}{2 * r * d})$$

Therefore, the angle of rotation is $C = B - A$ (see figure 2 (a) and (b)).

Next, we need to either rotate left or right along the arc of the palate depending on where the vertex was before. In case of a left rotation along the arc of the palate, again two cases need to be considered (see figure 2 (c) and (d)). Two angles need to be computed:

$$D = \arcsin(\frac{y}{r}) \quad \text{and} \quad E = D - C$$

The new coordinates of the vertex will be finally:

$$x_{new} = r - (r * \cos(E)) \quad \text{and} \quad y_{new} = r * \sin(D))$$

The case of right rotation is similar. This completes the penetration check with the sphere.

5.1.2 Penetration of a Plane

If a vertex crosses the plane (for simplicity, the coordinate system is assumed to be perpendicular to this plane), the algorithm takes the following steps:

- compute the normal distance *dist* of the vertex from the plane:

$$dist = \frac{p_0 * x + p_1 * y + p_2}{\sqrt{p_0^2 + p_1^2}},$$

  where p_0, p_1, p_2 define the plane.

- next the vertex is moved perpendicularly onto the surface. This is done iteratively, i.e. by moving the point along the normal in several steps:

$$x_{new} = x - (\frac{dist * p_0}{step}),$$

  where again x_{new} is the new x coordinate and *step* is the step of the iteration. We do the same for y coordinate. At each step, the new vertex is checked if it is inside the virtual palate. If it is the algorithm terminates; otherwise it reiterates : the vertex moves along the normal and the check is done once more.

5.1.3 Expand / Contract

In order to conserve the tongue volume, the change in length in one direction should be compensated by a change in the other direction. Using the soft object technique allows producing objects from few primitives and to define their animation in an intuitive way. Nevertheless, it does not guarantee volume conservation of the equi-potential surface. For our purpose we ignore these volume changes. At the skeleton level, however, we can strive for area conservation of the triangles forming the skeleton. The penetration avoidance algorithm modifies segment lengths in the sagittal plane; so the algorithm should adjust accordingly the segment lengths in the coronal plane.

In the coronal plane the skeletal frame is expanded and contracted as follows. After detecting a penetration of the virtual palate in the sagittal plane, the algorithm compares the segment's lengths and computes the ratio of the change in length:

$$ratio_{change} = \frac{(oldlength_{sag} - newlength_{sag})}{oldlength_{sag}}$$

Then in the coronal plane, the corresponding sides of the segment are extended or contracted according to

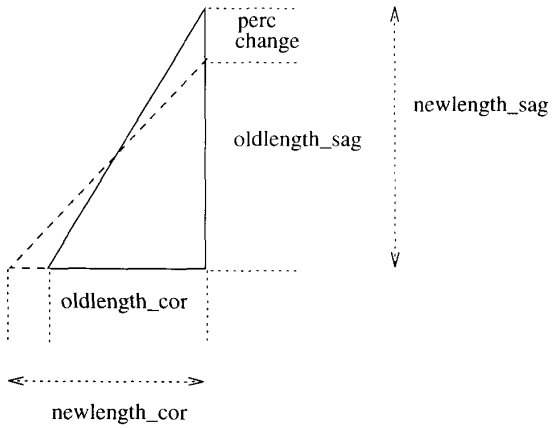

Figure 3: Expansion / Contraction of the tongue

this ratio (see figure 3)

$$newlength_{cor} = oldlength_{cor} + (ratio_{change} * oldlength_{cor}).$$

So if a segment length is shortened, the side length is increased thus (approximately) preserving the area of the associated triangle.

5.2 Second Pass

In the second pass, the penetration check is done on the final equi-potential surface. Since penetrations with the skeleton were detected already, only a few points of the triangular mesh representing the equi-potential surface are expected to penetrate the palate. These points are simply projected onto the palate.

6 Speech

Different tongue shapes differentiate phonemic elements. Consonants and vowels show different characteristics; for vowels, the entire tongue surface matters as well as its curvature while for consonants it is mainly the points of contact between the tongue and the palate or teeth that matter. For consonants, the tongue touches the palate with more tension than for vowels.

Jaw actions occur during accented vowel production. During jaw opening the tongue has greater distances to cover to reach its maxima positions. Depending on the speech rate, the tongue might not have time to reach these positions.

As it is noted in [10], there is not a universal tongue shape for each articulation, but the constraints on the tongue are such that to each articulation corresponds a particular shape which can appear in various positions in the oral cavity. The relevant issue here is the relation between the different tongue positions.

As speech rate increases the tongue does not have time to reach its extreme positions; the tongue shows less displacement, but its curvature is not affected by higher speech rates. Curvature is accentuated with loudness.

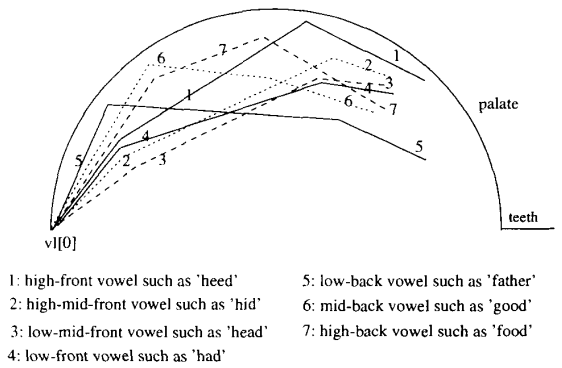

Figure 4: Tongue shape during vowel production

For slow speech-rate, steady-state tongue behavior occurs where the tongue remains still. [10] found coarticulation effects in tongue motion during speech production. To compute lip shapes, our model uses a look-ahead model with some temporal and geometric constraints [18]. Our tongue model uses also the look-ahead model to compute the tongue shape.

7 Coarticulation

Many studies have characterized tongue shape during speech production [11], [21]. Using the results of these studies and the tool we discussed in the previous sections, we specified a tongue shape for each vowel and consonant (see figure 4 (figure adapted from [11])). Even though there is no universal shape for each phonemic item, we define one tongue shape to each phonemic item for the sake of simplicity [11].

Next, speech is decomposed into a sequence of discrete units such as syllables and phonemes. The lip shape and tongue shape of any given phoneme are influenced by its predecessors and successors due to a phenomenon called *coarticulation*.

7.1 Computation of the Lip Shapes

In previous work ([18]) we implemented an algorithm computing the lip shapes. This algorithm is based on lip reading techniques. The lip shapes are defined using the Facial Action Coding System (**FACS**) developed by P. Ekman and W. Friesen. This system describes any visible facial action by the changes occurring beneath the muscular activity. An Action Unit (**AU**) corresponds to an action produced by one or more related muscles (we refer the reader to [4] for a detailed description of each **AU**.). Vowels and consonants are divided into clusters corresponding to their associated lip shapes. Such clustering depends on the speech rate. The faster a person talks, the more marginally visible segments will lose their characteristic lip shapes.

The computation of the lip shapes is done in 3 steps:

- First, we apply coarticulation rules such as forward and backward rules derived from the look-ahead model. These rules deal with the fact that

a segment may be influenced by a following or preceding vowel;

- Next, we look at temporal constraints where we consider the relaxation and contraction time of each **AU**. Indeed, we check that each **AU** has time to contract after the previous segment or relax before the next one. If not, the previous segment will be influenced by the contraction of the current segment and similarly for relaxation time.

- Finally, we look at geometric constraints by considering the surrounding phonemes. The intensity of an action is rescaled to take into account the geometric relation between successive segments. For example, when saying the word "popcorn", the 'o' of 'pop' is less open due to the 2 surrounding p's which are formed by the closure of the lips.

7.2 Computation of Tongue Shapes

We applied a similar look-ahead algorithm to compute the tongue shape. Phonemic segments show a slightly different clustering scheme for the tongue shapes in comparison with the lip shapes. The look-ahead model assures that for some highly deformable phonemic segments, the tongue shape will be influenced by the surrounding segments. If no tongue shape is associated to a particular segment, the program uses the property of the tongue which states that when a gesture is not involved in a particular segment but is in the next one, this gesture starts earlier [21]. In this case, the program starts the tongue movement on the previous segment which shows no tongue movement (e.g for /be/, the tongue associated to phoneme /e/ in not engaged in the production of /b/ and therefore starts as the same time as /b/ is pronounced [21]). To insure realism in the final animation, we compare the computed tongue shapes for given phonemes with pictures and diagrams found in the literature [13], [21].

The program outputs the different values of the tongue skeleton for each key-frame (each phonemic item correspond with a key-frame). The final tongue shapes are computed using the soft object program. An implementation of this technique was already available to us. We did not need any major work on this part (but this technique alone could not allow us to do tongue movement). The animation is displayed using *Jack*®a graphics package developed at the University of Pennsylvania. The figure 8 shows a sequence of different tongue shapes produced during speech.

8 Conclusion

We have presented a tool to model tongue movement during speech production. Tongue movement plays an important part in speech production. It helps to differentiate some phonemic elements when they can not be differentiated by their associated lip shapes. Tongue shape is very complex and flexible. The soft object technique allows one to model a complex and flexible shape; at the same time these shapes are defined by few primitives which can be easily modified to create other shapes.

Acknowledgements

We would like thank particularly Dr. Norman I. Badler for his very useful comments. We are also grateful to Dr. Mark Steedman and Scott Prevost for generating the synthesized utterances and letting us use their research.

References

[1] R.A.W. Bladon and F.J. Nolan. A video-fluorographic investigation of tip and blade alveolars in english. *Journal of Phonetics*, 5:185–193, 1977.

[2] C.P. Browman and L. Goldstein. Gestural specification using dynamically-defined articulatory structures. *Journal of Phonetics*, 18:299–320, 1990.

[3] Michael M. Cohen and Dominic W. Massaro. Modeling coarticulation in synthetic visual speech. In D. Thalmann N. Magnenat-Thalmann, editor, *Computer Animation '93*. Springer-Verlag, 1993.

[4] P. Ekman and W. Friesen. *Facial Action Coding System*. Consulting Psychologists Press, Inc., 1978.

[5] J.W. Folkins, R.N. Linville, J.D. Garrett, and C.K. Brown. Interactions in the labial musculature during speech. *Journal of speech and hearing research*, 31:253–264, 1988.

[6] G. Heike, R. Greisbach, and B.J. Kroger. Coarticulation rules in an articulatory model. *Journal of Phonetics*, 19:465–471, 1991.

[7] Eric Keller. Factors underlying tongue articulation in speech. *Journal of Speech and Hearing Research*, 30:223–229, june 1987.

[8] R.D. Kent. Some considerations in the cinefluorographic analysis of tongue movements during speech. *Phonetica*, 26:16–32, 1972.

[9] R.D. Kent. *The Production of Speech*, chapter The Segmental Organization of Speech. Springler-Verlag, 1983.

[10] R.D. Kent and K.L. Moll. Tongue body articulation during vowel and diphthong gestures. *Folia Phoniatrica*, 24, 1972.

[11] P. Ladefoged. *A course in Phonetics*. Harcourt Brace Javanovich, 1982.

[12] J.P. Lewis and F.I. Parke. Automated lip-synch and speech synthesis for character animation. *CHI + GI*, pages 143–147, 1987.

[13] B. Lindblom. *The Production of Speech*, chapter Economy of Speech Gestures. Springler-Verlag, 1983.

[14] N. Magnenat-Thalmann and D. Thalmann. The direction of synthetic actors in the film *rendez-vous à montrèal*. *IEEE Computer Graphics and Applications*, pages 9–19, December 1987.

[15] S.E.G. Ohman. Coarticulation in vcv utterances: Spectrographic measurements. *Journal of Acoustical Society of America*, 39:151–168, 1966.

[16] S.E.G. Ohman. Numerical model of coarticulation. *Journal of Acoustical Society of America*, 41(2):311–321, 1967.

[17] F.I. Parke. Control parametrization for facial animation. In N. Magnenat-Thalmann and D. Thalmann, editors, *Computer Animation '91*, pages 3–14. Springer-Verlag, 1991.

[18] C. Pelachaud, N.I. Badler, and M. Steedman. Linguistic issues in facial animation. In N. Magnenat-Thalmann and D. Thalmann, editors, *Computer Animation '91*, pages 15–30. Springer-Verlag, 1991.

[19] Elliot L. Salztman and Kevin G. Munhall. A dynamical approach to gestural patterning in speech production. *Ecological Psychology*, 1(4):333–382, 1989.

[20] M. Stone. A three-dimensional model of tongue movement based on ultrasound and x-ray microbean data. *Journal of Acoustical Society of America*, 87(5):2207–2217, 1990.

[21] M. Stone. Toward a model of three-dimensional tongue movement. *Journal of Phonetics*, 19:309–320, 1991.

[22] M. Stone, K.A. Morrish, B.C. Sonies, and T.H. Shawker. Tongue curvature: A model of shape during vowel production. *Folia Phoniatrica*, 39:302–315, 1987.

[23] M. Unser and M. Stone. Automated detection of the tongue surface in sequences of ultrasound images. *Journal of Acoustical Society of America*, 91(5):3001–3007, 1992.

[24] C.W.A.M. van Overveld and B. Wyvill. Potentials, polygons and penguins: An adaptive algorithm for triangulating and equi-potential surface. 1993.

[25] D.H. Whalen. Coarticulation is largely planned. *Journal of Phonetics*, 18:3–35, 1990.

[26] Sidney A.J. Wood. X-ray data on the temporal coordination of speech gestures. *Journal of Phonetics*, 19:281–292, 1991.

[27] G. Wyvill, C. McPheeters, and B. Wyvill. Data structures for Soft Objects. *The Visual Computer*, 2(4):227-234, April 1986.

List of Figures

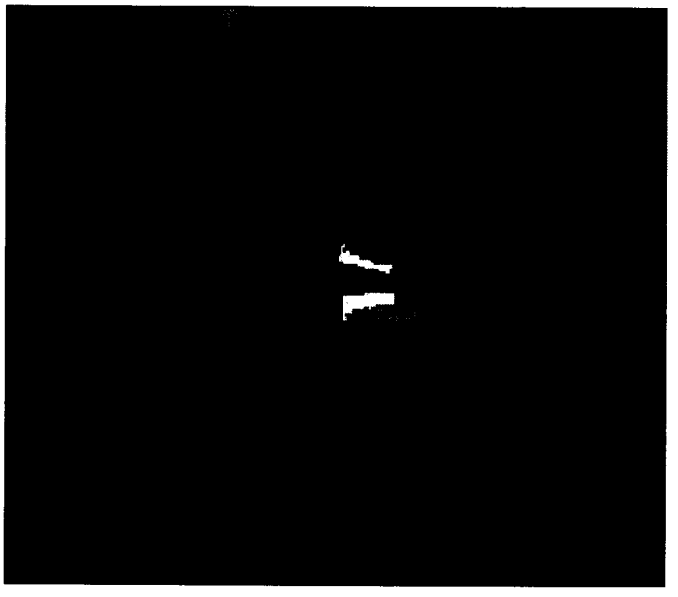

Figure 5: Clipping view of the face with the tongue and palate

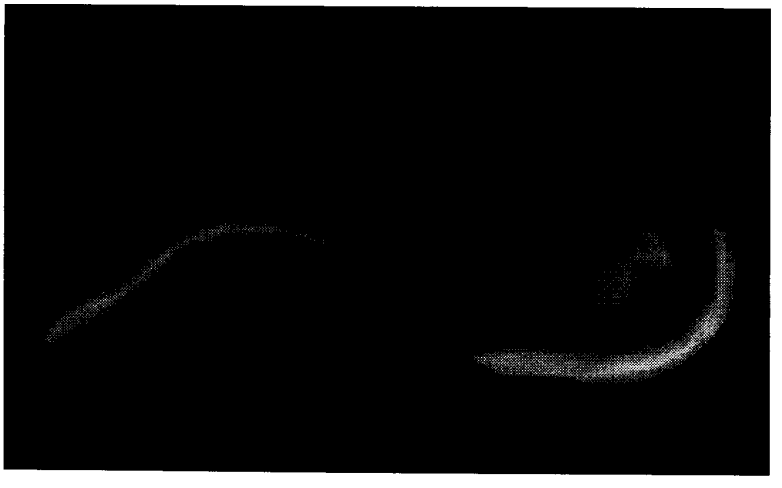

Figure 6: Two examples of tongue shapes

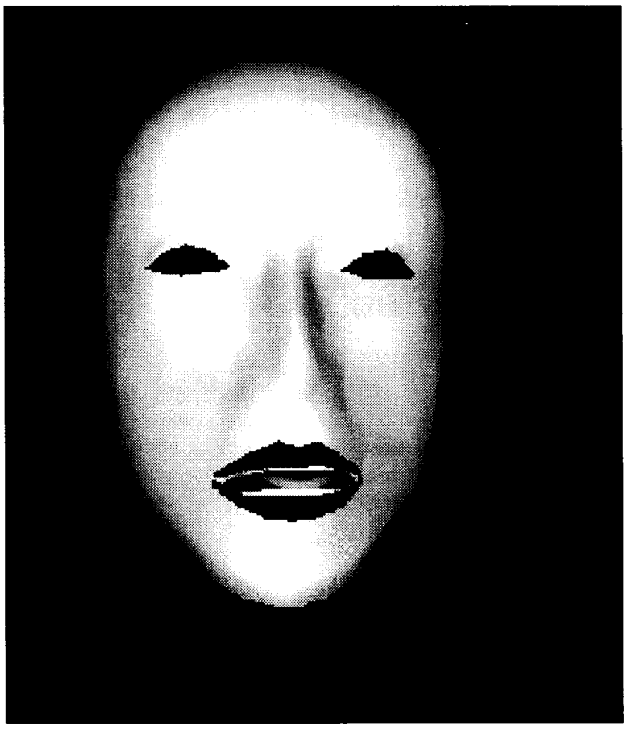

Figure 7: Frontal view

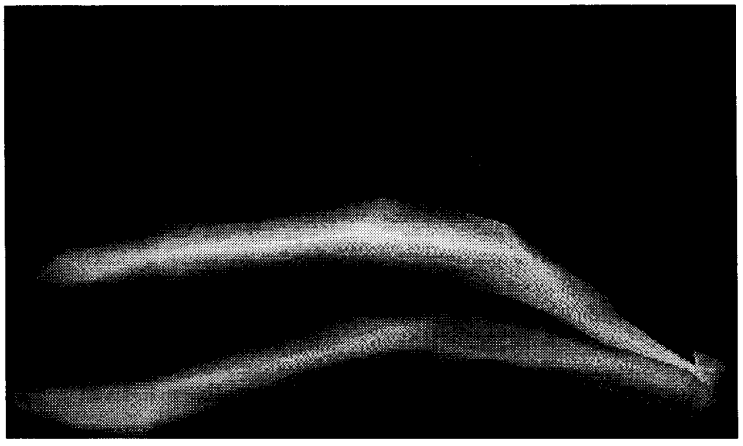

Figure 8: Examples of tongue interpolation during speech