# Real-Time Facial Animation based upon a Bank of 3D Facial Expressions

Cyriaque Kouadio†‡ Pierre Poulin† Pierre Lachapelle‡

†Département d'Informatique et de Recherche Opérationnelle
Université de Montréal

‡TAARNA Studios inc.,
Montréal

## Abstract

*The importance of faces in human interaction explains the desire for synthetic faces as a communication vehicle in computer graphics. Unfortunately, animating a face is a very complex process, partly because of our very familiarity with the human face. Moreover, depending upon the application, there exists a wide range of faces to animate, whether realistically or artistically.*

*We present here an animation system that captures live facial expressions from a performance actor, and uses them to animate in real-time a synthetic character. Our approach is based upon a bank of 3D facial expressions of a synthetic model. Critical points on the face model are matched to live markers. A linear combination of the basic expressions obtained by minimizing Euclidean distance between corresponding points and markers is then used to construct the intermediate facial expression. Mapping the motion into a bank of expressions produces subtle motions more dependent on the characteristics of the model itself. This is more difficult to achieve with previous techniques deforming a model without any additional information on how the model should move. Our technique draws mostly on the technical skills of an artist to model facial expressions, and on the skills of a performance actor to bring life to a face, with most animation issues being automated.*

*We also present improvements on the control of markers, on specific details for more delicate control of the eyelids, and on the use of filters applied to the sequence of markers displacements. More flexibility in the bank of expressions is provided by subdividing some of the expressions into independent components, and by controlling accessories such as teeth and the tongue only by means of the reconstructed expression.*

*The resulting system is very flexible, intuitive to use, and the real-time animations provide immediate feedback to express and refine animator and performer skills.*

## 1. Introduction

We are very familiar with and sensitive to human facial expressions. The fact that a human face can convey a large amount of information makes it one of the most common and favored communication channels between humans. This omnipresence of faces in our world has a parallel in many applications of computer graphics where faces are desired. Each application has its specific requirements, and although impressive results have been produced, much research is still needed to improve the modeling and control of faces [12][10].

### 1.1. Previous Work

The literature on facial animation is fairly extensive and appears in many research areas such as medicine, anatomy, psychology, computer vision, computer graphics, and many others. The reader interested in several aspects of this topic is referred to the excellent book of Parke and Waters [10].

We can divide computer facial animation into two categories: interpolation and parametrization. Interpolation is probably the most widely used technique for facial animation due to its simplicity and the full control it provides. Key-frames, or basic expressions, in 2D or 3D are located at different moments in the animated sequence. Intermediate expressions are simply interpolated between two successive basic expressions. The technique has several disadvantages. The modeling of a complete bank of basic expressions can be quite time consuming, and all these basic expressions occupy space. The intermediate expressions are also limited by the type of interpolation, and if two expressions do not blend nicely, more intermediate expressions must be added to the sequence. Furthermore, a complete bank must be built for any new facial model. Notwithstanding these limitations, this technique has proven to be quite effective. Parke showed that with a single simple topology,[1]

---

[1] In this context, a unique topology means each vertex is matched with a unique vertex in each expression, and the connectivity between the polygons remains the same between all models.

interpolation can lead to reasonable transitions. Many short animations such as *Tony de Peltrie*, *Breaking the Ice*, *Sextone for President*, and more recently *Dragon Heart* showed convincing results using these interpolation schemes.

To reduce some of the problems mentioned earlier, Parke [9] introduces the concept of parametrization in facial animation. He applies it to the face model, *conformal control*, to build a face out of all parametrized facial features, and to the animation, *expression control* independently of any specific face. The technique requires only a single basic model of a face, eliminating the need for a complete bank of models. Its animation control is also suitable for the *Facial Animation Coding System*, which denotes many of the basic facial movements. Systems derived from this concept represent a large portion of current developments in facial animation systems.

Any parametrization can be supported in these systems. One such parametrization distributes two types of muscles and their attachments on the face in order to induce complex non-linear motion for a face [15]. The addition of more physically-based models such as elastic surface layers to simulate the highly deformable skin over the muscles and skeleton provides even more realism [7], which can also be extended to produce skin deformations such as wrinkles and other aging features [17].

The key aspect in these parametrized conformal and expression controls relies upon the parametrization itself. Developing a parametrization flexible enough to create any possible face, and allowing it to take any desired expression with simple and intuitive controls, is a very complex task. Too few parameters will offer only a limited spectrum of expressions, while too many parameters will overwhelm an animator creating a specific expression on a specific face. The right balance depends upon the application, but it seems thus far no unique parametrization has proven to be sufficient [10]. Moreover, when faces must include all kinds of human, animal, and cartoon faces, it is not clear any unique parametrization will ever be possible.

Once all the models and parameters are established, an animator needs to manipulate the controls in order to animate his synthetic character. One powerful technique consists in tracking features via markers, snakes, or stereo-matching, on a performance actor and mapping this motion onto the character. It usually provides a more natural motion much faster. Williams [16] and Patterson *et al.* [11] deform a neutral model by translating a few markers displacements on an actor into control points of Coons patches on the character. Pighin *et al.* [13] also deform a basic model but using stereoscopy, thus achieving more complex motion. In both cases, the captured motion *deforms* the model with little considerations for the properties of the model such as different mouth shapes and motions between the performer and the character. In fact, it is generally difficult to map the tracked motion onto the non-linear parameters of a parametrized expression control.

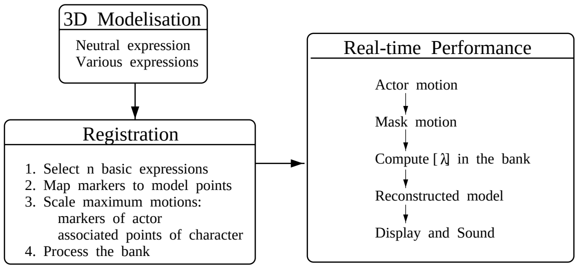

**Figure 1. The three steps in our facial animation system**

## 1.2. Our Facial Animation System

We present in this paper a facial animation system that captures the facial deformation on a performance actor and computes in real-time a new expression for a model of 2000 points as extracted weights applied to a bank of facial expressions. It draws on the skills of an artist to freely create a bank of 3D models of basic facial expressions, and then on the skills of a performance actor to breathe life into the face. The system offers several advantages. Because a real-time animated synthetic sequence results from the tracking of markers on a performance actor, it becomes possible to refine both the performance and the facial expressions, and to deliver quality animations in shorter time. Voice synchronization with performer expressions are immediately mapped in the animated sequence [16]. It allows for any facial configuration, whether cartoon, animal, or human. Moreover, because we use interpolation within a bank of expressions, the synthetic character retains some of its features through the movements. This key feature is difficult to achieve if a model is simply deformed by points tracked onto a performance actor.

Fig. 1 shows a diagram of the different steps in our system. The first step consists of creating or selecting basic facial expressions to form the bank. It is followed by a registration step that maps markers on the performance actor (in his neutral posture) to points on the neutral expression. The markers neutral configuration is projected onto the model of the character neutral expression. This projection takes into account the geometric scaling discrepancies between the character model and the actor's face. This scaled correspondence forms the link between the character and the actor, which transfers proper markers displacement. We are then ready for real-time animation. The system automatically generates by a fast least-squares algorithm any resulting expression as a weighted combination of all expressions in the model. Any recorded sequence from the performer can be mapped to any bank of any 3D facial model. The animator can create a signature by designing the neutral expression and by selecting different expressions for the bank. While good results can be obtained automatically, the animated sequence can also be refined by tuning some controls such as modifying the mapping of the markers, modifying the bounds on the weights of basic expressions, splitting the expressions into independent components, and applying filters.

In the next two sections, we present the mathematical framework on which the bank of 3D facial expressions is based, and show how the animation process is cast into a least-squares problem controlled by markers tracked on a real actor to model real-time facial animations. The results of this basic system are then discussed. The subsequent section addresses more specific issues improving the quality and control over the basic animated sequences. Finally, we conclude and suggest directions for further improvements.

## 2. The Bank of Facial Expressions

Through the years, very powerful CAD tools have been developed to produce 3D synthetic models. Artists and designers have acquired great skills with these tools to produce complex shapes. Our facial animation system primarily focuses on these skills as the basis for animation. It requires no special knowledge about muscle attachments, skin deformations, or spring constants. The artist needs only to create a neutral facial model and $n$ basic expressions derived from this neutral model. He has complete freedom to model in great detail each feature in each expression, as long as the expressions keep the same topology. To help in this task and to reduce the modeling time, a set of tools mapping several basic expressions across different facial models is provided [9] [4]. Therefore the artist does not have to create all $n$ basic expressions from scratch each time.

The facial models can be constructed by interactive techniques, or scanned. Each expression is represented as a set of 3D points. There exists a one-to-one correspondence between the points in an expression and the points in the neutral model. Therefore the number of points and their configuration remain fixed for each facial model within a single bank. Fig. 2 shows a typical bank of expressions, with its neutral facial model in the top left corner.

A 3D facial model made of $M$ 3D points is interpreted as a point in $3M$ dimensions. The neutral expression $E_0$ is the origin of the coordinate system. Any facial expression $E_i$ can be transformed as a vector in $3M$ dimensions $\xi_i = E_i - E_0$, describing the motion of the points from the neutral expression $E_0$ to the expression $E_i$. The $n$ facial expressions in a bank form the axes of an expression coordinate frame originated from the neutral expression $E_0$. The set of $n$ facial expressions spawns a subspace $\mathcal{E}$ of $\mathbf{R}^{3M}$ by choosing weights for each expression according to criteria presented later in this paper. Thus $\mathcal{E}$ defines the set of all facial expressions exactly representable in this subspace. Any facial expression $E_f$ in this subspace $\mathbf{R}(E_0; \xi_1, ..., \xi_n)$ is computed as the result from a linear combination as

$$e = \sum_{i=1}^{n} \lambda_i \xi_i \qquad (1)$$

$$E_f = E_0 + e \qquad (2)$$

where $e$ describes the displacements of the points.

The choice of a *linear* combination is fundamental in our approach because the resulting set of equations can be solved fast enough for *real-time* purpose. By assigning values to all $\lambda_i$, the motion vector for each point of the 3D model is computed in $e$, and a facial expression $E_f$ can be constructed from this motion. To create an animation, the $\lambda_i$ can be used as keys for interpolation [8][4], producing more complex and subtle facial animations than can be obtained by simply interpolating between only two basic expressions.

While some facial expressions might not be reconstructed from such a linear combination, we observed that a small bank of well-chosen expressions is often sufficient to create a wide enough spectrum of expressions. The fact that points on a face move in a more constrained way with respect to each other seem to allow the subtle facial movements to be well approximated by this simple scheme. This linear combination from a bank of facial expressions is also used commercially in systems such as the module *ShapeShifter* of *PowerAnimator* from *Alias*$|$*wavefront* and *VActor* from *SimGraphics* [3]. However neither of them describe in detail how this is achieved. As far as we know, none of them provide an automatic computation for expressions weights. The animator has to create keys by himself or use channels to control the weights. Moreover, our contribution will show in the next section how the weights $\lambda_i$ can be computed automatically in real-time from the motion of a performance actor. We will describe in Section 4 the techniques we developed to improve on the basic approach.

## 3. Controlling the Animation

The final animation is made by constructing one facial expression for each frame. A facial expression synthesis consists in a linear combination of bank expressions as stated in Eq. 1 and 2. To control the animation, we need to compute the values of $\lambda_i$. In this section, we present a mathematical formulation for the determination of an expression's coordinates. It is followed by techniques used to map the desired facial expression on the synthetic face. Finally, some features of the system are discussed through their influence on the resulting animations.

Neutral $E_0$ Jaw Open Jaw Right Jaw Left Pucker Right Pucker Left

E Sound F Sound Frown Brows Sad Brows Surprise Squeeze Eyes

Sad Pushed Lip Sneer Smile Jaw Down Kiss

**Figure 2. A typical bank of facial expressions and the neutral expression**

### 3.1. Mathematical Formulation

The movement expression $e$ is formulated in matrix form as $[\xi]\,[\lambda] = [e]$, or in its extended form as

$$\begin{bmatrix} x_{\xi_{1,1}} & \cdots & x_{\xi_{n,1}} \\ y_{\xi_{1,1}} & \cdots & y_{\xi_{n,1}} \\ z_{\xi_{1,1}} & \cdots & z_{\xi_{n,1}} \\ \vdots & \ddots & \vdots \\ x_{\xi_{1,M}} & \cdots & x_{\xi_{n,M}} \\ y_{\xi_{1,M}} & \cdots & y_{\xi_{n,M}} \\ z_{\xi_{1,M}} & \cdots & z_{\xi_{n,M}} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} x_{e_1} \\ y_{e_1} \\ z_{e_1} \\ \vdots \\ x_{e_M} \\ y_{e_M} \\ z_{e_M} \end{bmatrix} \tag{3}$$

where for example $y_{\xi_{i,j}}$ represents the $Y$-coordinate of the $j^{th}$ point ($1 \leq j \leq M$) in the $i^{th}$ motion expression ($1 \leq i \leq n$). Therefore a column of the matrix represents the movements of all points of a single basic expression. A line represents the movement of one coordinate of one point in all the expressions in the bank.

The values for $[\lambda]$ can be computed, assuming $[\xi]$ and $[e]$ are known, provided a solution to Eq. 3 exists. Because the displacement $e$ to achieve the desired expression $E_f$ might not be exactly representable in the space $\mathbf{R}(E_0; \xi_1, ..., \xi_n)$, we try instead to find the closest approximation $\hat{e}$ according to the Euclidean norm $\|\ [\xi]\,[\lambda] - [e]\ \|$. We use a very efficient and robust least-squares solver with inequalities [14], LSI, which provides the additional benefit of constraining values for all $[\lambda]$ to be within an acceptable range, if desired. Any value for a $\lambda_i$ can exaggerate some motions for $\lambda_i > 1$ or even $\lambda_i < 0$. However extreme values can have objectionable results that proper bounds simply eliminate. Casting the resolution into a least-squares problem offers the additional advantage of always providing a solution. Remember the solution of this system of equations corresponds to the *motion* of the points to reach the desired expression $E_f$. In the remaining sections of this paper, we will consider a facial expression as the motion to reach this expression. Once added to the neutral expression $E_0$, the 3D model $E_f$ is easily reconstructed as

$$E_f = E_0 + e.$$

Unfortunately a bank of 20 basic expressions, each of them comprising 2000 3D points, would require a *much more* efficient algorithm to resolve this system of equations in real-time as we would like.

### 3.2. Control Points

Instead we choose $m$ critical points out of the $M$ points of the model ($m \ll M$). These critical points are manually located at *strategic* positions such as eyebrows, lower jaw, chin, cheeks, and cheek-bones, to capture the face global movements, and at more localized features such as mouth

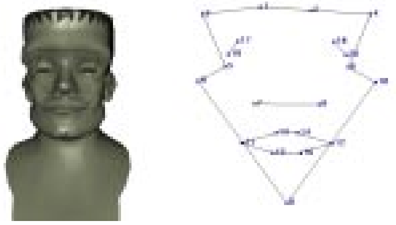

Figure 3. Manipulating the mask

deformations and eyelids. This smaller system of equations is easily solved in real-time for $m \approx 100$. The complete expression $E_f$ is then reconstructed for all its $M$ points, but with the solutions for $[\lambda]$ computed with a matrix $[\xi]$ of only $3m$ rows instead of $3M$ rows.

The set of $m$ critical points is linked into a polygonal mesh called the *mask*. This mask can be manipulated interactively to produce a complete interpolated expression over all $M$ points of the facial model, as illustrated in Fig. 3. Once again, the computed values for $[\lambda]$ can themselves be used as keys for interpolation, or the positions of the points of the mask can be used as keys to be interpolated. In the latter, new values for $[\lambda]$ are computed for every frame of the animated sequence. This mask provides a type of high level control for the animation. This general tool also offers more flexibility to control facial expressions of non-human facial structures, such as ears of animals.

### 3.3. Live Tracking

The system reveals much of its full potential when displaying an animated sequence controlled by the tracking of a real actor, all in real-time.

Many cell-animation studios such as *Disney* record voices of famous actors for their characters. Drawing artists often transfer some facial expressions of the actors during these recording sessions into expressions for their animated characters. Live tracking captures directly these facial postures, which are decomposed into the bank of basic expressions. Because of the real-time nature of the system, the actor can try several expressions and enhance his performance. A director can also run a recorded tracking session through the banks of several characters to choose the best synthetic character.

Any tracking system can be used, as long as it provides 2D or 3D positions of points on the face of a real actor. We selected a system[2] based upon a camera mounted on a helmet, tracking reflective markers glued on the face of the real actor similar to Williams [16]. It offers real-time, stable, relatively unobtrusive, and fairly accurate 2D tracking for about 30 markers. We actually never used more than 32 markers in all our animations. The left image of Fig. 4 shows the tracking apparatus. The distribution of 19 markers on the real actor, and the corresponding points on the neutral facial model, are highlighted in the middle and right images, respectively. Because of our formulation of the problem, even though we get only 2D displacements from the tracking system, the system of equations can be solved with only X and Y coordinates. Then when the entire expression is reconstructed with all its points and its three coordinates, the motion inferred by all the basic expressions is 3D. Problems might occur when most markers move only along the Z axis, which is a rare situation. A more complete 3D tracker could trivially be inserted into our resolution, thus eliminating this problem.

Figure 4. The tracking system

The first step to live tracking consists in matching each marker on the real actor to a unique point on the neutral expression of the model. Then the actor perforsm a set of extreme facial gestures. The maximum displacements reached by each marker on the actor along the X and Y axes (because our tracking system is 2D) are matched with the maximum displacements of its corresponding point in the bank. Automatic scaling factors for the synthetic character are then computed according to the extreme displacements. If necessary, the animator can choose to manually modify the associated displacements in order to attenuate or amplify certain movements.

At this point, the system is calibrated for the real actor and the particular bank of expressions. Any change in the number of markers on the actor would require performing some of the registrations described in the preceding paragraph.

### 3.4. Results

The facial animation system presented here provides real-time animation of a synthetic character automatically controlled by the facial expressions taken by a live performance actor. It allows the animator's talent to be expressed by the construction of the bank of basic expressions. The animator has also access to a few parameters in order to control the synthetic expressions. Therefore an animator can interactively set boundaries for each $\lambda_i$ or add/remove a basic expression from the bank. The animator can amplify or reduce the effect of a marker by scaling its displacements on

---

[2] *FaceTrax*$^{\textit{TM}}$ from *Adaptive Optics*

the synthetic character. By recording the marker displacements and the contribution $[\lambda]$ of basic expressions at each frame, the animator is able to control the animated sequence by finely tuning the values of $[\lambda]$. The voice of the performance actor is recorded simultaneously during the tracking session, providing synchronization of the sound with the lips motions.

The system has been used extensively in the production of the animated film "*The Boxer Trailer*" [5]. Some animated sequences designed to illustrate the techniques described in this paper can be viewed from the site associated with this paper from `www.iro.umontreal.ca/labs/infographie/papers` We used 24 markers, none on the forehead.

The 3D models and banks of expressions for the two synthetic actors were provided by Taarna Studios. An artist was simply recorded (tracked) for all the sequences, and the first author mapped the trackers with the 3D model, and recorded the computed motion for all frames. The final rendering was realized in texture mapped for better image quality. No artists were involved in improving the raw results presented in these sequences, and the entire sequences were accomplished in less than one hour. All facial animation sequences of the animated film *"The Boxer"* (to appear soon) were captured with our system, and improved by artists with the tools described in the next sections.

## 4. Improvements

The results of this basic automatic system are quite satisfying for real-time animation. In fact, with a good bank of expressions and appropriate registration, many of the animated sequences are already considered of good quality without any additional efforts.

It is important, however, to provide animators with additional tools to control and refine some aspects of the results. In this section, we discuss other components we integrated to our system, present some tuning parameters, and explain in more details some of the basic concepts.

### 4.1. Markers

The number of markers on the real face, their distribution, and their association with points on the facial model are crucial to produce high quality animations. A marker is typically associated with a single point of the facial model. For some facial models represented by fewer points, the closest point might not be accurate enough. In such situations, the marker is instead associated with the barycentric coordinates of one triangle selected from the mesh of triangles representing the facial model. Because of the unique topology between basic expressions, each triangle is guaranteed to match another one in each expression; there exists a one-to-one mapping between all the triangles in all the expressions.

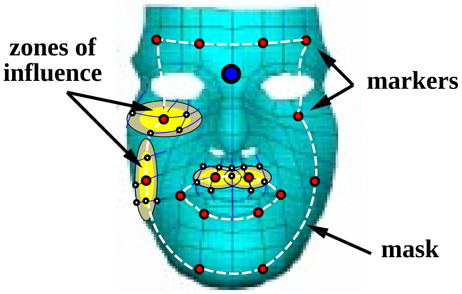

**Figure 5. Zone of influence for real markers**

Any point on the performer's face affected by detailed motions must be properly sampled to effectively control the animation of the facial model. The most important regions include the outline of the mouth, the eyebrows, and the eyelids. Fewer markers are necessary to control other regions such as the chin or the cheeks. The combined effect of all these markers offers both global and local controls. However, because of the tracking system physical limitation on the maximum number of markers, some under-sampled regions could require additional information in order to improve the approximation for the computed $[\lambda]$. This is achieved by introducing *virtual markers*. An animator can determine a zone of influence around each point in the neutral model associated to a real marker; this zone is currently bounded by an ellipsoid, although more complex boundaries could be defined. A zone can enclose many points of the facial model, and a point of the facial model can be enclosed in more than one zone, as illustrated in Fig. 5. The *influence* function is usually a continuous decreasing function, for instance a Gaussian on the positive half-line, which is related to the distance to the real marker. This notion of zone can be used to simulate simple skin deformations associated with the motion of each marker. Any point on the neutral model enclosed in a zone of influence can become a virtual marker. When real markers move, the motion of each virtual marker is computed as the weighted motion of all the real markers for which it is enclosed within their zones of influence.

### 4.2. Bank of Expressions

#### 4.2.1. Improving the Matrix

The most important control provided to an animator lies in the choice of the basic expressions to compose the bank. All interpolated expressions will be the result of a linear combination in this bank. Therefore the bank must be as complete as possible, keeping in mind that each new basic expression increases the least-squares computation and the reconstruction of the $M$ points of the model representing the desired expression.

Some observations of the matrix $[\xi]$ with respect to the control points can help detect and correct deficiencies in a bank of expressions. Rows and columns of 0's can simply be eliminated as they cannot influence the resulting expression. The initial system of equations is transformed into a system of normal equations [6], $[\xi]^T[\xi][\lambda] = [\xi]^T[e]$, to make the computational technique of constrained least-squares less dependent on the number of control points $m$, but more dependent on the number of basic expressions $n$, which is usually smaller than the number of control points. The normal equations can only be used as described above if the initial system of equations is not under-constrained, i.e. the number of equations is greater or equal to the number of unknowns. It is therefore preferable to have fewer expressions than three times the number of control points ($n \leq 3m$). If this condition is not respected, the constrained least-squares technique is used instead with the initial system of equations [6].

Another way to influence the result of the computation of $[\lambda]$ is to give different weights to markers. Particular markers can have more influence than others. The importance of a marker is controlled by the artist. It is introduced into the initial system of equations by scaling the equations corresponding to that marker, thus representing its importance. Solving the least-squares will take into account that weighting factor.

### 4.2.2. Splitting Expressions and Sub-banks

A desired expression is decomposed as coefficients $[\lambda]$ in the set of basic expressions. In this original form, each marker, and even each coordinate of each marker has the same weight as all the others, and the final contribution of the displacement of a basic expression $\xi_i$ is a single factor $\lambda_i$. This has the advantage of producing global actions. For instance, if a large smile always comes with squeezed eyes, such a basic expression will squeeze the eyes each time a large smile appears, unless another basic expression compensates with unsqueezed eyes and little impact on the smile. This suggests the quality of a bank is also function of the independence of the basic expressions.

One improvement of the bank consists in splitting one basic expression, replacing it by two new basic expressions composed of the neutral expression and its side of the split expression. A vertical split of an expression down the middle offers the opportunity of creating a wink, or a smirk, as illustrated in Fig. 6. These split expressions reduce both the modeling and the storage space. They also provide more flexibility without restraining too much the artistic creativity.

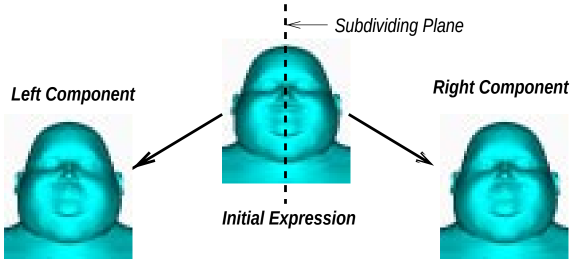

**Figure 6. Splitting a bank vertically**

While splitting is an essential element to improve upon the flexibility and the quality of the animation, it must however be used with care. Points on the borders of two split regions can move in opposite directions, and produce some visible discontinuity. For instance, splitting an *"O Sound"* expression introduces discontinuities around the mouth as an actor cannot produce a facial posture equivalent to half an *"O Sound"*. It is therefore not realistic to split such an expression. To reduce the discontinuities, points in the facial model surrounding the borders are included in both regions, and their new positions in each half expression are weighted accordingly. We observe in general fewer large and negative values for the computed $[\lambda]$ even if they are computed without bounds.

The bank of facial expressions can also be subdivided into sub-banks in order to treat some regions of the facial model independently. One typical sub-bank can cover the regions around the eyes, another can include the region around the eyebrows. A sub-bank is usually smaller than a bank of complete expressions. The smaller system of equations is solved only for the markers' displacements associated with its sub-bank. The final expression is reconstructed from all independent solutions, once again allowing for some continuity by considering points on bordering regions to be affected proportionally by the adjacent regions solutions.

## 4.3. Accessories

A facial model can be very complex, even more when it is augmented by a complete set of teeth, tongue, eyeballs, glasses, hair, etc. All of these items could be incorporated in every basic expressions, but the cost in modeling each basic expression, storing it, and using it in the reconstruction of an interpolated expression is prohibitive.

Some items such as glasses or eyeballs are not deformed by changes in a face and their corresponding standard rigid transformations are computed from the facial points they are anchored to. These items can also be animated independently, for instance via dynamics.

Other items, which we call *accessories*, can have their

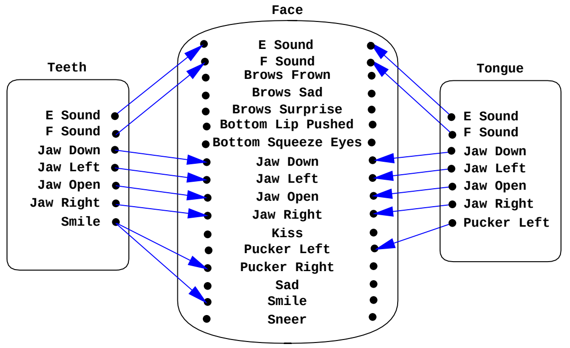

**Figure 7. Links between the basic expressions of a face, and those of two accessories**

own motion induced by some elements of some facial expressions. For instance, the tongue can take different postures depending upon the shape the mouth takes to pronounce some sounds. The animation of such accessories is also based upon a bank, but this time upon a bank of expressions for each accessory. The computed $[\lambda]_{face}$ for the desired facial expression are transformed as factors $[\lambda]_{accessory}$ in the bank of accessory expressions. Note that more than one basic facial expression can control an accessory expression, their weighted contributions then being simply summed. Fig. 7 gives an example of a dependency graph between a bank of facial expressions and two accessory banks, one for the tongue and one for the teeth. For instance, the expression *smile* in the bank of accessory teeth is weighted as $\lambda_s + \lambda_{pr}$, where $\lambda_s$ is the weight of facial expression *smile* and $\lambda_{pr}$ the weight of facial expression *pucker-right*.

We reserve a special treatment for the eyelids. The top and bottom eyelids must slide on the surface of the sphere representing the eyeball. A marker positioned on the top eyelid and another one on the bottom eyelid move mostly vertically. This restricted motion allows us to define an opening factor, which is mapped into a small (2-5) bank of eyelid expressions.

### 4.4. Filtering

The output signal of the tracking system is subject to noise that can cause slight undesirable motions in the final animation sequence. These effects can largely be eliminated by applying a low-pass filter on the marker positions for the entire tracking session.

Another filtering consideration is devoted to the sequence of computed $[\lambda]$ keys. Because we are using a least-squares solution, two consecutive expressions are computed completely independently. Although we did not witness objectionable consecutive reconstructions, some level of continuity should be expected. Once again, a filtering operation can be incorporated in the least-square solver in order to filter the sequence of computed $[\lambda]$. The wider the filter kernel, the smoother the sequence of expressions. Various filters can be used to affect the expressiveness of the sequence [2], and some controlled noise can even be re-introduced to produce controlled shaking in specific parts of a facial motion.

Finally, these two filterings can also be applied to the incomplete sequence as it is being driven by the real-time tracking system. This is carried out by the use of a predictive temporal filtering which can be applied on the markers displacements at each frame before making the expression synthesis.

## 5. Conclusion

We presented in this paper a facial animation system based upon a bank of 3D facial expressions. It exploits mainly the skills of an artist to model static 3D facial expressions, and is flexible to animate most types of synthetic characters. The expressions are transformed into a vector space and movements are induced by varying the coefficient $\lambda_i$ of each expression $\xi_i$.

To provide a more intuitive and powerful control of the animation, we introduce markers associated to points in the facial model. Minimizing the Euclidean distance between the mask and the corresponding face points positions generated from the bank (or the matrix of expressions) efficiently identifies the coefficients $[\lambda]$ then used to construct the desired expression $E_f$.

The system reveals its full potential when the mask is linked directly to points tracked on the face of a performance actor. The facial expressions of the actor are transferred in real-time into the synthetic character, producing real-time facial animation synchronized with the voice.

Several improvements are proposed to refine the resulting animations. The limited number of real markers are alleviated by the introduction of virtual markers to better control the results. The distribution of markers and their precision are also discussed. Splitting expressions of the bank improves the reconstruction of asymmetrical expressions. Subdividing a bank into independent regions reduces the number of basic expressions necessary to cover a wide range of representable expressions. Continuity is provided in both techniques by allowing points around a subdivision to belong to more than one region and be weighted accordingly.

Accessories are animated by transforming the computed coefficients in certain basic facial expressions as coefficients in their own bank of accessory expressions. This offers the possibility of sharing some of these accessory banks among several characters.

Finally, filtering on the tracked markers' displacements and on the computed coefficients $[\lambda]$ ensures some continuity in time while reducing noise. These filters can be applied on incomplete sequences. Filter shapes are also used to vary the expressiveness of an animated sequence.

The system is currently used by trained computer animators in a few production environments around the world. They rate the resulting animations from good to very good, which is quite encouraging. We expect that a longer experience with the system and a better understanding of the criteria affecting the quality of a bank of basic expressions should improve even more the resulting animated sequences. Taarna Studios inc. has filed a patent application covering this animation process.

## 6. Future Work

Although we are quite satisfied with the system, there is still room for improvement. The first need involves the precision of the tracked data. More precise 3D markers, maybe using simple stereo algorithms will provide a better mask, specially for the lips and chin, therefore refining the accuracy of the representation in the vector space. The video sequence could be better analyzed [1] to extract more precise positions into highly deformable regions such as around the lips. The motion of the iris can also be efficiently extracted from the video sequence, thus controlling the eyes' orientation during the performance session.

With each facial expression in a bank can be associated surface properties such as local skin deformation. This would provide some control to animate variations of skin blood and fat concentrations affecting its appearance as the face expression changes.

Finally, it would be interesting to try to combine the results obtained in our technique with a more elaborate parametrized facial animation model, such as muscle-based. This technique could be used to construct the basic expressions. This would also offer a better control when the animator wants to finely adjust portions of the resulting animations. Another improvement from muscle-based systems would be to spread a skin model over points on the reconstructed expression. The skin could then be relaxed to provide an even more continuous surface when necessary.

### Acknowledgement

We would like to thank all the artists and researchers at Taarna Studios who made this work so interesting and challenging. Michiel van de Panne and several anonymous reviewers provided helpful comments. We acknowledge financial support from Taarna Studios, NSERC, and the Université de Montréal.

## References

[1] A. Blake and M. Isard. 3D position, attitude and shape input using video tracking of hands and lips. In *Proceedings of SIGGRAPH '94*, Annual Conference Series, pages 185–192. July 1994.

[2] A. Bruderlin and L. Williams. Motion signal processing. In *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 97–104. Aug. 1995.

[3] M. J. Fusco and S. E. Tice. VActor animation creation systems. SIGGRAPH '93 Character Motion Systems seminar notes. Aug. 1993.

[4] K. Kardan. Tools for the animation of facial expressions using computer graphics. In *134th SMPTE Technical Conference*, pages 1–16, Nov. 1992.

[5] P. Lachapelle. The boxer, Taarna Studios Inc., 1997.

[6] C. L. Lawson and R. J. Hanson. *Solving Least Square Problems*. Prentice-Hall, 1960.

[7] Y. Lee, D. Terzopoulos, and K. Waters. Realistic face modeling for animation. In *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 55–62. Aug. 1995.

[8] M. Oka, K. Tsutsui, A. Ohba, Y. Kurauchi, and T. Tago. Real-time manipulation of texture-mapped surfaces. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, pages 181–188, July 1987.

[9] F. I. Parke. Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, 2(9):61–68, Nov. 1982.

[10] F. I. Parke and K. Waters. *Computer Facial Animation*. A.K. Peters, 1996.

[11] E. C. Patterson, P. C. Litwinowicz, and N. Greene. Facial animation by spatial mapping. In *Computer Animation '91*, pages 31–44, 1991.

[12] C. Pelachaud, N. I. Badler, and M.-L. Viaud. Final report to NSF of the standards for facial animation workshop. Technical report, Dept. of Computer and Information Science, University of Pennsylvania, 1994.

[13] F. Pighin, J. Auslander, D. Lischinski, D. H. Salesin, and R. Szeliski. Realistic facial animation using image-based 3D morphing. Technical Report UW-CSE-97-01-03, Dept. of Computer Science, University of Washington, May 1997.

[14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.)*. Cambridge University Press, 1992.

[15] K. Waters. A muscle model for animating three-dimensional facial expression. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, pages 17–24, July 1987.

[16] L. Williams. Performance-driven facial animation. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, pages 235–242, Aug. 1990.

[17] Y. Wu, N. M. Thalmann, and D. Thalmann. A dynamic wrinkle model in facial animation and skin ageing. *Journal of Visualization and Computer Animation*, 6(4):195–206, Oct. 1995.