Hindawi Publishing Corporation
International Journal of Computer Games Technology
Volume 2009, Article ID 397938, 11 pages
doi:10.1155/2009/397938

Research Article

A Facial Expression Parameterization by Elastic Surface Model

Ken Yano and Koichi Harada

Graduate School of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi Hiroshima, 739-8527 Hiroshima, Japan

Correspondence should be addressed to Ken Yano, d064016@hiroshima-u.ac.jp and
Koichi Harada, harada@mis.hiroshima-u.ac.jp

Received 28 January 2009; Accepted 11 June 2009

Recommended by Mingquan Zhou

We introduce a novel parameterization of facial expressions by using elastic surface model. The elastic surface model has been used as a deformation tool especially for nonrigid organic objects. The parameter of expressions is either retrieved from existing articulated face models or obtained indirectly by manipulating facial muscles. The obtained parameter can be applied on target face models dissimilar to the source model to create novel expressions. Due to the limited number of control points, the animation data created using the parameterization require less storage size without affecting the range of deformation it provides. The proposed method can be utilized in many ways: (1) creating a novel facial expression from scratch, (2) parameterizing existing articulation data, (3) parameterizing indirectly by muscle construction, and (4) providing a new animation data format which requires less storage.

1. Introduction

Recent interests in facial modeling and animation have been spurred by the increasing appearance of virtual characters in film and video, inexpensive desktop processing power, and the potential for a new 3D immersive communication metaphor for human-computer interaction. Facial modeling and animation technique is a difficult task, because exact classifications are complicated by the lack of exact boundaries between methods and the fact that recent approaches often integrate several methods to produce better results. The classification of these methods is described in the survey report [1]. Many efforts have been put on physic-based muscle modeling to model anatomical facial behavior more faithfully. These are categorized into three: mass-spring systems, vector representation, and layered spring meshes. Mass-spring methods propagate muscle forces in an elastic spring mesh that models skin deformation [2]. The vector approach deforms a facial mesh using motion fields in delineated regions of influence [3]. A layered spring mesh extends a mass-spring structure into three connected mesh layers [4]. Mesh deformation plays a central role in computer modeling and animation. Animators sculpt facial expressions and stylized body shapes. They assemble procedural deformation and may use complex muscle simulations to deform a character's skin. Despite the tremendous amount of artistry, skill, and time dedicated to crafting deformations, there are few techniques to help with reuse.

In this paper, a novel parameterization of facial expressions is introduced. The parameters can be learned from existing face models or created from scratch. The obtained parameters can be applied on target face models dissimilar to the source model from which the parameters are taken in order to generate similar expressions on the target models. We also adopt a muscle-based animation system to obtain the parameters indirectly. It is tedious and difficult to make expressions by manipulating each control point. The proposed system provides a new alternative to make expressions which is easier and more intuitive. Facial animation by the parameterization requires less storage especially for highly complex face models with huge articulation data without reducing the range of deformation it provides.

In Section 3, the elastic skin model is described. Section 4 describes the details of the proposed facial parameterization. In Section 5, the expression cloning technique is introduced. In Section 6, a simplified facial muscle model is used to indirectly generate the parameters by manipulating the muscles. Section 7 describes the advantages of the proposed method and perspective for continuous animation.

2. Related Works

Applying facial expressions from human faces to computer-generated characters has been widely studied [5–11]. To control facial movement, facial expressions are analyzed into the position of feature points [5, 6, 11, 12] or the weights for blending premodeled expressions [7–10]. Our work is mostly motivated by the work [6]. In order to transform motion data from a source model to a target model, their method requires a dense mapping between the source and the target model. Then the motion vector at each vertex on the source model is dissipated among the corresponding vertices on the target model. Our method does not require such a dense mapping and computationally more efficient. References [8, 9] adopt an example-based approach for retargeting facial expressions. Examples-based approach requires a set of precomputed basis models to synthesize a new expressions. This approach is effective since the animators can use their imagination to create a set of basis expressions so that a novel bending expression can possibly represent their artistry. However, creating basis expressions is not a trivial work and these methods might lack generality.

To compute facial parameters from existing models, we assume that there is a “point-to-point” correspondence between them in order to derive motion vectors for each expression. This assumption might be too restrictive in some cases; however there are several techniques to establish correspondences between two different models [13–15]. Harmonic mapping is a popular approach for recovering dense surface correspondences between them [16].

3. Elastic Facial Skin Model

In this section, the underlying theory of the elastic skin model is introduced. An intuitive surface deformation can be modeled by minimizing physically inspired elastic energies. The surface is assumed to behave like a physical skin that stretches and bends as forces are acting on it. Mathematically this behavior can be captured by the energy functional that penalizes both stretching and bending [17–19]. Let d be the displacement function defined on the surface and let k_s and k_b be the parameters to control the resistance to stretching and bending, respectively, the elastic energy E is defined as

$$\begin{aligned} E(d) = \int_{\Omega} & k_s\left(\|d_u\|^2 + \|d_v\|^2\right) \\ & + k_b\left(\|d_{uu}\|^2 + 2\|d_{uv}\|^2 + \|d_{vv}\|^2\right) du dv, \end{aligned} \tag{1}$$

where the notations $d_x,\ d_{xy}$ are defined as $d_x = \partial d/\partial x,\ d_{xy} = \partial^2 d/\partial x \partial y$.

In a modeling application one would have to minimize the elastic energy in (1) subject to the user-defined constraints. By applying variational calculus, the corresponding Euler-Langrange equation that characterizes the minimizer of (1) can be expressed as

$$-k_s\Delta_s d + k_b\Delta_S^2 d = 0. \tag{2}$$

The Laplace operator in (2) corresponds to the Laplace-Beltrami operator [20]. Using the famous cotangent discretization of the Laplace operator, the Euler-Lagrange PDE turns into a sparse linear system:

$$\begin{aligned} -k_s\Delta d + k_b\Delta^2 d &= 0, \quad p_i \notin H \bigcup F, \\ d(p_i) &= d_i, \quad p_i \in H, \\ d(p_i) &= 0, \quad p_i \in F, \end{aligned} \tag{3}$$

where H is the handle vertices, and F is the fixed vertices. Interactively manipulating the handle H changes the boundary constraints of the optimization. As a consequence, this system has to be solved in each frame. Note that restricting to affine transformation of the handle H allows us to precompute the basis functions of the deformation. So, instead of solving (3) in each frame, only the basis functions have to be evaluated [21]. We will elaborate the details in the next section.

Figure 1 shows the results of deformation for two extreme cases (a) pure stretching ($k_s = 1,\ k_b = 0$) and (b) pure bending ($k_s = 0,\ k_b = 1$).

In general, the order k of Laplacian operator corresponds to the C^{k-1} continuity across the boundaries. For the facial skin deformation, we use the pure bending ($k_s = 0,\ k_b = 1$) surface model because the model can retain the C^1 continuity around the handle vertex H which is proved to be a good approximation of the skin deformations of various expressions.

4. Facial Parameter Estimation

In this section, we parameterize a set of existing face models using the elastic surface model. The facial parameters are calculated so that obtained parameters are precise enough to approximate the deformation of certain facial expression. The input consists of a face model with neutral expression and a set of face models with key expressions. To match up every vertices, all the models share the same number of vertices and triangles and have identical connectivity. Equation (3) can be expressed in matrix form by reordering the rows:

$$L\begin{pmatrix} d(P_H) \\ d(P_F) \\ d(P) \end{pmatrix} = \begin{pmatrix} \mathbf{d_i} \\ 0 \\ 0 \end{pmatrix}, \qquad L = \left(\begin{array}{cc} I_{H+F} & 0 \\ \hline \multicolumn{2}{c}{\tilde{\Delta}} \end{array}\right). \tag{4}$$

The solution of (4) can be explicitly expressed in terms of the inverse matrix L^{-1}. We observe that the explicit solution of (4) is

$$\begin{pmatrix} d(P_H) \\ d(P_F) \\ d(P) \end{pmatrix} = L^{-1}D_i, \qquad D_i = \begin{pmatrix} \mathbf{d_i} \\ 0 \\ 0 \end{pmatrix}. \tag{5}$$

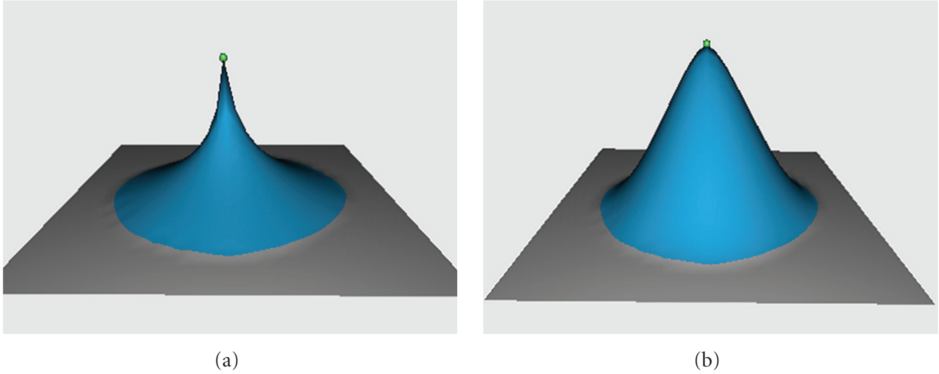


Figure 1: The surface is deformed by minimizing elastic surface energy subject to the user constraints. The gray area is the fixed region F, and the blue area is the free region. The vertex in green is the handle vertex H. (a) Pure stretching ($k_s = 1,\ k_b = 0$), (b) pure bending ($k_s = 0,\ k_b = 1$).

Let the kth column vector of L^{-1} be denoted by $L^{-1}(k) = B_k$, then the right-hand side of (5) can be decomposed as

$$\begin{aligned}\begin{pmatrix} d(P_H) \\ d(P_F) \\ d(P) \end{pmatrix} &= L^{-1}D_1 + L^{-1}D_2 + \cdots + L^{-1}D_m \\ &= L^{-1}(1) * d_1 + L^{-1}(2) * d_2 + \cdots + L^{-1}(m) * d_m \\ &= B_1 * d_1 + B_2 * d_2 + \cdots + B_m * d_m \\ &= \begin{pmatrix} B_1 & B_2 & \cdots & B_m \end{pmatrix} \begin{pmatrix} d_1 & d_2 & \cdots & d_m \end{pmatrix}^T, \end{aligned} \tag{6}$$

where m is the number of handle points. Note that the basis functions $\{B_i\}$ can be precomputed once the handle points are fixed and they can be reused for all the expressive face models of the same subject.

The left-hand side of the (6) can be computed for each expressive face by subtracting the neutral face. The facial parameters $\{d_i\}$ can be computed by using least square approximation method. We use the QR factorization method to solve the least square problem.

To obtain the basis functions $\{B_i\}$, handle region H corresponding to the facial control points needs to be defined. We adopt a subset of facial control points defined in MPEG-4 standard [22], which are distributed symmetrically over an entire front face. The total number of control points is forty-seven for our test model, and they are shown in Figure 2. Note that the number of facial control points and the location of each point are fully customizable. For instance, if mouth animation is the main objective such as a lip syncing animation, the majority of control points must be placed around the lips to increase the degree of freedom of the lip movement.

If no fixed region F is defined on the face model, undesired deformation would occur on the back of the head and around the ears. This is because the solution of (3) would try to keep the C^1 boundary conditions around the deformed handle region H. For our face models, the fixed region F is empirically defined on the vertices which are static under the change of expressions. In order to search for the fixed vertices, we let R to be the Euclidean distance between the tip of the nose and the center of the forehead, then if the Euclidean distance r between the vertex and the tip of nose is greater than the threshold value defined as $1.5\,R$ (for our test models), we put the vertex in the fixed region F.

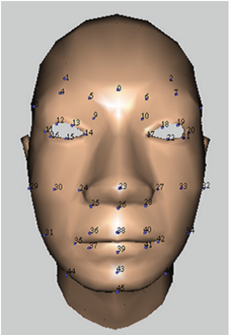


Figure 2: The control points are a subset of MPEG-4 feature points (FPs) and some additional points. The total number is forty-seven for the test models. We put more control points around eyes and mouth since those facial parts need to be deformed more flexibly than other facial parts to make facial expressions.

Figure 3 shows the generated face models by applying facial parameters computed to the neutral face model. In order to clarify the difference from the original, the corresponding original models are shown in the first row of the Figure 3. All the expressions are reproduced correctly enough even though we notice some slight differences between the two models such as the nostril area of the anger expression and the eyelid of the blink expression. We can mitigate the differences between the two models by placing additional controls points around the area.

4.1. Facial Expression Blending. Facial expression blending is a common technique for facial animation to create a novel expression by blending existing expressions.

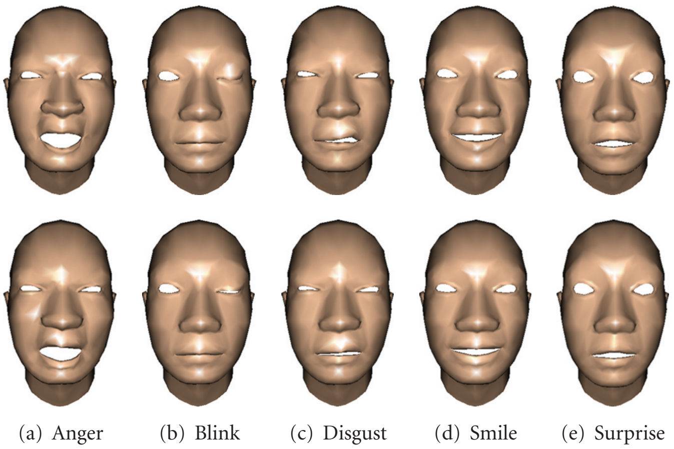


Figure 3: First row shows the original face model and the second row shows the face models generated using facial parameters calculated from the original models.

Given the set of facial parameters $\{d_i\}$ generated for each expression, a novel expression can be created by simply blending the facial parameter for each expression:

$$\begin{pmatrix} d(P_H) \\ d(P_F) \\ d(P) \end{pmatrix} = \Sigma_k w_k \begin{pmatrix} B_1 & B_2 & \cdots & B_m \end{pmatrix} \begin{pmatrix} d_1^k & d_2^k & \cdots & d_m^k \end{pmatrix}^T$$
$$= \begin{pmatrix} B_1 & B_2 & \cdots & B_m \end{pmatrix} \times \begin{pmatrix} \Sigma w_k d_1^k & \Sigma w_k d_2^k & \cdots & \Sigma w_k d_m^k \end{pmatrix}^T, \quad (7)$$

where $\{w_k \mid 0.0 \leq w_k \leq 1.0\}$ is the blending weight for each expression.

Figure 4 shows some examples of expression blending. The first row shows no textured images, and the second row shows textured images. The blending calculation is performed at the facial control points but not at every vertices, so the computational cost is relatively low.

We can also attenuate the displacement motion of each control point independently by adopting the importance map as suggested in [9] to increase the variety of expressions to be generated.

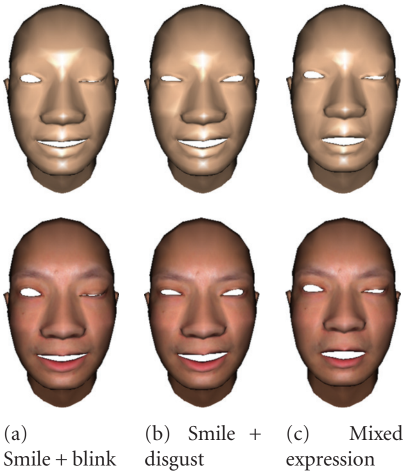


Figure 4: Expressions are blended by changing weight for each expression. (a) Blending smile (1.0) and blink (1.0), (b) blending smile (1.0) and disgust (0.5), (c) blending all expressions : anger (0.3) + blink (0.6) + disgust (0.3) + smile (0.1) + surprise (0.4).

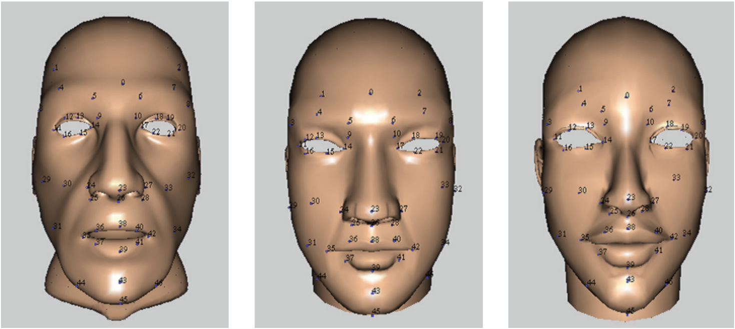
Table 1: Expression cloning. Time took to compute the basis functions for each target model.

	Vertex	Face	Time (basis function)
Source model	755	473	2.5 seconds
Target model 1	5226	3139	17.1 seconds
Target model 2	2362	1159	6.2 seconds
Target model 3	2254	997	5.5 seconds

5. Facial Expression Cloning

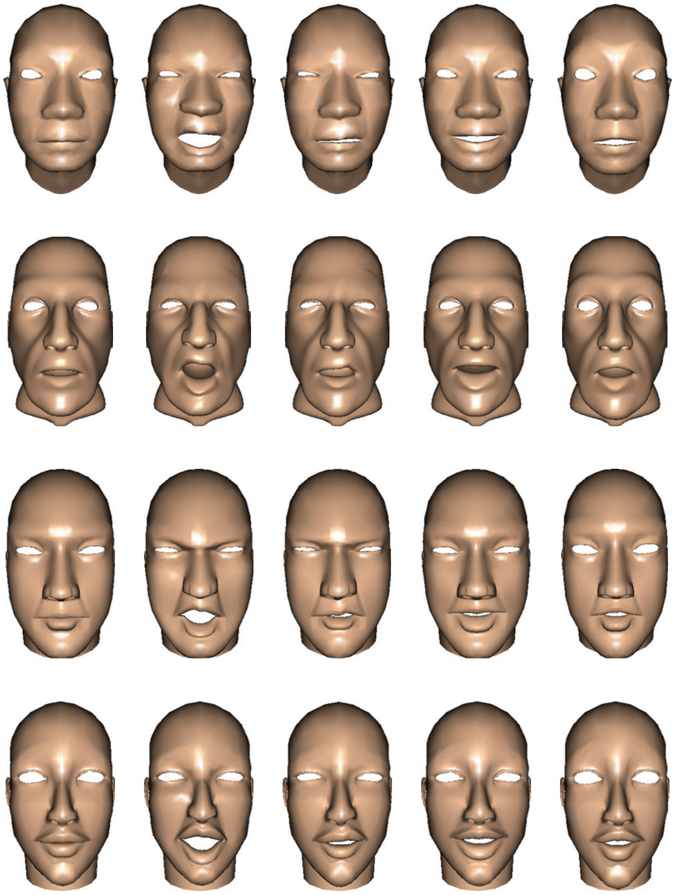
Expression cloning is a technique that copies expressions of a source face model onto a target face model. The mesh structure of the models needs not to be the same. Our proposed facial parameterization can be used for this purpose.

The first step selects the facial control points on the target model, each of which is exactly correspond to the control point on the source model. It takes no more than twenty minutes to select all facial control points on the target model. The second step computes the basis functions $\{B_i'\}$ for the target model as we did in the previous section. Table 1 shows the time took for computing the basis functions for each target model. In the third step, we copy the expressions on the target model. Given the facial parameters $\{d_i\}$ for each expression of the source model and the basis functions $\{B_i'\}$ obtained from the target model, each expressive target model is computed by using (6).



(a) Target model 1 (b) Target model 2 (c) Target model 3

Figure 5: Expression cloning. In order to copy expressions from the source model, exactly identical facial control points of the source model have to be defined on the target model. The selection of facial control points is done manually.



(a) Neutral (b) Anger (c) Disgust (d) Smile (e) Surprise

Figure 6: First row shows the source models and the other rows show target models generated by expression cloning. The expressions are retrieved from the source model and are copied on each target model. Left most column shows the neutral expression of each model.

To compensate for the scale difference between the source and the target model, each element of facial parameters $\{d_i\}$, a 3D displacement vector from the neutral face at the control point i, is normalized such that the norm is measure by the Facial Action Parameter Unit (FAPU). The FAPU is commonly set as the distance between the inner corners of the eyes of the model. We also assume that the model is aligned so that the y-axis points through the top of head, x-axis points through the left side of head and looking in the positive z-axis direction. If the target model is not aligned

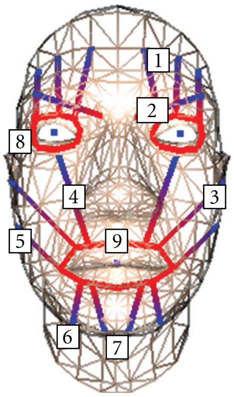


Figure 7: The major human facial muscles: (1) occipitofrontalis, (2) corrugator, (3) zygomaticus major, (4) levator, (5) buccinator, (6) depressor, (7) mentalis, (8) orbicularis oculi, and (9) orbicularis oris.

with the above coordinate system, it is aligned before the deformation is applied then moved back to the original position after the deformation.

In Figure 6 the source model and five basis expressions are shown in the first row and the cloned expressions on three different target models are shown in the following rows.

At the end of this paper we show the various expressions generated by a set of facial parameters in Figure 12. For each row, the facial parameter is same for all the models. The models are rendered with eyes to make each expression distinguishable.

6. Facial Deformation by Muscle Contraction

The facial animation by muscle contraction has been studied by many researchers [3, 23]. The simplified facial muscle model which is often used in computer animation should resemble that of anatomical models as much as possible. This is because the facial animation by muscle contraction must be general and fully understood by the animator. The basic canonical facial expressions have been throughly studied by Ekman and Friesen [24]; they have described in detail how the facial parts should move in order to make certain expressions. Even though there is little attention about the contraction of facial muscles underlying the facial skin, their analysis was very helpful to manipulate our simplified facial muscles to create certain expressions.

We define two types of muscles: a linear muscle that pulls and a sphincter muscle that squeezes the nearby skin elements. Similar pseudomuscle model is first proposed in [3]. In [3], they have succeeded to generate various facial expressions from a set of simple facial muscles. Figure 7 shows the muscle models we use in the following sections.

Most of the methods proposed before using a facial muscle model try to attach nearby skin elements to a pseudomuscle in the registration stage then deform the skin elements as the muscle contracts. The magnitude of the deformation is determined from the relative position of the skin element from the muscle. In a physically based facial modeling approach [25], the deformation is applied as nodal forces to three-layered mass-spring facial skin system. Some drawback of previous approaches is the complexity of the interaction between the muscle and nearby skin elements and the relatively high computational costs due to the finite elements method for skin deformation [25]. The uniqueness of the proposed method is that each muscle's contraction drives only the nearby facial control points that we define in the previous sections but not all the nearby skin elements as proposed before. The approach alleviates the painstaking task required to register all the skin elements to the corresponding muscle and has low computational costs. Finally, the skin elements other than the control points are deformed as the solution of the elastic skin model described in the Section 3. The details are described in the following section.

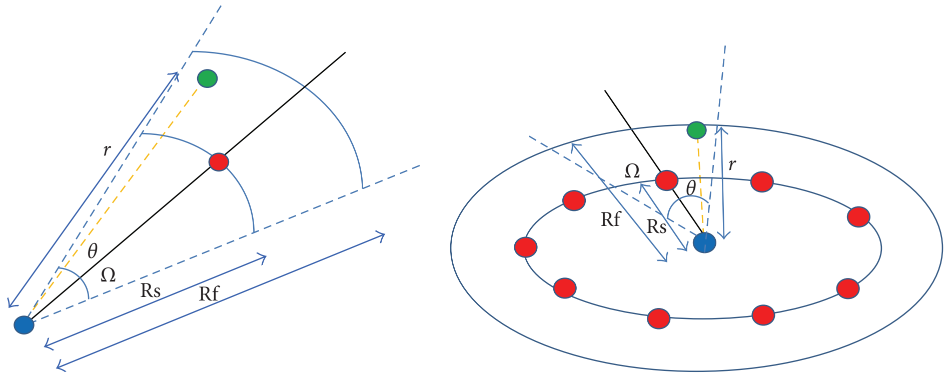
6.1. Facial Muscle Registration. In order to generate natural expressions and provide the animator easier operability, a set of simplified major facial muscles is defined and each of them is attached to the mesh by referring a anatomical model. These tasks are done manually and must be adopted for each different face model. Note that each muscle is a virtual edge connecting the two vertices (attachment points) of the mesh.

The fundamental of the linear muscle is that one end is the bony attachment that remains fixed, while the other end is embedded in the soft tissue of the skin. When the muscle is operated, it contracts along the two ends of the muscle. Each muscle has maximum and minimum zones of influence; however there is no analytic methods to measure them since only the surface points could be measured and the range of influence varies greatly from face to face [3].

The sphincter muscles around the mouth and the eyes that squeeze the skin tissue are described as an aggregate of linear muscles radially arranged from a pseudocenter point. The pseudo center is calculated by fitting an elliptic curve to the points defining the sphincter muscle.

Figure 8 shows the two kinds of the muscle model. The end point of each muscle is colored in blue at the bony attachment and in red at the skin attachment. As shown in Figure 7, the facial muscles are symmetrically defined across the center of the face however each muscle contracts independently.

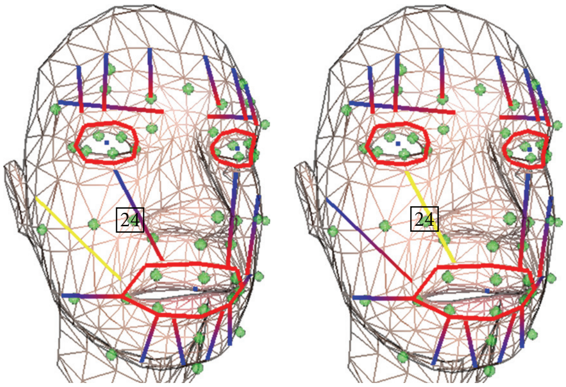
To compute the zone of maximum and minimum influences, we adopt the method proposed in [3]. Each linear muscle has the radial falloff and the angular falloff calculated from the rest position. At the skin attachment (not at the bony attachment), the influences is maximum (1.0) and gradually falloff to minimum (0.0) using cosine curve as in (8) and (9). Each muscle registers the nearby facial control points along with its own radial influence RI and angular influence AI if they reside in the influence region of the muscle. At the end of the registration, each control point is registered by one or more than one muscle elements depending on the zones of influence. In cases that there are any control points which no muscles register, the zones of



(a) Linear muscle

(b) Sphincter muscle

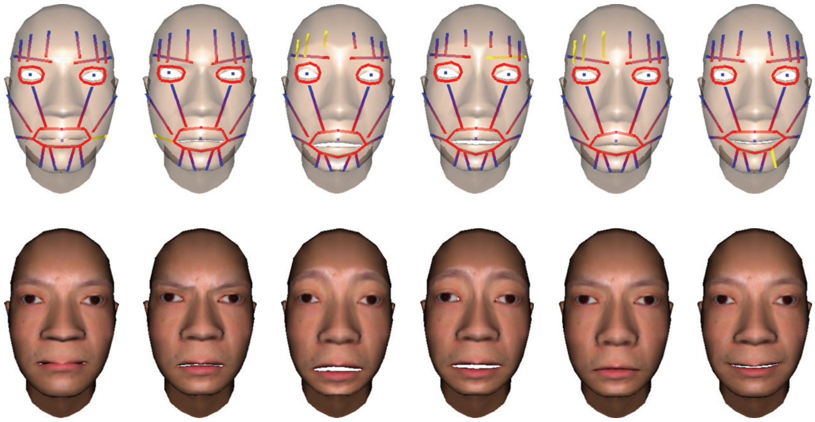
FIGURE 8: The blue vertex is the fixed bony attachment point, and the red vertex is the skin attachment point. The green vertex is a facial control point within the zone of influence. When the muscle contract, the red point is moved along the muscle endpoints. Rs: radial distant at the muscle's registration, Rf: radial fall off distance, Ω: angular fall off, θ: angle of the control point from the muscle.



(a)

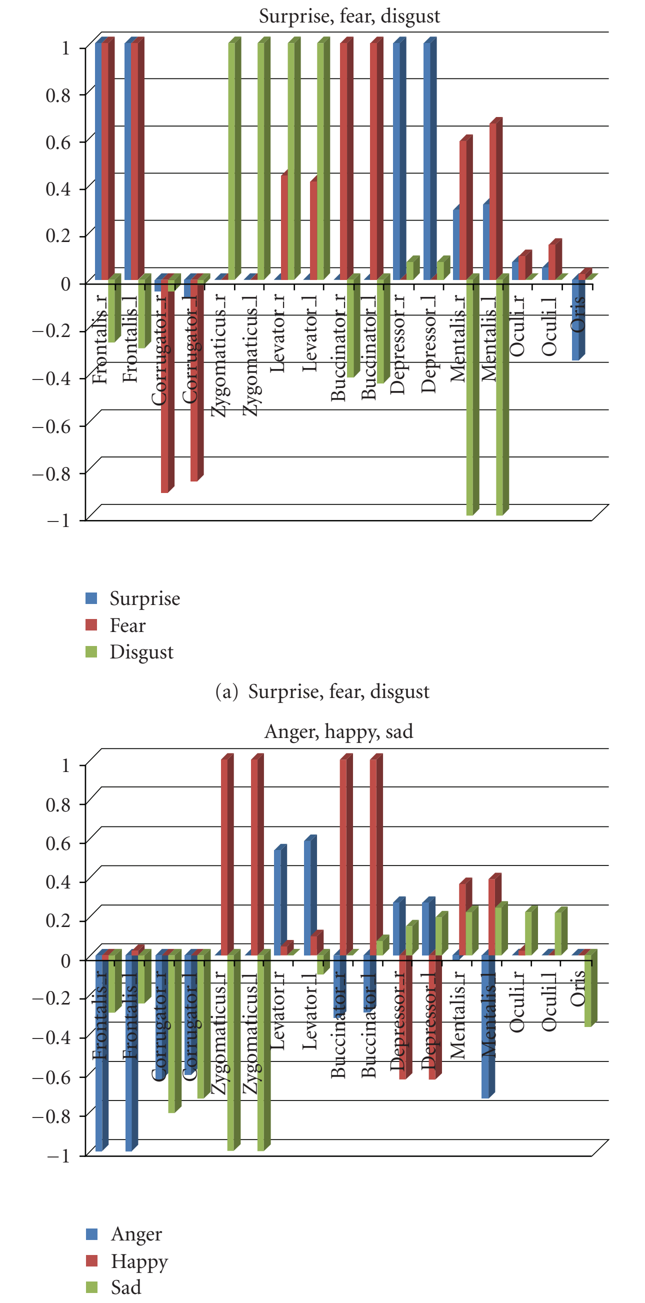
(b)

FIGURE 9: Facial muscle registration. The control point (index 24) is influence by two linear muscles. (a) Right levator (RI = 0.756, AI = 0.825), and (b) right zygomaticus major (RI = 0.951, AI = 0.468).



(a) Disgust (b) Anger (c) Surprise (d) Fear (e) Sad (f) Happy

FIGURE 10: Six canonical facial expressions by muscle contraction.



(a) Surprise, fear, disgust

(b) Anger, happy, sad

Figure 11: Amounts of muscle contraction for each expression. If the amount is positive, the muscle stretches and if negative the muscle shrinks.

influence must to be adjusted until every control point is registered by at least one muscle:

$$\begin{aligned} \mathrm{RI} &= \cos\left(\frac{1.0 - r}{\mathrm{Rs}}\frac{\pi}{2}\right) \quad \text{if } (r \le \mathrm{Rs}), \\ \mathrm{RI} &= \cos\left(\frac{r - \mathrm{Rs}}{\mathrm{Rf} - \mathrm{Rs}}\frac{\pi}{2}\right) \quad \text{if } (\mathrm{Rs} \le r \le \mathrm{Rf}), \end{aligned} \tag{8}$$

where r is the radial distance between the facial control point and the muscle's bony attachment point

$$\mathrm{AI} = \cos\left(\frac{\theta}{\Omega}\frac{\pi}{2}\right), \tag{9}$$

where θ is the angle between the linear muscle at rest position and the facial control point.

Figure 9 illustrates the registration of a facial muscle. The control point is influenced by two linear muscles and is registered with its own radial and angular influence values.

6.2. Facial Expression by Muscle Contraction. In our simplified muscle models, the linear muscle contracts only along the muscle's endpoints. When the muscle contracts, the displacement vector dv at the skin attachment point (the red point in Figure 8) from the rest position is calculated and it is dissipated among the registered facial control points. The amount of displacement of each registered control point $\{df\}$ is defined as a function of RI, AI, and dv. We use the following simple formula to drive the translation of the facial control point:

$$df = dv * \frac{\mathrm{RI} + \mathrm{AI}}{2.0}. \tag{10}$$

Since the facial control point might be registered by other muscles, the final displacement, the total of the displacement by each muscle, is given by

$$df^{\text{total}} = \Sigma_k df_k = \Sigma_k dv_k \frac{\mathrm{RI}_k + \mathrm{AI}_k}{2.0}. \tag{11}$$

Finally the deformed model by the muscle contractions is calculated by the product of the basis functions $\{B_i\}$ and the facial parameters $\{df_i^{\text{total}}\}$.

In Figure 10, six canonical expressions are created by contracting facial muscles by referring the analysis results [24]. In Figure 11, the amount of each muscle contraction of each expression is shown in two separate graphs to remove clutters. Note that each of the symmetric muscle pair can be contracted independently. If the amount is positive, the muscle stretches and if negative the muscle shrinks along the two ends of the linear muscle. The displacement vector of each muscle is calculated as

$$dv = w * \left(\frac{FAPU}{10.0}\right) * \overline{dm}(-1.0 \le w \le 1.0), \tag{12}$$

where $\overline{dm}$ is the normalized muscle vector, and w is the amount of muscle contraction from the user setting.

7. Discussion

The proposed parameterization of facial expression has advantages in terms of storage size. The conceptual storage

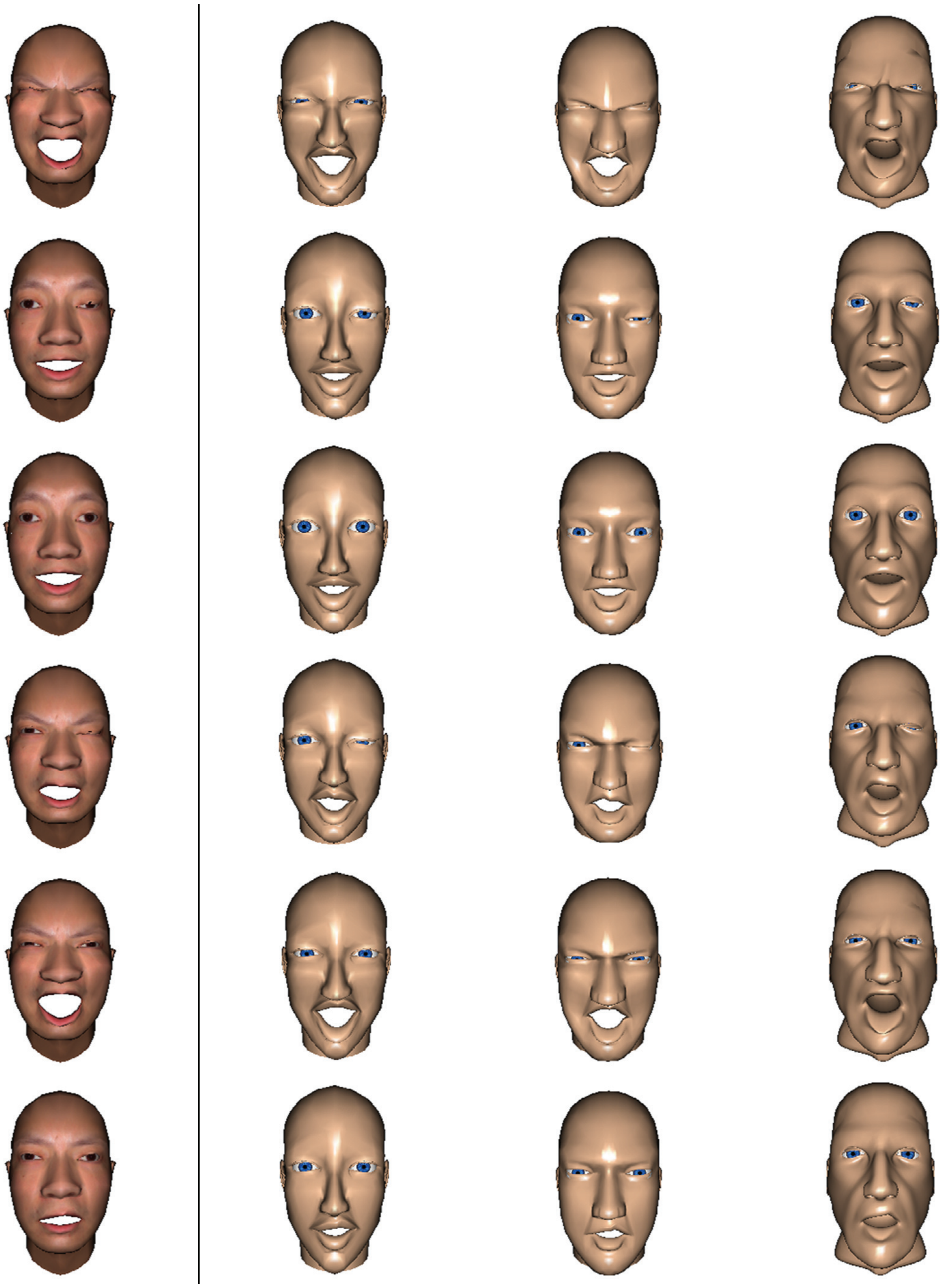


Figure 12: First column: the source model and expressions. Second column through the last column: the cloned expressions. Models have different shapes but expressions are well represented.

size required to store the animation data is given by

$$\begin{aligned} ST_{\text{param}} = {} & (\#\text{ of facial\_control\_points}) * (3\text{ floats}) \\ & * (\#\text{ of expressions}) \\ & + ST_B(\text{basis\_functions}) \end{aligned} \tag{13}$$

for the proposed method, where $ST_B(\text{basis\_functions})$ is the size of storage required to store the precomputed basis functions and it is defined as

$$\begin{aligned} ST_B(\text{basis\_functions}) = {} & (\#\text{ of vertices}) * (1\text{ floats}) \\ & * (\#\text{ of facial\_control\_points}). \end{aligned} \tag{14}$$

Meanwhile for a traditional system that simply stores the displacement vector at every vertex, it is given as

$$ST_{\text{noparam}} = (\#\text{ of vertices}) * (3\text{ floats}) * (\#\text{ of expressions}). \tag{15}$$

So the required storage size is decreased by the ratio

$$\frac{ST_{\text{param}}}{ST_{\text{noparam}}} = \left(\frac{\#\text{ of facial\_control\_points}}{\#\text{ of vertices}}\right) + \frac{1}{3} * \left(\frac{\#\text{ of facial\_control\_points}}{\#\text{ of expressions}}\right). \tag{16}$$

As this formula indicates, the proposed method requires less storage and has much lower memory footage if the size of the mesh is very large and the number of the expressions exceeds the number of the control points. It is a possible scenario since highly detailed facial animation might require a large number of blendshapes, for instance, the facial animation of Gollum in the feature film. *The Two Towers* require 675 blendshapes [26].

Character animation specifically facial animation requires continuous deformation in animation time frame.

Key framing is a common technique to continuously deform a model by interpolating key framed poses or channel data. The various methods of interpolation and extrapolation control the visual effects of the continuous deformation in the animation time frame. By using the proposed parameterization, the displacement of each facial control point can be set directly by tweaking the position at each key-frame. It is also possible to set the facial control points indirectly by blending canonical expressions described in Section 4.1 or by the muscle contraction described in Section 6.

The deformation by the limited control points requires low computational cost since only the sum of the scalar products of the basis functions $\{B_i\}$ and the facial parameters $\{d_i\}$ is required to get the resulting model. Smooth deformation by a limited facial control points is a key technique when creating a facial animation from motion capture (mocap) data. Several approaches have been studied. For instance Lorenzo and Edge [27] use Radial Basis Functions (RBFs) to transfer facial motion data to a target mesh dissimilar to the original actor from whom it was captured. Our proposed method can be nicely coupled with mocap data if the control points on the target mesh are exactly matched with the markers on the actor.

8. Conclusion

The elastic deformable surface model has been used as a deformation tool especially for elastic organic models. We have introduced a novel parameterization of facial expression using the elastic deformable surface model. The parameterization uses a small number of facial control points to deform a higher resolution surface smoothly and continuously.

In order to obtain parameters for each expression two approaches are introduced. The first approach retrieves parameters directly from existing models by least square minimization method. The parameter can also be created from scratch by moving each control points using the imagination of the animator.

The other approach indirectly obtains parameters by manipulating the facial muscles. The method by the muscle contraction is more intuitive and less daunting task compared with the former method.

The obtained facial parameters can be applied on other target model even if the mesh structure (number of vertices, number of triangles, and connectivity) is different from the source model. A key-framed facial animation can seamlessly use the proposed parameterization. The parameter values could be provided from mocap data.

The method could be used as a postprocessing tool to compress existing animation data, since it requires less storage especially for highly complex mesh objects with huge articulation data without sacrificing the quality of the original animation as much as possible.

Future research could explore the automated detection of the control points on the face model. Several heuristic approaches have been studied [6, 28]. If the target mesh is different from the source mesh, not only the ratio of the size but also the local features at control points, for example, the human head and the dog head, the facial parameters obtained cannot be directly applied. A possible solution is to define the parameters with respect to the local coordinate system at each control point of the source model then for the target model the parameter is reconstructed using the local coordinate system at the corresponding control point of the target model. Similar method is suggested in [6].

References

[1] J. Noh and U. Neumann, “A survey of facial modeling and animation techniques,” Technical Report 99-705, USC, 1998.

[2] S. M. Platt and N. I. Badler, “Animating facial expressions,” in *Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’81)*, vol. 15, pp. 245–252, Dallas, Tex, USA, August 1981.

[3] K. Waters, “A muscle model for animating three-dimensional facial expression,” in *Proceedings of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’87)*, vol. 21, pp. 17–24, Anaheim, Calif, USA, July 1987.

[4] D. Tezopoulos and K. Waters, “Physically-based facial modeling, analysis, and animation,” *Journal of Visualization and Computer Animation*, vol. 4, pp. 73–80, 1990.

[5] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, “Making faces,” in *Proceedings of the Annual Conference on Computer Graphics (SIGGRAPH ’05)*, pp. 17–24, 2005.

[6] J. Yong and N. Ulrich, “Expression cloning,” in *Proceeding of 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 277–288, 2001.

[7] F. Pighin, R. Szeliski, and D. H. Salesin, “Resynthesizing facial animation through 3D model-based tracking,” in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV ’99)*, vol. 1, pp. 143–150, Kerkyra, Greece, September 1999.

[8] B. Choe and H.-S. Ko, “Analysis and synthesis of facial expressions with hand-generated muscle actuation basis,” in *ACM SIGGRAPH Courses*, 2006.

[9] H. Pyun, Y. Kim, and W. Chae, “An example-based approach for facial expression cloning,” in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 167–176, 2003.

[10] P. Joshi and W. C. Tien, “Learning controls for blend shape based realistic facial animation,” in *ACM SIGGRAPH Courses*, 2006.

[11] L. Williams, “Performance-driven facial animation,” *Computer Graphics*, vol. 24, no. 4, pp. 235–242, 1990.

[12] Y. Wang, X. Huang, C.-S. Lee, et al., “High resolution acquisition, learning and transfer of dynamic 3-D facial expressions,” *Computer Graphics Forum*, vol. 23, no. 3, pp. 677–686, 2004.

[13] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, “Multiresolution analysis of arbitrary meshes,” in *Proceedings of the 22nd Annual ACM Conference on Computer Graphics and Interactive Techniques*, pp. 173–180, Los Angeles, Calif, USA, August 1995.

[14] R. W. Sumner and J. Popović, “Deformation transfer for triangle meshes,” in *Proceedings of International Conference on Computer Graphics and Interactive Techniques*, vol. 23, pp. 399–405, August 2004.

[15] A. W. F. Lee and D. Dobkin, “Multiresolution mesh morphing,” in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 343–350, 1999.

[16] T. Kanai, H. Suzuki, and F. Kimura, “Metamorphosis of arbitrary triangular meshes,” *IEEE Computer Graphics and Applications*, vol. 20, no. 2, pp. 62–75, 2000.

[17] G. Celniker and D. Gossard, “Deformable curve and surface finiteelements for free-form shape design,” in *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 257–266, 1991.

[18] D. Terzopoulos and J. Platt, “Elastically deformable models,” in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Technique*, pp. 205–214, 1987.

[19] W. Welch and A. Witkin, “Variational surface modeling,” in *Proceedings of the 19th Annual ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’92)*, vol. 26, pp. 157–166, Chicago, Ill, USA, July 1992.

[20] G. Taubin, “A signal processing approach to fair surface design,” in *Proceedings of the 22nd Annual ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’95)*, pp. 351–358, Los Angeles, Calif, USA, August 1995.

[21] M. Botsch and L. Kobbelt, “An intuitive framework for real-time freeform modeling,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 630–634, 2004.

[22] ISO, Mpeg-4 international standard, moving picture experts group, 2003.

[23] Y. Lee and D. Tezopoulos, “Realistic modeling for facial animation,” in *Proceedings of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’95)*, pp. 55–62, 1995.

[24] P. Ekman and W. V. Friesen, *Unmasking the Face: A Guide to Recognizing Emotions from Facial Clues*, Prentice-Hall, Upper Saddle River, NJ, USA, 1975.

[25] Y. Lee, D. Terzopoulos, and K. Waters, “Realistic modeling for facial animation,” *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’95)*, pp. 55–62, 1995.

[26] J. Fordham, “Middle earth strikes back,” *Cinefex*, no. 92, pp. 71–142, 2003.

[27] M. S. Lorenzo and J. D. Edge, “Use and re-use of facial motion capture data,” in *Proceedings of the Vision, Video, and Graphics*, pp. 1–8, 2003.

[28] Z. Deng and U. Neumann, “Data-driven 3D facial animation,” in *Expressive Visual Speech Generation*, pp. 29–59, Springer, Berlin, Germany, 2008.

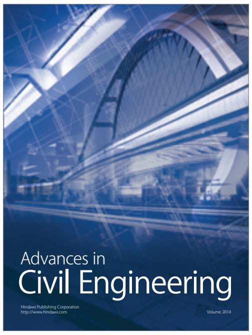
Journal of
Engineering

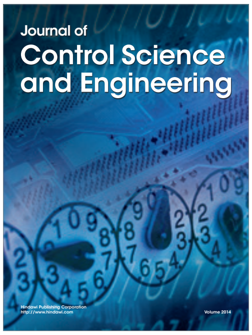
The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

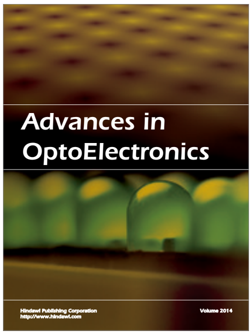
International Journal of
Distributed
Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

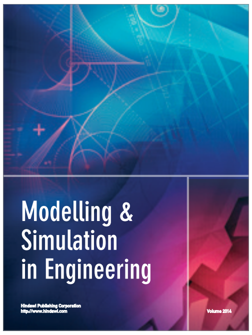
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

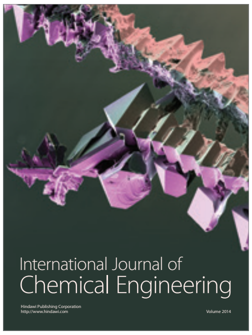

Advances in
OptoElectronics

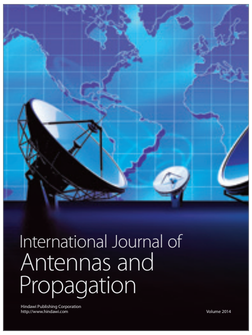
VLSI Design

International Journal of
Navigation and
Observation

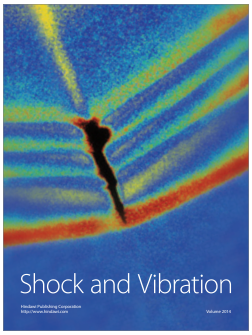

Modelling &
Simulation
in Engineering

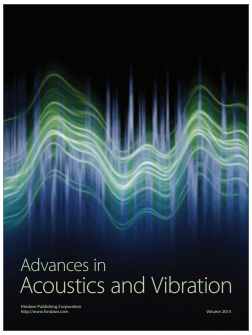
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration