EG UK Theory and Practice of Computer Graphics (2008)
Ik Soo Lim, Wen Tang (Editors)

Facial Expression Transferring with a Deformable Model

Guofu Xiang, Xiangyang Ju, Patrik O'B. Holt, Lin Shang

Cognitive Engineering Research Group, School of Computing, The Robert Gordon University, Aberdeen, AB25 1HG, UK

Abstract

This paper presents an automated approach to transferring facial expressions from a generic facial model onto various individual facial models without requiring any prior correspondences and manual interventions during the transferring process. This approach automatically detects the corresponding feature landmarks between models, and establishes the dense correspondences by means of an elastic energy-based deformable modelling approach. The deformed model, obtained through the deformation process, maintains the same topology as the generic model and the same shape as the individual one. After establishing the dense correspondences, we first transfer the facial expressions onto the deformed model by a deformation transfer technique, and then obtain the final expression models of individual models by interpolating the expression displacements on the deformed model. The results show that our approach is able to produce convincing results on landmark detection, correspondence establishment and expression transferring.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling – Geometry Algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – Animation

1. Introduction

Modelling and animating realistic facial models is a substantial challenge in computer graphics, especially for facial expressions, because we are so familiar with human faces and very sensitive to "unnatural" subtle changes in faces. There has been intensive research in this area, and the reader is referred to the recent surveys [HT04, Pig06].

Generally, to create realistic high-quality expressions for facial models from first principles requires a tremendous amount of artistry, skill and time. Thus, it has been popular to take advantage of existing data to produce new models instead of creating them [NN01, SP04]. This dramatically reduces the repeated work for artists.

In order to transfer facial expressions between different models, one key problem is to establish the correspondences between the models. The correspondence problem is also a fundamental problem in numerous fields such as shape registration [GMGP05], mesh morphing [Ale02], and computer vision. There are a large number of approaches that have been proposed to address this problem [HLS07, PMW05]. Most of the approaches require the user to specify several initial point-to-point correspondences (or landmarks) between the input models.

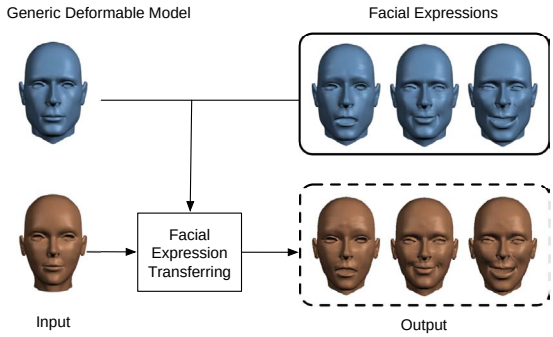

Figure 1: *Illustration of the outline of facial expression transferring.*

This paper describes an *automated* approach for facial expression transferring, which automatically detects the landmarks and establishes the dense correspondences by an energy-driven method. Figure 1 shows the outline of our

DOI: 10.2312/LocalChapterEvents/TPCG/TPCG09/117-124

research work. We built a generic deformable model with represented its facial expressions. Given an individual facial model, the facial expression transferring will automatically produce high-quality facial expressions for that model based on the generic facial expressions.

In the following section, we provide an overview of selected related work in this field. In Section 3, we present the details of our facial expression transferring approach. The results and the conclusion are presented in Section 4 and 5, respectively.

2. Related Work

Recently, a number of approaches have been proposed that reuse existing data to produce new models. Noh and Neumann first proposed the concept of expression cloning [NN01], which retargets facial expression animation created by any tools or methods to new models. This approach computed the dense surface correspondences by volume morphing with Radial Basis Function (RBF) followed by a cylindrical projection. Fifteen heuristic rules were used to identify dozens of initial correspondences which were needed for the RBF morphing. However, since the RBF morphing uses fixed markers, the final dense correspondences strongly depend the accuracy of the initial correspondences. Deformation transfer [SP04, BSPG06] applies the deformation exhibited by a source triangle mesh onto a different target triangle mesh. This approach uses the template-fitting algorithm [ACP03] to build the correspondence, which is controlled by dozens of user-selected markers. Vlasic *et al.* proposed a method, which used multilinear models for mapping video-recorded performance of one individual to facial animations of another [VBP05]. An example-based approach [PKC*03] proposed by Pyun *at al.* clones facial expressions of a source model to a target model while reflecting the characteristic feature of the target model.

Due to its great challenge in many research fields, numerous research efforts are devoted to establishing correspondences between different meshes [HLS07, PMW05]. The template fitting method [ACP03, SP04] deforms a template surface to a target surface subject to minimizing the combining errors of the smoothness of the template surface and the distance between the two surfaces. Our approach for the correspondence establishment is similar to this procedure, but we developed in the context of the deformable model, which uses the linear variational deformation technique [BS08]. Steinke *et al.* proposed a method to establish correspondences by learning a combination of features from a given training set of aligned human heads [SSB07] while Anguelov *et al.* proposed an unsupervised algorithm for registering 3D surface scans of an object undergoing significant deformations [ASP*05].

Deformable object modelling has been studied in computer graphics for more than three decades, across of a range of applications and there are detailed surveys of this field (see e.g. [GM97]). Recently, a self-organizing deformable model was proposed to project a 3D object mesh model onto a surface of another target object based on competitive learning and energy minimization [MMN07].

In this paper, we propose an automated landmark detection algorithm for finding the corresponding landmarks between the generic model and that of individuals so that our approach is fully automated and does not required the user to provide selected landmarks for each individual model. Our deformable model uses soft-assigning landmarks. We establish the dense correspondences by deforming the generic model onto that of individuals. This procedure is similar to [ACP03, SP04, MMN07], however, our deformation process is subject to minimizing the elastic bending and stretching energy of the surface, which is a physically based surface deformation. Moreover, this variational minimization problem can be efficiently solved [BS08]. The whole transferring procedure will be detailed in the next section.

3. Facial Expression Transferring

The goal of the facial expression transferring is to copy the facial expressions from the generic facial model onto the individual facial models considering the differences in scale, topology, and shape. Figure 2 shows the pipeline of the transferring process. Our approach consists of landmark detection, deforming the deformable model, and expression transferring. We first find the corresponding feature landmarks on the individual model for each predefined landmark on the generic model, deform the generic facial model toward the individual facial model to get the deformed facial model, and then transfer facial expressions from the generic facial model onto the deformed facial model. Finally, we get the facial expressions on the individual facial model by interpolating the expression displacements on the deformed facial model.

We choose triangle meshes as the representations for all our facial models. In this paper, the notations on triangle meshes described in [BPK*07] are used.

3.1. Automated Landmark Detection

For our generic facial model $\mathcal{S}$, we have specified twenty landmarks $\mathcal{L} = \{\mathbf{m}_i, 1 \leq i \leq 20\}$ according to the MPEG-4 specification on feature points of human faces [LP99] (See Figure 4(A)). These landmarks represent certain feature points on the face, such as the mouth corners, the nose tip, etc. In this paper, the landmarks are represented by the triangles they locate at and their barycentric coordinates.

Assuming that the input models are roughly globally aligned, the goal of the landmark detection is that, for each landmark on the generic model $\mathcal{S}$, find the corresponding landmark located at the similar feature point on the target model$\mathcal{T}$. Our landmark detection algorithm (Algorithm 1)

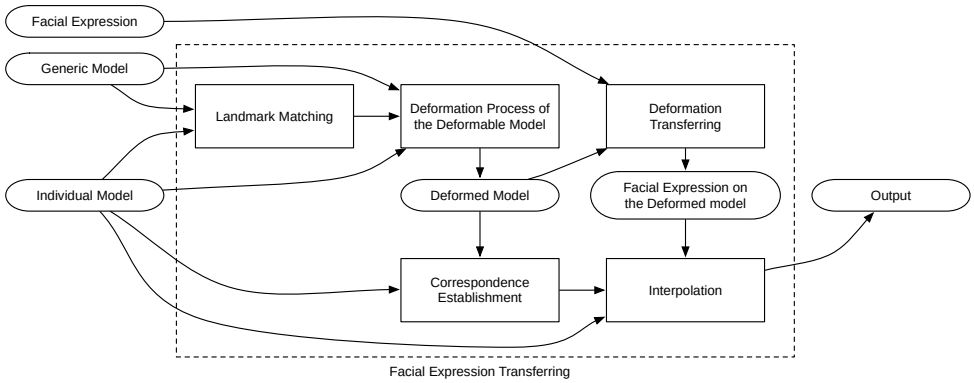

Figure 2: *The pipeline of our approach to transferring facial expressions.*

is developed from the iterative closest points (ICP) algorithm [RL01], which has been widely used for rigid shape registration.

Input : $\mathcal{S}, \mathcal{T}, \mathcal{L}$;
Output: The corresponding landmark set $\mathcal{L}' = \{\mathbf{m}_i', 1 \le i \le |\mathcal{L}|\}$ on $\mathcal{T}$.

foreach $\mathbf{m}_i \in \mathcal{L}$ *on* $\mathcal{S}$ **do**
  Initialize the registration error ε_i, *rotation* $\mathbf{R}_i$, *translation* $\mathbf{t}_i$, *initial registration states* $S = \{s_j = (\mathbf{R}_{j0}, \mathbf{t}_{j0}), 1 \le j \le |S|\}$
  Find the neighbourhood points $\mathcal{P}_i = \{\mathbf{p}_i, 1 \le i \le |\mathcal{P}|\}$ around $\mathbf{m}_i$ with the geodesic distance $r_i = g(\mathbf{p}_i, \mathbf{m}_i) < r_d$
  foreach $s_j \in S$ **do**
    Apply s_j to $\mathcal{P}_i \to \mathcal{P}_i'$
    Register $\mathcal{P}_i'$ to $\mathcal{T}$ using ICP $\to$ registration error ε_j, rotation $\mathbf{R}_j$ and translation $\mathbf{t}_j$
    if $\varepsilon_j < \varepsilon_i$ **then**
      $\varepsilon_i = \varepsilon_j, \mathbf{R}_i = \mathbf{R}_j, \mathbf{t}_i = \mathbf{t}_j$
    end
  end
  Apply $\mathbf{R}_i, \mathbf{t}_i$ to $\mathbf{m}_i \to \mathbf{m}_i^r$
  Find the closest point $\mathbf{m}_i'$ to $\mathbf{m}_i^r$ on $\mathcal{T}$.
end

Algorithm 1: Automated Landmark Detection

To find the neighbourhood surface points, we used the algorithm proposed by Surazhsky *et al.* [SSK*05] for computing the geodesic distance between two surface points. In our present implementation, we set the neighbourhood size to $r_d = 6\% \times w_{\text{eye}}$, where w_{eye} is the average width of the eyes of $\mathcal{S}$, and nine initial registration states are used, which are the translations with $\pm r_d/2$ in horizontal and vertical directions on the plane orthogonal to the view direction. During finding the closest points, we use a compatible condition test, which is the angle between the normal directions at the two points should be less than 60°, to get better matches in regions such as the lips.

3.2. Deformable Model

In order to establish the dense correspondences between the generic facial model and an individual facial model, we developed a deformable model, which deforms the generic facial model towards the individual facial model and the deformed facial model have the same topology as the generic facial model but in the same shape as the individual facial model.

The deformation process is controlled by two energy terms: the potential and the strain energies. The potential energy is the source of driving force to pull the generic facial model toward the target, while the strain energy is used to not allow dramatic changes in mesh.

Let us denote by $\mathcal{S}, \mathcal{T} \subset \mathbb{R}^3$ the source and the target two-manifold surfaces. The source surface is parameterized by a function $\mathbf{p} : \Omega \subset \mathbb{R}^2 \mapsto \mathcal{S} \subset \mathbb{R}^3$.

This surface $\mathcal{S}$ is to be deformed to $\tilde{\mathcal{S}}$ by adding to each point $\mathbf{p}(u,v)$ a displacement vector $\tilde{\mathbf{d}}(u,v)$, such that $\tilde{\mathcal{S}} = \tilde{\mathbf{p}}(\Omega)$ with $\tilde{\mathbf{p}} = \mathbf{p} + \tilde{\mathbf{d}}$ (See Figure 3).

The total energy E_{total} we try to minimize consists of two terms:

$$E_{\text{total}} = E_s + E_p, \tag{1}$$

where E_s is the strain energy of the surface (shell) and E_p is the potential energy cause by landmarks and closest points. E_p is the source of driving force and E_s is smooth term.

It is known from differential geometry that the first and second fundamental forms, $\mathbf{I}(u,v), \mathbf{II}(u,v) \in \mathbb{R}^{2\times 2}$, can be used to measure geometrically intrinsic (i.e., parameterization independent) properties of surface $\mathcal{S}$, such as lengths, areas, and curvatures. The change of fundamental forms

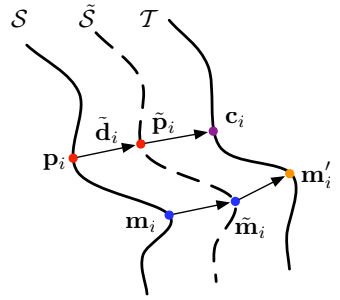

Figure 3: *Illustration of the surface deformation from $\mathcal{S}$ to $\mathcal{T}$.*

therefore yields a measure

$$E_s = \iint_\Omega (k_s \|\tilde{\mathbf{I}} - \mathbf{I}\|_F^2 + k_b \|\tilde{\mathbf{II}} - \mathbf{II}\|_F^2) \mathrm{d}u\mathrm{d}v, \qquad (2)$$

where $\|\cdot\|_F$ is the Frobenius matrix norm, k_s and k_b are parameters to control the resistance of stretching and bending deformation.

The potential energy

$$\begin{aligned} E_p &= E_m + E_c \\ &= \iint_\Omega k_m \|\tilde{\mathbf{m}} - \mathbf{m}'\|^2 \mathrm{d}u\mathrm{d}v + \iint_\Omega k_c \|\tilde{\mathbf{p}} - \mathbf{p}'\|^2 \mathrm{d}u\mathrm{d}v, \end{aligned} \qquad (3)$$

where E_m and E_c are the potential energies caused by landmarks and closest points, k_m and k_c are the weights, $\|\cdot\|$ is the vector L^2 norm.

Since the expression of the strain energy is non-linear, it is difficult to solve the problem directly. To simplify the problem, we have to linearize it [BPK*07] and then employ the variational approach to solve the minimization problem. Finally, we get the linearized problem of our deformable model (for continuous case).

$$\begin{aligned} (-k_s \Delta \tilde{\mathbf{d}} + k_s \Delta^2 \tilde{\mathbf{d}}) + 2k_m(\mathbf{m} + \tilde{\mathbf{d}} - \mathbf{m}') \\ + 2k_c(\mathbf{p} + \tilde{\mathbf{d}} - \mathbf{p}') = 0 \end{aligned} \qquad (4)$$

The discretized form of the above equation can be assembled into a linear system.

$$(-k_s \mathbf{L} + k_b \mathbf{L}^2 + \mathbf{M} + \mathbf{C})\tilde{\mathbf{d}} = \mathbf{b}_m + \mathbf{b}_c, \qquad (5)$$

where $\mathbf{L}$, $\mathbf{M}$, $\mathbf{C}$ are sparse matrices. The matrix $\mathbf{L}$ is the same as that in [BPK*07] which is derived from the discretized Laplace-Beltrami operator. The sparse matrix $\mathbf{M}$ only has non-zeros at the vertices, which belong to the triangles that the landmarks locate at and the values are $2k_m$ multiplied by the corresponding component of the barycentric coordinates. The sparse matrix C is a diagonal matrix with $2k_c$ at each diagonal element. $\mathbf{b}_m$ and $\mathbf{b}_c$ are the vectors related to the corresponding landmarks and closest points on the source mesh and the target mesh. This sparse linear system can be solved efficiently by the TAUCS sparse matrix solver.

Given the generic facial model $\mathcal{S}$ with the predefined landmarks $\mathcal{L}$ and an individual facial model $\mathcal{T}$, in order to deform $\mathcal{S}$ towards $\mathcal{T}$, we first use the Algorithm 1 to find the corresponding landmarks $\mathcal{L}'$ for $\mathcal{T}$ and then we employ an annealing style of the deformation schedule to vary the weights, k_s, k_b, k_m, and k_c, so that the generic facial model is gradually and smoothly deformed towards the target. At present we use 10 iterations to deform the generic facial model towards the individual facial model. At each iteration, we use the deformed model from the last iteration as input, and solve the linear equations.

At the initial stage, we set $k_s = 1.0$, $k_b = 1.0 \times 10^{-9}$, $k_m = 100$, $k_c = 0$. Since the two models might not be well aligned at the initial stage, the closest point term is meaningless. We use the corresponding landmarks to align the two facial models first, and the initial values of k_s and k_b ensure the deformation is rigid.

After initial stage, we iteratively vary the weights: k_s from 1.0 to 1.0×10^{-3}, k_b from 1.0×10^{-9} to 1.0×10^{-12}, k_m from 100 to 1000, k_c from 0.0 to 3000. We currently use a linear function for decreasing and increasing the weights, but many other different functions may be used.

After finishing these iterations, we obtained a deformed model from the generic model. The deformed model has the same topology structure as the generic model but in the shape of the individual facial model. Thus, the dense correspondences between the two facial models is trivial. We use the deformed facial model as the base mesh. For each vertex on the individual facial model, we find its corresponding closest surface point on the deformed model, and use the triangle and the barycentric coordinates for its parameterization on the generic facial model.

3.3. Expression Transferring

In this section, we are try to transfer facial expression from our generic facial model onto the deformed model (which is deformed from the generic through the method described in Section 3.2) and individual facial models. The central idea is to encode the facial expression as the difference between the neutral facial model and the expression facial model, and then to appropriately apply this difference to another individual neutral facial model. However, directly applying the facial expression encoded as vertex displacements to the individual face may result in an unrealistic face with artefacts because of different shape proportions. To resolve this problem, we employ the deformation transfer method [SP04] to encode the facial expression as a collection of affine transformations tabulated for each corresponding triangle.

For the generic neutral facial model $\mathcal{S}$ with N vertices $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$ and M triangles $\{t_1, t_2, \ldots, t_M\}$, and the ex-

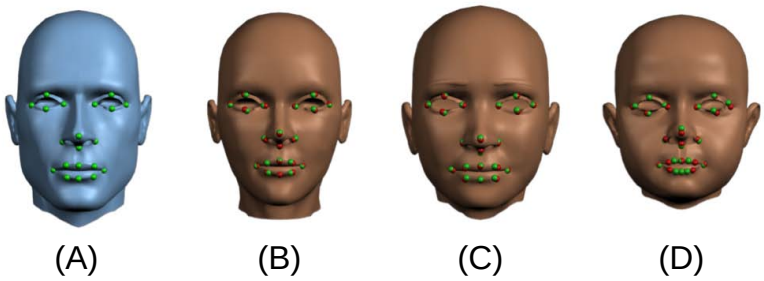

Figure 4: *(A): The generic facial model with 20 specified feature landmarks; (B), (C), (D): The comparison between the landmarks (green points) computed by the landmark detection algorithm and those landmarks (red points) manually selected by hand.*

pression facial model $\mathcal{S}'$, the facial expression of $\mathcal{S}'$ is represented as a collection of affine transformations for each corresponding triangle between $\mathcal{S}$ and $\mathcal{S}'$

$$E_{\mathrm{g}} = \{\mathbf{T}_i^{\mathrm{g}} | \mathbf{v}'_j = \mathbf{T}_i^{\mathrm{g}}\mathbf{v}_j + \mathbf{d}_i,\ \mathbf{v}_j \in t_i,\ \mathbf{v}'_j \in t'_i,\ j = 1,2,3,\ i = 1,2,\ldots,M\}, \quad (6)$$

where $\mathbf{v}_j$ and $\mathbf{v}'_j$ are the three vertices of the i-th triangle in $\mathcal{S}$ and $\mathcal{S}'$ respectively, $\mathbf{T}_i^{\mathrm{g}}$ is the affine transformation between the two corresponding i-th triangles, and $\mathbf{d}_i$ is the displacement vector containing only the translational portion. The non-translation portion $\mathbf{T}_i^{\mathrm{g}} \in \mathbb{R}^{3\times 3}$ encodes the changes in orientation, scale, and skew induced by the deformation on the triangle.

Since the three vertices $\{\mathbf{v}_{i1}, \mathbf{v}_{i2}, \mathbf{v}_{i3}\}$ of a triangle t_i before and after deformation do not fully determine the affine transformation $\mathbf{T}_i^{\mathrm{g}}$, a fourth vertex $\mathbf{v}_{i4}$ is introduced for each triangle as,

$$\mathbf{v}_{i4} = \mathbf{v}_{i1} + \frac{(\mathbf{v}_{i2} - \mathbf{v}_{i1}) \times (\mathbf{v}_{i3} - \mathbf{v}_{i1})}{\sqrt{|(\mathbf{v}_{i2} - \mathbf{v}_{i1}) \times (\mathbf{v}_{i3} - \mathbf{v}_{i1}|}},\ i = 1,2,...,M. \quad (7)$$

Then $\mathbf{T}_i^{\mathbf{g}}$ is given by

$$\mathbf{T}_i^{\mathbf{g}} = \mathbf{V}'_i \mathbf{V}_i^{-1} \quad (8)$$

with

$$\mathbf{V}_i = [\mathbf{v}_{i2} - \mathbf{v}_{i1}\ \ \mathbf{v}_{i3} - \mathbf{v}_{i1}\ \ \mathbf{v}_{i4} - \mathbf{v}_{i1}]$$
$$\mathbf{V}'_i = [\mathbf{v}'_{i2} - \mathbf{v}'_{i1}\ \ \mathbf{v}'_{i3} - \mathbf{v}'_{i1}\ \ \mathbf{v}'_{i4} - \mathbf{v}'_{i1}]$$

For a given individual facial model $\mathcal{T}$, the goal is to build a new facial model $\mathcal{T}'$ with the same expression as $\mathcal{S}'$. The expression $\mathbf{E}_{\mathrm{f}}$ on $\mathcal{T}'$ can be represented in the similar way as described above, while at this time $\mathbf{v}'_j$ represents the unknown vertices to be solved. So the problem becomes to match E_{f} to E_{g} as well as possible while maintaining consistency for shared vertices of each triangle, that is,

$$\min \|E_{\mathrm{g}} - E_{\mathrm{f}}\|_F^2 = \min \sum_{i=1}^{M} \left\| \mathbf{T}_i^{\mathrm{g}} - \mathbf{T}_i^{\mathrm{f}} \right\|_F^2. \quad (9)$$

The matrix norm $\|\cdot\|$ is the Frobenius norm, or the square root of the sum of the squared matrix elements. A solution of this optimization problem defines the desired expression on the individual face. This optimization problem can be solved by reformulating the problem in a system of linear equations, which contains a large sparse matrix. For efficiency, we use a sparse matrix solver with Cholesky factorization [Tol].

After transferring the facial expressions from the generic facial model to the deformed facial model, we obtain a series of facial expressions on the deformed facial model. Each of these transferred expression facial models has the same topology from the generic facial model, but the shape is a blend of the shape of the original expression model and the shape of the individual model. With the dense correspondences built by the method in the previous section, we can also easily obtain the facial expressions on the individual facial model by interpolating expression displacements on the deformed facial models.

4. Results

According to the MPEG-4 specification on feature points of human faces [LP99], we specified twenty prominent feature landmarks on our generic facial model. Figure 4 shows the results of automatically detecting the corresponding landmarks on three individual facial models (green points). To compare the results, we also show the user selected landmarks (red points) for each individual facial models. The distances between the computed and the user selected landmarks for Figure 4 (A) and (D) are shown in Figure 5 (A) and (B), respectively.

The results show that our landmark detection algorithm can precisely detect most corresponding landmarks on the individual facial models. Most distances between the computed landmarks and the user-selected landmarks are below $8\% \times w_{\mathrm{eye}}$ (Figure 5). However, if the shapes between the generic facial model and the individual facial model differ too much, for example, the lip in Figure 4(D), the distance

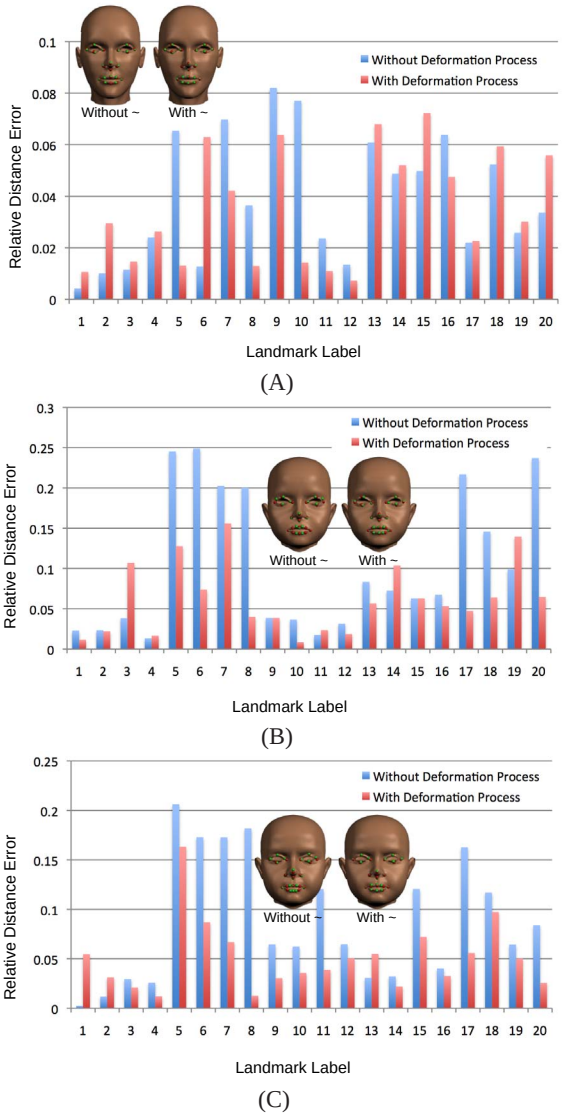

Figure 5: *The comparison between the results obtained by the automated landmark detection algorithm with and without an additional deformation step. The distance error, measured in the unit of the average eye width w_{eye} of the generic model, is the relative distance between each computed landmark and its corresponding landmark selected by hand. Figures (B), (C) show that the additional deformation step can improve the detected results.*

error would rise. Thus, we did the tests of combining the landmark detection with the deformable model to improve the detection results. The procedure is that, after detecting the landmarks, we deform the generic facial model using the algorithm described in Section 3.2, and then use the detection algorithm again to find the landmarks for each one on the deformed model which has the similar shape as the individual facial model. Figure 5 shows the comparison between the results obtained by the automated landmark detection algorithm both with and without an additional deformation step. From the results, we can see that the additional deformation step can significantly improve those landmarks that are not well matched.

For each individual facial model, we established the dense correspondences by deforming the generic facial model towards the individual facial model and use the deformed model as the base mesh to parameterize the individual facial model. Figure 6 shows the results of the dense correspondences established by our method.

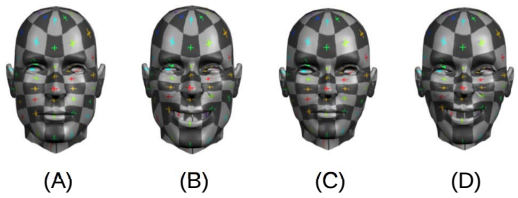

Figure 6: *(A) The chessboard texture rendering results of the generic model (A), the expression model (B), the target model (C) and the target expression model (D), to show the correspondences established by our method.*

As shown in Figure 7, we transferred six facial expressions onto four individual facial models. The number of vertices and triangles for each facial model are listed in Table 1 in the same order.

Table 1: *The topology information of the facial models*

Model Name	Number of Vertices	Number of Triangles
Generic	3861	7678
Boy	2674	5304
Girl	1773	3516
Judy	3861	7678
Penny	3861	7678

The time cost for transferring one facial expression onto an individual facial model depends on the number of vertices of the two models. In our example, the whole process takes about 2–3 minutes for one individual model, timing results on 2.2GHz Intel Core 2 Duo computer.

5. Conclusions

The work reported and discussed in this paper shows that the improved automated landmark detection algorithm can correctly find most corresponding landmarks on the individual facial models with distance errors below 8%w_{eye}. Combined with the deformable model, we can significantly improve

those landmarks that are not well matched and are able to establish fairly good correspondences between facial models and that facial expressions can be automatically transferred to different facial models. The results are currently being evaluated in detail with the aim of further improving the automated transfer of facial expressions through hybrid models.

6. Acknowledgments

This work is supported by a Robert Gordon University Research Development Initiative (RDI) PhD Studentship.

References

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (2003), pp. 587–594.

[Ale02] ALEXA M.: Recent advances in mesh morphing. *Computer Graphics Forum 21*, 2 (2002), 173–198.

[ASP*05] ANGUELOV D., SRINIVASAN P., PANG H., KOLLER D., THRUN S., DAVIS J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in Neural Information Processing Systems* (2005), p. 33.

[BPK*07] BOTSCH M., PAULY M., KOBBELT L., ALLIEZ P., LÉVY B., BISCHOFF S., RÖSSL C.: Geometric modeling based on polygonal meshes. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses* (2007).

[BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 213–230.

[BSPG06] BOTSCH M., SUMNER R., PAULY M., GROSS M.: Deformation transfer for detail-preserving surface editing. In *Vision, Modeling & Visualization* (2006), pp. 257–364.

[GM97] GIBSON S. F. F., MIRTICH B.: *A survey of deformable modeling in computer graphics*. Tech. rep., Mitsubishi Electric Research Laboratories, 1997.

[GMGP05] GELFAND N., MITRA N. J., GUIBAS L. J., POTTMANN H.: Robust global registration. In *SGP '05: Proceedings of the Third Eurographics Symposium on Geometry Processing* (2005), p. 197.

[HLS07] HORMANN K., LÉVY B., SHEFFER A.: Mesh parameterization: theory and practice. In *SIGGRAPH '07: ACM SIGGRAPH 2007 Course Notes* (2007).

[HT04] HABER J., TERZOPOULOS D.: Facial modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes* (2004), p. 6.

[LP99] LAVAGETTO F., POCKAJ R.: The facial animation engine: Toward a high-level interface for the desgin of MPEG-4 compliant animated faces. *IEEE Transactions on Circuits and Systems for Video Technology 9*, 2 (1999), 277–289.

[MMN07] MOROOKA K., MATSUI S., NAGAHASHI H.: Self-organizing deformable model for mapping 3d object model onto arbitrary target surface. In *3DIM '07: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling* (2007), pp. 193–200.

[NN01] NOH J., NEUMANN U.: Expression cloning. In *SIGGRAPH '01: ACM SIGGRAPH 2001 Papers* (2001).

[Pig06] PIGHIN F.: Performance-driven facial animation. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Course Notes* (2006).

[PKC*03] PYUN H., KIM Y., CHAE W., KANG H. W., SHIN S. Y.: An example-based approach for facial expression cloning. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 167–176.

[PMW05] PLANTIZ B. M., MAEDER A. J., WILLIAMS J. A.: The correspondence framework for 3D surface matching algorithms. *Computer Vision and Image Understanding 97*, 3 (2005), 347–383.

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling* (2001), pp. 145–152.

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (2004), pp. 399–405.

[SSB07] STEINKE F., SCHÖLKOPF B., BLANZ V.: Learning dense 3D correspondence. In *Advances in Neural Information Processing Systems* (2007), pp. 1313–1320.

[SSK*05] SURAZHSKY V., SURAZHSKY T., KIRSANOV D., GORTLER S. J., HOPPE H.: Fast exact and approximate geodesics on meshes. In *SIGGRAPH '07: ACM SIGGRAPH 2007 Papers* (2005), pp. 553–560.

[Tol] TOLEDO S.: A library of sparse linear solvers, version 2.2. http://www.tau.ac.il/~stoledo/taucs.

[VBP05] VLASIC D., BRAND M., POPOVIĆ H. P. J.: Face transfer with multilinear models. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (2005), pp. 426–433.

Figure 7: *The results obtained by our facial expression transferring method. In the first row are the generic model and those different facial expressions. The following rows are the individual models and their corresponding facial expressions transferred from the generic model.*