EUROGRAPHICS 2004 / M.-P. Cani and M. Slater
(Guest Editors)

*Volume 23 (2004), Number 3*

# Hierarchical Retargetting of Fine Facial Motions

Kyunggun Na and Moonryul Jung

Department of Media Technology, Graduate School of Media Communications
Sogang University, Seoul, Korea
e-mail : gun@sogang.ac.kr, moon@sogang.ac.kr

**Abstract**

*We present a novel technique for retargetting captured facial animation to new facial models. We use dense motion data that can express fine motions such as wrinkles. We use a normal mesh, which is a special multi-resolution mesh, to represent source and target models. A normal mesh is defined by the base mesh and sequence of normal offsets from it. Our retargetting consists of two steps: base mesh and detail mesh retargetting. For base mesh retargetting, we use an example-based technique to take advantage of the intuition of the user in specifying the similarity between source and target expressions. In detail mesh retargetting, the variations of normal offsets in the source mesh are hierarchically computed and transferred to the target mesh. Our retargetting scheme is able to produce robust and delicate results for unusual target models from highly detailed motion data.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism - Animation; I.3.5 [Computer Graphics]: Computational Geometry and Object Modelling - hierarchy and geometric transformation, object hierarchy

## 1. Introduction

3D animation by motion capture is now commonplace. To make captured motions reusable, retargetting motions to new bodies were first developed by Gleicher [Gle98], and extended by Lee [LS99], and Choi [CK00]. There are also some work [JT03, BBV03, PS03, NN01] in facial motion retargetting. Noh et al. [NN01] proposed a retargetting method which transfers facial motions of human to unusual facial models. Pyun et al. [PS03] developed an example-based retargetting for more stable and accurate results.

Here we consider using fine facial motion data and retargetting them to new facial models. Fine motions play an important role in making facial animation realistic, and are very difficult for animators to create. The problem of retargetting highly detailed facial motion to different facial models has not been addressed. The goal of this paper is to construct a robust retargetting system that can transfer highly detailed facial motion to nonhuman target models.

The retargetting method of Noh et al. [NN01] consists of two steps. First, it computes the displacement or motion of each vertex on the source mesh at each frame from the neutral frame. Secondly, it computes the scaled version of the displacements and applies them to the neutral frame of the target mesh. The displacements are measured with respect to the local coordinate systems defined on the neutral frames of the source and target meshes. This method works well only when shapes of the source and target are similar or the displacements of vertices from the neutral frame are small. A more stable method is needed to overcome this restrictions.

Pyun et al. [PS03] proposed an example based retargetting scheme. In this approach, the system uses a set of correspondences between example expressions of the source and target facial models provided by the animator. This method produces more stable and accurate results than that of Noh et al. [NN01], because of sufficient mapping data between source and target expressions. In general, example-based approaches [JT03, BBV03, PS03] are attractive because of stable and accurate outcomes. However, the more detailed the facial expressions are, the more examples are required and the more effort are needed to craft them. Hence, it is difficult to use an example-based approach to retarget highly detailed facial motions.

To take advantage of an example-based approach and to handle highly detailed facial motions as well, we present a hierarchical retargetting system. Our approach is to represent

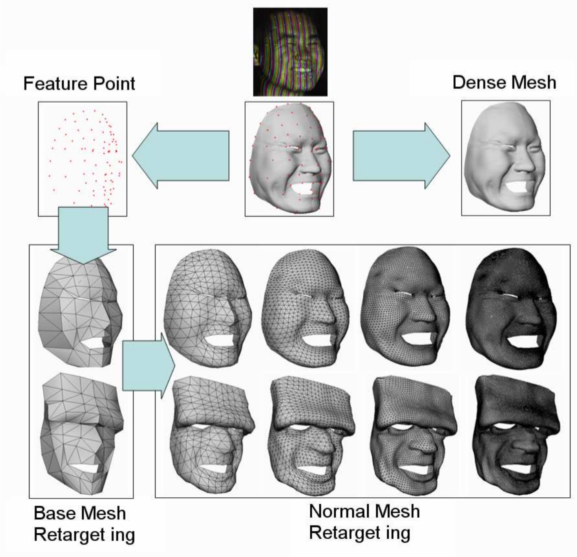

**Figure 1:** *Hierarchical retargetting system overview: (1) At each frame, the shape of the source base mesh (human) is first mapped to the target base mesh (Frankenstein) by an example-based approach. (2) The details of the source normal mesh at each level are then transferred to the target normal mesh level by level.*

a facial motion as a sum of the base motion and a hierarchy of fine motions. Retargetting is performed in two steps: base mesh retargetting and detail mesh retargetting, as shown in Figure 1.

We use an example-based approach to retarget the base motion. Our retargetting requires only a few example expressions of the source and target models at base level. Each point on base mesh has its mapping data, which are extracted from example expressions of source and target. Retargetting of each base point is performed by scattered data interpolation.

We use a normal mesh [GVS*00] to represent a hierarchy of fine motions. A normal mesh, one of the remeshing methods [GSS99, Hop96, KCV*98], is a multi-resolution mesh that is specified by the base mesh and a sequence of normal offsets from it. For detail mesh retargetting, we represent the motion of the source mesh as a sequence of normal offset difference from the neutral frame. Then we apply the normal offset difference to the target mesh level by level.

Compared to Noh et al. [NN01], our method is more robust when motion of the source model (from the neutral frame) is large and the target model is very different from the source. This is because we use an example-based approach for base mesh retargetting and a hierarchical approach for detail mesh retargetting. Compared to Pyun [PS03], our method needs less example expressions because an example-based approach is only used for base motion. However, it can produce good results for highly detailed motion because of hierarchical detail mesh retargetting process, as shown in Figure 2.

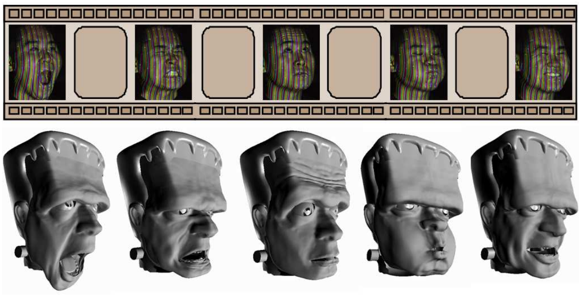

**Figure 2:** *Detailed expressions, e.g. wrinkles, are captured by a vision technique (top) and retargetted to Frankenstein (bottom)*

## 2. Fine facial motion data

In our experiment, we use fine facial motion data which are generated by a structured lighting method [JLP04, MV96]; a vision-based technique that recovers 3D shapes from stripe images made by camera and projector. The camera is the Sony XC-003 3-CCD color camera, and the projector is an Epson EMP-7700. Images are obtained at 30 frames per second. The maximum resolution of the resulting 3D shape can be as fine as 640*480. However, source meshes generated by this method at each frame don't have an identical topology and inter-frame correspondence, since all points on the face cannot be tracked. Only 78 white dots(feature points) that are drawn on the actor's face are tracked to determine frame-to-frame correspondence.

Figure 3 shows a stripe image of the actor's face and the 3D mesh recovered from the image. However, as the capture system is not yet so mature, there were some manual works involved in editing raw dense meshes. 3D meshes generated from stripe images by a structured lighting method [JLP04] are terrain meshes since the depths of each pixel in the stripe image correspond to the heights of vertices on the mesh. The boundaries of a terrian mesh are fuzzy. So, we manually trimmed the boundaries of the face, the eyes, and the mouth. We also cut off the other side of the face because only one side of the face is well captured. This is then duplicated and the other side obtained by mirror reflection and the two sides are joined. Since the original source mesh is noisy, we smoothed it with denoising method [FDC03]. Our retargetting system doesn't necessarily need data obtained by a structured lighting method. Any other motion data can be used if it is dense enough.

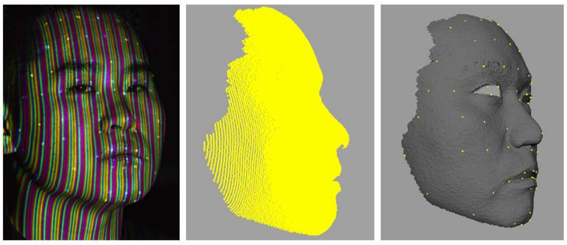

**Figure 3:** *A snapshot of stripe images(left) and a 3D mesh recovered from the image(right): The dots are drawn on the actor's face to specify feature points and are used to determine an interframe correspondence of the source mesh.*

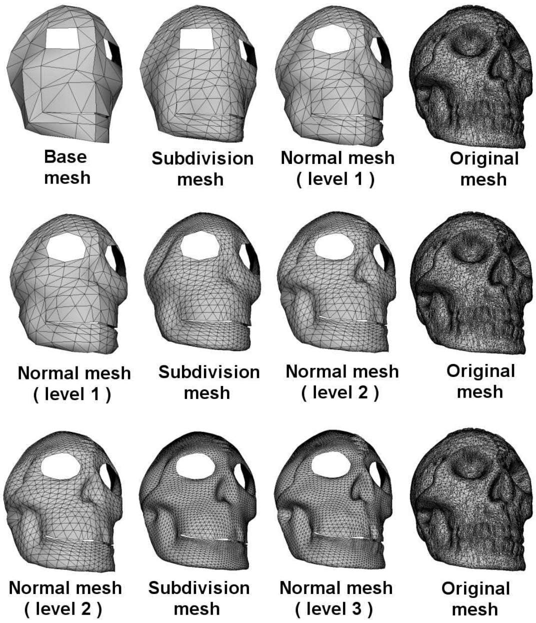

**Figure 4:** *The construction of a normal mesh level by level: a normal mesh starts from the base mesh, and the normal mesh at a level is generated by computing the normal offsets from the subdivision mesh of the previous level to the original mesh.*

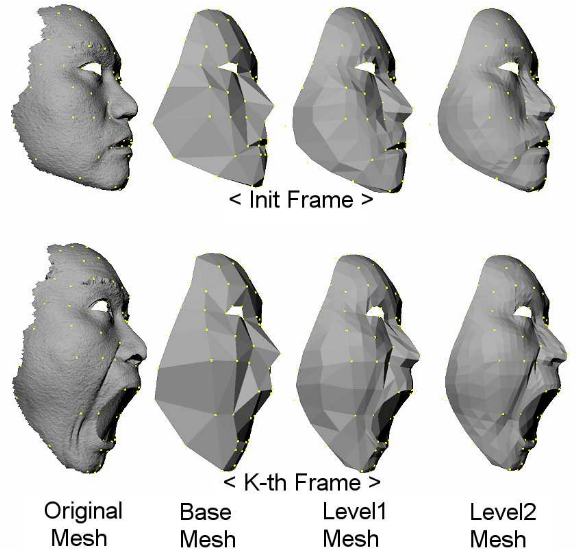

**Figure 5:** *The establishment of a temporal correspondence level by level*

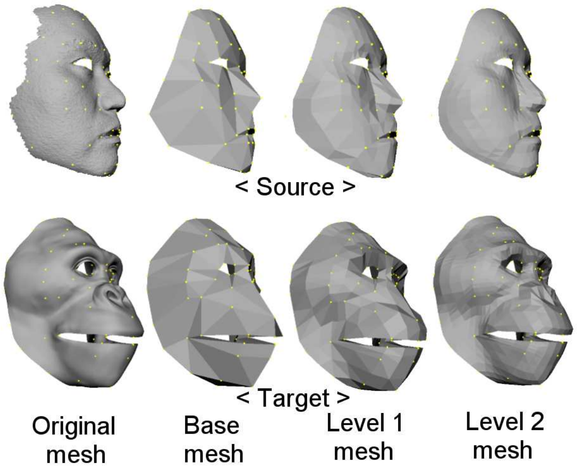

**Figure 6:** *The establishment of a spatial correspondence level by level*

## 3. Dense correspondence

Source meshes at each frame don't have temporal correspondence. In addition, source mesh and target mesh don't have spatial correspondence as they have different mesh structures. We can, however, establish temporal and spatial correspondences in a hierarchical manner by taking advantages of the normal mesh representation. A normal mesh starts from the base mesh, and an intermediate level of the normal mesh is defined by the normal offsets from the subdivision mesh of the previous level to the original mesh as shown in Figure 4. We use loop scheme in subdivision process for its smooth surface. In this paper, an intermediate level of the normal mesh whose level is $n$ is called *the normal mesh at level n*.

### 3.1. Temporal correspondence

The source meshes at each frame have only a partial correspondence by the tracked feature points. In our experiments, we manually construct the base mesh by connecting feature vertices. The source base meshes at each frame have the same topology. Hence, the intermediate level meshes of the source mesh at each frame have the same topology. So, a temporal correspondence is determined hierarchically by the intermediate level meshes, as shown in Figure 5.

### 3.2. Spatial correspondence

For motion retargetting, it is crucial to find a spatial correspondence between the source and target meshes. In our system, the user specifies feature points on the target mesh corresponding to the feature points on the source mesh, as shown in Figure 6. This process reflects the user's intuition and determines the partial correspondence between the source and target base meshes. Once the feature points are specified on the source and target meshes, normal meshes are generated from the source and target base meshes. The intermediate level meshes of the source and target meshes have the same topology and so a hierarchical spatial correspondence is established.

## 4. Hierarchical retargetting

### 4.1. Base mesh retargetting

Above all, facial retargetting must guarantee the similarity of expression between the source and target models. The approach of Noh et al. [NN01] constructs a retargetting mapping $F$ by using only one correspondence between the neutral expressions of the source and target models, as shown in Equation 1.

$$\begin{aligned} S_{neut} \Leftrightarrow T_{neut} :& \\ \text{Find } F \text{ such that } T_{new} - T_{neut} = F(S_{new} - S_{neut})& \end{aligned} \tag{1}$$

Here $S_{neut}$ and $T_{neut}$ denote the neutral expressions (faces) of the source and target models. $S_{neut} \Leftrightarrow T_{neut}$ denotes the user-given correspondences between $S_{neut}$ and $T_{neut}$. In contrast, Pyun et.al [PS03] constructs a retargetting mapping $F$ by using a set of correspondences between multiple pairs of example expressions, as shown in Equation 2.

$$\begin{aligned} T_1 \Leftrightarrow S_1, \ \ldots, T_M \Leftrightarrow S_M :& \\ \text{Find } F \text{ such that } T_{new} - T_{neut} = F(S_{new} - S_{neut})& \end{aligned} \tag{2}$$

This approach works well for even very different target models. But as the details of expressions increase, more examples are required to construct a retargetting mapping $F$. For instance, if the user wants to retarget specially shaped wrinkles or muscles to the target model, new example expressions must be crafted manually. However, in our system, we use an example-based approach only for the base mesh, because details are retargetted by other means. Moreover, base points are so sparse that the movement of each base point does not affect the movement of other base points so much. So, we retarget the motion of each base point independently of other base points. This idea is motivated by action units (AU), in FACS [PW96]. It can generate rich expressions by the combination of each AU parameter, which are assumed to move independently of one another. This makes it possible to reduce burdens for preparing examples, since combinations of each local components can generate rich expressions. Consequently, we use only 7 example expression pairs of source and target base mesh for an example-based retargetting, as shown in Figure 7.

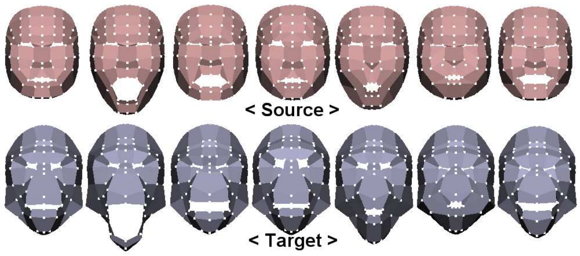

**Figure 7:** *Seven example source and target base meshes are used for base mesh retargetting.*

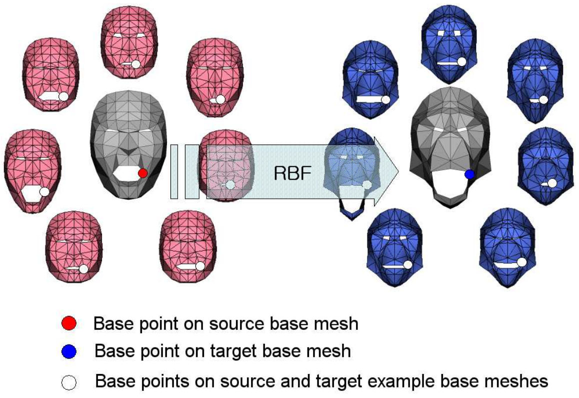
RBF
● Base point on source base mesh
● Base point on target base mesh
○ Base points on source and target example base meshes

**Figure 8:** *The function (RBF) from each base point on the source base mesh to the corresponding base point on the target base mesh: It is constructed from the correspondences between the base points on the example source meshes and those on the example target meshes.*

For base mesh retargetting, we need a retargetting map to transform each base point on the source base mesh to the corresponding base point on the target base mesh. We construct the retargetting map by using scattered data interpolation over a set of correspondences between the base points on the example expressions (faces) of the source and target models. We use a radial basis function to construct the retargetting map. Figure 8 shows how a base point on the source base mesh is mapped to the corresponding base point on the target base mesh.

Let $S_{new}$ be the source base mesh whose motion is to be retargetted and $T_{new}$ be the target base mesh to be created.

Let $S_j$, $T_j$ $(j = 1, M)$ be the example base meshes of source and target, where M is the number of examples. Let $[i]$ be the $i$-th base point on the base mesh. Let $S_{neut}$ and $T_{neut}$ be the base meshes with neutral expression of the source and target models. Then $T_{new}[i]$ is defined as follows:

$$T_{new}(i) - T_{neut}[i] = \sum_{j=1}^{M} \vec{w}_j[i] \;\; h(\, S_{new}[i] - S_{neut}[i] \,) \qquad (3)$$

Here $h$ is a multi-quadric radial basis function [Har71] and $\vec{w}_j[i]$ is a vector of weight coefficients, whose dimension is three. $h$ can be described by $h(x) = \sqrt{x^2 + c^2}$. A locality parameter $c$ is used to control the behavior of the basis function and how it is influenced by neighboring control-points.

The $i$-th base point of the $k$-th target example face, $T_k[i]$ correspond to the $i$-th base points of the $k$-th source example faces, $S_k[i]$. Equation (3) should be satisfied for $S_k[i]$ and $T_k[i]$ $(\, k = 1, M \,)$. Hence, by substituting $S_k[i]$ and $T_k[i]$ for $S_{new}[i]$ and $T_{new}[i]$ respectively in Equation (3), we have:

$$T_k[i] - T_{neut}[i] = \sum_{j=1}^{M} \vec{w}_j[i] \; h(S_k[i] - S_{neut}[i]) \;\; (k = 1, M) \quad (4)$$

The weights $\vec{w}_j[i]$ $\;(j = 1, M)$ for Equation 3 are computed by solving the linear system defined by Equation 4.

### 4.2. Detail mesh retargetting

Now suppose that the target base mesh at the current frame has been obtained by example-based retargetting. The source mesh at the current frame has detail meshes on top of the base mesh, because the source mesh is represented as normal mesh. But the target mesh at the current frame does not. The detail meshes on top of the target base mesh at the current frame should be created. For this, we first compute the normal offsets difference of the source mesh between the neutral frame and the current frame. Then, we add the normal offsets difference to the normal offsets of the neutral frame of the target mesh. Figure 9 illustrates the simple process for normal mesh retargetting.

Suppose that the initial frame, frame 0, is the neutral face of the source model. The source normal offsets $SN_0[1]$ and the target normal offsets $TN_0[1]$ of level 1 at frame 0 are defined as follows:

$$\begin{aligned} SN_0[1] &= S_0[1] - sub(S_0[0]) \\ TN_0[1] &= T_0[1] - sub(T_0[0]) \end{aligned} \qquad (5)$$

where $SN$, $TN$ are normal offsets of the source and target meshes and their values are scalar. $S_0[1]$, $S_0[0]$, are the source normal mesh at level 1 and at level 0. Similarly with $T_0[1]$, $T_0[0]$. $sub()$ denotes the subdivision process. The normal mesh at level 0 is the base mesh. The normal offsets of the source mesh at level $i$ at the frame $k$ are defined as

$$SN_k[i] = S_k[i] - sub(S_k[i-1]). \qquad (6)$$

The normal offset differences of the source mesh at level $i$ at frame $k$ are measured from the initial frame and obtained as follows:

$$\triangle D_k[i] = SN_k[i] - SN_0[i] \qquad (7)$$

The normal offset differences are applied to the normal offsets of the target mesh at frame 0 level by level, to obtain the target normal mesh at frame $k$, as follows:

$$\begin{aligned} T_k[1] &= sub(T_k[0]) + (TN_0[1] + \delta \triangle D_k[1]) \\ T_k[2] &= sub(T_k[1]) + (TN_0[2] + \delta \triangle D_k[2]) \\ T_k[3] &= sub(T_k[2]) + (TN_0[3] + \delta \triangle D_k[3]) \\ &\ldots\ldots\ldots\ldots\ldots\ldots \end{aligned} \qquad (8)$$

$\delta$ is a global scale factor, which is defined as follows:

$$\delta = \sum f_T \,/ \sum f_S \qquad (9)$$

where $\sum f_S$, $\sum f_T$ are the total surface areas of the source and target mesh, respectively. Figure 10 shows the process in which the details of the source mesh at the current frame is transferred to the target mesh at the current frame level by level, starting from the base mesh.

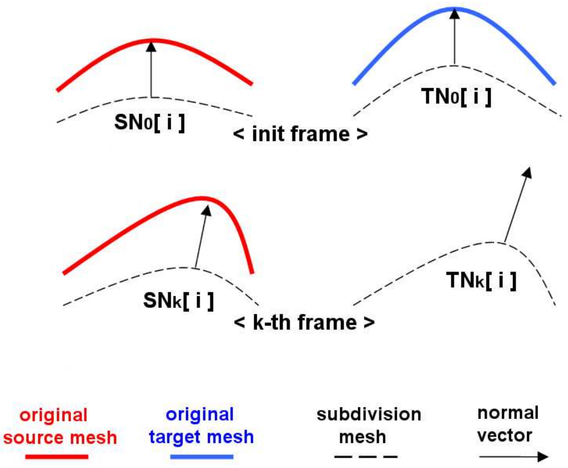

**Figure 9:** *A normal offset of the target mesh at level i at frame k, $TN_k[i]$, is obtained by adding the normal offset difference of the source mesh, $SN_k[i] - SN_0[i]$, to the normal offset of the target mesh at the neutral frame, $TN_0[i]$.*

### 4.3. Using normal map

So far we have described how the motion of the source normal mesh is hierarchically transferred to the target normal mesh. The higher level normal meshes are, the more detailed result we have. However, the high level mesh with fine motions is very expensive for animation and rendering. To express the details at low level, we use normal map. We get normal maps from high level mesh at each frame and apply them to low level mesh. This is analogous to place texture map on a coarse polygonal mesh, as shown in Figure 11.

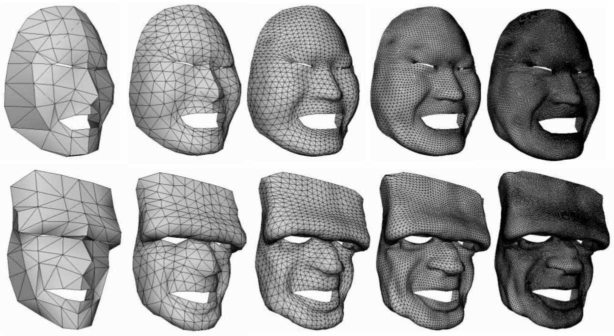

**Figure 10:** *Detail mesh retargetting: The details of the source normal mesh (top) are transferred to the target normal mesh (below) level by level and the details become to appear.*

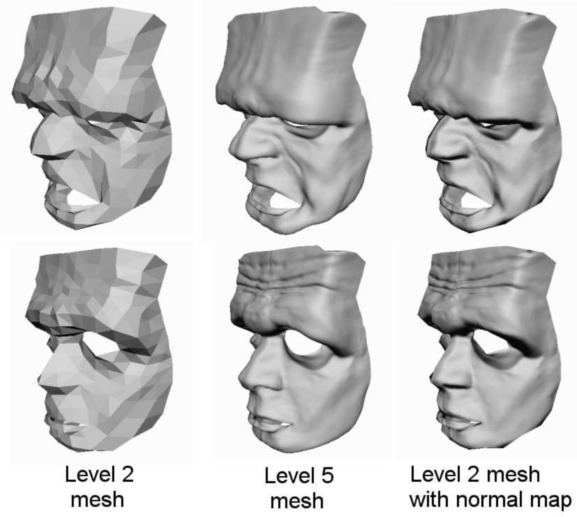

**Figure 11:** ***Top**: Expression 1. (left) The target normal mesh at level 2. (center) The target normal mesh at level 5. (right) The target normal mesh at level 2 rendered with the normal map, which is made from normal mesh at level 5. Its surface detail is similar to that of the center figure, but it is more efficient for rendering. **Bottom**: Expression 2. The same with Top.*

This technique is not for motion retargetting, but is only for rendering the retargetted motion.

### 4.4. Treatment of special cases

#### The normal offsets of boundary vertices

We subdivide normal meshes with preserving the properties of the boundary edges in subdivision process. Our facial meshes have boundaries at the mouth or eyes or border of face, and normal vectors of boundary points on subdivision mesh may not pass the boundaries of original mesh. In this case, we cannot compute the normal offsets. As indicated in Figure 12, we compute the displacement vectors, which start from the boundary points of the subdivision mesh and are perpendicular with boundary edges of the original mesh. For boundary points of target meshes at each frame, the displacement vectors of target mesh at initial frame are used instead of the normal offsets.

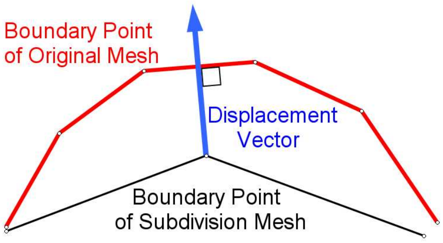

**Figure 12:** *For boundary vertices of the subdivision mesh at some level, we use the displacement vectors.*

#### Piercing error

As described in Guskov [GVS*00], the process of constructing the normal mesh from the original mesh is not clear. When piercing normal vectors into the original mesh, there may be intersection points that are not correct. These cases are called piercing errors. In our experiments, normal meshes up to level 4 or 5 for the source and target meshes have the piercing errors, as shown in Table 1.

| mesh name | vertices | vertices with piercing error | % |
|---|---|---|---|
| source | 69,246 | 10 ( average of all frames ) | 0.0144 % |
| gorilla | 69,246 | 24 ( init frame ) | 0.0346 % |
| ox | 17,470 | 6 ( init frame ) | 0.0343 % |
| Frankenstein | 17,470 | 4 ( init frame ) | 0.0228 % |
| skull | 17,470 | 14 ( init frame ) | 0.0801 % |

**Table 1:** *The piercing error*

If some pierced point is problematic, we solve the problem by replacing the point with the average coordinate of the neighboring vertices instead of applying the problematic normal offset.

## 5. Experimental results

We retargetted a motion of the actor's face to gorilla, ox, Frankenstein, and skull models. These target meshes were crafted by the View Point corporation. The experiments are performed on a 2.4 GHz Pentium-4 PC. Our retargetting system spends quite a time in generating normal meshes for temporal and spatial correspondence. It takes $O(NM)$ to find normal offsets from N vertices of the subdivision mesh to M triangles of the original mesh. Figure 13 shows snapshots of the results of our retargetting. The gorilla mesh is represented as a normal mesh with 5 levels. The rest of the meshes have 4 levels. Figure 14 shows how the present system can retarget wrinkles of the human actor to the gorilla model. For an animation, see the supplementary video

## 6. Conclusion and future work

In this paper, we presented a novel and effective method to retarget highly dense facial motion to new target faces. The results of our retargetting experiments demonstrate that fine details of the source motion are transferred to the target model without visible distortion. The basic idea of the method is to represent the source and target meshes as a sum of the base mesh and the detail mesh. We represent the source and target meshes as a normal mesh which represents the detail mesh as a hierarchy of normal offsets on top of the base mesh. Given a source motion, our method transforms the source base mesh at each frame to the corresponding target base mesh. The detail mesh of the target mesh is obtained from the motion of the source detail mesh.

To transform the source base mesh to the target base mesh, we used an example-based approach. The idea is that the global motion of the target base model should be constructed based on the user-supplied correspondences between example expressions of the source and target base models. In the case of detail motion, the system takes the detail motion of the source mesh and scales it and adds it onto the target base mesh at the initial frame. Base mesh retargetting via an example-based approach requires some labor, although it gives good results. It is worthwhile, however, because the results of detail mesh retargetting looks good only when base mesh retargetting is good enough. Also, the approach is not too expensive, as the base mesh retargetting via an example-based approach requires only 7 example source and target expressions.

It turns out that mesh representation is critically related to motion retargetting. Our method also depends on normal mesh representation. Continuing this line of investigation, we plan to extend our retargetting method so that it represents the motion of the source mesh in terms of the motion of feature curves, e.g. crease lines and ridge curves, and transfers them onto the target mesh.

## References

[BBV03] Blanz V., Basso C., Vetter T.: Reanimating faces in images and video. In *Eurographics 03* (2003). 1

[CK00] Choi K., Ko H.: On-line motion retargeting. In *Journal of Visualization and Computer Animation* (2000), pp. 223–243. 1

[FDC03] Fleishman S., Drori I., Cohen-Or D.: Bilateral mesh denoising. In *SIGGRAPH 03* (2003), pp. 950–953. 2

[Gle98] Gleicher M.: Retargetting motion to new characters. In *SIGGRAPH 98* (1998), pp. 33–42. 1

[GSS99] Guskov I., Sweldens W., Schröder P.: Multi-resolution signal processing for meshes. In *SIGGRAPH 99* (1999). 2

[GVS*00] Guskov I., Vidimce K., Sweldens W., Schröder P.: Normal meshes. In *SIGGRAPH 00* (2000). 2, 6

[Har71] Hardy R.: Multiquadric equations of topography and other irregular surfaces. In *J. of Geophysical Research* (1971), vol. 76(8), pp. 1905–1915. 5

[Hop96] Hoppe H.: Progressive meshes. In *SIGGRAPH 96* (1996), pp. 99–108. 2

[JLP04] Je C., Lee S. W., Park R.-H.: High-contrast color-stripe pattern for rapid structured-light range imaging. In *European Conference on Computer Vision (ECCV 2004)* (2004). 2

[JT03] Joshi P., Tien W. C.: Learning controls for blend shape based realistic facial animation. In *Eurographics 03* (2003), pp. 187–192. 1

[KCV*98] Kobbelt L., Campagna S., Vorsatz J., Seidel H. P.: Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH 98* (1998), pp. 106–114. 2

[LS99] Lee J., Shin S.: A hierarchical approach to interactive motion editing for human-like figures. In *SIGGRAPH 99* (1999), pp. 39–48. 1

[MV96] McIvor A. M., Valkenburg R. J.: Calibrating a structured light system. In *Image and Vision Computing* (1996), Industrial Research Limited, pp. 167–172. 2

[NN01] Noh J., Neumann U.: Expression cloning. In *SIGGRAPH01* (2001), pp. 271–288. 1, 2, 4

[PS03] Pyun H., Shin S. Y.: An example-based approach for facial expression cloning. In *ACM SIGGRAPH/Eurographics Symposium on Comp. Aniamtion* (2003), pp. 167–176. 1, 2, 4

[PW96] Parke F., Waters K.: *Computer Facial Animation*. A K Peter, 289 Linden Street, Wellesley, MA 02181, 1996. 4

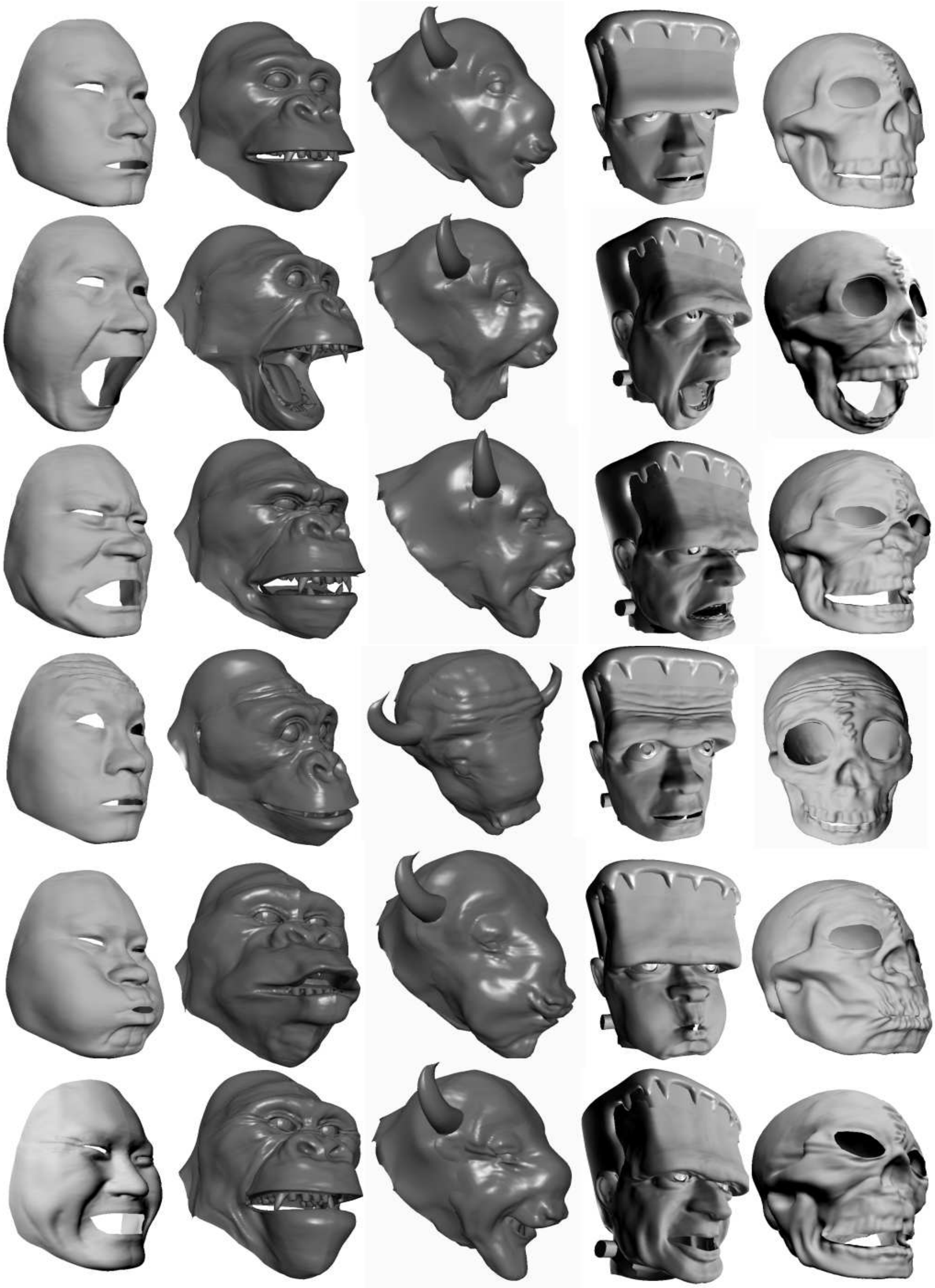

**Figure 13:** *The first column shows the snapshots of the source normal mesh. The other columns show the snapshots of the retargetted expressions of gorilla, ox, Frankenstein, and skull models.*

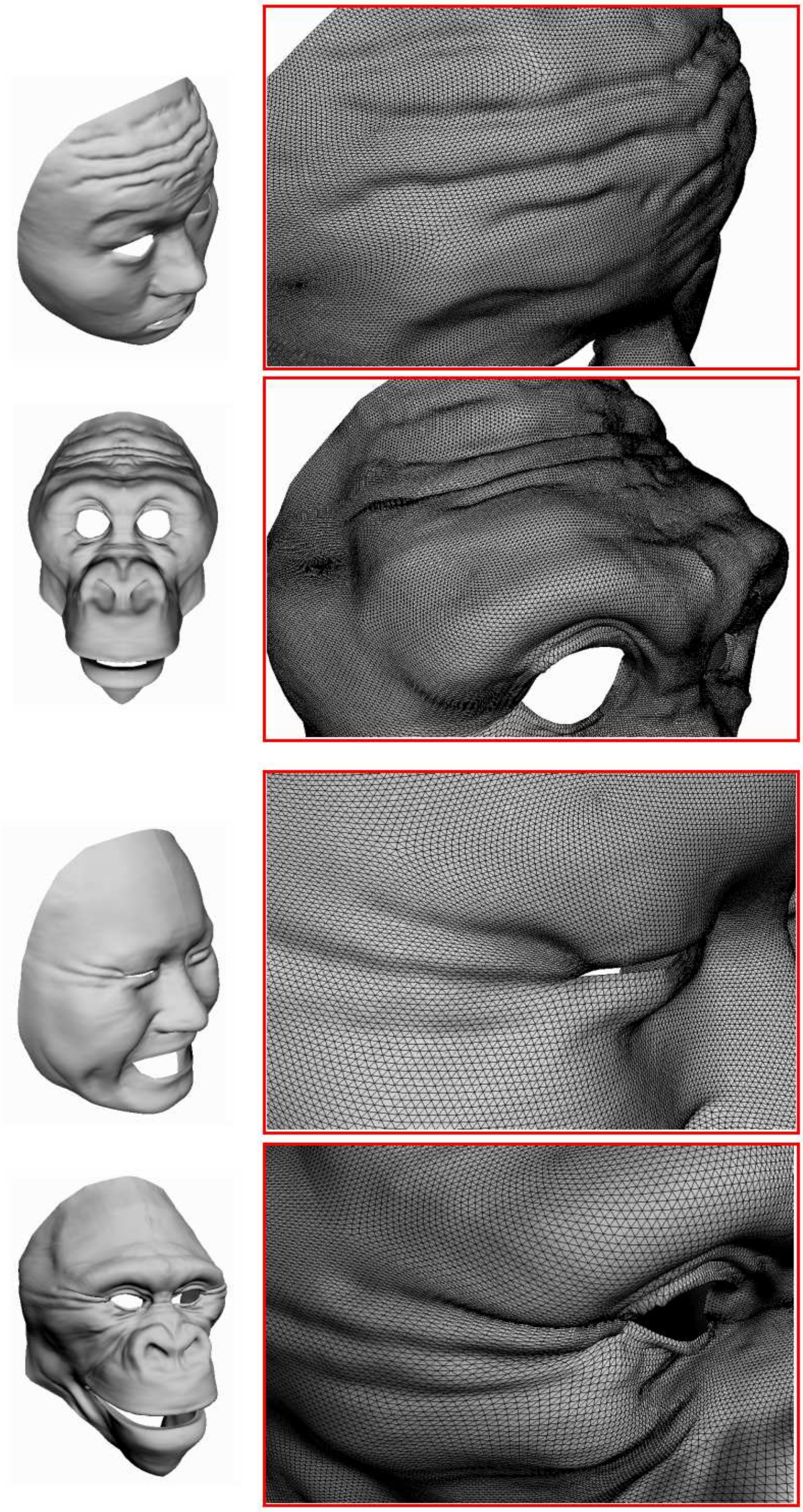

**Figure 14:** *The two top figures show how the wrinkles on the head of the human are transferred to the gorilla model. The two bottom figures show how the wrinkles of the right eye in the human are transferred to the gorilla model.*