Localized Optimization for Mocap-Driven Blendshapes

Ko-Yun Liu[13] Wan-Chun Ma[2] Jackson Lee[1] Chuan-Chang Wang[1] Chun-Fa Chang[4]

[1]Media Laboratory
Next Media Animation

[2]Institute for Creative Technologies
University of Southern California

[3]Dept. of Computer Science
National Tsing Hua University

[4]Dept. of Computer Science
National Taiwan Normal University

Figure 1: *A screenshot from the final animation result.*

The main purpose of the proposed technique is to allow a blendshape rig to create facial expressions that are independent from the underlying blendshape poses. [JTDP03] proposed an automatic segmentation technique to divide the face into regions that have similar amount of deformation. Each region finds its best blendshape weights that conforms to the motion capture data and then propagates the weights to each vertex by radial basis functions (RBFs) [LCF00].In our case, we ask the user to provide desired facial segmentation and establish a corresponding set of motion capture data for each blendshape. The benefits are:

1. An animator can simply control facial segmentation by drawing the masks without much efforts. It is not guaranteed that [JTDP03] would produce a segmentation that is good for producing asymmetrical expressions. (An RBF-based low-frequency warping is applied in their technique to deal with this problem.)
2. The masks can be reused as long as the parametrization of the blendshape rig remains the same. Whereas [JTDP03] would produce various segmentations according to different manually-selected thresholds. Besides, deformation-based segmentation is usually inconsistent over different characters.
3. It is not easy to find an precise mapping between motion capture markers and the corresponding locations in blenshapes when using RBFs, while we needs only a training session that records motion capture data that matches a predefined set of facial expressions for direct retargeting.

1. Facial Motion Retargeting

To drive the blendshape model, we first created a corresponding set of motion capture data for each of the blendshape poses. The poses bijectively map to the blendshape expressions. Since the blendshape weights will be computed solely based on motion capture data, the motion capture actor may not necessarily be the same person whom we scan for the digital actor as long as the bijective mapping is established. Here we define a motion capture basis vector s_i and the target vector t as:

$$\begin{aligned} s_i &= [\mathbf{x}(p_1^{s_i}), \mathbf{y}(p_1^{s_i}), \mathbf{z}(p_1^{s_i}) \cdots, \mathbf{x}(p_{n_m}^{s_i}), \mathbf{y}(p_{n_m}^{s_i}), \mathbf{z}(p_{n_m}^{s_i})]^T, \\ t &= [\mathbf{x}(p_1^{t}), \mathbf{y}(p_1^{t}), \mathbf{z}(p_1^{t}), \cdots, \mathbf{x}(p_{n_m}^{t}), \mathbf{y}(p_{n_m}^{t}), \mathbf{z}(p_{n_m}^{t})]^T. \end{aligned}$$

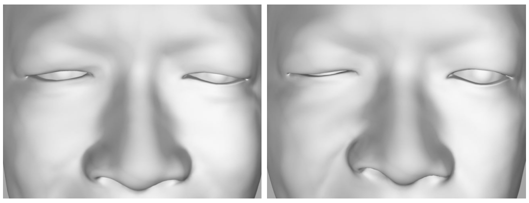
n_m is the number of markers. $\mathbf{x}(p_i)$, $\mathbf{y}(p_i)$, and $\mathbf{z}(p_i)$ are functions that return the X, Y and Z Cartesian coordinates of a marker position p_i. We can put all the basis vectors into a single matrix $S = [s_1, s_2, \ldots, s_{n_b}]$, where n_b is the number of the blendshape poses. The recovery of blendshape weights w for a given shape t can be posed into a constrained least square problem:

$$\underset{1 \geq w(i) \geq 0,\ \sum_{i=1}^{n_b} w(i)=1}{\arg\min} \quad \| Sw - t \|^2 . \qquad (1)$$

Applying the same weight vector w to the blendshape model yields a face shape which conforms the best to the input motion capture data t.

2. Localized Optimization

However, all of our blendshapes have symmetrical facial expressions. By just linearly combining the shapes we will



Traditional Localized

Figure 2: *Comparison of traditional (left) and localized (right) blendshape controls for a single-eye blinking expression.*

not produce asymmetrical facial expressions.Usually digital artists will split each one of symmetrical shapes into several localized shapes, but this method produces a large number of shapes. To avoid this problem, the face is separated into different regions based on predefine masks, then blendshape weights are computed based on these local regions.

We denote M_k as the scaling matrix for mask k. M_k scales the motion capture marker positions according to the user-defined mask value:

$$M_k = \{m^k_{ij} | 1 \le i \le 3n_m, 1 \le j \le 3n_m\},$$

$$m^k_{ij} = \left\{ \begin{array}{ll} \mathbf{m}_k(q) & \text{if } i = j, q = \lfloor \frac{i-1}{3} \rfloor + 1 \\ 0 & \text{otherwise.} \end{array} \right. ,$$

$$\sum_{k=1}^{n_r} m^k_{ii} = 1, \qquad 0 \le m^k_{ii} \le 1.$$

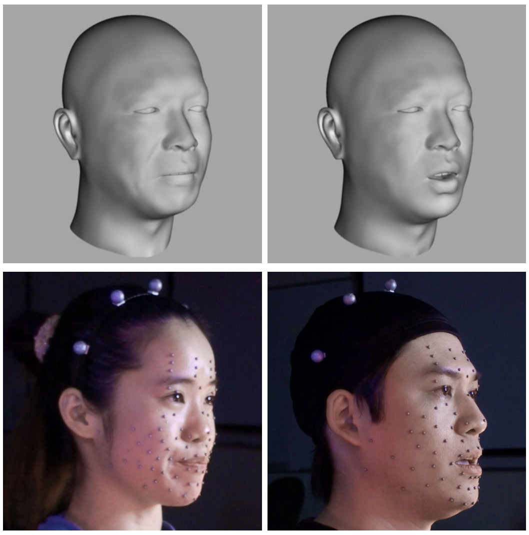
n_r is the number of masks (regions). Function $\mathbf{m}_k(q)$ returns how important p_q is in mask k (q is equivalent to the identification number of the motion capture marker). Since motion capture data is a lower resolution version of face, it is straight forward that both the motion capture markers and face geometry can share the same texture space. Therefore, one way to implement function $\mathbf{m}_k(q)$ is to compute the average mask value within p_q's Vonoroi diagram cell in the texture space. The blendshape weights are calculated region by region. For each region, we formulate the following linear system to compute regional blendshape weights:

$$\underset{1 \ge w_k(i) \ge 0,\ \sum_{i=1}^{n_b} w_k(i) = 1}{\arg\min} \quad \| M_k S w_k - M_k t \|^2, \qquad (2)$$

Figure 2 shows a comparison of traditional and localized blendshape controls for a single-eye blinking expression.

3. Results

A Vicon system with ten infrared cameras was set up for facial motion capture. Eighty-five markers were put on the actor's face for capturing facial expressions and five additional markers were put on his head for determining head rigid transformation during the performance. We also provided images of predefined nineteen different expressions to the actor in order to practice and perform similar expressions for retargeting. We set up a 3D scanner to scan our digital double. Eight of these face scans are based on Preston Blair's phoneme series. Post-processing of the scans and building the corresponded blendshape poses were done manually by digital modelers with Maya and ZBrush software.



(a) (b)

Figure 3: *Retargeting different motion capture actors.*

A complete human head including hair and neck was created for demonstration. We manually placed custom 3D models of teeth and tongue into every blendshape poses. A color face texture which is free of specular reflections is acquired using cross-polarization technique, because relighting based on specular-free texture is usually more physically plauible. Mental Ray with custom shaders was used to render skin, teeth, and eyes with image-based lighting. Hair was simulated by Maya Fur and rendered with Mental Ray's fur shader. Figure 1 shows one screenshot from the final animation results. Figure 3 contains results that are retargetted from two different motion capture actors.

References

[JTDP03] Joshi P., Tien W. C., Desbrun M., Pighin F.: Learning controls for blend shape based realistic facial animation. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 187–192. 1

[LCF00] Lewis J. P., Cordner M., Fong N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proc. of SIGGRAPH* (2000), pp. 165–172. 1