Synthesis of Arbitrarily Oriented Face Views from Two Images

Yasuhiro MUKAIGAWA Yuichi NAKAMURA Yuichi OHTA

University of Tsukuba, Institute of Information Sciences and Electronics
Tennodai 1-1-1, Tsukuba, Ibaraki 305,Japan
mukaigaw@image.is.tsukuba.ac.jp

Abstract

In this paper, we propose a method for synthesizing arbitrarily oriented facial views from only two facial images. The method includes the following features: unlike the 2-D interpolation method, the views by true 3-D rotation can be generated; unlike the stereo, the camera parameters of the two input images are not necessary; although it is based on the structure-from-motion theory, the 3-D structure of a face is not explicitly reconstructed.

Assuming rigidity and orthographic projection, the 2-D coordinates of feature points in any view can be expressed as a linear combination of those in the two input images. The coefficients of the linear combination are determined based on the facial orientation to be generated. The facial image is synthesized by adaptively mapping the texture of the input images to the 2-D triangular patch model whose vertices are the feature points.

1 Introduction

For a variety of applications, the synthesis of arbitrarily oriented facial views is important. Texture mapping to a 3-D facial model[1], that is usually a polyhedron with a lot of patches, is a popular way for this purpose. It, however, requires the accurate measurement of an individual face. This requirement is infeasible in many applications.

On the other hand, 2-D interpolation methods[2] can be used to synthesize intermediate facial views from two reference images without the 3-D shape model. In this method, the position of each point on the face in a synthesized view is estimated by a linear interpolation between those in reference images. The advantage of this method, which does not need 3-D information, limits the range of its applicability; it can not generate a view with arbitrary orientation.

To cope with this problem, this paper proposes a new method which can generate facial views with arbitrary orientations by using only two facial images. Unlike the usual stereo case, the viewing direction of the two input images can be unknown. Unlike the usual structure-from-motion case, the 3-D structure of the face is not explicitly reconstructed.

2 Key idea

According to the structure-from-motion theory[3], the 3-D structure of an object can be reconstructed from some views taken from distinct positions. The reconstructed 3-D structure can be used for the synthesis of arbitrarily oriented views. The reconstruction from images, however, is usually unstable and sensitive to noise. Therefore a number of studies, which try to stabilize the reconstruction by using a least square method with a large number of image sequences, have been made[4]. However, the reconstruction of 3-D structure is not necessary for the purpose of the synthesis of 2-D images. In other words, it is sufficient for our purpose if we could make 2-D images of arbitrary views directly from the 2-D input images.

In this paper, we show a method for synthesizing arbitrarily oriented facial views from only two facial images without 3-D reconstruction under the conditions of rigidity and orthographic projection. The principle is related to the structure-from-motion theory, which indicates that the 3-D structure can be reconstructed from three input images. Koenderink et al.[5] showed that the 3-D structure can be reconstructed from two images within the degree of freedom of 3-D affine transformation. Ullman et al.[6] proved that an arbitrarily oriented object can be recognized by a linear combination of the 2-D coordinates of feature points in a few images of the object. They discussed about the application of the 2-D view models to the recognition of 3-D rigid objects.

We apply the principle, that the 2-D coordinates of feature points in any view can be expressed by a linear combination of the 2-D coordinates of feature points in a few images, to the image synthesis and try to synthesize arbitrarily oriented facial views by the combination of two facial images.

3 Calculating coordinates of feature points

3.1 Basic principle

Let $\boldsymbol{P}$ be a set of points in a 3-D space, and let $[x_i, y_i, z_i, 1]^T$ be the coordinates of the i th point. $\boldsymbol{P}$ is represented by a $4\times n$ matrix as shown in the following.

$$\boldsymbol{P} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & \cdots & z_n \\ 1 & 1 & \cdots & 1 \end{bmatrix} \quad (1)$$

As shown in equation(2), $\boldsymbol{B}_j$ represents a set of points which is a transformation by 4×4 affine matrices $\boldsymbol{R}_j (j = 1, 2, \ldots, 5)$.

$$\boldsymbol{B}_j = \boldsymbol{R}_j \boldsymbol{P} \quad (j = 1, 2, \cdots, 5) \quad (2)$$

Let $[x_{ji}, y_{ji}, z_{ji}, 1]^T$ be the coordinates of the i th point after the affine transformation by $\boldsymbol{R}_j$. $\boldsymbol{B}_j$ is a $4 \times n$ matrix as shown in equation(3).

$$\boldsymbol{B}_j = \begin{bmatrix} x_{j1} & x_{j2} & \cdots & x_{jn} \\ y_{j1} & y_{j2} & \cdots & y_{jn} \\ z_{j1} & z_{j2} & \cdots & z_{jn} \\ 1 & 1 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_j \\ \boldsymbol{y}_j \\ \boldsymbol{z}_j \\ \boldsymbol{1} \end{bmatrix} \quad (3)$$

Let $\boldsymbol{x}_j$ be the row vector of $\boldsymbol{B}_j$ which represents the X-coordinates, and let $\boldsymbol{r}_j$ be the first row vector of $\boldsymbol{R}_j$. These vectors satisfy the following equation.

$$\boldsymbol{x}_j = \boldsymbol{r}_j \boldsymbol{P} \quad (j = 1, 2, \cdots, 5) \quad (4)$$

If $\boldsymbol{r}_1$ through $\boldsymbol{r}_4$ are linearly independent, a set of coefficients $a_k (k = 1, 2, 3, 4)$ which satisfy equation(5) should exist.

$$\boldsymbol{r}_5 = a_1 \boldsymbol{r}_1 + a_2 \boldsymbol{r}_2 + a_3 \boldsymbol{r}_3 + a_4 \boldsymbol{r}_4 \quad (5)$$

It is because the dimension of $\boldsymbol{r}_j$ is 4. So $\boldsymbol{x}_5$, which represents a set of the X-coordinates transformed by $\boldsymbol{R}_5$, is expressed by a linear combination of $\boldsymbol{x}_1$ through $\boldsymbol{x}_4$ as shown in equation(6). In the same way, Y-coordinates are expressed using the coefficients $b_k (k = 1, 2, 3, 4)$ as shown in equation(7).

$$\boldsymbol{x}_5 = a_1 \boldsymbol{x}_1 + a_2 \boldsymbol{x}_2 + a_3 \boldsymbol{x}_3 + a_4 \boldsymbol{x}_4 \quad (6)$$

$$\boldsymbol{y}_5 = b_1 \boldsymbol{y}_1 + b_2 \boldsymbol{y}_2 + b_3 \boldsymbol{y}_3 + b_4 \boldsymbol{y}_4 \quad (7)$$

Therefore assuming the orthographic projection, the X- and Y-coordinates on arbitrary images can be expressed as a linear combination of four vectors which are linearly independent. Now, we know that the explicit reconstruction of 3-D structure is not necessary.

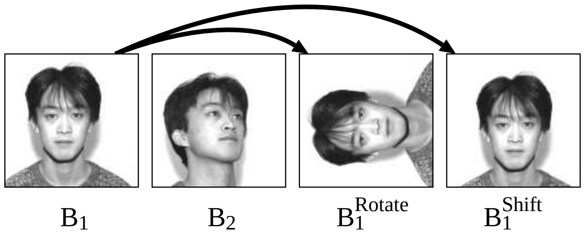


Figure 1: Four *base images*

3.2 Using two images

If we have four images, the coordinates in any image are expressed by a linear combination of the four vectors ($\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, $\boldsymbol{x}_4$) which are linearly independent. However, these four vectors are not necessarily taken from distinct four images. If there are two images (B_1, B_2), the four linearly independent vectors can be obtained easily.

As shown in Figure 1, an image B_1^{Rotate} is created by rotating B_1 around the perpendicular axis to the image plane, and another image B_1^{Shift} is created by shifting B_1 on the image plane. We call these four images (B_1, B_2, B_1^{Rotate}, B_1^{Shift}) '*base images*'. Let $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_1^{Rotate}$, $\boldsymbol{x}_1^{Shift}$ are the vectors, representing the X-coordinates of all feature points on the *base images*. If the relation of two images B_1 and B_2 is neither a rotation around the perpendicular axis to the image plane nor a shift on the image plane, these four vectors become linearly independent.

The X-coordinates vector $\hat{\boldsymbol{x}}$ of another view $\hat{B}$ is expressed as a linear combination as shown in equation(8). Similarly the Y-coordinates vector $\hat{\boldsymbol{y}}$ is expressed as shown in equation (9).

$$\hat{\boldsymbol{x}} = a_1 \boldsymbol{x}_1 + a_2 \boldsymbol{x}_2 + a_3 \boldsymbol{x}_1^{Rotate} + a_4 \boldsymbol{x}_1^{Shift} \quad (8)$$

$$\hat{\boldsymbol{y}} = b_1 \boldsymbol{y}_1 + b_2 \boldsymbol{y}_2 + b_3 \boldsymbol{y}_1^{Rotate} + b_4 \boldsymbol{y}_1^{Shift} \quad (9)$$

3.3 Determining the coefficients

In order to determine the coefficients for the linear combination to synthesize an image, an image called '*indicator image*' which specifies the facial orientation in the synthesized image is used. At least four feature points have to be detected on the *indicator image*. Given a set of feature points, the coefficients(a_1 through a_4) which satisfy equation(10) are calculated.

$$\hat{\boldsymbol{x}'} = a_1 \boldsymbol{x}'_1 + a_2 \boldsymbol{x}'_2 + a_3 \boldsymbol{x}'^{Rotate}_1 + a_4 \boldsymbol{x}'^{Shift}_1 \quad (10)$$

The X-coordinates vector $\hat{\boldsymbol{x}'}$ in equation(10) is obtained from the feature points detected on the *indicator image*, and four vectors $\boldsymbol{x}'_1$, $\boldsymbol{x}'_2$, $\boldsymbol{x}'^{Rotate}_1$ and

x'^{Shift}_1 are the X-coordinates of the same feature points on the four *base images*. If the number of the feature points on the *indicator image* is more than four, the coefficients can be calculated using a least square method. Then 2-D coordinates of all feature points are calculated using the equations(8) and (9).

4 Synthesizing facial image

4.1 Texture mapping

In order to synthesize a facial image, a triangular patch model whose vertices are the feature points is created, and textures taken from the *base images* are mapped to this patch model. Since the texture can be selected from the two facial images, there are two possible ways for selecting the texture:

(1) Choose all texture from one image whose facial orientation is closer to the synthesized image.

(2) Choose each texture from the image in which the patch has a larger area than in the other image.

In the method(1), the discontinuity of texture between neighboring patches does not occur since all textures are obtained from a same image. However, when we synthesize an image sequence whose facial directions are gradually changing, the selected texture will change drastically at a certain orientation of the face. In the method(2), the discontinuity of texture between neighboring patches occurs as shown in Figure 2.

Although both methods have disadvantages, the method(2) has some good characteristics. One is that the mapped texture is independent of the facial direction being synthesized. Another is that the mapped texture will be in good quality, because the texture is taken from the larger patch. Therefore, we employ the method(2) and try to reduce the discontinuity of the texture.

4.2 Reducing the discontinuity

We compare the pixel on the boundary and reduce the discontinuity of texture by the following two methods.

- Patch division
- Intensity revision

Details are shown in the following sections.

4.2.1 Patch division

Modeling a facial surface as a set of flat patches causes the texture misalignment at the patch boundaries, because the texture is warped independently. To align the texture, we divide the patch as shown in Figure 3. The process of the division is shown below.

Figure 2: An example of the discontinuity of the texture

1. Pixels along the both sides of the boundary are resampled to generate a pair of intensity profiles. Then their edges are detected by differentiation.
2. The pair of intensity profiles are matched using dynamic programming[7].
3. The patches are divided at each edge point. Then they are expanded or contracted according to the correspondences obtained by the previous step.

4.2.2 Intensity revision

Another serious problem is the different lighting conditions of the two *base images*. As illustrated in Figure 4, there are two types of regions according to the images B_1 or B_2 from which the texture was selected. Serious intensity discontinuity occurs at the boundary of such two regions. We reduce this discontinuity by revising the intensity.

At the boundary, the internal pixel P_i and the external pixel P'_i should have similar intensity. Let $I(P_i)$ be the intensity of pixel P_i, and $I(P'_i)$ be the intensity of pixel P'_i. The intensity $I(P_i)$ is revised according to the equations(11) and (12).

$$I(P_i) \leftarrow I(P_i) + d_{P_i} \tag{11}$$

$$d_{P_i} = \frac{I(P'_i) - I(P_i)}{2} \tag{12}$$

This means that each intensity at the boundary is replaced with their mean value. For the pixels inside of the region, for example Q, the intensity $I(Q)$ is revised by equation(13).

$$I(Q) \leftarrow I(Q) + d_Q \tag{13}$$

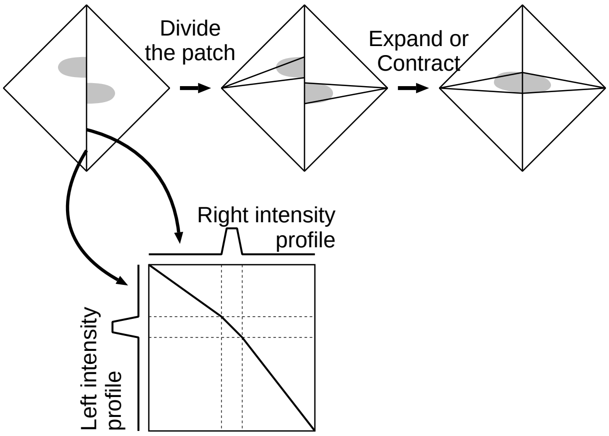


Figure 3: Alignment of patch boundaries using dynamic programming and additional division of patches

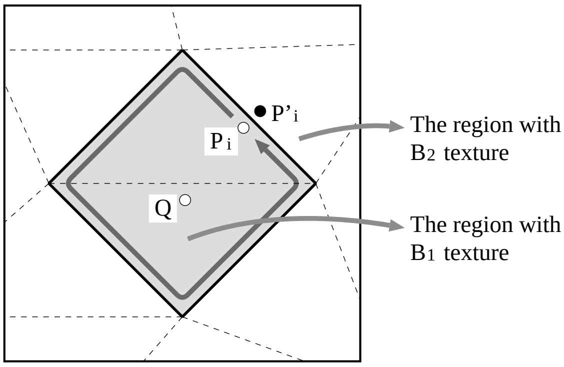


Figure 4: Intensity revision

Here d_Q is calculated as a weighted summation of the d_{P_i} of all boundary pixels as shown in equation(14).

$$d_Q = \sum_i w_i d_{P_i} \tag{14}$$

Where w_i is the weight inversely proportional to the distance between Q and P_i, and $\sum w_i = 1$.

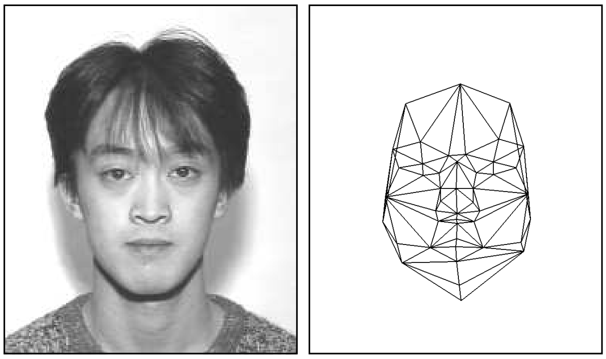
By these methods, even if the location of the light and the solid shape of the head are unknown, the facial image can be synthesized without discontinuity of texture.

5 Experiments

The input images are shown in Figure 5(a)(b). The 2-D coordinates of forty five feature points were given manually. The front view shown in Figure 6(a) is given as an *indicator image* in which 2-D coordinates of five feature points (the outer corners of both eyes, the nose peak, and the both corners of mouth) were detected. By using this *indicator image*, the coefficients for the

(a) (b)

Figure 5: Two input images



(a) (b)

Figure 6: *Indicator image* and 2-D patch model

linear combination were calculated and the 2-D triangular patch model was created as shown in Figure 6(b).

The results of texture mapping onto this patch model are shown in Figure 7. Fig.7(a) shows the result created by the method(1) described in 4.1. Fig.7(b) shows the result created by the method(2). In this image, discontinuity of texture occurs. Fig.7(c) shows the result after the elimination of discontinuity described in 4.2. Other *indicator images* and synthesized images are shown in Figure 8. We can see that the arbitrarily oriented facial views can be synthesized from only two images.

6 Conclusion

We have proposed a method for the synthesis of arbitrarily oriented facial views from two images without explicit reconstruction of the 3-D shape. We implemented this method and showed its effectiveness by experiments using real images.

References

[1] Y.Suenaga, Y.Watanabe, "A method for the synchronized acquisition of cylindrical range and color

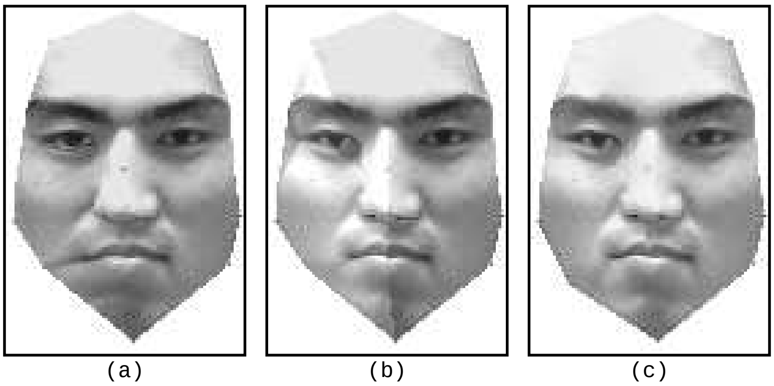


Figure 7: Synthesized images

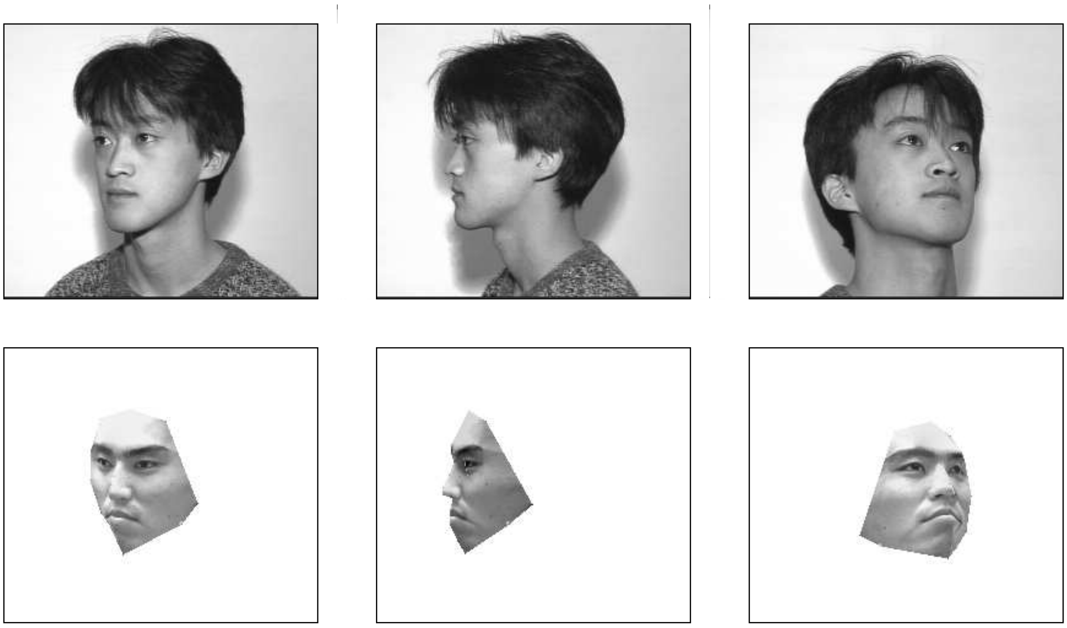


Figure 8: Synthesis of various facial views
(top: *indicator images* specifying facial orientation, bottom: synthesized views)

data", *IAPR Workshop on MVA*, Tokyo, Nov. (1990)

[2] R.Skerjanc, and J.Liu, "Computation of Intermediate Views for 3DTV", *Proc. 5th Workshop 1992 on Theoretical Foundations of Computer Vision*, pp.190-201 (1992)

[3] S.Ullman, "The Interpretations of Visual Motion", MIT Press, Cambridge, MA (1979)

[4] C.Tomasi, T.Kanade, "The Factorization Method for the Recovery of Shape and Motion from Image Streams", *In Proceedings of Image Understanding Workshop*, pp.459-472 (1992)

[5] J.J.Koenderink, A.J.van Doorn, "Affine structure from motion", J.Opt.Soc.Am.A, Vol.8, No.2, pp.377-385 (1991)

[6] S.Ullman, R.Basri, "Recognition by Linear Combinations of Models", *IEEE Trans. PAMI*, Vol.13, No.10, pp.992-1006 (1991)

[7] Y.Ohta, T.Kanade, "Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming",*IEEE Trans. PAMI*, Vol.7, No.2, pp.139-154 (1985)