APPEARANCE MODEL BASED FACE-TO-FACE TRANSFORM

Takayuki Nagai † *and Truong Nguyen*‡

†Dept. of Electronic Engineering, The University of Electro-Communications, Tokyo, Japan
‡Dept. of Electrical & Computer Engineering, Uiversity of California, San Diego, USA
Email:tnagai@ee.uec.ac.jp, nguyent@ece.ucsd.edu

ABSTRACT

In this paper, a novel approach to face-to-face transform is presented. The face-to-face transform is a technique, which transforms one person's facial actions to the others. In general, the 3D models of faces are used for such transformation. Therefore the facial action parameters should be estimated from the 2D input images, which is not an easy task. On the contraly, our proposed approach is based on the 2D appearance model instead of the 3D model so that the model is acquired by learning directly from training images. To achieve this, we investigate making use of the Hidden Markov Model (HMM) framework, which models the correspondence between an input face and the other's one as well as the appearances of both faces. The experimental results show the effectiveness of the proposed method.

1. INTRODUCTION

The face-to-face transform is a technique, which transforms one person's facial actions to the others. Such systems have been studied extensively, since animated characters and talking heads are playing an important role in computer interfaces and so forth. In general, the 3D models of faces are used for such transformation. Thus the facial action parameters must be estimated from the 2D input images, which is not an easy task. On the other hand, image-based methods have been proposed recently[4][5]. Torre *et al.* [5] have proposed the method, which is based on Robust Principal Component Analysis (RPCA). Although any 3D model is not employed in the method, facial parts must be correctly recognized. Furthermore, a lot of coupled training samples of both persons are required, since the method is based on PCA.

In this paper, a novel approach to the face-to-face transform is presented. Our proposed approach is based on the 2D appearance model instead of the 3D model. To achieve this, we investigate making use of the Hidden Markov Model (HMM) framework, which models the correspondence between an input face and other's one as well as the appearances of both faces. The model is trained directly from training images by using EM algorithm. The advantage of the proposed system is that it requires only one image for each person. Results of the experiments illustrate the validity of our proposed method.

2. APPEARANCE MODELS USING 2D HMM

In this section, we propose the appearance models of image sequences. Since the proposed method uses Pseudo 2-D Hidden Markov Models, we briefly describe them first. Then, the image representation using PHMMs is introduced.

(a) (b) (c)

Figure 1: PHMM representaion of the image. (a)Input image. (b)Reconstructed image. (c)State sequence.

2.1. Pseudo 2-D Hidden Markov Model

To represent a 2-D signal, 2-D HMM is desirable. Due to the high complexity of fully connected 2-D HMMs, Pseudo 2-D HMM (PHMM) is used in this approach. PHMMs are shown to be effective for both character[1] and face recognition[2]. PHMMs consist of a set of super states (super-states) and normal states(embedded-states), which are embedded in super-states. The super-states model the vertical direction, while the embedded states model the horizontal one. The PHMM is parameterized by initial super-state distribution $\boldsymbol{\Pi}_s$, super-state transition probability matrix $\boldsymbol{A}_s = \{a_{kj}\}$, initial embedded-state distribution $\boldsymbol{\Pi}_e^{(k)}$, and embedded-state transition probability matrix $\boldsymbol{A}_e^{(k)} = \{a_{ij}^{(k)}\}$. Furthermore, embedded-states are characterized by Gaussian mixture densities with mixture weights $\boldsymbol{C}_e^{(k)} = \{c_{jm}^{(k)}\}$, mean vectors $\boldsymbol{M}_e^{(k)} = \{\boldsymbol{\mu}_{jm}^{(k)}\}$ and covariance matrices $\boldsymbol{U}_e^{(k)} = \{\boldsymbol{\Sigma}_{jm}^{(k)}\}$. Hereafter, model of the PHMM is represented as

$$\boldsymbol{\lambda} = \{\boldsymbol{\Pi}_s\ \boldsymbol{A}_s\ \boldsymbol{\Lambda}_e\}, \tag{1}$$

where

$$\boldsymbol{\Lambda}_e = \{\boldsymbol{\Lambda}^{(1)}\ \boldsymbol{\Lambda}^{(2)}\ \cdots\ \boldsymbol{\Lambda}^{(N_s)}\}, \tag{2}$$

$$\boldsymbol{\Lambda}^{(k)} = \{\boldsymbol{\Pi}_e^{(k)}\ \boldsymbol{A}_e^{(k)}\ \boldsymbol{C}_e^{(k)}\boldsymbol{M}_e^{(k)}\ \boldsymbol{U}_e^{(k)}\}, \tag{3}$$

and N_s denotes the number of super-states.

2.2. PHMM image representation

This subsection describes how PHMMs encode the input image information. Assuming that the trained PHMM has been obtained. Then, the input image can be decoded by

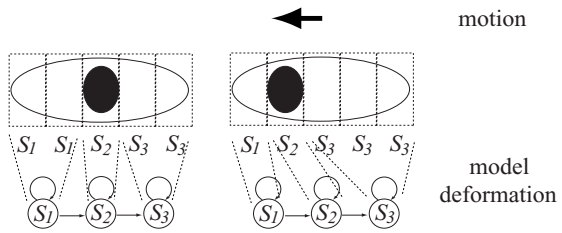

Figure 2: Two different interpretations of the motion.

Figure 3: Reconstructed images using the model of Fig.1. (a)Input images. (b)Reconstructed images. (c)State sequences.

using doubly embedded Viterbi algorithm and the decoding process provides us the state sequence (and mixture indices). Since each state has a gaussian (gaussian mixture) PDF, it is obvious that the most likely output image for given state sequence is obtained by arranging the mean value of each state (gaussian PDF) according to the state sequence.

The model can be trained easily by using EM algorithm. Figure1 shows an example of miss america image (1st frame of the sequence). Figure 1 (a) is used for the training and (b) is reconstructed from the model and the state sequence (c). It can be seen that the input image is well reconstructed from the model and the state sequence.

2.3. Motion as a model deformation

Since the video sequences contain some object movements, the model must be able to represent motions. In the proposed model, motions are considered as model deformations. This idea is depicted in Fig.2. From this figure, one can see that the model deformation corresponds to the change of the state sequence. Figure 3 shows actual examples of miss america sequence. The model, which is trained by Fig.1(a) was used to represent Figs.3 (a). Doubly embedded Viterbi algorithm gives the state sequences Figs.3 (c) and the images Figs.3 (b) are reconstructed by using the model and the state sequences. It can be confirmed that the model is able to represent the images with movements. In fact, PSNRs of both images (b) are around 37[dB]. It should be noted that the performance of the reconstruction largely depends on the umber of states and mixtures.

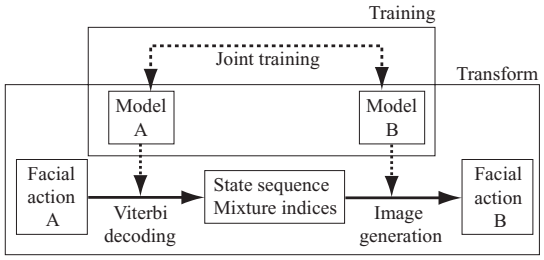

Figure 4: The basic idea of the face to face transform.

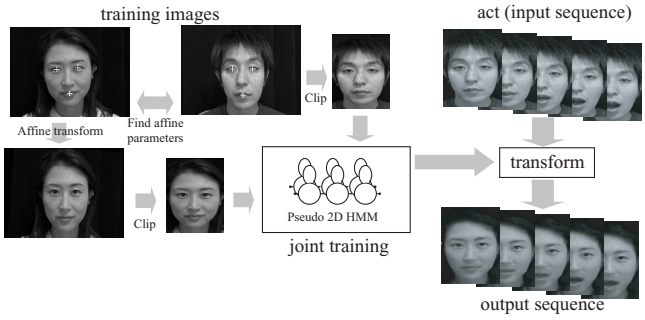

Figure 5: An overview of the proposed system.

3. FACE-TO-FACE TRANSFORM

This section introduces the face-to-face transform, which is based on the 2D appearance models proposed in the foregoing section.

3.1. System's overview

The basic idea of the transform is to represent the input sequence using person A's model, and then person B's model is used to reconstruct the images (Fig.4). As we have described, the encoded data (the state sequence) represent the motion, hence the person A's facial action is expected to be transformed to the person B's face.

Figure 5 illustrates an overview of the proposed system. In the training phase, two images (one facial image for each person) are prepared. At first, eyes and mouth are localized to find the affine parameters. In our system, faces are first detected by [6] and then facial parts are localized by using modular eigenspace[7]. Then face B is affine transformed so that the model can learn the correspondence easily. The details of the training phase are discussed in the next subsection. The face-to-face transform is carried out by applying the trained models to the input image sequence of person A. The actual procedure of the transform is presented in sections **3.3** and **3.4**.

3.2. Joint training of PHMM

To model the relationship between both faces, the joint training of PHMMs is introduced. The basic idea is illustrated in Fig.6. As the figure shows, the coupled PHMMs share a common state transition probability between coupled states. However, mean vectors and covariance matrices

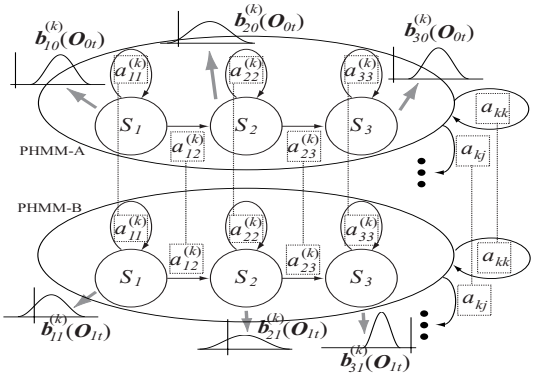

Figure 6: The basic idea of the joint PHMM.

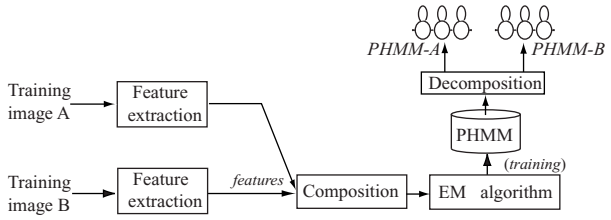

Figure 7: Details of the joint training.

are different in each state.

Now, let j, k, ℓ and m be the embeded state index, the super state index, the index of the person (0 or 1) and m-th mixture on the state j, respectively. $\boldsymbol{O}_{\ell t}$ represents an observed vector of the person ℓ at a pixel t. The output probability of the state j is

$$\begin{aligned} \boldsymbol{b}_j^{(k)}(\bar{\boldsymbol{O}}_t) &= \sum_{m=1}^{N} c_{jm}^{(k)} \mathcal{N}(\bar{\boldsymbol{O}}_t; \boldsymbol{\mu}_{jm}^{(k)}, \boldsymbol{\Sigma}_{jm}^{(k)}) \\ &= \sum_{m=1}^{N} \left[c_{jm}^{(k)} \prod_{\ell=0}^{1} \left\{ \mathcal{N}(\boldsymbol{O}_{\ell t}; \boldsymbol{\mu}_{jm\ell}^{(k)}, \boldsymbol{\Sigma}_{jm\ell}^{(k)}) \right\} \right], \end{aligned} \quad (4)$$

where

$$\bar{\boldsymbol{O}}_t = [\boldsymbol{O}_{0t}^T \ \boldsymbol{O}_{1t}^T]^T, \quad (5)$$

$$\boldsymbol{\mu}_{jm}^{(k)} = [\boldsymbol{\mu}_{jm0}^{(k)\ T} \ \boldsymbol{\mu}_{jm1}^{(k)\ T}]^T, \quad (6)$$

$$\boldsymbol{\Sigma}_{jm}^{(k)} = \begin{bmatrix} \boldsymbol{\Sigma}_{jm0}^{(k)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{jm1}^{(k)} \end{bmatrix}, \quad (7)$$

and $\boldsymbol{\mu}_{jm\ell}^{(k)}$, $\boldsymbol{\Sigma}_{jm\ell}^{(k)}$, $c_{jm}^{(k)}$, and N represents mean vector, covariance matrix, mixture weight and number of mixtures, respectively. We also assume that $\boldsymbol{\Sigma}_{jm\ell}^{(k)}$ is a diagonal matrix. Here, the decomposition of the PHMM $\boldsymbol{\lambda}$ into two persons' models $\boldsymbol{\lambda}_\ell$ using Eqs.(6)(7) is considered. The marginalization of Eq.(4) results in

$$\begin{aligned} \boldsymbol{b}_j^{(k)}(\boldsymbol{O}_{0t}) &= \int \boldsymbol{b}_j^{(k)}(\bar{\boldsymbol{O}}_t) d\boldsymbol{O}_{1t} \\ &= \sum_{m=1}^{N} c_{jm}^{(k)} \mathcal{N}(\boldsymbol{O}_{0t}; \boldsymbol{\mu}_{jm0}^{(k)}, \boldsymbol{\Sigma}_{jm0}^{(k)}). \end{aligned} \quad (8)$$

Thus $Pr(\boldsymbol{O}_0|\boldsymbol{\lambda}) = Pr(\boldsymbol{O}_0|\boldsymbol{\lambda}_0)$ holds true, where $\boldsymbol{O}_\ell = [\boldsymbol{O}_{\ell 1}^T \ \boldsymbol{O}_{\ell 2}^T \ \cdots \ \boldsymbol{O}_{\ell\tau}^T]^T$ and τ is a number of pixels. This implies that the likelihood of the observation $\boldsymbol{O}_0$ can be evaluated by the decomposed model $\boldsymbol{\lambda}_0$. $\boldsymbol{O}_1$ can also be evaluated by the decomposed model $\boldsymbol{\lambda}_1$. In the training phase, features are extracted from both face images respectively and are combined according to Eq.(5) to share the same transition probabilities among $\boldsymbol{\lambda}_\ell$. A set of parameters of the PHMM $\boldsymbol{\lambda}$ is obtained through EM-algorithm using the combined feature vectors. Then, applying Eqs.(6)(7) inversely results in the PHMM $\boldsymbol{\lambda}_\ell$.

Figure 7 depicts the actual procedure of the training. First of all, a set of training images is prepared and features are extracted. As the features, the pixel value itself, Gaussian, Laplacian and first order derivatives (both vertical and horizontal directions) are used. These feature vectors are combined as one feature vector and PHMMs are trained. After the training, each PHMM is divided into two models PHMM-A and PHMM-B.

3.3. Face transform using PHMM

The objective of the transform is to obtain the image $\boldsymbol{O}_1$, which maximizes the conditional probability $Pr(\boldsymbol{O}_1|\boldsymbol{O}_0)$. $Pr(\boldsymbol{O}_1|\boldsymbol{O}_0)$ can be rewritten as

$$Pr(\boldsymbol{O}_1|\boldsymbol{O}_0) = \int\int Pr(\boldsymbol{O}_1, \boldsymbol{S}_1, \boldsymbol{S}_0|\boldsymbol{O}_0) d\boldsymbol{S}_1 d\boldsymbol{S}_0, \quad (9)$$

where $\boldsymbol{S}_1$ and $\boldsymbol{S}_0$ represent a set of the state transition sequence $\boldsymbol{q}$ and the sequence of mixture's index $\boldsymbol{m}$ as $\boldsymbol{S}_1 = \{\boldsymbol{q}_1, \boldsymbol{m}_1\}$, $\boldsymbol{S}_0 = \{\boldsymbol{q}_0, \boldsymbol{m}_0\}$. Equation(9) implies that we must search $\boldsymbol{O}_1$, which maximize the likelihood Pr for given $\boldsymbol{O}_0$ under all combinations of $\boldsymbol{S}_1$ and $\boldsymbol{S}_1$. In practice, it seems that $Pr(\boldsymbol{O}_1|\boldsymbol{O}_0)$ is dominated by the value of $Pr(\boldsymbol{O}_1, \boldsymbol{S}_1, \boldsymbol{S}_0|\boldsymbol{O}_0)$, in which $\boldsymbol{S}_0$ represents $\boldsymbol{O}_0$ best. Therefore, the following problem is solved instead,

$$\underset{\boldsymbol{O}_1, \boldsymbol{S}_1, \boldsymbol{S}_0}{\text{argmax}} \ Pr(\boldsymbol{O}_1, \boldsymbol{S}_1, \boldsymbol{S}_0|\boldsymbol{O}_0). \quad (10)$$

Since the above maximization is still intractable, we further rewrite Eq.(10) using the conditional independence,

$$\begin{aligned} &Pr(\boldsymbol{O}_1, \boldsymbol{S}_1, \boldsymbol{S}_0|\boldsymbol{O}_0) \\ &= Pr(\boldsymbol{O}_1, \boldsymbol{O}_0|\boldsymbol{S}_1, \boldsymbol{S}_0) \frac{Pr(\boldsymbol{S}_1, \boldsymbol{S}_0)}{Pr(\boldsymbol{O}_0)} \\ &= Pr(\boldsymbol{O}_1|\boldsymbol{S}_1) Pr(\boldsymbol{S}_1|\boldsymbol{S}_0) Pr(\boldsymbol{S}_0|\boldsymbol{O}_0). \end{aligned} \quad (11)$$

Hence, instead of the direct maximization of Eq.(10), Eq.(11) is maximized step by step from the right term on RHS.

First, the maximization of $Pr(\boldsymbol{S}_0|\boldsymbol{O}_0)$ can be seen as the viterbi decoding. Therefore, the input signal is decoded using the model PHMM-A and then the most probable state transition sequence is obtained. Second term $Pr(\boldsymbol{S}_1|\boldsymbol{S}_0)$ is maximized when the same state transition sequence is selected for given $\boldsymbol{S}_0$. The remaining problem is a maximization of $Pr(\boldsymbol{O}_1|\boldsymbol{S}_1)$. Since the set of $\boldsymbol{\lambda}_1$, $\boldsymbol{q}_1$ and $\boldsymbol{m}_1$ is given in the preceding step, it is clear that the most probable $\boldsymbol{O}_1$ can be obtained as a sequence of the mean vectors of each selected state. However, this results in a discontinuous image since no information between pixels is taken into account.

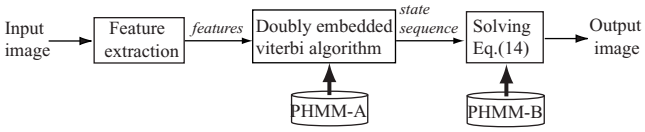

Figure 8: Details of the transform.

3.4. Maximization with inter-pixel information

In order to obtain smooth results, maximization of the likelihood considering some neighboring pixels is required. Recall that Gaussian, Laplacian and derivatives are included in the feature vectors of the input image; they retain information on eight neighbors of the target pixel.

Let $\boldsymbol{p}_t$ be the target pixel values to be obtained. The feature vector $\boldsymbol{O}_{1t}$ at a position t of the image is

$$\boldsymbol{O}_{1t} = [\boldsymbol{p}_t^T \;\; \boldsymbol{G}^T(\boldsymbol{p}_t) \;\; \boldsymbol{L}^T(\boldsymbol{p}_t) \;\; \boldsymbol{H}^T(\boldsymbol{p}_t) \;\; \boldsymbol{V}^T(\boldsymbol{p}_t)]^T, \quad (12)$$

where $\boldsymbol{G}(\cdot)$, $\boldsymbol{L}(\cdot)$, $\boldsymbol{H}(\cdot)$ and $\boldsymbol{V}(\cdot)$ represent Gaussian, Laplacian, horizontal derivative and vertical derivative operations, respectively. Therefore, the problem here becomes the estimation of the image $\boldsymbol{p}= [\boldsymbol{p}_1^T \;\; \boldsymbol{p}_2^T \;\; \cdots \;\; \boldsymbol{p}_\tau^T]^T$, which maximizes the likelihood $Pr(\boldsymbol{O}_1|\boldsymbol{S}_1)$ under the constraint $\boldsymbol{O}_1 = \boldsymbol{W}\boldsymbol{p}$, where the matrix $\boldsymbol{W}$ consists of the coefficients of Gaussian, Laplacian, horizontal derivative and vertical derivative kernels. The log likelihood function to be maximized can be written using the constraint,

$$\begin{aligned}\log P(\boldsymbol{O}_1|\boldsymbol{m}_1, \boldsymbol{q}_1) = &\sum_{j,m\in\boldsymbol{m}_1} \log c_{j,m}^{(k)} \\ + \sum_{i,j\in\boldsymbol{q}_1} \log a_{i,j} + &\sum_{k,i,j\in\boldsymbol{q}_1} \log a_{i,j}^{(k)} - \frac{N_f\tau}{2}\log 2\pi \\ -\frac{1}{2}(\boldsymbol{W}\boldsymbol{p}-\boldsymbol{\mu}_1)^T&\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{W}\boldsymbol{p}-\boldsymbol{\mu}_1) - \frac{1}{2}\log|\boldsymbol{\Sigma}_1| \end{aligned} \quad (13)$$

where $\boldsymbol{\mu}_1$ represents a mean vector, which is constructed by arranging the means of selected states according to $\boldsymbol{q}_1$ and $\boldsymbol{m}_1$. $\boldsymbol{\Sigma}_1$ is a diagonal covariance matrix whose diagonal elements consist of the variances of selected states. N_f denotes the dimension of the feature vector. Since Eq.(13) is in a quadratic form of $\boldsymbol{p}$, it is maximized when

$$\frac{\partial \log Pr(\boldsymbol{O}_1|\boldsymbol{m}_1, \boldsymbol{q}_1)}{\partial \boldsymbol{p}} = \boldsymbol{0}. \quad (14)$$

It can be seen that Eq.(14) is a linear equation with respect to the pixels of the target image $\boldsymbol{p}$. The actual procedure is given in Fig.8.

4. EXPERIMENTAL RESULTS

In this section, we show some results of the face-to-face transform. The topology of the PHMM used in the experiment is as follows; the number of super states, embedded states and mixtures are 8, 10 and 10, respectively. In Fig.9, results of the experiment are illustrated. These figures validate the proposed algorithm. Even though there is no tongue in the training images, the model is rich enough to generate the tongue image. It is interesting to see that the lip part of the model deforms to represent the tongue.

Figure 9: Results of the proposed system. Left column: training images. Top row: the face images to be transformed. Middle and bottom rows: transformed results.

5. CONCLUSION

In this paper, the PHMM-based face-to-face transform system has been presented. The proposed approach is based on appearance models of faces, which are acquired and represented by PHMMs. The experimental results show the effectiveness of the proposed method.

Although the consideration of big head motions is left for feature research, small head motions can be handled by the proposed method since the model is based on HMMs.

6. REFERENCES

[1] S.-S.Kuo, and O.E.Agazzi, "Keyword Spotting in Poorly Printed Documents Using Pseudo 2-D Hidden Markov Models", *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol.16, no8, pp.842–848, Aug. 1994.

[2] A.V.Nefian, and M.H.Hayes III, "An Embedded HMM - Based Approach for Face Detection and Recognition", in Proc. of *Int. Conf. on Acoust. Speech and Signal Proc.*, vol.6, pp.3553–3556, May 1999.

[3] T.Goto, S.Kshirsagar, and N.M.Thalmann "Automatic Face Cloning and Animation", IEEE Signal Processing Magazine, Vo. 18, No. 3, pp 17-25, May, 2001.

[4] E.Cosatto, and H.P.Graf, "Photo-Realistic Talking-Heads from Image Samples", *IEEE Trans. on Multimedia*, vol.2, no3, pp.152–163, Sept. 2000.

[5] F.D.Torre, and M.J.Black, "Dynamic coupled component Analysis", Proc. of *Computer Vision and Pattern Recognition*, Vol.II, pp.643-650, Dec. 2001.

[6] R.Lienhart, and J.Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", Proc. of *IEEE Int. Conf. on Image Process.*, Vol.1, pp.900-903, Sep. 2002.

[7] B.Moghaddam, and A.Pentland, "Face Recognition using View-Based and Modular Eigenspaces", *Automatic Systems for the Identification and Inspection of Humans*, SPIE, Vol.2277, July 1994.