COMPONENT-WISE POSE NORMALIZATION FOR POSE-INVARIANT FACE RECOGNITION

Shan Du and Rabab Ward

Department of Electrical and Computer Engineering
The University of British Columbia, Vancouver, BC, Canada
{shand, rababw}@ece.ubc.ca

ABSTRACT

The pose variation involved in facial images significantly degrades the performance of face recognition systems. In this paper, a component-wise pose normalization method for facilitating pose-invariant face recognition is proposed. The main idea is to normalize a non-frontal facial image to a virtual frontal image component by component. In this method, we first partition the whole non-frontal facial image into different facial components and then the virtual frontal view for each component is estimated separately. The final virtual frontal image is generated by integrating the virtual frontal components. The proposed method relies only on 2D images, therefore complex 3D modeling is not needed. The experimental results using the CMU-PIE database demonstrate the advantages of the proposed method.

Index Terms: pose-invariant face recognition, component-wise pose normalization, virtual view generation, face recognition.

1. INTRODUCTION

Face recognition has attracted much attention due to its wide applications in commerce, law enforcement and other areas. Existing work in face recognition has demonstrated good recognition performance on frontal, expressionless views of faces with controlled lighting conditions. However, a practical face recognition system needs to work under differing imaging conditions, such as differing poses, expressions, and illumination changes. In this paper, we focus on face recognition under differing pose conditions.

It is not difficult for human beings to recognize the same individual in varying poses. However, for automated systems, this is a difficult task because the differences in images of two varied poses of a person would be more significant than the differences between two distinct persons in the same pose.

When the face is rotated in the image plane, it can be normalized by detecting at least two facial features. However, when the face is subjected to in-depth 3D rotation, simple geometrical normalization is not possible. Many approaches have been proposed for automated face recognition under 3D rotation. Having multi-view images stored in the gallery is one strategy for dealing with the pose-variant problem, and is a direct extension of frontal face recognition. An algorithm of this type is presented in [1]. In [2], the popular eigenface approach is extended to handle multiple views. The authors compare the performance of a parametric eigenspace (computed using all views from all subjects) with view-based eigenspaces (separate eigenspaces for each view). In the experiments, the view-based eigenspaces outperformed the parametric eigenspace. Other approaches that use 2D model-based multi-view algorithms have been proposed for face tracking across significant pose changes. In [3], separate active appearance models are trained for profile, half profile and frontal views.

Another popular solution is to generate virtual views. A generic 3D model of the human face can be used to predict the appearance of a face under different pose parameters [4][5]. Once a 2D face image is texture mapped onto the 3D model, the face can be treated as a traditional 3D object in computer graphics, undergoing 3D rotations. In 3D methods, we need to construct a precise 3D model of the face from the current image, which will require many techniques such as active camera calibration, feature points selection/detection, correspondent points labeling from different views, 3D model translation, rotation and projection, and a database of 3D heads. The enormous computational complexity involved may preclude it from becoming a real-world application.

There are also some 2D example-based view synthesis methods that can generate virtual views under multiple poses. [6] propose an algorithm to synthesize novel views from a single image by using prior knowledge of the facial images and apply them to face recognition. In this method, prior face knowledge is represented by 2D views of prototype faces. The underlying assumption of the method is that the 3D shape of an object (and 2D projections of 3D objects) can be represented by a linear combination of prototype objects. It follows that a rotated view of the object is a linear combination of the rotated views of the prototype objects. The so-called Linear Object Classes (LOC) idea is used to synthesize rotated views of facial images from a single example view. In LOC, a facial image is first separated into shape vector and texture vector, and then LOC is applied to them respectively. The virtual "rotated" images are then easily generated using a base set of 2D prototypical views. The synthesized virtual views are highly dependent on the correspondence between the images. However, building accurate pixel-wise correspondence between facial images is a difficult problem.

In [7], a Local Linear Regression (LLR) method, which starts from the basic idea of LOC, is proposed. The authors show that, in the case where the given samples are well aligned, there exists an approximate linear mapping between two images of one person captured under variable poses. This mapping is consistent for all persons, if their facial images are aligned in a pixel-wise manner. Unfortunately, pixel-wise correspondence between images is still a challenging problem. In most real-world face recognition systems, facial images are only coarsely aligned, based on very few facial landmarks, such as the two eye centers. In this case, the above-mentioned assumption of linear mapping no longer holds

theoretically, since it becomes a complicated nonlinear mapping. LLR proposes that by partitioning the whole surface of the face into multiple uniform blocks, the linearity of the mapping for each block is increased because of a consistent normal and better control over alignment.

In this paper, we start from the basic idea of LOC and LLR, that is, we try to generate the frontal view of a given non-frontal facial image based on the 2D prototypes in a training set with corresponding image pairs of some specific poses. However, unlike previous methods, we apply the generation algorithm on multiple facial components rather than on separate shape and texture vectors (LOC) or on uniform image blocks (LLR). Accurate dense correspondence between facial images is not required; what we need is just a coarse alignment based on the two eye centers. In the proposed method, the whole non-frontal facial region is partitioned into multiple facial components where different normalization parameters are applied to different components for the generation of their frontal counterpart.

The remainder of this paper is organized as follows: Section 2 describes the framework of the proposed method. In Section 3, the pose alignment algorithm is discussed in detail. The performance evaluation of the proposed method is presented in Section 4. Section 5 concludes the paper.

2. FRAMEWORK OF THE PROPOSED APPROACH

We focus our attention on developing a technique for synthesizing images that are different from the viewing positions of the sample model images, using only 2D views and information derived from prototype faces. Our motivation for using the example-based approach lies in its potential as a simple alternative to the more complicated 3D model-based approach.

The idea of segmenting an image into patches was inspired by LLR. For the case of coarse alignment, the developers of LLR considered that the corresponding local facial regions of the frontal and non-frontal view pair satisfy the linear assumption much better than the whole facial region (GLR). From the viewpoint of LOC, we can understand that estimating the linear combination coefficients will be easier and more accurate using small patches rather than the whole image.

Distinct from LLR, which segments an image into uniform blocks, we partition images according to the facial components positions. The reason for this innovation lies in our observation that using uniform blocks may break facial components into pieces. Moreover, the size of the blocks is not easy to select. If too large, the resulting image is blurred. If too small, the coarse cross-pose correspondence may be meaningless, resulting in many annoying artifacts.

Our method has two merits. First, the patches are more meaningful than in LLR, and therefore the establishment of coarse cross-pose correspondence is easier. The sizes of patches are neither too large nor too small. They are only related to the image size and are assigned automatically; no manual selection of the block size is needed, as in LLR does. Also the patch size is not uniform - different components have different sizes. Second, we do not break facial components into pieces. Thus, the blocking artifacts introduced by the block-based method will not ruin those facial components that are more important than others in face recognition.

The procedure for the proposed method is as follows (see Figure 1):

1. The images are segmented into facial components, including the training set and the probe image.
2. For each component patch,
   (a) the linear combination coefficients of the probe's non-frontal patch in terms of the training non-frontal patches are computed.
   (b) the virtual frontal patch is generated using the above coefficients and the training frontal patches.
3. The virtual frontal patches are integrated to form the virtual frontal image.

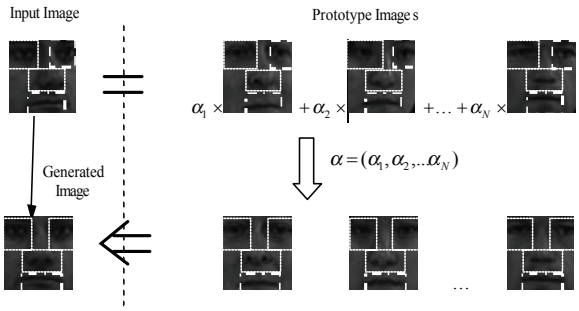

Figure 1. Component-based virtual frontal view generation

3. COMPONENT-WISE POSE ALIGNMENT

Given a non-frontal facial image, our aim is to generate its virtual frontal view based on a training set. Simply speaking, we first represent the given non-frontal image using a linear combination of the training non-frontal images. Then, using the linear combination coefficients and the training frontal images, we generate the virtual frontal view of the given image. Because our method is component-based, we represent each component patch of the given non-frontal image using a linear combination of the corresponding training non-frontal patches. Finally, using the linear combination coefficients and the corresponding training frontal patches, we generate the virtual frontal patch.

Estimating the linear combination coefficients for each patch becomes much easier and more accurate than estimating the global one because of the much lower dimension of the patches. This is factor especially important when the given training set is of limited size.

3.1. Component Segmentation

Because the positions of the two eyes' are already known, we can use them to roughly segment facial images into facial components, as shown below.

Let (x_l, y_l) and (x_r, y_r) be the coordinates of the two eyes' centers. The distance between the two eyes is $d = x_r - x_l$. Please note that we do not consider the difference between y_l and y_r. Even though they are not equal, the difference between them is much smaller compared with the difference between x_l and x_r, and can therefore be ignored.

Normally, the distance between the inner corners of the two eyes is similar to the length of one eye. Therefore, we can use this information to segment a face (see Figure 2).

The facial image is segmented into seven different-sized patches. Each patch contains one facial component, e.g., eye, the area between the eyes, nose, mouth, cheek.

As shown in Figure 2, moving from left to right and up to down, the sizes of the seven patches are calculated as follows:

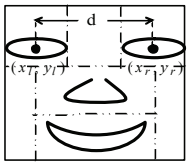

Figure 2. Component segmentation

1. $\left(1 : x_l + \frac{1}{4}d,\quad 1 : y + \frac{1}{4}d\right)$
2. $\left(x_l + \frac{1}{4}d + 1 : x_r - \frac{1}{4}d,\quad 1 : y + \frac{1}{4}d\right)$
3. $\left(x_r - \frac{1}{4}d + 1 : w,\quad 1 : y + \frac{1}{4}d\right)$
4. $\left(1 : x_l,\quad y + \frac{1}{4}d + 1 : h\right)$
5. $\left(x_l + 1 : x_r,\quad y + \frac{1}{4}d + 1 : y + \frac{1}{4}d + 1 + \frac{h - (y + \frac{1}{4}d)}{2}\right)$
6. $\left(x_l + 1 : x_r,\quad y + \frac{1}{4}d + 1 + \frac{h - (y + \frac{1}{4}d)}{2} + 1 : h\right)$
7. $\left(x_r + 1 : w,\quad y + \frac{1}{4}d + 1 : h\right)$

where w and h are the width and height of the image; $y = \max(y_l, y_r)$.

Using this segmentation, we can avoid breaking facial features into pieces. In describing the following experiments, we will show the merit of this aspect of our procedure. Moreover, since this segmentation directly results in meaningful patches, the rough cross-pose correspondence is established automatically. Figure 3 shows the component segmentation on images with different poses.

3.2. Coefficients Estimation

Let $\left\{\Phi^{p_0}, \Phi^{p_k}\right\}$ be the training set of one component patch, where Φ^{p_0} denotes the frontal view composed of N subjects $\left\{x^{p_{0,1}}, x^{p_{0,2}}, ..., x^{p_{0,N}}\right\}$, and $\Phi^{p_k} = \left\{x^{p_{k,1}}, x^{p_{k,2}}, ..., x^{p_{k,N}}\right\}$ is the corresponding non-frontal view under pose p_k. Note that $x^{p_{k,i}}$ is the counterpart of $x^{p_{0,i}}$ from the same person but with different poses.

Following the LOC theory, we can use "prototypical" 2D views and their known transformations to synthesize an operator that will transform a 2D view into a new 2D view when the object is a linear combination of the prototypes.

$$x^{p_k} = \sum_{i=1}^{N} \alpha_i x^{p_{k,i}} \tag{1}$$

$$x^{p_0} = \sum_{i=1}^{N} \alpha_i x^{p_{0,i}} \tag{2}$$

The decomposition of a given view x^{p_k} in (1) and the composition of the new view in (2) can be understood as a single linear transformation. First, we compute the coefficients α_i for the optimal decomposition (in the sense of least square). The given view is decomposed into the "example" N given prototypes by minimizing

$$\left\| x^{p_k} - \sum_{i=1}^{N} \alpha_i x^{p_{k,i}} \right\|^2 \tag{3}$$

We rewrite (3) as $x^{p_k} = \Phi^{p_k} \alpha$, where Φ^{p_k} is the matrix formed by the N vectors $x^{p_{k,i}}$ arranged column-wise, and α is the column vector of the α_i coefficients. Minimizing (3) gives

$$\alpha = (\Phi^{p_k})^{+} x^{p_k} \tag{4}$$

Then the new view x^{p_0} is given by

$$x^{p_0} = \Phi^{p_0} \alpha = \Phi^{p_0} \Phi^{p_k}{}^{+} x^{p_k} \tag{5}$$

and thus can be learned from the 2D example pairs (Φ^{p_0}, Φ^{p_k}), where

$$\Phi^{p_k}{}^{+} = (\Phi^{p_k}{}^{T} \Phi^{p_k})^{-1} \Phi^{p_k}{}^{T} \tag{6}$$

Figure 3. Component segmentation on images with different poses

3.3. Virtual Generation

The virtual frontal view can be obtained using Equation (5). After all virtual frontal components are generated, they are integrated to form the virtual frontal image.

4. EXPERIMENTS

We conducted experiments on the 5 pose subsets of the CMU-PIE database, which includes pose 29, 05 (turning left and right at 22.5 degrees), 11, 37 (turning left and right at 45 degrees), and 27 (near frontal) [8]. The generation of the virtual frontal views used the leave-one-out strategy. In the final face recognition experiment, a total of 68 subjects were used with the frontal facial images (pose 27) forming the gallery, while the non-frontal facial images were used as probes to match against the frontal images in the gallery.

We implemented four different recognition modes: without preprocessing (i.e., using the original non-frontal image directly as input), the global generation method (GLR) [7], the local generation method with uniform blocks (LLR) [7], and our proposed component-based generation method.

4.1. Virtual View Generation (Visual Quality)

In Figure 4, we show some examples of virtual frontal view generation results. Column (a) shows the input non-frontal images, column (b) the virtual frontal view generated by the global method GLR, columns (c) and (d) the results produced by the local method LLR with different block sizes, and column (e) the results generated by the component-based method. The last column shows the real frontal faces.

From these results, we can see that virtual generation using GLR is somewhat blurred; LLR can generate better results. In [7], the authors obtained the best results with block size 20×20.

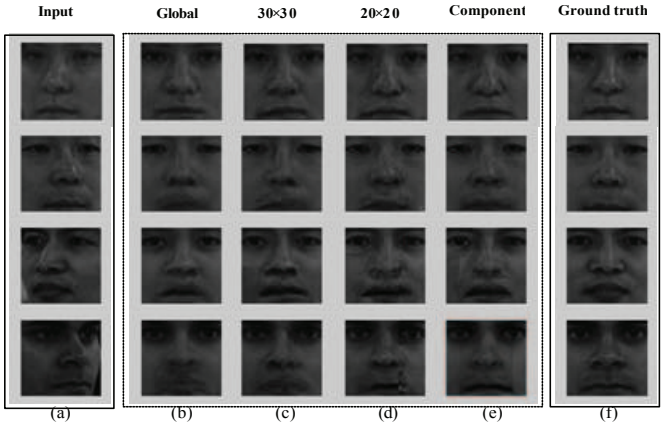

Figure 4. Examples of virtual frontal view generation results

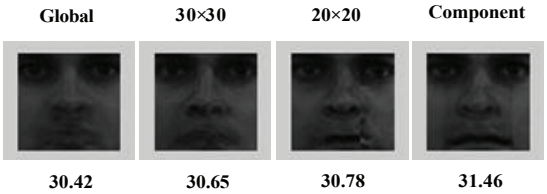

Figure 5. PSNR for virtual images

With the block size reduced to 10×10, the results became worse due to blocking artifacts. Compared with LLR, our method can generate a smoother image with fewer blocking artifacts. Most important, the facial components are not broken into pieces.

4.2. Peak Signal-to-Noise Ratio

To evaluate the virtual generation accuracy quantitatively, we compute the Peak Signal-to-Noise Ratio (PSNR) value of the generated image with the ground truth frontal image. The PSNR is calculated by

$$PSNR = 10 \times \log_{10} \frac{255 \times 255}{\frac{1}{wh}\sum_{i=1}^{w}\sum_{j=1}^{h}\left[I(i,j) - \hat{I}(i,j)\right]^2} \qquad (7)$$

where $I(i,j)$ is the ground truth frontal image, $\hat{I}(i,j)$ is the generated virtual frontal image, and w and h are the width and height of the image, respectively.

Figure 5 shows the PSNR values of different generated images. Our proposed method generated the best image.

4.3. Pose-invariant Face Recognition using Virtual Views

In this section, we describe the pose-invariant face recognition experiments we carried out on the virtual frontal views to evaluate the proposed algorithm.

We implemented four different recognition modes: without preprocessing (original), the global generation method (GLR) [7], the local generation method with uniform blocks (LLR) [7] (with block sizes of 30×30 and 20×20), and our proposed component-based generation method. From Figure 6, it can be seen that our method outperformed others. In Table 1, we also show the comparison of our method with the eigen light-field (ELF) method [9] that is well known for recognizing faces across pose and achieving good performance. Our method outperformed it.

5. CONCLUSIONS

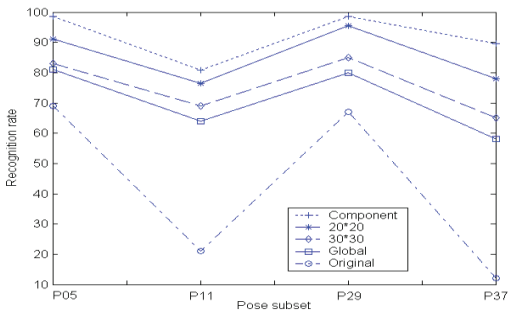

Figure 6. Recognition rate comparison

Table 1. The performance comparison between our method and other methods.

Methods	P05	P11	P29	P37
ELF [9]	88%	76%	86%	74%
LLR (20×20) [7]	91.2%	76.5%	95.6%	77.9%
Our method	98.5%	80.9%	98.5%	89.7%

In this paper, we proposed a component-based pose normalization method for pose-invariant face recognition. The effectiveness of the proposed method was evaluated by face recognition experiments on the CMU-PIE database. The experimental results showed that partitioning facial images into facial components is more meaningful than partitioning them into uniform blocks, resulting in better pose normalization results in both visual quality and recognition rate.

6. REFERENCES

[1] David J. Beymer, "Face recognition under varying pose," A. I. Memo No. 1461, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1993.

[2] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 84-91, 1994.

[3] T. Cootes, G. Wheeler, K. Walker, and C. Taylor, "View-based active appearance models," Image and Vision Computing, 20: 657-664, 2002.

[4] V. Blanz, P. Grother, P. J. Phillips, and T. Vetter, "Face recognition based on frontal views generated from non-frontal Images," IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 454-461, 2005.

[5] Volker Blanz and Thomas Vetter, "Face recognition based on fitting a 3D morphable model," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, pp. 1063-1074, 2003.

[6] Thomas Vetter and Tomaso Poggio, "Linear object classes and image synthesis from a single example image," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 733-742, 1997.

[7] Xiujuan Chai, Shiguang Shan, Xilin Chen, and Wen Gao, "Local linear regression (LLR) for pose invariant face recognition," Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR'06).

[8] T. Sim, S. Baker and M. Bsat, "The CMU Pose, Illumination, and Expression Database," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 12, pp. 1615-1618, 2003.

[9] R. Gross, I. Matthews, and S. Baker, "Appearance-based face recognition and light-fields," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 26, no. 4, pp. 449-465, 2004.