Efficient Face Frontalization in Unconstrained Images

Mallikarjun B R
CVIT, IIIT Hyderabad, India
Email: mallik.jeevan@gmail.com

Visesh Chari
Visiting Faculty, CVIT, IIIT Hyderabad, India
Email: visesh@gmail.com

C.V. Jawahar
CVIT, IIIT Hyderabad, India
Email: jawahar@iiit.ac.in

***Abstract*—Face *frontalization* is the process of synthesizing a frontal view of a face, given its non-frontal view. Frontalization is used in intelligent photo editing tools and also aids in improving the accuracy of face recognition systems. For example, in the case of photo editing, faces of persons in a group photo can be corrected to look into the camera, if they are looking elsewhere. Similarly, even though recent methods in face recognition claim accuracy which surpasses that of humans in some cases, performance of recognition systems degrade when profile view of faces are given as input. One way to address this issue is to synthesize frontal views of faces before recognition.**

We propose a simple and efficient method to address the face frontalization problem. Our method leverages the fact that faces in general have a definite structure and can be represented in a low dimensional subspace. We employ an exemplar based approach to find the transformation that relates the profile view to the frontal view, and use it to generate realistic frontalizations. Our method does not involve estimating 3D model of the face, which is a common approach in previous work in this area. This leads to an efficient solution, since we avoid the complexity of adding one more dimension to the problem. Our method also retains the structural information of the individual as compared to that of a recent method [4], which assumes a generic 3D model for synthesis. We show impressive qualitative and quantitative results in comparison to the state-of-the-art in this field.

I. Introduction

Facial analysis in images for recognition/manipulation is a widely addressed and commercially important problem. Its applications range from surveillance to automatic tagging of photos on social websites. Recently, there are papers producing convincing results on *in-the-wild* datasets [4, 18]. These datasets differ from previous ones in their unconstrained nature of image capture. However such methods have two drawbacks. Firstly, a lot of these methods have degraded performance in profile view vs frontal view. Secondly, they require lot of training data [5]. One way to alleviate both problems is to be able to generate realistic frontal view faces for any person. This can be achieved, because faces have a definite structure. Eigen analysis [1], for example, has shown that faces exist in low dimensional sub spaces and can be represented as linear combinations of other faces. Also, it has been shown earlier that many face characteristics like expressions, hair etc. can be *transferred* from one person to another, in a very realistic manner [11].

In this paper, we show that a pre-processing step of synthesizing frontal pose of the face significantly improves the accuracy of face recognition. Face frontalization is the process of synthesizing frontal pose of the face, given a profile view of the face as shown in Figure 1. This step helps in simplifying the task of face recognition as recognition systems have more information and less occlusion to work with. Few methods counter this *frontalization* problem, by choosing to extract features only at the salient locations. Unfortunately, this leads to loss of structural relation between various parts of the face. However, as we will show in this paper, our method preserves this information as well.

978-1-4673-8564-0/15/$31.00 ©2015 IEEE

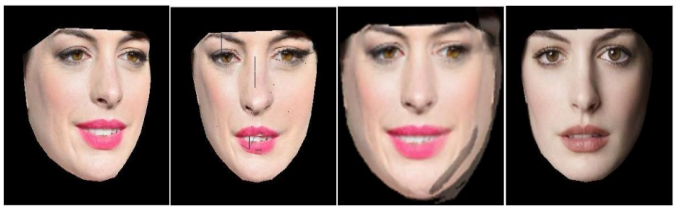

Fig. 1: Left image shows the profile face. Second image is *face frontalized* by our method. Third image is of Hassner *et al.* [4] method. Right image is the natural frontal view of the individual. *Frontalization* helps in face recognition.

Apart from aiding face recognition systems, frontalization techniques can also be used to generate a video out of a single image and can find applications in animation [11]. For example, if a family photograph has some people looking away from the camera, our approach can be used to correct this discrepancy [10].

Recent methods [13] [8] have proposed different ways of addressing the challenging problems of pose variations in images. Simonyan *et al.* [8] [12] choose to define features extracted out of large image regions to counter mis-alignments. Wolf *et al.* [6] [16] choose to align faces before extracting features. Sun *et al.* [5] use large datasets to create models robust to these challenges. In line with our approach, some recent works try to counter these challenging conditions of pose variation by synthesizing pose neutral faces from input images. Taigman *et al.* [14] try to estimate a 3D model of each input image. They then use this 3D information to synthesize the frontal view. On the other hand, [4] assumes a generic 3D model for all input images and produces convincing frontalization results. Even though the approach of [14] seems to be good, estimating 3D model from a single image is a hard problem. And assuming a generic 3D model in [4], leads to loss of structural information unique to an individual. Thus, in this work, we turn towards an exemplar based approach to fill the 3D information gap required by the previous approaches.

Lately, we have seen a surge of papers [9] [15] based on exemplar methods for solving computer vision problems. In these type of approaches, exemplars of the problem category are used instead of defining a generic model to solve the the problem at hand. For example, in the case of object detection, [9] trains a set of models using one positive exemplar each, instead of all the training set. And they show

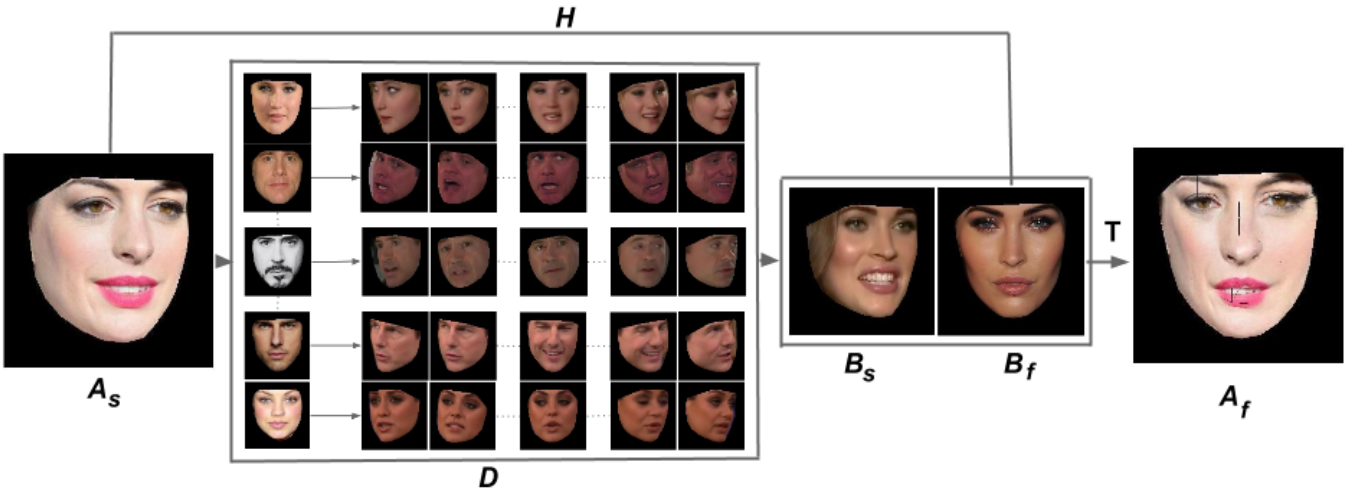

Fig. 2: Figure shows the generic pipeline used in our approach. Given the input image (left most block) we use the exemplar database (second block) to compute the nearest profile view (third block, first image). We then use the correspondences between the profile and frontal views of the selected exemplar pair (third block) to compute the affine transformation H between the input image and the frontal exemplar, and use it to produce the *frontalized output* (right most block).

that the ensemble of such models give surprisingly good generalization. Similarly, our method is based on an exemplar based approach toward face frontalization. Consider a huge dataset of profile, frontal view face pairs of different. Chances of finding individuals with similar face structures to an input profile image is thus very high. Given such a match, the frontal view of the person in the database can then be used to frontalize the input image. Therefore, for our method, we collect a database of profile views and corresponding frontal view of a large number of individuals. We leverage the fact that faces lie in a low dimensional subspace and thus, many characteristics, like pose, expressions, etc. are transferable between people.

II. Face Frontalization

Our method takes as input, a profile view face, A_p, and an exemplar database, D, consisting of wide range of profile, frontal pose pairs for different persons. We then proceed to frontalize the face in two steps. First, we run facial landmark detection [7] on the input face, and using it we retrieve the most similarly posed face I_p^i and its corresponding frontal view face I_f^i, from database D. When profile views of two faces match, there is high likelihood that the two persons have similar facial structures. We exploit this property to get geometrical transformations required for frontalization of the input face. Simply put, we obtain the frontal view of A_p by using the affine transformations between A_p and I_f. One recently proposed state-of-the-art method [4] uses a generic 3D model for computing this transformation. This leads to loss of important discriminate structural information unique to an individual. Since we are finding a nearest profile exemplar and its corresponding frontal view face, structural information is still preserved for an individual face in our case.

Let $P = (X, Y)$, denote the landmark locations on the face, where $X = (x_1, x_2, .., x_{68})$ and $Y = (y_1, y_2, .., y_{68})$ are vectors of X and Y coordinates respectively. We consider 68 landmarks which includes feature points such as eye corners, nose tip, mouth line and jaw line. We use the dlib [7] implementation for landmark localization. Note that landmarks of images in D have been pre-computed. We also manually predefine 110 planes for a face using these landmarks. For example, the ends of two eyebrows and the beginning of the nose form a plane (see Figure 4). This has to be done only once since these planes connect the same fiducial points irrespective of the face. Each plane is defined by 3 landmark locations.

Let $T = \{t_1, t_2, \ldots, t_{110}\}$ represent the set of planes defined in 3D for a face image. Given planes of one profile-frontal image pair $T_p^m \& T_f^m$, in D, we define $H^m = \{H_1^m, H_2^m, \ldots, H_{110}^m\}$ as the affine transformations between corresponding planes, computed using point correspondences from the facial landmarks. That is,

$$t_f^{m,i} = t_p^{m,i} \times H_i^m \tag{1}$$

where the subscript p denotes profile, and the subscript f denotes frontal views.

Given the landmarks P for images in the database D, we now proceed to frontalize using the following steps.

A. Nearest Exemplar Selection

To retrieve the closest exemplar to the input face, we first need to define a similarity measure between faces. Let P^m and P^n represent landmarks of two faces. The similarity score between poses of two faces can then be defined as the Euclidean distance between P^m and P^n.

$$ds^{mn} = \sqrt[2]{\sum_{i=1}^{68}((x_i^m - x_i^n)^2 + (y_i^m - y_i^n)^2)} \tag{2}$$

However P^m and P^n are defined in different coordinate systems, separated by translation, rotation and scaling. We need to nullify the effect of translation and scaling and bring both sets of landmark positions to one coordinate system. Note that rotation is not considered as it is one of the parameters of pose and our exemplar database is exhaustive enough to take care of rotation variations in the input face. To remove the translational

Fig. 3: First row of images are the input profile images. Second row shows the retrieved faces from database.

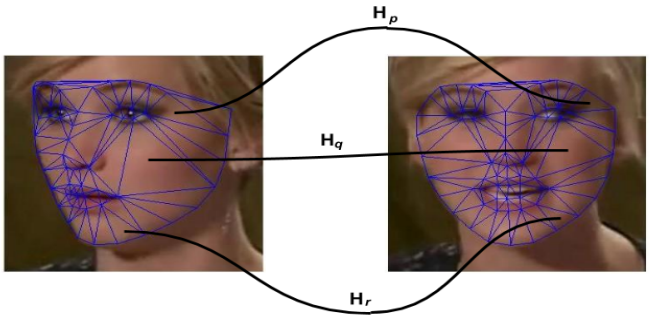

Fig. 4: Image shows the planes represented as triangles and correspondences between two views of the same face. Note that each plane contains a fixed set of points irrespective of pose. For example, one plane contains two ends of the eyebrows and the top of the nose.

effect, we subtract the mean of X and Y coordinates from both the landmark vectors, $X = (X - \mu_X), Y = (Y - \mu_Y)$. To remove the scaling effect, we multiply P_m by a factor of s, given by

$$s = \frac{\sum_{i=1}^{68}((x_i^m \times x_i^n) + (y_i^m \times y_i^n))}{\sum_{i=1}^{68}((x_i^m)^2 + (y_i^m)^2)} \tag{3}$$

which follows from a straightforward optimization procedure that minimizes *root mean squared error* (rmse) error between corresponding landmark positions. The derivation is omitted for brevity.

Pose is accurately defined by the position of landmarks on the face. We concatenate the landmark locations obtained on the input image into a single vector called the pose vector. We then use Euclidean distance as the metric of comparison to retrieve the most similarly posed face from the database D. To get an accurate measure, we convert the pose vector of input face to exemplar face coordinate system. Let P^t represent the landmarks of profile input face and P_p^i represent the landmarks of profile exemplars available in the database. The nearest exemplar is the one which has the least ds^{ti}.

$$i^* = \arg\min_i d^{ti} \tag{4}$$

Given the nearest profile image I_p^i, we retrieve its frontal image and pose I_f^i, P_f^i. The first row of Figure 3 shows sample input faces and second row shows the nearest exemplars retrieved from D. Observe that men and women have slightly different facial structure, and this captured by our method, since women are retrieved as top exemplar candidates for input images of women.

B. Triangulation and Transformations

Once the frontal view of the nearest exemplar is obtained, we need to transform the input profile face to a frontal view. To do this, we first *transfer* correspondences between the exemplar pairs to the input image. This is done by replacing positions of the profile exemplar landmarks with those of the input image. Using the landmarks obtained, we define around 110 triangles on the face, each of which can be considered as a plane in the face coordinate system. Since the triangles are defined based on particular set of landmarks, we have correspondences between planes in the input image and corresponding frontal view exemplar. We obtain the affine transformations between the corresponding planes and then synthesize the frontal view of the input image using these transformations.

Figure 2 pictorially represents our method. For a given profile view input face, A_p, we retrieve most similar exemplar, I_p^i from the D along with its corresponding frontal view face, I_f^i. We then compute affine transformations, H^i, to transform planes of A_p to generate its frontal view.

C. Face Recognition

Our face recognition pipeline is based on the framework of Write *et al.* [17] who claim that faces of a particular individual lie in a low-dimensional subspace. In their method, training samples are represented as a 2D matrix, where each column represents a feature extracted from one training image. The input test sample should then be represented as linear combination of samples from the corresponding training samples of the same class (person). This problem has to be posed as an $l0$ minimization problem as it selects a combination of samples from training set. Based on recent advancement in sparse representation and compressed sensing, the authors claim that when the solution is sparse enough, solving $l1$ minimization is equivalent to the $l0$ minimization problem. Using this insight, a solution can be obtained in polynomial time using linear programming models. We use their implementation as the basis for our experiments, while we train using our dataset.

III. Experiments and Results

A. Exemplar Database

For the exemplar database, we collected various face poses for 22 individuals (11 male and 11 female) from the talk shows available online. We selected sections which have complete swing of pose and expression changes. Approximately 15 exemplars and a frontal view were selected per individual. In total of around 400 exemplars and 22 frontal view faces were collected. For face recognition experiment, we collected approximately 50 training and 50 input faces of 6 celebrities online. We call this dataset the $PoseInTheWildFaceDataSet$ (PIWFDS). We consider a new dataset, as existing datasets do not contain profile-frontal image pairs and even state-of-the-art recognition systems perform poorly on it.

All our experiments were conducted using MATLAB. We used HoG [3] feature based face detector to find faces and

Fig. 5: First row shows the output of our method and the second row is of Hassner *et al.* [4] for LFPW [2] dataset. Observe ghost like appearances, structure distortion, mirroring effects in Hassner *et al.* [4] output.

Fig. 6: First row shows the output of our method and the second row is of Hassner *et al.* [4] for PIWFDS dataset.

its output is re-scaled to a 300×300 image for both the database images and our input. This is given to facial landmark detection code based on [7], which is publicly made available. This provides 68 landmarks on each face. Using the landmarks we divide the face surface into 110 planes (triangular in shape). Using the corresponding planes between input profile image and the exemplar frontal view image, we compute the homography transformations matrix using publicly available implementation of vgg_Haffine_from_x_MLE. Using this set of homographies, we synthesize the frontal view of the input image.

B. *Comparison with Hassner* et al. *[4]*

Figure 6 shows the comparative results between our method and that of Hassner *et al.* [4] for PIWFDS dataset. To show that our exemplar database is generic enough to extend to standard datasets, we provide qualitative results for LFPW [2] dataset in Figure 5. We use Hassner *et al.* publicly released code to obtain the result.

Observe ghost like appearances present in most of the cases from Hassner *et al.* [4] output. Also take into consideration, that the face structure of the actress in second row has been changed to a generic one. This is because they use a generic 3D model of the face to achieve the result.

C. *Quantitative Results*

For quantitative analysis, we used around 50 testing and 50 training samples of 6 classes for the face recognition task. Each sample is re-sized to a 300x300 image. After converting each sample from color to gray scale, we concatenated gray scale values to form a 90000 dimensional vector. We use Principal Component Analysis to reduce the dimensions to 40 using the training dataset. Each testing sample is also represented as 40 dimension vector as described above. We use publicly available implementation of Wright *et al.* [17] to recognize each input face.

Accuracy is calculated as fraction of testing samples classified correctly over the total number of samples. Our method achieved an accuracy of 31%, which is significantly better than 27% achieved by Hassner *et al.*.

IV. Conclusion

Pre-processing steps such as alignment, pose correction are vital in object recognition systems. We solve one such problem in face recognition domain, by neutralizing the pose of input image. Our method is novel, efficient and simple. These techniques also find applications in intelligent photo editing. We produce decisive qualitative and quantitative results.

References

[1] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *PAMI*, 1997.

[2] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. *CoRR*, 2014.

[5] J. Hu, J. Lu, and Y. Tan. Discriminative deep metric learning for face verification in the wild. In *CVPR*, 2014.

[6] G. B. Huang, V. Jain, and E. G. Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, 2007.

[7] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014.

[8] C. Lu and X. Tang. Surpassing human-level face verification performance on LFW with gaussianface. In *AAAI Conference on Artificial Intelligenc*, 2015.

[9] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[10] B. M. Oh, M. Chen, J. Dorsey, and F. Durand. Image-based modeling and photo editing. In *Conference on Computer Graphics and Interactive Techniques*, 2001.

[11] J. M. Saragih, S. Lucey, and J. F. Cohn. Real-time avatar animation from a single image. In *FG*, 2011.

[12] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *BMVC*, 2013.

[13] Y. Sun, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. *CoRR*, 2014.

[14] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. 2014.

[15] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *CVPR*, 2008.

[16] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *ACCV*, 2010.

[17] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 2009.

[18] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.