A Bayesian Mixture Model for Multi-view Face Alignment

Yi Zhou[1], Wei Zhang[2], Xiaoou Tang[1], and Harry Shum[1]
[1]Microsoft Research Asia
Beijing, P. R. China
{t-yizhou, xitang, hshum}@microsoft.com
[2]DCST, Tsinghua University
Beijing, P. R. China
w-z02@mails.tsinghua.edu.cn

Abstract

For multi-view face alignment, we have to deal with two major problems: 1. the problem of multi-modality caused by diverse shape variation when the view changes dramatically; 2. the varying number of feature points caused by self-occlusion. Previous works have used non-linear models or view based methods for multi-view face alignment. However, they either assume all feature points are visible or apply a set of discrete models separately without a uniform criterion. In this paper, we propose a unified framework to solve the problem of multi-view face alignment, in which both the multi-modality and variable feature points are modeled by a Bayesian mixture model. We first develop a mixture model to describe the shape distribution and the feature point visibility, and then use an efficient EM algorithm to estimate the model parameters and the regularized shape. We use a set of experiments on several datasets to demonstrate the improvement of our method over traditional methods.

1. Problem description

Parametric deformable models [1, 2, 3, 4, 5, 6] have been widely used in such vision tasks as image segmentation and object localization. Among different models, statistical shape models [3, 6] have shown state of the art performance for shape registration by using statistical techniques to describe shape distribution in order to regularize shapes in shape registration. However, the shape models used in these methods are generally Gaussian linear models and are only capable of describing faces with limited view changes. When view changes dramatically, the Gaussian linear models often fail to model shapes properly. To solve the multimodal shape registration problem, two main approaches have been proposed. One is the view based method which uses a set of different models to represent shapes from different views [4], and the other is the non-linear model method which uses nonlinear models to represent shape variations [5]. The view-based methods treat a set of discrete models separately without a uniform criterion to regularize shape in a multimodal framework. It is very difficult to cover unlimited view change possibilities with a small set of discrete models. Therefore, the method has high requirements on the views of the training data, which has to be close to the testing data. Since every view has to be computed, the computation cost significantly increases. On the other hand, the non-linear model methods generally assume all the feature points are visible. But this assumption might not hold for multi-view faces. In fact, an important issue for large view variation is that some feature points might become invisible at some views because of self-occlusion. So this requires the model to be able to handle varying feature point sets. However, the varying dimensional model optimization itself is an ill posed problem of exponential complexity. Instead of solving such a varying dimension optimization problem, we will treat the point visibility as a random variable and then infer the probability of the visibility of each point.

In this paper, we propose a multimodal Bayesian framework for multi-view face alignment. First, the problems of multi-modality and variable feature points are formulated in a unified Bayesian framework. Specially, we use a mixture model to describe the shape distribution and point visibility, and then derive the posterior of the model parameters given an observation of unknown valid feature points. Second, an EM algorithm is given to estimate the model parameters, the regularized shape, and the visibility of its points. Extensive experiments on several datasets clearly show the efficacy of the new algorithm.

2. Problem formulation

Our probabilistic formulation includes a prior mixture model of the shape and point visibility and a likelihood model about the observation of variable feature points in the image space. At the end of this section, we derive the posterior about the model parameters and formulate a hierarchical hidden variable model for the problem.

2.1. Prior model

A shape with N landmark points is labeled by a $3N$-dimensional vector $(x_{11},x_{12},v_1,\ldots,x_{N1},x_{N2},v_N)$, in which the pair (x_{i1}, x_{i2}) is the coordinates of the *ith* point and v_i is a 0-1 variable indicating the visibility status of the *ith* point. Using $\boldsymbol{x}=(x_{11},x_{12},\ldots,x_{N1},x_{N2})^T$ to denote all point coordinates and using $\boldsymbol{v}=(v_1,\ldots,v_N)^T$ as the visibility indicator for all points, we represent a shape as $(\boldsymbol{x}, \boldsymbol{v})$. The prior model includes a mixture shape model of $\boldsymbol{x}$ and a mixture visibility model of $\boldsymbol{v}$. A mixture shape model is learned after

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

aligning all the shapes to a common coordinate frame. Computing the common coordinate frame and aligning all the shapes is a typical Generalized Procrustes Analysis (GPA) [3, 5, 6]. In this paper, we learn a 2-cluster mixture PPCA model [9],

$$p(\mathbf{x}\,|\,\mathbf{b}) = \sum_{i=1}^{m} \pi_i f_i\left(\mathbf{x}\,|\,\mathbf{b}^{(i)}\right) = \sum_{i=1}^{m} \pi_i \left(2\pi\sigma_i^2\right)^{-N} \exp\left\{-\frac{1}{2\sigma_i^2}\left\|\mathbf{x}-\boldsymbol{\mu}^{(i)}-\boldsymbol{\Phi}^{(i)}\mathbf{b}^{(i)}\right\|^2\right\}, \quad (1)$$

where m is the cluster number, π_i is the cluster weight and $f_i(\cdot|\boldsymbol{b}^{(i)})$ is a normal density function; $\boldsymbol{b}=(\boldsymbol{b}^{(1)},\ldots,\boldsymbol{b}^{(m)})$ is called the shape parameter and each $\boldsymbol{b}^{(i)}$ follows a Gaussian distribution $p(\boldsymbol{b}^{(i)})=N(\boldsymbol{0}, \boldsymbol{\Lambda}^{(i)})$ in which $\boldsymbol{\Lambda}^{(i)}$ is a diagonal matrix; $\boldsymbol{\mu}^{(i)}$ is the center of the *ith* cluster and $\boldsymbol{\Phi}^{(i)}$ is the principal matrix whose columns are the eigenvectors.

The prior landmark point visibility model is defined as a mixture Bernoulli model,

$$p(\mathbf{v}) = \sum_{i=1}^{m} \pi_i \prod_{k=1}^{N}\left[\left(q_k^{(i)}\right)^{v_k}\left(1-q_k^{(i)}\right)^{1-v_k}\right], \quad (2)$$

where the cluster weight π_i is the same as Eq. (1) and $q_k^{(i)}$ is a value between 0 and 1 which defines the probability of the *kth* point to be visible for the shapes in the *ith* cluster. Each $q_k^{(i)}$ is learned after the mixture shape model is obtained. For a training shape $\boldsymbol{x}$, its probability for belonging to the *ith* cluster is denoted as $f_i(\boldsymbol{x})$. We compute $f_i(\boldsymbol{x})$ by first projecting $\boldsymbol{x}$ into the principle subspace $\boldsymbol{\Phi}^{(i)}$ to get its shape parameter $\boldsymbol{b}^{(i)}$ and then multiplying the probability $p(\boldsymbol{b}^{(i)})$ with $f_i(\boldsymbol{x}|\boldsymbol{b}^{(i)})$. In other words, we compute the Mahanalobis distance of $\boldsymbol{x}$ to the cluster center $\boldsymbol{\mu}^{(i)}$ as the summation of the M-distance in the principal subspace and the M-distance from the principal subspace [9]. So far, with the training set $\{(\boldsymbol{x}^{(1)},\boldsymbol{v}^{(1)}),\ldots,(\boldsymbol{x}^{(L)},\boldsymbol{v}^{(L)})\}$, the prior visible probability $q_k^{(i)}$ of point k in cluster i is

$$q_k^{(i)} = \sum_{l=1}^{L} v_k^{(l)} f_i\left(\mathbf{x}^{(l)}\right) \Big/ \sum_{l=1}^{L} f_i\left(\mathbf{x}^{(l)}\right). \quad (3)$$

Figure 1 shows the two cluster centers of the mixture shape model with the visible probability of each landmark point illustrated as the degree of the darkness. The darker the point is, the more likely it might be occluded.

2.2. Likelihood model

For the current searched image, the observation for the shape regularization is the updated shape after local texture matching [3, 6]. We call this updated shape as the observed shape [6] and denote it as $\boldsymbol{y}$. The observed shape $\boldsymbol{y}$ is in the image space and there is a similarity transformation difference between $\boldsymbol{y}$ and the shapes in the underlying shape space. We use a vector $\boldsymbol{\theta}=(c_1,c_2,s,\theta)^T$ to represent this set of transformation parameters, in which $\boldsymbol{c}=(c_1,c_2)^T$ is the translation, s is the scale and θ is the rotation. We use $T_\theta(\cdot)$ to denote the similarity transformation incurred by this set of pose parameter, i.e. for a shape vector $\boldsymbol{x}$, $T_\theta(\boldsymbol{x})=s(\boldsymbol{I}_N\otimes\boldsymbol{U}_\theta)\boldsymbol{x}+\boldsymbol{I}_N\otimes\boldsymbol{c}$. Here $\otimes$ is the Kronecker product, $\boldsymbol{I}_N$ is a N-dimensional column vector whose elements are all one, $\boldsymbol{I}_N$ is the N-dimensional identity matrix and $\boldsymbol{U}_\theta$ is the rotation matrix with the angle θ. We denote $T_\theta^{-1}(\cdot)$ as the inverse transformation of $T_\theta(\cdot)$.

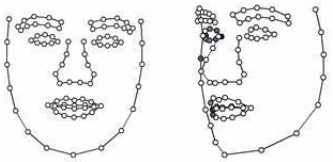


Figure 1: Centers of the two clusters in the mixture shape model.

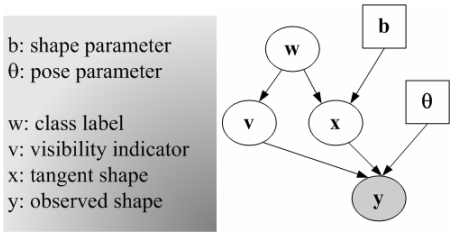


Figure 2: The hierarchical hidden variable model.

The observed shape $\boldsymbol{y}$ is known to be noisy and some of its feature points might be occluded. So given the underlying shape $\boldsymbol{x}$ and its point visibility $\boldsymbol{v}$, $\boldsymbol{y}$ is assumed to follow a distribution in Eq. (4),

$$p(\mathbf{y}\,|\,\mathbf{x},\mathbf{v},\boldsymbol{\theta}) = \left(2\pi\rho^2\right)^{-|\mathbf{v}|}\exp\left\{-\frac{1}{2\rho^2}\left\|(\mathbf{v}\otimes\mathbf{1}_2)\circ\left[\mathbf{y}-T_\theta(\mathbf{x})\right]\right\|^2\right\}, \quad (4)$$

where the notation $\circ$ is array multiplication which is the entry-by-entry product of two vectors and the notation $|\cdot|$ is the L_1 norm. The variance ρ^2 is estimated from the update of the shape in the local texture matching step [6]. The visibility variable $\boldsymbol{v}$ indicates the valid feature points of the observed shape $\boldsymbol{y}$.

2.3. Hierarchical hidden variable model

The parameters to be estimated here are the shape parameter $\boldsymbol{b}$ and the pose $\boldsymbol{\theta}$. Based on the prior models (1), (2), and the likelihood model (4), the posterior of the parameters is obtained by integrating the hidden variables,

$$p(\mathbf{b},\boldsymbol{\theta}\,|\,\mathbf{y}) \propto \int p(\mathbf{y}\,|\,\mathbf{x},\mathbf{v},\boldsymbol{\theta})\,p(\mathbf{x}\,|\,\mathbf{b})\,p(\mathbf{v})\,p(\mathbf{b})\,\mathbf{dxdv}. \quad (5)$$

The variables $\boldsymbol{x}$ and $\boldsymbol{v}$ are the hidden variables in our formulation. In addition, the cluster indicator variable is denoted as $\boldsymbol{w}=(w_1,\ldots,w_m)^T$, which only takes value like $(0,\ldots,0,1,0,\ldots0)^T$ and its distribution function is,

$$p(w_i=1)=\pi_i \text{ or } p(\mathbf{w}) = \prod_{i=1}^{m}\pi_i^{w_i}. \quad (6)$$

And then given the cluster variable $\boldsymbol{w}$, the distribution of $\boldsymbol{x}$ and $\boldsymbol{v}$ can be rewritten as

$$p(\mathbf{x}\,|\,\mathbf{w},\mathbf{b}) = \prod_{i=1}^{m}\left[f_i\left(\mathbf{x}\,|\,\mathbf{b}^{(i)}\right)\right]^{w_i}, \quad (1)'$$

$$p(\mathbf{v}\,|\,\mathbf{w}) = \prod_{i=1}^{m}\prod_{k=1}^{N}\left[\left(q_k^{(i)}\right)^{v_k}\left(1-q_k^{(i)}\right)^{1-v_k}\right]^{w_i}. \quad (2)'$$

With $\boldsymbol{x}$, $\boldsymbol{v}$, and $\boldsymbol{w}$, the whole framework can be described by a hierarchical hidden variable model in Figure 2.

3. Shape regularization by EM algorithm

In this section, we describe an EM algorithm to compute the parameter MAP estimation in the hierarchical hidden variable model in Figure 2.

3.1. EM algorithm for MAP of parameters

Instead of directly optimizing the posterior function in Eq. (5), we apply the EM algorithm to find the MAP estimation of parameters. The E step computes the expectation of the log-posterior. It is done through the computation of the expectation of several sufficient variables. The derivation of these sufficient statistics is presented in Appendix A and B. And we summarize the E step as follows.

E1) Estimation of the cluster weight: $\hat{w}_i = E[w_i \mid \mathbf{y}]$.

E2) Estimation of the point visibility: $\hat{\mathbf{v}}_i = E[\mathbf{v} \mid \mathbf{y}, w_i = 1]$.

E3) Estimation of the shape: $\hat{\mathbf{x}}_i = E[\mathbf{x} \mid \mathbf{y}, \hat{\mathbf{v}}_i, w_i = 1]$

$= \left((1-\hat{\mathbf{v}}_i)\otimes \mathbf{1}_2\right)\circ \boldsymbol{\Phi}^{(i)}\mathbf{b}^{(i)} + (\hat{\mathbf{v}}_i \otimes \mathbf{1}_2)\circ\left(p_i \boldsymbol{\Phi}^{(i)}\mathbf{b}^{(i)} + (1-p_i)T_\theta^{-1}(\mathbf{y})\right)$.

In the above steps, refer to Eq. (12) in Appendix B for the computation of w_i and Eq. (11) for the computation of $\boldsymbol{v}_i$. Each element of $\boldsymbol{v}_i$ is a value between 0 and 1, and it determines how much we will use the information of each point from the observation when we estimate the underlying shape $\boldsymbol{x}_i$. The form of the estimation of the shape $\boldsymbol{x}_i$ tells us: 1) if some element of $\boldsymbol{v}_i$ is small, that is, the corresponding point is supposed to be very likely to be occluded, we will put less confidence on that point; 2) when the observed shape is used for the estimation of $\boldsymbol{x}_i$, its weight depends on the ratio between its noise and the noise in the shape space. Please refer to Eq. (10) in Appendix B for the derivation of $\boldsymbol{x}_i$. With the expectation of the sufficient statistics computed in the E step, the M step tries to maximize the expected log-posterior. It is done separately for the shape and pose parameters,

M1) update of shape: $\tilde{\mathbf{b}}^{(i)} = \dfrac{\hat{w}_i \boldsymbol{\Lambda}^{(i)}}{\hat{w}_i \boldsymbol{\Lambda}^{(i)} + \sigma_i^2}\left(\boldsymbol{\Phi}^{(i)T}\hat{\mathbf{x}}_i\right)$,

M2) update of pose: $\tilde{\boldsymbol{\theta}} = \min\left\{\sum_{i=1}^{m} \hat{w}_i \left\|(\hat{\mathbf{v}}_i \otimes \mathbf{1}_2)\circ\left(\mathbf{y} - T_\theta(\hat{\mathbf{x}}_i)\right)\right\|^2\right\}$.

From Eq. M1), we can see when projecting the shape $\boldsymbol{x}_i$ to the PCA subspace $\boldsymbol{\Phi}^{(i)}$, it is regularized by shrinking its weight along each principal direction. And the intensity of shrinkage depends on the energy $\boldsymbol{\Lambda}^{(i)}$ at the principal direction. If the energy $\boldsymbol{\Lambda}^{(i)}$ is small, that means the distribution along the principal directions is compact, then the shrinkage will be significant; otherwise, the shrinkage will be minor. In addition, the cluster weight w_i is also used to compute the shrinking factor. If the cluster weight is small, i.e. it is unlikely that the shape comes from this cluster, the shrinkage will be much significant for the shape parameter of this cluster. Eq. M2) shows that the cluster weight also contributes to the weighted least square process to update the pose parameter $\boldsymbol{\theta}$. And the point visibility $\boldsymbol{v}_i$ determines the contribution of each feature point in the update of pose.

3.2. The regularized shape and its reliability

Once we get the MAP parameter ($\boldsymbol{b}$, $\boldsymbol{\theta}$), we can regularize the observed shape $\boldsymbol{y}$ as the transformed expected underlying shape $\boldsymbol{x}$ with the pose parameter $\boldsymbol{\theta}$. And the expected tangent shape $\boldsymbol{x}$ is

$$E\left[\mathbf{x} \mid \mathbf{y}, \tilde{\mathbf{b}}, \tilde{\boldsymbol{\theta}}\right] = \sum_{i=1}^{m} \hat{w}_i \hat{\mathbf{x}}_i \ . \tag{7}$$

So the regularized shape is a weighted average of shapes in different clusters which are the best matched to the observed shape. It is weighed by the weight of the cluster, which tells us how likely the current shape belongs to this cluster. And we can also get the expected point visibility $\boldsymbol{v}$ of the regularized shape $\boldsymbol{x}$ as,

$$E\left(\mathbf{v} \mid \mathbf{y}, \tilde{\mathbf{b}}, \tilde{\boldsymbol{\theta}}\right) = \sum_{i=1}^{m} \hat{w}_i \hat{\mathbf{v}}_i \ . \tag{8}$$

The point visibility estimation is a value between 0 and 1, and it actually gives us a measurement of the reliability of each feature point of the regularized shape. In fact, although the shape $\boldsymbol{x}$ and the point visibility $\boldsymbol{v}$ are assumed independent in the prior mixture model, this independence is not tenable any more after the observation $\boldsymbol{y}$ is incorporated. This is illustrated by the formulation in Figure 2. The estimation of shape $\boldsymbol{x}$ is based on its visibility probability $\boldsymbol{v}$. And the point visibility probability measures how consistent one point is in the shape model, or in other words, to all other points.

4. Experimental analysis

We apply the hierarchical mixture model to multi-view face alignment. In this section, we describe the training procedure in details and evaluate the performance of the algorithm in terms of accuracy and stability. We demonstrate that, with a set of discrete view models, we can handle face images within a range of views. The mixture model can not only compute positions of the feature points, but also estimate the reliability of each point. An interesting byproduct of the new algorithm is that we can also use the model to estimate the view direction of a face.

4.1. Training the mixture model

The database we use for training and testing includes both the real and synthetic images. In order to better evaluate the algorithm, we divide our database into three parts: Dataset I, Dataset II, and Dataset III. Dataset I is composed of the frontal, 40 and 50 degree viewed face images, which will be used to train the prior mixture model. There are 230 frontal face images in Dataset I which come from the FERET database [7] and the AR database [8]. And there are 100 face images for the 40 and 50 degree views respectively, which are generated from the USF Human ID 3D database [10]. Dataset II also contains viewed face images generated from these 3D data. The faces in Dataset II are of 10, 20, and 30 degree

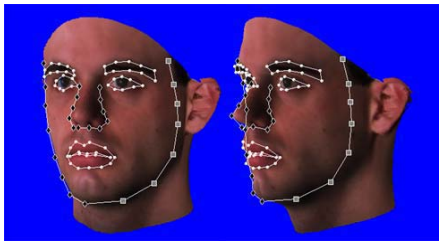


Figure 3: Faces with labeled landmarks.

views and there are 100 images for each view. The images in Dataset II are well labeled, and since they are different from those in Dataset I, they will be used as the ground truth for testing. The images in Dataset III are all synthetic images. We use 490 images in the FERET and AR databases and apply the technique in [11] to generate the 3D model of the face in each image. Then, using the 3D model, we can synthesize viewed face images at arbitrary views. In Dataset III, we have totally 490x51 synthetic face images, so there are 490 images at each view direction from 0 to 50 degrees.

The faces in Dataset I and II are labeled with 83 feature points. Figure 3 shows labeled faces at 10 and 50 degree views. We draw the points with three different shapes to represent different meanings. The circle points (white), i.e. the points on eyes, eyebrows, and mouth, are those of texture significance in the image and have the same geometrical meanings at each view. These points are chosen to correspond to the same points on the 3D model. More precisely, the locations of these points at different views all have corresponding points on the frontal view face at which all the points are visible. Unlike the circle points, the diamond ones (black) do not have the same geometrical meaning for all the views. In fact, they are selected because they have strong texture significance on the image, for example the edges, and have important functions in image searching. This kind of points includes those on the nose and the left part of silhouette (boundary with the background). All the other points (gray) are drawn with squares for those without texture meanings on the images. They are chosen to correspond to the same points on the 3D face model. These points are on the right part of the silhouette. After we finish labeling all the locations of these feature points, their visibilities are automatically computed with the 3D model of each face.

Training the hierarchical model includes learning a mixture shape model for all the 83 feature points and a mixture visibility model for each point. In the prior model, we learn a mixture PPCA model for the shape and a mixture Bernoulli model about the visibility of each point as described in Section 2.1. Training is done on the images in Dataset I and two clusters are learned for the mixture model. Figure 1 shows the two cluster centers. One cluster is the faces of nearly the frontal views and the other is the faces of 40 and 50 degree views. Given a shape $\boldsymbol{x}$, the probability $p(w_i=1|\boldsymbol{x})$ describes the likelihood that $\boldsymbol{x}$

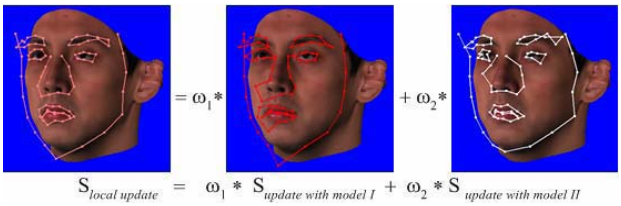


Figure 4: Local updating step.

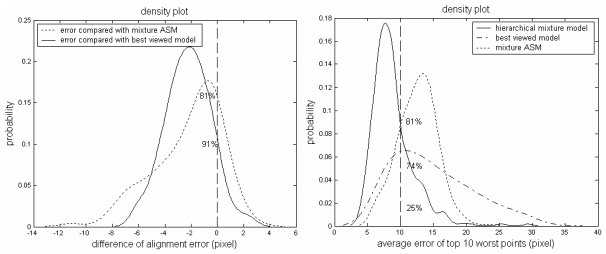
comes from the cluster i and how close it is to this cluster center, so this probability may give us some information on the view of $\boldsymbol{x}$. Based on this consideration, we learn a regression function of the view over the cluster weight estimation from Dataset I and II. The probability $p(w_i=1|\boldsymbol{x})$ is proportional to $p(w_i=1)f_i(\boldsymbol{x})$. Since the view distribution in Dataset I and II is very sparse, which only takes values as 0, 10, 20, 30, 40 and 50, the regression function we learned is a very simple 2nd-order polynomial. Thus, with this regression function and the estimated weight we can roughly estimate the view direction of an input face.

4.2. Local texture update

Similar to ASM and BTSM, the process of our method is composed of two steps: local texture matching and global shape regularization. However, because of the multimodality of the images we model, we use a slightly different local texture matching strategy. The local texture model is learned for each cluster separately. Specifically, for each cluster, the local texture model is trained on the faces whose shapes are within the three standard deviations of the PPCA model of each cluster. Our local texture model is the same as ASM and BTSM, which is based on gray gradients at the feature point. Also, our method is implemented in a multi-resolution framework and a three-layer Gaussian image pyramid is formed on each image by down-sampling its pixels to obtain images of coarser resolutions. And the local models are trained for each layer. Therefore, for each point, there are two local models at each layer. When we use the two models to do local updating, we first calculate the update of the point with each local model and then weigh the results to get a final one. And the weight is just the weight estimation of the cluster. Figure 4 shows this local updating process. After the local updating, in the shape regularization step, we use the EM algorithm in Section 3 to estimate the shape and pose parameters, and finally get the estimation of the regularized shape and the reliability of each point.

4.3. Alignment accuracy

The accuracy of our algorithm is tested on Dataset II which has labeled ground truth. It should be noted that the faces in Dataset II are of 10~30 degree views, which do not appear in our training data in Dataset I. We intend to show that using only a set of discrete view models, we



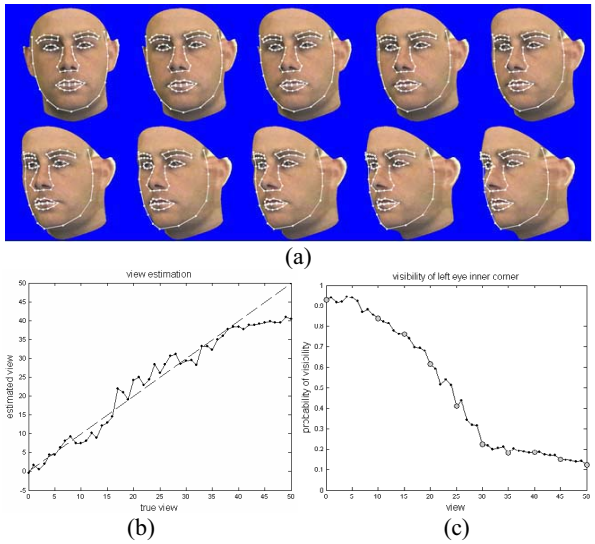
(a) (b)

Figure 5: Statistics of alignment errors. (a) Densities of error differences. (b) Densities of errors of top 10 worst points.

Figure 6: Results on testing images.

can handle face images in a range of views. To demonstrate the accuracy, we compare our method with another two existing methods: the view based model and the nonlinear model. The alignment error is calculated as the average point-to-point distance between the aligned points and the labeled ground truth. For the view based model, we train two linear BTSM models [6] for frontal view and 40-50 view respectively with the face images in Dataset I and present the best result of the individual viewed BTSM models. For other nonlinear models, we compare ours with that using a mixture Gaussian model for representing shape variations in ASM [5]. The three algorithms use the same local texture description and start with the same initialization. The alignment accuracy of the three methods is compared in Figure 5. Figure 5 (a) shows the overall improvement of our model over the other two. It shows that our method is better than the mixture ASM on 81% images and outperforms the best viewed BTSM on 91% images. The improvement can be seen more clearly if we only concentrate on the worst aligned points. Figure 5 (b) shows the average errors of top 10 worst aligned points: while 81% images aligned by the mixture ASM and 74 % images aligned by the viewed BTSM have this error above 10 pixels, our model has an error below 10 pixels on 75% images. Some results of the three methods are also shown in Figure 6.

These results show two aspects of our algorithm. First, compared with view based methods, our method can reasonably weigh each mixture component model. Since the



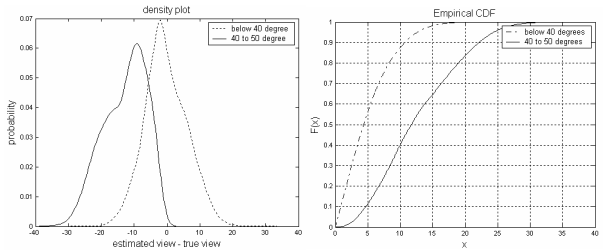
(a) (b) (c)

Figure 7: (a) Aligned results. (b) View estimation. (c) Estimation of the visibility probability of the left eye inner corner point.

weighted combination of mixture components implies a much bigger sample space, our method can also work on the cases even if similar viewed samples are not in the training data. Second, compared with other nonlinear models, our model considers the visibility of the feature points and uses this visibility measure to prevent the missing or misleading features to ruin the overall result.

4.4. View and point visibility estimation

One interesting result of our method is that we can estimate the face view direction using only the 2D model. We do this through the estimation of the cluster weights. In our model, this weight estimation represents how far the face is from the cluster center in terms of the Mahalanobis distance. We have one cluster of nearly frontal faces and the other of about 40~50 degree viewed faces. It is reasonable to assume that the faces between 0 and 50 degrees have different distances to the two cluster centers. And this conjecture is confirmed by our experimental results. The dotted line in Figure 7 (b) shows view estimation results when the view of a face changes from 0 to 50 degrees and Figure 7 (a) shows the alignment results. We can see the estimation is consistent for the continuous change of views. It shows the weight estimation is rather stable. Although the results seem not fitted well between 40 to 50 degrees, this is reasonable since 40 and 50 degree viewed faces are in the same cluster in our model so that the views between them cannot be classified well. Another interesting outcome is the estimation of point visibility and this quantity is a probability between 0 and 1 which can give us a reliability measure for each point. Figure 7 (c) shows the curve of the estimated visible probability of the left-eye inner corner point with the change of views. The gray dot points exactly correspond to the results visualized in Figure 7 (a).



(a) (b)

Figure 8: Statistics of view estimation errors. (a) Densities. (b) Empirical CDFs.

Finally, we test the behavior of our view estimation method on Dataset III. Dataset III is composed of synthetic viewed face images from 0 to 50 degree views and there are 490 images at each view. Since the weight will become undiscriminating between 40 and 50 degree views, we present the results in these views separately. Figure 8 (a) shows the density of the view estimation errors. The x-axis is the difference of the estimated view and the true view. We can see that for the views under 40 degrees, the view estimation is roughly a Gaussian distribution around the true view with a standard deviation of about 10 degrees. Figure 8 (b) further plots the empirical c.d.f. of the absolute errors. It shows that for the face views bellow 40 degrees, 80% of them have the view estimation errors under 8 degrees and 90% under 10 degrees.

5. Conclusion and discussion

This paper has presented a new nonlinear model for multimodal shape registration problem. By modeling shape and point visibility in a mixture probabilistic model, we have handled the large shape variation and variable feature points in a unified framework. Within this framework, an efficient EM algorithm for shape regularization is developed. When leveraging all the modalities in our model, we reasonably weigh their contributions in a sense of goodness-of-fit. Since this kind of continuous weight values are more accurate than the discrete values used by traditional view based methods, our method is more stable when the view changes. That partially explains why our model can work on a range of views with the prior model learned only on several views. On the other hand, by introducing a hidden variable to indicate the visibility status of each point, we handle the self-occlusion stochastically.

6. References

[1] A. Blake and M. Isard. Active Contours. *Springer,* 1998.

[2] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1:321--331, 1987.

[3] T. F. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models – their training and their applications. *Computer Vision and Image Understanding*, 61(1):38-59, January 1995.

[4] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor. View-based Active Appearance Models. *Image and Vision Computing*, 2002.

[5] T. F. Cootes and C. Taylor. A mixture model for representing shape variation. *BMVC*, 1997.

[6] Y. Zhou, L. Gu and H. J. Zhang. Bayesian Tangent Shape Model: Estimating Shape and Pose via Bayesian Inference. *IEEE Conf. on CVPR*, 2003.

[7] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. on PAMI*, 22(10):1090--1104, 2000.

[8] A.M. Martinez and R. Benavente. The AR Face Database. *CVC Technical Report* #24, June 1998.

[9] M. E. Tipping and C. M. Bishop. Mixtures of Probabilistic Principal Component Analysers. *Neural Computation*, July 1998.

[10] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D-faces. *ACM SIGGRAPH*, 1999.

[11] Y.X. Hu, D. L. Jiang, S. C. Yan, and H. J. Zhang. Automatic 3D Reconstruction for Face Recognition. *Intl. Conf, on AFGR*, 2004.

Appendix

A. Expectation of log-posterior

The expectation of the log-posterior of the complete variable (***y***,***x***,***v***,***w***) is computed by the following function,

$$Q(\mathbf{b},\boldsymbol{\theta}\,|\,\tilde{\mathbf{b}},\tilde{\boldsymbol{\theta}}) = -\frac{1}{2}\sum_{i=1}^{m}\left\{\mathrm{E}_{w|y}\left[w_i \mathrm{E}_{v|w_i,y}\mathrm{E}_{x|v,w_i,y}\left[\mathbf{b}^{(i)T}\Lambda^{(i)-1}\mathbf{b}^{(i)} + \rho^{-2}\left\|(\mathbf{v}\otimes\mathbf{1}_2)\circ(\mathbf{y}-T_\theta(\mathbf{x}))\right\|^2 + \sigma_i^{-2}\left\|\mathbf{x}-\boldsymbol{\Phi}^{(i)}\mathbf{b}^{(i)}\right\|^2\right]\right]\right\} + const \quad (9)$$

So the computation of Q-function is to compute the sufficient statistics $\hat{w}_i = E[\mathbf{w}\,|\,\mathbf{y}]$, $\hat{\mathbf{v}}_i = E[\mathbf{v}\,|\,\mathbf{y}, w_i = 1]$, and $\hat{\mathbf{x}}_i = E[\mathbf{x}\,|\,\mathbf{y}, \hat{\mathbf{v}}_i, w_i = 1]$.

B. Some conditional probabilities

1) Posterior of the shape ***x*** in cluster *i* with visibility ***v***:

$$p(\mathbf{x}\,|\,\mathbf{y},\mathbf{v},w_i=1) \propto f_i\left(\mathbf{x}\,|\,\mathbf{b}^{(i)}\right)\cdot p(\mathbf{y}\,|\,\mathbf{x},\mathbf{v},\boldsymbol{\theta}) \propto \exp\left\{-\frac{1}{2\sigma_i^2}\left\|\mathbf{x}-\boldsymbol{\Phi}^{(i)}\mathbf{b}^{(i)}\right\|^2 - \frac{1}{2s^{-2}\rho^2}\left\|(\mathbf{v}\otimes\mathbf{1}_2)\circ\left[\mathbf{x}-T_\theta^{-1}(\mathbf{y})\right]\right\|^2\right\} \quad (10)$$

So the expectation of ***x*** is that of Eq. E3) in Section 3.1.

2) Posterior of point visibility ***v*** for each cluster:

$$p(\mathbf{v}\,|\,\mathbf{y},w_i=1) \propto p(\mathbf{v}\,|\,w_i=1)\cdot\int p(\mathbf{y}\,|\,\mathbf{x},\mathbf{v},\boldsymbol{\theta}) f_i\left(\mathbf{x}\,|\,\mathbf{b}^{(i)}\right)\mathbf{dx} \propto \prod_{k=1}^{N}\left[\left(q_k^{(i)}\right)^{v_i}\left(1-q_k^{(i)}\right)^{1-v_i}\right]\cdot\left(\frac{\left(2\pi\delta_i^2\right)}{\left(2\pi\rho^2\right)\left(2\pi\sigma_i^2\right)}\right)^{|\mathbf{v}|} \cdot\exp\left\{-\frac{\left\|(\mathbf{v}\otimes\mathbf{1}_2)\circ\mathbf{z}'\right\|^2}{2\delta_i^2} + \frac{\left\|(\mathbf{v}\otimes\mathbf{1}_2)\circ\mathbf{z}_1\right\|^2}{2\sigma_i^2} + \frac{\left\|(\mathbf{v}\otimes\mathbf{1}_2)\circ\mathbf{z}_2\right\|^2}{2\rho^2 s^{-2}}\right\} \quad (11)$$

3) Posterior of view class:

$$p(w_i=1\,|\,\mathbf{y}) \propto p(w_i=1)\,p(\mathbf{y}\,|\,w_i=1) \propto \pi_i\prod_{k=1}^{N}\left[\left(1-q_k^{(i)}\right) + \frac{q_k^{(i)}\delta_i^2}{2\pi\sigma_i^2\rho^2}\cdot\exp\left\{-\frac{\left\|z'^k\right\|^2}{2\delta_i^2} + \frac{\left\|z_1^k\right\|^2}{2\sigma_i^2} + \frac{\left\|z_2^k\right\|^2}{2\rho^2 s^{-2}}\right\}\right] \quad (12)$$

where $\delta_i^2 = \left(\sigma_i^{-2} + s^2\rho^{-2}\right)^{-1}$, $p_1 = \left(s^{-2}\rho^2\right)/\left(\sigma_i^2 + s^{-2}\rho^2\right)$, $p_2 = 1-p_1$, $\mathbf{z}_1 = \boldsymbol{\Phi}^{(i)}\mathbf{b}^{(i)}$, $\mathbf{z}_2 = T_\theta^{-1}(\mathbf{y})$, and $\mathbf{z}' = p_1\mathbf{z}_1 + p_2\mathbf{z}_2$, z^k is the *kth* point of ***z***.