3D Alignment of Face in a Single Image

Lie Gu and Takeo Kanade
Computer Science Department
Carnegie Mellon University
{gu+, tk}@cs.cmu.edu

Abstract

We present an approach for aligning a 3D deformable model to a single face image. The model consists of a set of sparse 3D points and the view-based patches associated with every point. Assuming a weak perspective projection model, our algorithm iteratively deforms the model and adjusts the 3D pose to fit the image. As opposed to previous approaches, our algorithm starts the fitting without resorting to manual labeling of key facial points. And it makes no assumptions about global illumination or surface properties, so it can be applied to a wide range of imaging conditions. Experiments demonstrate that our approach can effectively handle unseen faces with a variety of pose and illumination variations.

1. Introduction

Automatically locating detailed facial landmarks across different subjects and viewpoints, i.e. 3D alignment of face, is a challenging problem. Previous approaches can be divided into two categories: view based and 3D based. View-based methods [1, 2, 3] train a set of 2D models, each of which is designed to cope with shape or texture variation within a small range of viewpoints. 3D-based methods [4, 5, 6, 7], in contrast, deal with all views by a single 3D model. The early work of Blanz et.al. [4] on 3D morphable model interprets a face by minimizing intensity difference between the synthesized image and the given image. Zhang et.al [6] proposed an approach that deforms a 3D mesh model so that the 3D corner points reconstructed from a stereo pair lie on the surface of the model. Dimitrijevic et. al. [7] proposed the use of a 3D morphable model similar to that of Blanz's, but discarded the texture component from the model in order to reduce the sensitivity to illumination. Both [6] and [7] minimize shape difference instead of intensity difference, but rely on stereo correspondence. These 3D-based methods are reported to require proper initialization, which typically involves manual labeling of certain key points.

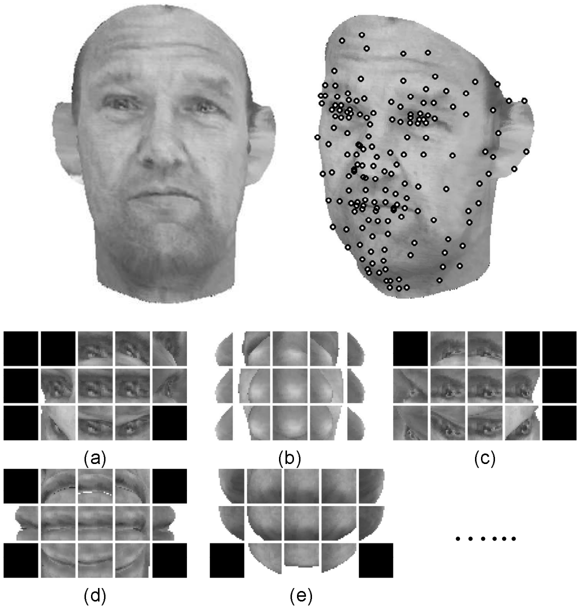

Figure 1. *A 3D face (upper left) is represented by a sparse set of 3D points (upper right) and the view-based 2D patches associated with every point (bottom). In this example, we choose* 190 *points from the face surface, in which* 89 *points are located on the contours of facial components. The other points are approximately uniformly distributed. Patches are sampled from images rendered under 15 different views:* $\{-90^o, -45^o, 0^o, 45^o, 90^o\}$ *for roll and* $\{-35^o, 0^o, 35^o\}$ *for pitch.* $(a \sim e)$ *shows patches that correspond to left eye center, nose tip, right eye center, mouse center and chin tip, respectively. Black patches indicate self-occlusion under the corresponding viewpoint.*

Our approach is a 3D patch-based approach. A face, as shown in Figure 1, is modeled by a set of sparse 3D points (shape) and the view-based patches (appearance) associated with every point. Working on the patch level, as opposed to the holistic face region, offers us two advantages: it is easier to compensate illumination locally; and the variance of texture within a patch is considerably smaller than that of the whole face. However, patch information alone is not enough to localize a facial point. The point could be occluded, or a similar patch pattern could present in a neighboring region. It is essential to constrain the spatial arrangement of the facial points by shape priors. We construct a compact 3D shape prior on the sparse 3D point set, and apply it to constrain the 2D facial points in different views.

The initial positions of the shape points in a given image are located using a simple gradient feature detector designed for each point. Those independently found positions are rather noisy and contains localization error. They also provide only an incomplete observation of the underlying 3D shape because the depth is missing and some points may be invisible. We formulate the alignment process of the 3D model as a Bayesian inference problem with missing data, whose task is to solve 3D shape and 3D pose from the noisy and incomplete 2D shape observation. To resolve the uncertainties we develop an EM-based algorithm, which decouples the model fitting into three separate steps: shape augmentation, shrinkage regularization, and pose estimation. The algorithm first produces an "augmented" 3D shape by a weighted combination of the shape observation and the current shape estimate; then regularizes the 3D shape by shrinking it in principal subspace toward the mean shape; and estimates its pose by solving a constrained optimization problem. The algorithm iteratively refines the 3D shape and the 3D pose until convergence.

In training we utilize a gigantic database of over $50,000$ synthetic 3D faces. We have constructed the faces from labeled 2D images, so that the correspondences among 2D faces are automatically conveyed to 3D. Shape priors are learned from the database, and view-based patch statistics are collected from the synthesized images. The algorithm is tested on CMU PIE database [10]. We demonstrate it can handle unseen faces with a variety of pose and illumination variations.

2. Problem Formulation

The 3D geometrical structure of a face is described by a set of 3D points concatenated into a vector $S_{3n\times 1} = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n)^t$, aligned with a normalized reference frame. We parameterize S as a linear deformable model,

$$S = \mu + \Phi b + \varepsilon \tag{1}$$

where the mean shape μ and the principal subspace matrix Φ are computed from training shape samples using PCA. The vector b represents the *deformation parameters* that describe the deformation of S along each principal direction. Assume that b is distributed as a diagonal gaussian

$$b \sim \mathcal{N}(0, \Lambda), \ \Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_r\} \tag{2}$$

with zero mean and variance λ_i's, which are the eigenvalues of PCA. The *shape noise* term ε measures the deviation of S from the principal subspace. We model ε as an isotropic gaussian with its variance set to be the average residual energy [12] that is not captured in the principal subspace,

$$\varepsilon \sim \mathcal{N}(0, \sigma^2 I), \ \sigma^2 = \frac{1}{3n}\sum\nolimits_{i=r+1}^{3n} \lambda_i \tag{3}$$

Given an input image, let $q_{2m\times 1} = (u_1, v_1, \ldots, u_m, v_m)^t$ denote the vector of m visible 2D points located in it. In general, q represents an incomplete and noisy observation of S. We relate q with S by noised weak perspective projection,

$$q = s\mathrm{M}PRS + t + \zeta \tag{4}$$

The 3D shape $S_{3n\times 1}$ is first rotated by $R_{3n\times 3n} = I_n \otimes R_3$, then projected onto the image plane with $P_{2n\times 3n} = I_n \otimes [(1,0,0);(0,1,0)]$, scaled by $s_{1\times 1}$, and translated on the image plane by $t_{2m\times 1} = 1_{m\times 1} \otimes t_2$. Here, $\otimes$ denotes "Kronecker Product". The matrix $M_{2m\times 2n}$ is a 0/1 matrix that indicates the visibility of points. The variables R_3, s, t_2 are the 3D *pose parameters* $\theta = \{R_3, s, t_2\}$, six parameters in total, to be determined by the alignment algorithm. The *observation noise* is modeled by ζ

$$\zeta \sim \mathcal{N}(0, \Sigma), \Sigma = \Sigma_m \otimes I_2, \Sigma_m = diag\{\rho_1^2 \ldots \rho_m^2\} \tag{5}$$

It is an *anisotropic* diagonal Gaussian, which reflects the assumption that all points are located independently with different confidence ρ_i^{-2}. The assignment of the value ρ_i will be clear as the paper proceeds.

We formulate 3D alignment as a Bayesian inference problem: given the 2D observation q, estimate the *deformation parameters* b and the *pose parameters* θ by maximizing their log posterior,

$$\log p(b, \theta | q) = \log p(q | b, \theta) + \log p(b) + \text{const} \tag{6}$$

The shape prior term $p(b)$ is learned from training samples (2). The likelihood term is a mixture distribution,

$$\log p(q | b, \theta) = \log \int_S p(q|S,\theta) p(S|b) dS \tag{7}$$

where $p(S|b)$ is defined by (1), and $p(q|S,\theta)$ is defined by (4). It measures the possibility that q is generated from parameters $\{b, \theta\}$ with weak perspective projection.

3. The Algorithm

By choosing S as a hidden variable, we rewrite the log posterior (6) as

$$\begin{aligned} \langle \log p(b, \theta | S, q) \rangle &= \langle \log p(q | S, \theta) \rangle \\ &+ \langle \log p(S|b) \rangle + \log p(b) + \text{const} \end{aligned} \tag{8}$$

where the expectation $\langle\cdot\rangle$ is taken with respect to S. Note that b and θ were coupled in (7) which could not be further factorized. However, by treating S as the hidden variable and treating the pair of q and S as the complete data, the resultant *complete log posterior* (8) allows us to decompose the optimization of b and θ into separate steps, each of which can be solved effectively. The EM algorithm is designed accordingly. In the E step, we compute the "averaging" distribution used in $\langle\cdot\rangle$, i.e. $p(S|q, b, \theta)$, the posterior of S given the observed shape q and the previous estimate of the parameters $\{b^{(t)}, \theta^{(t)}\}$; In the M step, we maximize $\langle \log p\left(q|S, \theta\right)\rangle$ and $\langle \log p\left(S|b\right)\rangle + \log p(b)$ over θ and b separately.

In order to simplify our subsequent expressions, let us introduce the following notations. $\tilde{I}_3 = diag\left\{1, 1, 0\right\}$. $\widetilde{\rho}^- = \left(\rho_1^{-2}, \rho_2^{-2}, 0, \rho_3^{-2}, \ldots\right)^t$, a $n \times 1$ vector obtained by elongating the inverse variance vector of the observation noise (5), where zeros are filled into the entries of $\widetilde{\rho}^-$ that correspond to the occluded points. $\widetilde{\Sigma}^- = diag\{\widetilde{\rho}^-\}$.

3.1. E Step

First we compute the "averaging" distribution $p(S|q, b, \theta)$. Given the parameters b, θ and the observation q, the conditional distributions of S and q are as follows,

$$S|b \sim \mathcal{N}\left(\Phi b + \mu, \sigma^2 I\right) \tag{9}$$

$$q|S, \theta \sim \mathcal{N}\left(sMPRS + t, \Sigma\right) \tag{10}$$

The distribution of S is a product of (9) and (10),

$$p(S|q, b, \theta) \propto p(q|S, \theta)p(S|b) \tag{11}$$

which is still a Gaussian. Its mean and variance are

$$\begin{aligned} \mathrm{E}[S|q, b, \theta] = \mathrm{Var}[S|q, b, \theta]\, \big[(\Phi b + \mu)/\sigma^2 + \\ sR^t q^t M^t \Sigma^{-1}\left(q - t\right)\big] \end{aligned} \tag{12}$$

$$\mathrm{Var}[S|q, b, \theta] = \left(s^2 \widetilde{\Sigma}^- \otimes \left(I - r_3^t r_3\right) + \sigma^{-2} I\right)^{-1} \tag{13}$$

Observe that (13) is 3×3 block diagonal, which means that the points of S are conditionally independent. So we divide S into two parts in terms of their visibility $S = \left\{S_o, S_v\right\}$.

For the i-th occluded point S_o^i, whose shape coordinates are not changed by the observation q, (12) and (13) are rewritten as,

$$\mathrm{E}[S_o|q, b, \theta] = [\Phi b + \mu]_o^i \tag{14}$$

$$\mathrm{Var}[S_o|q, b, \theta] = \sigma^2 I_3 \tag{15}$$

Neither the mean nor the variance is changed from (9), because there is no information collected from the observation.

For the visible part, let $S_v^i = (S_v^{i,1}, S_v^{i,2}, S_v^{i,3})$ denote the i-th visible point, where $S_v^{i,\cdot}$ denotes individual coordinate. We rewrite its conditional mean and variance as,

$$\mathrm{E}[S_v^i|q, b, \theta] = R^t \begin{pmatrix} w_1 S_b^{i,1} + w_2 S_q^{i,1} \\ w_1 S_b^{i,2} + w_2 S_q^{i,2} \\ S_b^{i,3} \end{pmatrix} \tag{16}$$

$$\mathrm{Var}[S_v^i|q, b, \theta] = \left(s^2 \rho_i^{-2}\left(I - r_3^t r_3\right) + \sigma^{-2} I\right)^{-1} \tag{17}$$

where

$$S_b = R\left(\Phi b + \mu\right) \tag{18}$$

$$S_q = s^{-1} P^t M^t\left(q - t\right) \tag{19}$$

$$w_1 = \rho_i^2/(s^2\sigma^2 + \rho_i^2), \;\; w_2 = s^2\sigma^2/(s^2\sigma^2 + \rho_i^2) \tag{20}$$

The expected i-th visible point $\mathrm{E}[S_v^i|q, b, \theta]$ is indeed a weighted average of two 3D points, S_b^i and S_q^i. S_b is the shape vector computed by using the deformation parameters b; S_q is the constructed shape vector by using the 2D observation q. The weights w_1 and w_2 in (20) are determined by the variances of noises ε and η: if ρ_i is small, the observation on the i-th point is reliable so the algorithm assigns larger weight to S_q; otherwise it gives more weight to S_b. In the subsequent discussions we refer $\langle S\rangle = E[S|q, b, \theta]$ as the *augmented 3D shape*, because it is a 3D shape vector "augmented" from 2D observation.

3.2. M Step

Given the averaging distribution of S, the maximization of (8) is decomposed into two independent problems: 1) estimate the deformation parameters b by maximizing its expected log-posterior $\langle \log p\left(S|b\right)\rangle + \log p(b)$; 2) estimate the 3D pose θ by maximizing the first term $\langle \log p\left(q|S, \theta\right)\rangle$.

For the first problem we note that the posterior of b, given S, is again a Gaussian,

$$\log p\left(b|S\right) = \log p\left(S|b\right) + \log p\left(b\right) \tag{21}$$

whose mean and variance are

$$\mathrm{E}[b|S] = \Lambda\left(\Lambda + \sigma^2 I\right)^{-1} \Phi^t\left(S - \mu\right) \tag{22}$$

$$\mathrm{Var}[b|S] = \sigma^2 \Lambda(\Lambda + \sigma^2 I)^{-1} \tag{23}$$

By taking the conditional expectation $\langle\cdot\rangle$ over (21), computing its derivative with respect to b, and setting it to zero, the optimal $\widehat{b}$ is obtained as

$$\widehat{b} = \Lambda\left(\Lambda + \sigma^2 I\right)^{-1} \Phi^t\left(\langle S\rangle - \mu\right) \tag{24}$$

Note that $\widehat{b} = \mathrm{E}[b|\langle S\rangle]$ by comparing with (22). In other words, $\hat{b}$ is nothing but the mean of b given the augmented 3D shape $\langle S\rangle$. If we rewrite (24) for each element of b, we have

$$\begin{aligned} \widehat{b}_i &= \beta_i \Phi_i^t\left(\langle S\rangle - \mu\right) \\ \beta_i &= \lambda_i/(\lambda_i + \sigma^2) \end{aligned} \tag{25}$$

Eq.(25) can be viewed as a regularization process: the augmented 3D shape $\langle S\rangle$ is projected onto the principal subspace; then the projection coefficient is shrunk toward the mean shape along each principal direction. The degree of shrinkage is controlled by the eigenvalue λ_i.

Next, we proceed to compute the optimal pose $\widehat{\theta} = \{R, s, t\}$. Maximizing $\langle \log p\,(q|S,\theta)\rangle$ would be obtained by minimizing the following conditional expectation of a weighted distance error d,

$$L_1 = \langle d^t \Sigma^{-1} d\rangle,\ d = q - sMPRS - t \tag{26}$$

Minimizing the loss function L_1 will involve the second order statistics $\langle SS^T\rangle$. Instead, let us consider an alternative by replacing S with the augmented 3D shape $\langle S\rangle$

$$L_2 = \widetilde{d}^t \Sigma^{-1} \widetilde{d},\ \widetilde{d} = q - sMPR\langle S\rangle - t \tag{27}$$

Minimization of L_1 can be approximated by minimization of L_2, because given q, b, θ the variance (17) of S is consistently small. Indeed, the difference between L_1 and L_2

$$L_1 - L_2 = \sum_{i=1}^{m} s^2\sigma^2/(s^2\sigma^2 + \rho_i^2) \tag{28}$$

is always considerably smaller than the value of L_2. By discarding the occluded points from $\langle S\rangle$ and assigning weights to the visible points,

$$\begin{gathered} q' = Wq,\ S' = W\langle S\rangle_v \\ W = diag\{\rho_1^{-1}, \ldots, \rho_m^{-1}\}/\Sigma\rho_j^{-1} \end{gathered} \tag{29}$$

Minimizing (27) leads us to the solution of $\widehat{\theta}$ as

$$\widehat{t} = c_{q'} - \widehat{s}P\widehat{R}C_{S'} \tag{30}$$

$$\widehat{s} = \left\langle q'_c, P\widehat{R}S'_c \right\rangle / \|P\widehat{R}S'_c\|^2 \tag{31}$$

$$\widehat{R} = \operatorname*{argmax}_R \langle q'_c, PRS'_c\rangle^2 / \|PRS'_c\|^2 \tag{32}$$

where $c_{q'}$ and $C_{S'}$ are the centroids of q' and S', respectively. Here q'_c and S'_c denote the centralized vectors. See the appendix for the detailed derivation.

3.3. Iterative EM Shape and Pose Estimation

The E Step and the M step developed in 3.1 and 3.2 are now put together into an iterative algorithm. Given the 2D shape observation q, the current shape estimate b^t, and the current pose θ^t, it performs:

1. *Shape augmentation*: generate the “augmented” 3D shape $\langle S\rangle$ from $\{q, b^t, \theta^t\}$ in two steps: 1) For the occluded points, simply replicate the coordinates of occluded points from S_b (14). 2) For the visible points of $\langle S\rangle$, replicate the depth by that of S_b (16); compute their image plane coordinates by a weighted average of S_b (18) and S_q (19).

2. *Shape regularization*: smooth $\langle S\rangle$ by projecting it into the principle subspace with shrinking the projection coefficients by (25). That gives the updated deformation parameters b^{t+1}.

3. *Pose estimation*: estimate θ^{t+1} ($30 \sim 32$) by minimizing L_2 (27).

Steps $1 \sim 3$ are repeated until convergence.

3.4. Discussion

Weighting and regularization may merit a few discussions.

Weighting for outlier resistance: Resistance to outliers is achieved by weighting. Note that in both the shape augmentation step and the pose estimation step, the weight (20) (29) associated with each observed point is inversely proportional to its observation variance ρ_i. For outliers with higher variance, smaller weights are assigned to suppress their influence. The value of ρ_i is adjusted in an iterative reweighting manner. The initial value of ρ_i is set to be same for all i. Suppose q^{t-1} is the previous 2D observation, and p^t is the 2D projection of the current 3D shape $S_b^t = \phi b^t + \mu$. We measure the *fitting error* for every point $e_i^t = \|q_i^{t-1} - p_i^t\|$, then use it to update $\rho_i^t = ce_i^{t-1}$. Normally outliers produce larger matching errors. The weights associated with every point in (20) (29) are adjusted accordingly.

Regularization by shrinkage: In deformable model matching, regularization is crucial for generating a smooth model from a noised observation. Our algorithm regularizes the 3D shape $\langle S\rangle$ by a ”shrinkage” process (25). It gives a greater amount of shrinkage in the directions with smaller eigenvalues λ_i, and vice versa. Recall that large shape deformations, such as expression change or varying degrees of fatness, are encoded in the first several principal directions. Our algorithm therefore encourages to preserve these “meaningful” deformations, and penalizes minor deformations that often correspond to random landmark perturbations. It also controls the overall penalty by the variance of shape noise σ^2, in other words, by the number of principal components. It can be shown that $\hat{b}$ is a Bayes estimator of the shape deformation in $\langle S\rangle$.

4. Aligning 3D Model to an Input Image

Given an input image, we construct a three-layered Gaussian pyramid, and apply the alignment algorithm sequentially from the coarsest layer to the finest. The alignment algorithm works as shown in Table 1. With the iterative EM estimation process at its core, the algorithm consists of initialization and main loop.

Initialization: The initialization procedure is automatic: The deformation parameters are initialized as zero; The 3D pose parameters are initialized by Schneiderman&Kanade face detector [8]. The initial roll angle is set to be one of $\{-90^o, -45^o, 0^o, 45^o, 90^o\}$, according to the output channel of the face detector; the initial pitch and yaw angles are

1. Initialize the deformation parameters $b^0 = 0$; initialize pose parameters θ^0 by a face detector [8] that gives the rough location and orientation of the face.
2. *Occlusion query*: render the 3D face $S = \mu + \Phi b$ under pose θ; identify the visible points by querying Z-buffer to update the "visibility" matrix M.
3. *2D observation*: update the location of visible points q by the feature detectors.
4. *Shape and pose inference*: estimate shape and pose by applying the core algorithm described in section 3.3.
5. *Re-weighting*: evaluate the fitting errors and update the weights for every point.
6. Outer loop: repeat steps $2 \sim 5$ until convergence.

Table 1. Fitting the model to an image.

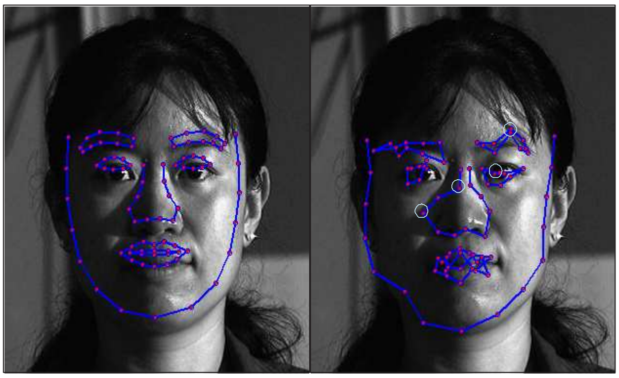

Figure 2. *Local feature point detection. Left: the initial projection of the 3D mean face; Right: the resultant positions obtained by individual local searching. Only part of points are shown and connected for display purpose. Since the feature detectors respond to gradient magnitude, it can be observed that many points are moved to the positions with strong edge response (as highlighted by white circles), but not necessarily the correct positions. It demonstrates that patch information alone is not enough to locate these points.*

set as zero. The initial scale and translation are computed from the outputs of the face detector.

Feature descriptor: In our system we use a very simple descriptor that is constructed from a 9×9 patch. The gradient magnitude is computed for every pixel in the patch, then stacked into a feature vector. We normalize the l_1 norm of the vector to one.

Feature point detector: We sample the patches from all training faces, and compute their mean and variance for each point under each view. Given the initial pose estimate, we find the closest training pose, and the associated feature detectors. We project the 3D shape onto the image plane, search over the neighboring region for each 2D point individually, and find the best matching. The goodness of matching is measured by the Mahalanobis distance. Figure 2 gives us an example of the feature detection results. We highlight several failed points by circles. In general, it is very difficult to localize their positions using only patch information.

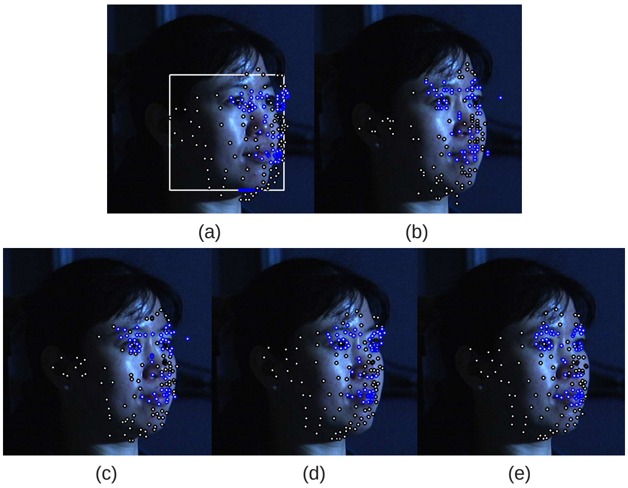

Figure 3. *Illustration of the fitting process. (a) initialization (b) 2D observation (c) augmented shape (d) regularized shape (e) converged*

Illustration of Fitting Process: Figure 3 visualizes the whole model fitting process for an example input of a half profile face. The figure shows the 2D projection of the 3D shape obtained at each stage. Figure (a) shows the projection of the mean shape using the initial pose. The position of visible points is then updated by individual local searching, and the result is shown in (b). Observe that although many points are moved toward the correct positions, the others are affected heavily by the specular light and the image noises around the face area. Also note that the spatial positions between the points are often inconsistent, because their position is updated independently for each point. The algorithm then produces an augmented 3D shape (c) by averaging the observation and the current shape estimate (mean shape in the first iteration). At this moment the algorithm has not identified outliers since the weights associated with all points are equal. Next, the augmented shape is regularized by the shrinkage process, and the resultant regularized shape is shown in (d). Note now that the protrusion appeared in (c) is smoothed out, and the spatial arrangement of the whole pattern is more regular. Meanwhile, the major shape deformations captured by the 2D observation is preserved in (d). The (c) and (d) steps are repeated iteratively until the EM algorithm converges. Then the points are reweighted according to their fitting errors, that is, the pairwise distance between (b) and (d). Observe that the outlier points are assigned with lower weights in the subsequent computations. The image (e) shows the final result.

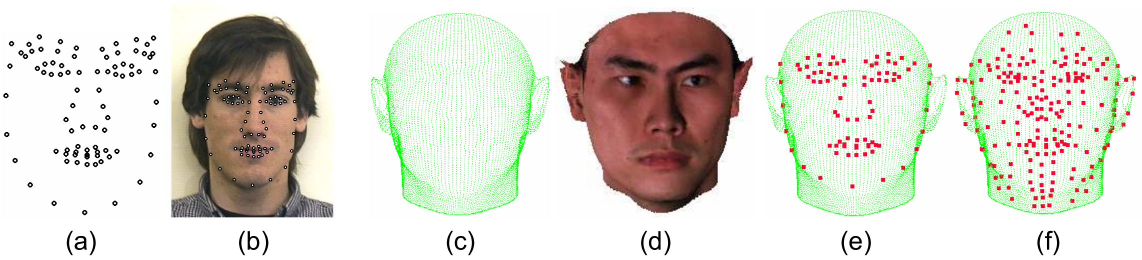

Figure 4. *Create synthetic 3D faces. (a) 2D reference model with 83 landmarks; (b) 2D frontal face image; (c) 3D reference model with 8895 vertices; (d) 3D laser scan; (e) correspondence between the 2D and 3D reference models; (f) additional points selected from the 3D face surface*

5. Experiments

Our experiments involve three face databases: A) *AR Database* [9]: 720 frontal face images, each image is manually labeled with 83 landmarks. B) *USF Human-ID Database* [4]: 100 laser scanners aligned to a 3D reference model of 8895 points. C) *CMU PIE Database*[10]: 4488 images of 86 people varied in pose and illumination. There is no overlap among these databases. We create a synthetic 3D face database from A and B for training. We test the algorithm on database C.

5.1. Synthetic Training Faces

Collecting and labeling a large number of 3D faces is itself a difficult problem. Several techniques have been developed to establish the correspondence among 3D laser scans automatically, but no guarantee can be made as to the correctness. In our work we adopt a different strategy: use synthetic faces instead of real faces for training.

Recall that the 2D face model uses 83 points (Figure 4(a)), and each image in database A are labeled with those landmark points (4(b)). In the meantime, each 3D laser scanned face in database B (4(d)) are marked with 8895 reference points (4(c)). Therefore once we establish manually the correspondences (4(e)) between those 83 points and 8895 points, we know the texture mapping between any pair of image I in database A and 3D face L in database B. A new virtual 3D face is generated automatically from I and L by follows,

1. Estimate pose: Compute the relative 3D pose of I with respect to L that minimizes their shape difference on the image plane.

2. Generate shape: Project L onto the image plane according to that pose; Warp the projected shape to fit I by RBF regression; Replicate the image plane coordinates of the virtual face from I; Replicate the depth from L.

3. Generate texture: Extract texture from the 2D image, and if holes (missing textures) exist, fill them in by interpolation.

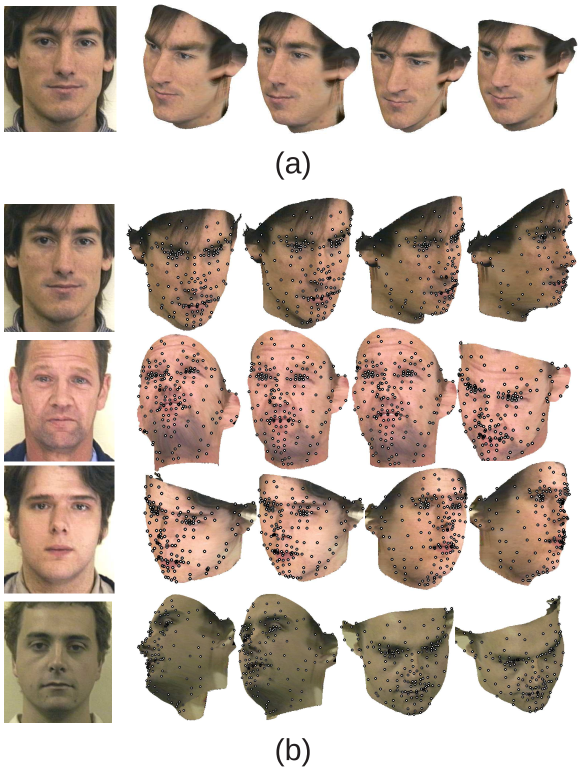

Figure 5. *Synthetic 3D face database. The original 2D images are shown in the first column; novel views synthesized from the generated 3D faces are shown in other columns. (a) by replicating different depth from different laser scans we can create multiple 3D faces from one image; (b) the correspondence across different faces and difference views.*

Repeating this process for every pair of image in databases A and 3D laser scan in databases B, we produce a gigantic database of over $50,000$ synthetic 3D faces with established correspondences. Observed from Figure 5, the synthetic 3D faces may or may not correspond to any "real" person, but visually they are all plausible face instances. Although there exists redundancy among the 3D faces generated from the same image, the synthesized 2D patches are all different because the depth is different.

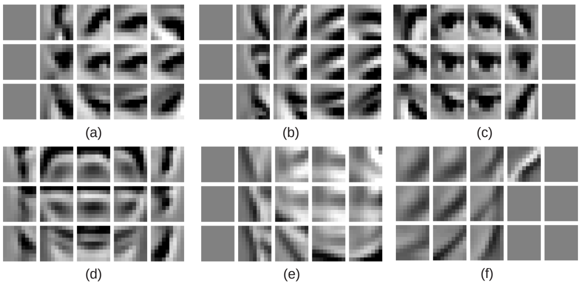

Figure 6. *The mean patches sampled at six key feature points from fifteen views. (a) left outer eye corner; (b) left eyebrow corner; (c) right eyeball center; (d) the lowest point on mouth contour; (e) the point at the center of left lower cheek; (f) the point on the contour of right lower cheek.*

5.2. Training

The 3D shape priors are learned from the synthetic training database via PCA. To construct the feature detectors, we render all 3D faces in 15 different views: $\{-90^o, -45^o, 0^o, 45^o, 90^o\}$ for roll, and $\{-35^o, 0^o, 35^o\}$ for pitch. The direction and intensity of the light source are varied randomly within a certain range. A three-layer Gaussian pyramid is constructed on the synthesized image, and for each layer, the gradient magnitude is computed for every pixel. The 9×9 patches centered on every visible point are sampled, and their mean and variance across training samples are computed. In figure 6, we show the mean patches of several key facial points sampled from the coarsest layer.

5.3. Testing Results

Figure 7 shows some typical alignment results on images from PIE database. The rectangles and the bars at the bottom denote the initial pose. The resultant 3D shape and pose are shown by projecting the 3D shape into the image plane with the estimated 3D pose. Our algorithm successfully align those key facial parts, such as eyes, noses and mouthes, as well as the occluding contours under different viewpoints, for the test images contain substantial pose variation and illumination changes. It has been observed that the algorithm has difficulty to align ears, which in turn may also affect the estimate of the scale. This is due to the missing texture of ears in the synthetic database.

6. Summary and Future Work

For automatic 3D face alignment, we have proposed a deformable model consisting of a sparse 3D points and view-based patch appearance. We have also proposed an algorithm to estimate 3D shape and pose from a noisy 2D shape observation. Given a single image, the resultant alignment provides locations of facial landmarks, as well as 3D shape and pose. They are useful for many applications, such as model based tracking and 3D reconstruction, which currently heavily depends on human intervention for initialization.

A problem of fitting 3D model to a single image is the depth ambiguity. Within the same framework, we have extended the inference algorithm to simultaneously align pairs of or multiple views of a rigid face [11]. Future work includes generalizing the prior model or the projection model for the alignment of other objects.

Appendix: Estimate Rotation

The loss function (32) can be reduced to a simple form as follows,

$$\begin{array}{c} \text{maximize} \;\; |a^t r|^2 \;\; \text{over} \;\; r \\ \text{s.t.} \;\; 1) r^t \Lambda r = 1 \;\; 2) \;\|r_1\| = \|r_2\| \;\; 3) {r'}^t_1 r'_2 = 0 \end{array} \tag{33}$$

We rewrite (32) as (34) by re-arranging the elements of q' and S',

$$\frac{\mathrm{Tr}\left\{\mathcal{Q}^T_{n\times2}\mathcal{S}_{n\times3}R_{3\times3}P_{3\times2}\right\}}{\mathrm{Tr}\left\{P^T_{3\times2}R^T_{3\times3}\mathcal{S}^T_{n\times3}\mathcal{S}_{n\times3}R_{3\times3}P_{3\times2}\right\}} \tag{34}$$

The rows of $\mathcal{Q}_{n\times2}$ and $\mathcal{S}_{n\times3}$ correspond to the points in q' and S'. Applying SVD decomposition to $\mathcal{S} = VDU^t$, and after a sequence of substitutions $R_{3\times3} = [r_1, r_2, r_3]$, $r'_i = U^t r_i$, $r_{6\times1} = \left[{r'}^t_1, {r'}^t_2\right]^t$, $\mathcal{Q}_{n\times2} = [q_1, q_2]$, $q'_i = DV^t q_i$, $a_{6\times1} = \left[{q'}^t_1, {q'}^t_2\right]^t$, we end up with a simplified form as (33). It is an over-constrained nonlinear optimization problem that can be solved effectively by a generalized Newton method in $10 \sim 30$ iterations.

References

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

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

[3] Y. Zhou, W. Zhang, X. Tang, H. Shum, A Bayesian Mixture Model for Multi-View Face Alignment. CVPR 2005.

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

[5] V. Blanz and T. Vetter, Face Recognition Based on Fitting a 3D Morphable Model, *IEEE Transactions on PAMI*, 2003.

[6] Z. Zhang, Z. Liu, D. Adler, M. F. Cohen, E. Hanson, Y. Shan, Robust and Rapid Generation of Animated Faces From Video Images - A Model-Based Modeling Approach, *International Journal of Computer Vision*, 2004.

[7] M. Dimitrijevic, S. Ilic, P. Fua, Accurate Face Models from Uncalibrated and Ill-Lit Video Sequences, *In Conference on Computer Vision and Pattern Recognition*, 2004.

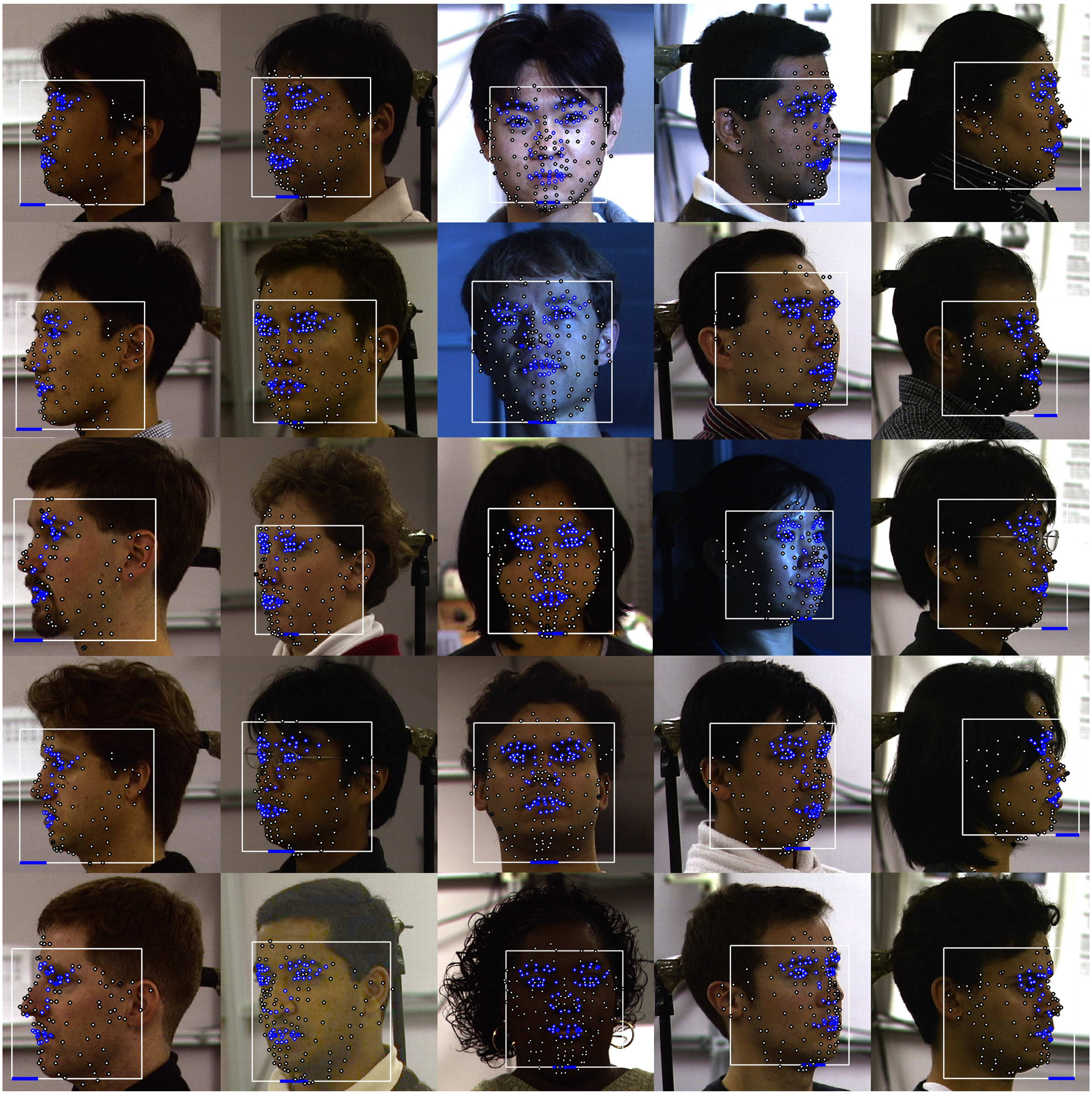

Figure 7. *Alignment results. The white rectangle denotes the initial face location, and the blue bar at the bottom of the rectangle denotes the initial orientation. The 3D shape is projected onto image plane according to their 3D pose estimate. The points along the contour of key facial parts (eye, nose, mouth and eyebrow) are highlighted in blue for display purpose. All other points are white.*

[8] H. Schneiderman, T. Kanade, A Statistical Method for 3D Object Detection Applied to Faces and Cars, *In Conference on Computer Vision and Pattern Recognition*, 2000.

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

[10] T. Sim, S. Baker, and M. Bsat, The CMU Pose, Illumination, and Expression Database, *IEEE Transactions on PAMI*, Vol. 25, No. 12, December, 2003.

[11] L. Gu and T. Kanade, 3D Alignment of Face in Multiple Images, in draft, 2006.

[12] Moghaddam B. and Pentland A, Probabilistic Visual Learning for Object Representation, *IEEE Transactions on PAMI*, 19 (7), pp. 696-710, July 1997.