An Integrated Model for Accurate Shape Alignment

Lin Liang, Fang Wen, Xiaoou Tang, and Ying-qing Xu

Visual Computing Group, Microsoft Research Asia, Beijing 100080, China
{lliang, fangwen, xitang, yqxu}@microsoft.com

Abstract. In this paper, we propose a two-level integrated model for accurate face shape alignment. At the low level, the shape is split into a set of line segments which serve as the nodes in the hidden layer of a Markov Network. At the high level, all the line segments are constrained by a global Gaussian point distribution model. Furthermore, those already accurately aligned points from the low level are detected and constrained using a constrained regularization algorithm. By analyzing the regularization result, a mask image of local minima is generated to guide the distribution of Markov Network states, which makes our algorithm more robust. Extensive experiments demonstrate the accuracy and effectiveness of our proposed approach.

1 Introduction

Shape alignment is a fundamental problem in computer vision with applications in many areas, such as medical image processing [1], object tracking [2], face recognition and modeling [3], and face cartoon animation [4]. Accurate alignment of deformable shapes or contours depends on estimation of optimal deformable shape parameters such that the deformed shape model matches the image evidence collected from images or video.

A number of different shape models have been proposed for shape alignment. One approach is to postulate the deformation parameters by reducing the shape deformation correlations. The shape prior is then modeled by the distribution of the deformation parameters. The leading work of this approach is the active shape model (ASM) [5]. In light of this work, several improved methods have been developed. A Bayesian tangent shape model and an EM based searching algorithm are proposed in [6] to make the parameter estimation more accurate and robust. To alleviate the local minima problem, Liu et.al. [7] designed a hierarchical shape model and DDMCMC inference algorithm. To handle the nonlinear shape variance, a mixture of Gaussians [8] and kernel PCA [9] are used to model the distribution of deformation parameters. For optimization, these methods usually generate an observed shape by sampling each feature point independently from a local likelihood and then regularize it using the shape prior model. The main advantage of these methods is that the global regularization step based on a shape prior may help to assure an overall shape reasonably in line with the object. However, since each feature point is sampled without considering its relationship with neighbor points, the observed location

© Springer-Verlag Berlin Heidelberg 2006

of each individual point is very sensitive to noise. To avoid the influence of the faraway outliers, some works such as [10] imposed a simple smoothness constraint between the neighbor points and used Dynamic Programming (DP) to find an observed optimal shape. Unfortunately, the observed shape optimized by DP was still directly regularized by the PCA shape model. The problem is that the regularized shape is usually placed at a mean position with the minimal sum of points distance to the observed optimal shape. Thus it cannot guide each "bad" point of observed shape to the accurate location. Moreover, it may even drag away the already accurately aligned points. Therefore, alignment accuracies of these methods are in general insufficient for many applications.

Recently, a different type of shape alignment method based on the Markov Random Field model has been proposed [11]. In this method, each feature point is considered as a node in a graph, and a link is set between each pair of feature points with the interaction energy designed to impose the local structure constraints between them. The benefit of such a model is that the shape prior is distributed in a Markov network of components and the image observation is still distributed by modeling the image likelihood of each individual component. The close interaction between the local image observation and structure constraints leads to far more accurate local shape estimation. The shortcoming of such an approach is that it models the shape only in a local neighborhood. Such a low level model cannot capture high level semantics in the shape. The lack of a global shape prior often leads the methods to nonstable results.

In this paper, by combining the advantages of the above two approaches, we propose an integrated model for accurate shape alignment. At the low level, the shape is modeled as a Markov Network with simple structure to effectively capture the local geometry constraints. At the high level, a global points distribution model based on PCA is adopted to regularize the inferred shape from the Markov Network. In order to avoid a decrease in accuracy during regularization, a constrained regularization algorithm is developed to keep the "good" point positions. The information from the global model is also fed back to the next Markov Network inference step by a mask image to guide the distribution of Markov Network states. This scheme effectively prevents the Markov Network from sticking to the local minima. The accuracy of the proposed approach has been demonstrated by extensive experiments.

2 A Two-Level Shape Model

In this section, we present a two-level model for shape alignment. The shape is split into a set of line segments and modeled as a Markov Network with simple structure to make the searching stage more effective and robust. At the same time, all the line segments are correlated through a global point distribution model to guarantee a globally reasonable shape.

2.1 Low Level Shape Model Based on Markov Network

Assuming that a shape $\mathbf{S}$ is described by N feature points (x_i, y_i) in the image, we can represent it by a $2N$-dimensional vector $\mathbf{S} = \{(x_i, y_i), i = 1, ..., N\}$.

We break the shape $\mathbf{S}$ into a set of line segments by the feature points. The parameters of each line segment $\mathbf{q}_i$ are the coordinates of its two endpoints $\mathbf{q}_i = [\mathbf{w}_i^s, \mathbf{w}_i^e]$. As shown in Figure 1 (a), these line segments are the nodes in the hidden layer of the graph. If two nodes are correlated, there will be an undirected link between them. For a deformable shape, we put a link between the connected line segments.

Assuming the Markovian property among the nodes, the shape prior can be modeled as $p(\mathbf{Q}), \mathbf{Q} = \{\mathbf{q}_0, \mathbf{q}_1, ..., \mathbf{q}_K\}$, which is a Gibbs distribution and can be factorized as a product of all the potential functions over the cliques in the graph:

$$p(\mathbf{Q}) = \frac{1}{Z} \prod_{c \in C} \psi_c(\mathbf{Q}_c) \tag{1}$$

where C is a collection of cliques in the graph and $\mathbf{Q}_c$ is the set of variables corresponding to the nodes in clique c, and Z is the normalization constant or the partition function.

In the context of deformable shapes, we adopt a pairwise potential function $\psi_{ij}(\mathbf{q}_i, \mathbf{q}_j)$ to present the constraint between two connected line segments. Thus we write the shape prior $p(\mathbf{Q})$ as:

$$p(\mathbf{Q}) = \frac{1}{Z} \prod_{(i,j) \in C^2} \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) \tag{2}$$

The pairwise potential function is defined by the constraints of the distance of two endpoints ($\mathbf{w}_i^e$ and $\mathbf{w}_j^s$) and the angle γ_{ij} between the two line segments, as illustrated in Figure 2:

$$\psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) = G(d_{ij}; 0, \sigma_{ij}^d) \cdot G(A_{ij}; \mu_{ij}^A, \sigma_{ij}^A) \tag{3}$$

where $d_{ij} = |\mathbf{w}_i^e - \mathbf{w}_j^s|$ is the distance between $\mathbf{w}_i^e$ and $\mathbf{w}_j^s$, $A_{ij} = \sin(\gamma_{ij})$, and σ_{ij}^d and σ_{ij}^A are variance parameters that control the tightness of the connectivity constraint.

Given the image observation I, as shown in Figure 1 (a), each segment $\mathbf{q}_i$ is also associated with its image observation, denoted as Γ_i. Assuming the local observation is independent of other nodes given $\mathbf{q}_i$, the likelihood is factorized as:

$$p(I|\mathbf{Q}) = \prod_i p_i(\Gamma_i|\mathbf{q}_i) \tag{4}$$

Then the posterior can be factorized as:

$$p(\mathbf{Q}|I) \propto \frac{1}{Z} \prod_i p_i(\Gamma_i|\mathbf{q}_i) \prod_{(i,j) \in C^2} \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) \tag{5}$$

For a complex shape, such as the face that contains multiple parts (brows, eyes, etc.), only the connected line segments belonging to the same part are linked, as shown in Figure 1 (b). We do not add links between different parts. This will guarantee that the good parts (with strong local image observation) move to its accurate position while not being affected by other bad parts. Such a simple structure is also easy for inference. The global relationship between

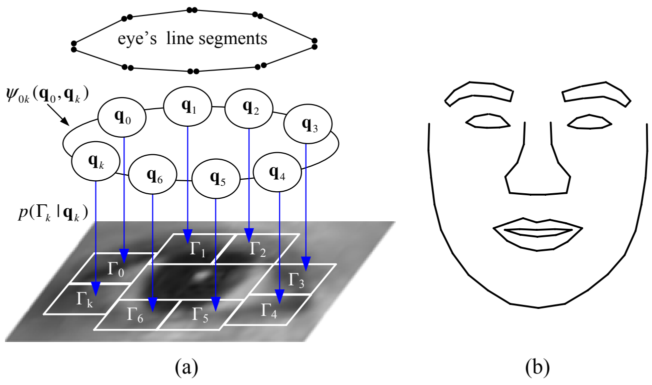

Fig. 1. The Markov Network for face. (a) The graph for the eye shape. (b) For the graph of face, the links are added within each black line.

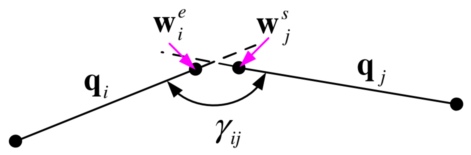

Fig. 2. The constraint of two connected line segments

different parts is constrained by a global shape model as explained in the following sections.

2.2 Global Shape Model Based on PCA

We adopt principal components analysis (PCA) to model the points distribution of a shape $\mathbf{S}$ as in [5]. To remove the shape variation caused by the global transformation, the shape is first aligned to the tangent space:

$$\mathbf{S} = cR_\theta\mathbf{x} + \mathbf{t} \tag{6}$$

where $\mathbf{x}$ is the tangent shape vector, c is the scaling parameter, R_θ is the rotation matrix and $\mathbf{t}$ is the translation parameter.

PCA is then adopted to find a set of main deformable modes. The deformable shape can be generated by a linear combination of these modes. Suppose r modes are retained in PCA, then:

$$\mathbf{x} = \mu + \phi_r\mathbf{b} + \varepsilon \tag{7}$$

where μ is the mean shape, and ϕ_r consists the first r columns of the projection matrix. Each column of ϕ_r corresponds to a deformable mode. ε is an isotropic noise in the tangent space. $\mathbf{b}$ is a hidden variables vector to generate the shape. Each item of $\mathbf{b}$ is independent and $\mathbf{b}$ is distributed as a multivariate Gaussian:

$$p(\mathbf{b}) = G(\mathbf{b}; 0, \Lambda) \tag{8}$$

where $\Lambda = diag(\lambda_1, ..., \lambda_r)$. λ_i is the ith eigenvalue.

Under such model, the global shape prior is modeled as $p(\mathbf{S}) \sim p(\mathbf{b})$.

2.3 An Integrated Model

In our integrated model, a deformable shape is parameterized by the hidden variable $\mathbf{b}$, the pose parameters $\Theta = (c, \theta, \mathbf{t})$ and the positions of the line segments $\mathbf{Q} = \{\mathbf{q}_i\}$. Given an observation image I, we want to maximize the following posterior $p(\mathbf{b}, \Theta, \mathbf{Q}|I)$ to get the optimal shape. Notice that given $\mathbf{Q}$, $\mathbf{b}$ and Θ can be considered as independent of I, thus

$$p(\mathbf{b}, \Theta, \mathbf{Q}|I) \propto p(\mathbf{b}, \Theta|\mathbf{Q})p(\mathbf{Q}|I) \tag{9}$$

$p(\mathbf{Q}|I)$ is defined in Equation (5), and the likelihood and low level's shape prior are factorized in this part. The final shape $\mathbf{S}$ can be easily got from $\mathbf{Q}$ by setting the position of one point s_i as the average position of the connected segments' endpoints, which can be written as $\mathbf{S} = T\mathbf{Q}$, so $p(\mathbf{b}, \Theta|\mathbf{Q})$ can be evaluated as $p(\mathbf{b}, \Theta|T\mathbf{Q})$, which is:

$$p(\mathbf{b}, \Theta|\mathbf{Q}) \propto p(T\mathbf{Q}|\mathbf{b}, \Theta)p(\mathbf{b}) \tag{10}$$

Maximizing the posterior equation (9) is equal to minimizing the following energy:

$$E = \lambda E_p + E_{mn} \tag{11}$$

where

$$E_p = -\log p(\mathbf{b}, \Theta|\mathbf{Q}) \tag{12}$$

$$E_{mn} = -\log p(\mathbf{Q}|I) \tag{13}$$

λ is a weighting parameter to balance the strength of the global shape prior and the Markov Network.

Based on this integrated model, the local and global shape constraints are imposed by Markov Network and the global PCA model respectively. The local structure constraints make the feature points more consistent with the image observation and robust to the local noise, thus a more accurate result can be achieved. And the global shape constraint regularizes the globally unreasonable shape to obtain a more stable result.

Besides, in order to make the inference more efficient, we adopt a hierarchical shape model similar to [7]: the shape resolution changes from coarse to fine corresponding to a Gaussian pyramid of the image. The located coarse shape is used to initialize the search for a finer shape in the higher resolution image.

Hierarchical Shape Model. Suppose $\mathbf{S}_l$ is the shape at the resolution level l and $\mathbf{S}_{l+1}$ is the shape at the coarser resolution level $l+1$, we model the conditional distribution $p(\mathbf{S}_l|\mathbf{S}_{l+1})$ as a Gaussian, as in [7]. Given the located coarse shape $\mathbf{S}_{l+1}$, we initialize the finer shape $\mathbf{S}_l$ as:

$$\mathbf{S}_l^0 = \max_{\mathbf{S}_l} p(\mathbf{S}_l|\mathbf{S}_{l+1}). \tag{14}$$

3 Shape Alignment by the Two-Level Shape Model

To find the optimal shape which minimizes the energy Equation (11), we update the shape to minimize E_{mn} and E_p iteratively. To minimize E_p, the previous methods such as [5] usually place the regularized shape at a mean position with the minimal sum of points distance to the shape obtained in the E_{mn} minimization step. Thus some points that have already converged to the good positions will be dragged away by those "bad" points. Also, although some "bad" points are regularized in the E_p minimization step, they may fall into the same false positions again in the next E_{mn} minimization step.

In order to solve these problems, we propose a new iterative optimization strategy. To avoid a decrease of the accuracy during the E_p minimization step, a constrained regularization algorithm is proposed to detect and keep the "good" points. Also to prevent the search from falling into the same mistake twice, the information from the global shape model is fed back to the next E_{mn} minimization step by recording the "bad" areas. Such optimization strategy helps to achieve more accurate and robust alignment result.

3.1 Alignment Algorithm Overview

We design a multi-resolution iterative search strategy to find the optimal shape. For searching at resolution level l, the algorithm is summarized as follows:

1. Given a located coarse shape $\mathbf{S}_{l+1}$, initialize the finer shape $\mathbf{S}_l^0$ by Equation (14);
2. In the neighboring region of current shape $\mathbf{S}_l^t$, find the optimal line segments set $\mathbf{Q}_l^t$ to minimize the E_{mn} in Equation (13), see subsection 3.2;
3. Update the global shape and pose parameters $\{\mathbf{b}, \Theta\}$ to fit $\mathbf{Q}_l^t$ using constrained regularization algorithm, and try to find a better shape $\tilde{\mathbf{S}}_l^t$ with lower posterior energy in Equation (11), see subsection 3.3;
4. If a better shape $\tilde{\mathbf{S}}_l^t$ is found: $\mathbf{S}_l^t = \tilde{\mathbf{S}}_l^t$, and also create the mask image for guiding the distribution of Markov Network states. Else, $\mathbf{S}_l^t = T\mathbf{Q}_l^t$, and repeat until convergence, see subsection 3.4.

The algorithm is converged if the posterior energy of current step is larger than the previous step, i.e. $E(\mathbf{S}_l^t) > E(\mathbf{S}_l^{t-1})$, or the changing is small enough, i.e. $E(\mathbf{S}_l^t) - E(\mathbf{S}_l^{t-1}) < \delta$.

3.2 Local Search by Markov Network

Given the current shape $\mathbf{S}_l^t$, we first discretize the state space of the Markov Network in the neighbor of $\mathbf{S}_l^t$. Then for an open curve, such as the facial contour, we adopt DP to find the optimal solution which is very efficient. For a closed curve, such as the eye part, we use loopy belief propagation (BP) [12] which produces excellent empirical results for a graph containing loops [13].

To make the inference more efficient, we generate an efficient state set by pruning states according to the strength of the local likelihood, then generating new possible states based on the Markovian property.

In more details, as shown in Figure 3, inspired by ASM [5], we distribute the possible states along the profiles of $\mathbf{S}_l^t$. Then we select the states from coarse to fine: the line segments connecting the profile points sampled at a larger step are tested first and those with strong local likelihood are selected. Then we select stronger ones in the neighborhood of the kept line segments.

States pruning based on local likelihood does not consider the relationship between one node and its neighbors, thus some key states may be missed. As shown in Figure 3, if the two green states are selected for $\mathbf{q}_{i-1}$ and $\mathbf{q}_{i+1}$, but the red one of $\mathbf{q}_i$ connecting them is not selected, we cannot find a good solution. So we add new possible states based on the Markovian property: for a node $\mathbf{q}_i$, given the states of its neighbors, we add the connections with a large enough potential as the new states of $\mathbf{q}_i$. Based on the above states discretization mechanism, 15 states kept for each node are enough for inference, which makes the algorithm more efficient.

Local search by the Markov Network will make the algorithm more robust to local noise. As shown in Figure 4, in the previous approaches [5][6], each feature point is moved independently. The problem is that although some points have moved to good positions, the final shape will still be dragged to a bad position after the regularization because of those bad points, as shown in Figure 4 (b)

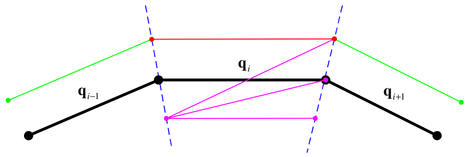

Fig. 3. Generate Markov Network states. The black line (most thick black line in printed version) is the current shape $\mathbf{S}_l^t$. $\mathbf{q}_i, \mathbf{q}_{i-1}, \mathbf{q}_{i+1}$ are linked Markov Network nodes. The pink line segments (3 line segments starting from the same point) are some candidate states of $\mathbf{q}_i$. If the two green line segments (line segments up the $\mathbf{q}_{i-1}$ and $\mathbf{q}_{i+1}$) are selected by $\mathbf{q}_{i-1}$ and $\mathbf{q}_{i+1}$, the red line segment (line segment up the $\mathbf{q}_i$) should be added to $\mathbf{q}_i$ for inference.

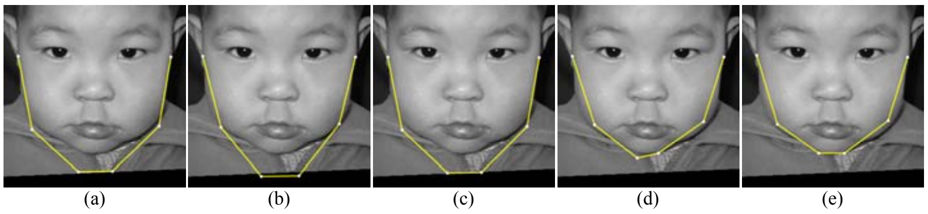

Fig. 4. Comparing the result of one step local search by Markov Network with ASM's approach. The search is executed at the low resolution (64x64) level with a ± 4 pixels search range along the shape profile for both methods. (a) The initial shape. (b) The result of ASM's approach. (c) The result of regularizing the shape of (b) with the shape model. (d) The result of local search by Markov Network. (e) The result of regularizing the shape of (d) in our constrained way.

and (c). As a result, a precise alignment result cannot be achieved. In our integrated model, by distributing the shape prior into the nodes of the Markov Network, the movement of one point is affected by the local image observation and the messages from its neighbors together, thus if the strength of the good points are strong enough, other points can be dragged away from the bad areas, as shown in Figure 4 (d) and (e). From our experiments, in many cases our algorithm can drag the shape out of local minima and generate a more precise result.

3.3 Constrained Regularization

Only minimizing E_{mn} can not guarantee a globally reasonable shape, especially when a strong edge appears nearby the ground truth, as shown in Figure 6 (a). Given the shape $\mathbf{S}^*$ obtained by Markov network inference, we expect the regularization of $\mathbf{S}^*$ will drag the "bad" points toward the better positions and maintain the positions of those "good" points. However, directly minimizing E_p as in ASM [5] does not meet this expectation. Some "good" points are dragged away as shown in Figure 6 (b). As a result, E_p is minimized but E_{mn} becomes much larger. To solve this problem we propose a constrained regularization algorithm . The key idea is to select some points of $\mathbf{S}^*$ that need to be fixed, then add these constraints during E_p minimization step.

We formulate Equation (12) more clearly:

$$E_p = \Delta^T \Sigma_l^{-1} \Delta + \mathbf{b}^T \Sigma_b^{-1} \mathbf{b} \tag{15}$$

where $\Delta = \mathbf{S}^* - (cR_\theta \mathbf{x} + \mathbf{t}), \mathbf{x} = \mu + \phi_r \mathbf{b}$. We add constrains by the likelihood variance matrix $\Sigma_l = diag(\sigma_1^2, \sigma_2^2, ..., \sigma_N^2)$. σ_i is set as a smaller value, if the corresponding point needs to be constrained. To find the optimal $\{\mathbf{b}, c, \theta, \mathbf{t}\}$, we adopt EM algorithm as [6].

To explain how we add constraints, we define the local energy of a feature point s_i first. Denoting $\{\mathbf{q}_k^i\}_{k=1}^n$ as the line segments connecting to point s_i, the local energy is defined as the mean of these line segments' likelihoods and their potentials:

$$e(s_i) = -\frac{1}{n} \sum_k \log p(\Gamma_k | \mathbf{q}_k^i) - \frac{1}{m} \sum_{k,j} \log \psi_{kj}(\mathbf{q}_k^i, \mathbf{q}_j^i) \tag{16}$$

Potential $\psi_{kj}(\mathbf{q}_k^i, \mathbf{q}_j^i)$ is defined as in Equation (3). m is the number of added potentials.

To add the constraints, we first minimize Equation (15) without any constraint (set σ_i as the same value) and denote the regularized shape as $\mathbf{S}_0$. Then we calculate the changing ratio of the local energy of each point: $\Delta e_i = (e(s_i^0) - e(s_i^*))/e(s_i^*)$. It's obvious that when $\Delta e_i <= 0$, the points are moved to a better location and need not to be constrained, while when $\Delta e_i > 0$, the points may be moved to a worse location, and the larger Δe_i is, the priority to constrain this point is higher. So we query those points with $\Delta e_i > 0$ in the order of Δe_i from large to small, then test which point to be constrained following this order.

To guarantee that the added constraint will drag the whole shape to a better location, we accept the constraint only when the shape's energy $E(\mathbf{S}_t)$ becomes lower, where $E(\cdot)$ is defined as Equation (11). If one constraint point is accepted, we re-order the points based on the local energy changes between current $\mathbf{S}_t$ and the Markov Network result $\mathbf{S}^*$. The algorithm is summarized in Figure 5.

```
Given the shape $\mathbf{S}^*$ from Markov Network, the minimal energy $E_{min} = E(\mathbf{S}^*)$, the
optimal shape $\tilde{S} = \mathbf{S}^*$, the constraint marks of all points: $\{b_i = false\}$, the likelihood
variance $\Sigma_l = diag(1, 1, ..., 1)$, $v_k$ is the point index, $\{v_k\}$ is the points order.

1. Set $\Sigma_l$: $\sigma_i = 0.01$ if $b_i = true$, minimize Equation (15) to find an optimal shape $\mathbf{S}_t$
2. Evaluate the shape energy $E(\mathbf{S}_t)$.
   If $E(\mathbf{S}_t) < E_{min}$: set $\tilde{\mathbf{S}} = \mathbf{S}_t, E_{min} = E(\mathbf{S}_t)$
   else: set $b_{v_k} = false, b_{v_{k+1}} = true$, goto 1.
3. Calculate the local energy changes: $\{\Delta e_i\}$, update the points order $\{v_k\}$,
   Set $b_{v_0} = true$, goto 1.
4. Repeat the above process, until the allowed maximum tested points number has
   achieved
```

Fig. 5. Constrained Regularization Algorithm

Figure 6 (b) and (c) shows the procedure of constrained regularization. Our algorithm can generate reasonable shape while keeping the positions of those good points.

3.4 Guiding the Distribution of Markov Network States

After the regularization, if a better shape $\tilde{\mathbf{S}}^t$ is found, this suggests that some parts of the Markov Network result $\mathbf{Q}^t$ is really bad to make the shape deviates the global shape prior. To prevent the Markov Network from getting stuck into the same mistake in the next local search step, we generate a mask image that

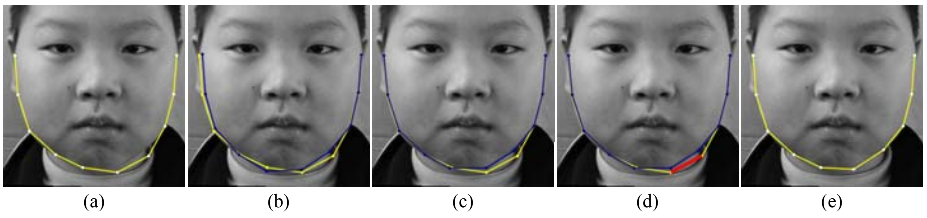

Fig. 6. The procedure of constrained regularization and guiding Markov Network states distribution. The yellow shape (white shape in printed version) is the result of Markov Network. The dark blue shape (black shape in printed version) is the result after regularization. (a) The result after one step Markov Network local search. (b) Regularization without constraints. (c) Regularization with constraints. (d) The states are forbidden to distribute in the red area (thick gray line in printed version). (e) After pruning the states in the blue area in (d), the result of Markov Network is much better.

points out the danger regions. Then we prune those states that mostly drop into the marked regions.

To mask the danger regions, we first calculate the distance of each line segment $\mathbf{q}_i^t$ to the better shape $\tilde{\mathbf{S}}^t$. Then if one line segment's distance is much larger than the mean distance, which shows that its location is really bad, we will draw it as a line with a certain width on the mask image, as shown in Figure 6 (d). Since after the constrained regularization, the shape $\tilde{\mathbf{S}}^t$ fits the good points quite well, detecting the bad line segments by this simple method is robust. By preventing the states from dropping into the detected wrong regions, the Markov Network will converge to the right position, see in Figure 6 (e).

4 Experiments

In this section, we apply the integrated shape model for face alignment and compare it with BTSM [6] and demonstrate that our algorithm improves accuracy greatly. The reason for comparing with BTSM is that it is an improvement of the classic ASM algorithm and extensive experiments have demonstrated its good performance in the context of face alignment.

To compare the accuracy of the two algorithms we use a set of child face images with size of 512×512 and divide the data set into 428 images for training and 350 images for testing. Each image contains a face with a size ranging from 270×270 to 330×330. A total of 87 feature points are manually labeled on each image for both the training and testing data sets. This data set contains the photos of children from 2 to 15 years old with different expressions. Thus the face shape variance is large. Consequently, although the images have good quality, the data set is still difficult for precise alignment.

In our hierarchical search scheme, a four-level Gaussian pyramid is built by repeated sub-sampling. For each image layer from coarse to fine, the corresponding face shape contains 28, 37, 57, 87 feature points respectively. For each test image, an initial shape is generated by randomly rotating (from $-20°$ to $20°$) and scaling (from 0.9 to 1.1) the mean shape of the training set, and it is fed into the two algorithms.

To quantitatively evaluate the accuracy of the algorithm, we calculate the estimation error by a curve difference measurement. Defining D_k as the distance of one point P_k of the searched shape to its corresponding ground true curve as explained in Figure 7(a), the estimation error is calculated as:

$$dist(A)_j = \sum_{k=1}^{N} D_k^A \tag{17}$$

where $dist(A)_j$ denotes the estimation error of algorithm A on the image j, and N is the number of feature points. For those closed sub-parts such as the eyes, brows and mouth, we break one closed contour into two open curves as shown in Figure 7(b). Comparing with the error measurement by summarizing the distance between searched point and annotated point, such a curve measurement

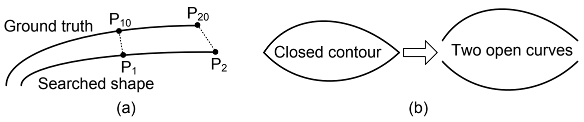

Fig. 7. Illustration for shape distance definition. (a) For the point P_1 of an open curve, D_1 is defined as the minimum distance $|P_1P_{10}|$. For the endpoint P_2, D_2 is defined as the distance between two endpoints $|P_2P_{20}|$. (b) The closed contour is broken into two open curves to calculate the error.

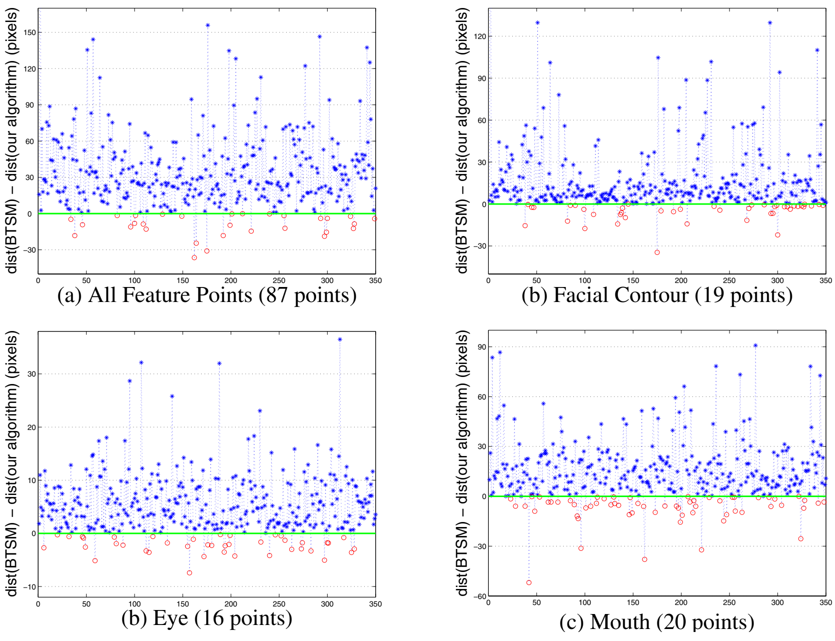

Fig. 8. Comparison of the accuracy of our algorithm and BTSM for the whole facial shape (a), facial contour (b), the eyes (c) and the mouth (d), respectively. The x-axis denotes the index j of test images and the y-axis denotes the difference of the estimation errors $dist(BTSM)_j - dist(our\ algorithm)_j$. Points above $y = 0$ (blue stars) denote images with better accuracy by our algorithm and points below $y = 0$ (red circles) are opposite.

is more reasonable for the comparison of alignment accuracy, because in many cases the fitted curves are almost the same although the positions of two sets of control points are different.

We have plotted $j \sim dist(BTSM)_j - dist(our\ algorithm)_j$ in Figure 8(a) for the whole face alignment. It is shown that on 320 of 350 (91.4%) images, the search results of our algorithm are better than that of BTSM. Since a human is more sensitive to the alignment accuracy for facial contour, eyes and mouth, we also compare the accuracy of these three parts respectively. For the facial contour, the eyes and the mouth, 308(88.0%), 309(88.3%), 289(82.6%) of 350

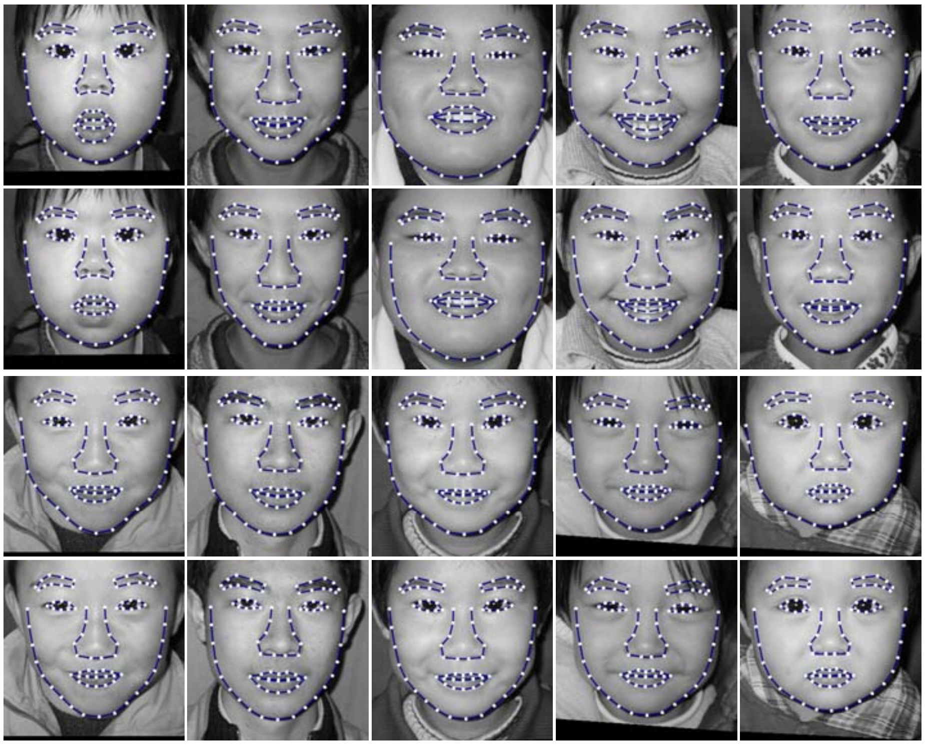
Fig. 9. Comparison of our algorithm and BTSM results. The first and third rows are our results, the second and fourth rows are BTSM results.

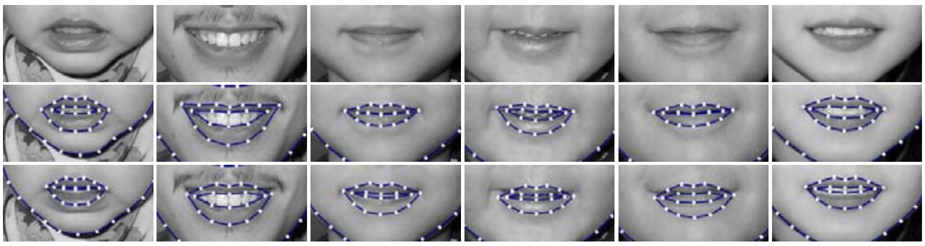
Fig. 10. Comparison of our algorithm and BTSM searching results for the mouth part. The first row is the mouth part cut from the test image, the second row is our results, and the third row is BTSM's results.

results of our algorithm are better than that of BTSM. As shown in Figure 8(b), 8(c) and 8(d), the improvement is distinct.

Figure 9 shows a set of searching results of our algorithm and BTSM. In the case that the facial contour or other facial sub-parts is largely variant from the average shape or there are wrinkles and shadings on the face, our algorithm can give more accurate results than BTSM. In many cases, BTSM can localize the

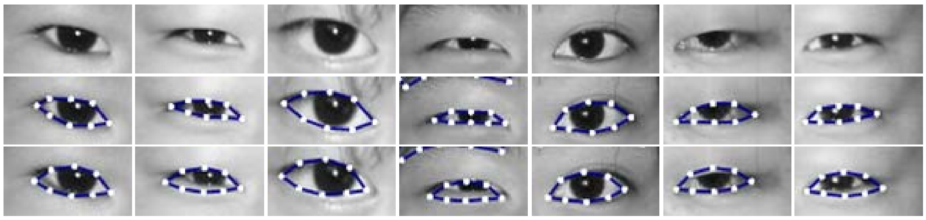

Fig. 11. Comparison of our algorithm and BTSM searching results for the eye part. First row is the eye part cut from the test image, the second row is our results, and the third row is BTSM's result.

whole face well, but when you look at each part closely, the results are often not accurate enough, as shown in Figure 10 and Figure 11, while our algorithm can get more accurate results.

5 Conclusion

In this paper, we present an integrated model for accurate shape alignment. The low level shape model based on a Markov Network serves to align the feature points to the image clues more accurately, while the global shape model based on PCA guarantees the shape to be globally reasonable. Constrained regularization and the mask image for guiding the distribution of Markov Network states make the algorithm more effective and robust. We have compared our algorithm with BTSM and demonstrated its greatly improved accuracy.

References

1. Thodberg, H.H., Rosholm, A.: Application of the active shape model in a commercial medical device for bone densitometry. In: BMVC, Manchester, UK (2001)
2. Zhang, T., Freedman, D.: Tracking objects using density matching and shape priors. In: ICCV, Nice, France (2003)
3. Hu, Y., Jiang, D., Yan, S., Zhang, L., Zhang, H.J.: Automatic 3d reconstruction for face recognition. In: FGR, Seoul, Korea (2004)
4. Liu, C., Zhu, S.C., Shum, H.Y.: Learning inhomogeneous gibbs model of faces by minimax entropy. In: IEEE Int'l Conf. on Computer Vision, Vancouver, Canada (2001)
5. Cootes, T.F., Taylor, C.J., Graham, J.: Active shape models – their training and application. Computer Vision and Image Understanding **61** (1995) 38–59
6. Zhou, Y., Gu, L., Zhang, H.J.: Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference. In: IEEE Conf. on Computer Vision and Pattern Recognition, Madison, WI (2003)
7. Liu, C., Shum, H.Y., Zhang, C.: Hierarchical shape modeling for automatic face localization. In: Enropean Conf. on Computer Vision. (2002) 687–703
8. Cootes, T.F., Taylor, C.: A mixture model for representing shape variation. Image and Vision Computing **17**(8) (1999) 567–574

9. Romdhani, S., Cong, S., Psarrou, A.: A multi-view non-linear active shape model using kernel pca. In: 10th British Machine Vision Conference, Nottingham, UK (1999)
10. Mitchell, S.C., Lelieveldt, B.P.F., van der Geest, R., Bosch, H.G., Reiber, J.H.C., Sonka, M.: Multistage hybrid active appearance model matching: Segmentation of left and right ventricles in cardiac mr images. IEEE Transactions on Medical Imaging **20**(5) (2001) 415–423
11. Coughlan, J., Ferreira, S.: Finding deformable shapes using loopy belief propagation. In: The Seventh European Conference on Computer Vision, Copenhagen, Denmark (2002)
12. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, San Mateo, California (1988)
13. Freeman, W., Pasztor, E., Carmichael, O.: Learning low-level vision. Int. J. Computer Vision **40**(1) (2000) 25–47