Hierarchical Shape Modeling for Automatic Face Localization

Ce Liu[1], Heung-Yeung Shum[1], and Changshui Zhang[2]

[1] Visual Computing Group, Microsoft Research Asia, Beijing 100080, China
[2] Department of Automation, Tsinghua University, Beijing 100084, China
lce@msrchina.research.microsoft.com

Abstract. Many approaches have been proposed to locate faces in an image. There are, however, two problems in previous facial shape models using feature points. First, the dimension of the solution space is too big since a large number of key points are needed to model a face. Second, the local features associated with the key points are assumed to be independent. Therefore, previous approaches require good initialization (which is often done manually), and may generate inaccurate localization. To automatically locate faces, we propose a novel hierarchical shape model (HSM) or multi-resolution shape models corresponding to a Gaussian pyramid of the face image. The coarsest shape model can be quickly located in the lowest resolution image. The located coarse model is then used to guide the search for a finer face model in the higher resolution image. Moreover, we devise a Global and Local (GL) distribution to learn the likelihood of the joint distribution of facial features. A novel hierarchical data-driven Markov chain Monte Carlo (HDDMCMC) approach is proposed to achieve the global optimum of face localization. Experimental results demonstrate that our algorithm produces accurate localization results quickly, bypassing the need for good initialization.

1 Introduction

Face detection and face localization have been challenging problems in computer vision and machine perception. *Face detection*, for example, explores possible locations of faces from an input image, and *face localization* accurately locates the facial shape and parts, often from an initialized model. *Appearance models* have been successfully used for face detection, where typically a square region with an elliptic mask is used to represent a face image. Based on a large amount of positive (face) and negative (non-face) samples, machine learning techniques such as PCA [13], neural networks [9,11], support vector machines [6], wavelets [10] and decision trees [14], are always used to learn the separating manifold of faces and non-faces. By verifying patterns in a shifting window, the position of a face can be derived.

However, an appearance model alone is not flexible enough to model shape deformations and pose or orientation variations. Shape models, in particular deformable shape models such as deformable template matching [15] and graph

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2351, pp. 687–703, 2002.

matching [4], have been used for face localization, *i.e.*, finding accurate facial shape and parts. A good example is the active shape model (ASM) [1] where a Bayesian approach using a mixture of Gaussians is adopted. The prior of shape and the likelihood of local features given each point are separately learnt, and Bayesian inference is chosen to obtain the maximum a *posteriori* (MAP) solution. They also developed and improved active appearance models (AAM) [2] to locate faces.

There are, however, two problems with previous shape models for locating faces. First, face localization is not automatic because of the huge solution space of shape and position. Typically we have to use a large number of feature points (*e.g.*, around 80 in [1]) to represent faces. Good initialization must be provided so that the optimization with MAP would converge to the global optimum. Often manual initialization is required. Second, even if a good initialization is provided, the localization results may not be accurate because the likelihood for local features was not modeled properly in previous approaches. For instance, distributions for features are assumed to be independent in the ASM model.

To address these two problems, we propose a *hierarchical shape model* (HSM) for automatic face localization. Multiple levels of shapes (with feature points) from coarse to fine, are employed to represent faces in a face image pyramid from low-resolution to high-resolution. First, the coarsest shape model is located in the lowest resolution image. Then we can gradually infer a finer shape in a higher-resolution image from the located coarse shape in lower-resolution image. Therefore, the uncertainty of the solution space is significantly reduced. Our system can automatically find the face shape and location robustly and quickly.

In HSM, we model two types of priors: single-level distribution and conditional distribution of a lower level given its higher level. Both of them are modeled and learnt by a mixture of Gaussians. A key idea in HSM is the likelihood modeling. The local image patterns associated with the feature points are NOT assumed independent, but conditionally independent with respect to a hidden variable. Specifically, we propose a novel *global and local* (GL) distribution to model the joint distribution, also with a mixture of Gaussians. In addition, we need to learn the data driven proposal density to guess the location of face based on local image evidence.

To pursue global convergence of the solution, we employ a hierarchical data driven Markov chain Monte Carlo (HDDMCMC) [12] method to explore the solution space effectively. It is not only globally optimal compared with traditional gradient descent methods, but also efficient compared with common Monte Carlo methods. All the distributions in HSM are modeled with Gaussian mixtures, which can be reliably learnt by a reversible jump Markov chain Monte Carlo method [8,3].

This paper is organized as follows. Section 2 introduces the framework of HSM, including its formulation, Bayesian inference and the main concept of HDDMCMC. The sampling details of HDDMCMC are introduced in Section 3. Section 4 talks about how to model four types of distributions in HSM and

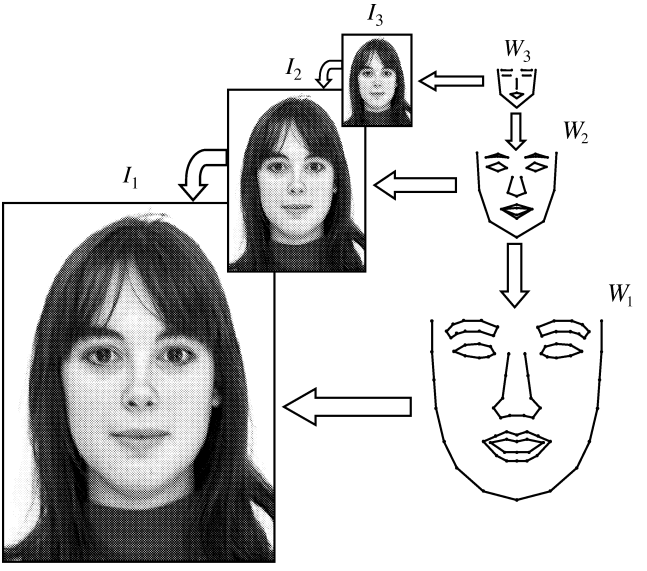

Fig. 1. Illustration of hierarchical shape model with three levels. Left is a Gaussian pyramid of a face image. Right is the hierarchical shapes explaining the image in corresponding levels.

briefly introduces the learning strategy. Experiments are shown in Section 5. Section 6 summarizes this article.

2 Hierarchical Shape Model

2.1 Hierarchical Modeling for Facial Shape

Feature points used in face shape models may have semantic meanings. For example, we usually choose corner and edge points of eyes, eyebrows, nose, mouth and face contour to model a face. Let $W = \{(x_i, y_i), i = 1, \cdots, n\}$ denote the shape, where (x_i, y_i) is the ith key point and n is the number of key points. Let I denote the image containing the face. The task of face localization is to infer W from I.

A *hierarchical shape model* (HSM) has multiple levels, $\mathcal{W} = \{W_l, l = 1, \cdots, L\}$, where W_1 is the finest level of shape. The number of feature points in W_l is n_l and the jth feature point of W_l is denoted as $W_l^{(j)}$. Each feature point in coarse levels $(W_2, \cdots, W_L)$ is generated as the weighted sum of chosen feature points in W_1. In practice, we choose n_{l+1} to be approximately half of n_l. A three-level HSM is shown in Fig 1. Let the Gaussian pyramid of image I be $\{I_1, \cdots, I_L\}$. Then the correspondence is established between shape domain W_l and image domain I_l. The most important property of HSM is that significant semantic information is preserved across levels. As shown in Fig. 1, eyes, mouth, face contour are all modeled even in the coarsest level.

2.2 Bayesian Inference in Hierarchical Shape Model

Our task is to infer $\mathcal{W} = \{W_1, \cdots, W_L\}$ from image I_1:

$$\begin{aligned}\mathcal{W}^* &= \arg\max_{\mathcal{W}} p(\mathcal{W}|I_1) = \arg\max_{\mathcal{W}} p(W_1, \cdots, W_L|I_1) \\ &= \arg\max_{\mathcal{W}} p(W_L|I_1) \prod_{l=1}^{L-1} p(W_l|W_{l+1}, \cdots, W_L, I_1) \\ &= \arg\max_{\mathcal{W}} p(W_L|I_L) \prod_{l=1}^{L-1} p(W_l|W_{l+1}, I_l). \end{aligned} \tag{1}$$

$p(W_L|I_1) = p(W_L|I_L)$ because the information of I_L is enough to determine W_L, and so on to get $p(W_l|W_{l+1}, I_l)$. Obviously given I_l and W_{l+1}, W_l only depends on I_l. We may have

$$\mathcal{W}^* = \arg\max_{\mathcal{W}} \prod_{l=1}^{L} p(W_l|I_l) = \arg\max_{\mathcal{W}} \prod_{l=1}^{L} p(I_l|W_l)p(W_l), \tag{2}$$

which is equivalent to

$$W_l^* = \arg\max_{W_l} p(I_l|W_l)p(W_l), l = L, \cdots, 1. \tag{3}$$

In HSM, we shall gradually optimize W_l^* in Eqn.(3).

We decompose shape model W_l into two parts: the *external* parameters including centroid Z_l, scale s_l and orientation θ_l, and the *internal* parameters w_l. With a linear transition matrix $T_{(s,\theta)}$ to scale and rotate the shape w_l, we get

$$W_l = T_{(s,\theta)} w_l + Z_l, \; T_{(s,\theta)} = s \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{4}$$

It is reasonable to assume that the external and internal parameters are independent

$$p(W_l) = p(w_l)p(Z_l)p(s_l)p(\theta_l). \tag{5}$$

2.3 Hierarchical Data-Driven Markov-Chain Monte Carlo

The Markov chain Monte Carlo (MCMC) method is a tool to sample high-dimensional distributions. It can be used in optimization if the objective function itself is a pdf or could be converted to a pdf. Particularly if the objective function is very complex with multiple peaks, MCMC has the good property of global convergence because it ensures the Markov chain to reach the global optimum with a certain probability. The inefficiency of MCMC could be improved by data driven MCMC (DDMCMC)[12]. The traditional MCMC method randomly walks through the parameter space while DDMCMC employs some heuristics from data to guide the walks. In HSM, we should devise the salient proposal density including both the heuristics given by the localization result on the higher level, and local cues directly from the image. This leads to a hierarchical

DDMCMC or HDDMCMC, which starts at the top level and propagates the optimal solution from higher level to lower level.

At the top level, the optimal W_L of I_L is determined by Metropolis-Hastings sampling. The Markov chain $\{W_L(t)\}$ to sample $p(W_L|I_L)$ is driven by the transition probability at time t

$$\alpha = \min\{1, \frac{p(W'_L|I_L)q(W_L(t); W'_L, I_L)}{p(W_L(t)|I_L)q(W'_L; W_L(t), I_L)}\} \tag{6}$$

where W'_L is sampled from proposal density $q(W'_L; W_L(t), I_L)$ and it is accepted as $W_L(t+1)$ with probability α. The proposal density has two components, the shape prior $p(W_L)$ and *data-driven* part, or local hints from image I_L to the jth feature point $q(W_L^{(j)}; W_L^{(j)}(t), I_L)$. The optimal W_L^* is selected from the samples $\{W_L(t)\}$ with maximum a posterior (MAP) $p(W_L^*|I_L)$.

The next is to find the optimal W_l^* from the higher level W_{l+1}^*. The sampling strategy is slightly different from Eqn.(6) because the localization W_{l+1}^* will guide the Markov chain in proposal density by

$$\alpha = \min\{1, \frac{p(W'_l|I_l)q(W_l(t); W'_l, I_l, W^*_{l+1})}{p(W_l(t)|I_l)q(W'_l; W_l(t), I_l, W^*_{l+1})}\} \tag{7}$$

where the proposal density $q(W'_l; W_l(t), I_l, W^*_{l+1})$ relies on W^*_{l+1} as well. The proposal density again includes two parts, shape prior propagation $p(W_l|W^*_{l+1})$ and local hints $q(W_l'^{(j)}; W_l^{(j)}(t), I_l)$ from I_l.

The hierarchical DDMCMC method (Eqn.(6) and (7)) is globally optimal since we shall finally sample the finest posterior $p(W_1|I_1)$. The information of higher level shape W^*_{l+1} propagates to lower level W_l via proposal density, and guides the Markov chain exploring the solution space $p(W_l|I_l)$. Thus, the error of W^*_{l+1} does NOT propagate to W_l. Since the entropy of $p(W_l|W^*_{l+1}, I_l)$ is much smaller than $p(W_l|I_l)$, the Markov chain will hardly walk to those unlikely samples. Therefore, HDDMCMC is much more efficient than conventional DDMCMC that directly samples W_1 from input image I_1.

3 Sampling Hierarchical DDMCMC

In this section, we explain the details of Metropolis-Hastings sampling Eqn.(6) and (7). Since the propagation from higher level to lower level is more general than sampling the top level, we focus on Eqn.(7). We shall also discuss how to give good initializations in the top level.

3.1 Sampling Strategy

The basic task of MCMC is to sample a target density $p(x)$, but in most cases it is very difficult to directly sample $p(x)$. Therefore a proposal density $q(x'; x(t))$

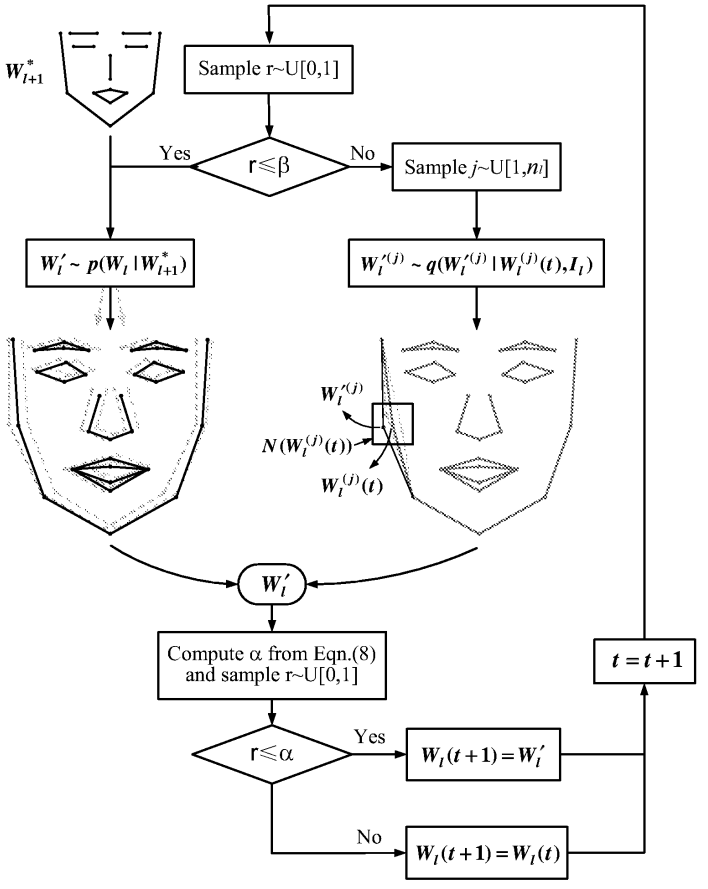

Fig. 2. The flowchart of hierarchical DDMCMC in HSM. We use the propagation form W_3 to W_2 as an example. The left branch is sampling the conditional prior from higher level result, while the right branch is sampling the position of each feature point.

is designed so that it is easy to draw samples $x' \sim q(x'; x(t))$. In our HSM, the proposal density $q(W'_l; W_l(t), I_l, W^*_{l+1})$ could be decomposed to

$$q(W'_l; W_l(t), I_l, W^*_{l+1}) = \beta p(W'_l|W^*_{l+1}) + (1-\beta)q(W'_l; W_l(t), I_l), \tag{8}$$

where β is the probability of choosing *prior propagation* process $p(W'_l|W^*_{l+1})$ and $1-\beta$ is the probability of choosing *data-driven* process $q(W'_l; W_l(t), I_l)$, or sampling the feature points directly from the image.

Sampling the first part of Eqn.(8), $p(W_l|W^*_{l+1})$, is high-dimensional and non-trivial. But if we model the joint distribution $p(W_l, W_{l+1})$ by a mixture of Gaussians, then the conditional density $p(W_l|W_{l+1})$ is also a mixture of Gaussians that can be derived from $p(W_l, W_{l+1})$. At the top level without any prior propagation, we simply sample $p(W_L)$ which is also modeled as a mixture of Gaussians.

To make it plausible to sample the second part of Eqn.(8), we design an individual proposal $q(W_l'; W_l(t), I_l)$ for each feature point $W_l'^{(j)}$, $j = 1, \cdots, n_l$:

$$q(W_l'; W_l(t), I_l) = \prod_{j=1}^{n_l} q(W_l'^{(j)}; W_l^{(j)}(t), I_l). \tag{9}$$

We may use Gibbs sampling to simply flip the position of one feature point at each time. Suppose the jth feature point is chosen, then after sampling, W_l' differs from $W_l(t)$ only at $W_l'^{(j)}$. Sampling $q(W_l'^{(j)}|W_l^{(j)}, I_l)$ means that we should find a better position for the jth feature point, merely considering the local likelihood. Let $\Gamma_{(x,y)} \subset I_l$ denote a 5×5 image patch centered at (x, y) and $N(W_l^{(j)})$ be a neighborhood, *e.g.* a 7×7 region centered at $W_l^{(j)}$. We merely take into account the possible positions of $W_l'^{(j)}$ in the neighbor $N(W_l^{(j)})$

$$q(W_l'^{(j)}; W_l^{(j)}(t), I_l) = \frac{P(W_l'^{(j)} = (x, y)|\Gamma_{(x,y)})}{\sum_{(x,y)\in N(W_l^{(j)}(t))} P(W_l'^{(j)} = (x, y)|\Gamma_{(x,y)})}, \tag{10}$$

where $P(W_l'^{(j)} = (x, y)|\Gamma_{(x,y)})$ is the probability of the jth feature point lying at position (x, y) given the local image pattern $\Gamma_{(x,y)}$. Thus it is easy to draw a new sample $W_l'^{(j)}$ via Eqn.(10). We define a *salient map* $p(W_l^{(j)}|I_l)$ as

$$p(W_l^{(j)}|I_l) = \frac{P(W_l^{(j)} = (x, y)|\Gamma_{(x,y)})}{\sum_{(x,y)\in I_l} P(W_l^{(j)} = (x, y)|\Gamma_{(x,y)})}, \tag{11}$$

to denote the distribution of the jth feature point at each position of image I_l according to local likelihood only. Eqn.(10) may be rewritten as

$$q(W_l'^{(j)}; W_l^{(j)}(t), I_l) = \frac{p(W_l'^{(j)}|I_l)}{\sum_{W_l'^{(j)}\in N(W_l^{(j)}(t))} p(W_l'^{(j)}|I_l)}. \tag{12}$$

Before the sampling process, we pre-compute the salient maps for all feature points such that it is very fast to draw proposals.

3.2 Initialization by Generalized Hough Transform

Although HDDMCMC is insensitive to initializations, good initializations always help searching algorithms both in efficiency and accuracy. In HSM, the initialization is given in the top level to initialize $W_L(0)$ in I_L. Since the dimension of W_L is fairly high, we first give an estimate of the global parameters $\{Z_L(0), s_L(0), \theta_L(0)\}$, and then estimate the position of each key point.

Suppose the lattice of image I_L is Ψ. The proposal density of Z_L associated with s_L and θ_L is

$$q(Z_L, s_L, \theta_L | I_L) = \sum_{j=1}^{n_L} \sum_{W_L^{(j)} \in \Psi} p(Z_L, s_L, \theta_L, W_L^{(j)} | I_L)$$
$$= \sum_{j=1}^{n_L} \sum_{W_L^{(j)} \in \Psi} p(Z_L, s_L, \theta_L | W_L^{(j)}) p(W_L^{(j)} | I_L), \quad (13)$$

where the salient map $p(W_L^{(j)}|I_L)$ generates a hypothesis of the positions of each feature point, and then the feature point would propagate the hypothesis to the 4D global parameter space by $p(Z_L, s_L, \theta_L|W_L^{(j)})$. This is in fact a *generalized Hough transform* (GHT). The initialization of the outer parameters $\{Z_L(0), s_L(0), \theta_L(0)\}$ is sampled from Eqn.(13), and the key points most likely to be connected to $Z_L(0)$ via $s_L(0)$ and $\theta_L(0)$ are chosen to initialize $W_L(0)$.

4 Distribution Modeling and Learning

In the previous section we introduced the statistical framework of HSM and its four distributions. Overall, in HDDMCMC, there are basically two densities, *i.e.* conditional prior $p(W_l|W_{l+1})$ and salient map $p(W_l^{(j)}|I_l)$ for us to draw proposals, and other two densities, prior $p(W_l)$ and likelihood $p(I_l|W_l)$ to evaluate the posterior. In this section we design different strategies to model them. We show that all of them can be decomposed to a mixture of Gaussians model, which could be reliably learnt by reversible jump Markov chain Monte Carlo method.

4.1 Prior $p(W_l)$

From prior decomposition Eqn.(4) and independence assumption Eqn.(5), we model the distribution of *external* and *internal* parameters separately. The prior distribution of the position Z_l is uniform and omitted. The priors of scale s_l and orientation θ_l are modeled by Gaussians

$$p(s_l) = \frac{1}{\sqrt{2\pi}\sigma_{s_l}} \exp\{-\frac{(s-\mu_{s_l})^2}{2\sigma_{s_l}^2}\}, \; p(\theta_l) = \frac{1}{\sqrt{2\pi}\sigma_{\theta_l}} \exp\{-\frac{(\theta-\mu_{\theta_l})^2}{2\sigma_{\theta_l}^2}\}, \quad (14)$$

where μ_{s_l}, μ_{θ_l} and σ_{s_l}, σ_{θ_l} are the means and variances of s_l and θ_l respectively. They can be easily estimated from training samples.

To model the position, scale and orientation irrelevant shape $p(w_l)$ is non-trivial due to its high dimensionality $2n_l$. Here we apply principal components analysis (PCA) to reduce the dimension and obtain the principal components h_l ($dim(h_l) < dim(w_l)$)

$$h_l = B_{w_l}^T (w_l - \mu_{w_l}), \; W_l = B_{w_l} h_l + \mu_{w_l}, \quad (15)$$

where μ_{w_l} is the mean of $p(w_l)$, and each column vector of B_{w_l} is the eigenvector of the covariance matrix of $p(w_l)$. Since h_l can approximate w_l very well with much lower dimension, we may learn $p(h_l)$ rather than $p(w_l)$. We model $p(h_l)$ with the Gaussian mixture

$$p(h_l) = \sum_{i=1}^{K_l} \alpha_l^{(i)} G(h_l; \mu_l^{(i)}, \Sigma_l^{(i)}), \tag{16}$$

where $G(h_l; \mu_l^{(i)}, \Sigma_l^{(i)})$ is a Gaussian distribution with mean $\mu_l^{(i)}$ and covariance $\Sigma_l^{(i)}$. $\alpha_l^{(i)}$ is the corresponding weight such that $\sum_{i=1}^{K_l} \alpha_i = 1$ and $\alpha_l^{(i)} > 0, \forall\, i$. K_l is the number of Gaussian kernels.

4.2 Conditional Prior $p(W_l|W_{l+1})$

The conditional density $p(W_l|W_{l+1})$ plays an essential role in HSM and HDDM-CMC because the localization of higher level W_{l+1}^* will propagate down via it. Similar to Eqn.(5) we may have

$$\begin{aligned} p(W_l|W_{l+1}) &= p(w_l, Z_l, s_l, \theta_l | w_{l+1}, Z_{l+1}, s_{l+1}, \theta_{l+1}) \\ &= p(w_l|w_{l+1}) p(Z_l|Z_{l+1}) p(s_l|s_{l+1}) p(\theta_l|\theta_{l+1}). \end{aligned} \tag{17}$$

The conditional distributions $p(Z_l|Z_{l+1}), p(s_l|s_{l+1}), p(\theta_l|\theta_{l+1})$ are all modeled as 1D or 2D Gaussians, *e.g.*, $p(s_l|s_{l+1}) \propto \exp\{-(s_l - s_{l+1})^2/\lambda_{s_l}\}$ where λ_{s_l} scales the variance of s_l.

We, however, take a two-step approach to modeling $p(w_l|w_{l+1})$. We first learn the joint distribution $p(w_l, w_{l+1})$ with a Gaussian mixture model. Then the conditional prior $p(w_l|w_{l+1})$ has a closed form distribution, directly computed by $p(w_l, w_{l+1})$ with parameters controlled by w_{l+1}. We again use PCA to reduce dimensions and in fact model $p(w_l|w_{l+1})$ by $p(h_l|h_{l+1})$.

4.3 Likelihood $p(I_l|W_l)$

To evaluate the likelihood of an image given the shape in HSM, we only need to take into account the pixels nearby each feature point. Let $\mathit{\Gamma}_{W_l^{(j)}} \subset I_l$ denote a 5×5 square patch around the jth feature point $W_l^{(j)}$. Then we have

$$p(I_l|W_l) = p(\mathit{\Gamma}_{W_l^{(1)}}, \cdots, \mathit{\Gamma}_{W_l^{(n_l)}}). \tag{18}$$

Directly modeling the above joint distribution is difficult. This is why previous shape models (*e.g.*, [1]) assumed independent distributions, *i.e.*

$$p(\mathit{\Gamma}_{W_l^{(1)}}, \cdots, \mathit{\Gamma}_{W_l^{(n_l)}}) = \prod_{j=1}^{n_l} p(\mathit{\Gamma}_{W_l^{(j)}}). \tag{19}$$

However, this assumption is an oversimplification for the likelihood. For example, what the left corner of the left eye looks like definitely depends on the appearance of the left corner of the left eyebrow.

We now introduce *global and local* (GL) distributions to model likelihood.

Definition 1. *The joint distribution of random variable $X = \{x_1, \cdots, x_n\}$ is called a GL distribution if*

$$p(X) = \int p(v) p(x_1, \cdots, x_n|v) dv = \int p(v) \prod_{i=1}^{n} p(x_i|v) dv, \tag{20}$$

where $v = f(X)$ is the hidden variable of X.

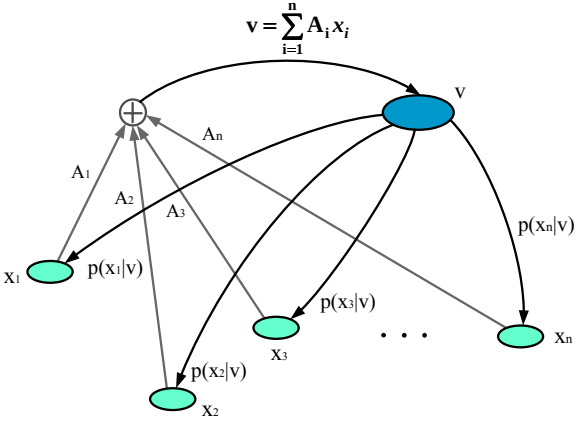

Fig. 3. Illustration of a GL distribution when the hidden variable is chosen as the principal components. The hidden variable v is determined by $\{x_i\}$, but it also controls each x_i. $\{x_i\}$s are never independent because x_j would affect x_i via v.

An intuitive explanation of the GL distribution is that each random variable $x_1, \cdots, x_n$ is conditionally independent with respect to the hidden variable v, and its distribution $p(v)$ captures the global properties of X. Therefore each random variable is not independent because they are connected by the hidden variable, and meanwhile not too correlated because the conditional densities $p(x_i|v), i = 1, \cdots, n$ may be different. What we should do for GL is to select hidden variable $v = f(X)$ and do the integration.

Theorem 1. *Let $v = AX$, where A is the principal components of X and $dim(v) \ll dim(X)$. Assume $p(v)$ and $p(x_i|v)$ $(i = 1, \cdots, n)$ to be continuous functions with finite optimums. The GL distribution can be approximated by*

$$p(X) \approx \lambda p(AX) \prod_{i=1}^{n} p(x_i|AX), \tag{21}$$

where λ is a constant.

Proof. Since matrix A is the principal components of X, for a particular X and a small ε there exists a small neighborhood $N_v(X) = \{p(X|v) > \varepsilon\}$. Since the integration of $p(v) \prod_{i=1}^{n} p(x_i|v)$ in the whole set is $p(X) < \infty$, the integration can be approximated in $N_v(X)$. The volume of the neighborhood $N_v(X)$ exists and is assumed to be δ due to the condition of $p(v)$ and $p(x_i|v)$ $(i = 1, \cdots, n)$. According to the mid-value theorem, there exists $\xi \in N_v(X)$ such that

$$\begin{aligned} p(X) &= \int p(v) \prod_{i=1}^{n} p(x_i|v) dv \\ &\approx \int_{N_v(X)} p(v) \prod_{i=1}^{n} p(x_i|v) dv \\ &= \delta p(\xi) \prod_{i=1}^{n} p(x_i|\xi). \end{aligned} \tag{22}$$

The point $v = AX$ must lie at the center of $N_v(x)$ because $X \approx A^T v$ and the conditional density $p(X|AX)$ is fairly high. Since both ξ and AX lie in the

very small neighbor $N_v(X)$, we may also have $\xi \approx AX$. This naturally leads to Eqn.(21). □

Theorem 1 gives us an approximation to evaluate the GL distribution by PCA. The hidden variable lies in the eigenspace of the observed data which captures the global correspondence as illustrated in Fig. 3. From another point of view, the distribution of the hidden variable $p(v)$ in eigenspace approximates the observed one, and the approximation error is compensated by the local densities $p(x_i|v)$.

When applying a GL distribution to modeling likelihood Eqn.(18), the dimension of $\Gamma_{W_l^{(j)}}$ is 25, still too high. We again employ PCA to reduce the dimension of $\Gamma_{W_l^{(j)}}$ to $u_l^{(j)}$. And v_l is the hidden variable or principal components of $\{u_l^{(1)}, \cdots, u_l^{(n_l)}\}$. Thus the likelihood is approximated by

$$p(I_l|W_l) \approx p(v_l) \prod_{j=1}^{n_l} p(u_l^{(j)}|v_l) = p^{-(n_l-1)}(v_l) \prod_{j=1}^{n_l} p(u_l^{(j)}, v_l). \tag{23}$$

Both $p(v_l)$ and $p(u_l^{(j)}, v_l)$ are assumed mixture of Gaussians.

4.4 Salient Map $p(W_l^{(j)}|I_l)$

From the definition of salient map Eqn.(11), the probability $P(W_l^{(j)} = (x, y)|\Gamma_{(x,y)})$ is essential. Based on Bayesian law we may have

$$P(W_l^{(j)}=(x, y)|\Gamma_{(x,y)}) \propto p(\Gamma_{W_l^{(j)}})p(W_l^{(j)}). \tag{24}$$

$p(W_l^{(j)})$ is the prior distribution of the jth feature point, *e.g.*, the left eye would not lie at the upper-right of the image. $p(\Gamma_{W_l^{(j)}})$ is just the independent component of Eqn.(19). We also apply PCA to reduce $\Gamma_{W_l^{(j)}}$ and learn a Gaussian mixture model in the reduced space.

4.5 Learning Gaussian Mixture by Reversible Jump MCMC

We have so far modeled all the distributions in HSM as a mixture of Gaussians because of its flexibility in fitting arbitrary distributions. A traditional algorithm of learning Gaussian mixture model is Expectation-Maximization (EM), which needs as input the kernel number and often gets stuck in local minimums. To solve this problem, we formulate the objective function under a MAP criterion instead of MLE, with prior that restricts the number of Gaussian kernels based on the *minimum description length* (MDL) criterion. Let $\{Y_1, \cdots, Y_m\}$ be observed examples. The number of Gaussian kernels is k, and the parameter of the ith kernel is α_i, μ_i and Σ_i. Let $\theta_k = \{\alpha_i,\ \mu_i,\ \Sigma_i\}_1^k$. The Gaussian mixture model is learnt via

$$(k^*, \theta_k^*) = \arg\max_{k,\theta_k} p(k) \prod_{j=1}^{m} p(Y_j; \theta_k) \tag{25}$$

where $p(k) \propto \exp\{-\lambda k \log k\}$ is the prior of the kernel number, and

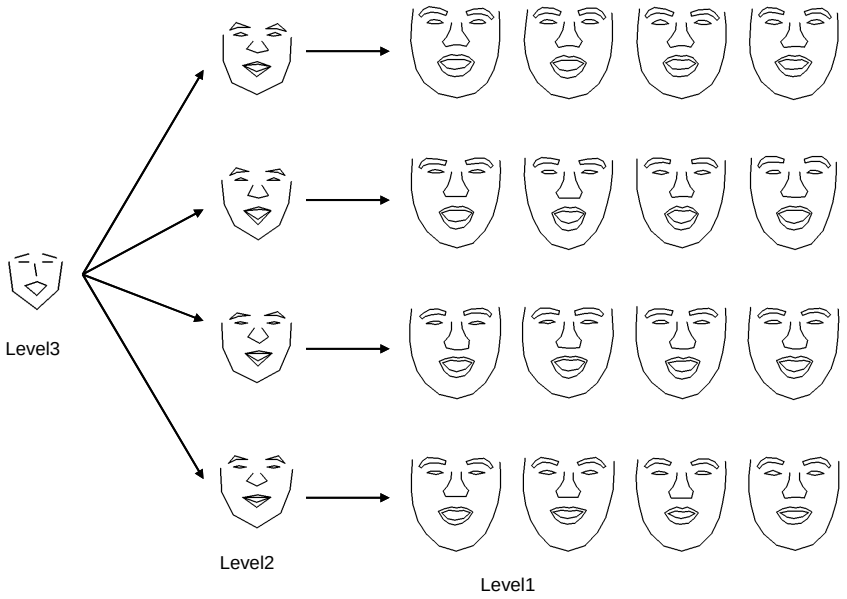

Fig. 4. A sampling tree of face prior shape from the top to the bottom level. For each parent node w_{l+1}, four child nodes are randomly sampled from $p(w_l|w_{l+1})$.

$$p(Y_i;\theta_k) = \sum_{j=1}^{k} \alpha_i G(Y_i;\mu_i,\Sigma_i).$$

Then a reversible jump Markov chain Monte Carlo is developed to explore varying probability spaces, with the guarantee of global convergence [8,3]. There are three processes in the reversible jump MCMC: *diffusion* to explore the same space, *split* to divide one Gaussian kernel to two, and *merge* to combine two kernels to one. So the sampler may randomly walk to samples with different kernel numbers. The learning by reversible jump MCMC is robust, efficient and reliable.

5 Experimental Results

Our experiments are conducted with a large number of frontal face images in the FERET data set [7], AR data set [5] and other collections, with different races and varying illuminations. We have selected 721 images as training data and use others for testing. We also collected some face images with complex backgrounds and lightings to test the robustness of our algorithm. Each training image is normalized to the same scale and manually labelled with 83 key points, including the most semantically important feature points such as the corners of eyes, mouth and face contour. These samples form the training set of shape $\{W_1(i), i = 1, \cdots, N\}$. Then we design $\{W_2(i), i = 1, \cdots, N\}$ and $\{W_3(i), i = 1, \cdots, N\}$ with 34 and 19 feature points, respectively.

Once the three levels of shape samples and their corresponding Gaussian pyramid are generated, we employ reversible jump Markov chain Monte Carlo to learn the four elementary distributions, *i.e.*, single level prior $p(W_l)$, conditional prior $p(W_l|W_{l+1})$, likelihood $p(I_l|W_l)$ and salient map $p(W_l^{(j)}|I_l)$. To justify the reliance of our learning algorithm, we build a sampling tree of hierarchical facial prior shape. The root node of this tree is the coarsest shape w_3 sampled from $p(w_3)$. Then for each parent node in the tree, *e.g.* $w_{l+1}(j)$, we may get four child nodes $\{w_{lj}(1), w_{lj}(2), w_{lj}(3), w_{lj}(4)\}$ randomly sampled from $p(w_l|w_{l+1}(j))$, as shown in Fig 4. The samples generated in this hierarchical shape tree demonstrate the reliability of both the conditional density modeling via Gaussian mixture and learning by reversible jump MCMC. For example, we observe that the root sample w_3 seems to have a smile, and so do four child samples at level 2 and sixteen samples at level 1. Obviously the magnitude of a smile differs from level to level. Going from level 2 to level 1, we also observe that the difference between the four children at level 1 generated by the same parent node at level 2 is much smaller than that between those four nodes on layer

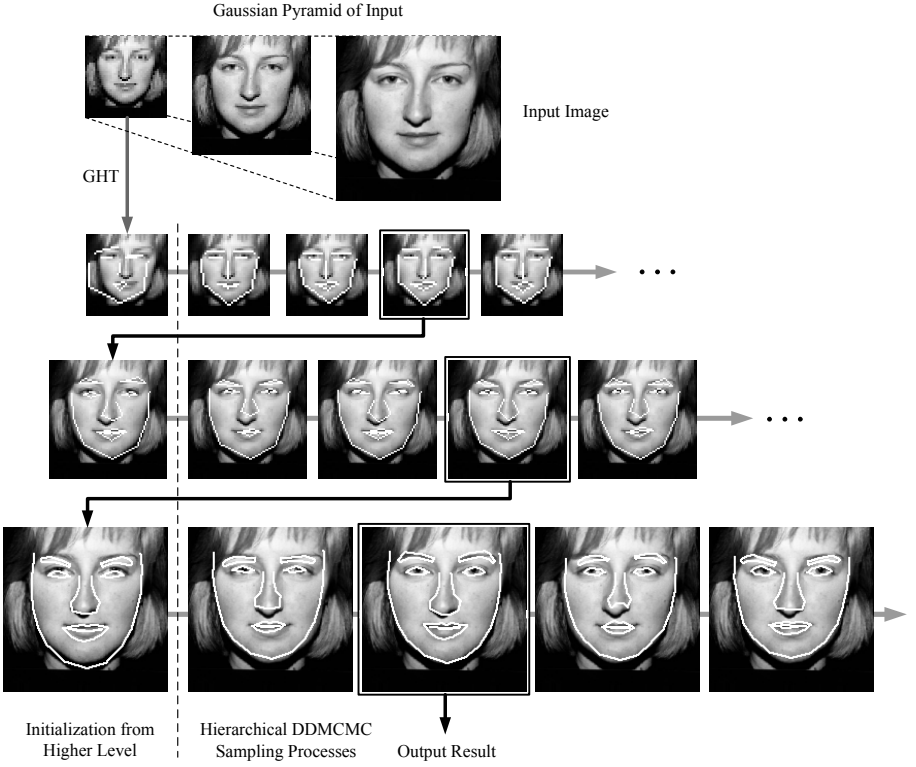

Fig. 5. The flowchart of gradually locating a face from low-resolution to high-resolution in HSM. In this display, the pyramid does not go up by 2 (the size of the image at higher level is more than a half of that at lower level), but in experiment it does.

Fig. 6. Comparison between likelihood assumptions. (a) Input face image. (b) Localization result with independence assumption to local features. (c) Localization result with GL distribution for local features.

2, reflecting the fact that the finer level model represents the higher frequency information.

Once all the distributions are learnt (there are 280 distributions to learn), we employ HDDMCMC to locate a face in input image. We use a simple example to illustrate our algorithm in Fig. 5. First we build the Gaussian pyramid of the input image, and do the generalized Hough transform (GHT) to get the initialization in the top level. Then the sampler of HDDMCMC draws random samples from the posterior, and the optimal solution is achieved in the samples by MAP criterion. By sampling the conditional face prior densities as shown in Fig. 4, the optimal solution from higher level generates the initialization at the lower level. This process is propagated to the finest level until the global optimal solution is obtained. In each hierarchical sampling process, we have found that 2000 samples are sufficient. In our experiment, we observe that the initialization from higher level is usually close to the ground truth. Therefore, our algorithm runs very fast, taking 0.5s, 1.5s and 6s to output the face shape from coarse to fine for an image of size 128×128.

We designed an experiment to demonstrate the importance of the GL distribution. We select a face image with significant side illumination and run two HDDMCMC algorithms with the only difference that the local likelihood of each feature point is independent or not[1]. The results are listed in Fig. 6, where (b) and (c) are the results of the independent local likelihood and GL distribution, respectively. It can be observed that the face contour is localized much more accurately in (c) than in (b). The global property of the likelihood plays an

[1] An interesting comparison could be between HSM with GL distribution and ASM (with independent local likelihood). It is, however, fair to compare HSM with GL distribution and with independent local likelihood.

important role in face localization, which is appropriately modeled in the GL distribution.

Finally we test our algorithm on some challenging face images, shown in Fig. 7. There are typically four cases: (a) intensive expression, (b) unusual lighting condition, (c) noisy and low-quality image and (d) face very different from the training data. Overall the results are satisfactory. It is interesting to note that in (a) the bottom lip is mismatched to the bottom teeth. This is the drawback of the shape model which merely takes into account the local image patterns associated with the feature points. Note that no Asian faces are used in our training data, yet we obtain good localization results in (b) and (d). Despite the poor lighting condition in (b), our algorithm is able to generalized the learnt distributions and obtain a good localization result. Because image (c) has very low resolution, we up-sample it and still obtain good localization without false alarm.

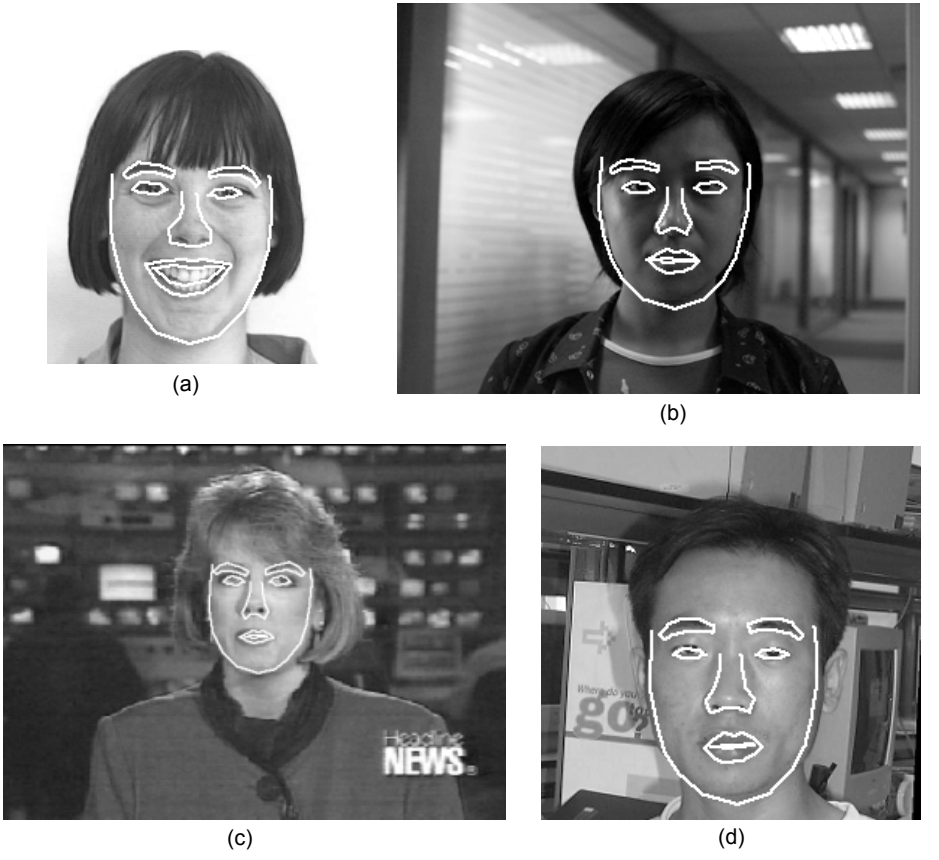

Fig. 7. The results of HSM in face localization with challenging conditions. (a) Intensive expression. (b) Unusual lighting. (c) Noisy and low-quality image. (d) Appearance that is very different from the training data.

6 Summary

In this paper, we build a hierarchical shape model for faces and employ HDDMCMC to automatically locating a face in an image. In this way, two major problems in previous shape models, *i.e.*, huge solution space and rather inaccurate model for likelihoods, are addressed. Even though MCMC is well known for its inefficiency, the HDDMCMC runs very fast because (a) it proceeds from coarse to fine with solution space sharply reduced and (b) salient proposal densities integrating both top-down and bottom-up processes are designed to guide the Markov chain. We model the joint distribution of local likelihoods via global and local (GL) distributions to reserve the global correspondence and the local details of local features associated with the key points. Our experimental results indicate that both modeling and learning of the distributions in HSM are accurate and robust.

A large part of our work focuses on how to deal with high dimensional distributions. The key idea in our approach is to simplify a complex correspondence by introducing hidden variable. We have also found principal components analysis and reversible jump MCMC are effective in linear dimensionality reduction and density learning in HSM. In fact, the GL distribution can be applied to general vision problems.

References

1. T.Cootes and C.Taylor. Statistical Models of Appearance for Computer Vision. Technical report, University of Manchester, 2000.
2. T.Cootes and C.Taylor. Constrained Active Appearance Models. *In Proceedings of the 8th ICCV*, July, 2001.
3. P.Green. Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. *Biometrika*, vol. 82, pp. 711-732, 1995.
4. T.Leung, M.Burl, and P.Perona. Finding Faces in Cluttered Scenes using Random Labeled Graph Matching. *In Proceedings of the 5th ICCV*, June, 1995.
5. A.Martinez and R.Benavente. The AR Face Database. CVC Technical report, No. 24, June 1998.
6. E.Osuna, R.Freund, and F.Girosi. Training Support Vector Machine: An Application To Face Detection. *In Proceedings of CVPR'97*, pages 130-136, 1997.
7. P.Philips. H.Moon, P.Pauss, and S.Rivzvi. The FERET Evaluation Methodology for Face Recognition Algorithms. *In Proceedings of CVPR'97*, pp.137-143, 1997.
8. S.Roberts, C.Holmes, and D.Denison. Minimum-Entropy Data Partitioning Using Reversible Jump Markoc Chain Monte Carlo. *IEEE Transactions on PAMI*, 23(8):909-914, August, 2001.
9. H.Rowley, S.Baluja, and T.Kanade. Neural Network-Based Face Detection. *IEEE Transactions on PAMI*, 20(1), January 1998.
10. H.Schneiderman and T.Kanade. A Statistical Method for 3D Object Detection Applied to Faces and Cars. *In Proceedings of the 7th ICCV*, May, 2000.
11. K.Sung and T.Poggio. Example-based Learning for View-based Human Face Detection. *IEEE Transactions on PAMI*, 20(1):39-51, 1998.
12. Z.Tu and S.Zhu. Image Segmentation by Data Driven Markov Chain Monte Carlo. *In Proceedings of the 8th ICCV*, July, 2001.

13. M.Turk and A.Pentland. Eigenface for Recognition. *Journal of Cognitive Neurosciences*, pages 71-86, 1991.
14. P.Viola and M.Jones. Robust Real-time Face Detection. *In Proceedings of the 8th ICCV*, July, 2001.
15. A.Yuille, P.Hallinan, and D.Cohen. Feature Extraction from Faces using Deformable Templates. *International Journal of Computer Vision*, vol. 8, no. 2, pp. 99-111, 1992.