International Workshop on Analysis and Modeling of Faces and Gestures, October 2005

An Integrated Two-stage Framework for Robust Head Pose Estimation

Junwen Wu and Mohan M. Trivedi

Computer Vision and Robotics Research Laboratory
University of California, San Diego
La Jolla, CA 92093, USA
{juwu, mtrivedi}@ucsd.edu

Abstract. Subspace analysis has been widely used for head pose estimation. However, such techniques are usually sensitive to data alignment and background noise. In this paper a two-stage approach is proposed to address this issue by combining the subspace analysis together with the topography method. The first stage is based on the subspace analysis of Gabor wavelets responses. Different subspace techniques were compared for better exploring the underlying data structure. Nearest prototype matching using Euclidean distance was used to get the pose estimate. The single pose estimated was relaxed to a subset of poses around it to incorporate certain tolerance to data alignment and background noise. In the second stage, the uncertainty is eliminated by analyzing finer geometrical structure details captured by bunch graphs. This coarse-to-fine framework was evaluated with a large data set. We examined 86 poses, with the pan angle spanning from -90^o to 90^o and the tilt angle spanning from -60^o to 45^o. The experimental results indicate that the integrated approach has a remarkably better performance than using subspace analysis alone.

1 Motivation and Background

Head pose can be used for analyzing subjects' focus of attention in "smart" environment [1][2][3]. Head pose is determined by the pan angel β and the tilt angle α, as shown in the right image of Fig. 1. For applications in driver assistance systems, accuracy and robustness of the head pose estimation modular is of critical importance [3]. Besides focus analysis, head pose estimation is also a very useful front-end processing for multi-view human face analysis. The accurate pose estimate can provide necessary information to reconstruct the frontal view face for a better facial expression recognition [4]. Pose estimation can also help select the best view-model for detection and recognition [5][6].

Over the past several years, head pose estimation has been an active area of research. If there are multiple images available, pose position in the 3D space can be recovered using the face geometry. The input could be video sequences [3][4][7][8] as well as multi-camera output [9][10]. Following techniques have been proposed: feature tracking, including tracking the local salient features [4][8] or the geometric features [3][7]; studying the joint statistical property of image intensity and the depth information [9][10]. With only static images available, the 2D pose estimation problem has presented a different challenge. Pose can only be determined in certain degrees of freedom (DOF), instead of the full 6 DOF as the 3D one does. 2D pose estimation can be used as the front-end for multi-view face analysis [5][11]; as well as to provide the initial reference frame for 3D head pose tracking. In [12], the author investigated the dissimilarity between poses by using some specific filters such as Gabor filters and PCA. This study indicates that identity-independent pose can be discriminated by prototype matching with suitable filters. Some efforts have been put to investigate the 2D pose estimation problem [5][6][11][13][14] and they are mainly focused on the use of statistical learning techniques, such as SVC in [5], KPCA in [11], multi-view eigen-space in [14], eigen-space from *best* Gabor filter in [13], manifold learning in [6] etc. All these algorithms are based on the features from entire faces. Although the identity information can be well-suppressed, one main drawback of such techniques is that they are sensitive to the face alignment, background and scale. Some researchers also explored the problem by utilizing the geometric structure constrained by representative local features [15, 16]. In [15], the authors extended the bunch graph work from [17] to pose estimation. The technique provides the idea to incorporate the geometric configuration for the 2D head pose estimation. However, the study is only based on 5 well-separated poses. The other poses not included can be categorized into these 5 poses by extensive elastic searching. Although this benefits the multi-view face recognition problem, it is not suitable for head pose estimation in a fine scale, since the elastic searching introduces ambiguity between similar poses. In [16], Gabor wavelets network, or GWN, which is constructed from the Gabor wavelets of local facial features, was used to estimate the head pose. One drawback is that it requires selected facial features to be visible, hence not suitable for head pose estimation with wide angle changes.

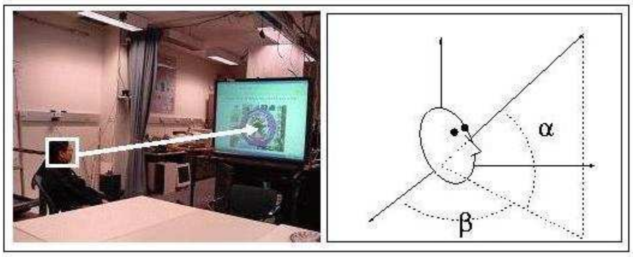

Fig. 1. Illustration of head pose estimation in focus analysis.

In this paper, our aim is to get a robust identity independent pose estimator over a wide range of angles. We propose a two-stage framework which combines the statistical subspace analysis together with the geometric structure analysis for more robustness. The main issue we want to solve is the robustness to data alignment and background. More details are discussed below.

2 Algorithm Framework

The proposed solution is a two-stage scheme in a coarse-to-fine fashion. In the first stage, we use subspace analysis in a Gabor wavelet transform space. Our study indicates that statistical subspace analysis is insufficient to deal with data misalignment and background noise, however, the noise does not drive the estimate far from its true value. Therefore, we can assume that the true pose locates in a subset of $p \times p$ neighboring poses around the estimate with a high accuracy. We use the subset of poses as the output from the first stage. This is similar to a fuzzy decision. The first-stage accuracy is evaluated accordingly: if the true pose locates in the $p \times p$ subset around the estimate, the estimate is determined as a correct one. Since geometric structure of the local facial features has the ability to provide the necessary detail for a finer pose assessment, in the second stage, we use a structural landmark analysis in the transform domain to refine the estimate. More specifically, we use a revised version of the face bunch graph [17]. The diagrams in Fig. 2 outline this algorithm.

To get a comprehensive view of the underlying data structure, we study four popular subspaces so that the best subspace descriptors can be found: Principle Component Analysis (PCA) [18]; Kernel Principle Component Analysis (KPCA) [19]; Multiple class Discriminant Analysis (MDA) [18] and Kernel Discriminant Analysis (KDA) [20, 21]. Results show that analysis in the kernel space can provide a better performance. Also, discriminant analysis is slightly better than PCA (please refer to Table 1).

To refine the estimate from the first-stage, *semi-rigid* bunch graph is used. Different from the face recognition task solved in [17], we only need to recover the identity-independent head pose. In [17], an exhaustive elastic graph searching is used so as to find the fiducial points that contains subjects' identity. However, the distortion in the geometric structure caused by the exhaustive elastic search would introduce ambiguity for close poses. Furthermore, for pose estimation, we do not require exact match of the fiducial points since the nodes from Gabor jets are actually able to describe the neighborhood property. That is the reason we use the "semi-rigid" bunch graph, in which the nodes can only be individually adjusted locally in legitimate geometrical configurations. We use multiple bunch graphs per pose to incorporate all available geometric structures. The reason is that the geometric structure captured by a single model graph is not subject-independent. Simply averaging is not sufficient to describe all subjects. Since the first stage estimation restricts the possible candidate in a small subset, the computational cost is still reasonable.

The data span pan angles from -90^o to 90^o and tilt angle from -60^o (head tilt down) to 45^o (head tilt up). 86 poses are included, as shown in Fig. 3.

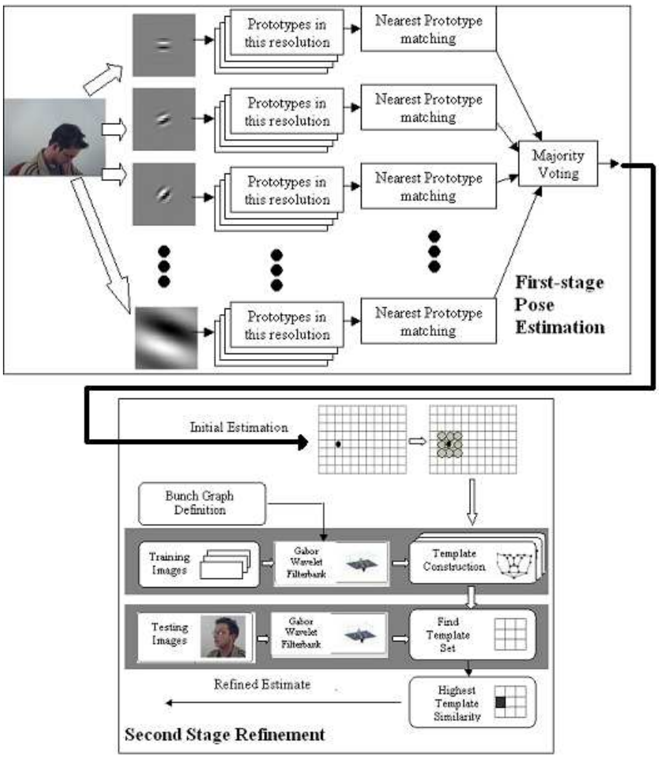

Fig. 2. Flowchart of the two-stage pose estimation framework. The top diagram is for the first-stage estimation and the bottom one is for the second-stage refinement. The output of the first stage is the input of the second stage.

3 Stage 1: Multi-resolution Subspace Analysis

Gabor wavelet transform is a convolution of the image with a family of Gabor kernels. All Gabor kernels are generated by a mother wavelet by dilations and rotations. Gabor wavelets provide a good joint spatial frequency representation. DC-free version of the Gabor wavelets can suppress the undesired variations, such as illumination change. Also, optimal wavelets can ideally extract the position and orientation of both global and local features [22]. Only magnitude responses are used in our algorithm since the phase response is too sensitive.

3.1 Subspace projection

The wavelet features suffer from high dimensionality and no discriminant information are extracted. Subspace projection is used to reduce the dimensionality as well as extracting the most representative information. In this paper, we compare four popular subspaces for better discovering

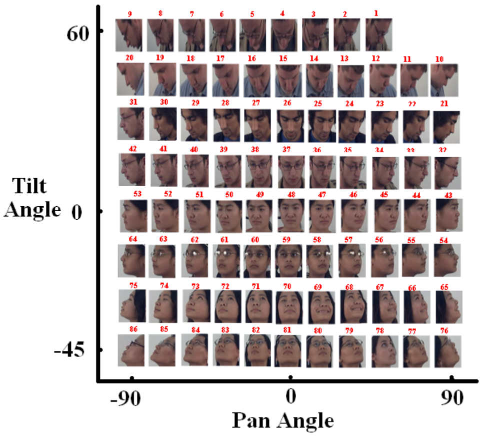

Fig. 3. Examples of the image set. The top two poses are not discussed because of lacking of enough samples.

the underlying data structure, which are PCA, MDA and their corresponding nonlinear pair. For the clarity of presentation, in the following sections, the data set is denoted as $\{\mathbf{x}_i\}_{i=1,\cdots,N}$ with C classes. Samples from c-th class are denoted as $\mathbf{x}_{c,i}, i = 1, \cdots, N_c$, where $N = \sum_{c=1}^{C} N_c$ and $\{\mathbf{x}_i\}_{i=1,\cdots,N} = \cup_{c=1}^{C}\{\mathbf{x}_{c,j}\}_{j=1,\cdots,N_c}$.

Linear subspace projection PCA aims to find the subspace that describes most variance while suppresses known noise as much as possible. PCA subspace is spanned by the principal eigenvectors of the covariance matrix, which is:

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^{\mathrm{T}}; \tag{1}$$

where $\boldsymbol{\mu}$ is the sample mean: $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$. The principal components are computed by solving the following eigen-decomposition problem:

$$\mathbf{\Sigma}\mathbf{V} = \mathbf{\Lambda}\mathbf{V}; \tag{2}$$

where $\mathbf{\Lambda}$ is the diagonal matrix whose non-zero entries are the eigenvalues λ_i of $\mathbf{\Sigma}$. $\mathbf{V}$ is the matrix from eigenvectors. λ_i indicates the information preserved on the corresponding eigenvector direction. By picking the eigenvectors with the largest eigenvalues the information lost is minimized in the mean-square sense.

While PCA looks for a projection subspace with minimal information lost, discriminant analysis seeks a projection subspace efficient for classification. The basic idea is to find a projection, in which the within class data are compactly represented while the between class data are well-separated. We use a multiple class discriminant analysis as introduced in [18]. The within-class scatter matrix $\mathbf{S}_W$ is used to evaluated the data compactness, defined as follows:

$$\mathbf{S}_W = \sum_{c=1}^{C} \sum_{i=1}^{N_c} (\mathbf{x}_{c,i} - \boldsymbol{\mu}_c)(\mathbf{x}_{c,i} - \boldsymbol{\mu}_c)^{\mathrm{T}}; \tag{3}$$

with $\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{x}_{c,i}$ as the class mean. The separability between data from different classes is evaluated by the between-class scatter matrix as follows

$$\mathbf{S}_B = \sum_{c=1}^{C} N_c (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^{\mathrm{T}}; \tag{4}$$

where $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$ is the overall sample mean. The subspace is found by Fisher's criterion, which maximize the Raleigh coefficient:

$$\mathcal{J}(\mathbf{V}) = \frac{\mathbf{V}^{\mathrm{T}} \mathbf{S}_B \mathbf{V}}{\mathbf{V}^{\mathrm{T}} \mathbf{S}_W \mathbf{V}}. \tag{5}$$

This turns out to be an eigen-decomposition problem. The solution can be found by solving the generalized eigen-decomposition problem $\mathbf{S}_B \mathbf{v}_i = \lambda_i \mathbf{S}_W \mathbf{v}_i$.

PCA and MDA provide powerful linear techniques for data reduction. However, most interesting data in real world assume certain non-linearities that linear projection can not model. This inspires the use of kernel machine, which explores the non-linearity of the data space. The extended nonlinear alternative, KPCA [19, 23] and KDA [20], are used.

Kernel machine: KPCA and KDA In [11] the use of KPCA for modeling the multi-view faces in the original image space was presented. Assuming data non-linearly distributed, we can map it onto a new higher dimensional feature space $\{\mathbf{\Phi}(\mathbf{x}) \in \mathcal{F}\}$ where the data possess a linear property. The mapping is $\mathbf{\Phi} : \mathbf{x} \mapsto \mathbf{\Phi}(\mathbf{x})$. KPCA is realized by a linear PCA in the transformed space $\mathcal{F}$. The covariance matrix now becomes:

$$\mathbf{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{\Phi}(\mathbf{x}_i) - \mathbf{\Phi}(\boldsymbol{\mu}))(\mathbf{\Phi}(\mathbf{x}_i) - \mathbf{\Phi}(\boldsymbol{\mu}))^{\mathrm{T}}. \tag{6}$$

Sample mean $\mathbf{\Phi}(\boldsymbol{\mu}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{\Phi}(\mathbf{x}_i)$. Only dot product $\mathbf{\Phi}(\mathbf{x}_i) \bullet \mathbf{\Phi}(\mathbf{x}_j)$ is involved, hence no explicit function is needed for the mapping $\mathbf{\Phi}$. Define the kernel as

$$\mathcal{K}(\mathbf{x}_i; \mathbf{x}_j) \equiv \mathbf{\Phi}(\mathbf{x}_i) \bullet \mathbf{\Phi}(\mathbf{x}_j)$$

and the Gram matrix $\mathbf{K}$ as a $N \times N$ matrix with its entry: $\mathcal{K}(\mathbf{x}_i; \mathbf{x}_j)$, $(i, j = 1, \cdots, N)$. The Hilbert space assumption constrains $\mathbf{v}$'s solution space within the span of $\{\mathbf{\Phi}(\mathbf{x}_1), \cdots, \mathbf{\Phi}(\mathbf{x}_N)\}$, which means $\mathbf{v} = \sum_i \alpha_i \mathbf{\Phi}(\mathbf{x}_i)$ $(\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_N]^{\mathrm{T}})$. The linear PCA problem in space $\mathcal{F}$ gives:

$$\mathbf{K}' \boldsymbol{\alpha} = N \lambda \boldsymbol{\alpha}, \tag{7}$$

where $\mathbf{K}'$ is the slightly different version from $\mathbf{K}$ by removing the feature's mean:

$$\mathbf{K}' = (\mathbf{I} - \mathbf{e}\mathbf{e}^{\mathrm{T}}) \mathbf{K} (\mathbf{I} - \mathbf{e}\mathbf{e}^{\mathrm{T}}); \tag{8}$$

$\mathbf{e} = \frac{1}{\sqrt{N}}[1, 1, \cdots, 1]^{\mathrm{T}}$.
The eigen-decomposition of the Gram matrix provides an embedding that captures the low-dimensional structure on the manifold. Hence, a better generalization ability can be achieved. In our implementation, we use the traditional Gaussian kernel.
The same as KPCA, KDA processes data in the transformed space $\mathcal{F}$. Hilbert space is assumed so that k-th projection direction is: $\mathbf{w}_k = \sum_{i=1}^{N} \alpha_i^{(k)} \mathbf{\Phi}(\mathbf{x}_i)$. Introduce the kernel $\mathcal{K}(\mathbf{x}_i; \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i) \bullet \mathbf{\Phi}(\mathbf{x}_j)$ and define an additional kernel matrix $\mathbf{K}_c$ as a $N \times N_c$ matrix whose entry is $\mathcal{K}(\mathbf{x}_i; \mathbf{x}_{c,j})$ $(i = 1, \cdots, N,\ j = 1, \cdots, N_c)$. Now the scatter matrices can be represented by:

$$\begin{aligned} \mathbf{W}^{\mathrm{T}}\mathbf{S}_B\mathbf{W} &= \mathbf{W}^{\mathrm{T}} \sum_{c=1}^{C} N_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{W} \\ &= \mathbf{V}^{\mathrm{T}}(\sum_{c=1}^{C} \frac{\mathbf{K}_c\mathbf{1}_c\mathbf{K}_c^{\mathrm{T}}}{N_c} - \frac{\mathbf{K}\mathbf{1}\mathbf{K}}{N})\mathbf{V}; \end{aligned} \tag{9}$$

$$\begin{aligned} \mathbf{W}^{\mathrm{T}}\mathbf{S}_W\mathbf{W} &= \sum_{c=1}^{C} \sum_{i=1}^{N_c} (\mathbf{x}_{c,i} - \boldsymbol{\mu}_c)(\mathbf{x}_{c,i} - \boldsymbol{\mu}_c)^{\mathrm{T}} \\ &= \mathbf{V}^{\mathrm{T}}(\sum_{c=1}^{C} \mathbf{K}_c\mathbf{K}_c^{\mathrm{T}} - \sum_{c=1}^{C} \frac{\mathbf{K}_c\mathbf{1}_c\mathbf{K}_c^{\mathrm{T}}}{N_c})\mathbf{V}. \end{aligned} \tag{10}$$

where $\mathbf{1}$ is an $N \times N$ matrix with all 1 entries and $\mathbf{1}_c$ is an $N_c \times N_c$ matrix with all 1 entries. The new projection matrix is $\mathbf{V} = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_m]$ with $\boldsymbol{\alpha}_k = [\alpha_1^{(k)}, \cdots, \alpha_N^{(k)}]^{\mathrm{T}}$. The Raleigh's coefficient now becomes:

$$\mathcal{J}(\mathbf{V}) = \frac{\mathbf{V}^{\mathrm{T}}(\sum_{c=1}^{C} \frac{1}{N_c}\mathbf{K}_c\mathbf{1}_c\mathbf{K}_c^{\mathrm{T}} - \frac{1}{N}\mathbf{K}\mathbf{1}\mathbf{K})\mathbf{V}}{\mathbf{V}^{\mathrm{T}}(\sum_{c=1}^{C} \mathbf{K}_c\mathbf{K}_c^{\mathrm{T}} - \sum_{c=1}^{C} \frac{1}{N_c}\mathbf{K}_c\mathbf{1}_c\mathbf{K}_c^{\mathrm{T}})\mathbf{V}}. \tag{11}$$

Similar as its linear alternative, KDA projection is pursued by maximizing the Raleigh's coefficient.
In Fig. 4 and Fig. 5, 2D toy examples are used to illustrate the four subspace analysis methods. In Fig. 4, the original 2D data are projected onto the 1D PCA and LDA subspace as shown. LDA can well-separate the data while PCA cannot. In Fig. 5, we illustrate the separation abilities for nonlinear data set. All four subspace projections are compared on a binary 2D toy data set. As can be seen, PCA and LDA are not able to produce a more discriminating representations due to the non-linearity of the data, whereas the KPCA and KDA transform the data into two well-separated clusters.

3.2 Prototype matching

We use the nearest prototype matching for the first stage classification. Each pose is represented by a set of subspaces, each of them computed from filter responses in one resolution. In each subspace the prototype

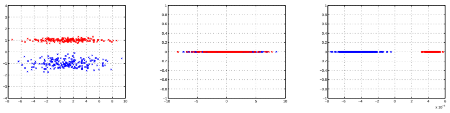

Fig. 4. Illustrative example of PCA and LDA subspace representation. The data from two classes are shown in red and blue individually. Left: original data; middle: projected data from PCA subspace; right: the projected data from LDA subspace.

from class mean is found as a template. Euclidean distances is used to measure the similarity in subspaces. The pose estimation is given by the prevailing class label from all resolutions as illustrated in Fig. 2. This gives a single pose as an estimate. We relax the single estimated pose label to a subset of 3 × 3 poses around it for additional robustness. A second-stage is applied thereafter to solve the sub-problem, where only poses in the subset are tackled.

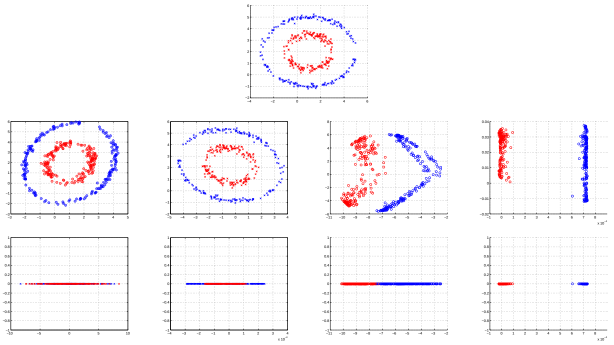

Fig. 5. Illustrative examples of the subspace representation for nonlinear data. Red color and blue color indicate samples from different classes. First row: the original data. Row 2-3: transformed data (top: 2D space; bottom: 1D space). From column 1 to column 4: PCA, LDA, KPCA, KDA. Kernel: same Gaussian kernel.

4 Stage 2: Geometric Structure Analysis

The second stage serves to refine the coarse pose estimation. In this section, we use a revised version of the face bunch graph introduced in [17] for this purpose. Face graph is a labeled graph which connects the local image features together with the image's geometric configuration. It exploits the local salient features on a human face, e.g. pupils, nose tip, corners of mouth, and etc. together with their locations.

4.1 Bunch graph construction

Each face image constructs a model graph. The model graph is a labeled graph with its nodes corresponding to the Gabor jets at the predefined

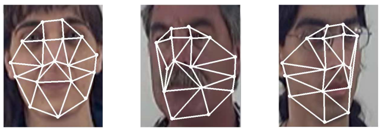

Fig. 6. Examples of the face model graph. Left: pan: 0^o tilt: 0^o; middle: pan: -15^o tilt: 0^o; right: pan: $+15^o$ tilt: 0^o.

salient facial features, and its edges labeled by the distance vector between nodes. Gabor jet is defined as the concatenation of the Gabor wavelet responses at an image point. Some examples of the model graphs are show in Fig. 6. Occlusion of the current view determines how many nodes are used. More nodes assert more geometric constraint useful for pose discriminating, however, more identity information could be preserved.

Each view is modeled by one set of bunch graphs from the model graphs of the same pose. The nodes of the bunch graph are the bundles of the corresponding nodes in model graphs. The geometric structure is subject-dependent in a certain degree. Subjects from different race, age group, or different gender possess different geometric configuration. Although a simple average of all the geometric configurations followed by an exhaustive search and match can still be used to find the identity-related fiducial points, this step will also add ambiguity to the global structure between close poses. In the purpose of retrieving the pose information while suppressing the subject identity, we keep every available geometric configuration and use a *semi-rigid* searching for matching, which means only local adjustment is allowed for refine the estimated face graph. Therefore, for each pose, we actually have the same number of bunch graphs as the model graphs. Each bunch graph inherits the edge information from an individual model graph. All the bunch graphs differ only in the edge labels. This is illustrated in Fig. 7. This strategy enables us to avoid large distortions in geometric structure that causes ambiguities between neighboring poses. This offline model construction step gives each pose a set of bunch graphs as the templates.

4.2 Graph matching

Denote the subset of poses confined by the first stage estimation as $\mathcal{P}_s$. Given a test image, every pose candidate in $\mathcal{P}_s$ gives an estimated face graph by searching the sets of nodes that maximize the graph similarity. Graph similarity is determined by both the similarity of the nodes and the distance in edge labels. We use the normalized cross correlation as the nodes similarity metric [17]. Let $\mathbf{J}(i) = (f_1(i), \cdots, f_F(i))$ be the Gabor jet for i-th nodes. Nodes similarity D is given by:

$$D(\mathbf{J}(i); \mathbf{J}(k)) = \frac{\sum_{m=1}^{F} f_m(i) f_m(k)}{\sqrt{\sum_{m=1}^{F} f_m^2(i) \sum_{m=1}^{F} f_m^2(k)}}. \qquad (12)$$

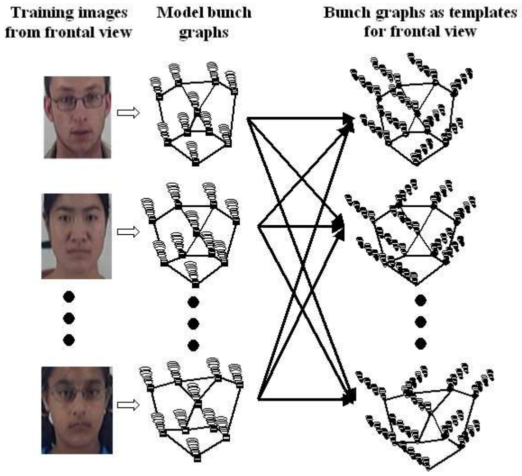
Training images from frontal view
Model bunch graphs
Bunch graphs as templates for frontal view

Fig. 7. Construction of the bunch graphs as the template for a single pose. Frontal view is used. The graphs shown here are just for illustration. In actual computation, more nodes are used, hence the graph structure is different from that shown here.

The graph similarity S between the estimated face graph $\mathcal{G} = (\mathbf{J}_m, \delta_e)$ and some bunch graph $\mathcal{B} = (\{\mathbf{J}_m^{B_i}\}_i, \delta_e^B)$ is defined as:

$$\mathcal{S}(\mathcal{G}, \mathcal{B}) = \frac{1}{M}\sum_{m=1}^{M} \max_i (D(\mathbf{J}_m; \mathbf{J}_m^{B_i})) - \frac{\lambda}{E}\sum_{e=1}^{E} \frac{(\delta_e - \delta_e^B)^2}{(\delta_e^B)^2}; \qquad (13)$$

where λ is the relaxation factor.
Since we have multiple bunch graphs for a single pose, each of them can generate a possible face graph for the testing image. The best matched one needs to be found as the representative face graph for this pose. This best face graph estimate is given by the following steps:

1. Scan the testing image. Each rigid topographic constraint ($\lambda = \infty$) determined by one bunch graph gives a set of matching nodes, and hence a graph $\mathcal{G}_t$. Out of which the best matched one is:

$$t^\star = \arg\max_t \mathcal{S}(\mathcal{G}_t, \mathcal{B}_t),$$

   with $\lambda = \infty$.
2. The nodes of the best matched estimated graph $\mathcal{G}_{t^\star}$ are individually adjusted locally to refine the match.
3. Refined nodes determines the graph.

The best geometric configuration $t^\star$ is selected and the graph similarity between the estimated face graph and the $t^\star$-th bunch graph is evaluated by equation 13. The pose with the highest similarity score gives the final pose estimation.

5 Experimental Evaluations

The data set used for evaluating this approach includes 28 subjects. Magnetic sensor is used to provide the ground-truth. Some poses are excluded due to lack of enough samples (see Fig. 3). We include 86 poses. The pan angle spans from -90^o to $+90^o$; with 15^o intervals from -60^o to 60^o, and then the poses with $\pm 90^o$ pan angles are also considered. The tilt angle has a consistent interval of 15^o from -45^o to 60^o. 3894 images of size 67×55 and their mirror images are used, so altogether 7788 images included. Each pose has 80$\sim$100 images, randomly split into two parts, one for training and one for testing. Some subjects may have multiple samples for one pose, assuming sufficient different facial expressions. We use Viola and Jone's face detector [24] to get the face area. 9 separate detectors are trained for different views. For each image, we manually select one detector according to the ground-truth of the head pose.

5.1 Stage 1: "coarse" pose estimation

Output of the first stage is a $p \times p$ subset of poses. The accuracy is evaluated accordingly: if the true pose does not belong to this subset, it is counted as a false estimate. In our implement $p = 3$ is used if not specially stated. Bigger p gives better accuracy, however, more computational cost will be needed for the second stage refinement. In table 1, the first-stage estimation for different subspaces are evaluated under different p. To better present the error distribution, in Fig. 8 we use a color coded error distribution diagram to show the accuracy for each pose for KDA subspace (evaluated under $p = 3$). Darker color shows more error. All four subspace didn't give a satisfactory results comparable with those reported when $p = 1$, which is actually the accuracy of using subspace analysis alone. This is not a surprise, since the subspace analysis is very sensitive to the data noise, such as background and data alignment. In our data set, the face position is not well-aligned. Also in some images parts of the hair and shoulder appears while not in the other. In such case, the subspace analysis alone is not capable to obtain as good performance. The use of the two-stage framework solves this problem. More experiments validate the advantage of the two-stage framework. We purposely translate the cropping window for the testing face images

	p=1	**p=3**	p=5
PCA	36.4	**86.6**	96.9
MDA	40.1	**88.0**	97.3
KPCA	42.0	**90.2**	99.2
KDA	50.3	**94.0**	97.9

Table 1. First-stage multi-resolution subspace analysis results evaluated under different p.

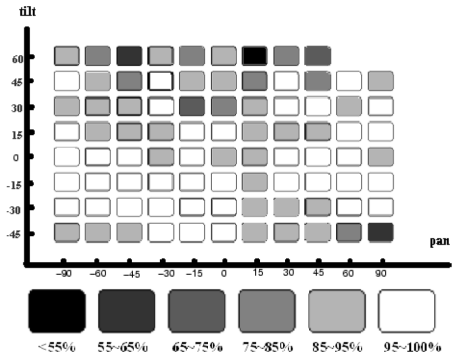

Fig. 8. Color coded error distribution diagram for KDA subspace ($p = 3$).

by $\pm 2, \pm 4, \pm 6, \pm 8, \pm 16$ pixels in both directions, which aggravates the misalignment. Use the same KDA subspace obtained in previous step to test the performance. The accuracy is evaluated for both $p = 1$ and $p = 3$, as show in Fig. 9. Experimental results indicate that when using $p = 3$ to evaluate the accuracy, the accuracy is actually quite stable with the aggravating misalignment. However, when $p = 1$, the accuracy keeps stable for small misalignment (<4 pixels), and drops fast with increasing misalignment. Since the second-stage is not affected by the misalignment, if we can get a stable output for the first-stage with increasing misalignment, the overall accuracy would be stable. This shows the advantage of the 2-stage framework.

5.2 Stage 2: refinement

We only use the best results, which is from KDA subspace analysis, as the first-stage output. The pose estimation accuracy after the refinement is summarized in Table 2. The accuracy was evaluated by the ratio of samples that were correctly classified. Pose with tilt angle 60^o get poor performance. It is because of the severe occlusion. Discarding these poses, the overall accuracy can be improved to 81.3%. For comparison, a second stage refinement by multi-resolution MDA analysis is also performed, using the poses confined by the first stage. The results are shown in Table 3. The comparison shows that by introducing the second-stage structure landmark matching, the estimation accuracy has a markable improvement.

6 Concluding Remarks

In this paper we discussed a two-stage approach for estimating head pose from static images. We use statistical subspaces analysis in Gabor wavelet

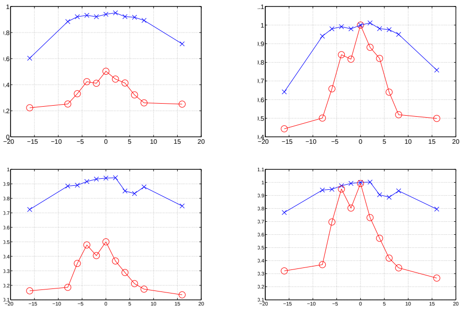

Fig. 9. The performance with the added misalignment $(\pm 16, \pm 8, \pm 6, \pm 4, \pm 2)$ in both directions. Top row: misalignment in the horizontal direction. Bottom row: misalignment in the vertical direction. Left column: accuracy change with misalignment. Right column: relative accuracy change with misalignment. Blue curve with x: evaluated on $p = 3$. Red curve with o: evaluated on $p = 1$.

domain to confine the possible range of the head pose. Semi-rigid bunch graph was used to systematically analyze the finer structural details associated with facial features, so as to refine the first-stage estimate. The combination of statistical analysis on features from entire images with the geometrical topograph driven approach provides a robust way to estimate the head pose in a fine scale. It solves the internal problem of the statistical analysis approach that requires a high-quality data set, as well as introducing the methodology of decomposing a large classification problem into smaller sub-problem, so that template matching is feasible. Experimental results show that better performance can be obtained than statistical analysis alone.

Acknowledgement

Our research was supported by grants from the UC Discovery Program and the Technical Support Working Group (TSWG) of the US Department of Defense. The authors are thankful for the assistance and support of our colleagues from the UCSD Computer Vision and Robotics Research Laboratory, especially Jens M. Pedersen, Pew Putthividhya and Daniel Norgaard. The authors also thank Thomas B. Moeslund for his contributions.

References

1. R. Pappu and P.A. Beardsley. Qualitative approach to classifying gaze direction. In *Proceedings of the IEEE Conf on Automatic Face and Gesture Recognition.*, 1998.
2. R. Stiefelhagen. Tracking focus of attention in meetings. In *Proceedings of the IEEE International Conference on Multimodal Interfaces (ICMI'02).*, 2002.

	-90^o	-60^o	-45^o	-30^o	-15^o	0^o	15^o	30^o	45^o	60^o	90^o
60^o	16.7	30.8	11.1	50.0	50.0	20.0	30.8	23.1	25.0		
45^o	80.5	77.5	65.6	65.4	79.5	88.2	76.7	83.6	58.3	77.8	67.1
30^o	80.8	74.7	81.5	73.0	76.3	85.3	71.1	82.2	87.6	83.3	86.6
15^o	75.4	71.4	80.5	84.4	94.3	84.0	79.4	82.4	85.7	71.4	87.3
0^o	87.5	83.4	77.0	88.8	91.6	86.0	87.9	79.0	83.1	79.6	79.3
-15^o	67.0	81.2	85.0	78.9	82.5	80.4	89.1	75.9	74.5	84.1	87.0
-30^o	80.9	84.1	92.2	67.7	89.5	64.7	94.2	75.9	87.0	72.1	76.8
-45^o	82.9	65.7	77.1	81.6	76.7	65.6	68.8	81.6	71.1	75.0	30.0

Table 2. The overall accuracy (%) using KDA subspace majority voting for the first stage estimation and the semi-rigid bunch graph matching as the second stage refinement. The accuracy is 75.4%.

KDA +BG	PCA +MDA	MDA +MDA	KPCA +MDA	KDA +MDA
75.4	43.1	44.0	47.3	53.4

Table 3. Comparison of results from different second-stage refinement.

3. K. Huang, M. M. Trivedi and T. Gandhi. Driver's View and Vehicle Surround Estimation using Omnidirectional Video Stream. In *Proceedings of IEEE Intelligent Vehicles Symposium*, Columbus, OH, pp. 444-449, June 9-11, 2003.
4. B. Braathen, M. S. Bartlett, and J. R. Movellan. 3-d head pose estimation from video by stochastic particle filtering. In *Proceedings of the 8th Annual Joint Symposium on Neural Computation.*, 2001.
5. Y.Li, S.Gong, and H. Liddell. Support vector regression and classification based multi-view face detection and recognition. In *Proceeding of IEEE International Conference on Automatic Face and Gesture Recognition.*, pages 300–305, July 2000.
6. S.Z. Li, Q.D. Fu, L. Gu, B. Scholkopf, Y.M. Cheng, and H.J. Zhang. Kernel machine based learning for multi-view face detection and pose estimation. In *Proceedings of 8th IEEE International Conference on Computer Vision.*, July 2001.
7. M. Cordea, E. Petriu, N. Georganas, D. Petriu, , and T. Whalen. Real-time 2.5d head pose recovery for model-based video-coding. In *Proceedings of the IEEE Instrumentation and Measurement Technology Conference.*, 2000.
8. T. Horprasert, Y. Yacoob, and L. S. Davis. An anthropometric shape model for estimating head orientation. In *Proceedings of the 3rd International Workshop on Visual Form*, 1997.
9. L. Morency, P. Sundberg, and T. Darrell. Pose estimation using 3d view-based eigenspaces. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures, in Conjunction with ICCV2003*, pages 45–52, 2003.
10. E. Seemann, K. Nickel, and R. Stiefelhagen. Head pose estimation using stereo vision for human-robot interaction. In *Proceedings of the*

6th IEEE International Conference on Automatic Face and Gesture Recognition, 2004.

11. L. Chen, L. Zhang, Y. Hu, M. Li, and H. Zhang. Head pose estimation using fisher manifold learning. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures, in Conjunction with ICCV2003*, 2003.
12. S. Gong J. Sherrah and E. Ong. Understanding pose discrimination in similarity space. In *Proceedings of the The Eleventh British Machine Vision Conference (BMVC1999)*, 1999.
13. Y. Wei, L. Fradet, and T. Tan. Head pose estimation using gabor eigenspace modeling. In *Proceedings of the IEEE International Conference on Image Processing (ICIP2002)*, volume 1, pages 281–284, 2002.
14. S. Srinivasan and K.L. Boyer. Head pose estimation using view based eigenspaces. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 4, pages 302–305, 2002.
15. M. Potzsch, N. Kruger, and C. von der Malsburg. Determination of face position and pose with a learned representation based on labeled graphs. Technical report, Institute for Neuroinformatik, RuhrUniversitat, Bochum, 1996. Internal Report.
16. V. Krger and G. Sommer. Efficient head pose estimation with gabor wavelet networks. In *Proceedings of the The Eleventh British Machine Vision Conference (BMVC2000)*, 2000.
17. L. Wiskott, J. Fellous, N. Krger, and C von der Malsburg. Face recognition by elastic bunch graph matching. In *Proceedings of the 7th International Conference on Computer Analysis of Images and Patterns(CAIP'97)*, 1997.
18. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-interscience, second edition.
19. B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
20. Y. Li, S. Gong, and H. Liddell. Recognising trajectories of facial identities using kernel discriminant analysis. In *Proceedings of the British Machine Vision Conference (BMVC2001)*, pages 613–622, 2001.
21. S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller. Fisher discriminant analysis with kernels. In *Proceedings of the IEEE Neural Networks for Signal Processing Workshop*, pages 41–48, 1999.
22. J.MacLennan. Gabor representations ofspatiotemporal visual images. Technical report, Computer Science Department, University of Tennessee, Knoxville., 1991. CS-91-144. Accessible via URL: http://www.cs.utk.edu/ mclennan.
23. J. Ham, D.D. Lee, S. Mika, and B. Scholkopf. A kernel view of dimensionality reduction of manifolds. In *Proceedings of the International Conference on Machine Learning.*, 2004.
24. P Viola and M. Jones. *Robust Real-time Object Detection*, In Proceedings of the Second International Workshop on Statistical and Compu tational Theories of Vision - Modeling, Learning and Sampling. Jointed with ICCV2001.