Pointing'04: Visual Observation of Diectic Gestures, ICPR Workshop, May, 2004.

A Two-level Pose Estimation Framework Using Majority Voting of Gabor Wavelets and Bunch Graph Analysis

Junwen Wu, Jens M. Pedersen, Duangmanee (Pew) Putthividhya, Daniel Norgaard and Mohan M. Trivedi

Computer Vision and Robotics Research Lab

University of California, San Diego

La Jolla, CA 92037, USA

{juwu, mejdahl, putthi, norgaard, mtrivedi}@ucsd.edu

Abstract

In this paper a two-level approach for estimating face pose from a single static image is presented. Gabor wavelets are used as the basic features. The objective of the first level is to derive a good estimate of the pose within some uncertainty. The objective of the second level processing is to minimize this uncertainty by analyzing finer structural details captured by the bunch graphs. The first level analysis enables the use of rigid bunch graph. The framework is evaluated with extensive series of experiments. Using only a single level, 90% accuracy (within ±15 degree) and over 98% (within ±30 degree) was achieved on the complete dataset of 1,395 images. Second level classification was evaluated for three sets of poses with accuracies ranging between 67-73%, without any uncertainty.

1 Introduction

In this paper, we present a two-level classification framework for the accurate pose determination, so as to determine the face pointing direction. The two-level approach is based upon the rationale that visual cues characterizing facial pose has unique multi-resolution spatial frequency and structural signatures. The first level of the approach has the objective of deriving pose estimates with some uncertainty. First level output confines the poses in a small range so that rigid bunch graph can be used thereafter. The objective of the second level processing is to minimize this uncertainty by systematically analyzing the finer structural details captured by the bunch graphs. Gabor wavelets are used as the features. In the coarse level, every Gabor wavelet response is classified using the subspace projection. Two different subspaces are used to get the best descriptors, which are PCA and Kernel Discriminant Analysis (KDA) [1]. The classification results from different Gabor wavelet are combined by majority voting. The first level localizes the poses up to an NxN (N=3) sub-window around the true poses. In the fine level, the pose estimation refined by using rigid bunch graph matching [2][3], which utilizes the geometrical details of the salient facial component.

2 Related research

Human-computer interaction is an active research topic in computer vision and intelligent systems. The essential aim is to determine human's identity and activity in different environment settings [4-6]. Development of practical systems for intelligent environments can utilize gestures; pointers or the direction in which a person's face is pointed

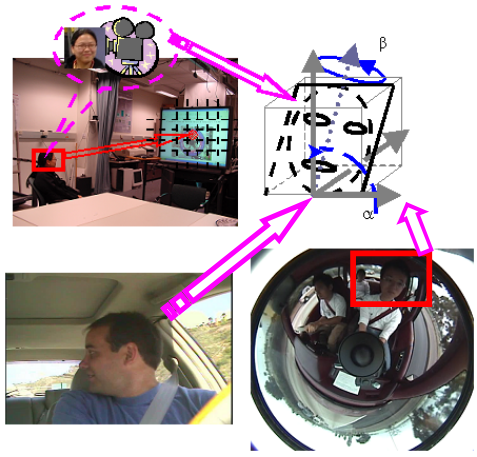

Figure 1. Illustration of face pointing problem and possible applications. Top left image shows the application of face pointing in intelligent room, where the face direction shows the user's focus of attention. The bottom two images are form out system of intelligent vehicles: driver's vigilance based on head pose analysis.

to identify an area of interest [7]. The top right image in Fig. 1 illustrates the face-pointing problem. Face pose is determined uniquely by both the pan angel β and the tilt angle α. The top left and bottom two images give some typical application scenarios for face pointing.

Existing pose estimation algorithms can be categorized into one of the following two classes: 3D pose estimation and 2D pose estimation. For 3D pose estimation, the problem setup is based on multiple inputs. The input could be

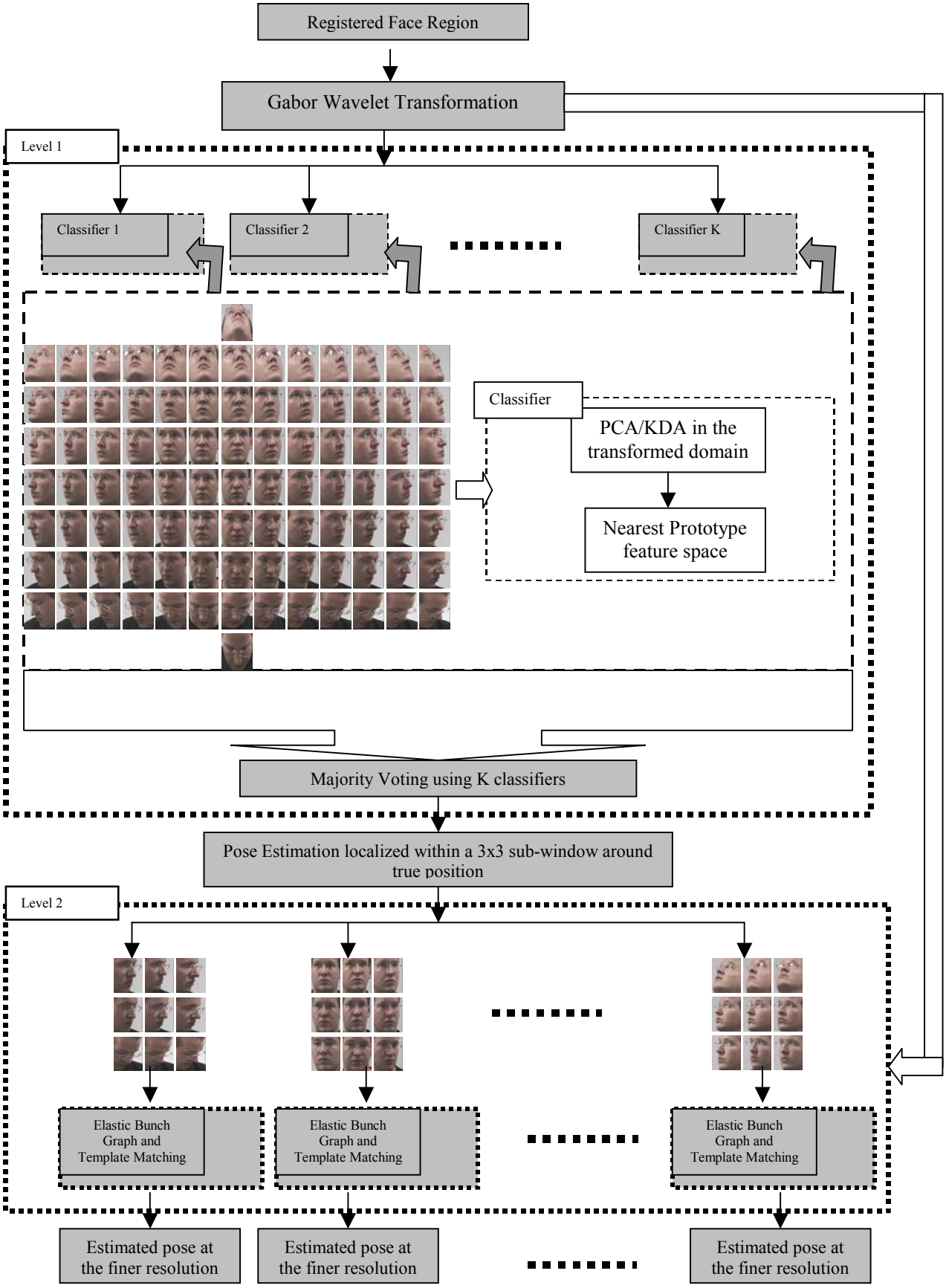

Figure. 2 The Two-level Pose Estimation Framework. Estimates provided by Level-1 processing are refined by considering finer structural details at Level-2.

subsequent frames from a time sequence [8-10], from which the motion of the face, including scaling, translation and rotation, can be obtained by head tracking. This can be used for a variety of computer vision systems. In our own

research we have considered this in the context of an intelligent meeting room [4][6], intelligent vehicles [11], and wide area surveillance [12].

The input could be stereo pair of the face images [13]. Correspondences between the stereo pair are established from salient facial features, using which the depth map can be reconstructed. The 3-D coordinates of the salient facial features are estimated hence after to determine the face pose. The 2D pose estimation problem poses a different challenge. In general, the input is limited to single images. Many approaches have been proposed to investigate the problem [14-16]. However, most efforts are not sufficient for face pointing due to insufficient resolution of the estimation. Also, many researches restrict themselves to the case that poses are different only in the pan angle (angle β as shown in Fig. 1). However, for face pointing applications, both the pan angle and the tilt angle need to be estimated accurately.

3. Face pose estimation approach

Aligned faces are transformed into the multi-scale spatial frequency domain by Gabor wavelets [17]. In our implementation, face region is registered manually to avoid error from alignment. Automatic face cropping can be realized by face detection algorithms [18][19] followed by alignment, or image registration. In Fig. 2 some examples of the cropped face images at different poses are given. PCA and KDA [1] are used to find the most discriminant subspace in the transform domain. A multi-level tree structure is presented to classify the face regions into different poses in a coarse-to-fine fashion. Considering the limited number of samples available, in the first level we use the nearest prototype as the basic classifier. The basic classifier output from wavelets in different scales and orientations is combined by majority voting. It gives estimation with some uncertainty, in the sense that it is accurate up to ±15 degree in both pan and tilt. In the second level, the output is refined by rigid bunch graph [2][3] to give the accurate position. The flowchart of the whole coarse-to-fine scheme is shown as follows in Fig. 2.

In section 3.1, the feature extraction algorithm is shown. In section 3.2, the details of the classification strategy are described.

3.1 Multi-resolution feature extraction

Gabor wavelets are joint spatial frequency domain representation. Frequency domain analysis techniques have a nice property in extracting the structural features as well as suppressing the undesired variants, such as changes of illumination, changes with person identity, etc. Due to its multi-resolution analysis methodology, wavelet is one of the most powerful frequency domain analysis techniques. However, frequency domain representation alone has its essential disadvantage: the localization information is lost. Naturally, people will seek a joint spatial frequency representation. Gabor wavelet is one solution. Gabor wavelets are recognized to be good feature detectors since the optimal wavelets can ideally extract the position and orientation of a local feature. There is considerable evidence [17] that images in primary visual cortex are represented in terms of Gabor wavelet, that is, hierarchically arranged, Gaussian-modulated sinusoids.

3.1.1 Gabor wavelets transformation

A Gabor wavelets transform is defined as a convolution of the image with a family of Gabor kernels. All Gabor kernels are generated by a *mother wavelet* by dilation and rotation. For Gabor Wavelets, the mother wavelet is a plane wave generated from complex exponential and restricted by a Gaussian envelop. In equation (1) - (3), a DC-free mother wavelet is given [2][3]:

$$\psi_{\vec{k}}(\vec{x}) := B(k,x)\left(\exp\left(i\vec{k}\cdot\vec{x}\right)-\exp\left(-\frac{\sigma^2}{2}\right)\right) \tag{1}$$

$$B(k,x)=\frac{k^2}{\sigma^2}\exp\left(-\frac{k^2}{2\sigma^2}x^2\right) \tag{2}$$

$$\left\|\psi_{\vec{k}}(\vec{x})\right\|^2 \sim k^2 \tag{3}$$

The set of Gabor kernel can be given as:

$$\psi_{\vec{k}}(\vec{x}) = k^2\cdot\psi_{\binom{1}{0}}\left(k\Re(\varphi)\cdot\vec{x}\right), \tag{4}$$

where $\vec{k}=(k,\varphi)$ is the spatial frequency in polar coordinates and

$$\Re(\varphi)=\begin{bmatrix}\cos\varphi & \sin\varphi\\ -\sin\varphi & \cos\varphi\end{bmatrix} \tag{5}$$

DC-free versions of Gabor kernels are of great interests to the researchers in computer vision area due to its invariance property to the uniform background illumination change [2][3]. To eliminate the diversity from varying contrast, all filter responses are normalized. An example of the Gabor kernel is shown in Fig. 3 (real part as well as imaginary part).

In our implementation, a family of Gabor kernels with 48 spatial frequencies is used, 6 scales and 8 in orientations. Only the magnitude of the wavelet transformation is used in the feature representation because the phase response is highly sensitive to the non-perfect alignment of the data. Example of the transformed data is shown in Fig. 4.

3.1.2 Feature selection in the transformed domain

The wavelets transform representation suffers from high dimensionality. Subspace projection is used to reduce the dimension. Two different subspaces are used individually, and their performance is compared. One is the PCA subspace projection, and another one is the KDA. PCA is a widely used method in subspace feature extraction. It selects the most representative subspace by finding the orthogonal projection directions that have large variances. However, PCA is calculated based on the second-order statistics of examples from all the classes, it is not clear if

the subspace from PCA contains most discriminant information for classification. KDA is a nonlinear variant of Linear Discriminant Analysis (LDA). For LDA, it finds the projection that maximizes the between-class variance as well as minimizing the within-class variance. However, it is still a linear projection, which will be problematic for severe nonlinear problem. By introducing the kernel trick, KDA is able to get good performance for the nonlinear problem as well. In the first level, both the PCA and the KDA with Gaussian kernel are implemented and the performance is compared. It is not surprising that KDA gets a

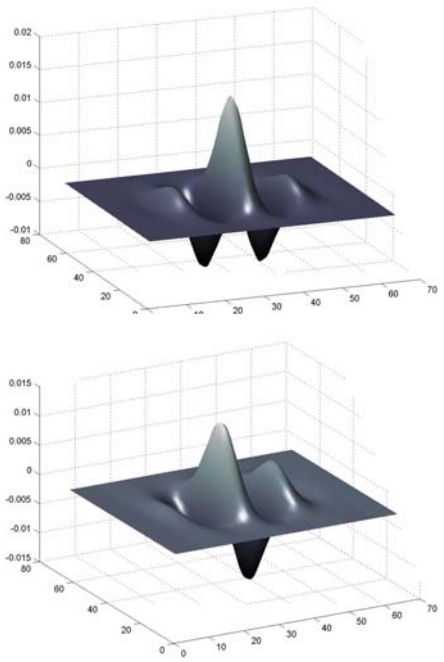

Figure. 3 Example of the Gabor kernel. Top one is the real part and the bottom one is the imaginary part.

better performance than PCA. The following equations (6)-(8) give the PCA transformation:

$$\Phi = \frac{1}{N}\sum_{i=1}^{N}\left((x_i-\mu)(x_i-\mu)^{\mathrm{T}}\right), \tag{6}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \ , \tag{7}$$

$$\Phi V = V\Lambda \ , \tag{8}$$

where $\Lambda = diag(\lambda_1, \lambda_2 \cdots \lambda_D)$ is a diagonal matrix whose elements $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$ are Φ 's eigenvalues. $V = [v_1, v_2, \cdots, v_D]$ is the matrix whose columns are the corresponding eigenvectors. The PCA subspace is formed by the first $M < D$ eigenvectors.

The KDA transformation we use in the implementation is given as follows [1]:

$$A = \left(\sum_{c=1}^{C}\frac{1}{N_c}K_c K_c^{\mathrm{T}}\right)^{-1}\left(\sum_{c=1}^{C}\frac{1}{N_c{}^2}K_c 1_{N_c} K_c^{\mathrm{T}}\right) \tag{9}$$

$$(K_c)_{ij} := k(x_i, x_j), \tag{10}$$

$$k(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}, \tag{11}$$

where K_c is an $N \times N_c$ matrix and N_c is the size of class c. For normalized filter responses, we let $\sigma = 1$. The subspace can be found by eigen-decomposition:

$$AV_A = V_A \Lambda_A, \tag{12}$$

where $\Lambda_A = diag(\lambda_{A1}, \lambda_{A2} \cdots \lambda_{AD})$ is a diagonal matrix with elements $\lambda_{A1} \geq \lambda_{A2} \geq \cdots \geq \lambda_{AD}$, which are A 's eigenvalues. $V_A = [v_{A1}, v_{A2}, \cdots, v_{AD}]$ is the matrix whose columns are the corresponding eigenvectors. The KDA subspace is:

$$U_A = [v_{A1}, v_{A2}, \cdots, v_{AM}], \ M_A < D \tag{13}$$

The KDA projection is obtained by:

$$y = U_A{}^{\mathrm{T}} k_x, \tag{14}$$

where $k_x = (k(x, x_1), \cdots k(x, x_N))$.

The projected vectors y in the subspace are the features we use.

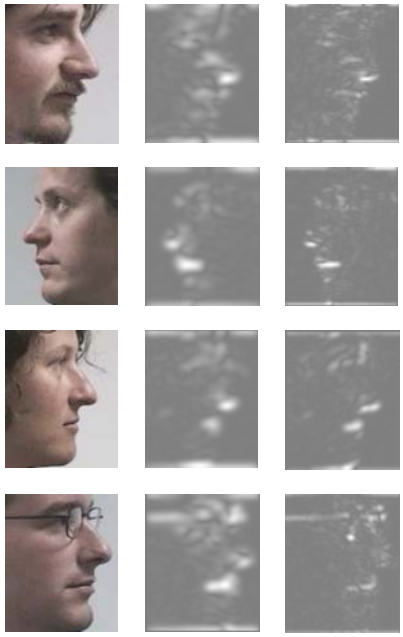

Figure.4 Example of the wavelet transforms. The leftmost column shows the original face regions. The middle column shows the 17th Gabor kernel responses, for which **k**=($2^{1.5}$,0). The rightmost column shows the 33rd Gabor kernel responses, for which **k**=($2^{-1.5}$,0)**.**

3.2 Classification

Two-level classification scheme is proposed. In the first level, the pose is estimated with localization ability up to ± 15 degree in both pan and tilt. It corresponds to the 3×3 neighborhood around the true pose position. Then the problem turns to a 9-class classification problem instead of a 93-class one. This makes it possible to use rigid bunch graphs in the second level to refine the estimation.

3.2.1 Level-1 classification by majority voting

We use the nearest prototype as the basic classifier for the first level classification. For every Gabor wavelet response, class mean in the transformed feature subspace is calculated and used as the prototype. For every Gabor kernel, we can get a basic classifier. Therefore, there are 48 basic classifiers altogether. Assuming that the 48 Gabor wavelets are equally important for the pose estimation, we use the majority voting to determine the pose. The prototype of each class is given by the mean of training samples in the transform domain subspace projection:

$$\mu_{y,c,f} = \frac{1}{N_c} \sum_{i=1}^{N_c} y_{i,f}, \tag{15}$$

where $f = 1, \cdots, 48$ and $c = 1, \cdots, 93$.

$$d(y,c,f) = \left\| y_f - \mu_{y,c,f} \right\|, \tag{16}$$

$$\ell(y,f) = \arg\min_c d(y,c,f). \tag{17}$$

The classification result is given by:

$$C(y) = \arg\max_c \{\#(\ell(y,f) = c)\} \tag{18}$$

Both the feature set from PCA and KDA are used for the first level classification.

3.2.2 Level-2 classification by bunch graph template matching

The coarse pose estimation is refined in the second level. The use of filter responses computed from the entire face image poses certain drawbacks to the problem of accurate pose estimation. Due to a small difference between neighboring poses, PCA and KDA might not be able to select the features that best discriminate poses that are strikingly similar. In this section, we present a landmark based approach which attempt to exploit accurate localization of salient features on a human face, e.g. pupils, nose tip, corners of mouth, and etc. together with their geometric configuration to aid in pose classification. The motivation behind the use of geometric relationships between salient points on a face lies in a simple observation that with different degrees of rotation in depth (both in the pan and tilt directions), the distances between salient points correspondingly change. In this step, we proposed the use of face bunch graph algorithm [2][3] to first accurately locate a predefined set of salient features on a face. Template matching is used in the second level refinement.

3.2.2.1 Face representation & Model Graph Generation

The basic object representation that we use is labeled graph. In our implementation, we adopt the same representation of face bunch graph as used in [2][3] for the task of face recognition. A face is represented as a graph with nodes corresponding to the wavelet responses of Gabor kernels in different scales and orientations. The nodes are connected and labeled with distance information. Our implementation uses the responses from 5 scales and 8 orientations of Gabor kernels. For each pose, a model graph for is generated. First, the issue of which salient points on a face to be used as nodes is addressed. In the frontal parallel view case as shown in the leftmost image of Fig. 5, 19 nodes are selected. In a more oblique view in the middle and right of Fig. 5, only 11 nodes are used. To generate a model graph for each pose, all the 15 training images are used. A face bunch graph is constructed by bundling the model graph from each training image together as shown in Fig. 5 [2][3].

3.2.2.2 Similarity Measurement

The cascade of the wavelets responses for each node is called jets. Matching between different graphs is realized by evaluating the similarity between the ordered Gabor jets [2][3]. The similarity function is used as proposed in, where $x_j(f)$ corresponds to the j^{th} sample's magnitude response of the f^{th} filter.

$$S_x(J_i, J_j) = \frac{\sum_f x_i(f) x_j(f)}{\sqrt{\sum_f x_i{}^2(f) \sum_f x_j^2(f)}} \tag{19}$$

A graph similarity between an image graph, G^I, and the m-th face bunch graph, B_m, is computed by searching through the stacked model graph for each node to find the best fitting jet in the bundle that maximizes the jet similarity function. The level-1 classification enables us to confine the graph to be rigid. Only the magnitude similarity is exploited. The average response over all the nodes is used as the overall graph similarity.

$$S_B(G^I, B) = \frac{1}{N} \sum_n \max_m (S_x(J_n^I, J_n^{B_m})) \tag{20}$$

3.2.2.3 Template Matching

In second-level pose classification, we attempt to classify 9 neighboring poses that are strikingly similar. The idea behind this step is simple. For each of the 9 poses, a model bunch graph is constructed from the 15 training images of

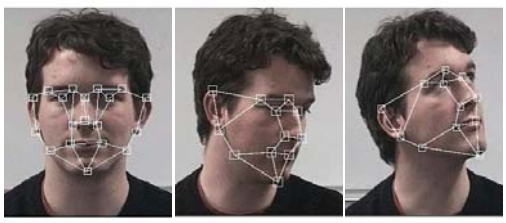

Figure 5. Examples of the elastic bunch graph

the same pose. The similarity between a test image and all the 9 model templates are computed and the model that gives the highest similarity response is declared a match.

4 Experimental valuations and analysis

Experimental results from both levels are discussed individually.

if the pose estimation falls out of the $N \times N$ sub-window around its true value, it is determined as falsely classified. In our implement N=3 is used. Bigger N gives better accuracy, however, the localization ability is weaker, which will cause more difficulty for the second level refinement.

In Fig. 6, the errors from PCA and KDA are shown respectively. In these plots, each block represents the $N \times N$ sub-window around the true pose. The color shows the number

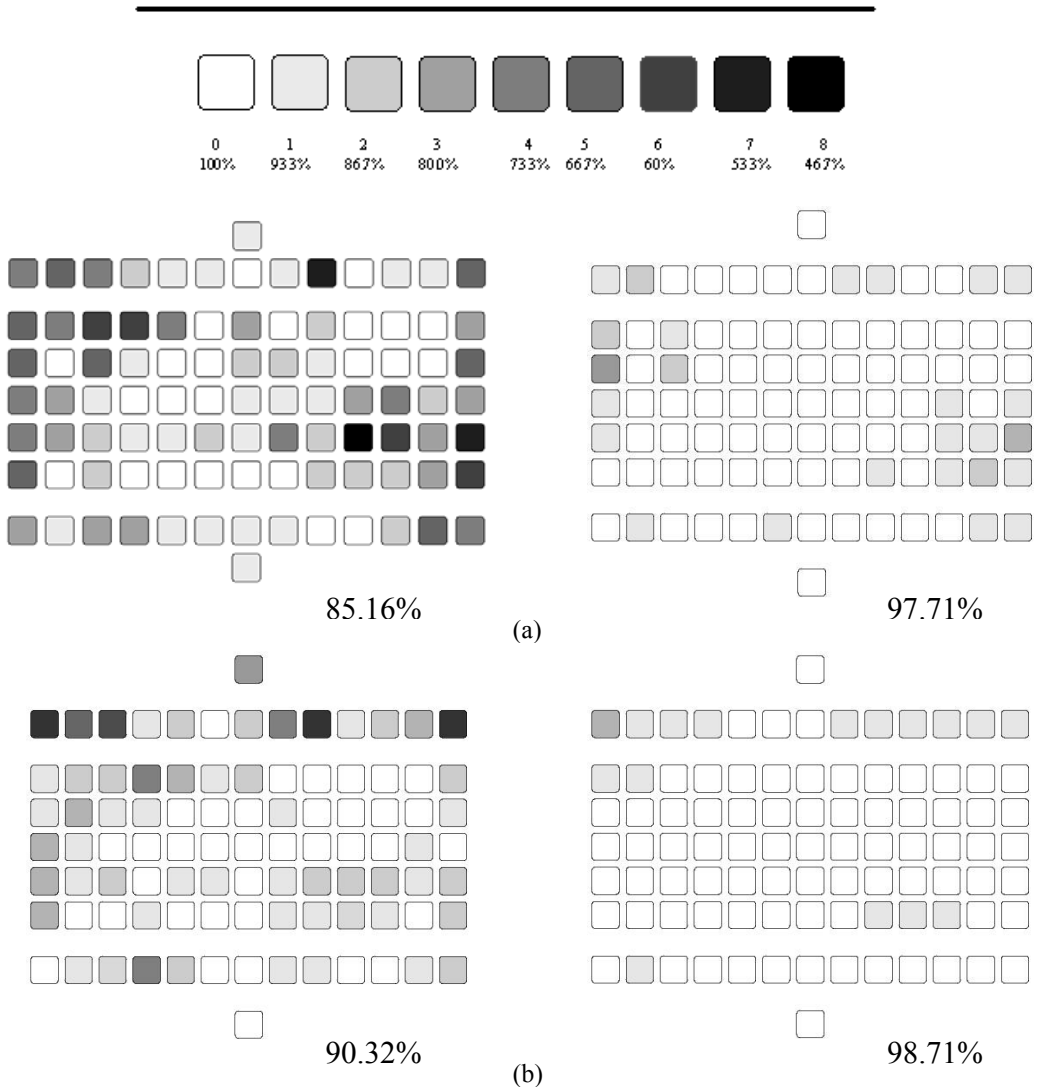

Figure 6. Results evaluated for the first level classification in PCA and KDA subspaces. The top column image gives the legend. The middle co1umn (a) gives the errors in the PCA subspaces of 48 wavelets. The left figure evaluates the localization ability up to the 3x3 sub-window around the true pose, which corresponds to ±15 degree; the right figure evaluates error on the localization ability up to 5x5 sub-window around the true pose, which corresponds to ±30 degree. The bottom column (b) gives the similar error evaluation for KDA subspace (Gaussian kernel).

4.1 Level-1 classification

The purpose of the first level is to localize the poses at the accuracy up to the $N \times N$ sub-window around the true pose. The accuracy is evaluated according to this purpose:

of the false classified samples. The left diagrams show the error rate evaluated on the 3x3 sub-windows, which means ±15 degree uncertainty. PCA subspace projection can give us a total accuracy of 85.16%. As expected, KDA can improve the accuracy to 90.61%.

To get a better understanding of how these errors distributed, we also evaluated the error on the 5x5 sub-windows, corresponding to ±30 degree uncertainty. The results are shown as the right diagrams in Fig. 6. The PCA gives a total accuracy of 97.71%, while KDA gives 98.71%. It shows that only few samples has large estimation deviation from its true value.

4.2 Level-2 refinement

Due to the labor-intensive step in generating templates for each pose, the landmark based pose refinement has been evaluated in the neighborhood of only a few representative poses, which are shown in Fig. 5. The refinement step works on the 3x3 sub-window located in the first level. Fig. 7 gives some example for the bunch graphs in a 3x3 sub-window. Using the templates consisting of 19 nodes as shown in the leftmost image of Fig. 5, we are able to obtain 10 correct classifications for 15 testing images. For the pose shown in the middle image, we use templates consisting of 11 nodes as shown in the middle image of Fig. 5, we obtained 11 correct classifications for 15 testing images. The third set of poses we analyzed is shown in the rightmost image of Fig. 5. Eleven nodes are used in the template. In this case, we obtained 10 correct classifications for 15 test images. The final classification results are summarized in Table 1. The results show that face bunch graph template matching is a promising candidate for the level-2 refinement.

Table 1. Second-level refinement on some representative poses

	Number of Nodes in Template	% Accuracy
Pose 46 (Pan 0 degree; tilt 0 degree)	19	66.7
Pose 16 (Pan –60 degree; tilt –30 degree)	11	73.3
Pose 68 (Pan –60 degree; tilt +30 degree)	11	66.7

In analyzing the errors made by the template matching classifier, we encounter a few misclassification errors that arise from the use of templates with inadequate structural details to distinguish between similar poses. In an example shown below in Fig. 8, the green nodes corresponds to the correct template being matched to the correct pose, while the red nodes corresponds to the wrong template being matched. However, the indicated by the red markers yielded a higher similarity response and thus is declared a correct match. The inadequacy of discriminant structural details could be fixed by adding more nodes and edges to constrain the templates. In the current pose shown below, a few extra nodes around the right eye and eyebrow of the subject will help constrain the structure of the template and allow for matching to be more accurate. However, further investigation of this idea must be carried out.

Several misclassification errors result from the inherent ambiguity prevalent in both the training and the testing images. As seen in the Fig. 9 below, in the leftmost image pair, pose 16 (upper image) is misclassified as pose 28 shown in the lower image. However, these 2 poses are supposed to be different by 15 degrees in the pan and tilt direction, which obviously is not the case. In the right im-

Figure 7. Examples of face bunch graphs for the 3x3 sub-windows to be examined in the second level. Top 3 rows: sub-window around pose 68; middle 3 rows: sub-window around pose 46; bottom 3 rows: sub-window around pose 16.

age pair, pose 46 (upper image) is misclassified as pose 45 as shown in the lower image. Again, the 15-degree angle difference is not apparent.

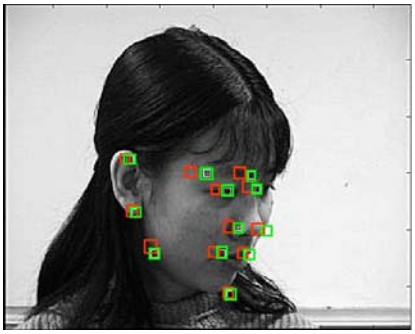

Figure.8 Example of error from inadequate nodes

Also the cropping procedure is important. In the experiment, we noticed that the images with large estimation deviation from the first level classification are mostly from subject 11th with high tilt angles (looking up). After carefully comparison of the image set, we found that the 11th subject seems to be cropped too closely. It is suspected that missed chin is the reason of the high error rate for this subject. Figure 10 gives the example of the 11th subject compared with other subjects.

Figure. 9 Example of the ambiguity in data.

Figure 10. Subject 11th with high tilt angle compared with some other subjects

5 Conclusion and discussions

In this paper we discussed a two-level approach for estimating face pose from a single static image. The rationale for this approach is the observation that visual cues characterizing facial pose has unique multi-resolution spatial frequency and structural signatures. For effective extraction of such signatures, we use Gabor wavelets as basic features. For systematic analysis of the finer structural details associated with facial features, we employ rigid bunch graphs. The first level of the approach has the objective of confining the estimation into a smaller range; therefore rigid bunch graph is sufficient in the second level refinement. Bunch graph exploits the structural details in the facial features, which makes it capable for pose location refinement Extensive series of experiments were conducted to evaluate the pose estimation approach. Using only a single level, 90% accuracy (within ±15 degree) was achieved on the complete dataset of 1,395 images. Second level classification was evaluated for three sets of poses with accuracies ranging between 67-73%, without any uncertainty. Having verified the basic efficacy of the proposed approach, further research for improving the computational performance and for evaluation using data sets with more precise ground truth information is desired.

Acknowledgements

Our research was supported in part by grants from the UC Discovery Program and the Technical Support Working Group of the US Department of Defense. We are thankful for the guidance of and interactions with our colleagues Dr. Doug Fidaleo, Joel McCall and Kohsia Huang and, Shinko Cheng from the CVRR Laboratory. We also thank Professor Thomas Moselund of the Aalborg University for his encouragement and support. Finally, we thank the organizers of the Pointing'04 Workshop and the PRIMA group of INRIA for providing the dataset used in our evaluation.

References

[1]. Y Li, S Gong and H Liddell. Recognising Trajectories of Facial Identities Using Kernel Discriminant Analysis, In *Proceedings of The British Machine Vision Conference*, 2001

[2]. M. Potzsch, N. Kruger and C. von der Malsburg. Determination of face position and pose with a learned representation based on labeled graphs. Institut for Neuroinformatik. *Internal Report,* RuhrUniversitat, Bochum, 1996.

[3]. L. Wiskott, J. Fellous, N. Krüger 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, Kie

[4]. K. Huang and M. Trivedi, Video arrays for real-time tracking of person, head, and face in an intelligent room. *Machine Vision and Applications*, vol. 14, no. 2, pp. 103-111, June 2003.

[5]. K. Huang and M. Trivedi, Robust Real-Time Detection, Tracking, and Pose Estimation of Faces in Video Streams, In *Proceedings of International Conference on Pattern Recognition* 2004, (*to appear*).

[6]. M. Trivedi, K. Huang, and I. Mikic, Dynamic Context Capture and Distributed Video Arrays for Intelligent Spaces, *IEEE Transactions on Systems, Man, and Cybernetics, special issue on Ambient Intelligence.* (*To appear* in July 2004)

[7]. Crowley, J., Coutaz, J., Berard, F., *Things That See*. *Communications of the ACM*, March 2000, p. 54-64

[8]. K. Nickel and R. Stiefelhagen. Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. *Proceedings of the 5th international conference on Multimodal interfaces.* 2003

[9]. K Seo, I. Cohen, S. You and U. Neumann. Face pose estimation system by combining hybrid ICA-SVM learning and re-registration, In *Proceedings of Asian Conference on Computer Vision (ACCV)*, Jeju, Korea, Jan. 27-30 2004.

[10]. M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):322--336, 2000

[11]. K. Huang, M. M. Trivedi, Distributed Video Arrays for Tracking, Human Identification, and Activity Analysis, *Proceedings of the 4th IEEE International Conference on Multimedia and Expo, Baltimore, MD,* pp. 9-12, July 6-9, 2003

[12]. K. Huang, M. M. Trivedi, T. Gandhi, "Driver's View and Vehicle Surround Estimation using Omnidirectional Video Stream," *Proc. IEEE Intelligent Vehicles Symposium, Columbus, OH,* pp. 444-449, June 9-11, 2003

[13]. M. Xu and T. Akatsuka. Detecting head pose from stereo image sequences for active face recognition. In *Proceedings of Int. Conf. on Automatic Face and Gesture Recognition*, pages 82-87, Nara, Japan, April 14-16, 1998.

[14**].** S. Z. Li, H. Zhang, X. Peng, X. Hou and Q. Cheng Multi-View Face Pose Estimation Based on Supervised ISA Learning. In *Proceedings of Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, May 20 - 21, 2002

[15]. S. Gong, S. McKenna, and J.J. Collins. An investigation into face pose distributions. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 265-270, Vermont, USA, October 1996

[16]. L. Chen, L. Zhang, Y. Hu, M. Li, H. Zhang, Head Pose Estimation Using Fisher Manifold Learning, in *Proceeding of IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, in conjunction with ICCV-2003, Nice, France, 2003

[17]. R.-L. Hsu, M.~Abdel-Mottaleb, and A.K. Jain. Face detection in color images. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on, .* 24(5):696--706, May. 2002

[18]. B. J.MacLennan. Gabor representations ofspatiotemporal visual images, *Technical Report CS-91-144*, Computer Science Department, University of Tennessee, Knoxville. Accessible via URL http://www.cs.utk.edu/~mclennan., 1991.

[19]. P.Viola and M.Jones. Robust real-time object detection. In *ICCV 2001. Workshop on Statistical and Computation Theories of Vision.* Volume 2, 2001.