FACE POSE ESTIMATION SYSTEM BY COMBINING HYBRID ICA-SVM LEARNING AND RE-REGISTRATION TECHNIQUE

K. Seo, I. Cohen, S. You and U. Neumann

ABSTRACT

We presents automatic face pose estimation system combined with real-time face detection and tracking technique. Foreground region in each frame from a video device is extracted by simple background subtraction method. Among the regions, candidate regions of faces are estimated by sparse run-length coding based analysis. Real-time detection System based on hybrid ICA-SVM can select the face region among the candidate regions. The pose estimation method by optimizing perspective projection of head cylinder model is applied to the detected faces. By combination of face detection and tracking technique with motion estimation algorithm, we show the more stable system applicable to the perceptual user interface system. The proposed system produces head pose information at the interactive rate (6 $\sim$ 15Hz) from video data acquisition to model visualization.

1. INTRODUCTION

Three dimensional head pose estimation technique has been highlighted as a good alternative user interface. Instead of keyboard or mouse, information like expression, pose, and gaze from face has been considered as a good candidate for user interaction with computer.

1.1. Related work

To construct the pose estimation system that can be applied to the user interface system, three key techniques such as face detection, tracking region, and pose estimation should be combined. In this paper, we presents automatic pose estimation system using real-time face detection and tracking and pose estimation algorithm.

First element necessary for proposed system is face detection technique. Especially, real-time performance is required for the live video processing. Among many kinds of techniques, feature-based approaches(ex. color, and edges) are suitable for real-time face detection system [1][2][3]. But the instability and the unpredictability of the system is the problem. Model-based approach like snake, deformable template is more robust but need initialization for first[4]. Otherwise, image-based approach is robust but still have time consuming process so it is not suitable for real-time in general. Recently, face recognition techniques have been applied to detect faces by checking if the selected region is face or not. Several pattern recognition and learning techniques like Neural Networks[5], PCA(Principal Component Analysis)[6], Subspace method[7], Kullback information[8] have been applied for face detection.

Recently, Support Vector Machine has been highlighted for its robustness and efficiency for face detection[9]. Heisele et al[10] are propose a group of SVM classifiers for processing of each face component. Nowadays, Feature reduction and boosting algorithm are used for increasing process speed of face detection [11][12].

There are also mixture method of several approaches of face detection. After obtaining the candidate region of using feature-based method, robust image-based method can finally determine and adjust the region of face. In this paper, we implement the face detection method using support vector machine and instead of using raw face image as input, we uses the extracted feature from the Independent Component Analysis(ICA)[13][14].

The position of the detected face is used for initial information to the pose estimation algorithm. Basu et al[15] used 3D ellipsoidal model to extract rigid motion by calculating optical flow information. Cascia [16] used texture map onto the cylinder model by registration between consecutive frames. Black and Yacoob invented optical flow based regularization method to extract parameters of 2D planar model[17]. To the contrary, the facial pose is estimated by tracking the feature such as nose, eyes, and mouth[18][19].

1.2. Our Approach

In this paper, we propose the robust head pose estimation and tracking method applicable to user interaction system. Three key techniques such as face detection, face tracking, and face pose recovery method are combined to realize the robust head pose estimation system. We present a real-time face detection system by combining estimation of candidate region and face classification technique. To recover the head motion parameters, we present a simplified cylindrical model-based head motion recovery method inspired by Kanade's method[20]. Based on these techniques, we propose the robust head pose estimation system.

The organization of this paper is as follows. Section 2 describes how to collect the candidate regions from the subtracted region and how to track the regions after success of detection. In the section3, we explain the face detection method using hybrid statistical learning method. Section 4 discusses about the simplified head pose estimation method based on cylinder model. In the section 5, We propose the automatic head pose estimation system combining three key techniques. In the section 6, we shows the results and performance of our proposed system. Finally we conclude this paper in the section7.

2. ESTIMATION OF HEAD POSITION USING SPARSE RUN-LENGTH STRUCTURE

Instead of whole area of input image, collected regions are tried to apply the algorithm for fast processing. We extract the connected information from foreground regions as a result of background subtraction. Under the fixed exposure condition of the camera, we get a foreground image by differentiate the current frame from a reference frame including only background.

For the foreground pixels in the image, we encode the run-length This section is a description of how to calculate the estimated size of the head and choose the region of candidate for face detection. In order to detect faces of various sizes, we propose a simple estimation technique based on the segmented foreground. Commonly, a multi-resolution strategy is used. It consists of searching for faces in various levels of the pyramid. Is is robust and accurate but not suited to real-time application. In this paper we propose to infer the size and image location of the face from the segmented foreground regions.

At first, foreground pixels are encoded to horizontal run-length for foreground region. While we scan the horizontal line in each frame, connected pixels are grouped. Let us denote the run-length as R_i with index i. Each run-length structure includes properties like start position in x direction $\sigma(R_i)$, end position in x direction $\varepsilon(R_i)$, and y position $\phi(R_i)$. For fast processing, instead of constructing run-length at every scan line, we make a sparse run-length information by skipping specified interval pixels.

For estimating size, we group horizontal run-lengths into connected components by considering the distance between run-lengths. The run-lengths with close distance are clustered into same connected component group. We denote C_k as a connected component which is grouped according to two distance criteria as denoted by Equation (1).

$$C_k = \{R_l, R_m \in C_k \| d_1(R_l, R_m) < \lambda_1, d_2(R_1, R_m) < \lambda_2\} \tag{1}$$

where $d_1(R_l, R_m)$ is the height (y direction) distance function for judging the proximity and $d_2(R_1, R_m)$ is the percentage of difference between sizes of two run-lengths. By these two distance value, all run-length structure are clustered into the unit of connected component.

For each connected component, we calculate position of left($l(C_k)$)), right($r(C_k)$), top($t(C_k)$), bottom($b(C_k)$) as shown below in equation (2) :

$$l(C_k) = \frac{1}{n}\sum_{l=1}^{n} \sigma(C_k^l) \tag{2}$$

$$r(C_k) = \frac{1}{n}\sum_{l=1}^{n} \varepsilon(C_k^l) \tag{3}$$

$$t(C_k) = max(\phi(C_k)) \tag{4}$$

$$b(C_k) = min(\phi(C_k)) \tag{5}$$

As you can see, the position of top and bottom are determined by choosing uppermost and lowermost of run-length. The position of left and right is done by averaging all of left and right position to overcome noise.

For each frame, we calculate the four boundary values of all connected components as shown above and try to collect the possible candidates. By checking ratio of width and height of the candidate, we ignore the component with too big and small ratio because we have anatomical knowledge that most human has a width to height ratio close to 1 or 1.2. We also discard small connected components. These collected components are used to initialize the search region for face classificaion and tracking.

Whenever the face recognizer succeeds in finding face regions, boundary information of face region and corresponded estimated head boundary information are stored and combined. If there is no detected result in previous frame, the estimated region is given to face recognizer as initial size and position. Within chosen boundary from the estimated region, our proposed algorithm detects face regions. In case of successful detection, we get the positions of estimated regions and detect regions for current frame. These two positions information are combined with relation of scaling and translation as formalized in the following equation (6).

$$D_c = \mathbf{S_1} E_c + \mathbf{T_1} \tag{6}$$

D_c shows the position of detected region and E_c is the position of estimated region in current frame. $\mathbf{S}$ and $\mathbf{T}$ means the transformation matrix of scaling and translation.

The characterization of the relationship between estimation and detection allows us to reuse detection results between without iterative usage of recognition module. If there exists a detected region in the previous frame, we check the amount of overlap between estimated regions of current frame and previous frame. For highly overlapped regions, instead of applying recognition process, we update the detected result using geometric relations between overlapped regions.

First, we calculate scaling and translation factor using the geometric relation between estimation results of previous and current frame as described in equation (7).

$$E_c = \mathbf{S_2} E_p + \mathbf{T_2} \tag{7}$$

where E_p is the position of estimated region in previous frame. If we assume that there is not much transformation between highly overlapped estimated regions in consecutive frames, we directly estimate the detected region using just geometric relations with previous frame without trying to use face recognition module. As shown in equation (8), we update detection result $\widetilde{D}_c$ by substituting equation (7) for E_c.

$$\widetilde{D}_c = \mathbf{S_1}(\mathbf{S_2} E_p + \mathbf{T_2}) + \mathbf{T_1} \tag{8}$$

The updated result $\widetilde{D}_c$ can replace the final detection result of current frame, D_c by the assumption above. In comparison to recognition module, the estimation and update algorithm has lower computation complexity so we can expect faster speed of our system.

3. FACE DETECTION BASED ON HYBRID ICA-SVM METHOD

Real-time face detection system is necessary for automatic face pose estimation. In this section, we describe the construction of face detection module using hybrid ICA and SVM based learning method. Among the classification technique, SVM has been recently used for detecting faces. Also low computational complexity is convenient for real-time applications.

The core problem for SVM-type classifiers is the training component. The choice of features considered dictates the expected performances of the classification. Raw images can be considered as feature vectors but the capabilities of the classification in detecting faces of different complexion or in scene of variable illuminations will be very limited. Instead of using directly raw images as an input vector for training the SVM, an ICA representation of the images is considered [21]. It allows to reduce the dimension of the data as well as increasing the performance of the classification by relying on statistical features of the grey level distributions.

ICA algorithm is to estimate matrix W when we assume a linear mixture model(equation (9) and a reconstruction model equation (10).

$$X = As \tag{9}$$

where s are independent unknown sources X is observed data.

$$U = WX \tag{10}$$

X is the input vector which each column vector is one face image is constructed by database images. Because the size of matrix X is big, we use m eigenvectors of X denoted by P_m. and then we apply the ICA algorithm on P_m instead of X as follows:

$$WP_m^T = U \tag{11}$$

$$P_m^T = W^{-1}U \tag{12}$$

X can be approximated as shown in equation (13).

$$X_{rec} = XP_mP_m^T \tag{13}$$

By Replacing P_m^T with equation (12), we can get equation (14).

$$X_{rec} = XP_mW^{-1}U \tag{14}$$

Using equation (14), ICA representation for each test image can be calculated as follows:

$$c = IP_mW^{-1} \tag{15}$$

where I is $1 \times N$ row vector of a test image and P_mW^{-1} is calculated during ICA denoted by K

We apply ICA algorithm twice for face images and non-face images and then we get a pair of ICA feature per a test image as shown in equation (16) to build a SVM training vector.

$$c_f = IK_f \tag{16}$$

$$c_{nf} = IK_{nf} \tag{17}$$

We combine two feature vectors into one training vector with class label as follows:

$$\tau_{i=1}^l = (c_f^i, c_{nf}^i, d^i)_{i=1}^l \tag{18}$$

where $d_i = 1, -1$ is the class label which 1 means the face class and -1 is non-face class and l is the number of training images. $\tau_{i=1}^l$ is used to construct SVM. After extracting support vector by execution of SVM algorithm, we can classify whether the image is face or not by calculating the distance value as following format:

$$f(x) = sign(\sum_{i=1}^{l} y_i \lambda_i K(x, x_i) + b) \tag{19}$$

The constructed recognizer is applied within specified boundary from the estimated region. The algorithm detects whether there exists a face or not in the considered region.

4. HEAD MOTION ESTIMATION AND TRACKING

Detected faces in the input image are used as an initialization of pose estimation since frontal faces are recognized and tracked. To recover the motion information of head we adopt a modified re-registration technique based on Kanade's method [20]. The shape of the head is assumed as a cylinder model. 3D points on the surface of the cylinder model are mapped onto pixels of the captured image by perspective projection. The transformation parameter of each pixel between consecutive frames is calculated by Lucas-Kanade registration method[22].

The perspective transformation matrix is recovered by optimizing the objective function as following equation.

$$minE(\mu) = \sum_{u \in \Omega} (I(F(u,\mu), t+1) - I(u,t)^2 \qquad (20)$$

where $I(u,t)$ means an image at time t with pixels u and $F(u,\mu)$ is parameter motion model using twist representation of rigid motion as shown by Bregler[23].

The Lucas-Kanade solution including weight compensation process can be formulated as:

$$\mu = -(\sum_{\Omega}(w(I_u F_\mu)^T (I_u F_\mu)))^{-1} \sum_{\Omega}(w(I_t(I_u F_\mu)^T)) \qquad (21)$$

where I_t and I_u means image gradient. The weights w are updated in the direction of reducing the effects of outliers and increasing importance of the center of the head cylinder model.

The modified part of the original method is as follows. Firstly, we add the re-initialization process while the algorithm executes re-registration process. Whenever tracking of head position is failed or overlapped region between current template and warped image of last template is under the specified percentage, the head pose estimation process is paused and try to detect frontal face for re-initializing the head pose estimation algorithm. Secondly, the abrupt change between consecutive frames can cause the accumulation of errors that cannot be compensated by weight calculation. We forcibly adjust the translation parameter among the rigid body motion parameters by considering the displacement of head position value tracked by face detection and tracking method.

Though these modified methods don't produce exact compensation in the domain of 3D space, they are simply useful for the purpose of user interaction and enable the propose system to guarantee more long-term robustness.

5. AUTOMATIC HEAD POSE ESTIMATION SYSTEM

With three techniques presented in previous section, we propose an automatic pose estimation and tracking system from a live video. The proposed system is composed of face detection, face position estimation, and head pose estimation and tracking module.

At first, background subtraction enables to classify each pixel is foreground or background. Pixels that corresponded difference value between current frame and reference background frame is over specified threshold value are considered as foreground region. In the foreground pixels, we construct sparse run-length structure and group adjacent runs into connected components. By ignoring the connected components that have extraordinary ratio of height to width, regions including remained components are regarded as candidate region to apply face detection algorithm. Face detection algorithm using hybrid statistical learning method is applied to the candidate region. If the detection algorithm succeed in classifying the region as face, the position of the classified region is used to initialize the head pose recovery algorithm. The extracted motion information is used for interacting with computer.

While tracking the pose, the proposed system checks the validity of the motion parameter. Whenever the system detects the invalid motion parameter value, face detection algorithm is executed for reset of head motion algorithm. In this paper, we decided the case that the rotation value is not inside -70 degree and 70 degree in the direction of panning and yawing. The modified pose estimation algorithm is not robust in case of abrupt change of position ,i.e., big translation. So, we use the displacement value as a result of tracking face region to compensate the pose tracking value about translation. If the translation value by pose recovery algorithm is much smaller than the displacement of face region between consecutive frames, we modify the translation value by adding displacement value.

6. EXPERIMENTS

In this section, we present how to implement our proposed system and discuss about results. Prior to execute the proposed system, two preparation processes are necessary. Firstly, to construct real-time face detection algorithm, we need statistical learning process. We use total 3000 images (Faces : 2000, Non-faces : 1000) for training. 2000 face images are collected from FERET face database[24]. After specifying reference feature points such as 4 eye corners, 1 tip of nose, and 2 mouse corners, all face images are aligned by affine transformation. For non-face images, we take tens of background pictures including both outdoor and indoor scenes using a digital camera. 1000 images are generated by selecting regions randomly and cropping the part of image with different size. Both face and non-face images are also histogram equalized and the boundary of each image is filled with black called oval masking to reduce the ambiguity from the boundary of face.

We separately make ICA kernel for the prepared face and non-face images. Figure (1) shows the basis images by result of ICA algorithm. 24×24 resolution of images are gathered into one matrix as a one image vectors. We make a matrix including 64 eigenvectors from the gathered matrix and apply ICA on this eigenvector matrix. The feature extracted by ICA is labeled with +1 or -1 to train the SVM.

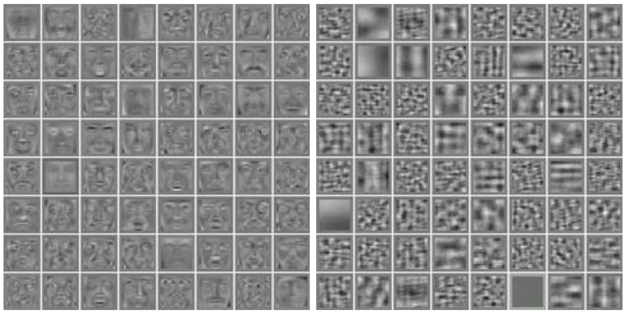

Fig. 1. 64 ICA Basis Images for 2000 Face Images and 1000 Non-Face Images

As a result of SVM training, we can get 384 support vectors from 3000 ICA-based feature vectors for each image. Later, these support vectors are used to classify if the class of the arbitrary image is belong to the face class or not.

Secondly, to obtain absolute pose and position of head from the camera, we need accurate calibration process to get the reference information between captured image and real geometry. Instead of using standard calibration algorithm, we use simple method to get the position of head by relationship between size and distance from camera. We set the head size is approximately same so the we can get the distance from camera by getting the size of head. At first, we capture the paper including the black rectangle with some amount of size (4cm) with several different distances. Using this information, we can infer the relationship between actual pixel size, actual size and distance. Under the assumption that head size is fixed, we can calculate approximate distance from the camera.

The proposed system applies background subtraction from video input. For this segmented image, estimation, tracking, and detection process are executed. The detected and estimated results for the single and multiple person are shown in figure 2. The white rectangle shows the estimation position and blue one describes the final detected position. Face recognition module searches the region around estimated region and tries to find the exact position of face.

The processing speed of face detection algorithm including data acquisition and visualization of results is about 40Hz on average. When the system uses the face recognition module,most time consuming part, the speed of all the processing is above 25Hz. In most cases in the video sequence, after succeeding in detecting face region, the system tracks the face region instead. The process speed of the case of tracking is above 50Hz. When more people is coming inside the view of video, more processing time is consumed for multiple usage of face recognition module. In our experiment for two people, the speed is above 22Hz on average. Table 1 shows the processing speed of each part in the proposed system.

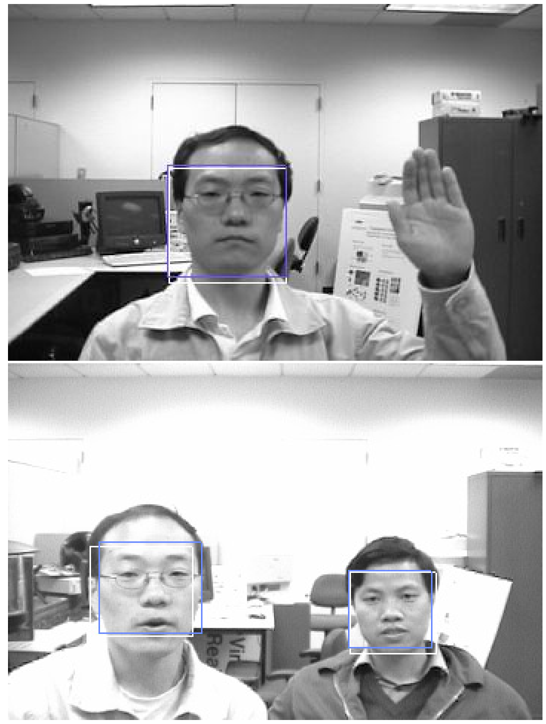

Fig. 2. The estimation and detection results

In order to evaluate the detection rate of the proposed algorithm, the off-line test using the captured frames is done. At first, we record video of 6 different single persons and 3 pairs of people including different kinds of movements. Each set consists of 396 450 frames. We classify the captured sets of video into 4 classes such as normal, rapid, small rotation, and large rotation. Normal means that the movement in the video is smooth and it has no special movement. Rapid is a footage including fast movement of head. Small rotation means that the video data contains the normal rotation including panning and large means that the footage has the movement of large rotation including panning , tilting, and yawning. Table 2 shows the accuracy of experiment for single person. Especially, in the case of large rotation, the algorithm failed more frequently than any other cases. We defined the metrics to quantify the performance of our system. The detection rate (DR) and the false alarming rate (FAR) are based on the scalars such as

- True Positive (TP) : number of detected regions that correspond to faces,
- False Positive (FP) : number of detected regions that do not correspond to faces,
- False Negative (FN) : number of faces not detected.

These numbers are used for generating rates using $DR = \frac{TP}{TP+FN}$ and $FAR = \frac{FP}{TP+FP}$.

The other metrics, the estimation rate and the corresponding false alarm rate, are generated with similar scalars for the estimation results.

	Normal	Small R	Big R	Fast	Total	
Estimation	ER	93.94 %	91.17 %	80.42 %	89.47 %	91.40 %
	FAR	2.42 %	5.15 %	9.06 %	4.28 %	3.77 %
Detection	DR	85.66 %	85.98 %	73.15 %	88.26 %	81.26 %
	FAR	4.34 %	3.68 %	16.15 %	3.29 %	7.58 %

(ER : Estimation Rate, DR : Detection Rate, FAR: False Alarming Rate)

Table 1. Performance result for single person

We also get the accuracy for the multiple person situation. We recorded videos with 3 different types of movements. As you can see in table 3, We classify the video data into overlapped movements between two persons, fast movement, and rotation movement. Complex situations between two people cause lower estimation rate and detection rate.

		Overlap	Rotation	Fast	Total
Estimation	ER	33.32 %	84.80 %	85.12 %	85.57 %
	FAR	2.10 %	4.38 %	5.32 %	3.74 %
Detection	DR	23.73 %	39.81 %	78.54 %	65.21 %
	FAR	7.24 %	5.62 %	4.52 %	5.51 %

(ER : Estimation Rate, DR : Detection Rate, FAR: False Alarming Rate)

Table 2. Performance result for multiple person

We apply the head motion recovery algorithm to the detected face region obtained by real-time detection algorithm. The average processing speed from video data acquisition to visualization of approximate position and direction of head shows from 8Hz to 15hz. The varied speed of the system depends on iteration number of the head motion recovery algorithm and whether the detection algorithm is executed or not during whole system are executed.

The calibration process enable the system to obtain the absolute position and orientation from the installed camera. Figure 3 shows that 3D cylinder model are perspectively projected onto the face region in the captured image and the model is appropriately transformed and fitted to the change of face position and orientation. Three images of right columns in the figure 3 shows the simple 3D model visualization of absolute position and orientation from the monitor on which the camera is attached. Whenever face pose are estimated, the visualization results represent which points of the monitor the person are looking at. In many experiments, we verify the system is working properly.

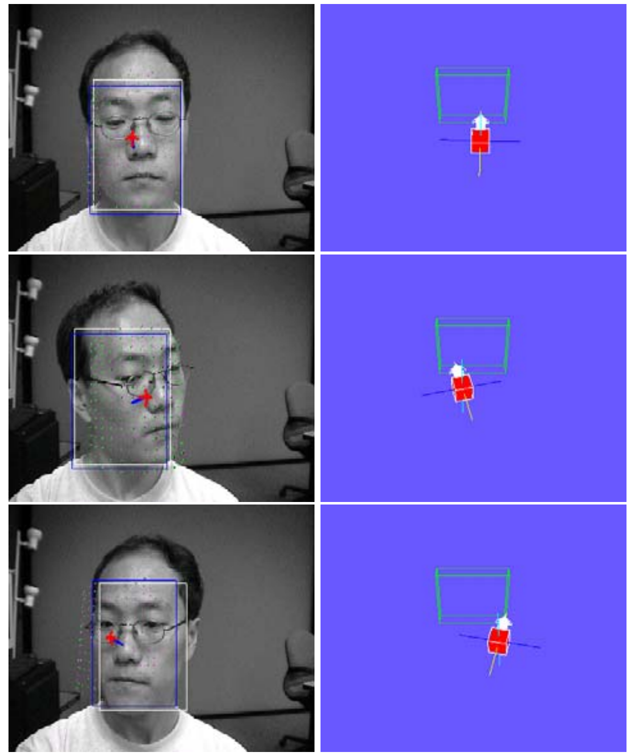

Fig. 3. Pose estimation and visualization

7. CONCLUSIONS AND FUTURE WORKS

In this paper, we have described automatic head pose estimation system working at the interactive rate(8Hz $\sim$ 16Hz). Real-time face detection algorithm using statistical learning method and estimation technique of head candidate contribute to make the system fully automatic. Moreover, face region tracking method for the detected region compensates weakness of the system for the abrupt change of position so increases the robustness of the system and enables to endure long-term processing. By modification of Kanade's pose recovery, we obtain accurate head motion parameter information. Three techniques such as face detection, the estimation and tracking of face region, and head pose recovery algorithm are well combined to realize robust and fast system that is applicable to user interface.

In the future, improved face detection algorithm that find multiple person under less constraints enables to multiple head pose estimation. Now, we uses simple calibration method. instead, more accurate calibration will produce more accurate position from the camera and obtain higher resolution result of interaction from head movement. Tracking technique also can overcome the limit of our system. Probabilistic approach like [25] contains the possibility of choosing the alternative position so the propagation of error is decreased.

The system presented in this paper can be used for the extension of many applications related to face processing. First of all, facial expression analysis can be added to this face detection system and online facial expression analysis and recognition system can be an important application. Also, face feature detection and tracking is strongly related to this system. We can use the detected region as a important input to facial expression analysis.

8. REFERENCES

[1] Erik Hjelmas and Boon Kee Low, "Face detection : A survey," *Computer Vision and Image Understanding*, vol. 83, pp. 236–274, 2001.

[2] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Anil K. Jain, "Face detection in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , no. 5, pp. 1046–1049, 2002.

[3] John G. Daugman, "Complete discrete 2-d gabor transformation by neural networks for image analysis and compression," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, July 1998.

[4] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," *Proceedings of European Conference on Computer Vision 98*, vol. 2, pp. 484–498, June 1998.

[5] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[6] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, pp. 84–91, June 1994.

[7] Ming-Hsuan Yang and D. Abuja, N. Kriegman, "Face detection using mixtures of linear subspaces," *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 70–76, March 2000.

[8] Antonio J. Colmenarez and Thomas S. Huang, "Face detection with information-based maximum discrimination," *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition 97*, pp. 782–787, June 1997.

[9] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines:an application to face detection," *IEEE Proceeding of Computer Vision and Pattern Recognition 97*, vol. 6, pp. 130–136, 1997.

[10] Bernd Heisele, Alessandro Verri, and Tomaso Poggio, "Learning and vision machines," *Proceedings of IEEE*, vol. 90, no. 7, July 2002.

[11] Gregory Shakhnarovich, Paul Viola, and Baback Moghaddam, "A unified learning framework for real time face detection and classification," *Proceedings of the Fifthe IEEE International Conference on Automatic Face and Gesture Recognition 2002*, pp. 16–23, May 2002.

[12] Stan Z. Li, Long Zhu, ZhenQiu Zhang, Andrew Blake, HongJiang Zhang, and Harry Shum, "Statistical learning of multi-view face detection," *Proceedings of The 7th European Conference on Computer Vision*, 2002.

[13] Yuan Qi., Doermann D., and Dementhon D., "Hybrid independent component analysis and support vector machine learning scheme for face detection," *IEEE International Conference on Acoustics, Speech, and Signal Processing 2001 Proceedings.*, vol. 3, pp. 1481–1484, May 2001.

[14] Tae-Kyun Kim, Sung-Uk Lee, Jong-Ha Lee, Seok-Cheol Kee, and Sang-Ryong Kim, "Integrated approach of multiple face detection for video surveillance," *International Conference of Pattern Recognition 2002*, vol. 2, pp. 394–397, August 2002.

[15] Sumit Basu, Irfan Essa, and Alex Pentland, "Motion regularization for model-based head tracking," *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3, pp. 611–616, August 1996.

[16] Marco La Cascia, John Isidoro, and Stan Sclaroff, "Head tracking via robust registration in texture map images," *IEEE Conference on Computer Vision and Pattern Recognition,*, pp. 508–514, June 1998.

[17] Michael J. Black and Yaser Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *International Journal of Computer Vision*, vol. 25, no. 1, pp. 23–48, October 1997.

[18] Volker Kruger, Sven Bruns, and Gerald Sommer, "Efficient head pose estimation with gabor wavelet networks," *Proceedings of British Machine Vision Conference*, September 2000.

[19] Pingping Yao, Glyn Evans, and Andrew Calway, "Face tracking and pose estimation using affine motion parameters," *Proceedings of the 12th Scandinavian Conference on Image Analysis*, pp. 531–536, June 2001.

[20] Jing Xiao, Takeo Kanade, and Jeffrey F. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*, pp. 156–162, May 2002.

[21] Aapo. Hyvarinen, Patric O. Hoyer, and Mika Inki, "Topographic independent component analysis," *Neural Computation*, pp. 1527–1558, July 2001.

[22] Bruce D. Lucas and Takeo Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 121–130, 1981.

[23] Christoph Bregler and Jitendra Malik, "Tracking people with twists and exponential maps," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 8–15, June 1998.

[24] P.J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The feret database and evaluation procedure for face recognition algorithms," *Journal of Image and Vision Computing*, vol. 15, no. 5, pp. 295–306, 1998.

[25] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.