# FACE POSE ESTIMATION SYSTEM BY COMBINING HYBRID ICA-SVM LEARNING AND 3D MODELING

*Kyoungchin Seo*

Sogang University
Department of Media Technology
Seoul, Korea

*Isaac Cohen, Suya You, Ulrich Neumann*

University of Southern California
Integrated Media System Center
Los Angeles, CA, USA

## ABSTRACT

We present real-time face detection and face pose estimation and tracking technique for collaborative workspace. Foreground regions in each frame are extracted by simple background subtraction method. Among these regions, candidate regions of faces are estimated by sparse run-length coding based analysis. Real-time detection system based on hybrid ICA-SVM is then used for detecting faces among the candidate regions and tracking them over time. An estimation of the head pose of the participants identifies the focus of attention during the collaborative work. The head pose is computed by approximating the shape of the head by a 3D cylinder. 2D velocities are mapped onto the 3D cylinder for updating and tracking the pose of detected faces.

The combination of face detection and tracking technique with motion estimation algorithm, demonstrates a more stable system applicable to head pose estimation in a perceptual user interface system. The proposed system produces head pose information at the interactive rate of 10Hz.

## 1. INTRODUCTION

Face detection techniques have been extensively studied in the context of surveillance applications and biometrics. In this paper we focus on the face detection of participants while interacting with a system. The objective here is to derive situation's context for gauging the user state and focus of attention. A first step towards this challenging problem is the detection of people faces and computing the six parameters characterizing the 3D head-pose of the participants. Also, real-time performances are required as we would like to adapt the system behavior to the participants' activity.

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 present automatic pose estimation system using real-time face detection and tracking and pose estimation methods.

The first component of such system is the face detection. As we are interested in a system that captures the user state while interacting with an information system, real-time performances is a crucial component. Feature-based and model-based approaches are well suited for real-time face detection system[1]. However, the instability of selected features and manual initialization of the models limit the use of such methods in an interactive setting.

Recent machine learning approaches, face recognition techniques have been proposed and a scanning of the image for detecting if a specific region is a face or not have shown very good performances. Several pattern recognition and learning techniques [2][3][4] have been used for detecting faces in the image. Among them, Support Vector Machine(SVM) has been highlighted for its robustness and efficiency for face detection[5]. Nowadays, Feature reduction and boosting algorithm are used for increasing process speed of face detection [6][7][8].

Second element indispensable to the proposed system is the 3D head-pose estimation. The position of the detected face is used as an initial estimate of the 3D head pose. Basu *et al*[9] used 3D ellipsoidal model to extract rigid motion by calculating optical flow information. Cascia [10] used a texture map onto the cylinder model by registration between consecutive frames. Black and Yacoob used an optical flow based regularization method for extracting parameters of 2D planar model[11]. Other approaches, estimate face pose by tracking faces features such as nose, eyes, and mouth[12][13].

This paper presents a robust head pose estimation and tracking system, that allows tracking head pose for long time durations. Three techniques are combined into the proposed system. A new real-time face detection method using statistical learning method is designed to find face's position from video sequences. To boost the speed of the method, we propose a sparse run-length encoding of the the foreground regions. This regions encoding is used as input to face detection method. Using the initial estimation of face's position, 3D head pose estimation and tracking method provides the 3D head-pose of the user. We present a modified version of Kanade's head motion recovery technique[14].

Moreover, the combination of the three components allows for compensating the defects of each method and provides a robust tracking of the head pose.

The organization of this paper is as follows: Section 2 describes the search for candidate face regions from the detected foreground. In section3, we explain the face detection method using hybrid statistical learning method. The tracking of detected faces is described in section 4. Section 5 discusses the simplified 3D head-pose estimation method based on cylinder model. In the section 6, we show the results and performances of our proposed system. Section7 concludes the paper and discusses future work.

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

In this section we describe the approach used for selecting candidate regions for faces in the image. 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.

Foreground pixels are represented using a horizontal run-length encoding. While scanning 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 number of lines.
For estimating the face size, we group horizontal run-lengths into connected components by considering the distance between two run-lengths. The closest run-lengths are clustered into the same connected component. We denote $C_k$ the obtained connected component which is grouped according to two distances criteria as defined by the following equation:

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

where $d_1(R_l, R_m)$ is the distance along the $y$-axis, characterizing the proximity of $\phi(R_l)$ and $\phi(R_m)$. $d_2(R_1, R_m)$ is the ratio between sizes of the two run-lengths. Using these functions we cluster the encoded run-length structures into 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 one can see, the position of top and bottom are determined by choosing uppermost and lowermost run-length. The left and right boundaries are obtained by averaging left and right positions of the run-lenght belonging to the connected component. 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 regions, we discard small connected components, and components with too large or too small ratio. Antropometric measurements have characterized the for most humans, the width to height ratio is close to 1 or 1.2. These collected components are used to initialize the search region for face detection and tracking.

## 3. FACE DETECTION USING A HYBRID ICA-SVM METHOD

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 [15]. The ICA representation reduces the dimension of the data, and increases 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 (6) and a reconstruction model equation (7).

$$X = As \tag{6}$$

where $s$ are independent unknown sources $X$ is observed data.

$$U = WX \tag{7}$$

$X$ 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:

$$WP_m^T = U \tag{8}$$

$$P_m^T = W^{-1}U \tag{9}$$

$X$ can be approximated as shown in equation (10).

$$X_{rec} = XP_mP_m^T \tag{10}$$

By Replacing $P_m^T$ with equation (9), we can get equation (11).

$$X_{rec} = XP_mW^{-1}U \tag{11}$$

Using equation (11), ICA representation for each test image can be calculated as follows:

$$c = IP_mW^{-1} \tag{12}$$

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 on face and non-face images for training the SVM. We combine the feature vectors $c_f$ and $c_{nf}$ representing respectively faces and non faces images into one training vector with class label as follows:

$$\tau_{i=1}^l = (c_f^i, c_{nf}^i, d^i)_{i=1}^l \tag{13}$$

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. After characterizing the support support vector, we can classify whether the image is face or not by calculating the distance value as following format:

$$f(x) = sign(\sum_{i=1}^{l} y_i\lambda_iK(x, x_i) + b) \tag{14}$$

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. 2D HEAD TRACKING

Whenever the face recognizer succeeds in finding face regions, boundary information of face region and 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 (15).

$$D_c = \mathbf{S_1}E_c + \mathbf{T_1} \tag{15}$$

$D_c$ 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 in consecutive frames without iterative usage of the recognition module. For a detected region in the previous frame, we check the amount of overlap between estimated regions in current frame. If the regions have a large overlap, 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 (16).

$$E_c = \mathbf{S_2}E_p + \mathbf{T_2} \tag{16}$$

where $E_p$ is the position of estimated region in previous frame. If we assume a small transformation between highly overlapped regions in consecutive frames, we can directly estimate the detected region using just geometric relations with previous frame without trying to use face recognition module. As shown in equation (17), we update detection result $\widetilde{D}_c$ by substituting equation (16) for $E_c$.

$$\widetilde{D}_c = \mathbf{S_1}(\mathbf{S_2}E_p + \mathbf{T_2}) + \mathbf{T_1} \tag{17}$$

The updated result $\widetilde{D}_c$ can replace the final detection result in current frame, $D_c$. In comparison to recognition module, the estimation and update algorithm has lower computation complexity and allows for small variations in face orientation.

## 5. 3D HEAD POSE ESTIMATION AND TRACKING

Detected faces in the input image are used as an initialization of the pose estimation since only frontal faces are recognized by the ICA-SVM classifier. To estimate the head pose we adopt a modified re-registration technique based on Kanade's method [14]. The shape of the head is approximated by a 3D cylinder model. 3D points on the surface of the cylinder model are mapped onto pixels of the captured image by perspective projection. The motion of each pixel between consecutive frames is calculated by Lucas-Kanade registration method[16]. The perspective transformation matrix is recovered by optimizing the following objective function:

$$minE(\mu) = \sum_{u\in\Omega}(I(F(u, \mu), t + 1) - I(u, t)^2 \tag{18}$$

where $I(u, t)$ represents the image at time t with pixels $u$ and $F(u, \mu)$ is the parameter motion model using twist representation of rigid motion as shown by Bregler[17]. $\mu$ represents the 3D rigid motion of the head and is defined by the 3D head rotation and translation.

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

$$\mu = -(\sum_{\Omega}(w(I_u F_\mu)^T(I_u F_\mu)))^{-1}\sum_{\Omega}(w(I_t(I_u F_\mu)^T)) \tag{19}$$

where $I_t$ and $I_u$ represents temporal and spatial image gradients. The weights $w$ are updated for reducing the effects of outliers and for emphasizing the center of the head cylinder model.

The proposed approach combines the face detection method with the head-pose estimation. Firstly, we add the re-initialization process while the algorithm executes re-registration process. Whenever tracking of head position fail or overlapping regions between current template and previous template is too small, the head pose estimation process is stopped and a new face detection is initiated for re-initializing the head pose algorithm. Secondly, abrupt changes 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.

## 6. EXPERIMENTS

In this section we present some experimental results obtained, as well as some an evaluation of the performances of the proposed approach.

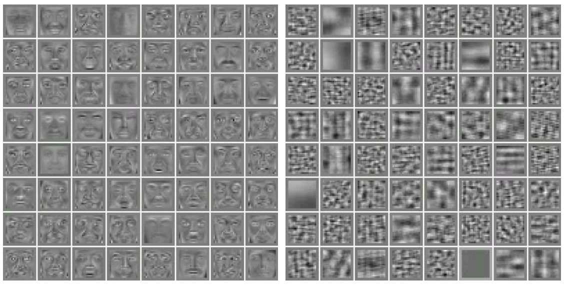

**Fig. 1**. 64 ICA Basis Images for 2000 Face Images and 1000 Non-Face Images

We use a total of 3000 images (Faces : 2000, Non-faces : 1000) for training. 2000 face images are collected from FERET face database[18]. We have separately built ICA kernels for the face and non-face images set. Figure (1) shows the basis images obtained by the ICA algorithm. $24 \times 24$ 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 training the SVM classifier provided 384 support vectors that are used for face detection. 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 represents the final detected face.

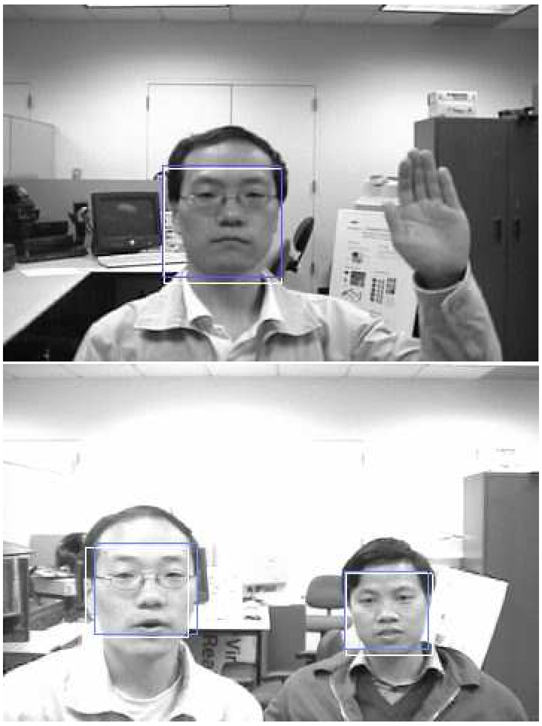

**Fig. 2**. Real-time face detection. The white rectangle depicts the estimated face locations and the blue rectangle depicts the detected faces.

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 to 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 the 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 and tilting. Table 2 shows the accuracy of experiment for single person. In the case of a 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 alarm rate (FAR) are based on the following scalars:1

- 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 Alarm Rate)

**Table 2**. Performance result for multiple persons

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 is 10Hz.

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 changes of face's 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.

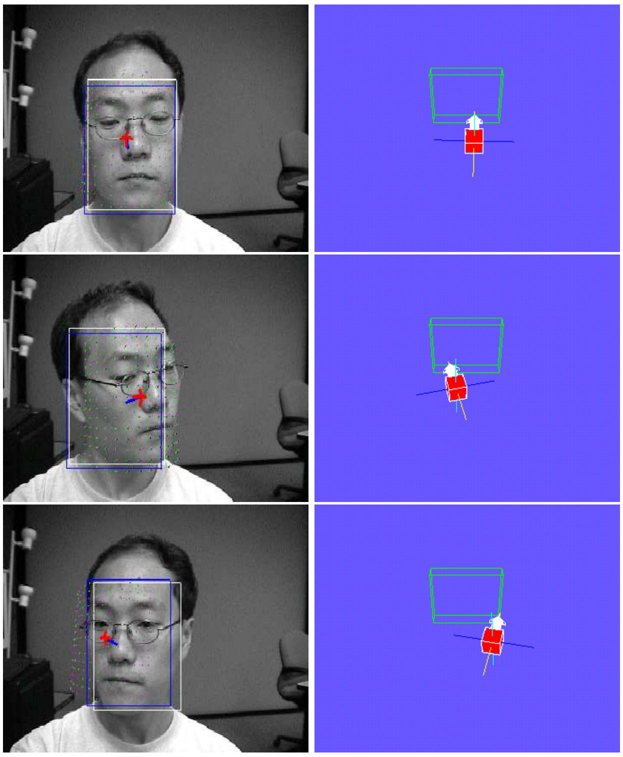

**Fig. 3**. Pose estimation and visualization

## 7. CONCLUSIONS AND FUTURE WORKS

In this paper, we have described an automatic head pose estimation system working at the interactive rate (10Hz). 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 when abrupt changes of face's position and orientation. The modified head-pose estimation algorithm, provides accurate head motion parameter information. The proposed approach for face detection and 3D head pose estimation are well integrated allowing an automatic and robust head-pose estimation for user interface.

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.

## Acknowledgments

This work is achieved during the first author's visiting at the Computer Graphics and Immersive Technologies (CGIT) Laboratory of the University of Southern California. We acknowledge support from the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

### 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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.