Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

STEREO VISION and STATISTICAL BASED BEHAVIOUR PREDICTION OF DRIVER

Haluk Eren, Ümit Çelik, Mustafa Poyraz

Abstract— *Abstract*—**The goal of this project is to develop a webcam-based system for monitoring the activities of automobile drivers. As in any system deployed for monitoring driver activities, the primary goal is to distinguish between safe and unsafe driving actions. There is no fixed list of actions that qualifies the unsafe driving behaviors. In general, an activity or an action that reduces a driver's alertness of their surroundings should be classified as unsafe driving behavior. Some examples of unsafe driving behavior include fatigue, talking on a cellular telephone, eating, and adjusting the controls of the dashboard stereo while driving. In this study, we also investigated the relationship between 2D and 3D face and pose recognition.**

I. INTRODUCTION

In this study in order to process later, we try to extract and recognize the driver's face image grabbed in 3D. As it is known, the first stage in image understanding is generally segmentation. Otherwise it is difficult to extract any knowledge using the raw image data. We have used two same webcams to acquire 3d data and to produce stereo image pairs [8]. In the first sight, it can be thought that only one webcam is feasible for this type of application. Nowadays, researches on 3D human machine interfaces are getting more popular [10]. If more robust algorithms can be designed, these types of studies will lead researchers to be opened new horizons and provided them new research topics. The outdoor environment is visible through the car's windows, but driver behaviors have still uncertainty. To address this problem, we only extract the disparity map and determine a threshold to segment driver face in the disparity map regardless of the background objects.

It is tried to determine face location considering the skin color of a face in color based segmentation method [12]. These type of methods are not efficient on face localization since background of driver changes dynamically and continuous. In our study, we localized the driver's face with respect to disparity map. Since the image appeared foreground is likely to be a driver face, disparity of that image would be higher. If we wish to eliminate specified disparity, the driver's face can be obtained on the captured frame. It can be appeared the driver's face data in 3D if we reconstruct using face depth image extracted in disparity map. Since resulted image data is in 3D, it can be compared to the images included in database in accordance with different point of views by rotating 3D image in a specified rate, and so more efficient recognition can be achieved. It can be saved the unsuitable behavior appeared in driver such as phone call, and then the unsuitable behaviors can be recognized.

The most basic cue about a driver's actions is his pose. However, tracking a driver's articulated motion in an environment with rapidly varying illumination and many potential self-occlusions is prohibitive both in terms of computational resources (for model-based tracking) and since the initialization of an articulated model is non-trivial in an automatic fashion.

We tried to use only one webcam at first. We have understood that the face recognition using by PCA or HMM approach needed sufficient illumination [13]. Therefore, it has been aimed not only to use 2D or 3D setup for processing driver's image but also to try mixed algorithm as a challenging task. If we use 2D images with illumination improvement or mentioned additional steps, the behavior recognition is likely to be easier but there would be possibly to impersonate the program using ordinary driver's own photograph.

In this study, we obtained the disparity map from the images that are captured using by webcams. Thus, segmentation would be easier in extracting needed sub-image using the disparity or intensity map.

In the disparity map, it can be easily segmented of the driver face from background images because it is supposed that background would have higher depth and driver face has lower depth than the others. Then, in the next step, we studied on the segmented face image. Furthermore, it would be possible to retrieve needed knowledge apart from driver face recognition.

For example, it can be obtained the range data or pose of driver via disparity map. The steps in our approach are as follows:

Haluk Eren, Department of Computer Technology, Fırat University, Elazığ-Turkey , heren@firat.edu.tr

Ümit Çelik, Department of Computer Engineering, Çankaya University, Ankara-Turkey, ucelik@cankaya.edu.tr

Mustafa Poyraz, Department of Electrical and Electronics Engineering, Fırat University, Elazığ-Turkey mpoyraz@firat.edu.tr

1-4244-1068-1/07/$25.00 ©2007 IEEE.

- Acquiring and producing stereo image pair
- Disparity extraction
- Thresholding with lower intensity image data on disparity map in order to extract face
- and so, assuming that segmented and remained disparity map is includes the face of driver.
- To assume the default image as 2D face.
- To analyze the image with some known algorithms such as face or pose recognition.

Firstly, stereo image pair has been produced. In the second phase, we extracted disparity map using the stereo pair. Then, we segmented the image in the 3 rd step. In this stage, we tried to give an estimated threshold value in order to be appeared driver face by segmenting the disparity map. To make recognition easier, we assumed the obtained image as 2D face and it was processed on 2D image data. It was used some functions embedded in OpenCV in order to understand if this image was a real driver's face [5]. To recognize the driver face, we tried the PCA and the HMM approaches.

A. *Acquisition of Stereo Image Pairs*

For vehicle, as an alternative, lower quality USB-based cameras were used in order to obtain the stereo image pair imagery required. Standard USB cameras have a significantly lower resolution to that of the Pulnix cameras. The pre-collected files and live images are acquired using the functionality available in the OpenCV library.

Fig. 1 The stereo head used in experiments

Our system is designed so that it is compatible with inexpensive USB cameras. These Logitech USB cameras are more affordable and portable, and perhaps most importantly, support a higher real-time frame rate of 30 frames per second.

II. THE ALGORITHM

Most of details on the known steps concerning intensive stereo calculations have not been included in this article. We obtained intrinsic and extrinsic camera calibration parameters and projection matrix in early stages of proposed system.

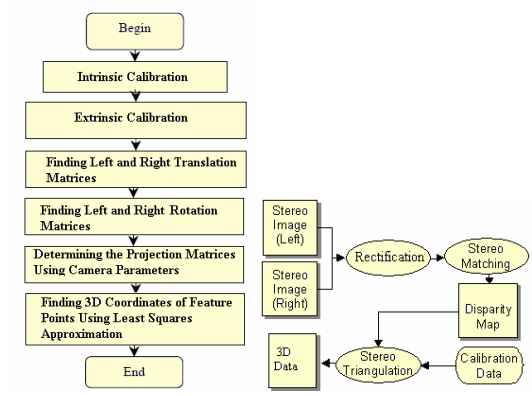

Fig. 2 The algorithm and stereo scheme used in this study

Assume the object frame coincide with the left camera frame and let (c_l, r_l) and (c_r, r_r) be the left and right image points respectively. The 3D coordinates (x, y, z) can be solved through the perspective projection equations from left and right image as

$$\lambda_l \begin{pmatrix} c_l \\ r_l \\ 1 \end{pmatrix} = W_l M_l \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad \text{and} \quad \lambda_r \begin{pmatrix} c_r \\ r_r \\ 1 \end{pmatrix} = W_r M_r \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

This equation system consists of 5 unknowns ($x, y, z, \lambda_l, \lambda_r$) and 6 linear equations, the solution can be obtained using the least-squares method. Let $P_l = W_l M_l$ and $P_r = W_r M_r$ represent the projective matrix for left image and right image, respectively.

The combination of the equations, we have

$$\begin{pmatrix} P_{l11} & P_{l12} & P_{l13} & -c_l & 0 \\ P_{l21} & P_{l22} & P_{l23} & -r_l & 0 \\ P_{l31} & P_{l32} & P_{l33} & -1 & 0 \\ P_{r11} & P_{r12} & P_{r13} & 0 & -c_r \\ P_{r21} & P_{r22} & P_{r23} & 0 & -r_r \\ P_{r31} & P_{r32} & P_{r33} & 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ \lambda_l \\ \lambda_r \end{pmatrix} = \begin{pmatrix} -P_{l14} \\ -P_{l24} \\ -P_{l34} \\ -P_{r14} \\ -P_{r24} \\ -P_{r34} \end{pmatrix}$$

The least-squares solution of the linear system $AX = B$ is given by $X = (A^T A)^{-1} A^T B$. The 3-D coordinates can be thus obtained from the two corresponding image points.

The above process should be done for each pixel, individually. We used the formula above to reach 3D data from 2D data of each point. As shown the matrix above, we need so many matrices to get matching each pixel. Some reconstructed 3D objects are shown in Figure 3.

Fig. 3 Reconstructed image samples

III. EXTRACTING DISPARITY MAP TO SEGMENT DRIVER'S FACE AND RECONSTRUCTION

A. *Disparity map processing*

A disparity map or "depth map" image is an efficient method for storing the depth of each pixel in an image. Each pixel in the map corresponds to the same pixel in an image, but the grey level corresponds to the depth at that point rather than the gray-shade or color. Objects nearer will have greater separation (this is the disparity), and objects very far away will line up very close (they will have less disparity). In fact, this is how stereo vision algorithms work. The computer attempts to match every pixel in an image with every pixel in the other using a correspondence algorithm. Viewing a gray level histogram equalized disparity map image, objects that are lighter are close, and darker object are farther away. To determine the depth, two images are processed to produce a disparity map and since there are two images, the disparity value at each pixel is referenced to one image. These images are directly from our stereo webcam system. The actual image produced for extracting depth is quite different. Disparity map can be become more clearly using image histogram equalization operation. Matlab code as shown below performs histogram equalization process.

```
A = imread ( 'disp.pgm');
data = bitand(A,255) ; % Strip off Fractional portion
A = uint8(data); % Get back to 8 bit format
figure, imshow(A) % Show the image before histogram
J = histeq(A); % Perform histogram equalization
figure, imshow(J) % Show the image after histogram equalization
```

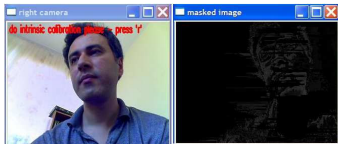

Fig. 4 Left is one of the input image from the stereo imagers. Second image is the disparity image produced by OpenCv– brighter areas is higher disparity, and closer to the camera.

A strategy to remove the floor data is to use the information from the Y direction and determine the height of the objects producing the signals. By removing the signal from the floor, amazingly clear signals produced by the actual objects-of interest can be seen.

```
A = imread ( 'bvz.pgm'); % Load the image
k1=230; % add 1 for level 22. Matlab starts at 1, pix starts at 0!
for i = 1:288 % Remove the floor data
      pixKill = find(A(i,:) < k1); % Find background pixels in row to cancel
      A(i,pixKill) = 80; % Cancel the background
end
imshow(A)
```

Fig. 5 Giving a threshold to extract only the real driver face

Disparity value of closer face is appeared as higher. Therefore, if there are some undesired stuffs, such as face, in the driver's frame they can be easily eliminated, and so we can focus on the driver face. In addition, we can obtain the pose of driver's body and recognize the driver's actions by processing the frame since obtained objects show the range knowledge. For example we can understand unsuitable driver actions such as phone call by using PCA or HMM approach. We captured a stereo image pair in the experiment below. There are two faces and many objects in the frame. It seems hard way to segment desired driver's face or body image in this frame unless using 3D. We considered that the closer person is likely to be driver in the disparity map which is earlier extracted by using stereo pair. Then we reconstructed the driver's face and body in 3D. In order to localize driver's face and body via obtained image, we used blob coloring algorithm which is based on the neighbor pixels.

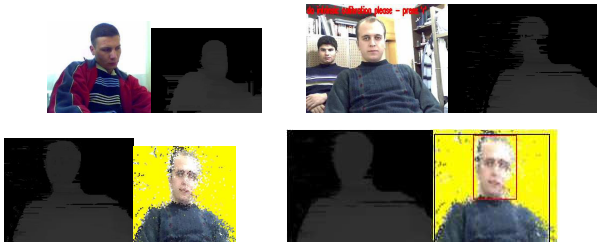

Fig. 6 Removing data at specified level from the original disparity map

Data desired at a specified level within disparity map can be removed in this manner. It can be done a ground truth for verifying pose estimation in the figure below:

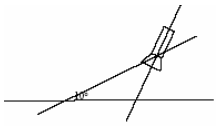

Fig. 7 Ground truth for verifying pose estimation

IV. CAMERA POSE ESTIMATION

The general concept of vision-based camera 3D pose estimation is to find the best set of camera position and orientation data (the six extrinsic parameters) to fit a known model in the target image. When camera intrinsic parameters are given or available from the initial calibration, it is also known as the *absolute orientation problem*. Unlike other sensing technologies, vision solutions directly estimate camera pose from the same imagery that is also used as the

real world background. Pose estimation is an essential step in many machine vision problems involving the estimation of object position and orientation relative to a model reference frame or relative to the object position and orientation at a previous time using a camera sensor or a range sensor.

In a method that uses depth information obtained from 3D camera, however, overall pose estimation error largely depends on instability of the depth information and feature points tracking error. This causes serious problems in real-time 3D camera motion tracking. On the other hand, a method which only uses geometric information and tracked feature points can only recover camera motion up to scale since we have no absolute measurements. Additionally, in a specific camera motion, we can not recover the motion due to triangulation uncertainty. Geometric relationship between two images obtained from calibrated cameras is represented as 3x3 Essential matrixes (E). E consists of camera motion parameters, rotation and translation matrix. Thus, if we know E, we can recover camera motions until up to scale. After calculating *E*, we decompose *E* into rotation and translation matrix. As a result, we can get totally four different rotation and translation matrix after separating *E* with *SVD* method. Among them, we have to select correct rotation and translation matrix by applying geometrical relationship between a camera and triangulated 3D points.

A. Homography estimation

The three-dimensional motion parameters of a rigid planar patch can be determined by computing the singular value decomposition (SVD) of a 3x3 matrix containing the eight so called "pure parameters." This planar transformation is called a homography. Furthermore, aside from a scale factor for the translation parameters, the number of solutions is either one or two, depending on the multiplicity of the singular values of the matrix. This method provides the rotation, the translation of the moving camera up to a scale factor and the orientation of the 3D plane of interest.

B. Direct pose estimation of 3D camera using depth information

The pose of 3D camera can be solved using calibrated camera theory and depth information. If the accuracy of depth information obtained from 3D camera is guaranteed, recovering camera motion is regarded as solving absolute orientation problem. It is resulted in finding rotation matrix and translation matrix with respect to corresponding feature points in two 3D point sets.

$$\sum_{i=0}^{n} \left\| B_i - (R \cdot A_i + t) \right\|^2$$

where, $\{A \leftrightarrow B\}$ represents corresponding 3D point sets, R and t mean rotation matrix and translation matrix between two 3D point sets, respectively.

V. DRIVER'S FACE RECOGNITION USING PCA AND HMM

A. Face Detection

Driver's face detection is the first step for these type of applications in accordance with human computer interface. Modeling skin color requires choosing an appropriate color space and identifying a cluster associated with skin color in this space. Since the skin-tone color depends on luminance, we nonlinearly transform the YCbCr color space to make the skin cluster luminosity-independent. This also enables robust detection of dark and light skin tone colors.

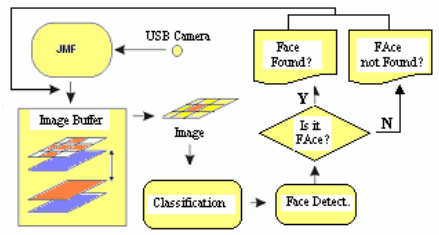

Fig. 8 Face detection algorithm.

B. Facial recognition algorithms

Some approaches to face recognition are as follows:

- Geometric feature based methods
- Template based methods
- Model based methods
- Various computer algorithms exist to recognize faces
  - Eigenface analysis (PCA)
  - Hidden Markov Models(HMM)

C. PCA (Principle Components Analysis) Algorithm

In PCA algorithm, face images included in database are firstly converted eigenfaces. Then, the significancy of the images in database and the existed image are calculated, and matched with experiment set. It can be decided that the image which has closest weight is to be desired image.

Main steps in PCA are basically as follows:

a. Segmenting a comlex image.
b. Producing some extractions from face. It can be divided in two steps: Some features based on whole image (characteristic features) and partial features (such as eyes, nose, hair).
c. Decision.

Resolution	Recognition Rate	Number of Eigenvectors (eigenvalues over 95%)
16*16	0.19	44
32*32	0.57	47
64*64	0.83	49
120*120	0.93	49

Table 1 Effect of resolution on PCA algorithm

Expressing a face as a vector: A face image can be expressed as a vector form. For example, a face image have components of pixels as many as aspect ratio.

Fig. 9 Face vector extraction from face image.

As known, face segments are similar to each other geometrically. Therefore it will lead vectors, to be obtained, to be closer to each others. The goal is to estimate the most similar vector which represent the face.

D. *Hidden Markov Model based Algorithms*

HMM is similar to eigenfaces. Set of characteristics are stored from a set of images of the same face. The set of images are used to compare if face in another picture matches.

Facial Features of a frontal face include the hair, forehand, eyes, nose and mouth. These features occur in a natural order, from top to bottom. Therefore, the image of a face may be modeled using a one dimensional HMM by assigning each of these regions to a state. The states themselves are not directly observable and observation vectors are statistically dependent upon the state of the HMM. These vectors are obtained from L rows that are extracted sequantially from the top of the image to the bottom. Since 2D HMM's are too complex for real time face recognition, embedded HMM is used. Embedded HMM uses observation vectors that are composed of two dimensional Discrete Cosine Transform coeffecients. The DCT helps separating the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality).

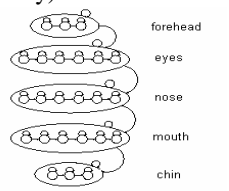

Fig. 10 Embedded HMM Model for face recognition

HMM is a Markov chain with finite number of unobservable states. These states have a probability distribution associated with the set of observation vectors. Things necessary to characterize HMM are as follows:

-State transition probability matrix.

-Initial state probability distribution.

-Probability density function associated with observations for each state.

Embedded HMM: Making each state in a 1-D HMM, makes an embedded HMM model with super states along with embedded states. Super states model the data in one direction (top-bottom). Embedded states model the data in another direction (left-right). Transition from one super state to another is not possible. Hence it is named "Embedded HMM".

1) *Embedded HMM for Face Recognition*

We implemented face recognition scheme "Embedded HMM" Embedded HMM approach uses an efficient set of observation vectors and states in the Markov chain [14].

Markov Chains: How to estimate probabilities S is a set of states. Random process {Xt|t=1,2….} is a Markov Chain if ∀t, the random variable Xt satisfies the Markov property.

Observation Vectors: P x L window scans the image from left-right and top-bottom, with overlap between adjacent windows is M lines vertically and Q columns horizontally. Size of observation vectors = P x L [14].

Pixel values don't represent robust features due to noise and changes in illuminations. 2D-DCT coefficients in each image block (low freq components, often only 6 coefficients). This helps reduce size of observation vector drastically.

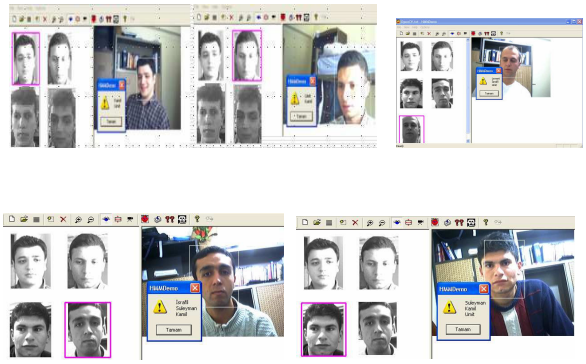

Fig. 11 Face recognition samples using HMM

PCA Algorithm: After it is dealt with face localization problems, we started to study face recognition via PCA algorithm [13]. We captured 10 people in 5 different positions and ranges, and then saved them as a database for experiment test set. At the first stage, PCA algorithm has been tested using some of face images and understood the advantages and drawbacks. We appeared that some factors related to image such as noise ratio, resolution were effected on the result of PCA algorithm. Then, we tested PCA algorithm in real time arrangement. If it is provided specified conditions that is suitable to PCA algorithm, we appeared that error rate was getting minimum and the result

was challenging. We assigned a folder which includes some face images with an aspect ratio of 160x190. These specified face images was kept in disc as JPEG graphical format. Then it has been obtained localization and size of faces on the captured real time images. Once face images reach to a resolution of 160x190, PCA algorithm has been executed. The recognition process has been accomplished by matching just captured new image and existed images in database. We observed that the result of the experiment was satisfied and true. The main difficulty in the experiment is that images included in database and real time captured images must be in same resolution. In following pictures, it is shown some sample results using PCA algorithm.

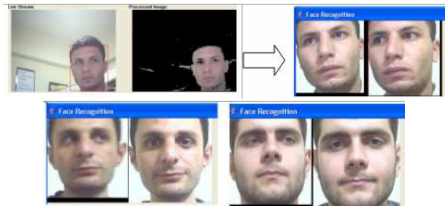

Fig. 12 Detection of face location and recognition

Although captured image is very noisy as shown in the figure, it is appeared that the accuracy of the result is satisfied. Therefore, it shows the performance of PCA algorithm is better.

VI. CONCLUSION

In this study, we preferred different type of webcams in order to use in vehicles and tried to get performance characteristics of them. The use of an embedded HMM model for the human face is just justified by the structure of the face, and is invariant for a large range of orientations, gestures, and face appearances. The use of an embedded HMM increases over 10% the recognition rate of the one-dimensional HMM and the classical eigenfaces method. It has been compared some recognition methods such as HMM or PCA to get driver's image. Moreover, we tried to understand how to contribute 3D processes to these types of approaches. We have done some experiments to process driver behavior. As a result we consider that there is some possible application fields are as follows:

- We can determine driving durations of coach drivers. Therefore, it can be given some useful information (such as driver fatigue level, identification, "how long vehicle driver is driving") to passengers in accordance with their traveling safety.
- It can be got statistical information on driving habits of licensed drivers.
- Once we obtain drivers pose recognition, drivers can be noticed about their fatigue strengths or vehicle can be controlled according to driver's fatigue level.
- Vehicles can be prevented from burglary. Burglar can take webcam, but if driver's biometric information on the licensed and real driver at that moment doesn't match then vehicle will not work as a result of recognition phase.
- This situation is not actually practical for drivers since they don't want to wait for a short while in first ignition and moving. This arrangement leads driver to wait at least half minutes so they must be a little bit patient before moving the car. Therefore this arrangement can be used for specified group of vehicles such as commercial taxi or interurban coach drivers.

REFERENCES

[1] E. Trucco and A. Verri. Introductory Techniques for 3-D Computer Vision, Prentice Hall, 1998.

[2] "Depth Discontinuities by Pixel-to-Pixel Stereo" Stanford University Technical Report STAN-CS-TR-96-1573, July 1996.

[3] S. Birchfield and C. Tomasi. A Pixel Dissimilarity Measure That Is Insensitive to Image Sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 20, No 4, April 1998.

[4] L. Di Stefano, M. Marchionni, S. Mattoccia "A PC-based Real-Time Stereo Vision System" Machine GRAPHICS & VISION 13(4) pp 197-220, January 2004

[5] G. Bradski, A. Kaehler, V. Pisarevsky, Learning-Based Computer Vision with Intel's Open Source Computer Vision Library, Intel Technology Journal, Vol. 9, ISSN 1535, May 2005, Pg 119-130.

[6] Daniel Scharstein and Richard Szeliski, "A Taxonomy and Evaluation of Dense Two frame Stereo Correspondence Algorithms", to appear on *IJCV*, pp. 7-42, Vol. 47 No. 1, May 2002.

[7] Gorodnichy, D.O., Malik S. & Roth, G. (2002). Affordable 3D face tracking using projective vision. Proceedings of International Conference on Vision Interface (VI'2002), 383-390.

[8] Eren, H.; Celik, U.; Poyraz, M.; Approaches on the Selection of Web Cameras and Calibration Targets for Stereo Vision, Signal Processing and Communications Applications, 2006 IEEE 14th, 17-19 April 2006.

[9] M Kaya, H. Eren, Spider Robot (Genetic Programming Approach), The 7th Mechatronics Forum International Conference, Georgia University, Georgia, USA, 6th - 8th September 2000.

[10] Haluk Eren, Ümit Çelik, M. Poyraz, Factors Affected On The Performance Of Stereo Vision And Key Aspects, P. 537-549, 4th FAE International Symposium, Gemikonagı – Lefke TRNC, 1 December 2006.

[11] H.A. Rowley, S. Bluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (1) (1998), pp. 23-38.

[12] M. Yang, D. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, no. 1, January 2002.

[13] Turk M., and Pentland A., 1991, Face recognition using eigenfaces, In *Proc. of Computer Vision and Pattern Recognition*, pp. 586-591, IEEE

[14] Ara V. Nefian and Monson H. Hayes III. "An embedded HMM based approach for face detection and recognition", IEEE International Conference on Acoustics, Speech and Signal Processing, March 1999, p.3553-3556, vol VI.