

A Feasible Face Pose Estimation by Evaluating 3D Facial Feature Vectors from 2D Features

Wongki Kim, Hyungwoo Lee, Wonjin Lee, Hyejin Song, Junchul Chun

Department of Computer Science, Kyonggi University, San 94-6 Yiui-Dong, Yeongtong-Gu, Suwon, S. Korea

{bemong1, simoni88, dnjswls23}@naver.com, sonuser1234@gmail.com, jcchun@kgu.ac.kr

Abstract— Estimating head pose correctly from input face images is important for developing applications of vision-based human computer interaction. In this paper, we present a robust 3D head pose estimating approach using a skin-color model and haar-like feature detection. Using the detected primitive facial features of an obtained facial region, the corresponding 3D head pose of the 2D face is estimated. As for facial region detection, we adopt YC_bC_r model for skin-color model. For facial feature detection from the detected facial region, Haar-like feature is utilized along with AdaBoost learning. The 3D head pose of an input face image can be obtained by evaluating 3D information of facial features from the detected 2D eye-points and nose. Without any constraints for head pose estimation such as using initial face pose template of a frontal face, our proposed method can estimate face pose and retarget the motion to a 3D face model in real time. From the experiments, the proposed approach shows robustness in face and facial feature detection and eventually produces better results in estimating head pose rather than simply using Haar-like feature for both face and facial feature detection.

Keywords— Head pose estimation, Skin-color model, Haar-like feature, AdaBoost algorithm, Face detection

I. INTRODUCTION

Head pose estimation in computer vision is a process to infer the orientation of a person's head from input video images relative to the view of a camera. The techniques of head pose estimation are widely used in human computer interaction, vision-based animation and gesture recognition and so on. Various approaches for head pose estimation from a still image or a sequences of input images are introduced[1].

Many studies have been done for recovering face motion from image sequences. One is to use distinct image features[2], which work well when the features may be reliably tracked over the image sequence. However, when good feature correspondences are not available, tracking the entire facial region using a 3D head model is more effective. In this case, both generic and user-specific model have been used for head motion recovery[3]. Much simpler geometric models such as planner model and ellipsoidal model have been also introduced[4].

II. PROPOSED METHOD

The proposed head pose estimation method consists of three major steps. The first step is detecting face, the second

step is detecting facial features from the detected facial region and the third step is estimating the face pose based on the vector values of the facial features. Fig 1 illustrates the overall steps for estimating face pose from a sequence of input video images.

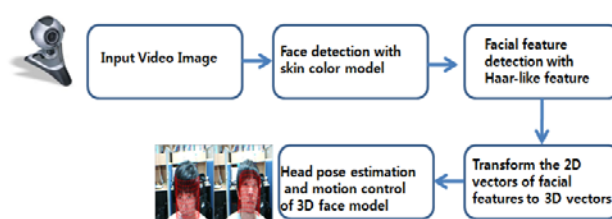


Figure 1. The overall procedures of the proposed face pose estimation approach

First, the facial region is separated by using a skin color model and a post-processing. In the subsequent stage, nose and eyes are detected with Haar-like feature and Adaboost learning algorithm. When major facial features such as nose and eyes are detected, two vectors between the position of nose and two eyes are evaluated and transformed to 3D vectors to obtain the depth value(z-coordinate of facial features) for face pose estimation. Finally, the head pose is retargeted to a mock face model.

A. Skin Color Model for Face Detection

Face detection is an initial step to extract facial features to estimate head pose based on tracking facial features. Face detection methods can be broadly classified into feature-based methods and image-based methods. In image-based methods, models are trained from a set of training set before detecting face. Using a large amount of training data, these methods show high accurate detecting rate against light and pose variation. Principal Component Analysis(PCA), Support Vector Machine(SVM) and Hidden Markov Model(HMM) are well-known for classifiers to detect face. However, these methods need a mounts of large face and non-face data set for training to keep the detecting accuracy.

Meanwhile, the feature-based methods make explicit use of face knowledge and follow the classical detection methodology in which low-level features such as color, edge, and are derived prior to knowledge-based analysis. Among them, skin color models have been available for detecting face from input images. Even though some models for face

detection are relatively sensitive for surrounding light variation and mislead for false detection, however it is known that skin color models against robust to light variation are efficient and simple to detect face in real time applications. Mostly the primary components of R,G,B are used for skin segmentation. Other models such as HSV, normalized RGB, YC_bC_r , YIQ etc. are used for the segmentation of skin-like regions[5]

In this work, we adopt YC_bC_r model for face detection with By discarding luminance (Y) we use define a region of skin tone pixels using only C_b, C_r values from samples of skin color pixels. With carefully selected threshold range values, a pixel value is classified to the face region if its value meets the range. To detect the facial region, we use the ranges of C_b, C_r experimented by Chai[6]. The binary image of the detected facial region is determined by following formula:

$$B_{face}(x, y) = \begin{cases} 1 & \text{if } (77 \leq C_b \leq 127) \cap (133 \leq C_r \leq 173) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Fig 1 illustrates a detected facial region using C_b, C_r skin color model and morphological operation which can reduce the noise from the segmented binary images.

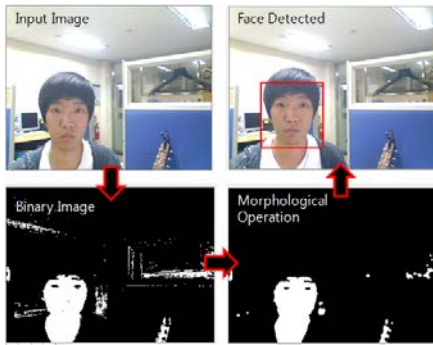


Figure 2. Face detection based on C_b, C_r and morphological operation for post-processing

B. Facial Feature Detection Using Haar-like Feature

To estimate head pose from an input video image, the detection of specific facial features are important. Because, in this work the locations of both eyes and nose are used to estimate head pose. We use haar-like feature and Adaboost learning algorithm to detect such facial features rather using pixel directly. Haar-like feature is originated from the Haar basis functions which have been used by Papageorgiou et al[7]. A Haar-like feature takes into account adjacent rectangular regions at a specific location in a window, sums up the pixel intensities in each region and calculates the difference between these sums. Subsequently the difference is used to categorize the subsections of an image.

The value of simple rectangle Haar-like feature is defined as the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the

detected face image. This rectangle features can be computed very rapidly by using summed area table called integral image[8]. Fig 2 shows examples of two-rectangle features and three-rectangle feature, respectively. The integral sum (I_{sum}) of location $1(x, y)$ in Fig 2 can be sum of the pixels above and the left of $1(x, y)$ and can be expressed by

$$I_{sum}(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \quad (2)$$

The sum of pixels in the rectangle D can be computed as $4+1-(2+3)$.

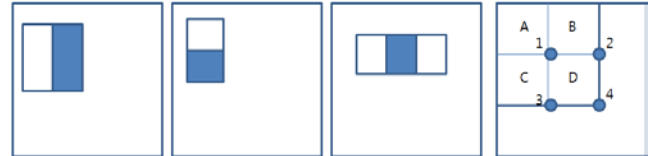


Figure 3. Examples of rectangle features(two-rectangle features and threerectangle feature) and rectangular sum using integral image

The calculated feature values are used for Adaboost classifier to recognize facial features such as eyes and nose.

C. Adaboost Learning Algorithm

When the face is detected, the facial features are recognized by using Adaboost learning algorithm which was originally developed to detect face by Viola[8][9]. The detector consists in scanning an image by shifting window at different scales. Each sub-window is tested by a classifier made of several stages. The sub-window is rejected when it is determined clearly as non-face. However, if it is difficult to discriminate the sub-window, more specific classifier in the late stage determines the sub-window non-face or face.

In order to reduce the effect of surrounding light variation for the facial region the histogram equalization is proceeded before applying Adaboost algorithm for facial feature detection.



Figure 4. The histogram equalization is used for detected face region before Adaboost learning is applied.(A detected facial region(left) and its enhanced image(right))

Adaboost is an algorithm for constructing a strong classifier $f(x)$ as a linear combination

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (3)$$

of simple weak classifiers $h_t(x)$ where $h_t(x)$ is weak or basis classifier(feature). The Adaboost algorithm only needs two inputs: a training data set and set of features. It is not

necessary to have any prior knowledge of facial features such as eye and nose.

For facial feature detection, we exploit the Adaboost learning algorithm which was used for face detection by Viola[9]. The discrete algorithm can be described as follows:

- Given: $(x_1, y_1) \dots (x_m, y_m)$, $x_i \in X$, $y_i \in \{-1, 1\}$
 - Initialize weight $D_1(i) = \frac{1}{m}$
 - For $t = 1 \dots T$
 - ① Call weak learn which returns weak classifier $h_t : X \in \{-1, 1\}$ with minimum error w.r.t D_t
 - ② Choose $\alpha_t \in \mathbb{R}$
 - ③ Update weight $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
- Where Z_t is a normalization factor chosen so that D_{t+1} is a distribution
- Produce the strong classifier:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

Figure 5 show the results of detected facial features using Harr-like feature and Adaboost learning algorithm.

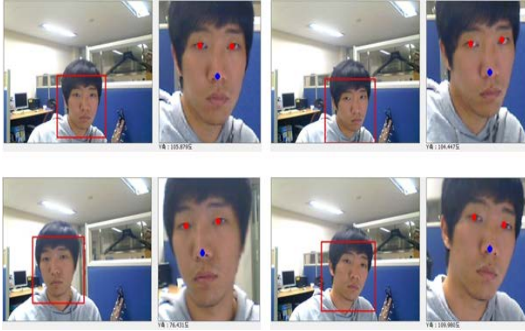


Figure 5. Detected facial features by using Haar-like feature and Adaboost learning algorithm

D. Head Pose Estimation

The proposed head pose estimation method only use the positions of detected eyes and nose and two direction vectors composed of nose and eyes. If $\vec{L}$ is a directional vector between the positions of nose and left eye and $\vec{R}$ is another vector between the positions of nose and right eye, then we can assume that the norms of $\vec{L}$ and $\vec{R}$ are equal in real environments. For 3D head pose estimation we estimate 3D positions of nose and eyes by expanding $\vec{L}$ and $\vec{R}$ from 2D to 3D. If the $\vec{L} = (a, b, 0)$ and $\vec{R} = (d, e, 0)$ then the norm of each vector can be represented by

$$\begin{aligned} \text{Norm}(\vec{L}) &= \sqrt{a^2 + b^2 + 0^2} \\ \text{Norm}(\vec{R}) &= \sqrt{d^2 + e^2 + 0^2} \end{aligned} \quad (4)$$

When norms of two vectors are evaluated from a specific input image, we regard the vector which has a bigger norm value as a basis vector and reevaluate the z value of another vector to equalize the magnitudes of two vectors. If X is the norm of the vector which is greater than the norm of the other vector, the later vector can be defined by $\vec{S} = (p, q, t)$. Since the norms of $\vec{L}$ and $\vec{R}$ also become equal in 3D space, we can obtain 3D positions of facial features simply adjusting z coordinate of the $\vec{S}$. The z coordinate of the $\vec{S}$, t can be redefined by

$$t = \sqrt{X^2 - (p^2 + q^2)} \quad (5)$$

Then the 3D vector of $\vec{S}$ becomes $\vec{S} = (p, q, t)$. Finally the head pose can be obtained by the cross product ($\vec{V} = (x, y, z)$) of $\vec{L}$ and $\vec{R}$. The vertically rotated angle θ can be expressed by

$$\theta = \arccos \left(\frac{y}{\text{norm}(\vec{V})} \right). \quad (6)$$

Figure 6 shows the results of estimated angle of head pose vertically (y direction).



Figure 6. Estimated head pos

III. EXPERIMENTAL RESULTS

Figure 7 includes experimental of face detection and facial feature detection. The face detection is performed based on the proposed skin color model and Haar-like feature, respectively. As illustrated in Figure 7, the proposed skin color model is more reliable to obtain facial region rather than using Haar-like feature. In facial feature detection, without equalizing the facial region before detecting facial features, some cases show failure in detecting facial feature. However, the equalization of the face region can make more reliable results in detecting facial features as shown in Figure 8.

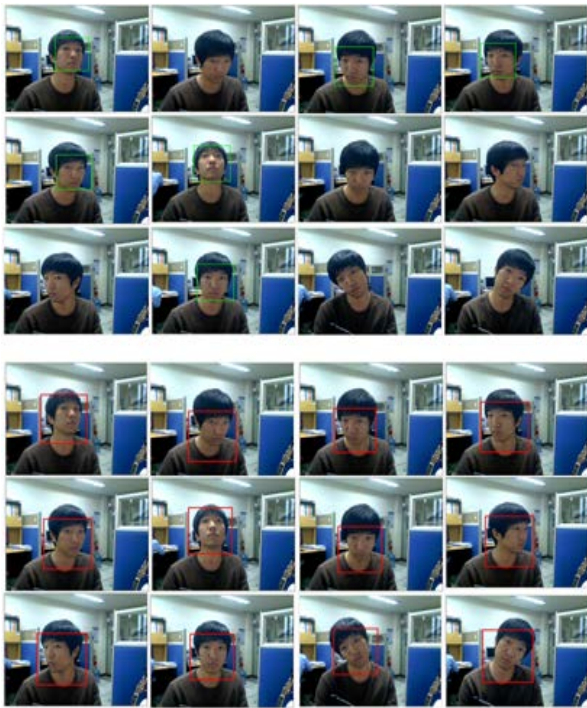


Figure 7. Results of face detection with Haar-like feature(top) and with $YCbCr$ skin color model(bottom)



Figure 8. Detected facial features without equalization(top) and with equalization of face region.

Figure 9 illustrates the estimated head pose using the vector values of facial feature explained in previous section and its corresponding retargeted head pose of a 3D face model.

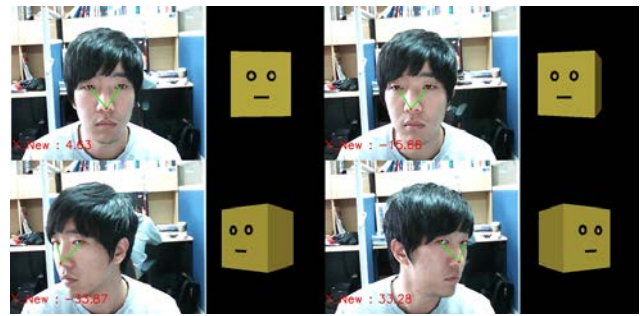


Figure 9. Estimated head pose and its retargeted head pose for a 3D avatar

IV. CONCLUSION

In this work, we propose and develop a real-time head pose estimation method using Harr-like feature and Adaboost learning. The head pose is easily estimated by obtaining the 3D information of nose and eyes from 2D information of facial features. The constraints in estimating head pose in previous works such as using stereo-vision or using a initial template from frontal face etc are not necessary in the proposed head pose estimation technique. It also efficiently used in real-time fashion for a sequence of input images with less computational cost.

ACKNOWLEDGMENT

This work is supported by GRRC (Gyeonggi Regional Research Center), the research title of “Developing and Industrializing Core-Technologies of the Smart-Space Convergence Framework” 2012.

REFERENCES

- [1] E. Murphy-Chutorian and M. Manubhai Trivedi, Head Pose Estimation in Computer: A Survey, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 4, pp. 607-627, 2009.
- [2] J. Xiao, T Kanade and J. Cohn, *Robust full-motion recovery of head by dynamic templates and reregistration techniques*, *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 163-169, 2002.
- [3] J.C Chun, O.R kwon, *A vision-based Approach for Facial Expression Cloning by Facial Motion Tracking*, *KSII Transactions on Internet and Information Systems*, Vol. 2, No. 2, pp. 120-133, 2008.
- [4] G.D. Hager and P.N. Belhumeur, *Efficient region tracking with parametric models of geometry and illumination*, *PAMI*, Vol. 20, No. 10, pp. 1025-1039, 1998.
- [5] K. Min, J. Chun, G. Park, *A Nonparametric Skin Color Model for Face Detection from Color Images*, *LNCS 3320*, pp. 115-119, 2004.
- [6] Chai. D. and Ngan. K.N, *Face Segmentation using-skin color map in video phone applications*, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, Pp. 551-564, 1999.
- [7] C. Papageorgiou, M. Oren, and T. Poggio. *A general framework for object detection*. In *International Conference on Computer Vision*, 1998.
- [8] Paul Viola and Michael Jones, *Rapid Object Detection using Boosted Cascade of Simple Features*, *Proceedings of the 2001 IEEE Conference on Computer Vision and pattern Recognition*, pp.
- [9] Paul Viola and Michael Jones, *Robust Real-Time Face Detection*, *International Journal of Computer Vision*, Vol. 57, No. 2, pp. 137-154, 2004.