Facial Feature Tracking and Head Orientation-based Gaze Tracking

Jong-Gook Ko*, Kyungnam Kim**, SeungHo Choi***, JinYoung Kim****, KiJung Kim*****,and JungNyo Kim*

*Electronics and Telecommunications Research Institute (ETRI)
1 Kusong-dong, Yusong-gu, Taejon 305-350, Korea
Tel +82-42-860-5940, Fax +82-42-860-5611
E-mail: jgko@etri.re.kr*
** Department of Information and Communications
KwangJu Institute of Science and Technology (K-JIST)
***Department of Information and Communications Dongshin University
**** Department of Electronics Chonnam University
*****Department of Internet Information Technology Kwangyang College

Abstract: In this paper, we propose a fast and practical head pose estimation scheme for eye-head controlled human computer interface with non-constrained background. The method we propose uses complete graph matching from thresholded images and the two blocks showing the greatest similarity are selected as eyes, we also locate mouth and nostrils in turn using the eye location information and size information. The average computing time of the image(360*240) is within 0.2(sec) and we employ template matching method using angles between facial features for head pose estimation. It has been tested on several sequential facial images with different illuminating conditions and varied head poses, It returned quite a satisfactory performance in both speed and accuracy.

1. INTRODUCTION

Multimodal user interface is attracting a special attention in recent times, including hand/ head gesture, facial expression, voice and eye gaze[1]. Conventional human computer interaction techniques such as keyboard and mouse are considered as bottlenecks in the information flow between humans and computer systems. In many speech recognition systems, voice signals are recognized with high success rates. However, the recognition ratio is quite dependent on in environment with noise such as car using eye-gaze and lip-reading. Human gaze has also the potential to be a fast input mode of computers. Eye-head controlled interface is used in a wide array of applications: Computer Interface, Virtual Reality and Games, Robot Control, Disabled Aid, Behavioral Psychology, Teaching and Presentation and so on. Facial features locating capability is needed in various applications.

In this paper, we propose a facial features tracking and head pose estimation schemes in order to do construct a novel image-based human computer interface controlled by eye and head, which is a subtask of a multimodal and intelligent interface of a car navigation system.

This paper consists of following. A brief description of related work is contained in Section 2. Section 3 and Section 4 describe the proposed method of locating the facial features and of head pose estimation respectively. Experimental results are provided in Section 5. The paper concludes with Section 6.

2. RELATED WORK

Due attention is being paid by the research community to face detection schemes, several kinds of approach such as template matching method, feature-based approach, color-based method, neural network method, and motion-based method to locate facial features have been proposed in this regard. Template matching method that was introduced by Yuille D.S. uses deformable templates[2][3]. This method is independent of size, slope, and illumination. But, at first, it requires knowledge of initial template of face. And feature-based approach searches the image for a set of facial features and groups them into face candidates based on their geometrical relationship. Yow and Cipolla[4], Leung et.al.[5] and Sumi and Ohta[6] employed this approach. And the color-based detection system[7][8] selects pixels that have similarity to skin color, and subsequently defines a subregion as a face if it contains a large of skin color pixels. But different camera conditions produce significantly different color values even for the same person under the same lighting condition and human feature colors differ from person to person. And the neural network approach detects faces by subsampling different regions of the image to a standard-sized subimage and then passing it through a neural network filter. Sung and Poggio,[10] and Rowley[9] reported this approach. And motion-based approach [11] uses image subtraction to extract the moving foreground from the static background. But, this approach will not work well in the presence of a large number of moving objects in the image. In head pose estimation, Feature-based method and template matching method are also used[12].

In our system, we employ the feature-based approach to use the information that the iris and the pupil are darker than any other features except hair for locating the facial features. And we propose template matching method based on the estimated angles of pairs of facial features for head pose estimation.

3. FACIAL FEATURES LOCATING ALGORITHM

Facial features locating is needed for finding head pose. In this section, we describe a method of locating facial features. We assumed that one user is in front of computer. And we employed the dark information of features, complete graph matching and geometrical information for locating facial features. Figure 1. describe the flow of face tracking briefly.

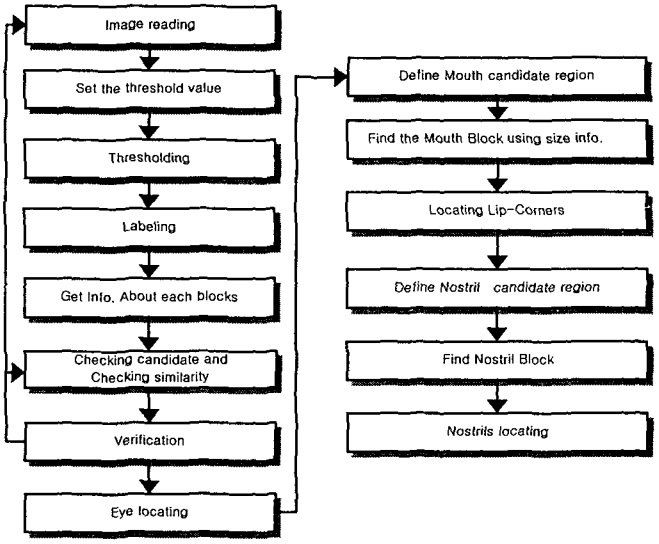

Figure 1. Block diagram of facial feature locating algorithm.

At first, the eyes is located using the dark information and geometrical information of the eyes. And the mouth is located using the information of eyes' position and size information. Finally, the nostrils are located using the information of eyes' and mouth's position and size information.

3.1. Locating the Eyes

Setting the threshold value and thresholding: It is important to find the proper threshold value in order to separate the eyes, nostrils and mouth from face. Among many methods to find the threshold value, we employed a heuristic P-Tile method [13]. After finding the weight center of histogram, the value is subtracted by constant value until the eyes is located for the first time like Figure 2.

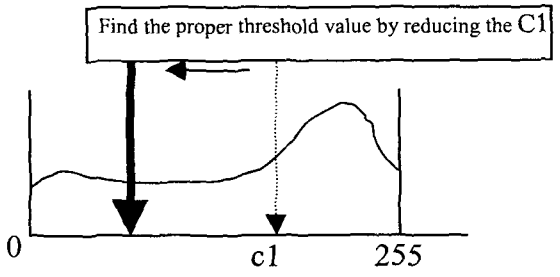

Figure 2. Set the threshod value using heuristic P-Tile method

After finding threshold value, the image should be binarized by that value. An example of a binarized image obtained in this manner is shown in Figure 3.

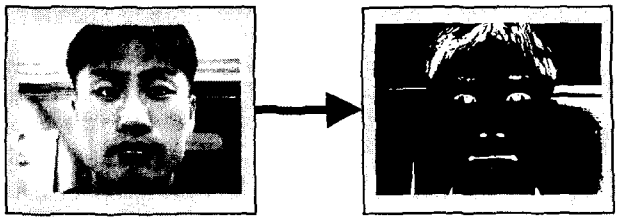

Figure 3. Binarized image

Edge detection may be employed to find the candidates for the eyes. But, it requires amount of computation intensive and it is difficult to find accurate edge pixels. So, thresholding method was employed instead to get the candidate of the eyes.

Finding the candidates of the eyes: After thresholding, We assign unique a tag to each isolated block by labeling the binarized image. In finding the candidates of the eyes, eliminating the blocks that is not satisfied in condition of being the eyes is much efficient than the finding the proper block which is satisfied in condition of being the eye. So we need standard as follows:

Suppose that the two points [x1,y1],[x2,y2] are the top-left point and bottom-right point of a circumscribed rectangle respectively. Let l(x,y) be the tag of the pixel.

$$\text{(i) } Size(i) = \sum_{x=x1}^{x2} \sum_{y=y1}^{y2} F(l(x, y))$$

$$\text{(if } l(x, y) = i \text{ then } F(i) = 1)$$

$$Min \le Size(i) \le Max$$

$$\text{(ii) } Ratio = Max\_Vertical / Max\_Horizontal$$

$$(Ratio \le 1)$$

If the block does not satisfy the conditions (i) and (ii) , then the block is eliminated from the candidate set. Condition (i) implies that the size of eye's block is between Max and Min value. Here, we define the Max and Min as 30 pixels and 300 pixels in size respectively by experimental results. By eliminating the blocks using the rough and simple size information, we could reduce the number of candidate blocks to a quarter. Condition (ii) means that the aspect ratio of the eye is less than 1.

Looking for similarity by complete graph matching: After eliminating the unsatisfactory blocks, a complete graph is composed with the candidate blocks and similarity for each pair is computed. Similarity is computed as follows:

$$1)Normal\_size(i, j) = Size(i)/Size(j)$$

$$2)Normal\_Average(i, j) = \frac{Average\_gray(i)}{Average\_gray(j)}$$

$$3)Normal\_Aspect\_ratio(i, j) = \frac{A.R(i)}{A.R(j)}$$

$$4)Normal\_Angle(i, j) = 1 - [y_{\_distance} / x_{\_distance}]$$

Normal_size(i,j) refers to similarity of two blocks in size while Normal_Average(i,j) and Normal_Aspect_ratio(i,j) refer to similarity of average gray value and aspect ratio between the blocks respectively. The small value is divided by larger value for normalization. Normal_Angle means the slope over x-axis. The pair of blocks that have the maximum sum of the above four factors are selected as the two eyes. Figure 4 shows the result of locating the two eye blocks.

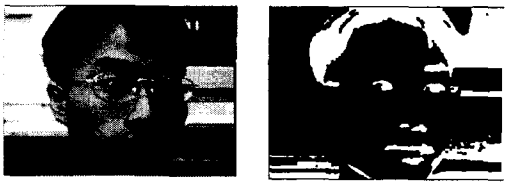

Figure 4. Locating the two eyes

3.2. Locating the Mouth and Lip-Corners

After locating eyes, We can define a rough region for the mouth by using the eye information. Figure 5 shows an example. We consider the largest blocks to be mouth in that region. After locating the mouth's block , we can find the lip-corners by scanning the first and last columns like Figure 6. If the face is tilt to the left like Figure 6, the first column is scanned from bottom to top and the last column is scanned from top to bottom. If the face is tilt to the right, it is scanned in opposition.

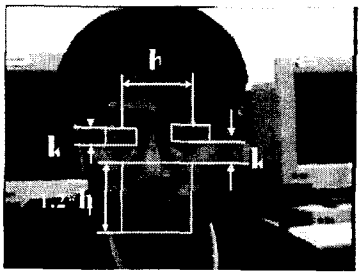

Figure 5. Defining the mouth region

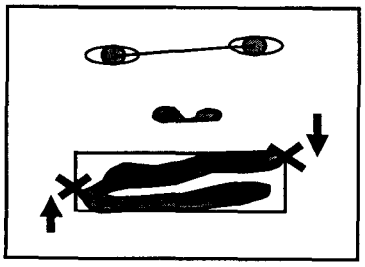

Figure 6. Locating the lip-corners

3.3. Locating Nostrils

We can define the region for nostrils using the two eyes and the mouth position information. Like locating mouth, we used size information of blocks in defined region for locating nostrils. But we should examine whether the nostrils in the image are appeared in one block or not.

3.4. Verification

After locating the facial features such as the eyes, lip-corners and nostrils, we should check whether the facial features have been located correctly using the geometrical information. For example, we can prevent the eyebrow from being selected as the eyes using the information that there are eye blocks under the eyebrows.

4. HEAD ORIENTATION-BASED GAZE TRACKING

We employed template matching using the angles of pairs of extracted facial features for head pose estimation. Each template consists of 6 angles like following Figure 7.

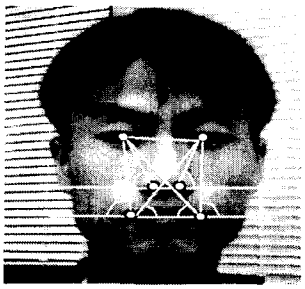

Figure 7. The angles of templates

We do not need consider the distance from user to camera because the angles are independent of the distance. We can create database of 11 templates from one person representing different poses. And each template indicates different gaze points. That is, we divided the computer screen into 11 blocks like Figure 8. We have made database of 55 templates from five persons.

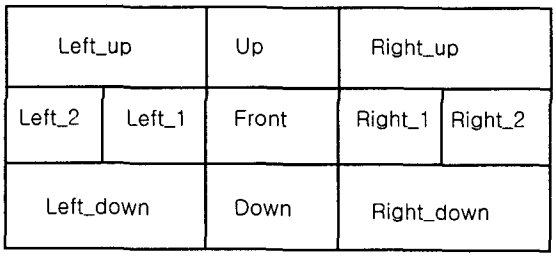

Figure 8. Screen Monitor Resolutions

We compare the angles of input image with those of each template for finding gaze point like following evaluation function:

$$E\_F(i) = \sqrt{\sum_{x=1}^{6} (T\_a(x) - I\_a(x))^2}$$

Where,
T_a(x) : xth angle of the template image
I_a(x) : xth angle of the input image
E_F(i) : evaluation value for the i th template

The template that has the minimum value among evaluation function values is selected as gaze point.

5. EXPERIMENTAL RESULTS

Experiments were conducted on a single-processor, 166MHz Pentium PC equipped with CCD camera and Coreco Ultra II frame grabber. Experimental results show that we can locate and track the eyes, the nostrils, and lip-corners in images with different resolutions and different illuminations in real-time(12+ Hz) as soon as the face appears in the field of the view of the camera. The accuracy is above 95% without any identifying mark on the user's face. We have also tested person wearing the glasses or not wearing the glasses. In the case of the subject' black glasses, unsatisfactory results are returned. And if the people have a mustache, then we can not locate the mouth exactly. You can see the result of gaze tracking using head orientation in Figure 9. The rectangles mean the gaze points.

6. CONCLUSIONS AND FUTURE WORK

Real-time facial feature tracking and head pose estimation for eye and head controlled human computer interface has been proposed. More intelligent gray-level threholding methods and verification techniques are desirable and more distinct features for head orientation must be included into the final target system. We can also expect better interface in car navigation systems that incorporate eye-gaze information.

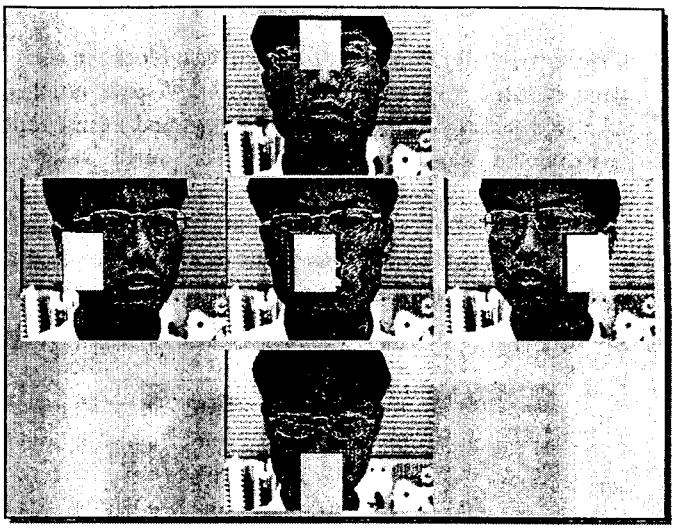

Figure 9 Experimental results of gaze tracking

References

[1] Rajeev Sharma, Vladimir I. Pavlovic, and Thomas S. Huang, "Toward Multi-modal Human-Computer Interface," Proc. IEEE, vol. 86, pp. 853-869, May, 1998.

[2]Kwon-Il Kang, Jong-Hei Ra and Moon-hyun Kim, "Tracking the Eye Trajectories in dynamic images using adaptive template matching," Proc. of the 9th Spring Conference of Korea Information Processing Society, 1988.

[3]A. L. Yuille, D. S. Cohen and P. W. Halinan, "Feature extraction from face using deformable template," Proc. IEEE Computer Soc. Conf. on Computer Vision and Patt. Recog., pp. 104-109, 1989.

[4]K. C. Yow, R. Cipolla, "Finding initial estimates of human face location," Proc. 2nd Asian Conf. on Computer Vision, vol. 3. Singapore, pp. 514-518, 1995.

[5] T. Leung, M. Burl and P. Perona, "Finding faces in cluttered scenes using labelled random graph matching," Proc. 5th Int. Conf. on Comp. Vision, MIT, Boston, pp. 637-644, 1995.

[6] Y. Sumi, Y. Ohata, "Detection of face orientation and facial components using distributed appearance modeling," Proc. Int. Workshop on Automatic Face and Gesture Recognition, Zurich, pp. 254-259, 1995.

[7]Q. Chen, H. Wu and M.Yachida, "Face detection by fuzzy pattern matching," Proc. 5th Int. Conf. on Comp. Vision, MIT, Boston, pp 591-596, 1995.

[8] Dai and Y. Nakano, "Face-texture model-based on SGLD and its application in face detection in a color Scence," Pattern Recognition, 29(6):1007-1017, 1996.

[9] H. A. Rowley, S. Baluja and T. Kanade, "Human face detection in visual scenes," Technical Report CMU-CS-95-158, CMU, July, 1995.

[10] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," Technical Report A.I. Memo 1521, CBLC Paper 112, MIT, 1994.

[11] T.I.P. Trew, R.D.Gallery, D.Thanassas, E.Badique, "Automatic face location to enhance videophone picture quality," Proc.4th Brit. Machine Vision Conf., Springer-Verlag, , pp. 488-497, 1993.

[12] R. Lopezand and T. S. Huang, "3D Head pose computation from 2D Images:Templates vs Features," ICIP, IEEE Washington DC, pp. 220-224 , Oct. 1995.

[13] X. Xie, R. Sudhakar and H. Zhuang, "Estimation of eye features from facial images," Proc. 4th Annu. Conf. Recent Advances Robot., Boca Raton, FL, pp. 73-80, 1991.