Real-time Face Tracking based on Facial Feature Matching

Wei-Ming Chen*, Chiou-Shan Chou*, Yi-Lung Lin*, Hao-Chun Wang* and Chia-Hung Yeh†

*National Ilan University, Yilan
Email: wmchen@niu.edu.tw, r9843003@ms.niu.edu.tw, dennis.bruces@msa.hinet.net, milk.lord@gmail.com
†National Sun Yat-sen University, Kaohsiung
Email: yeh@mail.ee.nsysu.edu.tw

***Abstract*—In this paper, we propose a real-time system to extract and track people's facial features effectively. It can also resist rotation, scaling, and parallax of the image. When camera captures video frame, the proposed system can recognize where the face is, and then uses our Dynamic Radial Kernel to record and match facial features in each frame. After getting all the facial features from those frames, we can realize what the user's movement is happening, such as face direction changing, face rotation, and depth changing, because every frame is in the same time sequence. At last, we map the 2D coordinate to 3D space by perspective transform. The experimental result shows that the proposed method is successful. It can recognize human facial features in several environments robustly. In addition, we implement a human interface system using the proposed method to display an augmented reality (AR) application. In the future work, we will try to improve our algorithm of feature recording, matching, and make it suitable to any content of image.**

I. Introduction

A. Related Work

In computer graphing, colors are described by RGB color model, but it is easily influenced by illumination [1]. If we want to analysis an image robustly, we have to overcome this situation first, so there have been several studies about color model conversion. NCC (Normalized Color Coordinates) [2][3] is a solution in general. It can separate the brightness from original color. Using YCbCr is another way. The advantage of YCbCr model is that it has a specific component Y to storage the brightness value, and the conversion between RGB and YCbCr is not too complex. Garcia and Tziritas [4] proposed a function to decide a color is a skin color or not. After that, YCbCr has been used in face detection and tracking very often [5][6][7]. However, HSV [8] describes colors by hue, saturation, and value, and it is more intuitional to the color sense of human. Thus HSV has better ability of correcting error derived from shadows. But the drawback is that the conversion between RGB and HSV is much more complex than others.

On the other hand, before tracking a face, we have to know where the face is. Some studies use color-based algorithm to search faces. Soriano et al. [9] proposed a Nogatech camera skin locus. It makes us easily to find a skin region. Some studies use template-based algorithm, because everyone's facial feature is arranged in the same regulation. The region of eyes will be darker, and the region of skin will be brighter. According to it, Viola et al. [10] proposed the integral image to find faces rapidly.

At last, we need to extract facial features. Some researchers, such as He et al. [11] and Minagawa et al [12], proposed methods to locate facial features, but there is a restrict of them: the features may disappear when user turns or nods his head. Then, we think we have to find some robust facial feature to prevent this kind of problem. The Landmark Detector [13], the Most Likely-Landmark Locator (MLLL) [14], and the Viola-Jones detector [15] have shown that there are several strong features on our face like canthi, nostrils, and corners of mouth. These points may provide the best robustness of our feature matching and tracking.

B. Our Goal

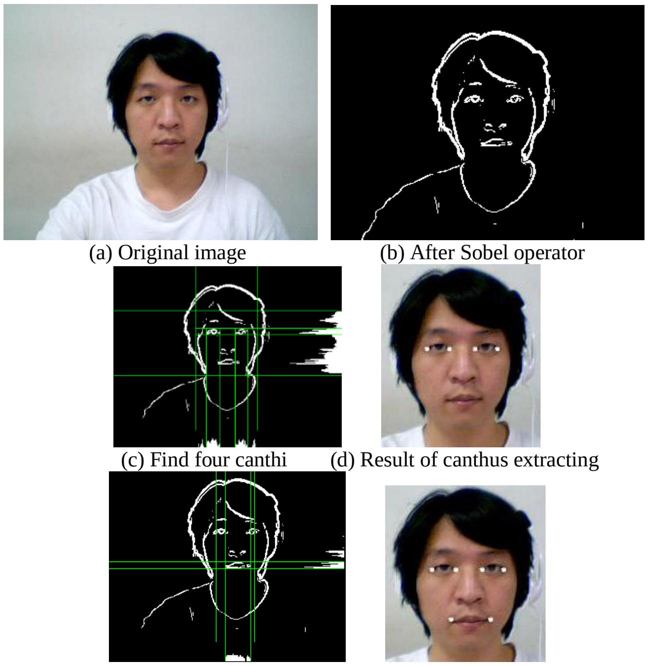
We propose a new approach to record data of facial features, and it can match another image to find the most similar region. Our goal is to implement a system which uses our algorithm and satisfies the following requirements:

- Working in real time
- Get the features automatically
- Rotation resistance when matching
- Scaling resistance when matching
- Parallax resistance when matching
- Analysis user's movement

II. Facial Features Recording

A. Features Extracting

In our system, what should be done first is to record user's facial feature. Thus, we need to know where the features are. Here we use a method proposed by Minagawa et al. [12] which is shown in Fig. 1. It uses an edge detector to filter obvious edges on user's face, and then extracts the positions of four canthi and two corners of mouth. These six feature points will be tracked in the following steps.



(a) Original image (b) After Sobel operator

(c) Find four canthi (d) Result of canthus extracting

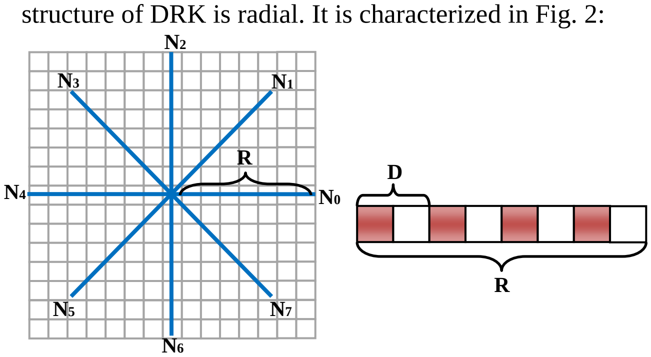
(e) Find two corners of mouth (f) Result of mouth extracting

Fig. 1. Extract six facial features we need.

B. Feature Recording

Most of approaches for facial feature detecting assume that user is face to camera directly. It will cause a problem: if user turns or nods his face slightly, the features may disappear. Therefore, we propose a novel feature recording method to solve it. It is called Dynamic Radial Kernel (DRK).

After getting a feature's coordinate, we use the DRK to record pixels near the feature point. As its name, the structure of DRK is radial. It is characterized in Fig. 2:



(a) A whole DRK. (b) One radiation of DRK.

Fig. 2. Structure of DRK.

We can find some rules of DRK: there are totally N radiations with length R in a DRK marked in blue shown in Fig. 2(a). Each radiation records R/D pixels. The recorded pixels are marked in red shown in Fig. 2(b).Normally, we set the default value of N is 8, R is 14, and D is 2. Each pixel in DRK will be recorded its color information. Here we use the HSV color model to describe each pixel. Sometimes, because of the illumination around the user, there will be some shadow on user's face, and it may bring errors when feature matching. HSV has good ability of correcting error derived from shadows. Color in different brightness will only affect its V value, so we just record the H and S value to prevent error shows up. In addition, we also need to record distance between each feature point. It helps our system to know what the scaling ratio is, and what the relative positions of each two feature points are.

III. FACIAL FEATURES MATCHING

Once DRK records nearing pixels' color information as we mentioned above, it has ability to match up other video frames. The pseudo code shown in Fig. 3 summarizes the matching algorithm:

```
function Matching(x, y)
   error←0
   for n←0 to DRK.N-1
      for p←0 to DRK.R / DRK.D
         error←error+
               Abs( DRK.color_info[n, p].H -
                    Img[x+posx(n, p), y+ posy(n, p)].H )+
               Abs( DRK.color_info [n, p].S -
                    Img[x+posx(n, p), y+ posy(n, p)].S )
      end for
   end for
   return error
end function

procedure Find_Feature(pre_x, pre_y)
   min_ error←∞
   for i← (Window_h-1) *z / -2 to (Window_h-1) *z / 2
      for j←(Window_w-1) *z / -2 to (Window_w-1) *z / 2
         this_error←Matching (pre_x+i, pre_y+j)
         if this_error < min_error then
            min_error←this_error
            new_x←i
            new_y←j
         end if
      end for
   end for
   pre_x←new_x
   pre_y←new_y
end procedure
```

Fig. 3. Pseudo code of DRK feature matching.

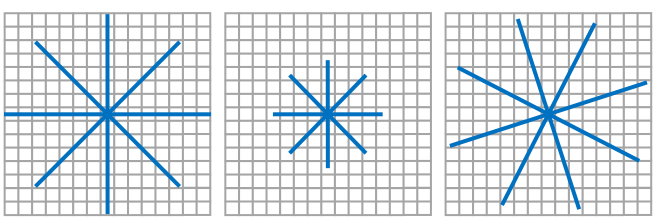
In the *Find_Feature* procedure, **Img** is the frame in processing, and its pixels described by HSV color model. After extracting a facial feature point, its coordinates will be recorded as well. The x and y will be stored in variance **pre_x** and **pre_y**. Then, we can define a detecting window according to the **pre_x** and **pre_y**. The variance **z** is the scaling ratio of face. If **z** increases, the detecting window gets bigger, and vice versa. All pixels in this window will be processed using the *Matching* function. The Matching function computes the difference between each pixel and facial feature's color information recorded in DRK.

In the *Matching* function, we compare the original color information and the processing pixel's nearing pixel. The difference will be computed in variance **error**, and returned to *Find_Feature* procedure. Therefore, the key point when matching is to know which pixel in the **Img** needs to be compared. In our algorithm, we propose two functions, *posx* and *posy*, to get the compared pixel's coordinate. These two functions have two parameters **n** and **p**. **n** represents the **n**th radiation of DRK, and **p** represents the **p**th recorded pixel in the radiation. With the two parameters, the compared pixel's coordinate can be computed using (1) and (2):

$$posx(n,p) = \cos\left(\frac{360^\circ}{N} \times n + t^\circ\right) \times D \times p \times z \,. \quad (1)$$

$$posy(n,p) = \sin\left(\frac{360^\circ}{N} \times n + t^\circ\right) \times D \times p \times z \,. \quad (2)$$

Variance **N** and **D** have been explained in Fig. 2. Variance **z** is the scaling ratio. DRK is dynamic, and it can adjust its size according to **z**. When the **z** value increases, the computed result is much bigger, and vice versa. Variance **t** is the degree of face tilting. If our system detects the face is rotating, the compared region can change as well. The computing of **z** and **t** will be discussed in next section. Fig. 4 characterizes the different DRKs:



(a) DRK with t=0, z=1. (b) DRK with t=0, z=0.5. (c) DRK with t=20, z=1.

Fig. 4. Dynamic region.

Every pixel in the detecting window computes its error from DRK. After all pixels are processed, the point with the smallest error is the new feature. The *Find_Feature* procedure will repeat six times to find six new feature points and replace the six original feature points, and then the matching is finished at this frame.

IV. Movement Analysing

Video is composed from a continuous image, so the contents of each image are continuous, too. According to these continuous data, we can simply analyzing user's movement. We define three kinds of movements, and they will be explained in the following paragraph.

A. Face Closing or Leaving

When user closes to camera, user's face must get bigger, and vice versa. Therefore, if the distances of each feature points become bigger or smaller at the same time and the same ratio, we define this phenomenon is closing or leaving. In our system, we use the two outer canthi to compute scaling ratio. At first, after extracting the facial features, it records the distance between outer canthi as a reference distance D1, and when it is tracking, it records the distance between tracked outer canthi as current distance D2. Then the scaling ratio is D2 / D1. The ratio also determines the variance **z** which we computed in section III. Fig. 5 characterizes this movement:

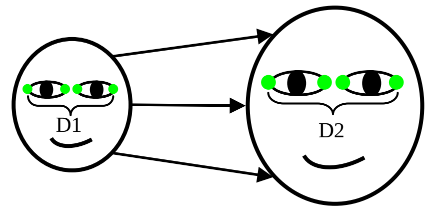
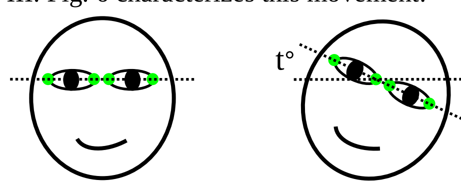


Fig. 5. The scaling rate is simply computed from D2 / D1. The green points represent the tracked canthis.

B. Head Tilting

In general case, the four canthi on human face are arranged in a straight line. The slope of this line defines the degree of tilting, and it also determines the variance **t** in section III. Fig. 6 characterizes this movement:



(a) No tilting (b) Tilt t degrees

Fig. 6. The slope of canthal line determines the tilting angle. The four canthi are marked as green points.

C. Head Shaking

Assume that human head is a sphere, the facial features are points on the surface of this sphere. According to sine and cosine function, we understand a phenomenon that the ratio of length of left eye to length of right eye changes with the rotation angle of user's head. Fig. 7 is a top view, and it characterizes this movement:

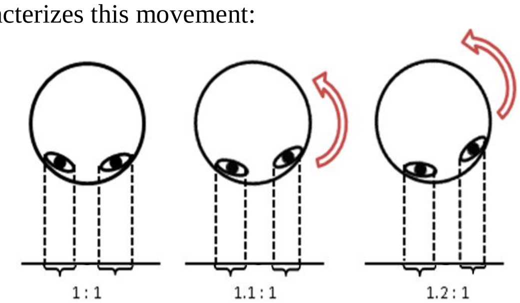
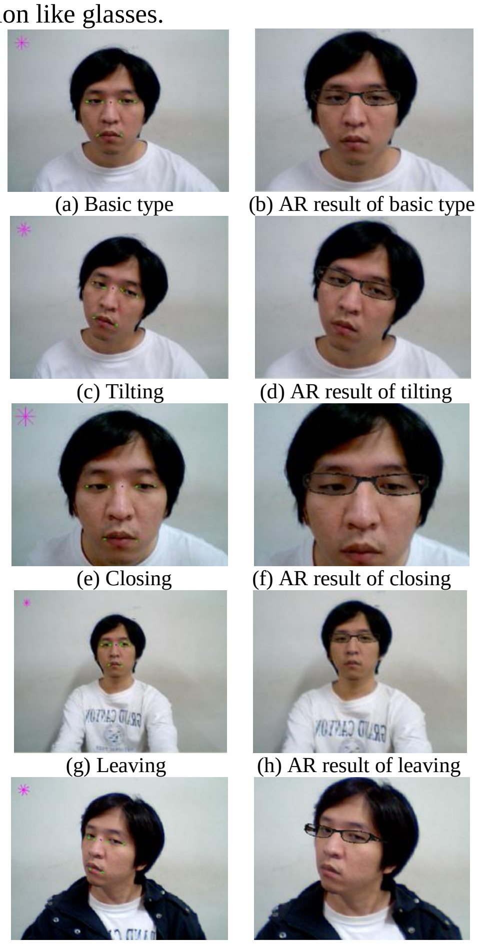


Fig. 7. The ratio changes when user's head shakes.

V. EXPERIMENTAL RESULT

Our prototype system uses a stand-alone PC with 1.96GB RAM and 2.66 GHz CPU. The captured device is Microsoft Webcam, its captured frame format is 320x240, and frame rate is 30 frames per second. This resolution is not high, but it is high enough to tracking facial features in mostly situations. However, if the degree of rotation is too high, DRK can't match features accurately.

In the results, we have successfully demonstrated that the system can track user's facial features under various movements and environments. Besides, our system has additional ability to compute a 3D rendering using OpenGL. It generates a pair of glasses 3D model on user's face, and its position and direction change with user's movement. It means our system becomes a useful Human Machine Interface (HMI) to help people wearing a virtual facial decoration like glasses.



(a) Basic type (b) AR result of basic type
(c) Tilting (d) AR result of tilting
(e) Closing (f) AR result of closing
(g) Leaving (h) AR result of leaving
(i) Head Shaking (j) AR result of head shaking

Fig. 8. Experimental results. (a)(c)(e)(g)(i) show the results of matching marked in green point, and the DRK is demonstrated in the upper left corner. (b)(d)(f)(h)(j) are the AR results.

VI. CONCLUSION

In this paper, we implement a real-time system which can record human facial features, and search the most similar region in another frame using our DRK. Most of the face tracking approaches can only track face when user is facing to camera directly. To overcome this problem, DRK provides a flexible structure to optimize matching region in each frame. Through analyzing user's movement, we can compute what is the most suitable structure of DRK in a specific frame. It increases the accuracy of our system. At last, we build an HMI system to generate a virtual decoration on user's face to prove our algorithm is accurate enough, and the costing is pretty low.

The DRK matching is still under development. We are still improving this approach. In the feature extracting step, we use Minagawa's method. It still has some restricts. We are still trying to design a new method to prevent these restricts. On the other hand, DRK is now only used in human face image, so the other further effort is trying to use the DRK in other content. To achieve this purpose, we may need to adjust the structure or matching algorithm of DRK. It will make our approach more useful and flexible.

REFERENCES

[1] B. A. Maxwell, R. M. Friedhoff, and C. A. Smith, "A bi-illuminant dichromatic reflection model for understanding images", Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2008.

[2] Soriano, M., B. Martinkauppi, S. Huovinen and M. Laaksonen, "Skin Color Modeling under Varying Illumination Conditions Using the Skin Locus for Selecting Training Pixels," Proc. IEEE Nordic Signal Processing Symposium, Kolmården, Sweden, 2000.

[3] M. Soriano, B. Martinkauppi, S. Huovinen and M. Laaksonen, "Skin Detection in Video under Changing Illumination Conditions," Proc. 15th International Conference on Pattern Recognition, Barcelona, Spain, 2000.

[4] C. Garcia and G. Tziritas, "Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis", Proc. IEEE Trans. On Multimedia, Vol. 1, No. 3, pp. 264-277, 1999.

[5] A. P. Chen, L. A. Pan, Y. B. Tong, and N. Ning, "Face Detection Technology Based on Skin Color", Proc. Second International Workshop on Education Technology and Computer Science, Vol. 2, pp. 708-711, 2010.

[6] K. M. Kryszczuk and A. Drygajło, "Color Correction for Face Detection Based on Human Visual Perception Metaphor," Proc. of the Workshop on Multimodal User Authentication, pp.138-143, 2003.

[7] Y. P. Lin, Y. P. Chao, C. C. Lin, and J. H. Chen, "Webcam Mouse Using Face and Eye Tracking in Various Illumination Environments," Proc. 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, 1-4, Sep., 2005.

[8] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd ed. Prentice Hall Press, 2002.

[9] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen, "Using the Skin Locus to Cope with Changing Illumination Conditions in Color-based Face Tracking," Proc. IEEE Nordic Signal Processing Symposium , pp. 383–386, 13-15, June, 2000.

[10] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, pp. 511, 2001.

[11] X. He, Z. M. Liu, and J. L. Zhou, "Real-Time Human Face Detection in Color Image," Proc. Second International Conference on Machine Learning and Cybernetics, pp. 2915-2920, 2-5, Nov., 2003.

[12] T. Minagawa, H. Saito, and S. Ozawa, "Face-Direction Estimating System Using Stereo Vision," Proc. International Conference on Industrial Electronics, Control and Instrumentation, pp. 1454-1459, 1997.

[13] G. M. Beumer, Q. Tao, A. M. Bazen and R. N. J. Veldhuis, "A Landmark Paper in Face Recognition", Proc. International Conference on Automatic Face and Gesture Recognition, pp.73–78, 2006.

[14] A. Bazen, R. Veldhuis and G. Croonen, "Likelihood Ratio-Based Detection of Facial Features", 14th Annual Workshop on Circuits, Systems and Signal Processing, pp.323–329, 2003.

[15] P. Viola and M. Jones, "Robust Real-Time Object Detection", International Journal of Computer Vision, 2002