SHAPE CONTEXT BASED MATCHING FOR HAND GESTURE RECOGNITION

[1]Lawrence Y. Deng and [2]Dong-Liang Lee
Department of [1]CSIE and [2]IM,
St. John's University
[1]Lawrence@mail.sju.edu.tw

[3]Huan-Chao Keh and [4]Yi-Jen Liu
Department of Computer Science and
Information Engineering
Tamkang University
[4]f08572@ms7.hinet.net

ABSTRACT

The shape context is often taken as a basis for shape matching. It can be regarded as a global characterization descriptor to represent the distribution of points in a set with scale and rotation invariance. In this paper, we developed a perceptual interface for human-computer-interaction based on real-time hand gesture recognition. User could interact with computer program by performing body gesture instead of physical contact. We use a shape context based approach for matching hand gestures. First, image of hand gesture was captured from video device. Then, shape information was extracted through computer vision techniques. After that, the hand gesture image was transformed into proper instruction according to the shape information. Finally, the instruction was transferred to an appropriate program to excute.

1. INTRODUCTION

As our technology rapidly advances, mouse and keyboard are not the only two necessities to control computer. Many substitutions for mouse and keyboard are being hard developed, such as speech recognition and gesture recognition.

In this paper, the 'Virtual Petting Game' was taken as an example. We tried to present a novel interaction style on this game. By computer vision techniques, users could play game more intuitively and could interact with computer game without mouse and keyboard.

This computer game system captured the image of user's hand gesture by video device. Users could hand down an order by performing hand gesture in front of a video camera. During playing with 'Virtual Pet' or applying this interaction style on other games, users would have exceptional experience. Through this mode of manipulation, users could be more active, explored more deeply, had more fun in the virtual world and felt that the game could be a part of real life.

We utilized computer-vision based approach in this study. Users didn't need to mark any colored sign on the hand or wear any glove or sensor.

With only a single digital camera, the image of user's hand gesture was captured and analyzed. The methodology could be divided into five parts:

(1) The image was captured through CCD camera.

(2) The hand gesture image was segmented from the image.

(3) The identification of hand gesture.

(4) Transform hand gesture into proper instruction.

(5) The instruction was transferred to Flash program to excute.

2. RELATED WORK

In this study, we tried to remove the background image and eliminate impropriated pixel from the captured image initially. Then the hand region would be detected and segmented from the image. The contour of hand gesture would be described. The significant information would be extracted by shape context based approach. Using this information, we could search corresponding hand gesture in pre-defined gesture-base, such that user's hand gesture could be transformed into proper instruction to trigger correct action.

There are many involved issues for implementing this system, such as Background Subtraction, Color Tracking, Hand Extraction, Hand Gesture Recognition, Shape Context. We will discuss in further detail in the following section.

2.1 Background Subtraction

For the purpose of background subtraction, we could set up a reference image to describe the conformation for the background model [1]. And the background model would provide the upper limit and lower limit of each pixel's variation. It would be a foreground image if the pixel variation was exceeded those limit value obviously.

It was also caused the pixel's variation value over the upper limit or lower limit if a worse background existed occasionally. We could solve this problem with an observation from background subtraction mask periodically.

Generally, the mask was a binary image that composed with black and white elements that provided foreground and background information individually.

The foreground area was a large adjacent block of white area which was noticed in a user's interaction mode normally. It took about one corner or one to two block area of the image approximately.

There were two kind of worse situations to be considered:

(1) When the situation of the shooting scene was worse, such as the lighting changed slowly, the decorated objects switched or the position of camera was unstable, etc. The mask would contain more small area of extension area for the expected block or image everywhere. This small area could be classified as an unnecessary noise that could be detected by the searching of small individual area with white image value as well.

(2) If the environment condition was improved thoroughly, such as the lighting was switched on/off or the camera was shielded by a large object, we also considered it was a stain without any small area of noise that the whole mask was consisted by a large white stain conditionally.

There were two measure methods for classifying of noisy signal:

(1) We defined a minimal foreground block; it would be a noisy signal if the area was smaller than that area.

(2) We defined a maximum foreground block of relative image; it would be an interference area if the area was bigger than that area.

2.2 Color Tracking

The purpose of color tracking was tracking a moving color area in a target image. It could input the result of extracted background with CAMSHIFT algorithm [2] or Kalman Filter Tracking algorithm [3] alternatively.

2.3 Hand Extraction

We had extracted a hand shape with Canny Edge Detector. It was proposed by J. Canny for the edge detection algorithm. There was a grayscale image for the input, and came out a bi-level image that marked as detected edge with non-zero pixel.

2.4 Hand Gesture Recognition

There were three patterns defined for the hand gesture basically: The Static Hand Poses Gestures, The Simple Hand Path Gestures and The Staged Hand Path Gestures.

(1) The Static Hand Poses Gestures was a single hand pose in a space room invariably. It could be a one symbol in the sign language alphabet system probably.

(2) The Simple Hand Path Gestures was a hand pose that described a normal tracking route to compare with the primitive shape. It could estimate the characteristic coordinate of hand shape, such as the mass of the hand. And it also compared the sketch of produced data to a primitive shape simultaneously.

(3) The Staged Hand Path Gestures was a hybrid gesture that combined with the static hand poses gestures and the simple hand path gestures together. It similar to a vector based trajectories with the poly-lines certainly. Therefore, we could divide up the hand gesture to the control points of multi-lines with localized hand postures naturally.

The relative techniques of these three hand gestures were discussed as followings separately:

(1) The Static Hand Pose Recognition

We had applied the concept of shape context into the static hand pose recognition currently [4][5][6]. The shape context was a new shape descriptor that provided the correspondence recovery of shape information and the identification of shape-based object practically.

A shape context of each point recorded the distributional status of its relative position. It described a shape configuration with a valuable, local descriptor generally. And the shape context simplified the recovery of correspondences of two known shape points mass effortlessly.

The shape context introduced an exhaustive and comprehensive data that could measure the similarity of shape easily. And the descriptors of shape context could accept all shape deformation

without any special landmarks or key points fundamentally.

(2) The Simple Hand Path Recognition

We could describe a tracking route of a moving hand with a projected image on a flat board, and compared it with a simple sketched shape, such as triangle, circle, square or ellipse. Thus we could identify the trajectory or path with the geometric property of the projected image briefly. There were three mainly concepts for the recognition of hand path which shown as followings [7][8]:

- The identifier was depended on the geometry information principally.
- To filter unnecessary shape in a specific standard with decision tree accordingly.
- To differentiate the degrees of certainty of the identified shape with fuzzy logic algorithm every so often [16].

(3) The Hybrid hand path recognition

The method of the hybrid hand path recognition was based on the vector analysis and localized hand pose identification generally. We could describe the tracking and path by user's hand which was according to a group of limited control points that determined by the video capture rate and speed of hand pose. And it was not suitable for the vector analysis directly. Therefore, the key point should be identified in a plotted curve previously.

We could summarize a group of sector by a high curve rate was zero with a multi-sector searching basically. We collected these sectors with its geometry property with the algorithm of Douglas-Peucker approximation and a proper parameter precisely. And the vertex point of shown hand gesture would be verified as soon as it needed.

2.5 Shape Context

Shape context is a new shape descriptor presented by Serge Belongie and Jitendra Malik. They proposed this idea in their paper "Matching with Shape Contexts" in 2000[10]. The shape context describes the coarse arrangement of the shape with respect to a point inside or on the boundary of the shape. It can be used for measuring shape similarity and recovering point correspondences.

2.6 Microsoft Project Natal

Project Natal is the code name for a "controller-free gaming and entertainment experience" by Microsoft for the Xbox 360 video game platform[11]. It enables users to interact with the Xbox 360 without touching a game controller physically through a perceptual user interface using body gestures, spoken commands, or presented objects and images.

3. HAND GESTURE MATCHING BASED ON SHAPE CONTEXT

In this paper, the description of hand gesture was based on the Shape Context algorithm [4]. We also focused on the how to search the possible position of hand gesture and the technique of image comparison simultaneously. It was integrated with the algorithm that included the shape sampling, image shape calculation, calculation of the shape descriptors, and variation of the shape descriptors with cost matrix. The whole idea was input a hand video image from video camera, and positioned the sample matrix of hand gesture from the points of shape context. The following chapter would discuss more for the key process as well.

3.1 Shape Sampling

The first step of shape context analysis was translated the edge elements of image shape to a group of feature points with N value. These points could be inside or outside of the image shape simultaneously. Also, there would not be the key-point of the shape normally, such as an apex. We took these sample shapes with a roughly equal range generally.

For example, we got a shape sample data from the digitized gesture image. Meanwhile, the sampling point of edge elements would be collected as Fig 1. Then, we could calculate every point of shape context for the shape descriptors which would be used in the following analysis,

The ‘C’ would be the collection of all shape context point.

$$C = \{C_1, C_2, ...C_t\}, C_i \in R^2 \quad (1)$$

The t would be the total number of the shape context points. And the D2 would be a two-dimension matrix of $t \times t$:

$$D2_{(i,j)} = (C_i \cdot x - C_j \cdot x)^2 + (C_i \cdot y - C_j \cdot y)^2 \quad (2)$$

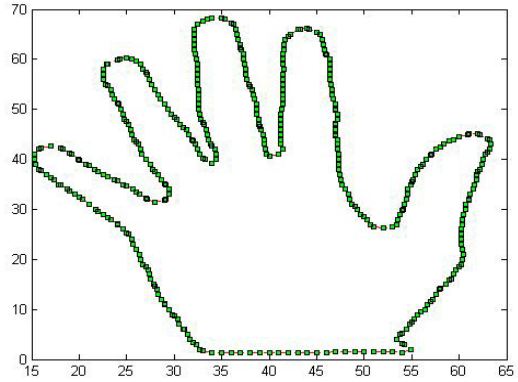

Fig 1. The image of Shape context point

We sampled the number of 'num_sample' points from the shape context with following algorithms:

while (Length(C) > num_sample) {
 //while collection C was bigger than num_sample, the loop would continued.
 [a,b]=min(D2);
 //a and b would be a vector of the row. The 'a' was the minimum value of each row in the D2 matrix, and 'b' was the number of row of the 'a' in that row of D2 matrix.
 [c,d]=min(a);
//The 'c' was the minimum value in the vector 'a'. And the 'd' was the index point of the minimum value in the vector 'a'.
 I=b[d];
//'I' was the number of row of the minimum value in the matrix D2.
 J=d;
//'J' was the number of column of the minimum value in the matrix D2. It deleted the Jth element from C.
//The Jth element was deleted from D2 matrix. It removed the Jth row from D2.
//The Jth column was deleted from D2 matrix.
}
//The 'C' was the point collection of the size of 'num_sample'. It also meant we collected a shape context after sampling with the number of 'num_sample' precisely (Refer to Fig 2).

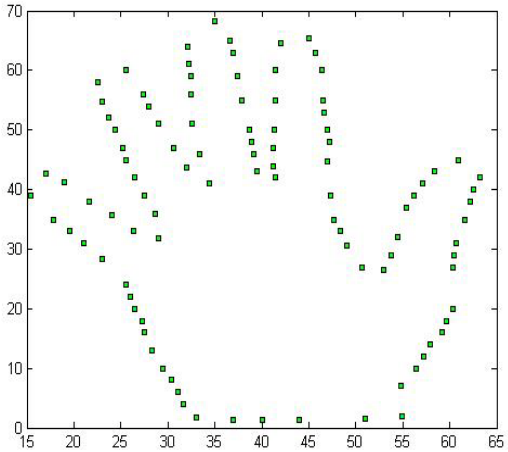

Fig 2. The image of sample points after shape context sampling

3.2 The calculation of distance and angle within sample points

We took the coordinate of each sample point and other n-1 points relatively. And the data was stored into a distance array [r] and an angle array [theta] individually. The calculation of the distance array [r] was defined as following:

The 'r_array' was a $t \times t$ 2-dimension array, the t was the length of the collection of sample points C. And the 'r_array' stored the distance of every point and its next point orderly.

$$r\_array_{ij} = \sqrt{(c_i - c_j)^2} \qquad (3)$$

The calculation of angle array [theta] was defined as following:

The 'theta_array' was a $t \times t$ 2-dimension array, the t was the length of the collection of sample points C. And the 'theta_arrayij' stored the tangent value of every two points and its next point orderly.

theta_arrayij = $\tan^{-1}$(Ci.y - Cj.y, Ci.x - Cj.x) (4)

3.3 Normalized Range

We could defined a 'dist' with a 'mean' function to calculate the average distance of every point. And the 'r_array_n' was a $t \times t$ 2-dimension array that stored the divided value of the distance of each points and its average distance precisely.
dist = mean(r)
The 'r_array_n' was a $t \times t$ matrix. // t = Length(C)
where r_array_nij = r_arrayij/dist

3.4 To divide the distance and the angle into equal parts

3.4.1 The distance was divided into equal parts

We classified the distance into several range with a logarithm method. We resulted a 'nbins_r' points between 'r_inner' and 'r_outer', then stored these points into 'r_bin_edges' simultaneously.

r_bin_edges=logspace(log10(r_inner),log10(r_outer),nbins_r);

The 'r_array_q' and 'fz' were a t × t 2-dimension array. And we classified the distance of each points into 'nbins_r' sections.

```
r_array_qij = 0, fzij = 0
for (m = 0; m < nbins_r; m++)
for (i = 0; i < t; i++)
for (j = 0; j < t; j++)
if (r_array_nij < r_bin_edges(m))
r_array_qij++;
```

The point was a outer boundary if the r_array_n(i,j) was not in the range of r_bin_edges. Also, those were recorded by a 'fz' matrix particularly.

```
for (i = 0; i < t; i++)
for (j = 0; j < t; j++)
if (r_array_qij > 0) fzij = 1;
```

3.4.2 The angle was divided into equal parts

We set the opposite angle of every point to a range between 0 and 2*π. And the 'theta_array_2' was a t × t two-dimension array (where t = Length(C)).

theta_array_2(i,j) = ((theta_array(i,j) mod 2*π)+2*π)

And the 'theta_array_q' was a t × t two-dimension array (where t = Length(C)).
theta_array_q(i,j) = 1+floor(theta_array_2(i,j)/(2*π/nbins_theta));

3.5 Calculation of Shape Descriptors

We tried to describe the shape descriptors with recording the characteristic of an image contour. The image contour was expressed by a series of discontinuous points with value of 'n'. Therefore, there were n-1 points left for the opposite position. And the recorded results could be rotated when the shape contour was rotated somehow. We could group these points by a symbol if the rotated relationship existed. Thus, a 'n' points would be represented by the 'n' symbols. This is how we could find the similarity between these shape contours so quickly.

We applied this comparison technique of shape contour to an existed image capture system. It also provided a related feedback with an area-based image capture method simultaneously. It would be have a high accuracy for the captured image that quite fit with a human thinking model at all.

We could calculate the parameters of shape contour with following program where all elements of BH would set to zero for the initialization.

```
for (n = 0; n < sample; n++)
   for (i = 0; i < t; i++)
      for (j = 0; j < t; j++)
         if (fz(i,j) > 0)
            BH(n,theta_array_qij,r_array_qij)++;
```

The shape context would be represented by the shape information which described with the parameters of each sample point totally.

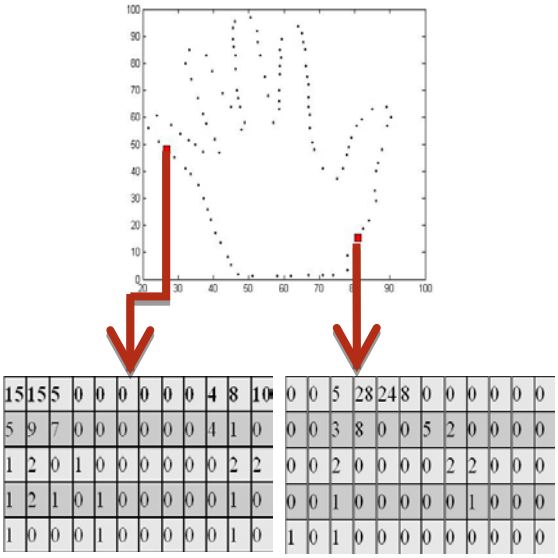

Fig 3. The Shape Context of Hand Gesture

3.6 Cost Matrix

We could calculate the cost of every point from these sample points. Thus, the cost from i point to j point was equal to the Chi-squared similarity of the shape descriptors from i row to j row approximately.

$$C_{(i,j)} \equiv C(p_i, q_j) = \frac{1}{2}\sum_{K=1}^{K}\frac{[h_i(k)-h_j(k)]^2}{h_i(k)+h_j(k)} \quad (5)$$

Now we got all the matched points Ci,j from first shape to second shape. We also minimized the cost of all these pair points with one by one conditionally. It was a problem of the square assignment or the weighted bipartite matching. The complexity of time O(N^3) was involved with Hungarian method probably.
We set a slightly more efficient algorithm in this study [5]. We made a input matrix with Ci,j matrix

instead, and the result was a permutationπ(i), which was ΣiCi , and the π(i) was the minimum value.

4. IMPLEMENTATION

Generally speaking, the study of vision-based hand gesture recognition usually involved image capturing, gesture analyzing and the tolerability of ambiguous hand gesture. Our system captured the image of user's hand gesture by video device. The significant information would be extracted, such that user's hand gesture could be transformed into proper instruction. Finally, the instruction was transferred to an appropriate program to trigger correct action.

4.1 Definition of Hand Gesture

We had designed and implemented an interface between human and computer for the fully hand gesture system in this study. There were five model of hand gestures which defined as system control commands. The user could apply its command with hand gesture to replace any heavy equipment on the body.

The system would grab user's hand movement with video camera automatically. Then, it determined the color range of skin and transferred the specified color range to a binary data. And a smooth processing and a noisy elimination would be applied on those selected data simultaneously. Finally, the data would be compared in the Matlab. The sequential procedures were described as followings:

Step 1: A completed hand was placed statically.

Fig 4. A photo of the grabbed hand gesture

Step 2: The static hand gesture was grabbed and transferred to a binary data.

Fig 5. The binary image of the grabbed hand gesture

4.2 Hand Gesture Capturing

The experiment was setup in a 30 m2 square space with a 10~30 cm range, and the average brightness was 217 lm/m2. We used a CCD camera to grab the hand images for the recognizable gesture with 60 frames per minute. The resolution of these images was up to 640 x 480. (Refer to Fig 7)

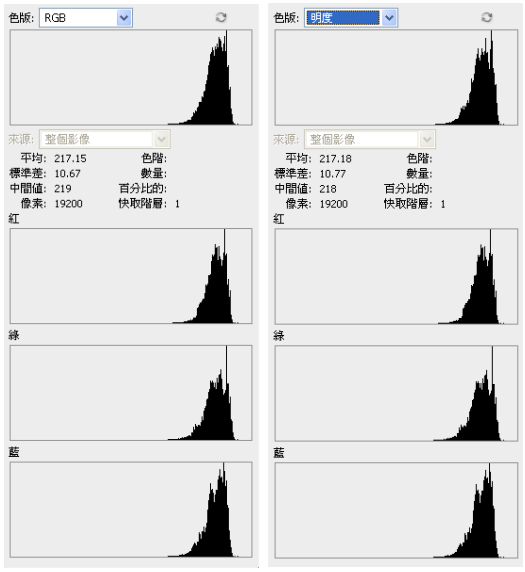

Fig 6. The average brightness
(Left) The fluorescent lamp
(Right) The lamp in the laboratory

Fig 7. The interface of Hand Gesture Grabbing

4.3 Detection of Skin Color Range

4.3.1 Detection of Skin Color

We could locate the hand gesture area by the skin color detection in a normal and simple background simply. Unfortunately, there was much more complicated background actually. Therefore, we need setup some other terms to search the possible gesture area precisely.

According to G. Kukharev, A. Novosielsk's theory, it was determined to a skin color while the value of YCbCr was fitted in with Y>80, 85>Cb<135, 135<Cr<180, where Y,Cb,Cr = [0, 255]. [9]

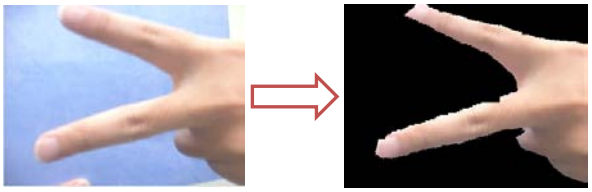

Fig 8.(b) The image of Skin color range

After the detection of color skin, we could easy to identify most of the skin color with the filter actually (Refer to Fig 8).

4.3.2 The Binary Processing of Hand Gesture

It was only need some gray level to determine the threshold value of gesture recognition with its histogram on the processing, such as black and white image. It would be easily to translate a gray level image to a black and white image directly. The definition of the binary image was summarized as followings:

g(x,y) = 1, if f(x,y) > T
g(x,y) = 0, if f(x,y) < T

f(x,y): the original image
T: the threshold value
g(x,y): '1' means black. And '0' means white.

In the consideration of brightness of light and the factors of hardware, the threshold value was given in the skin color range with the adjudgment by YCbCr. And the threshold value were T=45 < Y < 180, 126 < Cb < 143, and 122 < Cr < 130.

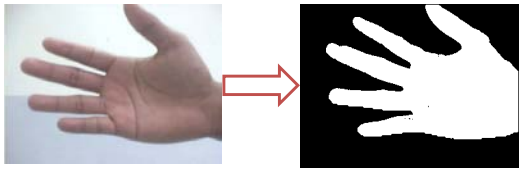

Fig 9. Smooth Processing

4.3.3 Smooth Processing

There were some noisy found in the image process of digitizing of hand gesture. It was very common for the noise signal of a capture image from a normal video camera generally. This also made easier to wrong distinguish for the further binary processing. Therefore, we need to eliminate the noisy by the erosion and dilation process before the binary image processing which would made our digitizing image had more smooth and higher noisy ration concurrently.

4.3.4 Elimination of Noisy Signals

We could find some small skin color points around the hand gesture after the binary imager processing with skin color detection. For the influence avoidance by the further procedures, we would apply the Opening operation in Morphology that could eliminate these minor noisy additionally.

The Opening operation was included erosion and dilation processing, that could narrow a binary image with erosion, then magnified that area image continuously. We also calculated the new binary image with mask operation after the skin color detection both for the erosion and dilation completely.

The Erosion: it determined the value of the position pixel P of the mask that was '1' or not. If it matched the value, the other 8 surrounding points would be determined repeatedly.

That was : $P=P1 \cap P2 \cap P3 \cap P4 \cap P5 \cap P6 \cap P7 \cap P8$

The Dilation: It resembled to erosion. It would determine the value of the position pixel P of the mask that was '1' or not. If it matched the value, the other 8 surrounding points would be determined repeatedly.

That was : $P=P1 \cup P2 \cup P3 \cup P4 \cup P5 \cup P6 \cup P7 \cup P8$

The binary skin color image would eliminate those minor noisy spot after the Opening operation (Refer to Fig 10 and 11).

Fig 10. The elimination of noisy signals with the algorithm: $S = (B \odot S) \oplus S$

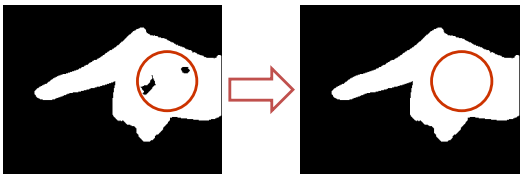

Fig 11. The elimination of noisy signals with the algorithm: B • S = (B⊕S) ⊙ S

5. EXPERIMENTAL RESULT

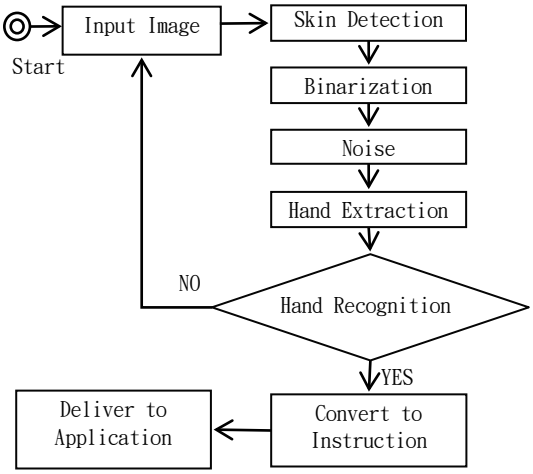

Fig 12. The Diagram of system flow

We used a video camera to shoot the images sequentially. The pixel quality of the images was 640 by 480 precisely, and the average luminance was 217cd/m2. The shooting speed was set to 0.5 frame per second actually.

The first step, we started the BCB to select the interface of video to get hand gesture directly (refer to Fig 13).

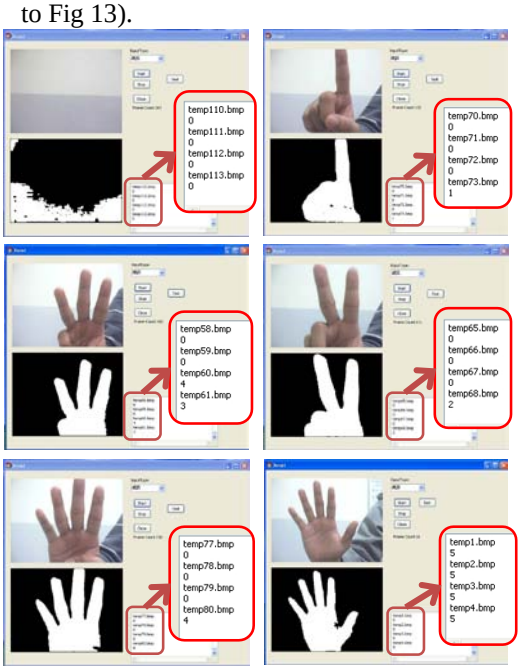

Fig 13. An interface of Hand Gesture

The second step, we select the video image with 'Input Type', then pressed 'Start' to execute the program. And the CCD camera would grab the specified image frame automatically. It also displayed the hand motion and binary skin color range on the left area of the interface menu in the same time (refer to figure 15). There was some function key on the right area, such as 'Start', 'Stop' and 'Close'. Finally, the total count of shooting frames currently was displayed on the bottom of those function keys. Also, the analyzed result was shown on the right bottom area apparently.

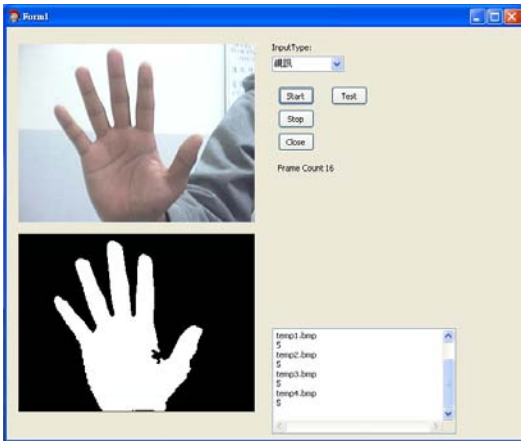

Fig 14. The Interface Menu that displayed the hand motion and the binary skin color range

The third step, the system would pass the analyzed data to the game system. The game system would response the suitable action according to the analyzed data previously. Therefore, the game system would execute a 'Head petting' motion if the hand gesture was analyzed as '3', and it executed 'Bone picking' motion if the hand gesture was analyzed as '4', same as the 'Feeding' motion for the '5' individually (refer to fig 15).

We got the experimental data for 100 times from each hand gesture, and the accuracy was summarized as Table 1.

Table 1. The Accuracy of hand gestures

Gesture Shape	1	2	3	4	5
Precision	70.8%	71.2%	75.0%	89.7%	81.7%
Correct Sample					
Error Sample					

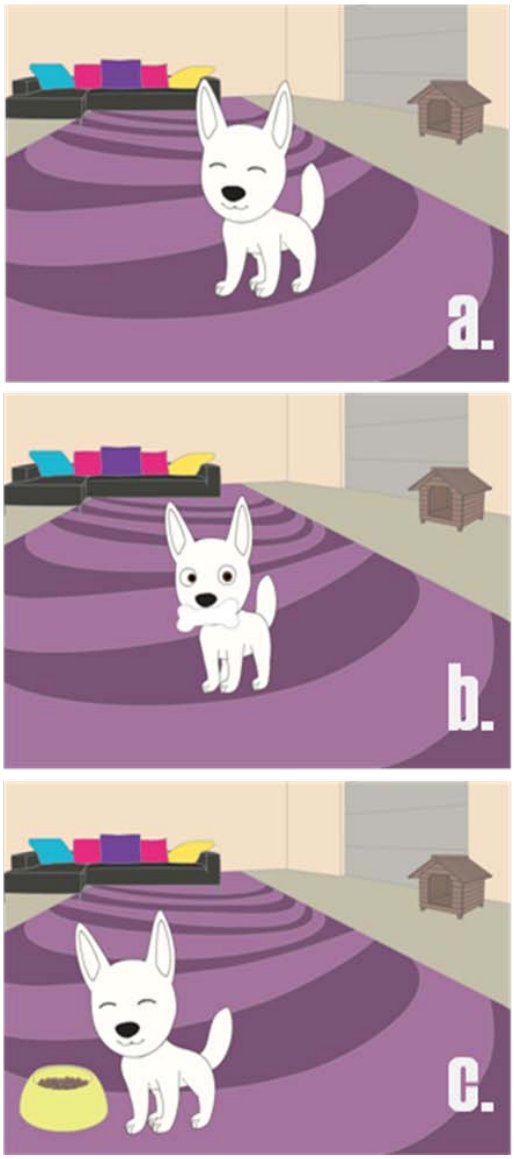

Fig 15. The Movement from Game Image (a) Head petting (b) Bone picking (c) Feeding

6. CONCLUSIONS

In this paper, we developed a perceptual interface for human-computer-interaction based on real-time hand gesture recognition. User could interact with computer program by performing body gesture instead of physical contact. We use a shape context based approach for matching hand gestures. We had proposed a simple and integrated method to recognize the hand gesture from a video image with several procedures, such as skin color detection, noisy signals elimination, comparison of hand gesture, and game command transferring. And it would trigger the control system of the virtual pet game with the previous transferred command consequently. We also solved a shield problem with a single video camera to avoid the influence factors such as the CCD position and the shooting angle. Finally, we had integrated with the game system for the amusements successfully.

7. REFERENCES

[1] Larry S. Davis Thanarat Horprasert, David Harwood. : 'A statistical approach for real-time robust background subtraction and shadow detection', Technical report, Computer Vision Laboratory University of Maryland, 1999.

[2] Gary R. Bradski : 'Intel open source computer vision library overview', 2002.

[3] N. Liu and B.Lovell : 'MMX-Accelerated Real-Time Hand Tracking System', Proceedings of IVCNZ 2001. pp. 381-385.

[4] S. Belongie, J. Malik, and J. Puzicha : 'Shape context: A new descriptor for shape matching and object recognition', In NIPS, pages 831–837, 2000.

[5] S. Belongie, J. Malik, and J. Puzicha : 'Shape Matching and Object Recognition Using Shape Contexts', IEEE Trans. Pattern Analysis and Machine Intelligence, vol.24,no. 4, pp. 509-522,Apr. 2002.

[6] Eng-Jon Ong; Bowden, R.: 'A boosted classifier tree for hand shape detection', Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on 17-19 May 2004 Page(s):889 - 894.

[7] Manuel J. Fonseca Joaquim A. Jorge : 'A simple approach to recognize geometric shapes interactively', Technical report, Departamen to de Engenharia Informática, IST/UTL, 1999.

[8] Ajay Apte, Van Vo, and Takayuki Dan Kimura : 'Recognizing Multistroke Geometric Shapes: An Experimental Evaluation', In Proceedings of the ACM (UIST'93), pages 121{128, Atlanta, GA, 1993.

[9] Udo Ahlvers et al. : 'Model-Free Face Detection And Head Tracking With Morphological Hole Mapping', Germany 2005，http://www.ee.bilkent.edu.tr/~signal/defevent/papers/cr1214.pdf.

[10] S. Belongie and J. Malik. : 'Matching with Shape Contexts', IEEE Workshop on Contentbased Access of Image and Video Libraries (CBAIVL-2000)

[11] http://en.wikipedia.org/wiki/Project_Natal