Real-time Hand Gesture Recognition by Shape Context Based Matching and Cost Matrix

Lawrence Y. Deng  
Department of CSIE, St. John’s University  
Lawrence@mail.sju.edu.tw

Jason C. Hung  
Department of Information Management  
Overseas Chinese University  
jhung@ocu.edu.tw

Huan-Chao Keh, Kun-Yi Lin, Yi-Jen Liu, and Nan-Ching Huang  
Department of CSIE, Tamkang University  
f08572@ms7.hinet.net

Abstract—How to recognize the shape gesture for new human-computer interface without controller required and bring entertainment, games industries and information appliances in new ways. In this paper, we would illustrate a real-time hand gesture recognition system by using shape context matching and cost matrix. The shape context is taken as a basis description 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. The image of hand gesture was captured from CCD. The hand gesture image was transformed into proper instruction according to the shape information respectively. The instruction was transferred to an appropriate program to execute. The experience of our preliminary results shown the precision rates was up to 70% ~ 90%.

Index Terms—Hand Gesture Recognition, Shape Matching, Cost Matrix and Human-Computer Interface

I. INTRODUCTION

As 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 proposed system of scenario contained following possible five steps (as shown in fig.1):

1. The image was captured through CCD camera.
2. The hand gesture image was segmented from the image.
3. Identifying the hand gesture.
4. Recognizing hand gesture and matching with proper instruction.
5. Triggering the entertainment games.

Figure 1. The scenario of real-time hand gesture recognition system

In this paper, we outlined the implementation procedures of real-time hand gesture recognition and then to interact with entertainment game. The related technologies and processes of hand gesture segmentation were illustrated in section II. The section III discussed the hand gesture matching based on shape context and cost matrix. The implementation considerations of the real time hand gesture human-computer interface system were discussed in section IV. Finally, we made a brief conclusion in section V.

II. RELATED TECHNOLOGIES OF HAND GESTURE SEGMENTATION
The first step is trying to segment the background image and eliminate improper 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 were several interesting image processing issues for system implementation that included Background Separation, Color Tracking, Object Segmentation, Hand Extraction, Pattern Recognition, Hand Gesture Recognition, and Shape Context etc.. We will discuss in further detail in the following sections.

The separation of the Background and foreground

For the purpose of background subtraction, we could set up a reference image to describe the configuration 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.

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.

Hand Extraction

We had extracted a shape of hand 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.

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 pose 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]: (a) The identifier was depended on the geometry information principally. (b) To filter unnecessary shape in a specific standard with decision tree accordingly. (c) 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.

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.

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.

III HAND GESTURE MATCHING BASED ON SHAPE CONTEXT
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.

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 figure 2. 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.

\[ C = \{C_1, C_2, ..., C_t\}, C_i \in \mathbb{R}^2 \]  

(1)

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

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

(2)

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

```c
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 figure 3).

![Figure 3. The image of sample points after shape context sampling (200 pixels)](image)

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:

\[
r_{array_{ij}} = \sqrt{(c_i - c_j)^2}
\] (3)

The calculation of angle array \[ \theta \] was defined as following: The \[ \theta_{array} \] was a \( t \times t \) 2-dimension array, where \( t \) was the length of the collection of sample points C. And the \( \theta_{array_{ij}} \) stored the tangent value of every two points and its next point orderly.

\[
\theta_{array_{ij}} = \tan^{-1}(c_{i,y} - c_{j,y}, c_{i,x} - c_{j,x})
\] (4)

Normalized Range

We define 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.

\( l/t = Length(C) \), where \( r_{array_{nij}} = r_{array_{ij}}/dist \)

To divide the distance and the angle into equal parts

To classifying 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. Therefore, the distance was divided into equal parts.

\[
r_{bin_edges} = \text{logspace}(\log10(r_{inner}), \log10(r_{outer}), \text{nbins}_r);
\]

We set the opposite angle of every point to a range between 0 and \( 2\pi \). And the \( \theta_{array_2} \) was a \( t \times t \) two-dimension array (where \( t = Length(C) \)). Therefore, the angle was divided into equal parts.

\[
\theta_{array_2(i,j)} = ((\theta_{array(i,j)} \mod 2\pi) + 2\pi)/2\pi/nbins_{theta});
\]

Calculation of Shape Descriptors

The shape descriptors recorded 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, an ‘n’ point 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 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.

\[
for (n = 0; n < sample; n++)
for (i = 0; i < t; i++)
for (j = 0; j < t; j++)
if (fz(i,j) > 0) BH(n,\theta_{array_2(i,j)},r_{array_2(i,j)})++;
\]

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

\[
C_{\text{ji}} = C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \left( h_i(k) - h_j(k) \right)^2
\]

(5)

Now we got all the matched points \(C_{\text{ij}}\) 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 \(C_{\text{ij}}\) matrix instead, and the result was a permutation \(\pi(i)\), which was \(\Sigma_i C_i\), and the \(\pi(i)\) was the minimum value.

**IV. SYSTEM 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.

**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 cost matrix. The sequential procedures were described as followings:

- **Step 1**: A completed hand was placed statically.
- **Step 2**: The static hand gesture was grabbed and transferred to a binary data.

**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 6)

**Detection of Skin Color Range**

We located 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 configured 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]

**Figure 7. The image of Skin color range**
After the detection of color skin, we could easily identify most of the skin color with the filter actually (Refer to Figure 7).

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:

\[
g(x,y) = 1, \text{if } f(x,y) > T \\
g(x,y) = 0, \text{if } f(x,y) < T \\
f(x,y): \text{the original image} \\
T: \text{the threshold value} \\
g(x,y): '1' \text{ means black. And '0' means white.}
\]

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

**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.

**Elimination of Noisy Signals**

We fund some small skin color points around the hand gesture after the binary image 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.

\[P=P_1 \cap P_2 \cap P_3 \cap P_4 \cap P_5 \cap P_6 \cap P_7 \cap P_8\]

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.

\[P=P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5 \cup P_6 \cup P_7 \cup P_8\]

The binary skin color image would eliminate those minor noisy spot after the Opening operation (Refer to Fig 8 and 9).

The implementation of the process flow was illustrated in figure 10. 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 frames per second actually. The first step, we started the program to select the interface of video to get hand gesture directly (refer to Fig 11).

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 12). 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 11. An interface of Hand Gesture

Figure 12. 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 13). We got the experimental data for 100 times from each hand gesture, and the accuracy was summarized as Table 1.

V. CONCLUSION

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.

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Shape 1</th>
<th>Shape 2</th>
<th>Shape 3</th>
<th>Shape 4</th>
<th>Shape 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>70.8%</td>
<td>71.2%</td>
<td>75.0%</td>
<td>89.7%</td>
<td>81.7%</td>
</tr>
<tr>
<td>Correct Sample</td>
<td><img src="image1" alt="Sample 1" /></td>
<td><img src="image2" alt="Sample 2" /></td>
<td><img src="image3" alt="Sample 3" /></td>
<td><img src="image4" alt="Sample 4" /></td>
<td><img src="image5" alt="Sample 5" /></td>
</tr>
<tr>
<td>Error Sample</td>
<td><img src="image6" alt="Error Sample 1" /></td>
<td><img src="image7" alt="Error Sample 2" /></td>
<td><img src="image8" alt="Error Sample 3" /></td>
<td><img src="image9" alt="Error Sample 4" /></td>
<td><img src="image10" alt="Error Sample 5" /></td>
</tr>
</tbody>
</table>

Figure 13. The Movement from Game Image (a) Head petting (b) Bone picking (c) Feeding

REFERENCES


Yi-Jen Liu Mr. Liu is an P.H.D. candidate in Department of CSIE of TamKang University, since 2002. He received the B.S. and master degree in Computer Science Information Engineering from TamKang University in 2001. His current research interests are Distance Learning, Ontology and Information Systems, Image Processing. The contact address of Mr. Liu is: Department of Computer Science and Information Engineering, TamKang University, Tamshui, Taipei Hsien, 251, Taiwan, R. O. C.

Nan-Ching Huang Mr. Huang is an P.H.D. candidate in Department of CSIE of TamKang University, since 2007. His current research interests are Decision Tree, Data Mining, and Image Processing. The contact address of Mr. Lin is: Department of Computer Science and Information Engineering, TamKang University, Tamshui, Taipei Hsien, 251, Taiwan, R. O. C.

Lawrence Y. Deng Dr. Deng is an assistant professor in Department of Information Engineering of St. John’s University, since 2002. He is also the Director of R&D in St. John’s University since 2004. He received the B.S. degree in Computer Science Information Engineering from TamKang University in 1997, and the M.S. and Ph.D. degrees in Information Engineering from the TamKang University, in 1999 and 2002, respectively. His current research interests are Distance Learning, Distributed Multimedia Presentation, Web Server Architecture, Floor Control Mechanism in communication and Multimedia Database System.

Jason C. Hung is an Associate Professor of Dept. of Information Management, Overseas Chinese University, Taiwan. His research interests include Multimedia Computing and Networking, Distance Learning, E-Commerce, and Agent Technology. From 1999 to date, he was a part time faculty of the Computer Science and Information Engineering Department at TamKang University. Dr. Hung received his B.S and MS degrees in Computer Science and Information Engineering from TamKang University, in 1996 and 1998, respectively. He also received his Ph.D. in Computer Science and Information Engineering from TamKang University in 2001.

Huan-Chao Keh Dr. Keh is a professor in Department of Computer Science and Information Engineering of TamKang University. He is also a Dean of the office of Academic Affairs now. His current research interests contain Data Mining, Database System, Distributed Multimedia Presentation, Web Server Architecture, Mobile Agent and Multimedia Database System.

Lin Kun-Yi Mr. Lin is an P.H.D. candidate in Department of CSIE of TamKang University, since 2001. He received the B.S. and master degree in Computer Science Information Engineering from TamKang University in 2001. His current research interests are Distance Learning, Data Mining, Image Processing. The contact address of Mr. Lin is: Department of Computer Science and Information Engineering, TamKang University, Tamshui, Taipei Hsien, 251, Taiwan, R. O. C.