We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS

BOOK CITATION INDEX

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Automatic Hand-Pose Trajectory Tracking System Using Video Sequences

Yuan-Hsiang Chang and Chen-Ming Chang
Department of Information and Computer Engineering, Chung Yuan Christian University
Taiwan, Republic of China

1. Introduction

1.1 Background

Hand-pose is one of the most important communication tools in human's daily life. In a situation when people from different countries are trying to communicate, they may be able to roughly express their thought through hand-poses or hand-gestures. During an oral presentation, a presenter may use hand-poses as an auxiliary tool to convey his or her idea for technical communication. In practice, the use of "sign language" is a typical example that provides an effective mechanism for exchanging information among deaf people and the hearing society. In essence, "sign language" can be considered as a combination of many hand-poses that define the actual information.

With the continuous advances of speech, image, and video processing techniques, human-machine interaction is constantly making progress over the past decades. For example, speech recognition systems (Cooke et al., 2001; Gales, 1998) are designed to recognize and translate human's speech to text. Handwritten recognition systems (Liu et al., 2003; Palacios & Gupta, 2002; Zhai & Kristensson, 2003) are designed to recognize and translate human's handwritten to text. A palmprint identification system (Zhang et al. 2003) is a biometric approach to recognize human's palmprint for personal identification. A safety vehicle system (Trivedi et al., 2007) can be used to estimate the situation around a vehicle and convey information to warn the driver for potential dangers such that the vehicle safety could be improved. In summary, many of these systems have been found to be feasible in enhancing the human-machine interaction and integrated in daily applications (e.g., cell phones, notebooks, security system, etc.).

1.2 Related research

During the past years, researchers have proposed novel methods for the classification or recognition of hand-poses or hand-gestures. The techniques can be divided into two main categories: image-based approaches and glove-based approaches. The image-based approaches are generally designed to use images as inputs to the system for the hand-pose recognition. In contrary, the glove-based approaches are designed with a special hardware installment and/or sensors (e.g., a data glove) as inputs to the system for the hand-pose recognition. The two approaches can be described as follows.

1.2.1 Image-based approaches

Athitsos and Sclaroff presented an appearance-based framework for hand shape classification (Athitsos & Sclaroff, 2002). Given an input image of a segmented hand, the objective was to classify hand shapes by finding the most similar matches using a large database of synthetic hand images. Wachs et al. described the issue of reconfigure ability of a hand-gesture recognition system (Wachs et al., 2005). They addressed the difficult problem of simultaneous calibration of the parameters of image processing and fuzzy C-means (FCM) components of a hand-gesture recognition system. Froba & Ernst proposed a method based on the Modified Census Transform (MCT) for face detection (Froba & Ernst, 2004). Their method was further applied by Just et al. to the hand posture classification and recognition tasks with success (Just et al. 2006). Malima et al. proposed a fast algorithm for automatically recognizing a limited set of gestures from hand images for a robot control application (Malima et al., 2006). Their approach contained steps for segmenting the hand region, locating the fingers, and finally classifying the gesture. Argyros and Lourakis presented a vision-based interface for controlling a computer mouse via two-dimensional (2D) and three-dimensional (3D) hand gestures (Argyros & Lourakis, 2006). In their research, 2D and 3D vocabularies were based on intuitiveness, ergonomics, and ease of recognition criteria. However, only the first two factors were included with the authors' own consideration. Chen and Chang presented an image-based hand-pose recognition system (Chen & Chang, 2007). Their system combined both the shift-distances and the Fast Fourier transform (FFT) features to recognize a set of different hand-poses. Furthermore, both weak and strong classifiers were used to improve the classification accuracy.

1.2.2 Glove-based approaches

Fang et al. used the CyberGlove and presented an additional layer to enhance the hidden Markov models (HMM) architecture with self-organizing feature maps (SOFM), while introducing a fuzzy decision tree in an attempt to reduce the search space of recognition classes without loss of accuracy (Fang et al., 2004). Gao et al. used the CyberGlove and presented a SOFM/SRN/HMM model for signer-independent continuous sign language recognition (SLR) (Gao et al., 2004). This model applied the improved simple recurrent network (SRN) to segment continuous sign language in terms of transformed SOFM representations, and the outputs of SRN were taken as the HMM states in which the lattice Viterbi algorithm was employed to search for the best matched word sequence. Su et al. created a new data glove and presented a SOMART system for the recognition of hand gestures (Su et al., 2006). In addition, the concept of SOMART system could also be applied to hand movement trajectory recognition. Heumer et al. presented a comparison of various classification methods for the problem of recognizing grasp types involved in object manipulations (Heumer et al., 2007).

Because of the flexible structure of human hands, the implied information can be very different in terms of shapes of hand-poses, locations of human hands, or trajectory. The aforementioned systems focused on recognizing hand-postures by extracting features such as hand-shapes from single image with success. However, single hand-pose may not be sufficient for fully interpreting the dynamic information of the human user. In this regard, a number of researches have also been investigated for hand tracking (Chen et al., 2003; Shan et al., 2004; Stenger et al., 2006). Instead of using single image for hand-pose recognition, the techniques could be used for capturing the dynamic information by tracing the locations of human hands in a sequence of images (i.e., video).

1.3 Motivation and objective

To date, personal computers are mainly designed to use a mouse or a keyboard as the input device to interact with human users. Home television or entertainment system often requires a remote control as the input device to receive control commands from human users. We anticipate that human-machine interaction could be greatly improved using a video device (e.g., a webcam) as the sole input device that automatically captures the human's motion (e.g., hand-poses or gestures) and interprets the implied information of the human user. The design is typically aimed for the goal that the input device could be free of direct contact with the human user to improve the convenience (Graetzel et al., 2004).

In this content, we propose an "automatic hand-pose trajectory tracking system using video sequences." The objective is to automatically determine the hand-pose trajectory using image and video processing techniques. The system can be used to input video data and analyze the hand-pose in each frame, quantitatively characterize the hand-pose, and determine the hand-pose trajectory of fingertips with the assumption that the hand-pose remains invariant. Furthermore, the system is designed with an attempt to determine the hand-pose trajectory using video sequences such that the human user does not need to wear special motion sensors or markers. As a result, the hand-pose trajectory could ultimately be used as an input to a user-interface that is able to interpret the given information for computer or machine control.

2. Method

Our system is designed to process two-dimensional (2D) video sequence with single video camera. The system hypotheses include the following:
1.   The human hand must be the dominant object in images (frames) of the video sequence;
2.   The hand-pose can be formed by either front or back of the palm, but without overlapping or crossing fingers;
3.   Hand-pose is defined with a bare hand and not occluded by other objects; and
4.   A still video camera is required with sufficient environment illumination.

Fig. 1 shows an example of the automatic hand-pose trajectory tracking yielded by our system. The hand-pose trajectory is defined as the motion path of the fingertip (index finger) with the assumption that the hand-pose remains invariant during motion.

Fig. 2 shows the terminology of a human hand used in our system. In general, the human hand consists of the arm, the palm, and the fingers in a hand image. Hand-pose is defined as the hand shape formed by the palm and fingers only, regardless of the arm. The palm is located at the interior region, while the fingers are located at the exterior region of the human hand. The center axis is defined as the straight line passing through the geometric center of the hand-pose. In addition, a hand-pose trajectory is defined as the motion path of a fingertip. Therefore, multiple fingertips can generate multiple hand-pose trajectories.

Fig. 3 shows a simplified flow chart of our "automatic hand-pose trajectory tracking system using video sequences". The main processes include preprocessing, segmentation of palm and fingers, feature extraction, and trajectory tracking. The preprocessing is used to determine the location of the hand-pose, while removing irrelevant information (e.g., noise, background, and arm). A rule-based approach is then proposed for the segmentation of palm and fingers in an attempt to isolate each finger from the palm. In addition, the hand-pose is further characterized with a set of features (e.g., number of fingers, fingertip's coordinates, etc.). Finally, if the hand-pose remains invariant during motion, the system is
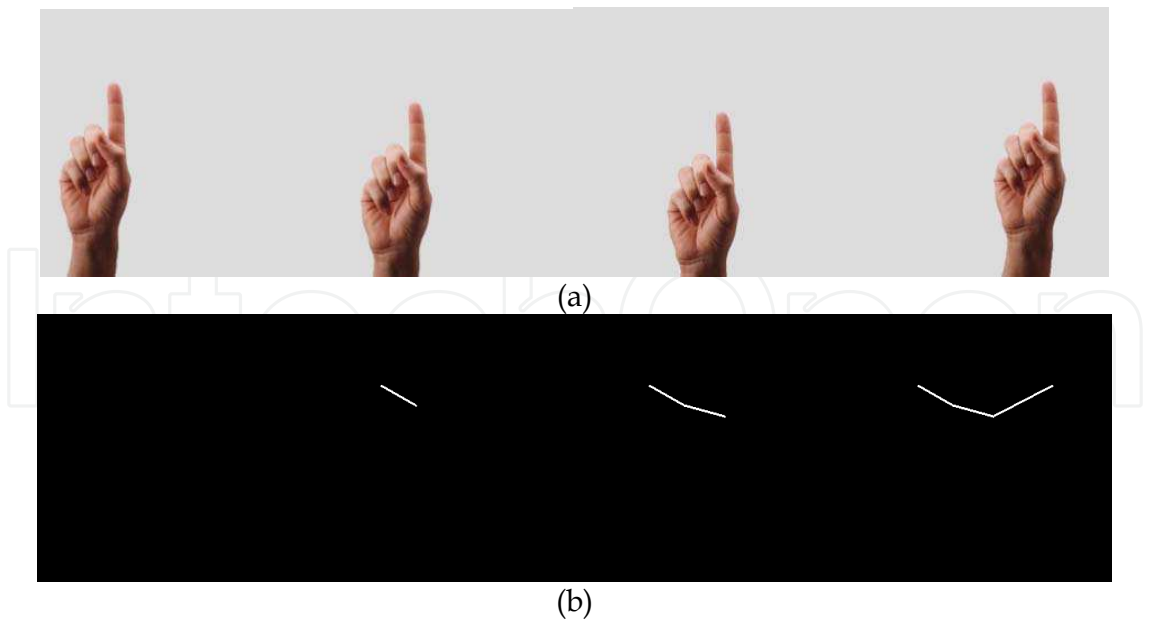
(a)

(b)

Fig. 1. An example of the hand-pose trajectory tracking yielded by our system. (a) Original video sequence with a human hand in motion; (b) Resulting hand-pose trajectory as defined by the motion path of the fingertip (index finger) with the assumption that the hand-pose remains invariant during motion.
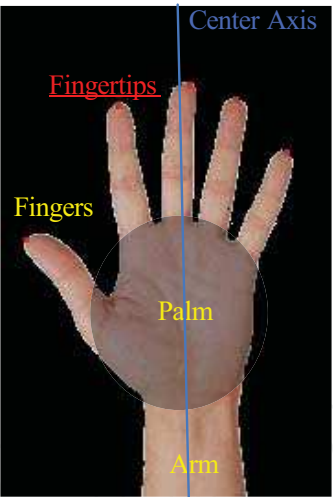

Fig. 2. Terminology of a human hand used in our system, where the human hand consists of the arm, the palm, and the fingers, respectively.

aimed to trace the hand-pose trajectory of the fingertip. In our system, the three processes (i.e., the preprocessing, the segmentation of palm and fingers, and the feature extraction) are performed in a frame-by-frame basis, while the trajectory tracking is performed to record the hand-pose trajectory in the whole video sequence.

2.1 Preprocessing

The objective of the preprocessing is to identify the hand-pose region in a hand image (frame), while removing irrelevant information (e.g., noise, background, and arm). The processes include gray-level transformation, smoothing, edge detection, hand-pose contour
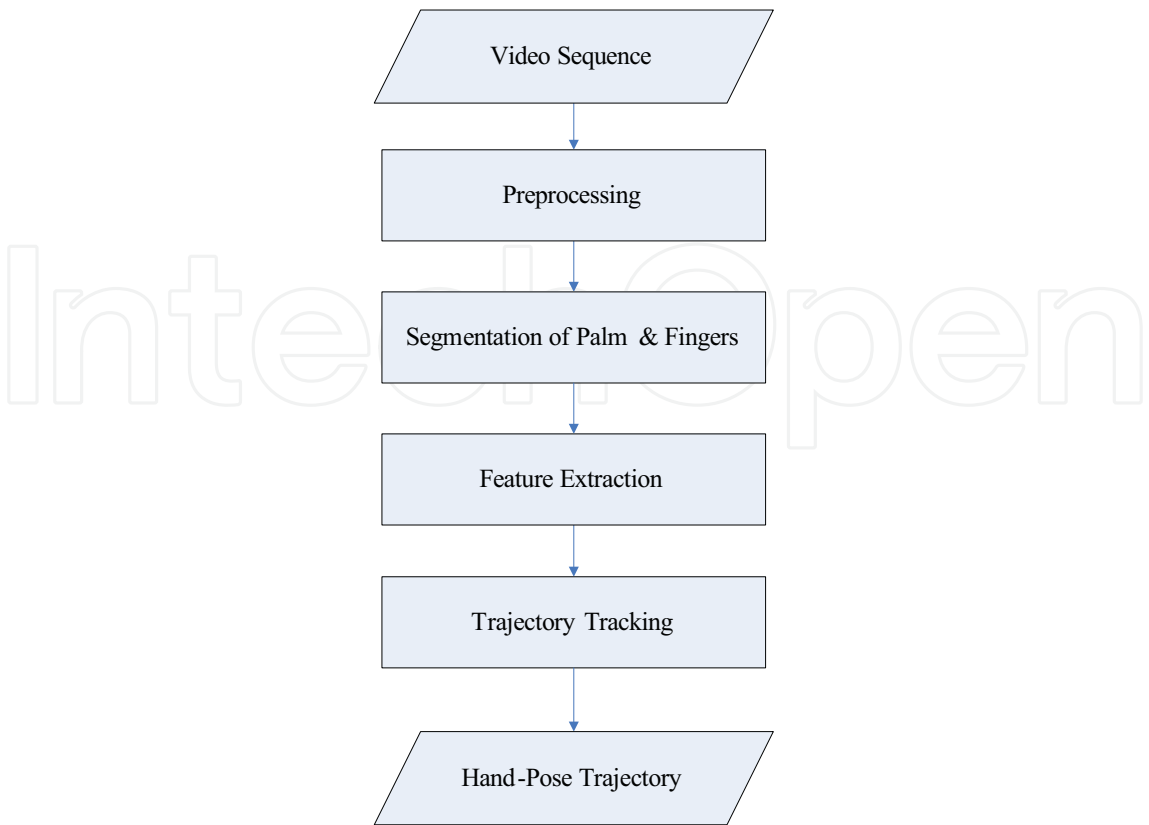
Fig. 3. A simplified flow chart of the automatic hand-pose trajectory tracking system using video sequences.

search, and arm removal. Fig. 4 shows an example of the preprocessing, where (a) is the original image, (b) is the image after gray-level transformation, (c) is the resulting image after smoothing and edge detection, (d) is the resulting human hand region after hand-pose contour search, and (e) is the identified hand-pose region.
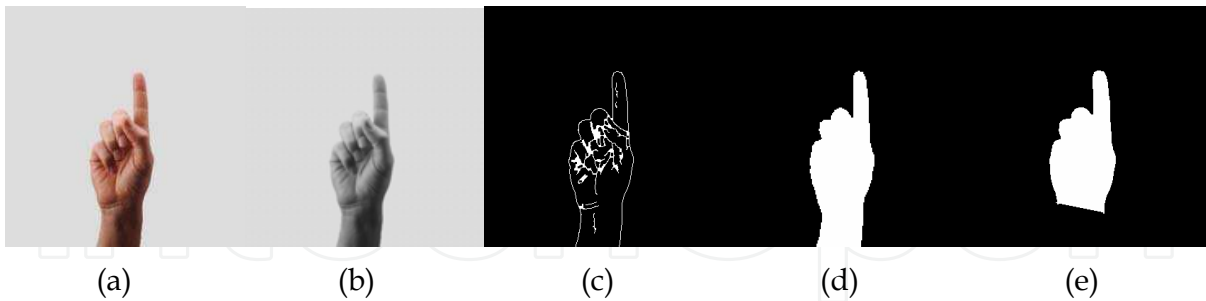

| (a) | (b) | (c) | (d) | (e) |

Fig. 4. An example of the preprocessing. (a) Original image; (b) The image after gray-level transformation; (c) The resulting image after smoothing and edge detection; (d) The resulting human hand region after hand-pose contour search; (e) The identified hand-pose region.

2.1.1 Gray-level transformation
The original color image is converted to gray-level image using the following equation:

$$Y = 0.299R + 0.587G + 0.114B \tag{1}$$

where Y is the intensity of the gray-level image; R, G, and B are the color components of the color image.

2.1.2 Smoothing
Smoothing is applied to remove image noise. The technique of "averaging with rotating masks" (Sonka 2007) is used in an attempt to enhance boundaries of the human hand in images, while removing image noise.

2.1.3 Edge detection
After smoothing, edge detection is applied by the system to detect edges or boundaries of the human hand. The Canny edge detection (Canny, 1986) is selected for the purpose. After the edge detection, the morphological processing (Gonzalez, 2008) is applied to refine the boundaries.

2.1.4 Hand-pose contour search
With the assumption that a human hand is the dominant object in the image, the hand-pose contour search is applied to extract the dominant object (main region) associated with the human hand. The process starts by filling the interior of each closed region, and then selects the region of the largest area (number of pixels) as the human hand region.

2.1.5 Arm removal
During the image acquisition, a human hand typically includes the arm that is irrelevant to the hand-pose recognition. This process is aimed to remove the arm and identify the hand-pose region for further processes. Fig. 5 shows an example of the arm removal. The system
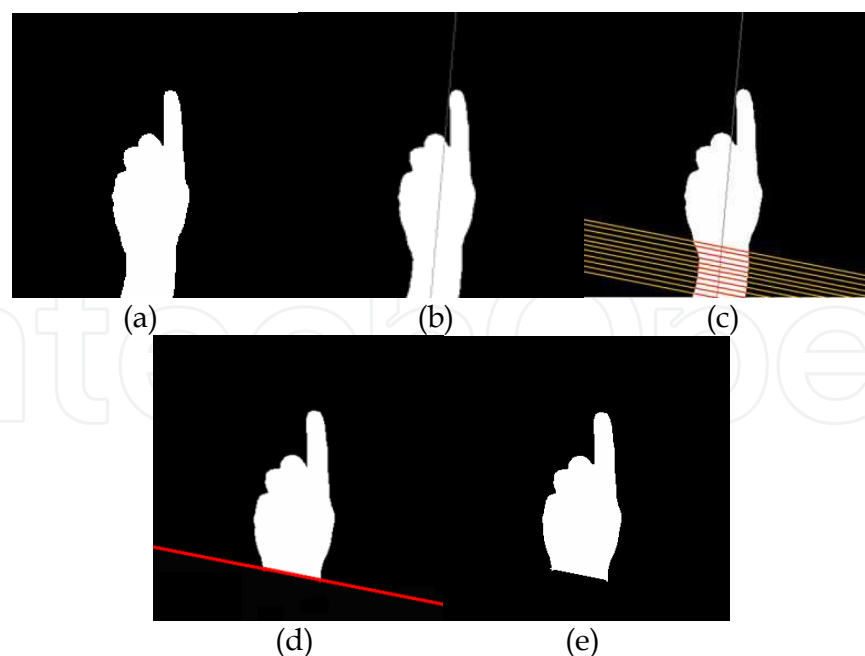

Fig. 5. An example of the arm removal. (a) The identified human hand region; (b) The center-axis for the human hand; (c) Normal lines (yellow) with respect to the center-axis and the measured arm widths (red); (d) Identified boundary for the arm and the palm; (e) The identified hand-pose region.

design for the arm removal includes the following procedures: (1) Detection of the center-axis; (2) Identification of the arm and palm boundary; and (3) Segmentation of the arm and hand-pose regions. Detail description of the technique follows.

Detection of the center-axis – Given the image with the identified human hand region (Fig. 5(a)), the objective is to detect the *center-axis* (Fig. 5(b)) that best represents the orientation of the human hand region. In practice, the least-square approximation (Cormen, 2001) is used to fit all the pixels (x_i, y_i), $i = 1…N$ in the human hand region. If the least-square line is defined as $y = r_0 + r_1 x$, then the vector r can be solved using the following equations:

$$r = \begin{bmatrix} r_0 \\ r_1 \end{bmatrix} = (A^T A)^{-1} A^T b \tag{2}$$

where

$$A = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \text{and} \ \ b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{3}$$

Identification of the arm and palm boundary – Based on the center-axis of the human hand region, the objective is to identify the arm and palm boundary. Our system is designed with the following two assumptions: (1) the boundary is perpendicular to the center-axis; and (2) the boundary can be located where the measured arm widths change dramatically. In practice, the system starts by locating the initial pixel at the image boundary and defines a set of normal lines that are perpendicular to the center-axis (Fig. 5(c)). Let (x_i, y_i) denote the pixel coordinates on the center-axis $y = r_0 + r_1 x$, the normal line equation can be defined as:

$$(y - y_i) = m(x - x_i) \tag{4}$$

where

$$m = -\frac{1}{r_1}. \tag{5}$$

Let W_j denote the measured widths of the *j*-th normal line, starting from the *n*-th normal line, we compute the variance V of the measured widths of K normal lines along the center-axis by:

$$V = \frac{1}{K} \sum_{j=n}^{n+K-1} (W_j - \overline{W})^2 \tag{6}$$

where

$$\overline{W} = \frac{1}{K} \sum_{j=n}^{n+K-1} W_j. \tag{7}$$

The arm and palm boundary can be located where the variance V exceeds a pre-defined threshold T_v.

Segmentation of the arm and hand-pose regions – After the arm and palm boundary is found, the segmentation is straightforward. If the line equation of the boundary is defined as $y = ax + b$. All foreground pixels can thus be classified as either in the hand-pose region or in the arm region by simply examining if $y \geq ax + b$ or $y < ax + b$ (Fig. 5(d)). As a result, the hand-pose region can be identified by setting all foreground pixels in the arm region as the background, resulting in the hand-pose region (Fig. 5(e)).

2.2 Segmentation of palm & fingers
In this content, the hand-pose is defined as the hand shape formed by palm and fingers only. Here, we propose a method for the segmentation of palm and fingers in hand images. The objective is to further segment the hand-pose region into palm region and finger region(s). Fig. 6 shows a simplified flow chart for the segmentation of palm and fingers in hand images. The method can be described in three processes: (1) Definition of centroid & quadrants; (2) Rule-based radius search; and (3) Region correction.

2.2.1 Definition of centroid & quadrants
Ideally, the system is aimed to identify the palm region first and segment finger regions from the palm region. In practice, we define the centroid $P(x_{mid}, y_{mid})$ that is close to the actual center of the palm by:

$$\left(x_{mid}, y_{mid}\right) = \left(\frac{1}{M}\sum x_i, \frac{1}{M}\sum y_i\right) \tag{8}$$

where $(x_i, y_i) \in R$ and R is the hand-pose region. M is the number of pixels in the hand-pose region. Fig. 7 shows an example of the identified centroid $P(x_{mid}, y_{mid})$ given the hand-pose region. Using the centroid as the origin in the polar coordinate system, the hand-pose region can be further partitioned into four quadrants I, II, III, and IV, respectively.
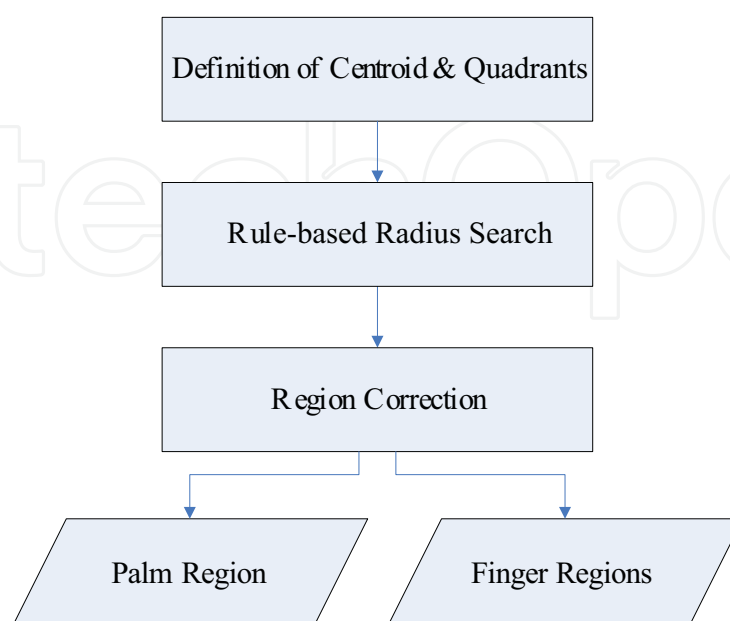
Fig. 6. A simplified flow chart for the segmentation of palm and fingers in hand images.
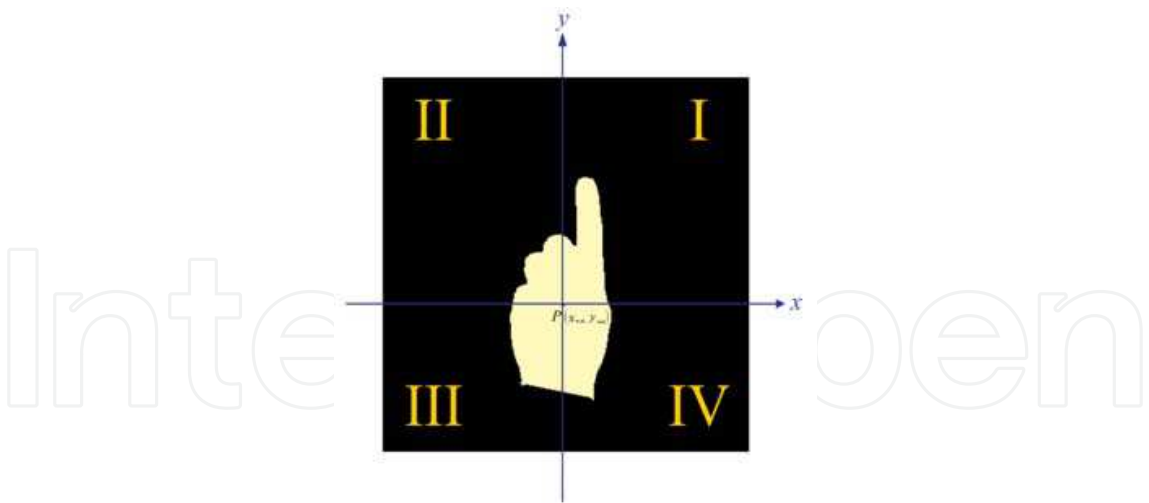
Fig. 7. Definition of the centroid P and the four quadrants for the hand-pose region.

2.2.2 Rule-based radius search

In this step, the system design is based on the assumption that the palm is located at the interior region, while the fingers are located at the exterior region of the hand-pose region. Because of the complex structure of human hands and various hand-poses, segmentation of palm and fingers is not an easy task. To overcome the problem, the system is designed to find a radius and draw a one-quarter circle for each of the four quadrants. As a result, in each quadrant, the interior of the circle is identified as the palm region, while the exterior of the circle is identified as the finger regions.

Given the polar coordinate system, the method is similar to the concept of "signatures" used in boundary description for object recognition (Gonzalez, 2008). Fig. 8 shows an example of the distance-versus-angle signatures $r(\theta)$ used to simplify the 2D hand shape in each quadrant into 1D signatures. The contour distance r is selected at the furthest point from the centroid, while the angle θ ranges from 0˚ ~ 90˚. The corresponding 1D signatures are shown in Fig. 9.

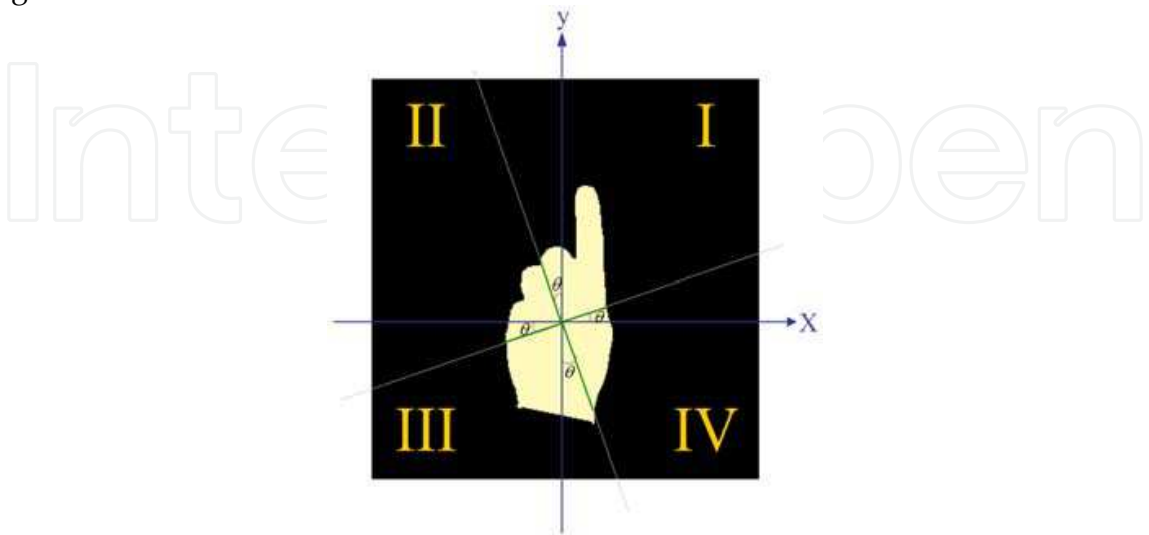Fig. 8. Distance-versus angle signatures using the polar coordinate system for the hand-pose region.
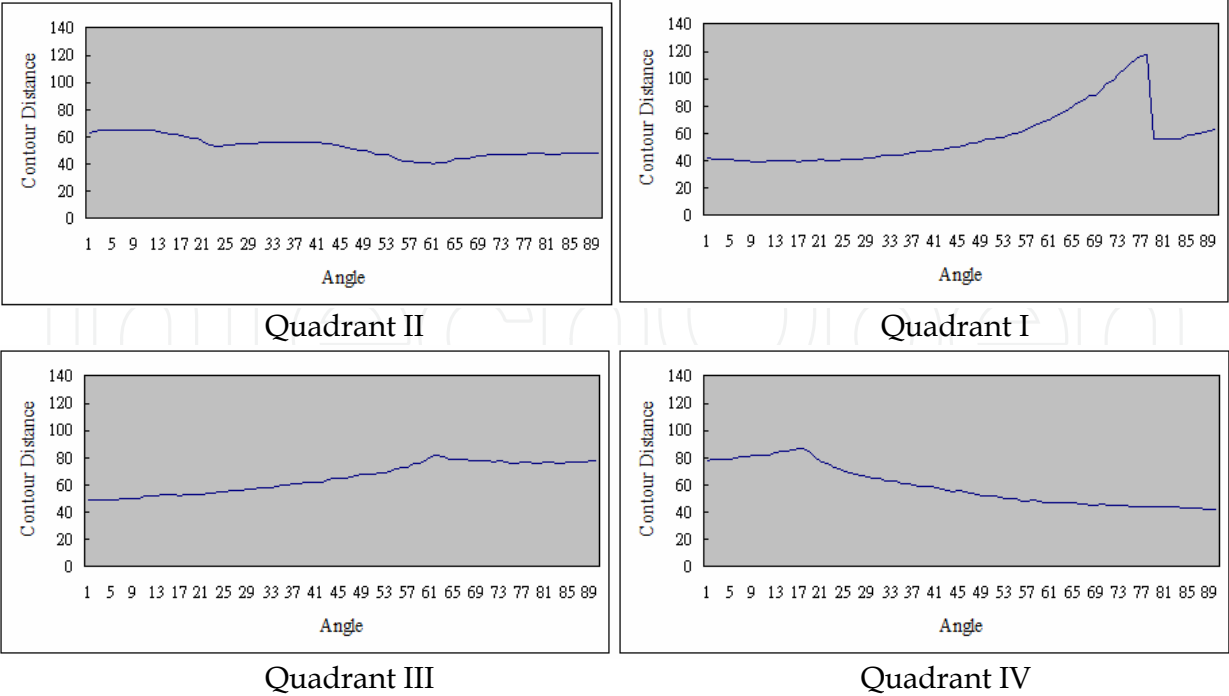
Quadrant II   Quadrant I

Quadrant III   Quadrant IV

Fig. 9. The computed 1D signatures for the hand-pose region in Fig. 8. For each of the quadrant I, II, III, or IV, the contour distance r given the angle $\theta = 0° \sim 90°$ is determined.

From the 1D signatures for the hand-pose region, the following properties can be observed:
1. The distances from boundary pixels of the fingers to the centroid are generally larger;
2. The distances from boundary pixels of the palm to the centroid are generally smaller;
3. At least one quadrant contains no fingers, or belongs to the palm region.

Based on the 1D signatures, the system then computes the maximum (MAX), the minimum (MIN), and the average (AVERAGE), respectively. In addition, the three values X, Y, and Z are computed as:

$$\begin{cases} X = MAX - AVERGE \\ Y = AVERAGE - MIN \\ Z = MAX - MIN \end{cases} \tag{9}$$

Therefore, the relationship X + Y = Z holds (Fig. 10).
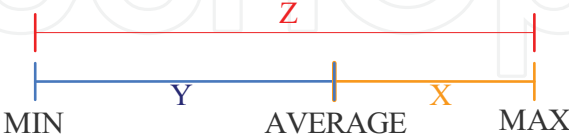

Fig. 10. The relationship of X, Y, and Z. Therefore, X + Y = Z.

Following the aforementioned three observed properties, a rule-based approach is presented for the segmentation of palm and fingers. In practice, the system finds a radius for each quadrant using the five rules given below:
1. **Rule 1 -** *The quadrant with the minimum Z.* Because the distances from the boundary pixels of palm to the centroid are generally smaller, this quadrant is most likely to contain the palm region with no fingers. The system selects MAX as the radius.

2. **Rule 2 -** *The quadrant with the maximum Z.* Because the value Z stands for the difference between MAX and MIN, this quadrant is most likely to contain the finger regions. The system selects $90 \times Y/Z$ samples with the smaller distances (more likely to be the palm boundaries) and compute the average as the radius.

3. **Rule 3 -** *The remaining quadrant with* X : Y > 2 : 1. Because the value X is larger than the value Y, this quadrant is most likely to contain a small portion of the finger regions. The system removes $90 \times Y/Z \times X/Z$ samples with larger distances (more likely to be the finger boundaries) and compute the average of the remaining sample as the radius.
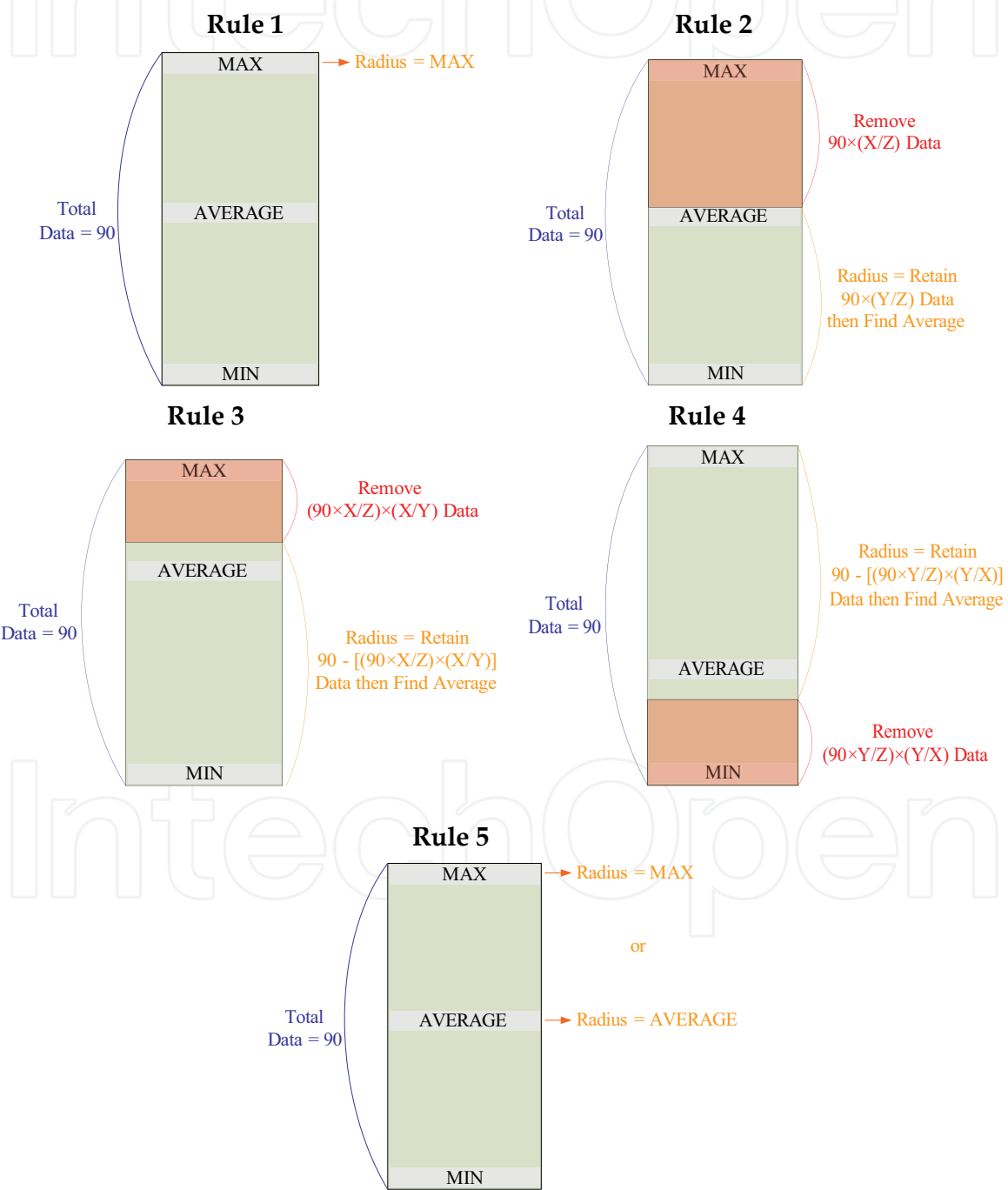

Fig. 11. Graphical demonstration of the five rules used for the rule-based radius search.

4. **Rule 4 -** *The remaining quadrant with* $X : Y > 1 : 2$. In contrary to the Rule 3, this quadrant is most likely to contain a large portion of the finger regions. The system removes $90 \times Y/Z \times X/Z$ samples with smaller distances (more likely to be the palm boundaries) and compute the average of the remaining sample as the radius.
5. **Rule 5 -** *The remaining quadrant not satisfying the Rule 1 through 4.* This quadrant is most likely to contain the palm region in hand-poses with few fingers (e.g., fist). If the quadrant is adjacent to the quadrant with minimum Z, the system selects MAX as the radius. Otherwise, the system selects AVERAGE as the radius.

A graphical demonstration of the five rules is given in Fig. 11.

Given the hand-pose region in Fig. 7, the rule-based radius search is applied for each of the four quadrants and the result is shown in Fig. 12. The interior of the one-quarter circle is classified as the palm region, while the exterior of the one-quarter circle is classified as the finger regions. As a result, the initial segmentation of palm and fingers is achieved. In this example, the quadrant I satisfies the Rule 2; the quadrant II satisfies the Rule 1; and the quadrant III and IV satisfy the Rule 5, respectively.
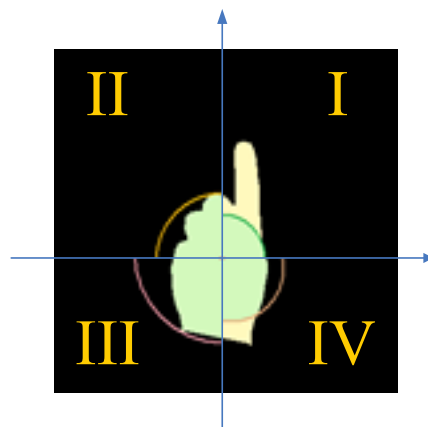

Fig. 12. An example of the rule-based radius search for the initial segmentation of palm and fingers.

2.2.3 Region correction

The objective of region correction is to verify if the initial segmentation of palm and fingers is correct. This process will retain misclassifications of the palm and finger regions if necessary. Region correction can be described in two steps: palm correction and finger correction.

Palm correction – Fig. 13 shows an example of the hand-pose image after the palm correction. In this example, the quadrant I satisfies the Rule 4, the quadrant II satisfies the Rule 2, the quadrant III satisfies the Rule 1, and the quadrant IV satisfies the Rule 5. As seen in the quadrant III, the thumb is not correctly classified as the finger region. Based on the assumption that palm region is a connected region in the hand-pose region, palm correction is applied to verify if the straight line as extended from the centroid to the boundary remains inside the hand-pose region. If so, the pixels are still classified as the palm region. Otherwise, the pixels are classified as the finger region instead.

Finger correction – The objective is to verify if each of the remaining regions is correctly identified as the finger region. Fig. 14 shows an example of the finger correction, where the
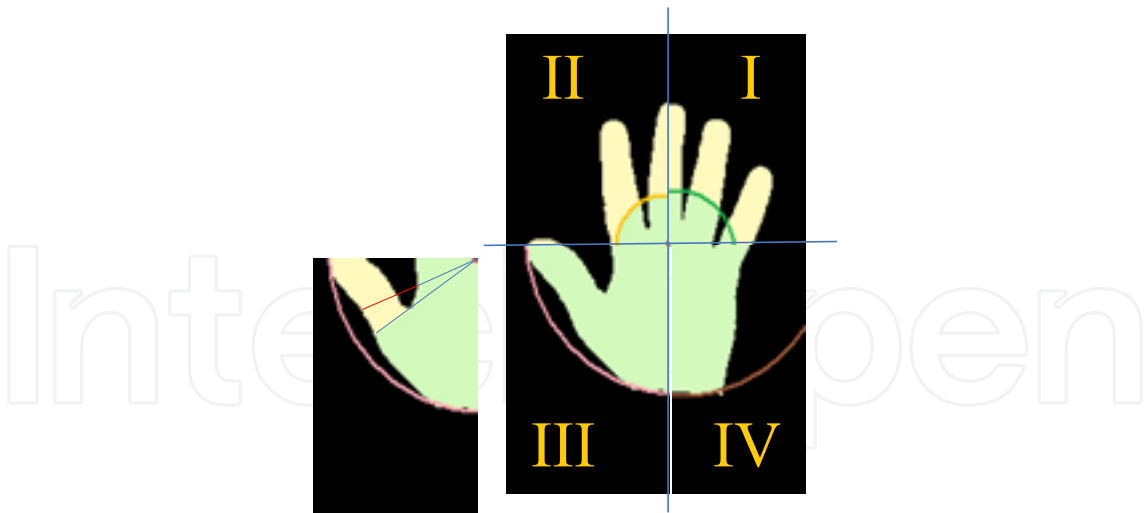
Fig. 13. An example of the palm correction. In the quadrant III, the thumb is misclassified as the palm region. After the palm correction, the thumb is re-classified as the finger region.

region ① is correctly classified as the finger region and the region ② is initially misclassified. Here, the system determines the center-axis using the least-square approximation for each of the remaining region. Then, the lengths either inside the palm region (green) or outside the palm region (yellow) are compared. If the green line is longer than the yellow line, the region remains. Otherwise, the region is re-classified as the palm region.
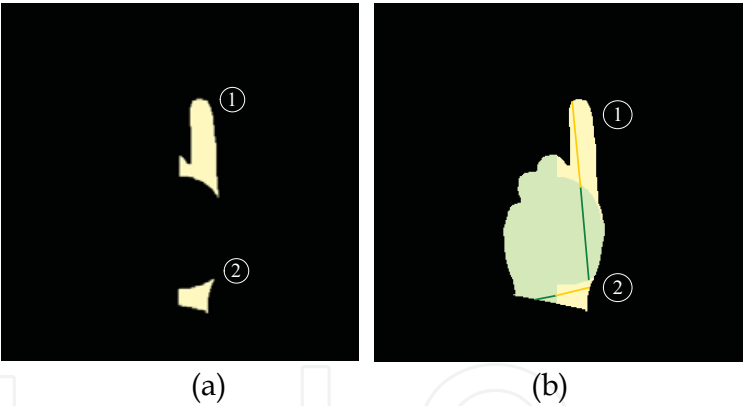

(a)   (b)

Fig. 14. An example of the finger correction. (a) The region ① is correctly classified and the region ② is initially misclassified; (b) The system determines the center-axis of each remaining region and the lengths either inside the palm region (green) or outside the palm region (yellow). After the finger correction, the region ② is re-classified as the palm region.

In summary, Fig. 15 shows an example of the segmentation of palm and fingers. In this example, the original hand-pose region (Fig. 15(a)) is segmented into two regions: the palm region (Fig. 15(b)) and the finger region (Fig. 15(c)), respectively.

2.3 Feature extraction
The objective of the feature extraction is to quantitatively determine several features using the hand-pose region. In our system, the following features are computed:
1. *Number of fingers*: The number of identified finger regions is used as the number of fingers;
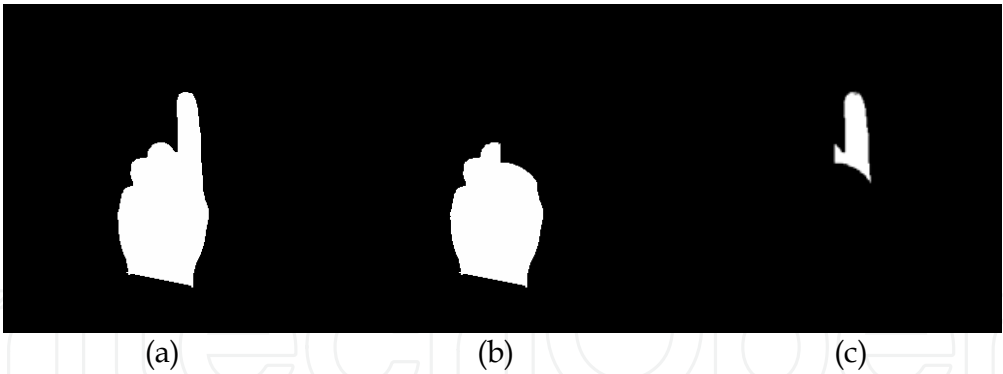
Fig. 15. An example of the segmentation of palm fingers, where (a) is the original hand-pose region, (b) is the segmented palm region, and (c) is the segmented finger region.

2. *Fingertip's coordinate*: Based on the center-axis of each finger region (following the finger correction), the (x, y) coordinate on the center-axis that is farthest from the centroid is recorded as the fingertip's coordinate;
3. *Fingertip-centroid distance*: The fingertip-centroid distance is calculated as the Euclidean distance from the fingertip to the centroid;
4. *Angle of two fingertips*: If multiple fingers are identified, the angle of two fingertips is also determined. Here, only the maximum angle is computed as the hand-pose feature.

Fig. 16 shows an example of the feature extraction. In this example, the number of finger is 1, the fingertip's coordinate is recorded as (91, 195), the fingertip-centroid distance is 118 (pixels), and the angle of two fingertips is 0° (only 1 finger identified).
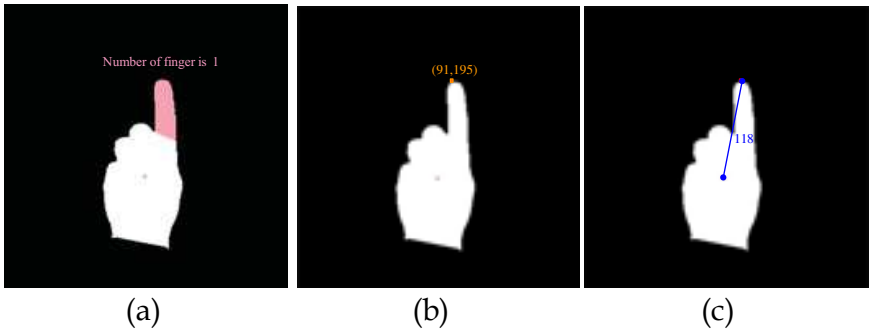

Fig. 16. An example of the feature extraction. (a) Number of fingers; (b) Fingertip's coordinate; (c) Finger-centroid distance.

2.4 Trajectory tracking

In this step, the system objective is to determine the hand-pose trajectory automatically. Our system is designed to trace changes of the fingertip's position with the assumption that the hand-pose remains invariant. Fig. 17 shows the flow chart of the system processes for the hand-pose trajectory. Here, the frame t_{old} is the old frame (or the anchor frame), while the frame t_{new} is the new frame (or the target frame). The system processes can be described as follows.

2.4.1 Hand-pose size-error

The first indication if the hand-pose remains invariant is that the hand-pose size remains approximately the same. Here, the system determines the hand-pose size (number of pixels in the hand-pose region) for both frames and calculates the hand-pose size-error ε by:
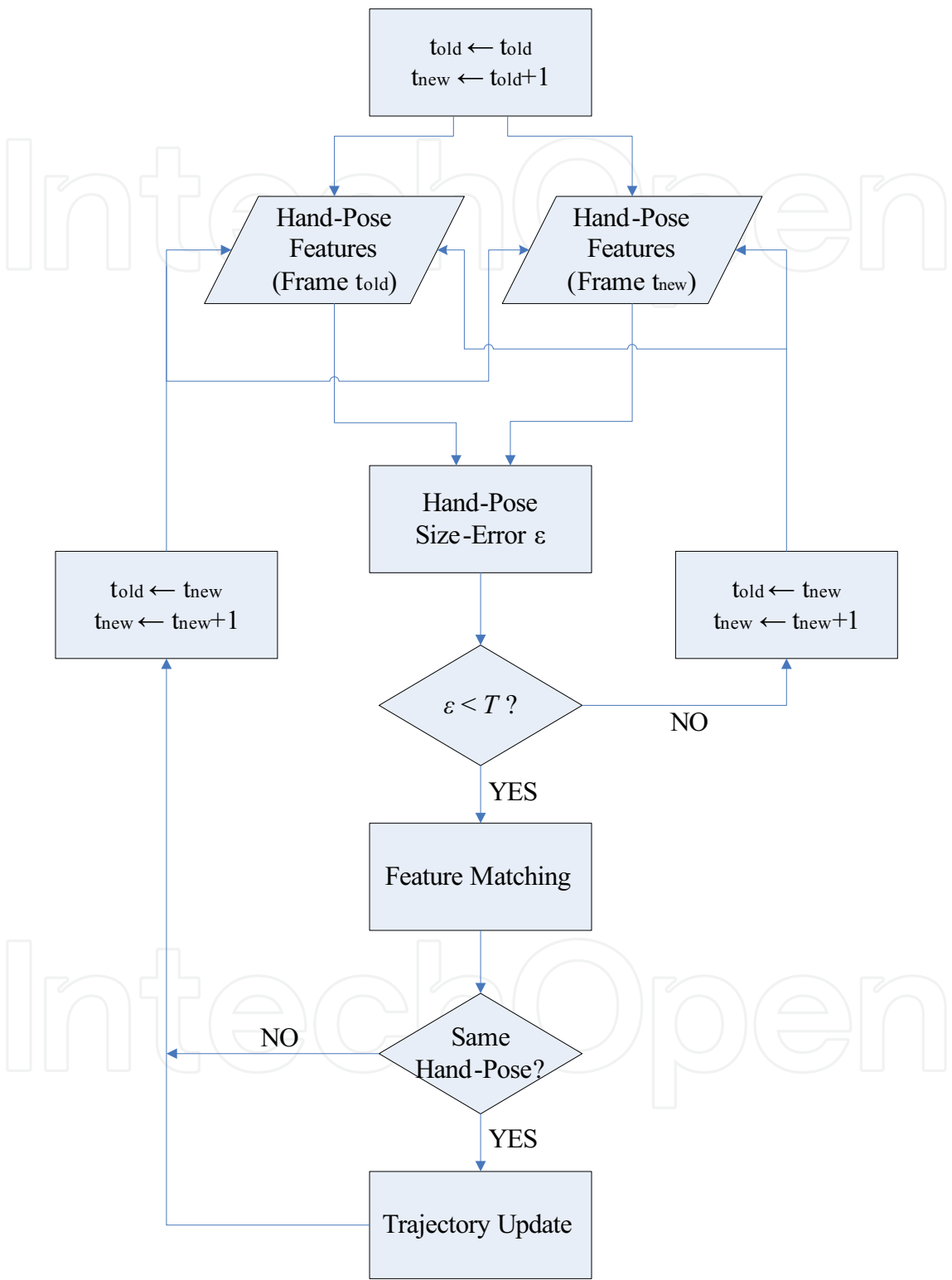
Fig. 17. A flow chart of the system processes for hand-pose trajectory. The main processes include: (1) Hand-pose size-error; (2) Feature matching; and (3) Trajectory update.

$$\varepsilon = \frac{|S_{old} - S_{new}|}{S_{old}} \times 100\% \qquad (10)$$

where S_{old} and S_{new} is the hand-pose size in the old frame and the new frame, respectively. If the hand-pose size-error ε is less than a pre-defined threshold T ($T = 75\%$ was empirically selected), the hand-pose is presumed to remain invariant. Further feature matching is applied. Otherwise, the system assumes that the hand-pose has changed and stops tracing the hand-pose trajectory temporarily.

2.4.2 Feature matching

The second indication if the hand-pose remains invariant is that the features (i.e., number of fingers, fingertip-centroid distance, and angle of two fingertips) remain approximately the same. The system assumes that the hand-pose has changed between the two frames if any of the following conditions occur: (1) the number of fingers has changed; (2) a $\pm 10\%$ change in the fingertip-centroid distances; or (3) a $\pm 3°$ change in the angle of two fingertips.

2.4.3 Trajectory update

If the hand-pose remains invariant between two frames, the system updates the hand-pose trajectory by recording the fingertip's coordinate as determined in the new frame. Otherwise, the system stops tracing the hand-pose trajectory.

3. Results

In this section, system results of the automatic hand-pose trajectory tracking are demonstrated. In our experimental design, the video camera NICON D90 was used to capture the video, and the Microsoft AVI file format was used for storage. All the experiments were carried out using the personal computer Intel Core Duo T5500 1.66G, RAM 2G. Software development included the Microsoft Visual Studio 2005 with the OpenCV 1.1 pre as the auxiliary software. In addition, all the hand-pose video data were acquired to meet the system hypotheses as described in Section 2.

Fig. 18 shows the results of the hand-pose trajectory of single fingertip (index finger) that forms a "D". Fig. 19 shows the results of the hand-pose trajectories of multiple fingertips (all five fingers) that form a "Scratch". Fig. 20 shows the results of the hand-pose trajectories in which the hand-pose has changed from single finger to double fingers.
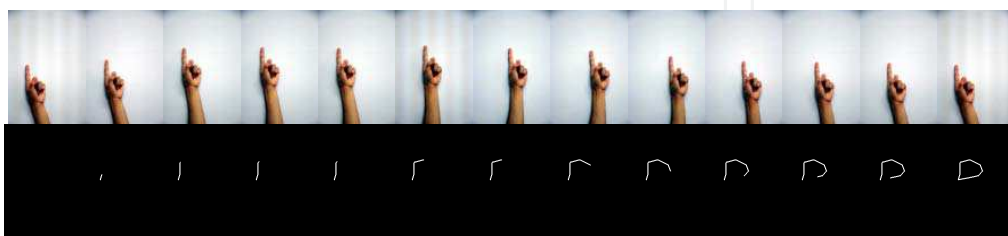

Fig. 18. Results of the hand-pose trajectory of single fingertip (index finger) that forms a "D".
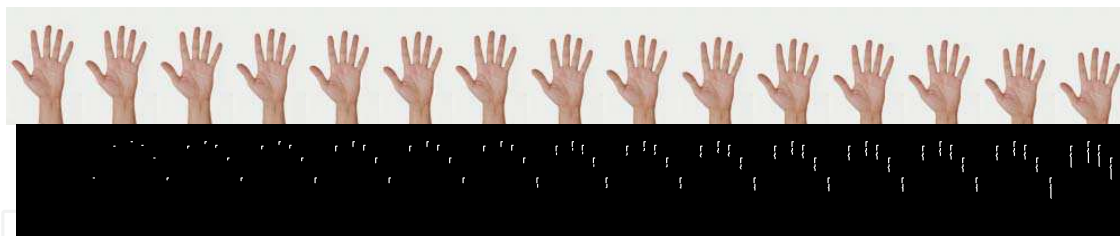
Fig. 19. Results of the hand-pose trajectories of multiple fingertips (all five fingers) that form a "Scratch".
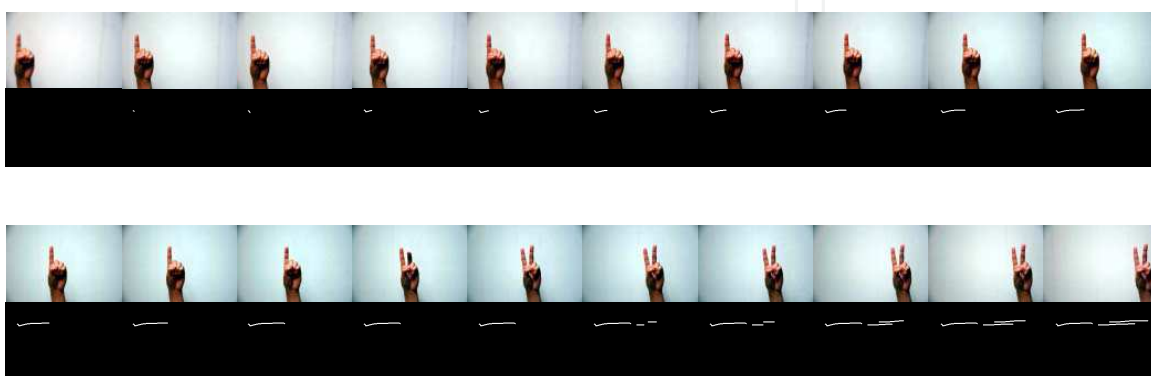

Fig. 20. Results of the hand-pose trajectories in which the hand-pose has changed from single finger to double fingers. Initially, only single hand-pose trajectory was traced. During the changes of the hand-poses, the system stopped tracing the hand-pose trajectory temporarily. Once the hand-pose remains invariant (double fingers), the system retained the tracing and double hand-pose trajectories were traced instead.

4. Conclusion

In this content, an automatic hand-pose trajectory tracking system using video sequences is presented. The results demonstrated that our system could reasonably trace the hand-pose trajectory with the assumption that the hand-pose remains invariant during motion. The techniques were based on the rule-based approach in segmenting palm and finger regions. In addition, feature extraction and matching were applied to trace the fingertip's position and determine if the hand-pose remains invariant. Even in a situation when the hand-pose changed, our system was demonstrated to be able to re-trace the hand-pose trajectories in the video sequences.

In essence, the objective of our system was very different from other hand-pose recognition systems with the primary goal to recognize various hand-poses. Instead, our system was designed to trace the hand-pose trajectory. Therefore, only a limited set of hand-poses were tested. Ultimately, our system could be integrated with the hand-pose recognition system if the tracing of hand-pose trajectory is limited in pre-defined hand-poses only (e.g., specific user-controlled commands) to enhance the functionality of a user-interface.

At present, our system was designed in an attempt to process in a frame-by-frame basis which is still time-consuming. For real-time applications, the system performance must be

further improved in terms of effectiveness and efficiency. A hardware implementation of the techniques could offer a potential solution to the problem. However, our system shows encouraging results that clearly define the future potentials in developing a convenient user-interface.

5. Acknowledgment

6. References

Argyros, A. A. & Lourakis, M. I. A. (2006). Vision-based Interpretation of Hand Gestures for Remote Control of a Computer Mouse, In *Computer Vision in Human-computer Interaction: Eccv 2006 Workshop on HCI*, pp.40-51, 978-3-540-34202-1, Austria, May 2006, Springer-Verlag, Graz

Athitsos, V. & Sclaroff, S. (2002). An appearance-based framework for 3D hand shape classification and camera viewpoint estim*ation, Proceedings of Fifth IEEE international Conference on Automatic Face and Gesture Recognition*, 0-7695-1602-5, USA, May 2002, Institute of Electrical and Electronics Engineers, Washington D.C.

Canny, J. F. (1986). A computational approach to edge detection, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, Nov. 1986, pp. 679-698, 0162-8828

Chen, F.; Fu, C. & Huang, C. (2003). Hand gesture recognition using a real-time tracking method and Hidden Markov Models, *Image and Video Computing*, vol. 21, no. 8, August 2003, pp. 745-758, 0262-8856

Chen, J. Y. & Chang, Y. H. (2007). A hand-Pose recognition system using a combined classifier of shift distances and Fourier features, *Computer Vision, Graphics and Image Processing Conference*, August 2007

Cooke, M. P.; Green, P. D.; Josifovski, L. & Vizinho, A. (2001). Robust automatic speech recognition with missing and unreliable acoustic data, *Speech Communication*, vol. 34, pp. 267-285, 0167-6393, June 2001, Elsevier

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; & Stein, C. (2001). *Introduction to Algorithms, Second Edition*, MIT Press, 0-262-03293-7, Cambridge, Massachusetts

Fang, G.; Gao, W. & Zhao, D. (2004). Large vocabulary sign language recognition based on fuzzy decision trees, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 34, no.3, May 2004, pp. 305-314, 1083-4427

Froba, B. & Ernst, A. (2004). Face detection with the modified census transform, *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 91-96, 0-7695-2122-3, Korea, May 2004, Institute of Electrical and Electronics Engineers, Seoul

Gales, M. J. F. (1998). Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition, *Computer Speech and Language*, January 1998, pp. 75-98, 0885-2308

Gao, W.; Fang, G.; Zhao, D. & Chen, Y. (2004). A Chinese sign language recognition system based on SOFM/SRN/HMM, *Pattern Recognition*, vol. 37, pp. 2389-2402, 0031-3203

Gonzalez, R. C. & Woods, R. E. (2008). *Digital Image Processing, 3rd Edition*, Prentice Hall, 9780131687288, New York

Graetzel, C.; Fong, T.; Grange, S. & Baur, C. (2004). A Non-Contact Mouse for Surgeon-Computer Interaction, *Technology and Health Care*, vol. 12, no. 3, pp. 245-257, 0928-7329

Heumer, G.; Amor, H. B.; Weber, M. & Jung, B. (2007). Grasp Recognition with Uncalibrated Data Gloves - A Comparison of Classification Methods, *Virtual Reality Conference, 2007. VR '07. IEEE*, March 2007, pp. 19-26, 1-4244-0905-5

Just, A.; Rodriguez, Y. & Marcel, S. (2006). Hand posture classification and recognition using the modified census transform, In *7th International Conference on Automatic Face and Gesture Recognition*, pp. 351-356, 0-7695-2503-2, UK, April 2006, Institute of Electrical and Electronics Engineers, Southampton

Liu, C.-L.; Nakashima, K.; Sako, H. & Fujisawa, H. (2003). Handwritten digit recognition: benchmarking of state-of-the-art techniques, *Pattern Recognition*, vol. 36, pp. 2271-2285, 0031-3203

Malima, A.; Ozgur, E. & Cetin, M. (2006). A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control, *Signal Processing and Communications Applications, 2006 IEEE 14th*, pp. 1-4, 1-4244-0238-7, Turkey, April 2006, Institute of Electrical and Electronics Engineers, Antalya

Palacios, R. & Gupta, A. (2002). A system for processing handwritten bank checks automatically, *Image and Vision Computing*, vol.26, no. 10, Feb. 2002, pp. 1297-1313, 0262-8856

Shan, C.; Wei, Y.; Tan, T. & Ojardias, F. (2004). Real Time Hand Tracking by Combining Particle Filtering and Mean Shift, In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 669-674, 0-7695-2122-3, Korea, May 2004, Institute of Electrical and Electronics Engineers, Seoul

Sonka, M.; Hlavac, V. & Boyle, R. (2007). *Image processing, analysis and machine vision, 3rd Edition*, Cengage-Engineering, 049508252X, Florence, KY

Stenger, B.; Thayananthan, A.; Torr, P. H. & Cipolla, R. (2006). Model-based hand tracking using a hierarchical Bayesian filter, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, Sept. 2006, pp.1372-1385, 0162-8828

Su, M. C.; Hung, C. H. & Zhao, Y. X. (2006). A SOMART System for gesture Recognition, *WSEAS Transactions on Computers*, vol. 5, July 2006, pp. 2764-2771, 1109-2750

Trivedi, M. M.; Gandhi, T. & McCall, J. (2007). Looking-In and Looking-Out of a Vehicle:Computer-Vision-Based Enhanced Vehicle Safety, *Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, March 2007, pp. 108–120, 1524-9050

Wachs, J. P.; Stern, H. & Edan, Y. (2005). Cluster Labeling and Parameter Estimation for the Automated Setup of a Hand-Gesture Recognition System, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: System and Humans*, vol. 35, no. 6, November 2005, pp. 932-944, 0018-9472

Zhai, S. & Kristensson, P. (2003). Shorthand writing on stylus keyboard, In *CHI '03 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 97-104, 1-58113-630-7, USA, April 2003, ACM Press., Ft. Lauderdale, Florida

Zhang, D.; Kong, W. K.; You, J. & Wong, M. (2003). Online Palmprint Identification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, Sept. 2003, pp. 1041-1050, 0162-8828

User Interfaces

Edited by Rita Matrai

ISBN 978-953-307-084-1

Hard cover, 270 pages

Publisher InTech

Published online 01, May, 2010

Published in print edition May, 2010

Designing user interfaces nowadays is indispensably important. A well-designed user interface promotes users to complete their everyday tasks in a great extent, particularly users with special needs. Numerous guidelines have already been developed for designing user interfaces but because of the technical development, new challenges appear continuously, various ways of information seeking, publication and transmit evolve. Computers and mobile devices have roles in all walks of life such as in a simple search of the web, or using professional applications or in distance communication between hearing impaired people. It is important that users can apply the interface easily and the technical parts do not distract their attention from their work. Proper design of user interface can prevent users from several inconveniences, for which this book is a great help.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yuan-Hsiang Chang and Chen-Ming Chang (2010). Automatic Hand-Pose Trajectory Tracking System Using Video Sequences, User Interfaces, Rita Matrai (Ed.), ISBN: 978-953-307-084-1, InTech, Available from: http://www.intechopen.com/books/user-interfaces/automatic-hand-pose-trajectory-tracking-system-using-video-sequences

INTECH
open science | open minds