Automatic Hand Gesture Segmentation Based on Multi-Feature Criteria

Linda Murphy  
Engineering and Computer Science  
Oakland University  
Rochester, MI, 48309-4401, USA  
Linda.C.Murphy@comcast.net

Ka C Cheok  
Engineering and Computer Science  
Oakland University  
Rochester, MI, 48309-4401, USA  
Cheok@Oakland.edu

Abstract

Detecting meaningful hand gestures in real-time with multi-variant room conditions presents many challenges. Variations such as room lighting, the detection for the presence of skin color, and determining the meaning of the hand gesture, if one exists, must be resolved for an automated hand gesture detection system. Solving these problems would contribute toward a non-verbal communication system that could benefit a variety of people and organizations who utilize hand gestures for communication. For the initial approach for this automated system, a multi-feature, three criteria human computer interface consisting of an object moving into a predefined three dimensional space, the presence of skin color, and non-motion is presented. A systematic method was developed to distinguish paused motion from hand movements from RGB and depth images so that pattern recognition techniques can be effectively utilized to interpret the hand gesture. The approach was successfully validated by experiments.

Keywords: HCI; hand gesture recognition; Lucas-Kanade; paused motion; pattern recognition; Kinect

1 Introduction

Real-time communication between human beings and computers presents many challenges. This paper outlines a touchless multi-feature automatic Human computer interface (HCI) that combines a computer along with the Microsoft Kinect sensor to interact with a human being using hand gestures in real time. Matlab software is used for processing. The user presents a hand gesture to the Kinect sensor. The hand gestures presented are static, meaning that the placement of the fingers and palm will be classified as a valid hand gesture by analyzing a set of multi-feature criteria once the hand exhibits no motion.

Unlike other gesture recognition algorithms which use sweeping motions in directions to communicate the meaning of hand gestures, the concept with this algorithm is that hand gesture communication starts when the motion stops.

Section 2 summarizes the multi-feature criteria structure for qualifying and interpreting hand motion. Section 3 describes the experimental setup for camera calibration. Section 4 outlines the hand gesture structure including skin color detection and automatic noise removal. Section 5 presents the topic of motion tracking and the importance of non-motion. Section 6 outlines the process of converting the hand gesture to a signal. Section 7 discusses the pattern recognition output and Section 8 presents the conclusions.

2 Multi-Feature Criteria

Figure 1 represents the multi-feature outline of the hand gesture acquisition for qualifying a hand gesture for pattern recognition. There are three main criteria that must be present for a possible hand gesture to be analyzed [1].

- An object must be present within the Kinect active region.
- An object must contain skin color.
- An object must have paused motion for a period of time.

Only objects that contain skin color in a predefined segment of three dimensional space viewed by the Kinect will be considered. This three dimensional space is called the Kinect active region (KAR). This predefined definition of an observed object that contains skin color in this region of space triggers the start of communication between a person and the computer. The Kinect is very noisy due to background clutter, lighting conditions, and degrees of accuracy and precision [2]. Defining a smaller active region reduces noise and activates the HCI.

Since this HCI system is automated, the trigger to start communication is non-haptic. The KAR is defined by a Kinect space consisting of x (width), y (height), and z (depth). If an object is observed to exist within this predefined region of space and it contains the second criterion which is the presence of skin color, the object will be tracked to further determine if communication is taking place. If the object travels outside the width, length, or depth requirements of the KAR, the HCI will stop observing the object.
This HCI system will know when to begin to analyze a possible hand gesture when motion of the hand stops for a predetermined amount of time after the first two criterion have been satisfied. If a hand gesture is held long enough and exhibits non-motion, this meets the third criterion which is the hand gesture hold time. A snapshot of this gesture is further analyzed by fuzzy logic and pattern recognition techniques to determine the sequence of hand gestures presented to the system.

Figure 1: Multi-feature criteria for qualifying a hand gesture

3 Calibration Experimental Setup

Since the RGB and depth cameras of the Kinect are spaced approximately one inch apart, camera calibration must be performed to combine both RGB and depth images. A box was positioned in front of the Kinect and measurements are taken from both RGB and depth images. All measurements are recorded in millimeters.

Figure 2 presents the experimental setup to determine the calibration parameters for aligning both RGB and depth images from the Kinect. RGB measurements are collected from the (u, v) pixel locations from 5 different locations positioned on the box. Depth measurements are recorded from the same corresponding RGB image locations.

Equation (1) refers to the matrix A as the camera intrinsic matrix and \( o_x \) and \( o_y \) are the coordinates of the principal point of the image. The focal lengths are conveyed as \( f_x \) and \( f_y \) and they are expressed in pixel units [3].

\[
A = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}
\] (1)

All the angle calculations are computed from the recorded box measurements from both the RGB and depth images using the set of equations represented by Equations (2) through (8). These calculations are performed for both the vertical and horizontal directions. The focal lengths \( f_x \) and \( f_y \) for each of the RGB and depth images are provided by Equations (9) and (10).

\[
aA = \cos^{-1} \left( \frac{B^2 + C^2 - A^2}{2BC} \right)
\] (2)
\[ aB = \frac{aA}{1 + (uC - uA)/ (uA - uB)} \]  
\[ aC = aA - aB \]  
\[ xB = -B \sin(aB) \]  
\[ zB = B \cos(aB) \]  
\[ xC = C \sin(aC) \]  
\[ zC = C \cos(aC) \]  
\[ f_x = \begin{bmatrix} \frac{xB}{uB - uA} \\ \frac{xC}{uC - uA} \end{bmatrix} \]  
\[ f_y = \begin{bmatrix} \frac{yB}{vB - vA} \\ \frac{yC}{vC - vA} \end{bmatrix} \]  
\[ \text{The combined calibrated depth and RGB images shown from Figure 4a and 4b are presented in Figure 4c. The next criterion for hand gesture segmentation is to determine if there is skin color present within the image. If skin color is found, this object is segmented from the image where noise is removed and the hand gesture is fortified by filling any skin color gaps from within the hand gesture.} \]

4 Hand Gesture Structure

One of the challenges with automatic object detection within an image is how to discern what data is valid and what is not. The computer does not know which pixels are noisy and which pixels contain information that is important for processing.

4.1 Skin Color Detection

The approach for detecting skin color within images was to combine two color spaces, normalized RGB represented by the term nRGB, and YCbCr. RGB images contain luminance within all three color channels where YCbCr separates the luminance represented by the Y channel [4, 5].

Since RGB images contain brightness within all three color channels, the RGB channels of the combined image are normalized along all three color channels in preparation for skin color detection [6, 7]. By normalizing these color channels, the effects of brightness within the images are reduced. Equation (11) represents each color channel for red, green, and blue for an RGB image. Equations (12), (13), and (14) normalize the color space for red, green and blue respectively. Equation (14) further attenuates the illumination within the image by multiplying the normalized RGB values. The conversion of RGB to YCbCr [8] is represented by Equation (16).

\[ r = \frac{R}{R + G + B}, \ g = \frac{G}{R + G + B}, \ b = \frac{B}{R + G + B} \]  
\[ R_n = \frac{r}{\sqrt{r^2 + g^2 + b^2}} \]  
\[ G_n = \frac{g}{\sqrt{r^2 + g^2 + b^2}} \]  
\[ B_n = \frac{b}{\sqrt{r^2 + g^2 + b^2}} \]  
\[ nRGB = nRGB \times nRGB \quad R_n, G_n, B_n \in [0, 1] \]  
\[ \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \end{bmatrix} \begin{bmatrix} R \\ G \end{bmatrix} \]  

Equations (17) through (20) represent the skin color detection formulas for nRGB and YCbCr color spaces [7, 8]. By using a fusion of both color detection results, more skin color information is retained from within the image.

\[ R_n > 0.373 \land G_n > 0.157 \land B_n > 0.078 \land R_n > G_n \land R_n > B_n \]  
\[ \| (R, G)_n \| > 0.059 \land R > G \land R > B \]  
\[ Y > 0.314 \]  
\[ 0.333 \leq Cb \leq 0.529 \]  
\[ 0.529 \leq Cr \leq 0.705 \]  

Figure 5a, 5b, and 5c represent the normalized RGB image, RGB to YCbCr image, and the combined binary normalized RGB and YCbCr images respectively. The
binary image of Figure 5c represents the results of the detection of skin color pixels along with false positive detections within the image.

![Figure 5: a. Normalized RGB image, b. RGB to YCbCr image, and c. Combined binary nRGB and YCbCr images.](image)

### 4.2 Automatic Noise Removal

Automatic noise removal from binary images can be a difficult process. The computer algorithms must be able to discern valid skin color as opposed to noisy pixels. If morphological operations are used, too much valuable data can be lost. Figure 6a and 6b display the same binary image along with example problems that algorithms face when trying to automatically determine which pixels are valid and how information must connect together to form a meaningful hand gesture.

Figure 6a shows missing skin color pixels in the lower arm region and a noisy boundary layer of pixels around the hand. Figure 6b presents the aftermath of a loss of skin color data after morphological operations are used. The little finger of the hand actually detaches and forms an island of pixels in the image.

![Figure 6: a. Noisy missing data and b. Distorted data.](image)

To retain skin color information, two techniques are employed. The first technique is use of a boundary layer that contains the majority of the skin color found within the image. This boundary layer is formed by creating a mask of the information retained from the depth image and overlaid onto the RGB image. The second technique, a skeleton layer connecting the sections of the image together, is then added to the image. Taking the image from Figure 5c, a boundary and skeleton layer are added to the image to form a new binary image shown in Figure 7a and 7b.

![Figure 7: a. Skeleton structure added to the image and b. Boundary layer added to the skeleton and skin color image.](image)

By determining the density of the skin color pixels within the rows and columns of the images within the boundary layer and by following the skeleton as a roadmap, the hand gesture image can be cleaned of noisy pixels. Any remaining holes in the image representing the hand gesture can be filled based on the density of the skin color pixels in the area of the image. Figure 8 shows the result of these cleanup operations.

![Figure 8: Skin color output.](image)

### 5 Motion Tracking – Paused Motion

Once the hand has been successfully segmented from the image after skin color is detected, motion tracking can begin. The motion tracking method is the Lucas-Kanade algorithm. [9] A template image T(x) is compared with an image P(x) where x is a vector of pixel coordinates from an image as represented by Equation (21). The algorithm is described in [8, pp. 453-454], consisting of Equations (21) through (25).

In order to determine motion versus non-motion (pause), the parameter updates of d must be minimized. The variable d is the offset between the pattern and target images and is represented by Equation (21). This is the variable that characterizes the amount of motion between the pattern and target images in the horizontal and vertical directions.

By setting \( k = 0 \) and initializing \( d_0 = 0 \) Equations (21) through (25) are computed by incrementing \( k \) by 1 until \( d_{k+1} \) converges. From the value \( d \), the threshold can be evaluated to determine if there is motion or non-motion that satisfies the criteria for motion detection between two successive images.
\begin{align}
  \mathbf{x} &= \begin{bmatrix} x \\ y \end{bmatrix} \\
  \nabla \mathbf{P} &= \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial x} & \frac{\partial \mathbf{P}}{\partial y} \end{bmatrix} \\
  \mathbf{H} &= \left( \sum_x (\nabla \mathbf{P})^T (\nabla \mathbf{P}) \right)^{-1} \\
  \Delta \mathbf{d} &= \mathbf{H}^{-1} \left( \sum_x (\nabla \mathbf{P})^T (\mathbf{T}(\mathbf{x}) - \mathbf{P}(\mathbf{x} + \mathbf{d})) \right) \\
  \mathbf{d}_{k+1} &= \mathbf{d}_k + \Delta \mathbf{d}
\end{align}

(21) (22) (23) (24) (25)

This value is arbitrarily set to a predefined threshold value. In this example, \( d \) is set to less than 3. When \( d \) is greater than 3, motion is detected as the template \( \mathbf{T}(\mathbf{x}) \) is compared to image \( \mathbf{P}(\mathbf{x}) \). The motion indicator screen is shown in the fourth subplot window in Figure 9 represented by the variable \( \text{THold} \). The motion indicator screen will show green, indicating that motion is detected, or red when motion is not detected based on the threshold of \( d \). In Figure 9 the motion indicator is green. This indicates that motion is detected. Figure 10 represents a stable hand gesture with no motion detected.

6 Hand Gesture to Signal

To determine the number of fingers within the image, the hand gesture is transformed into a signal which represents the hand gesture. To reduce the level of noise in the resultant signal, the image is scanned to determine where the fingertip locations are by locating where the minimum column position is that contains the least amount of skin pixels as compared to the maximum edge of the hand gesture which represents skin color. In this way, we remove any part of the signal that does not represent pertinent information needed to determine the hand gesture.

Figure 11a represents the boundary layer of the segmented hand gesture with the focus cut taken at the base of the hand based on fingertip location within the image. The information contained at the base of the hand does not contain meaningful information to determine the number of fingers present in the image so the new focus cut is taken and shown in Figure 11b.

The radial distances are measured based from the center location of the focus cut shown in Figure 11b. A hand gesture signal representation is shown in Figure 12a. The signal still contains noise so a low pass filter from Equation (26) smooths out the signal further. Figure 12b shows the filtered result.

\[ H(z) = \frac{0.4}{z - 0.6} \]

(26)

Figure 11: a. Hand gesture boundary and b. Focus cut to eliminate noise.

Figure 12: a. Noisy hand gesture signal and b. Filtered hand gesture signal.
7 Multi-Feature Pattern Recognition

The process of determining what pattern is contained within the hand gesture image is to take the first and second derivatives of the hand gesture signal. These first and second derivatives will be the basis for memberships in a Sugeno fuzzy logic decision process which will determine the classification of the hand gesture signal.

The process of taking the first and second derivatives of the signal is noisy and these signals are filtered using Equation (26). Figure 13a shows the noisy unfiltered first derivative and Figure 13b represents the filtered signal after the low pass filter is applied.

![Figure 13: a. First derivative signal and b. Filtered signal.](image)

By evaluating the output of the first and second derivatives by employing Sugeno fuzzy methods, the number of fingers can be determined. The red vertical lines shown in Figure 14a displays the intersection maximums and minimums of the signal and their correspondence to the derivatives. Figure 14b shows the fuzzy logic output for a four finger representation in the fourth subplot.

![Figure 14: a. Four finger hand gesture signal with first and second derivatives and b. Output gesture summary for finger representation 4.](image)

8 Conclusion

More development on recognizing hand gesture feature characteristics such as the area under the curve can be explored. The progression of work on recognizing and validating more complicated hand gestures that represent American Sign Language will require more techniques, refinement and validation [10].

In this paper we investigated and validated the concept of multi-feature criteria to capture paused motion of the hand using the Kinect. The investigation showed that the multi-feature criteria of object detection within the KAR, the presence of skin color, and non-motion are essential in recognizing the hand gesture. We also demonstrated the use of Sugeno fuzzy logic and pattern recognition techniques to interpret the gesture’s intended meaning.

9 References