Hand Gestures Recognition and Tracking

Deepak Gurung, Cansen Jiang, Jeremie Deray, Désiré Sidibé

University of Burgundy

Visual Perception

Hand Gesture Recognition and Tracking

Submitted By:
Deepak Gurung
Cansen Jiang
Jeremie Deray

Submitted To:
Prof. Desire Sidibe

June 11, 2013

1 Introduction

Man-machine interaction is an important field in robotics community. One example is the capability of robots to detect humans and recognize gestures. This permits to take measures against any possible collision or passage blockage. This is a passive interaction. An active interaction could be recognizing the gesture of man and controlling robots.

Human computer interaction based on vision is a natural way for computers to interact. A number of solutions have been proposed in the current literature, but the problem is still far from being solved. Problems like self-occlusion, illumination variation etc makes this task challenging. Furthermore, these task must be performed in real-time.

In this project we develop a system that uses low cost web cameras to recognise gestures and track 2D orientations of the hand. This report is organized as such. First in section 2 we introduce various methods we undertook for hand detection. This is the most important step in hand gesture recognition. Results of various skin detection algorithms are discussed in length. This is followed by region extraction step (section 3). In this section approaches like contours and convex hull to extract region of interest which is hand are discussed. In section 4 a method is describe to recognize the open hand gesture. Two additional gestures of palm and fist are implemented using Haar-like features. These are discussed in section 5. In section 6 Kalman filter is introduced which tracks the centroid of hand region. The report is concluded by discussing about various issues related with the embraced approach (section 9) and future recommendations to improve the system is pointed out (section 10).

2 Hand Detection

Hand detection in our implementation is more of segmentation. Segmentation is the process of grouping points that belong to the same object into segments. The idea is to extract from image the set of points that describe the user's hand.

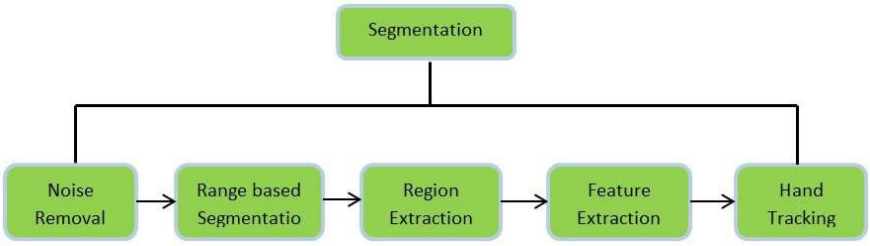


Figure 1: Flowchart of steps involved in skin segmentation

The cue mainly used for hand segmentation is the color information. The other method is to use the difference in image between consecutive video frames as cue for segmentation. We focused on using color information for segmentation. Some of the methods we tested are discussed below.

2.1 Method 1. Histogram based skin detection

In this approach 2D histogram for skin region and non-skin region are used for classification of each pixel. The method mentioned in [1] uses Bootstrap training method for an initial guess for skin pixels. Simply stated this means in-range thresholding. For the purpose of classification, this method uses normalized RGB color space.

2.2 Method 2. Naive Bayes using HSV color space

Our first approach was to train the classifier- Naive Bayes using dataset [2]. This dataset contains 153 images with 49 skin color images and 104 non skin color images. Each of the image has 500x500 pixels. That means 12.25 million skin color pixels and 26 million non-skin colored pixels. A skin probability map is developed using the Naive Bayes classifier.

2.3 Method 3. In-range thresholding in Log-Chromaticity Color Space (LCCS)

The LCCS is a 2D space obtained by taking the logarithms of ratios of color channels [3]. For our case the ratios $\log(R/G)$ and $\log(B/G)$ were considered. The paper [3] has explicitly provided ranges for 2D LCCS for which the skin colored pixels are projected. These ranges are $\log(R/G) = [0.15, 1.1]$ and $\log(B/G) = [-4; 0.3]$.

2.4 Testing Skin Detection Methods

Two test images are considered for measuring the variance of skin detection under varying condition of illumination. One of the image is taken outdoor under natural daylight illumination and the other image is taken inside room with lower relatively lower brightness condition.

(a) Image 1. Lower brightness condition

(b) Image 2. Natural day light illumination

Figure 2: skin pixel under varying illumination condition

Naive Bayes based skin probability map (method 2)

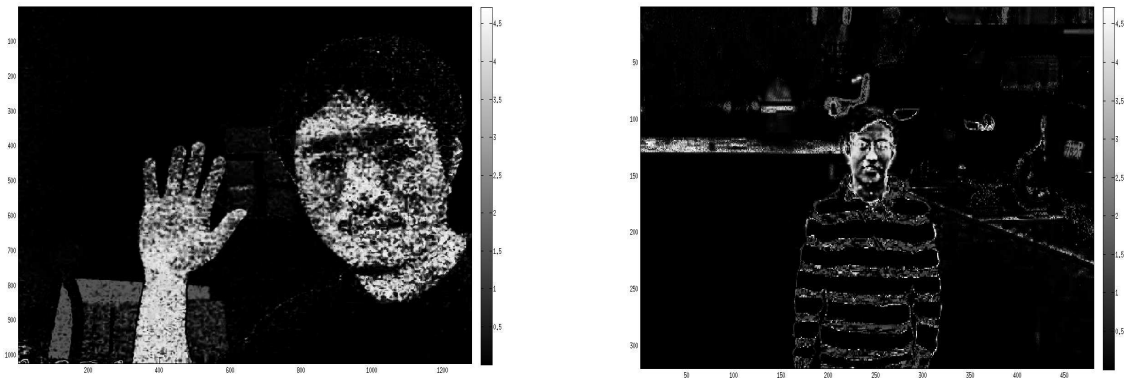


Figure 3: SPM generated using Naive Bayes. The grayscale bar on the right of each image shows the measure of probability of each pixel.

In-range thresholding in Log-Chromaticity Color Space (LCCS) (Method 3):

Figure 4: Binary mask obtained by in-range thresholding in LCCS

Discussion

From figure 3 and 4 we observe that the LCCS method yields better skin detection than that of spm based on histogram method. In image 1 spm gives a non zero probability for non skin region for example the book on lower left of the image. Even though for Image 2 false positive for LCCS method is observed to be higher than that of spm based on histogram (note the sidewalks and strips in shirt), we assume LCCS to be better mainly because of the extra amount of thresholding required for spm based method to obtain a final classified binary mask. It was observed that this threshold required for image 1 and image 2 are different thereby not invariant to varying illumination.

Comparision of LCCS with histogram based skin detection(Method 1) under varying illumination

We further compared LCCS method with that of histogram based skin detection. The results obtained are shown below.

(a) Binary mask obtained using method1 (histogram based detection)

(b) Binary mask obtained using method 3, LCCS

Figure 5: Skin detection for illuminated hand

(a) Binary mask obtained using method1 (histogram based detection)

(b) Binary mask obtained using method 3, LCCS

Figure 6: Skin detection for relatively low illuminated hand

Discussion

In figure 5 we observe that under properly illuminated condition both LCCS and spm based histogram method yeild good segmentation results. But under a relatively lower illumination condition histogram based method fails to detect any skin color (figure 6). The LCCS method outputs a superiro results compared to spm based on histograms.

3 Region Extraction

Once the skin regions have been detected, the next step is to determine which region corresponds to hands and which region corresponds to noise. There are possibilities that after skin detection there are small noisy regions. For example face region subtraction still leaves out some portions of head and neck. We assume that the regions corresponding to two hands will be the two largest regions.

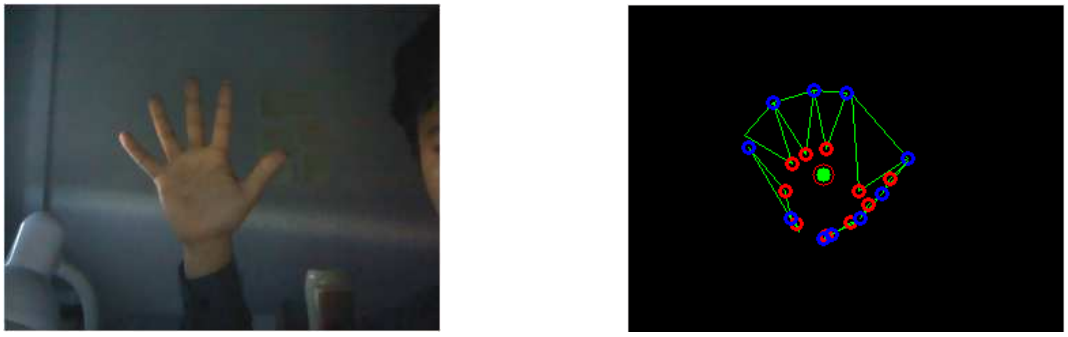
We first extract the contours of all the detected skin regions using the binary mask obtained from the skin detection process and connected component analysis. Suppose Ci is contour region I and the set of counter-clockwise perimeter coordinates $C_i(j) = (x_j, y_j)$ trace these contours. Let $N_i = |C_i|$ represent the total number of perimeter coordinates in the contour i. Small contours are rejected by thresholding. In order to be the potential hand region, the contour must have Ni above threshold σ ($\sigma = 90$ in the current implementation). Using Ni as a measure, we then choose the two largest contour corresponding to two hands.

4 Feature Extraction

After extracting hand region the next step is to compute gestures of hand by analyzing features of hand. For this we find the finger tips. Firstly, we compute the convex hull for set of points corresponding to contour of hands. The convex hull is used to find the envelope of set of points of contours we detected earlier in region extraction step.

In each of these convex hulls we compute the convexity defect. At each pixel j in a convex hull i, we compute the the curvature of defect using vector dot product. Say $[V_i(j), V_i(k)]$ and $[V_i(l), V_i(k)]$ are two vectors with $V_i(j)$ and $V_i(l)$ representing the starting and ending point of defect, and $V_i(k)$ is the point the deepest point of the convexity defect. Using the dot product and measuring the depth of deepest point we determine the finger tips of hand. We currently use a degree threshold $\theta = 30\,\text{deg}$ and depth threshold $\sigma = 30\text{pixels}$.

Figure 7 shows the result of detecting finger tips using convexity defect measures discussed earlier in this section.



(a) input image

(b) convex hull and convexity defects

Figure 7: Finger tip detection

In figure 7b convex defects are shown in blue colored circles. The blue colored point show the starting

or ending of convex defects. The line joining blue colored points and red colored points form an angle at the red colored points. This angle information is used as a cue for open hand detection. Futher more the depth of convex defects (the red colored point) is another cue. These cues of angle and defects can be used for open hand gesture recognition.

4.1 Open hand gesture

After computing the features and the determining the finger tips, we are able determine whether the hand is opened or closed. An open hand satisfies three constraints they are: no of points forming a contour, depth of defect in convex hull and the other is the angle between the fingers. The number of points that forms the contour must be greater than 90 ($N_i > 90$) (see section 3). The There must be atleast three fingers detected each of which separated by an angle of 30 degree near the defect (see section 4).

5 Haar-Like Features for hand gesture recognition

Voila and Jones used statistical measures for the task of face detection to handle the large variety of human faces. These faces may be rotated in three detections, different face colors, some faces with glasses, some faces with beard, etc. They use "Integral image" to compute rich set of Haar-like features. This approach has a very small computation time. The training algorithm of Viola-Jones considers samples to be, "positive" and "negative" samples. In training process, Haar-like features are selected at each stage to identify the images containing the object of interest.

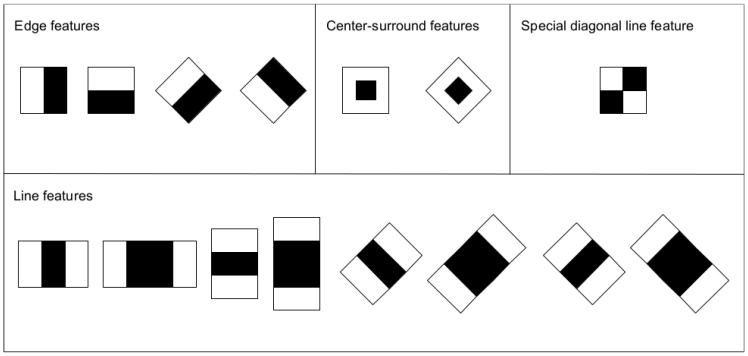


Figure 8: Rays back-projected from corresponding image points x, x' obtained from noisy camera do not meet at a point.

Haar-like features are described by a template that includes black and white rectangles, their relative coordinates to the origin of the search window and the size of the feature. The Haar-like feature set proposed by Lienhart [4] is shown in figure 8. The reason Haar-like features are used in classifiers instead of raw pixels is that these features can efficeintly reduce/increase the in-class/out-of-class variability which makes classification easier. And these features are more robust against noises and illumination variations than other features such as colors. Noises affect pixels values of all areas (black and white rectangles). Noises and illumination variations are effectively negated by computing the difference between these connected areas.

Haar-like features alone cannot identify an object with high accuracy. Adaboost learning algorithm is used by the object detection framework to both select the best features and to train classifiers that use them.

For our implementation we have used Haar-like features for detecting gestures. Each of these gestures must be trained separately. We used Haar-like features to train classifiers for palm gesture and fist gesture. Face detection is also performed using Haar-like feature trained classifiers. But face detection is performed solely to remove skin color pixels from video frames. They do not contribute to any gestures.

We used implemented classifiers using OpenCV function. OpenCV provides the trained data for front face. As for the palm detection we used the trained classifier from [5] and for fist detection we used the trained classifier form [6]. A comprehensive list of steps required to train the classifiers is given in [7].

5.1 Noise Removal and Face subtraction:

In practice, before applying classifiers onto the image sequences the image is first filtered by using Gaussian Low pass filters followed by median filtering.

In figure 9, 10 and 11 we can see that in the processing mask window, face is removed. As mentioned in section 5 face is detected using classifier based on Haar-like features and is subtracted from the input image sequence.

6 Tracking hand motion

To track the open hand gestures we consider the centroid of hand. The Kalman filter is used for tracking. It is a suitable tool for the purpose of predicting position of hand in the coming frame and after having measured the actual position of the hand in that frame, this prediction is corrected and the adjusted value is used for the prediction in the following frame. Kalman filter smooths tracking process.

Initialization

The state vector is represented as $\mathbf{x}(k) = (x(k), y(k), v_x(k), v_y(k))^\intercal$ where $x(k), y(k)$ represent the locations of the centroid in camera frame and $v_x(k), v_y(k)$ represent the velocity of the hand in the k^{th} image frame. It is assumed that between the $(k-1)^{th}$ and the k^{th} frames, the hand undergoes constant acceleration of a_k.

Kalman filter process model and measurement model consits of constructing a model transition matrix A, the process noise covariance matrix Q, the measurement covariance matrix R, the measurement transition matrix H found in the formulas,

$$\mathbf{x}(k) = A\mathbf{x}(k-1) + \mathbf{w}(k-1) \quad (1)$$

$$\mathbf{z}(k) = H\mathbf{y}(k-1) + \mathbf{v}(k) \quad (2)$$

where, $\mathbf{z}$ is the measurement vector of centroid position (x, y). $\mathbf{w}$ is model noise drawn from the distribution identified by Q and $\mathbf{v}$ is the measurement noise drawn from the distribution identified by R.

To formulate the parameters of Kalman filter [] is referred. All parameters are assumed to be independent and no converison of measurment is required. Thus H is identity matrix. Matrix A is given as,

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & \text{Drag} & 0 & 0 \\ 0 & 0 & 0 & \text{Drag} & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where Drag represents the drag on the velocity. This value is kept less than 1 for drag and 1 for no drag.

7 False Gesture removal

Sporadically when changing the hand gestures by humans the system detects false gestures. Such false detection can also occur in highly illuminated scene. We have dealt with this problem by designing a frame buffer of size 10 which stores the lastest 10 images of video. From this frame buffer the gesture having highest count is considered as a gesture.

8 Results of gesture recognition

Open hand gesture Recognition

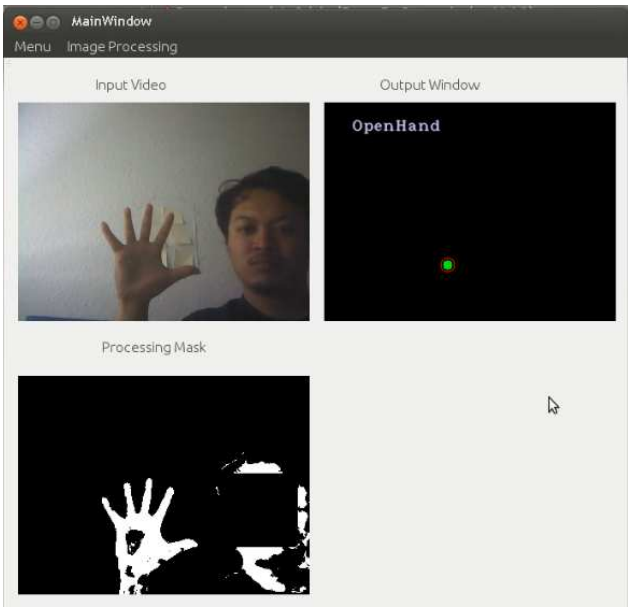


Figure 9: open hand gesture recognition.

Palm gesture Recognition

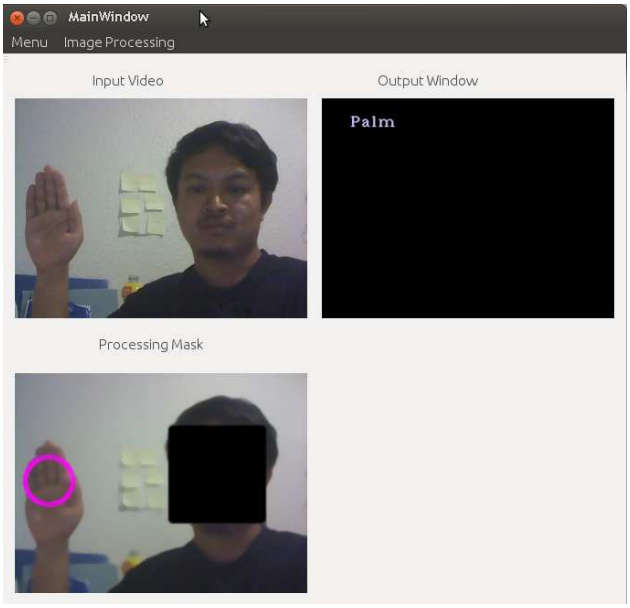


Figure 10: palm gesture recognition.

Fist gesture Recognition

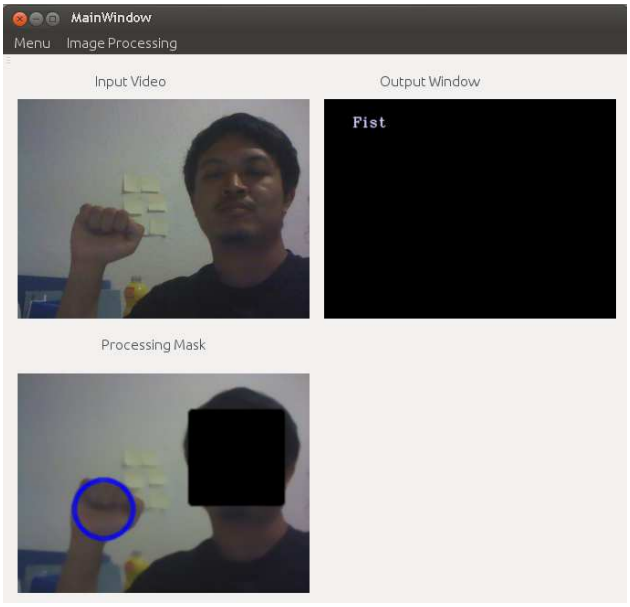


Figure 11: fist gesture recognition

Discussion

In final version of our system we implemented LCCS feature for skin detection mainly because of its robustness to varying illumination condition.

In figure 9, the output window shows centroid (green colored point). This point which is controlled by open hand gesture and smoothed using Kalman filter. In output figure 10 and 11 there is no green dot. This is the case of when no tracking is performed.

9 Overall Discussion

We built a gesture recognition software. Video feed from web camera is used for recognition. Recognition mainly consititutes of hand detection and feature extraction. Most of the existing gesture recognition system assume a static background. In developing our current system we were motivated to make gesture recognition work on varying illumination condition. For this a huge part of our research efforts were devoted on finding a robust skin detection algorithm.

Three different methods were tested for hand detection. The first one was histogram based pixel classification considering HSV color space, second was naive bayes approach in HSV color space and the third one was inrange thresholding in Log-Chromatic Color Space (LCCS). We compared performance of each of these methods under varying illuminatioon conditions. It was observed that using LCCS yielded better results. This has been implemented in our final software.

After detection of probable hand region contours are computed. Contour area is used as a cue for determinig hand region. Since there is a possibility that objects other than hands may also have same area as hand two more constraints are imposed. One is the depth of defect in convex hull and the other is the angle between the fingers. There must be atleast three fingers detected each of which separated by an angle of 30 degree near the defect.

Two other gestures of hand are recognised using Haar like features. Haar-like feature is used so as to make the recognition more robust. One may reason as to why an extra work of using Haar like features must be done on images since hand regions are already extracted in hand detection step. One of our underlying reason is to make the algorithm robust under varying condition of light. Using the constraints of contour

area and convexity defect alone is not enough to make the hand gesture recognition viable under varying illumination.

10 Future Work

While the system works fairly well for three different hand gesture recognition, there is still room for improvement. Even though we have designed the system for varying illumination the skin detection step still misclassifies some pixels.

One improvement to the current system would be gesture recognition based on template matching. On doing so we can differentiate between mutliple single-finger gestures. Left/right hand detection could be taken into account for additional gesture recognition.

The current sysyem performs a number of image processing including filtering and segmentation on every acquired image. Even thoug the speed of our system in gesture recognition is quite good, we could possibly improve the recognition rate significantly by tracking local features like hands and fingers. We have already used Kalman filters to track the hand position. But this tracking is limited to tracking position only. As pointed out in paper [8] this tracking can be transferred to the entire segmented hand region. The only downfall in this approach would be that the hand must not move too quickly. But in such a case reinitialization of tracker based on current implementation could be performed.

References

[1] Gomez, G., Morales, E. (2002, July). Automatic feature construction and a simple rule induction algorithm for skin detection. In Proc. of the ICML workshop on Machine Learning in Computer Vision (pp. 31-38).

[2] http://code.google.com/p/ehci/downloads/list

[3] Khanal B., and Sidibe D., "Efficient Skin Detection Under Severe Illumination Changes and Shadows", To appear in ICIRA 2011. 4th International Conference on Intelligent Robotics and Applications, Aachen, Germany, 6-9 December, 2011.

[4] R. Lienhart and J. Maydt, An extended set of Haar-like features for rapid object detection, in Proc. IEEE International Conference on Image Processing, vol. 1, 2002, pp. 900903.

[5] https://github.com/chenosaurus/drone/blob/master/xml/palm.xml (last accessed 10 June 2013)

[6] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, J. Handler. "A Real-Time Hand Gesture Interface for Medical Visualization Applications". Applications of Soft Computing : Recent Trends. Springer Verlag, Germany, 2006. vol. 36, pp. 153-163.
Available at: http://code.google.com/p/ehci/source/browse/trunk/data/aGest.xml?r=223

[7] http://note.sonots.com/SciSoftware/haartraining.html (last accessed 10 June 2013)

[8] H. Lahamy, D. Lichti., "Real-time hand gesture recognition using range cameras." Proceedings of the Canadian Geomatics Conference, Calgary, Canada. 2010.