Hand Gesture Recognition Using Microsoft's Kinect

Shashank Sonkar & Akshay Kumar
Advisor : Dr. Amitabha Mukerjee
Department of Computer Science and Engineering
{ssonkar,kakshay}@iitk.ac.in

April 15, 2012

ABSTRACT

Hand Gesture is one of the most natural ways to give commands to the computer or communiate with a robot. The other plausible possibility, speech recognition, requires a lot of learning to be done on part of the computer/robot as it is user specific. In this project, we use z-depth obtained by Kinect to determine accurately different hand gestures. The advent of low cost Microsoft's Kinect has provided new opportunities & spurred research in areas of human-computer interaction (HCI).However, Hand Gesture Recognition remains a relatively unexplored facet of Kinect. The complexities involved in hand recognition owing to its small size relative to the human body & its complex shape further complicates the scenario. We use a novel distance metric method for computing dissimilarity between two different hand gestures - Finger-Earth over's Distance (FEMD).

1 INTRODUCTION

Hand Gesture Recognition is one of the frontier areas of research in Computer Science because of its extensive applications in virtual reality, sign language recognition & computer games. Unfortunately, traditional hand gesture recognition approaches have faired poorly in real-life applications. Traditional hand gesture recognition methods use data from optical sensors too identify the posture of the hand. In this case, demarcating the portion in the image which has the hand is quite a non-trivial task. Morevoer, the quality of the captured image is sensitive to lightning conditions & cluttered background due to the limitations of the optical sensor. An innovative approach to avoid these shortcomings is to use data glove. These sensors are more reliable & are also relatively insensible to lightning conditions or cluttered background but it requires the user to wear a data glove and often requires calibration which impairs its usefulness.

Kinect works quite well to track a large object but due to its fairly low resolution, *viz.* 640X480, it is not that efficient in detecting & segmenting a small object from an image *e.g.* a human hand and hence it is still an open problem to use Kinect for hand gesture recognition. This situation is aptly described by the given figure:

Local distortions also add to Kinect inaccuracy. The two fingers in the hand shown above are indistinguishable by classical shape recognition methods like shape contexts and skeleton matching.

In order to overcome this difficulty, the method deploted is a distance metric called Finger-Earth Mover's Distance (FEMD). It is specifically designed for hand gestures and considers each finger as a cluster & penalizes unmatched fingers. This method has been tested by us on a 9-gesture set dataset and gives fairly accurate results.

2 HAND DETECTION

The Kinect sensor is used as the input device, which gives a XYZRGBA color image & depth map at 640X480 resolution.

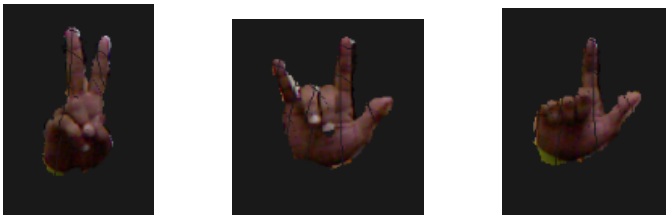
2.1 Hand Segmentation

We require the user to cooperate in two aspects, both of which are reasonable requirements in HCI. These two inherent assumptions are:

- The hand must be the frontmost object facing the sensor which, we believe, is quite a reasonale assumption we are making while working with Kinect as the basic Image Processing technique of the Kinect depend on the z depth of the image obtained. In this way, a portion of the image can be segmented by segmenting the nearest depth position with a certain gap as shown the figure.

Hand segmented for Gestures 1, 5, 6, 9 respectively

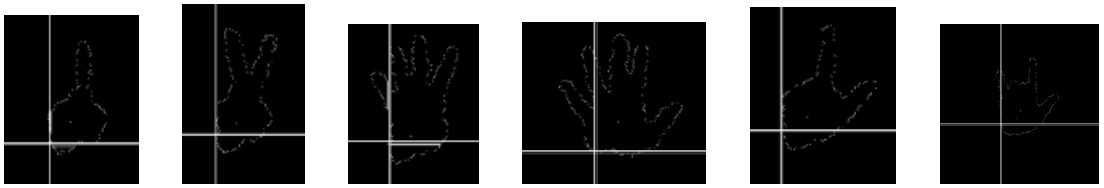
- The user needs to wear a green belt (or any belt of some suitable color not cofusing with the color of skin) on the wrist of hand used for the gesture. Has this constraint been not imposed, there has to be some other way of segmenting only the portion of the hand till the wrist. *For e.g.,* constraining the hand of the user to appear in a aproiri fixed 3D cube in space which in turn limits the usefulness of the Kinect.



Band segmented for Gestures 2, 8, 9 respectively

We get the location of the colored band from the image obtained after first point by finding the depth mode the color of the colored band. The image correspoding to depth varying from nearest depth position to nearest depth position plus 0.15 is segmented in the first step. Then, an array of size 30 is built each of whose element is equal to the number of points with corresponding depth. The element with the maximum value corresponds to the depth of colored band. Another incentive of this approach is that the effects of outliers are also eliminated completed. The method used in the paper [1] cited by us is RANSAC but it tends to work quite slow on a dynamic range of inputs.

After this the RGB image of the hand is coverted to Gray Scale by changing the RGB values of all the point with a non-NAN depth to 255. This is done because Distance Transform function needs a Gray Scale image rather than RGB image. Finally, the contour of this image is obtained and the one with the maximum number of points is selected. This contour is used for further analysis.



Contour for Hand Gestures 1, 2, 4, 5, 6, 8 respectively. The center point is the cyan point whereas the point of intersection of two lines is the red point.

2.2 Shape Representation

After detecting the hand shape, we represent it in the form of a histogram as shown in the screenshots on second last page. This histogram is used for classification & clustering of shapes. Firstly, two points, the center point - cyan point and the corner point - red point are determined. These two points are already depicted in the previous two figures (color not visible due to gray scale conversion).

The center point is the point with the maximal distance after applying distance transform on the shape hand. In Distance Transform, each point on the contour is given a value of 1 and every other point is assigned 0.

Nextly, the corner point, red point is determined by using the location of colored band. The mode value gives the depth of the band. Next, the red point is the point with depth equal to the mode value and situated at the left most upper region on the band. These two points are needed to get the histograms for the next step.

2.3 Threshold Decomposition

Now a histogram is obtained as shown in the screenshots on second last page. The horizontal axis denotes the angle between each contour vertex and the initial point relative to the center point. The vertical axis denotes the normalized Euclidean distance between the contour vertices and the center point. For normalizing, firstly the radius of the maximum inscribed circle is subtracted from each of the element of the array. If the value of a particular element drops below zero, it is adjusted to zero. Finally, this histogram is normalized by dividing each element by the sum of all elements of the histogram. The histograms obtained have been shown in the figure. It captures nice topological properties of the hand.

3 HAND GESTURE RECOGNITION

We now compare the output histogram to the template histograms of each gesture and find out their FEMD values. The input gesture is recognized as the class with the least dissimilarity value.

$$c = argmin_c FEMD(H, T_c)$$

where H = Input Hand, Tc = Template of Class C, FEMD (H,Tc) = FEMD between the input and each template.

3.1 Finger-Earth's Mover Distance

Earth Mover's Distance (EMD) is a measure of the distance between the probablity distributions (here two histograms) over a region D. In laymen's language, if there are two different ways of piling up a certain amount of dirt over a region, then EMD computes the minimum cost of turning one pile into the other, where cost is assumed to be amount of dirt moved times the distance by which it is moved. The locations of earth piles and holes denotes the mean of each cluster in the signatures, the size of each earth pile or hole is the weight of cluster and the ground distance between a piel and a hole is the amount of work needed to move a unit of earth. However, the current EMD has two disadvantages:

- Two hand shapes differ mainly in global features, the fingers, not local features. Large number of local features slow down its speed. Hence, it is better to look for global features in contour matching.
- EMD allows for partial matching *i.e.,* it gives zero dissimilarity value if a signature and its subset are measured. This is illogical.

FEMD addresses these two issues. It considers input hand as a signature with each finger as a cluster and adds penalt on empty holes to alleviate partial matches on global features.

Let $P = \{(p_1, u_1), ..., (p_{360}, u_{360})\}$ be the first histogram with 360 bins & $Q = \{(q_1, v_1), ..., (q_{360}, v_{360})\}$ be the second histogram with 360 bins. p_i & q_j denote the locations of the bin with weights u_i & v_j resectively. The EMD is then defined as:

$$EMD(P, Q) = min \frac{\sum_{i=1}^{360} \sum_{j=1}^{360} d_{ij} f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij}}$$

with following constraints:

$$\sum_{j=1}^{360} f_{ij} \leq u_i, \quad \sum_{i=1}^{360} f_{ij} \leq v_j, \quad \sum_{i=1}^{360} \sum_{j=1}^{360} f_{ij} = min\{\sum_{i=1}^{360} u_i, \sum_{j=1}^{360} v_j\}, \quad f_{ij} \geq 0$$

This can be viewed as a constrained optimization transportation Linear Programming (LP) problem

$\mathbf{D} = [d_{ij}]$ is the ground distance matrix of signature P and Q and is defined as the minimum moving distance for interval u_a to overlap with v_b, *i.e.:*

$$\begin{aligned} d_{ij} &= 0, && p_i \textit{ totally overlap with } q_j \\ &= |i - j|, && \textit{otherwise} \end{aligned}$$

The penalty term for the FEMD is $E_{empty} = |\sum_{i=1}^{360} u_i - \sum_{j=1}^{360} v_j|$ Hence, the value of FEMD is given by,

$$\begin{aligned} FEMD(R,T) &= \beta E_{move} + (1-\beta)E_{empty}, \\ &= \beta \sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij} f_{ij} + (1-\beta)|\sum_{i=1}^{360} u_i - \sum_{j=1}^{360} v_j|, \end{aligned}$$

Note that that parameter β is adjustable & can be varied according to the desired results. We have kept the value of β equal to 0.5.

4 RESULTS

The Hand Gesture Recognition used by us is effective in the first step in segmenting out the hand. However, it poses some difficulty during the band detection process due to cluttered background. Appreciable results are obtained only after few trials. After that the contour is obtained over which FEMD is applied the second gesture being the one's from the template & the one with the minimum FEMD value is the required gesture.

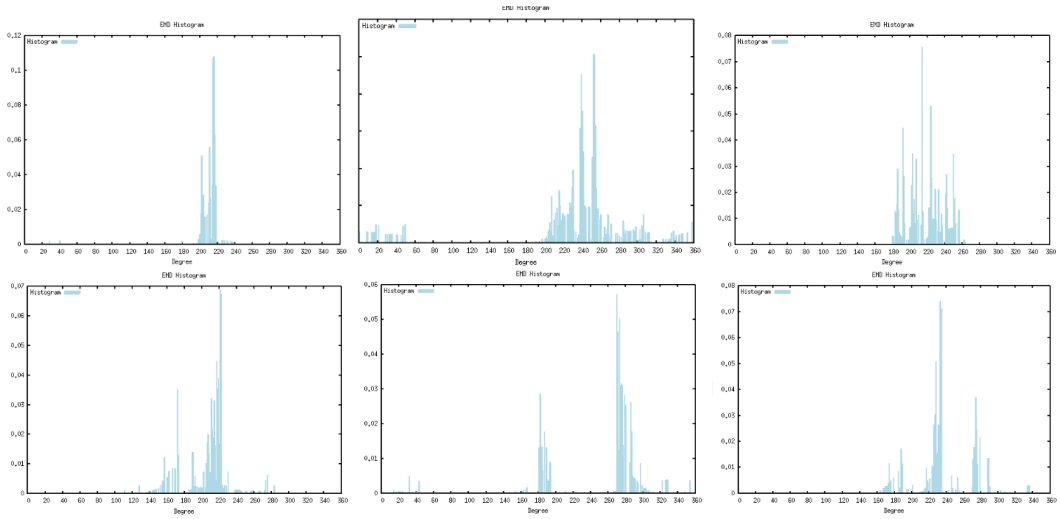
The table given here shows the FEMD values for a total of 9 different gestures:

Template →	1	2	3	4	5	6	7	8	9
1	2.22703	4.88081	12.5464	5.63506	6.00428	4.95483	30.4162	16.936	9.88055
2	8.11018	4.94938	8.41658	11.0984	7.74803	13.2025	23.2504	9.75817	11.1484
3	14.822	13.8366	3.86206	14.4521	13.9143	20.1565	16.5028	5.53534	12.6941
4	4.61536	3.7917	15.4221	2.42261	2.69318	4.77751	23.9533	14.4237	4.21001
5	4.77436	4.91516	13.9101	5.61602	1.67878	5.10229	21.0837	12.0395	4.95121
6	6.0795	9.36261	20.1011	7.88622	14.3552	3.97967	27.6912	16.9642	8.61706
7	22.7472	18.7734	14.4799	13.7488	14.3201	16.2942	6.13627	11.0334	17.1329
8	26.5933	29.3429	16.246	27.1864	30.1829	35.1021	12.9547	11.5799	15.7907
9	64.5633	63.7881	65.4581	56.2733	57.7328	58.0296	73.1934	62.9683	29.9087

As evident from the table, the red colored values along the principle diagonal are minimal thereby signifying the correct recognitioin of various hand gestures.

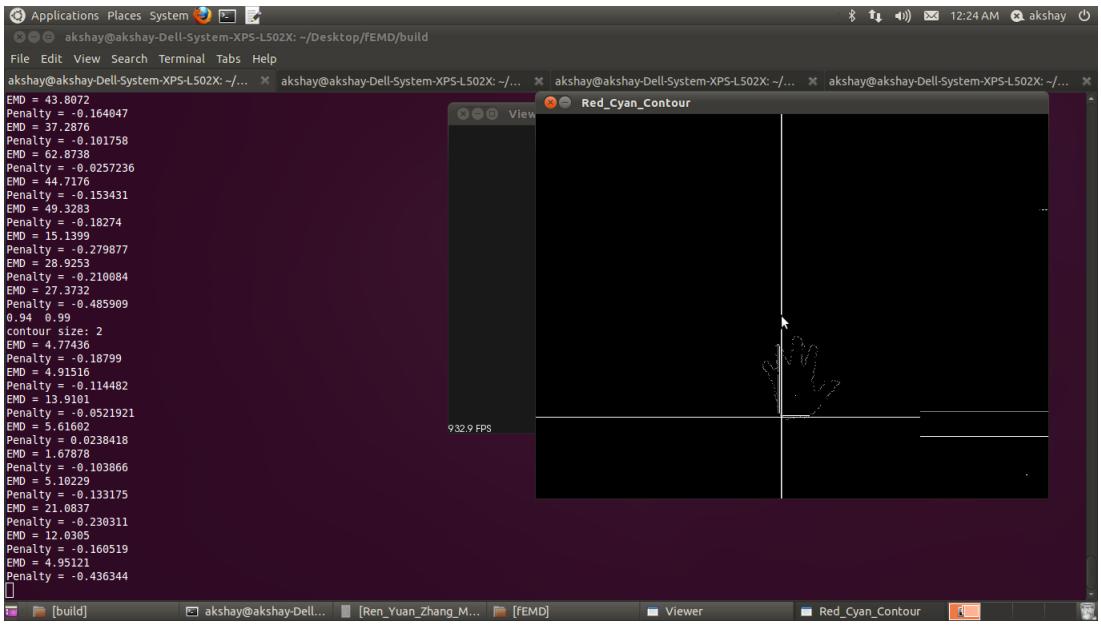
4.1 Screenshots

Furthermore the screenshots of the corresponding histograms for nine gestures are given on below.



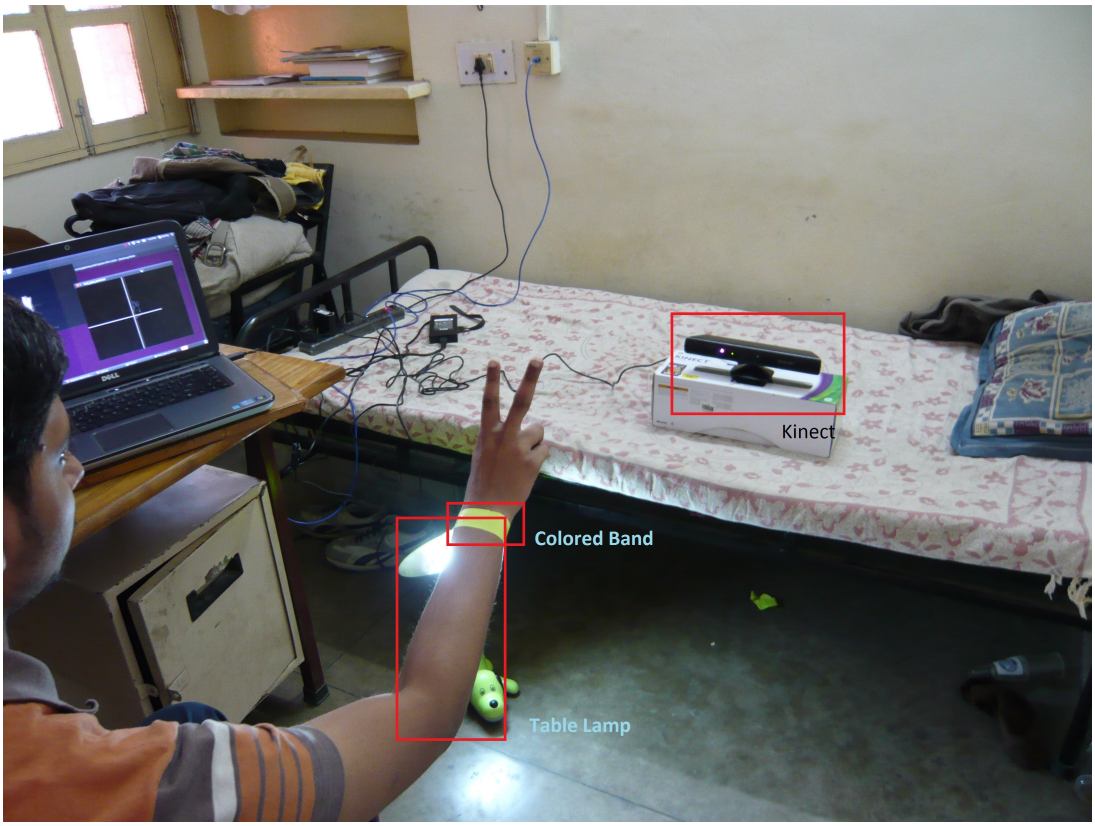
Histograms for Hand Gestures 1, 2, 4, 6, 7, 8 respectively

The screenshot here shows the result obtained for Gesture 5:



5 FURTHER IMPROVEMENTS

The results obtained by us have not been totally satisfactory according to our expectations. The primary reason of discrepency is the ineffectiveness of the Kinect to detect colored band accurately. The Kinect captures image at a low resolution and is highly affected by the positioning of the object with respect to its depth. If there is a way to somehow do away with presence of colored band over the wrist, the results can be improved significantly.



A real time image of the Kinect. The optimum results were obtained at the depth of around 1 meter.

6 REFERENCES

The whole code for the above project has been written by us from stratch. We have used PCL (Point Cloud Library) & OpenCV for the coding aspects and gnuplot for making histograms.

[1] Zhou Ren, Junsong Yuan, and Zhengyou Zhang. Robust Hand Gesture Recignition Based on Finger-Earth Mover's Distance with a Commodity Depth Camera. *MM'* 11: 1093-1096, November 28-December 1, 2011, Scottsdale, Arizona, USA.
[2] Haibin Ling, and Kazunori Okada. An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison. *ECCV' 06.*
[3] All the images used in the paper have been obtained by us except for the first image which has been taken from reference[1]