International Journal of Internet of Things 2015, 4(1A): 1-6
DOI: 10.5923/c.ijit.201501.01

Real-Time Hands Detection in Depth Image by Using Distance with Kinect Camera

Mostafa Karbasi[1,*], Zeeshan Bhatti[1], Parham Nooralishahi[2], Asadullah Shah[1], Seyed Mohammad Reza Mazloomnezhad[1]

[1]Faculty of Information Computer Technology, IIUM University, Malaysia
[2]Faculty of Computer Science and Information Technology, UM University, Malaysia

Abstract Hand detection is one of the most important and crucial step in human computer interaction environment. This paper presents a distance technique for hand detection based on depth image information get from Kinect sensor. Distance was used as the method of this study for hand detection. First, Kinect sensor was used to obtain depth image information. Second, background subtraction and iterative method for shadow removal applied to reduce noise from the depth image. Then, it used the official Microsoft SDK for extraction process. Finally, two hands could be segmented based on different color in specific distance. The experiment result shows that hands and head can be detected in a different position with good accuracy.

Keywords Kinect, Hand Detection, Depth Image

1. Introduction

Recently, the hand detection is being widely applied to many applications. One of the interesting research area is replacing control device with vision based Natural Human Interaction (NHI). Vision based methods are non-invasive, and many algorithms have been proposed using intensity or color images from one or more video camera. The difficulties of gesture recognition of using regular camera include following.

(a) Noisy segmentation of hand from complex background and changing illumination.
(b) Simple feature extracted from images produce ambiguities.
(c) Hand tracking methods suffer from initialization and tracking failures.

Thus, low computational-efficiency and lack of robustness motivate researchers use Microsoft Kinect camera for practical gesture recognition.

Fig. 1 shows a Kinect with the cover removed (Miles, 2012), included: infrared projector, infrared camera, RGB camera and microphones.

Infrared camera to detect environment space depth output for 640×480 resolution video (low frame rate can be up to 1280×1024 resolution).When an Xbox 360 gaming platform is connected, the available range of 1.2 ~ 3.5 meters, the maximum range reaches six square meters. Kinect sensor field is design with 57° level angle, 43 ° vertical angle, and 27 ° of elevation. Kinect depth image of map formats: an unsigned 16-bit 1 channel (grayscale) image, among them low 12 bits is effectively information. The actual distance has been converted into grayscale.

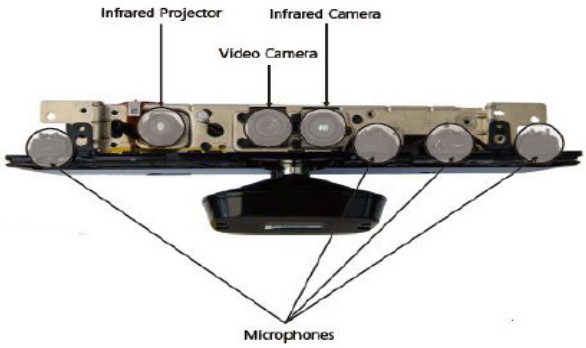


Figure 1. Kinect Camera

2. Literature Review

There is large body of information about hand detection. In recent years, there are many method has been developed for hand detection using depth by different devices such as Kinect and Zcam .some of researcher worked on hand segmentation, hand counter, color distribution, etc.

Kd-tree structure is used (Suau, Ruiz-Hidalgo, & Casas, 2012) to obtain color candidate cluster.

* Corresponding author:
mostafa.karbasi@live.iium.edu.my (Mostafa Karbasi)
Published online at http://journal.sapub.org/ijit

Blue and yellow cluster can be merged since Hausdorff distance between them is small enough. (Park, Hasan, Kim, & Chae, 2012) proposed adaptive hand detection approach by using 3-dimentional information from Kinect and tracks the hand using GHT based method. 3D hand model with label vertices was proposed by (Yao & Fu, 2012). During the training stage both RGB and depth data used as a input variable and produced two classifier such as random forest and Bayesian classifier. Per-78pixel hand part classification was obtained by forest classifier. (Chen, Lin, & Li, 2012) used hand region growing techniques which includes 2 steps. In the first stage, hand position detection was obtained by using hand moving with a velocity. In second stage tried to segment entire hand region by using region growing technique on 3D point. Hybrid RGB and TOF were proposed (Ren, Meng, Yuan, & Zhang, 2011) and (Van den Bergh & Van Gool, 2011). They used skin color segmentation based on two methods. a) Gaussian mixture model (GMM), was trained offline and it was able to detect skin color under lighting condition. b) Histogram-based method which was trained online. Hybrid method can be obtained by multiplying the GMM-based skin color probability with histogram based skin color probability. Representing hand in box used by (Frati & Prattichizzo, 2011). They used threshold approach to compute bounding box. Depth data which is too close and too far from Kinect are considered as zero with fixing lower and upper thresholds. The function “cvFindContours ()” was used to find the counter for the object. Combine both a training stage and estimation stage was used by (Yao & Fu, 2012). RGB and depth images used as an input in training stage and generate random forest classifier and a Bayesian classifier. The former is for the per-pixel hand parts classification; and the latter is employed as an object classifier to locate hand. Threshold Method is a simple method of depth which was used to isolate the hand by (Mo & Neumann, 2006) and (Breuer, Eckes, & Müller, 2007) and (Liu & Fujimura, 2004) and (Biswas & Basu, 2011). Depth thresholding determines the hands to be used to those points between some near and far distance thresholds around the Z (depth) value of the expected centroid of the hand – which can be either predetermined and instructed to the user, or determined as the nearest point in the scene. (Hamester, Jirak, & Wermter, 2013) perform foreground segmentation on depth images to reduce the region of interest. After that, edge detection in the foreground depth image provides a set of candidate contours. Randomized Decision applied this method for accurate hand detection and pose estimation with great accuracy result. (Keskin, Kıraç, Kara, & Akarun, 2012) introduced novel method to tackle the complexity problem. The idea is to reduce the complexity of the model by dividing the training set into smaller clusters, and to train PCFs on each of these compact sets. Most of researcher tries to detect hand and head by using Kinect skeleton which can detect and track hand and head easily. (Xiao, Mengyin, Yi, & Ningyi, 2012) and (Zainordin, Lee, Sani, Wong, & Chan, 2012) used skeleton model for hand detection. They usually crop hand based on coordinate x, y which obtains from the skeleton. However, when the subject's hand is far from the sensor, the captured hand is small in the captured image.

3. Shadow Modeling

Shadow modeling is basically model of a shadow proposed by (Khoshelham & Elberink, 2012). The Kinect is organized sensor consisting of 3 parts which are one infrared laser emitter, one infrared camera and one RGB camera. Firstly, the laser source emits a constant pattern of speckles into the scene. Secondly, infrared camera try to captured speckles are reflected. This measurement is called triangle process. Based on this structure, a model is built to present the cause of shadow in this section and a mathematical expression for shadow offset is also explained.

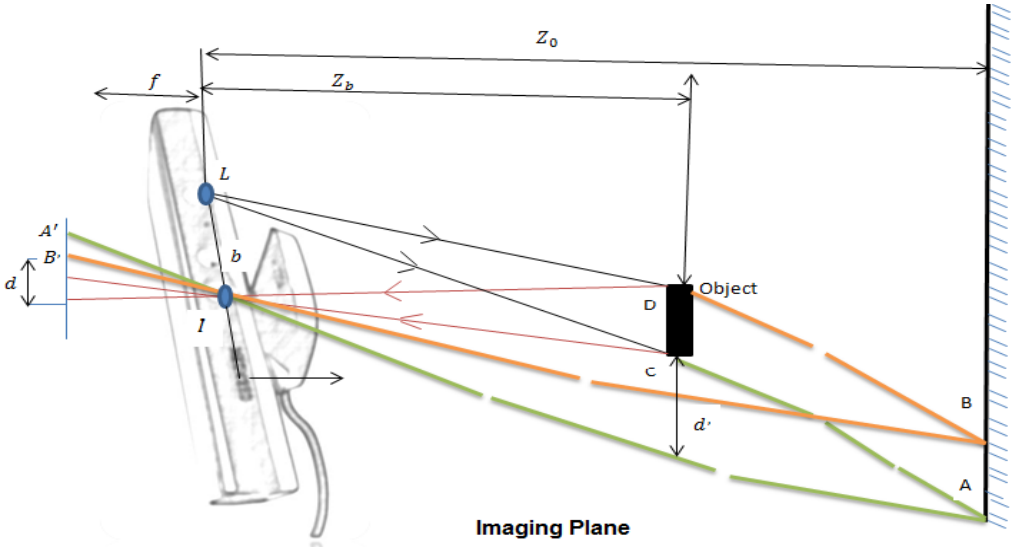


Figure 2. Cause of shadow

3.1. Cause of Shadow

Figure. 2 presents the cause of the shadow. It illustrates a simple scene consisting of one object and one background. Their distances to the sensor are z_o and z_b, respectively. L denotes the laser emitter and I denote the infrared camera. Suppose a number of speckles are projected onto the scene, as is shown in Figure 2 by solid lines. Some of the speckles hit the background directly while some are blocked by the object. We extend the line LC and LD to background and get a region marked as AB. Obviously, region AB is not reached by any speckles. As a result, A´B´, its corresponding region on imaging plane, receive no speckles from the scene. In other word, depth in region A´B´ is not measured and shadow is formed. From the projection model bellow, we conclude that the shadow is an area of the background where the speckles from the laser emitter cannot reach due to obstruction by an object. In other word, shadow is the projection of the object on the background. This model also explains why shadow is always presented on the right side of the object.

4. Methodology

This research was conducted to identify human hands in exact area .Given a collection of videos, we focused on the problem of interpreting the output of human hand model in order to identify with different color. The proposed approach is illustrated in Fig. 3.

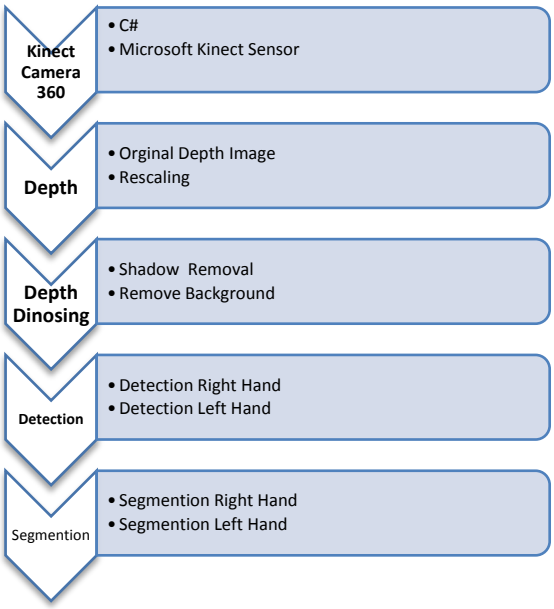


Figure 3. Overview of the proposed Work

4.1. Background Subtraction

Depth images, as they are provided by a Kinect, suffer from quite unusual noise. The best way was to calculate distance. In this method, by determining the minimum and maximum distance, the background can be removed. For Mindepth and Maxdepth, fix value were used as following statement.

Mindepth = Fix value
Maxdepth = Fix value

By comparing Figure 4 and 5, it can be seen that the pixel intensity was clustered in the middle of Figure4 while the pixel intensity in Figure 5 is discrete.

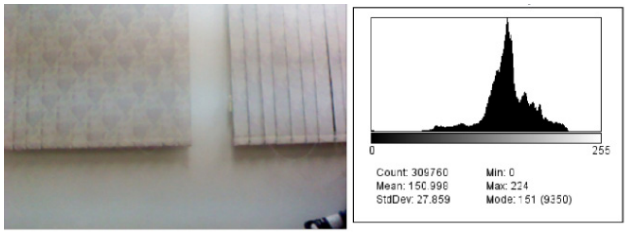


Figure 4. Original RGB image

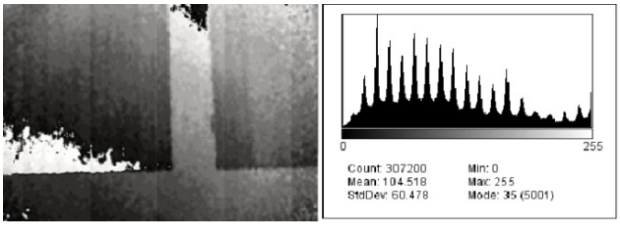


Figure 5. Depth Map of the image

Finally, all pixels from the background were removed as shown in Figure 6. Figure 7 shows more consolidated step by step results of all redundant data that has been removed from the scene.

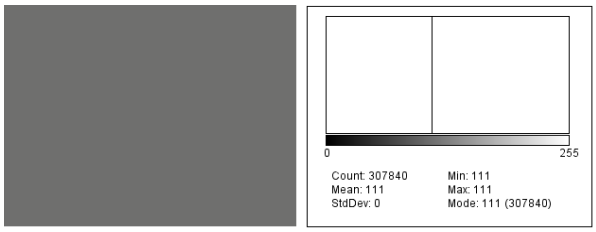


Figure 6. Background Subtraction

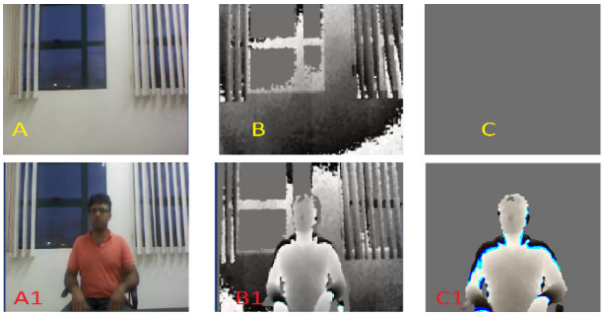


Figure 7. Steps of background subtraction

4.2. Shadow Removal

The shadow removal is one of the main contribution to this paper. This technique uses an iterative process that propagate values from known depth values to unknown depth values in the same segmented cluster. This solution provides a unique and novel. A bin is considered missing if its depth value is bellow MinDepth = 500 mm, and, after experimenting with a

few values, we chose MaxDepth = 2500 mm; and K = DepthLength. The source code performed very well, and with the exception of the image corners, it was able to get sensible estimates in depth. Because this process is related to a diffusion, the results automatically have the depth information interpolated. Figure 8 shows the result of denoising step by step in using this technique.

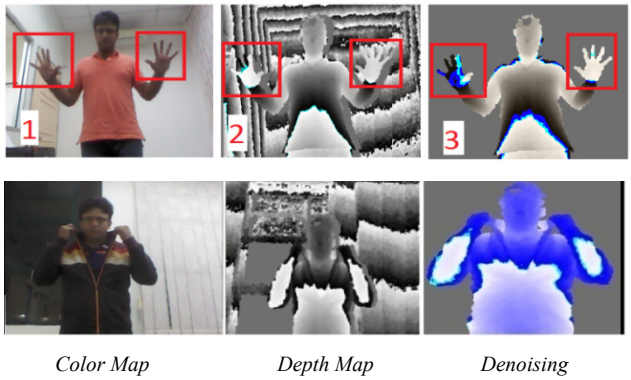


Figure 8. Shadow removal

Based on the Figure 8, we compared the pixel capacity of noising and denoising data given in the graphs shown in Figure 9. This result was based on 640 x 480=307200 depth resolution. It can be seen that the pixel capacity of depth map was around 160000 pixels, while under denosing process, this pixel capacity has decreased to around 60000 pixels values. Overall, this technique reached the best shadow removal and could successfully enhance the depth denoising by using Kinect 360 camera.

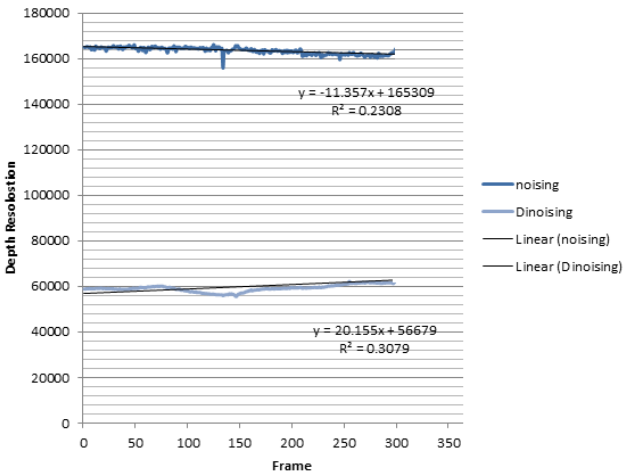


Figure 9. Comparison the pixel capacity of noising and denoising data

4.3. Hand Detection

Robust hand detection is the most difficult problem in building a hand gesture-based interaction system. There are several cues that can be used: appearance, shape, color, depth, and context. In problems like face detection, the appearance is a very good indicator. The eyes, nose and mouth always appear in the same configuration, with similar proportions, and similar contrasts. Unfortunately, this is not the case for hand detection; due to the high number of degrees of freedom, and the resulting shapes and shadows, the hand appearance can change almost indefinitely.

For hand detection user need to make sure that the hand is the front most objects facing the sensor. The hand region is cropped on the basis of centroid coordinate of hand obtained from skeleton model. The following algorithm shows the step involves in hand detection with different color for each contour.

```
DepthImagePixel[] depthPixels = new (depthPixels)

DepthImagePixel[depthFrame.PixelDataLength];

byte[] colorPixels = new(colorPixels)

byte[depthFrame.PixelDataLength * 4];

depthFrame .CopyDepthImage PixelData To
(depthPixels);

    int minDepth = 500;
    int maxDepth = 2500;
    int colorPixelIndex = 0;

for     int i = 0;
 i < depthPixels.Length     ++I;

short depth = depthPixels[i].Depth;
      intensity = ( byte)
(depth>= minDepth) &&
(depth <= maxDepth ? depth : 0);

intensity1 = 0;
intensity2 = 255;
intensity3 = 255;

End for

If      (depth<1500)

If       1000<Depth <1200

intensity1 = 0;
intensity2 = 200;
 intensity3 = 155;

If   depth <1000    then

intensity1 = 250;
 intensity2 = 200;
 intensity3 = 0;

else


intensity1++;
intensity2++;
```

```
intensity3++;

End If;

If      (depth < 1450) & (depth > 1200)
intensity1 = 0;
 intensity2 = 100;
 intensity3 = 155;
End If;

colorPixels[colorPixelIndex++] = intensity1;
colorPixels[colorPixelIndex++] = intensity2;
colorPixels[colorPixelIndex++] = intensity3;

Bitmap bitmapFrame
ArrayToBitmap (colorPixels, depth Frame.Width, depth Frame. Height, Pixel Format. Format32bppRgb );
```

5. Experimental Result

This tested on 7 videos, including detecting right hand beside the body (video 1), right hand front of the body (video 2), Left hand beside the body (video 3), left hand front of the body (video 4), both hand beside the body (video 5), both hand close to body (video 6), both hand front of the body (video 7).

These 7 videos show that background subtraction and denoising are removed completely. This is parallel with sign language recognition [2012] where she detected head and hands by RGB camera.

Video 1

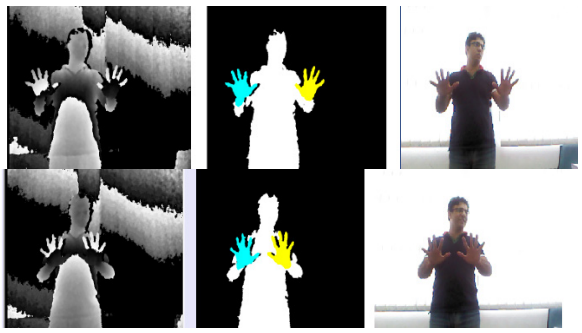
Video 2

Video 3

Video 4

Video 5

Video 6



Video 7

6. Conclusions

In this paper, a real time hand detection algorithm on depth images was proposed. it is tried to remove background subtraction and shadow completely, based on distance method also hands with different coulters has detected.

The findings from this study will help researchers to detect hands for human computer interaction (HCI). In addition, the study is significant to researchers who work in sign language area.

7. Future Work

This present study has some limitation based on situational restriction. Moreover, when the colored hand moves out of the defined boundaries, it cannot be detected by Kinect. So in future work, this limitation should be investigated further and solved by other researcher.

REFERENCES

[1] Biswas, KK, & Basu, Saurav Kumar. (2011). *Gesture Recognition using Microsoft Kinect®.* Paper presented at the Automation, Robotics and Applications (ICARA), 2011 5th International Conference on.

[2] Breuer, Pia, Eckes, Christian, & Müller, Stefan. (2007). Hand gesture recognition with a novel IR time-of-flight range

camera–a pilot study *Computer Vision/Computer Graphics Collaboration Techniques* (pp. 247-260): Springer.

[3] Chen, Li, Lin, Hui, & Li, Shutao. (2012). *Depth image enhancement for kinect using region growing and bilateral filter.* Paper presented at the Pattern Recognition (ICPR), 2012 21st International Conference on.

[4] Frati, Valentino, & Prattichizzo, Domenico. (2011). *Using Kinect for hand tracking and rendering in wearable haptics.* Paper presented at the World Haptics Conference (WHC), 2011 IEEE.

[5] Hamester, Dennis, Jirak, Doreen, & Wermter, Stefan. (2013). *Improved estimation of hand postures using depth images.* Paper presented at the Advanced Robotics (ICAR), 2013 16th International Conference on.

[6] Keskin, Cem, Kıraç, Furkan, Kara, Yunus Emre, & Akarun, Lale. (2012). Hand pose estimation and hand shape classification using multi-layered randomized decision forests *Computer Vision–ECCV 2012* (pp. 852-863): Springer.

[7] Khoshelham, Kourosh, & Elberink, Sander Oude. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors, 12*(2), 1437-1454.

[8] Liu, Xia, & Fujimura, Kikuo. (2004). *Hand gesture recognition using depth data.* Paper presented at the Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on.

[9] Miles, Rob S. (2012). *Learn Microsoft Kinect API*: O'Reilly Media, Incorporated.

[10] Mo, Zhenyao, & Neumann, Ulrich. (2006). *Real-time Hand Pose Recognition Using Low-Resolution Depth Images.* Paper presented at the CVPR (2).

[11] Park, MS, Hasan, Md Mehedi, Kim, Jaemyun, & Chae, Oksam. (2012). *Hand detection and tracking using depth and color information.* Paper presented at the Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV'12).

[12] Ren, Zhou, Meng, Jingjing, Yuan, Junsong, & Zhang, Zhengyou. (2011). *Robust hand gesture recognition with kinect sensor.* Paper presented at the Proceedings of the 19th ACM international conference on Multimedia.

[13] Suau, Xavier, Ruiz-Hidalgo, Javier, & Casas, Josep R. (2012). Real-time head and hand tracking based on 2.5 D data. *Multimedia, IEEE Transactions on, 14*(3), 575-585.

[14] Van den Bergh, Michael, & Van Gool, Luc. (2011). *Combining RGB and ToF cameras for real-time 3D hand gesture interaction.* Paper presented at the Applications of Computer Vision (WACV), 2011 IEEE Workshop on.

[15] Xiao, Zheng, Mengyin, Fu, Yi, Yang, & Ningyi, Lv. (2012). *3D human postures recognition using kinect.* Paper presented at the Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on.

[16] Yao, Yuan, & Fu, Yun. (2012). *Real-time hand pose estimation from RGB-D sensor.* Paper presented at the Multimedia and Expo (ICME), 2012 IEEE International Conference on.

[17] Zainordin, Faeznor Diana, Lee, Hwea Yee, Sani, Noor Atikah, Wong, Yong Min, & Chan, Chee Seng. (2012). *Human pose recognition using Kinect and rule-based system.* Paper presented at the World Automation Congress (WAC), 2012.