TOWARDS AUTONOMOUS DEPTH PERCEPTION FOR SURVEILLANCE IN
REAL WORLD ENVIRONMENTS

Thesis
Submitted to
The School of Engineering of the
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for
The Degree of
Master of Science in Electrical Engineering

By
Gayatri Mayukha Behara

Dayton, Ohio

December 2017

TOWARDS AUTONOMOUS DEPTH PERCEPTION FOR SURVEILLANCE IN
REAL WORLD ENVIRONMENTS

Name: Behara, Gayatri Mayukha

APPROVED BY:

Vamsy P. Chodavarapu, Ph.D.
Advisory Committee Chairman
Associate Professor
Department of Electrical and
Computer Engineering

Guru Subramanyam, Ph.D.
Committee Member
Professor
Department Chair
Department of Electrical and
Computer Engineering

Vijayan K. Asari, Ph.D.
Committee Member
Professor
Department of Electrical and
Computer Engineering

Robert J. Wilkens, Ph.D., P.E.
Associate Dean for Research and Innovation
Professor
School of Engineering

Eddy M. Rojas, Ph.D., M.A., P.E.
Dean, School of Engineering

© Copyright by
Gayatri Mayukha Behara
All rights reserved
2017

ABSTRACT

TOWARDS AUTONOMOUS DEPTH PERCEPTION FOR SURVEILLANCE IN REAL WORLD ENVIRONMENTS

Name: Behara, Gayatri Mayukha
University of Dayton

Advisor: Dr. Vamsy P. Chodavarapu

The widespread emergence of human interactive systems has led to the development of portable 3D depth perception cameras. In this thesis, we aim to expand the functionality of surveillance systems by combining autonomous object recognition along with depth perception to identify an object and its distance from the camera.

Specifically, we present an autonomous object detection method using the depth information obtained from the Microsoft Kinect sensor. We use the skeletal joints data obtained from Kinect V2 sensor to determine the hand position of people. The various hand gestures can be classified by training the system with the help of depth information generated by Kinect sensor. Our algorithm then works to detect and identify objects held in the human hand. The proposed system is compact, and the complete video processing can be performed by a low-cost single board computer.

This work is dedicated to my parents and my advisor Dr. Vamsy Chodavarapu
All I have accomplished is only possible due to their constant support and guidance.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my heartfelt gratitude to my thesis advisor, Dr. Vamsy Chodavarapu, without whom this thesis would have been a castle in air. I am extremely grateful for his constant support, guidance and the opportunities he provided me throughout my graduate studies at University of Dayton. He constantly gave me valuable suggestions which greatly boosted my morale and research progress. It is unimaginable how much time and effort he had to spend to discuss, proofread and correct all my works. I will always be indebted to Dr. Chodavarapu for encouraging me to present research papers in international conferences, which not only improved my confidence but also gave me an opportunity to get feedback from leading researchers. All in all, I consider myself lucky to be his student.

I am extremely delighted to express my gratitude to Dr. Vijayan Asari and Dr. Guru Subramanyam for taking time out of their busy schedule to review this work and provide insightful comments. Also, the guidance and inputs they provided in my meetings with them was extremely helpful for my research and to grow as an individual. My heartfelt thanks to the department of electrical and computer engineering for constantly supporting me and providing me the graduate research and teaching assistantships to pursue my educational goals.

I would like to thank the members of Integrated Microsystems Laboratory (IML) Balaadithya Uppalapati and Akash Kota for all the research help and educational discussions. Thanks to their patience for being excellent subjects for my research work. Without them this thesis would have been a dream.

Special gratitude goes to my mother Bharati for her unfailing love and support. Her tremendous faith in me, encouraged me to pursue my masters education. Thanks to my father Gopal Krishna for his constant motivation in my tough times. He has been a role model.

Thanks to my friends and extended family for their continuous motivation throughout my graduate studies.

TABLE OF CONTENTS

ABSTRACT.....	iv
DEDICATION.....	v
ACKNOWLEDGEMENTS.....	vi
LIST OF FIGURES	x
LIST OF TABLES.....	xii
LIST OF ABBREVIATIONS.....	xiii
LIST OF NOTATIONS	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Depth Perception.....	1
1.2 Comparison of Different 3D Depth Cameras	3
1.3 Objective of the Study	5
1.4 Significance of Study.....	6
1.5 Outline of Thesis.....	6
CHAPTER 2 LITERATURE SURVEY.....	7
2.1 Review of Related Work.....	7
2.2 Kinect V2 Sensor Specifications	8
2.3 Principle of Kinect V2 Sensor	8
CHAPTER 3 SYSTEM ARCHITECTURE AND DEPTH MAP ANALYSIS	9
3.1 Architecture of System	9
3.2 RGB-D Registration.....	10
3.3 Depth Map Rendering.....	10
3.4 Depth Map Normalization	11
3.5 3D Depth Map Visualization and Point Cloud Generation	12

3.6 Depth Image Segmentation for Human Recognition.....	15
3.7 Skeletal Tracking	17
 CHAPTER 4 3D HAND SEGMENTATION AND INTERACTION OF VARIOUS OBJECTS WITH HAND.....	21
4.1 Hand Tracking System.....	21
4.2 Background Subtraction Algorithm.....	22
4.3 Gesture Recognition System.....	24
4.4 Hand Object Interaction System Overview	28
4.5 Depth Based Object Reconstruction for Human Object Interaction.....	30
 CHAPTER 5 CONCLUSION AND FUTURE WORK	37
5.1 Conclusion	37
5.2 Future Work	37
 REFERENCES	38

LIST OF FIGURES

Figure 1.1: Different 3D depth cameras	2
Figure 1.2: Comparison of depth cameras in terms of resolution ranges	4
Figure 1.3: Comparison of depth cameras in terms of field of view	4
Figure 1.4: Kinect depth camera principle block diagram.....	5
Figure 2.1: Principle of Kinect v2 sensor [14].....	8
Figure 3.1: System architecture	9
Figure 3.2: Sample images from the Kinect V2 sensor (a) RGB color sensor output. (b) Depth sensor output.....	11
Figure 3.3: Depth map normalization [15]	12
Figure 3.4: 3D Point cloud tracking two humans	13
Figure 3.5: 2D colormap of depth image	14
Figure 3.6: 3D colormap of depth image	15
Figure 3.7: Human segmentation algorithm	16
Figure 3.8: ROI detected from peaks	17
Figure 3.9: Human segmentation results	17
Figure 3.10: Skeletal tracking outline	18

Figure 3.11: Human detection and tracking output	19
Figure 3.12: Color skeleton projection of human detection and tracking.....	20
Figure 4.1: Hand tracking	22
Figure 4.2 Background subtracted hand image.....	23
Figure 4.3 Hand detection.....	23
Figure 4.4: Open hand state	25
Figure 4.5: Both hands in open state.....	25
Figure 4.6: Closed hand state.....	26
Figure 4.7: Both hands in closed state	26
Figure 4.8: Lasso state recognition	27
Figure 4.9: Object from the category ‘cup’	28
Figure 4.10: Initial and final background subtracted image with object and human.....	30
Figure 4.11: Depth map at different depth levels for human hand interaction	31
Figure 4.12: 3D Volumetric reconstruction of hand object interaction	31
Figure 4.13: 3D reconstruction for hand object interaction with category cup and box .	32
Figure 4.14: Experimental result with no human detection and hand object interaction .	33
Figure 4.15: Experimental result with human detection and hand object interaction	34
Figure 4.16: Experimental result at near view of camera	35
Figure 4.17: Experimental result at far view of camera.....	36

LIST OF TABLES

Table 3.1 Calibration parameters	10
Table 4. 1: Experimental results with no human and hand object interaction	33
Table 4. 2: Experimental results with no human and hand object interaction	34
Table 4. 3: Experimental result with human detection and hand object interaction in near view of camera	35
Table 4. 4: Experimental result with human detection and hand object interaction at far view of camera	36

LIST OF ABBREVIATIONS

Fov	Field of view
fps	frames per second
HoG	Histogram of Gradients
HoD	Histogram of depths
ToF	Time of flight
CRF	Conditional Random Field

LIST OF NOTATIONS

f_vV	vertical field of vision
f_vH	horizontal field of vision
X_r	horizontal resolution of depth map
Y_r	vertical resolutions of depth map.
X_p	coordinates of each pixel.
Y_p	coordinates of each pixel.
f_{xd}	Depth camera intrinsic (focal lengths in pixel units)
f_{yd}	Depth camera intrinsic (focal lengths in pixel units)
C_{xd}	Depth camera intrinsic (optical centers/image centers)
C_{yd}	Depth camera intrinsic (optical centers/image centers)
f_{xrgb}	RGB camera intrinsic (focal lengths pixel units)
f_{yrgb}	RGB camera intrinsic (focal lengths pixel units)
C_{xrgb}	RGB camera intrinsic (optical centers/image centers)
C_{yrgb}	RGB camera intrinsic (optical centers/image centers)
R, T	Extrinsic matrix between color and IR cameras
b	Baseline between IR camera and IR projector
d_{off}	Depth offset

CHAPTER 1

INTRODUCTION

1.1 Depth Perception

Depth perception means the visual ability to see the world in three dimensions (3D) based on the distance of the objects from the vision system. In the past decade, depth acquisition has become a subject of extensive research and this has reflected in the extensive commercial production of depth cameras [1]. The term “depth camera” refers to specific hardware that can acquire the depth information in a scene either by using a sensor or by running a stereo vision algorithm on the color frames. However, the suitability of different depth cameras for any specific application depends on its technical capabilities. The widespread emergence of human interactive gaming and entertainment systems using body gestures for control has led to the development of portable 3D depth perception cameras [1], [2]. Many standalone systems capable of 3D depth perception are now commercially available. Examples of such systems include Kinect motion sensing input device developed by Microsoft for Xbox 360 and Xbox One video game consoles [2], Creative Labs Senz3D [3], and ZED camera from Stereo labs [4], which has combined a 3D Camera for depth sensing with motion tracking. Recently companies like Panasonic, Fotonics has announced and released new depth camera models [5].

Typically, the 3D camera systems work by illuminating the proximity scene with infrared radiation and then detecting the reflected infrared radiation to measure Time-of-Flight (ToF) difference between the illuminated and reflected light. Further, depth cameras are mostly based on two operating principles.

- i. Pulsed light modulation
- ii. Continuous wave amplitude modulation

Different commercially available 3D depth cameras are shown in Figure 1.1. The pulsed light modulation cameras like ZCam of 3DV systems has very short time intervals to achieve a resolution which corresponds to only a few centimeters in depth [6]. Whereas, the continuous wave modulation camera measures the phase shift between the emitted and received modulated light which directly corresponds to the Time-of-Flight and in turn to the depth, where problems in the form of multiples of the modulation wavelength may arise [7].

Figure 1.1: Different 3D depth cameras

The ToF sensors face two key issues, low resolution and low sensitivity due to high ambient noise. In contrast, depth cameras that use the structured light approaches, which are used in laser scanners, cannot give high frame rates for full images at good resolution. Thus, instead of a time varying pattern, the Microsoft Kinect V2 sensor uses irregular pattern which consists of many dots that are produced by the infrared laser and a diffractive optical element. It relies on a novel image sensor that indirectly estimates the time taken by the pulses of laser light to travel from a laser projector, to the target surface, and then back to an image sensor.

1.2 Comparison of Different 3D Depth Cameras

3D cameras such as the PMDTec, CamCube have measurement errors for objects that are very close to the camera due to over exposure [8]. The PMD based ToF cameras use a wobbling pattern. The emergence of Microsoft Kinect for Xbox gaming system has opened new avenues for 3D perception due to its color and depth video processing capabilities. The Kinect sensor will determine the disparities between the emitted light beam and the observed position of the light dot with using a 2-megapixel grayscale imager. The identity of the dot is determined by utilizing the irregular pattern. The distance to the reflecting object is triangulated using the on-board hardware.

Figure 1.2: Comparison of depth cameras in terms of resolution ranges

Kinect depth sensor has a good field of view, high resolution image capability, robustness and high computational efficiency compared to other cameras such as PMDTec, CamCube 41k, Zess Multicam and SR4000. Figure 1.2 shows the comparison of depth cameras with respect to their resolution ranges. Figure 1.3 shows the comparison of depth cameras with respect to their field of view.

Figure 1.3: Comparison of depth cameras in terms of field of view

1.3 Objective of the Study

The Microsoft Kinect depth perception camera which has lower distance calculation error in its working distance makes it a good prototype system for use in surveillance applications. The Kinect V2 is a time of flight (TOF) sensor and it captures data at 30 frames per second (fps) matching the camera, that's about 60 million bits per frame. It has the highest field of view (FoV) which is $57^\circ \times 47^\circ$ compared to all other commercially available portable 3D depth cameras [9]. We have selected the Microsoft Kinect sensor for designing our system because it can densely cover the full scene with its IR structured light pattern, and thus can obtain depth measurements for essentially every pixel [10]. It records depth in the same 11-bits per sample precision, same as the original Kinect, which is sufficient throughput needed for a 1080p resolution TOF sensor.

Figure 1.4: Kinect depth camera principle block diagram

The objective of this thesis is to expand the functionality of 3D depth perception by combining autonomous object recognition along with depth perception which could

provide the ability to both identify a object and its distance from the camera by using video processing algorithms that are fast and efficient. The Microsoft Kinect V2 sensor provides the color and depth videos as output for our surveillance application processing. Using both the color and depth videos, we implement a hand detection algorithm which ensures robustness to cluttered background and complex ambient scenes. The block diagram for the Kinect depth camera working principle is shown in Figure 1.4.

1.4 Significance of Study

The Microsoft Kinect sensor system uses direct time of flight (TOF) measurement. TOF sensors are essentially small infrared “radar-type” sensors that instantly create a depth map. There are several cues like the shape, color and depth based on which any object or a human can be monitored using a vision system. Of these cues, depth perception gives important useful information required for classification and detection of people and objects. Furthermore, depth based surveillance system will not get affected by any sudden illumination changes or moving shadows. Such capability for the system would prove invaluable to autonomous surveillance applications, where persons carrying any forbidden and dangerous objects are detected in real-time and appropriate warnings can be signaled.

1.5 Outline of Thesis

The thesis is organized into five chapters including this chapter. The second chapter deals with the system specifications and the literature survey. The third chapter gives a detailed description of the system and the fourth chapter discusses the experimental results of the surveillance system. The fifth chapter is a brief outline of the whole system and the future scope of the developed system.

CHAPTER 2

LITERATURE SURVEY

2.1 Review of Related Work

Real time hand detection from full body images has been previously proposed by Karbasi et al. [11], in which the users hand should be the front most body part facing the sensor. The hand region is then extracted based on the centroid coordinate of hand obtained from the skeleton model. The depth data generated from the motion of hand is applied to a hierarchical Conditional Random Field (CRF) which helps to recognize the hand gestures from the hand motions. The hierarchical CRF method is used to detect candidate segments of gestures using hand motions, and then a Boost Map embedding process is used to verify the hand shapes of the segmented signs. For detecting the 3D locations of the hands and face, the depth information generated from Microsoft Kinect is used [12]. The Kinect sensor and its SDK provide a reliable human tracking method if a constant line of sight is maintained. In another work, a human recognition method is developed based on color and depth information that is provided from any RGB-D sensor [13]. Initially a mask is created based on the depth information of the sensor to segment the shirt from the image, later the color information of the shirt is obtained for recognition. The shirt segmentation is based on depth information; hence it is light invariant compared to the color based segmentation.

2.2 Kinect V2 Sensor Specifications

The Kinect V2 sensor has a wide-angle time-of-flight camera, and it processes two gigabits of data per second. The new Kinect has greater accuracy with three times the fidelity over its predecessor Kinect V1 and it can track without any visible light by using an IR sensor. It has a 60% wider field of vision (FoV) compared to the first version. The sensor consists of an RGB color camera that can capture 1920×1080 pixel resolution video and a depth video with 512×424 pixel resolution. It can detect up to 13 feet or 4m from the user.

2.3 Principle of Kinect V2 Sensor

The Kinect 2.0 sensor works on the principle of ToF where each pixel is divided into halves. The device is turned on and off at a fast rate. When it is on, it absorbs the photons of the laser light. When it is off, it rejects the photons. The other half of the pixel does the same thing; however, at 180 degrees out of the phase. Also, the laser light source is pulsed in phase with the first pixel half such that, if the first half is on, so is the laser. And if the pixel half is off, the laser is off as well. The working principle of Kinect v2 sensor is shown in Figure 2.1 [14].

Figure 2.1: Principle of Kinect v2 sensor [14]

CHAPTER 3

SYSTEM ARCHITECTURE AND DEPTH MAP ANALYSIS

3.1 Architecture of System

The 3D depth perception based autonomous surveillance system consists of various sections as shown in Figure 3.1. Each section deals with specific video processing algorithms for autonomous human motion tracking and object recognition. The 3D depth maps of various objects are created for training the database and are later used for testing when the object to be detected is held in the subject hand.

Figure 3.1: System architecture

3.2 RGB-D Registration

The Kinect sensor provides a color and depth image that are not properly aligned. This occurs because of the difference in intrinsic parameters of the cameras and the locations of the cameras. Kinect calibration to align the depth image onto the color image is called registration. The main aim of calibration is to get the following parameters in Table 3.1

Table 3. 1: Calibration parameters

Parameter	Description
f_{xd} and f_{yd}	Depth camera intrinsic (focal lengths in pixel units)
C_{xd} and C_{yd}	Depth camera intrinsic (optical centers/image centers)
f_{xrgb} and f_{yrgb}	RGB camera intrinsic (focal lengths pixel units)
C_{xrgb} and C_{yrgb}	RGB camera intrinsic (optical centers/image centers)
R, T	Extrinsic matrix between color and IR cameras
b	Baseline between IR camera and IR projector
d_{off}	Depth offset

The extrinsics relate the RGB camera to the depth camera. It is a 4×4 matrix containing rotation and translation values.

3.3 Depth Map Rendering

The depth video continuously generates 30 frames per second for the area in the FoV. In our program, we generate one depth image for every 100 frames to minimize the memory and signal processing requirements. Figure 3.2 (a) and Figure 3.2 (b) shows the sample images of RGB color sensor output and depth sensor output respectively.

(a)

(b)

Figure 3.2: Sample images from the Kinect V2 sensor (a) RGB color sensor output.

(b) Depth sensor output.

3.4 Depth Map Normalization

The depth image from the Kinect sensor has pixel values which are equal to the calculated depth of the object. The areas that scatter from the Kinect are filled with zero pixel values as the information of the image is missing in those places. These pixels must be filled before using the depth images for surveillance system design. The filling process

takes place by replacing the zero-pixel values with statistical mode of surrounding with 25 pixels. Depth map normalization is shown in Figure 3.3 [15].

Figure 3.3: Depth map normalization [15]

3.5 3D Depth Map Visualization and Point Cloud Generation

To reconstruct any object surface, all the individual views must be combined. Each view of the object is transformed to form the point clouds and the relative transformation between the two views should be found. After the transformation is done, the point clouds can be formed. The process of finding the appropriate transformation is called point cloud registration. By repeating this process for all the viewpoints will align all the point clouds and thus recreate the entire object surface. The depth image can be visualized in 3D coordinates and the point cloud. A point cloud is plotted from the Kinect sensor images and it is returned as a M-by-N-by-3 matrix. The function returns the point cloud units in

meters. The origin of a right-handed world coordinate system is at the center of the depth camera. The 3D depth point cloud visualization of a depth image is shown in Figure 3.4.

The 3D coordinates coming from the depth map are combined with the corresponding color information (as RGB values) to constitute a colorized point cloud. The Kinect depth camera has limited range. Hence, certain pixel values in depth image will not have values for the corresponding 3D coordinates. These missing pixel values are set to NaN in the Location property of the returned point cloud.

Figure 3.4: 3D Point cloud tracking two humans

From the depth map, the depth values are stored in a matrix and they contain only the Z pixels which is the distance in space. The fvV and fvH are the vertical and horizontal field of vision of the Kinect sensor. The Xr and Yr are the horizontal and vertical resolutions of depth map. The Xp and Yp are the coordinates of each pixel. The X and Y coordinates of the camera are calculated from Eqs. (1) and (2), respectively.

$$X = Z \times 2 \tan\left(\frac{fvH}{2}\right) \times \left(\frac{Yp}{Yr}\right) \quad (1)$$

$$Y = Z \times 2 \tan\left(\frac{fvV}{2}\right) \times \left(\frac{Yp}{Yr}\right) \quad (2)$$

If colorimetric information is required, a transformation of the color frame must be performed because of its higher resolution. Figure 3.5 shows the 2D color map of the depth image and Figure 3.6 shows the 3D color map of the depth image.

Figure 3.5: 2D colormap of depth image

Figure 3.6: 3D colormapped of depth image

3.6 Depth Image Segmentation for Human Recognition

The depth image pixel values are analyzed, and noise pixel is assigned with a zero value. Also, the corresponding repetition of values is determined. Let S be the difference between the nearest and farthest distance of human body and $k = [k_1, k_2, \dots, k_n]$ be the vector which contains the peaks that are detected. We consider X as the vector for the sorted depth values which are ascending and Y be a vector of corresponding repetition of each measurement. An element Y_i is considered as peak if its value is greater than its two neighbors: The algorithm to segment the obtained depth map is shown in Figure 3.7.

Figure 3.7: Human segmentation algorithm

A region growing process is applied on separate objects which have same depth values. The main factors of the algorithm are the seed location and the growth threshold similar to the process described in reference [15]. All the detected regions are scanned horizontally and vertically to determine the location with highest density value. Figure 3.8 shows ROI detected from peaks. We can determine the threshold based on the histogram of the depth (HoD) of the region that is detected. The similarity between two pixels x and y are defined by:

$$Sim(x, y) = |depth(x) - depth(y)| \quad (3)$$

Figure 3.8: ROI detected from peaks

Finally, this region growing process will ensure that all neighboring pixels that are added belong to two existing depth layers of the detected region. Figure 3.9 shows the human segmentation results.

Figure 3.9: Human segmentation results

3.7 Skeletal Tracking

First, the depth image acquired from the depth video stream is passed through a threshold to extract the foreground. The outline of skeletal tracking is shown in Figure 3.10.

Figure 3.10: Skeletal tracking outline

The noise in the depth image is removed using morphological operations like erosion and dilation. Small blobs in the image are removed to extract a filtered foreground image. The upper body of the human is detected using the Viola-Jones algorithm with Haar feature classifiers [16]. The detector uses AdaBoost and constructs a rejection cascade of nodes [17]. Each node is again a multi-tree AdaBoost classifier with a low rejection rate. Hence, few regions are classified at each stage. The Distance Transform, DT , on the binary image is given by,

$$DT(p) = \min\{d(p, q).I(q) = 0\} \quad (4)$$

where I is the foreground segmented depth image. Equation (3) transforms the image with pixel values, where the coordinate is the Euclidean distance, EDT , to nearest intensity point. The distance transform may fail due to the projected lengths of the limbs. Hence, the extended distance transform is used for a point, p , with reference to a point, μ , and it is defined by the Equation (4) as,

$$EDT(p, \mu) = DT(p). \left(1 + \frac{|I_d(p) - I_d(\mu)|}{I_d(\mu)}\right) \quad (5)$$

From the Kinect sensor, the human skeleton is represented by 25 joint points and these points are projected on to the Kinect depth image. Up to six skeletons can be tracked by system in real-time in a complex environment. The skeletal projections are differentiated with different colors as overlays on the depth image. The skeletons can be tracked

irrespective of the body position of the human and then the skeletal projection can be viewed both on the RGB and depth image.

Figure 3.11: Human detection and tracking output

Also, the skeletal projection can be correlated with the RGB color image obtained from the color sensor camera in Kinect sensor. The differentiation of each individual skeleton is depicted through the different color projections of the skeleton on RGB images. The human detection and tracking output is shown in Figure 3.11.

Figure 3.12: Color skeleton projection of human detection and tracking

This implementation of the skeletal body image helps in tracking up to six individuals at a time in the FoV of the sensor. They are assigned different detection scores and each of the skeleton is assigned a unique number for differentiation. Figure 3.12 shows color skeleton projection of human detection and tracking.

CHAPTER 4

3D HAND SEGMENTATION AND INTERACTION OF VARIOUS OBJECTS WITH HAND

4.1 Hand Tracking System

The depth stream obtained from the Kinect has frame size of 512×424 in mono13 format and is uint16 data type. The z-axis value from the joint position of skeletal data is considered for hand tracking. Detection of a hand is the most complex part of this surveillance system. We set the Region of Interest (ROI) based on the center of coordinate and apply the background removal algorithm. Hand tracking plays a key role in the hand detection process. Most of the hand tracking methodologies are based on Kalman filter [18]. The hand motion is tracked based on the prediction of position of hand in current and previous frames. The length of the hand is calculated between both the converted points in the color coordinates. The center of the hand is calculated using the midpoint formula. In this approach, the left and right hands can be segmented from body image. By this method of hand tracking there will not be any effect with illumination changes. Figure 4.1 shows the hand tracking.

Figure 4.1: Hand tracking

4.2 Background Subtraction Algorithm

After the hand of the test subject is tracked, a background subtraction algorithm can be implemented. This will be useful for the gesture recognition and training process. The region of hand detected is segmented in the hand tracking process. The smallest depth value from this segmented image can be stored as the minimum depth value and the largest depth value can be stored as the maximum depth value. The wrist and the finger point data from skeletal image can be used to record the depth values. The pixels out of the depth range can be removed by thresholding. The threshold points are marked with black pixels. The color point value gives the maximum and minimum depth values for the color frame. The segmented image of hand from background subtraction algorithm is shown in Figure 4.2 and the detected hand can be seen in Figure 4.3.

Figure 4.2 Background subtracted hand image

Figure 4.3 Hand detection

4.3 Gesture Recognition System

Gesture recognition plays a key role for the detection of object in human hand. Kinect gives the joint position (X, Y and Z) of the human's hand. If some points of the hand move to relative positions for a given amount of time, then it is called a gesture. Most of the gesture recognition systems are based out of skin segmentation and contour tracking [19] [20]. But this is ineffective because the color images may be affected by illumination changes. Therefore, depth data is more useful to determine the hand gestures. The hand gestures can be recognized based on the joint position data provided by the Kinect sensor. There are totally 25 joint positions in a human body that a Kinect V2 sensor can identify. Therefore, after the human hand is tracked, the joint position data from the depth image will be helpful to identify the hand state. Based on the upper body joint positions the state of hand can be determined. The hand states data can be obtained for up to six individuals at any time.

Figure 4.4: Open hand state

Figure 4.5: Both hands in open state

Figure 4.6: Closed hand state

Figure 4.7: Both hands in closed state

The hand state identification helps with the object in hand interaction. The three main hand states that are needed for our surveillance system are The Open state, Closed state and

Lasso state. Figure 4.4 shows open hand state. Figure 4.5 shows the state of both hands open. Figure 4.6 shows closed hand state. Figure 4.7 shows the state of both hands closed. Figure 4.8 shows lasso state. The lasso state is the state where the fist is seen with index finger extending. This is the most suitable hand state for our surveillance applications as the interaction with the object will form a ring structure of hand around the object and only the index finger will be visible. Also, the joint positions give data about the finger positions. These can be useful for finger segmentation. The joint position consisting of the tip of left hand and the tip of the right hand can be considered. It is very important for identifying the finger position around the object for the detection.

Figure 4.8: Lasso state recognition

4.4 Hand Object Interaction System Overview

The hand object interaction plays a key role in the development of an autonomous surveillance system. Initially the depth image is captured after every 100 frames, this is preprocessed and a point cloud is computed along the normals of each vortex. After the hands of the human are tracked, the object model is incorporated in the humans hand. Different objects are trained to the system based on the color, shape and depth images. These are the three most essential cues for the 3D object training.

Figure 4.9: Object from the category 'cup'

The 3D depth and color images of the different objects are captured, and they are stored in the database like in Figure 4.9. We created a training database consisting of 6 different objects. These classes of objects are categorized based on the depth data from Kinect sensor.

The depth channel gives the 3D geometric information such as shape and size of the object. The color and shape are the two aspects based on which the object is categorized.

The depth gradients will not be affected by illumination changes, and thus, it can be used for the background subtraction. For the autonomous surveillance system, the real-time environment should be categorized into variety of objects. The database has 6 categories of objects. The 6 categories are ‘cups’, ‘boxes’, ‘mugs’, ‘books’, ‘bottles’ and ‘bowls’ respectively. Initially a mask is created for each image. And the edge detection is done by the canny edge detector. Finally, the normal map is calculated for each image. We use the histogram of gradients (HOG) method for the feature extraction [21]. The linear SVM is utilized for the learning task. A confusion matrix is generated for our algorithm and the average accuracy of our classifier is predicted for each case. The object in the hand is classified into one of the categories and it is extracted from the depth image in this step. The subtraction is done at various levels. This classifies the foreground pixel values and the background subtraction values. Later, the foreground pixels are divided into regions and tested for gradients at the boundaries as shown in Figure 4.10. The region of the object model is segmented to the input depth map using the 3D location of the user’s hand as described in the reference [22].

Figure 4.10: Initial and final background subtracted image with object and human

4.5 Depth Based Object Reconstruction for Human Object Interaction

Since the fingertips of the human will be in touch with the object, the seeds algorithm can be used to segment the hand and object. It should be noted that because of masking out the supporting plane, the object does not include points of that plane. The last step in object segmentation is to remove the object points that correspond to hands. To do so, we use the hand pose estimated by the hand tracker and we render a synthetic depth map of the user's hands. The volumetric construction of the hand object interaction utilizes the Kinect fusion algorithm [23]. Figure 4.11 shows the depth map for the human hand interaction. Figure 4.12 shows the volumetric reconstruction of hand object interaction. Figure 4.13 shows 3D reconstruction for hand object interaction with cup and box. The experimental results for different objects can be seen in Figure 4.14, Figure 4.15, Figure 4.16 and Figure 4.17. The object distance from camera and other parameters for surveillance system are tabulated in Table 4.1, Table 4.2, Table 4.3 and Table 4.4.

Figure 4.11: Depth map at different depth levels for human hand interaction

Figure 4.12: 3D Volumetric reconstruction of hand object interaction

Figure 4.14: Experimental result with no human detection and hand object interaction

Table 4. 1: Experimental results with no human and hand object interaction

Results Category	Number of detections
No of Humans detected (skeletons)	0
No of Hands segmented	0
Objects detected	0
Distance from 1 st object to camera	0 meters
Distance from 2 nd object to camera	0 meters

Figure 4.15: Experimental result with human detection and hand object interaction

Table 4. 2: Experimental results with no human and hand object interaction

Results Category	Number of detections
No of Humans detected (skeletons)	2
No of Hands segmented	2
Objects detected	2
Distance from 1 st object to camera	1.93171 meters
Distance from 2 nd object to camera	1.94426 meters

Figure 4.16: Experimental result at near view of camera

**Table 4. 3: Experimental result with human detection and hand object interaction in
near view of camera**

Results Category	Number of detections
No of Humans detected (skeletons)	1
No of Hands segmented	1
Objects detected	1
Distance from 1 st object to camera	1.40361 meters
Distance from 2 nd object to camera	0 meters

Figure 4.17: Experimental result at far view of camera

Table 4. 4: Experimental result with human detection and hand object interaction at far view of camera

Results Category	Number of detections
No of Humans detected (skeletons)	1
No of Hands segmented	1
Objects detected	1
Distance from 1 st object to camera	2.42961 meters
Distance from 2 nd object to camera	0 meters

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The autonomous object detection method using depth information obtained from the Microsoft Kinect sensor is useful for many surveillance applications. The results of the system proved to be very accurate and this system can be implemented in real world environments for surveillance. Our developed algorithm detects objects held in the human hand and this system is efficient in identifying the hand object interactions. The developed system is compact, and the complete video processing is performed by a low-cost single board computer.

5.2 Future Work

In a future work, the systems can be further enhanced to a portable system capable of identifying suspicious objects. The existing database can be further extended for various real-world objects.

REFERENCES

- [1] Tashev, “Recent Advances in Human-Machine Interfaces for Gaming and Entertainment”, Int. J. Inform. Tech. Security, 3, 3, 69-76 (2011).
- [2] XBOX one console. Available online: <https://www.xbox.com/en-US/>
- [3] Creative Senz3D. Available online: <https://us.creative.com/p/web-cameras/creative-senz3d>
- [4] ZED Depth Sensor 2K Stereo Camera. Available online: <https://www.stereolabs.com/>
- [5] Fotonic. Available online: <http://www.fotonic.com/>
- [6] ZCam 3DV systems. Available online: <http://ntuzhchen.blogspot.com/2011/04/3dv-systems-zcam-depth-camera.html>
- [7] Kinect principle and science behind it online:
https://www.gamasutra.com/blogs/DanielLau/20131127/205820/The_Science_Behind_Kinects_or_Kinect_10_versus_20.php
- [8] B. Langmann, K. Hartmann, and O. Loffeld, “Depth Camera Technology Comparison and Performance Evaluation”, Proc. ICPRAM, 438-444 (2012).
- [9] KINECT for Windows. Available online: <https://www.microsoft.com/en-us/kinectforwindows/>
- [10] J. Geng, “Structured-Light 3-D Surface Imaging: A Tutorial”, Adv. Optics Photonics, 3, 2, 128-160, (2011).

- [11] M. Karbasi, Z. Bhatti, P. Nooralishahi, A. Shah and S. Mazloomnezhad “Real-Time Hands Detection in Depth Image by Using Distance with Kinect Camera”, Int. J. Internet of Things, 4, 1-6 (2015).
- [12] C. Tang, Y. Ou, G. Jiang, Q. Xie, and Y. Xu “Hand Tracking and Pose Recognition via Depth and Color Information”, Proc. of 2012 IEEE Int. Conf. Robotics and Biomimetics (2012).
- [13] B. J. Southwell, and G. Fang, “Human Object Recognition Using Color and Depth Information from an RGB-D Kinect Sensor”, Int. J. Adv. Robotic Syst., 10, 171 (2013).
- [14] Kinect Principle and science behind it online:
https://www.gamasutra.com/blogs/DanielLau/20131127/205820/The_Science_Behind_Kinects_or_Kinect_10_versus_20.php
- [15] Depth Normalization. Available
online:<https://www.mathworks.com/matlabcentral/fileexchange/47830-kinect-depth-normalization>
- [16] T. H. Dinh, M. T. Pham, M. D. Phung, D. M. Nguyen, V. M. Hoang, and Q. V. Tran “Image Segmentation Based on Histogram of Depth and an Application in Driver Distraction Detection”, 13th International conference on control automation robotics & vision (icarcv), IEEE, 969 - 974, (2014),
- [17] D. Peleshko and K. Soroka, “Research of Usage of Haar-Like Features and AdaBoost Algorithm in Viola-Jones Method of Object Detection”, 12th International Conference on the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Polyana Svalyava, 2013, pp. 284-286., (2013).

- [18] F. Kadlcek, and O. Fucik, “Fast and Energy Efficient AdaBoost Classifier”,
Proceedings of the 10th FPGA world Conference, 1-5, September 10-12, (2013).
- [19] S. Vasuhi, M. Vijayakumar and V. Vaidehi, “Real Time Multiple Human
Tracking Using Kalman Filter”, 3rd International Conference on Signal Processing,
Communication and Networking (ICSCN), 1-6, (2015).
- [20] P. Dondi, L. Lombardi, and M. Porta, “Development of Gesture-Based Human-
Computer Interaction Applications by Fusion of Depth and Color Video Streams”,
IET Comput. Vis, 8, 568–578 (2014).
- [21] Y. Yao and Y. Fu, “Contour Model-Based Hand-Gesture Recognition Using the
Kinect Sensor,” in IEEE Transactions on Circuits and Systems for Video Technology,
24, (2014).
- [22] Jordan, Caleb. “Feature Extraction from Depth Maps for Object Recognition”,
(2013).
- [23] P. Panteleris, N. Kyriazis, and A. A. Argyros, “3D Tracking of Human Hands in
Interaction with Unknown Objects”, In BMVC,2, (2015)
- [24] <https://msdn.microsoft.com/en-us/library/dn188670.aspx>