3D Model Generation of Real-Objects Using Depth and Color Information

AliumBasir Sakib **14301010**
Hasnat Azam **14301024**
Tonni Das Jui **14301114**
FahimaAkter **14301015**

Supervisor: Dr. Md. Ashraful Alam

Department of Computer Science and Engineering,
BRAC University

Submitted on: 25 March, 2018

Declaration

We, hereby declare that this report is based on our own work and research for our thesis. The contents of the report are prepared by us for the thesis and sources of information and other materials that we used for help are acknowledged and mentioned here by reference. Thus, this report has not been submitted anywhere for any degree or award before.

Signature of Supervisor:

Dr. Md. Ashraful Alam

Assistant Professor

Department of Computer Science and Engineering

BRAC University

Signature of Authors:

Hasnat Azam

Fahima Akter

Tonni Das

Alium Basir Sakib

Acknowledgements

First of all we need to express gratitude toward Almighty ALLAH to begin with, the maker and the proprietor of this universe, who gave us direction, quality and capacities to finish this thesis Then, we would like to thank our thesis advisor Dr. AshrafulAlam for his valuable guidance, feedback and support till now. We are indebted to our parents, friends and teachers for their immense support and for the help they offered during various phase of our thesis.

Tonni Das,Md. AliumBasir

Fahima Akter, Hasnat Azam

Table of Contents

List of Figures

List of Tables

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
RGB	Red, Green, Blue

Abstract

We propose and demonstrate a method of generating 3D model using depth and color information of real-object using Microsoft Kinect sensor using our proposed pixel mapping algorithm. The primary idea of the method is to manipulate RGB and depth data within arrays to plot in three dimensional frame to generate the 3D colored model of any real-object. The proposed system comprises of 3 phases. The phases include acquisition of color and depth information of real objects, extraction and processing of RGB and depth data, generating 3D model using our proposed pixel mapping algorithm. Whereas, our proposed algorithm focuses on creating several 1D arrays for depth as well as the color data meaning Red, Green, Blue pixel values separately. By mapping the depth values accordingly pixel-wise 3D black and white model can be created where the background is also included. By limiting the depth values according to the objects position the background can also be removed to construct the focused object's 3d model. Furthermore, our algorithm maps the R, G, B arrays in the black and white model to add proper colors pixel-wise .As a result, the colorful RGB 3D model can be achieved.

Keywords: 3D model, Kinect Xbox ONE, Real-objects, Depth information, Color information.

Chapter 01

Introduction

1.1 Motivations and Goals

In this era, with the advancement of technology, 3D models have replaced the formal views of 2D images and models in every sphere to get more clarified idea and visualization of objects in various aspects. This idea is being widely used in technical purposes such as in film and animation industry, gaming industry and also for architectural purposes as well as for marketing purposes to grab people's' attention by providing them better overview of products. However, to create 3D models from scratch using basic software's and sketch is very time consuming and also need a great amount of effort to give it a definition that is closer to reality. Besides, it's costly too to create 3D models.

For the amazing presentation and visualization of 3D models, the implementation of 3D models are getting popular day by day among professionals and common people in every possible aspects. Programmers and designers who work in this area are continuously researching and working to improve previous methods and invent efficient newer methods to implement 3D models which are quite praiseworthy.

Due to this, we got motivation to work in this field for our thesis to contribute and propose a system or method to implement 3D models more easily which can generate models within a short amount of time; in other words, nearly within real-time .

1.2 Project Scopes

This thesis is done to create a 3D model taking images just from one angle .As this can be achieved only by an angel the images taken only once for depth and RGB without having to move the sensors position which will greatly help making 3D model creation much more easier and simple. As we are using Kinect sensor for making 3D there is no questions about its availability. Also if someone's on budget Xbox 360 Kinect can also be used for affordability which makes the whole generation of 3D model easy and low costing. Moreover, we can see now a day's console gamers are increasing day by day and they mostly have these Kinect sensors for playing.3D demonstrating in development additionally permits liveliness. Normally, the customer can envision a great deal more around an up and coming task that a level illustration would ever give. Customers can actually have a virtual walkthrough of the proposed assembling. Much the same as a 3D motion picture, a 3D show empowers the customer to get a vibe of how things will be laid out. They can stroll through the section of their future home, achieve the entryway and even picture visitors eating at the feasting region. They can see this even before a block has been laid out.

This thesis will inspire people to use those Kinect to some use and join the era of modeling easily and do further research on 3D models which can lead them to a much brighter future. This thesis will also help in the field of animation and model designing as the methods of creating animation model now are extremely high costing. Using our work to develop it further by capturing and creating models from more angles makes the whole 3D scenario much easier and cheaper with less effort.

1.3 Overview

In order to implement 3D model easily we researched and studied various available methods of generation of 3D models. In this thesis, we created 3D models from the color and depth data values acquired by Kinect Xbox-ONE with Matlab. Besides, we manipulated the data using one dimensional array in order to avoid the complexity of manipulating three dimensional arrays which are quite troublesome to understand.

Rest of the paper is organized as follows: Proposed method in chapter 3, Experimental setup and result in chapter 4, Conclusion in chapter 5 and references in the end.

Chapter 02

Literature Review

With the rapidly growing technological based industries, 3D model creation and its implementation in various sphere of life has been increasing in an enlarged scale. To create a 3D model of an object, the depth data and color data is needed at the very beginning. The demands of using 3D model varies in different industries as well as in case of applications according to need. Even the process and apparatus for building 3D models varies according to its environment. For instance, people use sonar and ocean satellite to construct 3D models in underwater. To enhance the comprehension of a human administrator driving a submerged Remotely Operated Vehicle,

U.Castellani reproduces 3D submerged condition by acoustic camera framework progressively, and the determination is around 5cm at the recurrence of the acoustic flag 500kHz[2-3].In gaming world, Video segmentation is an essential building hinders for extra ordinary state applications, for example, scene comprehension and communication investigation. While exceptional outcomes are accomplished in this field by best in class learning and model based techniques, they are confined to specific kinds of scenes or require a lot of clarified preparing information to accomplish protest division in nonspecific scenes. Then again, RGBD information, generally accessible with the presentation of purchaser profundity sensors, gives genuine world 3D geometry contrasted with 2D pictures [5]. Over the last several years numerous marketplaces specialized in 3D printing models have emerged. Some of the 3D printing marketplaces are combination of models sharing sites, with or without a built in e-com capability. Some of those platforms also offer 3D printing services on demand, software for model rendering and dynamic viewing of items, etc.

a constant framework for programmed formation of 3D confront models from a video succession is introduced in "An ongoing framework for programmed production of 3D confront models from a video arrangement". The framework comprises of a novel plan to extricate confront shape and a 3D show adjustment technique in light of a minimum square approach. The proposed confront shape extractor, which is exact and computationally shabby, depends on an oval controlled by three grapple focuses. The minimum square approach is chosen to adjust a non-specific 3D model to include confront, limiting mistakes between highlight focuses on a made

3D confront and separated component focuses from input video. Test comes about demonstrate that the proposed framework can productively manufactures 3D confront models, which are good with MPEG-4 facial items, from a video arrangement with no client mediation [6].
Another method of processing 3D model can be done in CAD/CAM system which is capable of modifying, archiving as well. The challenging part is the transformation procedure where data need to acquire from paper drawings. In this case, the different types of drawing and forms of displaying them and encountered errors and deviation from technical standards increases its complexity. Thus, an algorithm was proposed by J.Vasky, M.Elias, P.Bezak, Z.Cervenanska, L.Izakovic et al. to build 3D models from an orthogonal vector input. [4].
Again, 3D imaging methods, generation of 3D models and illustration of 3D points in cloud at different positions along with their applications and implementations are by discussed by the authors in [7]. A detailed assessment has been given by the authors about Kinect for Windows V2 Sensor for the purpose of reconstructing 3D models in [8]. In [9], the authors provided a comprehensive record of the variants (that are different from each other) of closest point algorithm which could be useful for the arrangement of various 3D models for the same object into a single position. In [10]. The study of Soumaya Bachtobji, Aymen Omri, Ridha Bouallegue, Kosai Raoof introduces a three-dimensional (3D) model for *K*-tier heterogeneous networks (HetNets) in order to evaluate the performance of using millimetre wave (mmWave) and radio-frequency (RF) bands. The model is based on 3D stochastic geometry that describes the different cells' positions in the network with realistic constraints.

A large portion of 3D demonstrating devices' working interface are intricate. In [11], a quick free outlining 3D demonstrating strategy with edge understanding drew nearer. Free outlining is this present technique's information. A 3D demonstrate was built in a brief span. Some self-convergence chart can be perceived however include some edge record data in this strategy.

3D remaking in view of vision implies getting picture information of the question and making a three-dimensional model in view of the learning of PC vision. This innovation isn't liable to the state of the protest. It is basic, helpful, and can create great three-dimensional models, with the goal that it is generally utilized as a part of numerous zones, for example, therapeutic application, robot route, building reproduction, and so forth. With the coming of some minimal effort RGB-D

cameras (e.g. Kinect), obtaining point mists containing shading data is substantially less demanding and quicker. In their paper, BoweiShen, Fang Yin &Wusheng Chou set forward a typical technique to manufacture 3D model of family unit question for automated getting a handle on. The point mists including object from various perspectives are caught from the Microsoft's Kinect v2 sensor. A pixel separating approach is utilized to process profundity picture and morphology calculation is executed to channel commotion focuses in the point billows of question. FPFH descriptor is extricated from each point. Test Consensus Initial Alignment and ICP calculation is utilized to enroll two adjoining point cloud precisely. In view of a shut circle enhancement strategy, the total blunder from persistent enlistment is decreased. We manufacture approximately 3D models of indoor protests through the proposed approach. The investigation comes about demonstrates that the strategy is advantageous and can meet the exactness necessities [12].

The three-dimensional display innovation isn't just the key of the virtual reality yet additionally the cellar of Virtual Reality (VR) framework, and the development of profundity observation which is accomplished by binocular divergence gave a huge advantage to 3D show. In the present market, the 3D impact is specifically dictated by utilizing twofold perspective strategy which contains the separation data of the scene. In view of stereo vision and the utilization of OpenGL, this paper is to separate multi-see pictures from the virtual 3D show, at that point change it into a stereoscopic uniqueness guide to take care of the 3D show issues. The perception framework contains the 3D display perusing and the production of binocular difference outline. The past part presents the perusing procedure of vertex data and the illustration of vertices. The follow-up is made by the monocular change calculation and the illustration of twofold perspective guide.[13].

Last but not the least, two other approaches for creation of 3D model is discussed briefly in [14-15].

Chapter 03

Proposed Method

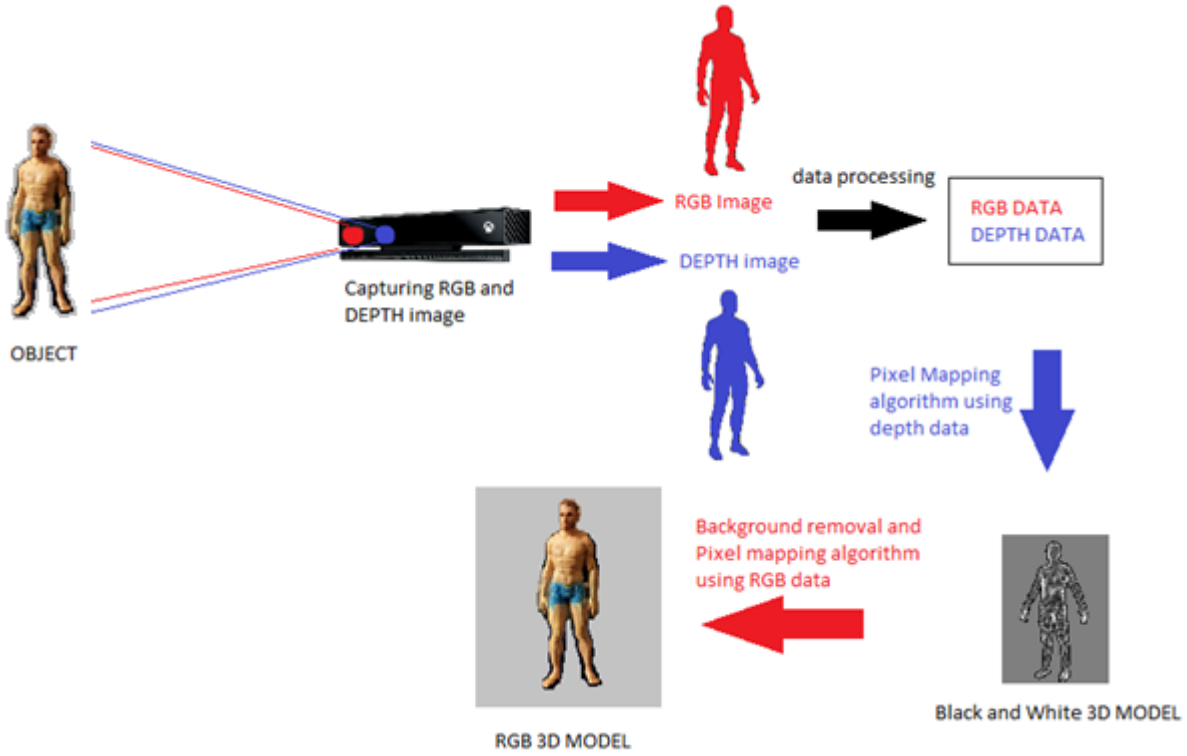

Fig. 3.1 System Architecture.

Our proposed method uses depth and RGB images from Kinect sensor to create the colored 3D model. The method emphasizeson our created algorithm which we call pixel mapping algorithm. The algorithm is based on 1D arrays of mainly 4 parameters which is acquired from the 2D array data's .These 2D array data's are processed from the images taken by Kinect. The 4 parameters we use in the process are X, Y, Z and RGB.So to get the model our proposed method consists of 3 phases which are acquisition of RGB and depth information of real object, extraction of RGB and depth data and generation of 3D model using proposed pixel mapping algorithm which are shown in figure 3.1.

Step 1: Acquisition of RGB and depth information of real object:

Fig. 3.2 RGB image.

Fig. 3.3 Depth image.

For getting RGB and Depth information, we take images using our Kinect sensor with attaching to Matlab. First of all, in the image acquisition section of Matlab, there are two options. One is for taking RGB image and other one for depth image. RGB section has to be selected first and then the preview option. After that, for capturing image, start acquisition must be pressed then Kinect will take a RGB picture in 1920*1080 resolution. For exporting the image, export data button has to be pressed andthe image must be stored in windows bitmap format. By following same process depth Image needs to be captured and stored in TIF format. Here the depth image is in 640*480 resolution. For processing data in the next step we need to resize the RGB image to 640*480 resolution as our pixel mapping algorithm requires both image data (RGB and depth) to be same resolution.

Step 2: Extraction of RGB and depth data:

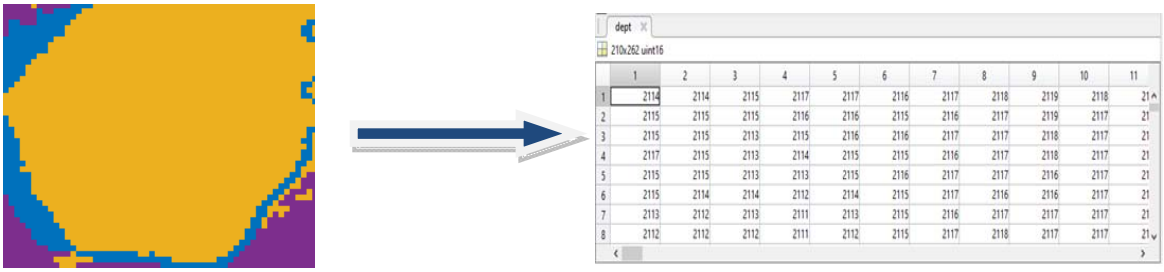

Fig. 3.4 Data extraction

Image acquisition gives us the RGB and Depth images, we import those images in our matlab for extracting the RGB and depth data. For extracting the data, we need to import the images in Matlab. RGB image need to be stored as Unit8 format which is a 2D array containing each pixel's red, green and blue values in this arrangement - [R G B]. We use a built in function to make 3 separate 2D arrays for RED, GREEN and BLUE from the previous array. We take depth image as Unit16 format which is a 2D array containing distance values from Kinect, pixel wise. So, from here we get four 2D arrays which needs to be used in our proposed algorithm in the next step.

Step 3: Generating of 3D model using the proposed pixel mapping algorithm

To create the model proposed algorithm needs to be used and the inputs need to be arranged in a proper way. To demonstrate the whole system our proposed a method refers to dividing the whole process into two segments. First the depth input should be arranged. Depth information's of the Kinect sensor are in 2D form. So in the first flowchart 3.3 Height and width of the array

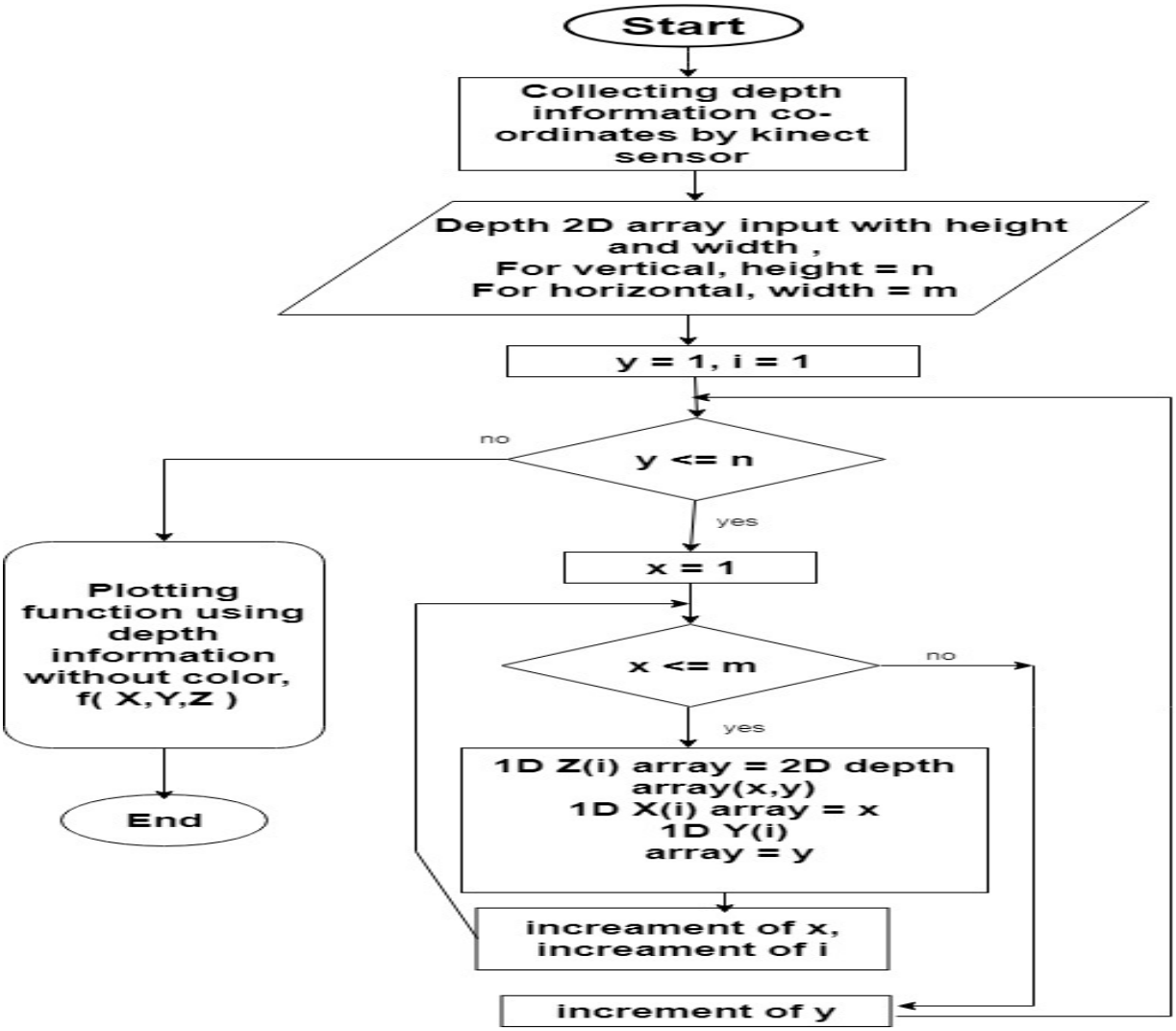

Fig. 3.5 Flow chart of the algorithm for the 3D model generation without color.

can be replaced with m and n temporary variable. In our flowchart in fig. 3.5 vertical and horizontal array places are filled with depth values. So for the 1st value that should be plotted, it is in the horizontal and vertical 2D matrices first index of the array list shown as Table. 3.1. Another 1D array should be created to transfer the 2D metric's depth values in a way so that when the first horizons last value is transferred, 2D metric's next horizons first value should be moved in the 1D array's next index shown as Table 3.2.

Width ⟶

DepthOf(1,1)	DepthOf(2,1)	DepthOf(3,1)	DepthOf(4,1)	DepthOf(5,1)
DepthOf(1,2)	DepthOf(2,2)	DepthOf(3,2)	DepthOf(4,2)	DepthOf(5,2)
DepthOf(1,3)	DepthOf(2,3)	DepthOf(3,3)	DepthOf(4,3)	DepthOf(5,3)

↑
Height

Table. 3.1 Two dimensional array of depth information.

DepthOf(1,4)	DepthOf(2,4)	DepthOf(3,4)	DepthOf(4,4)	DepthOf(5,4)

Table. 3.2 One dimensional array of depth information that has been constructed.

DepthOf(1,1)	……	DepthOf(5,1)	DepthOf(1,2)	…….	DepthOf(4,4)	DepthOf(5,4)

This way all the depth values (Z axis) are transferred in the 1D array. For plotting 3D model, there should be other two dimensions too (X axis and Y axis). For X axis's values and Y axis's values, other two 1D arrays are to be constructed. These arrays system are simpler than z-axis's data table construction. For both the x-axis's one dimensional array and y-axis's two dimensional array, for the 1st value that should be placed, it is in the horizontal and vertical 2D matrices first index number of the array list. Another 1D array should be created to transfer the 2D metric's depth values in a way so that when the first horizons last value is transferred, 2D metric's next horizons first index number should be moved in the 1D array's next index . Than the plot functions with X, Y, Z [1D arrays] values will plot a colorless 3dimentional model (it will plot height*width times, as each time it creates one pixel per X, Y, Z co-ordinate plane. So creating height*width times will draw the whole pixels in one X, Y, Z co-ordinate plane).

The second stage demonstrates for adding the color. After doing all instructions described in stage one flowchart shown as Fig. 3.5.Flowchart shown in Fig.3.6 should be followed.

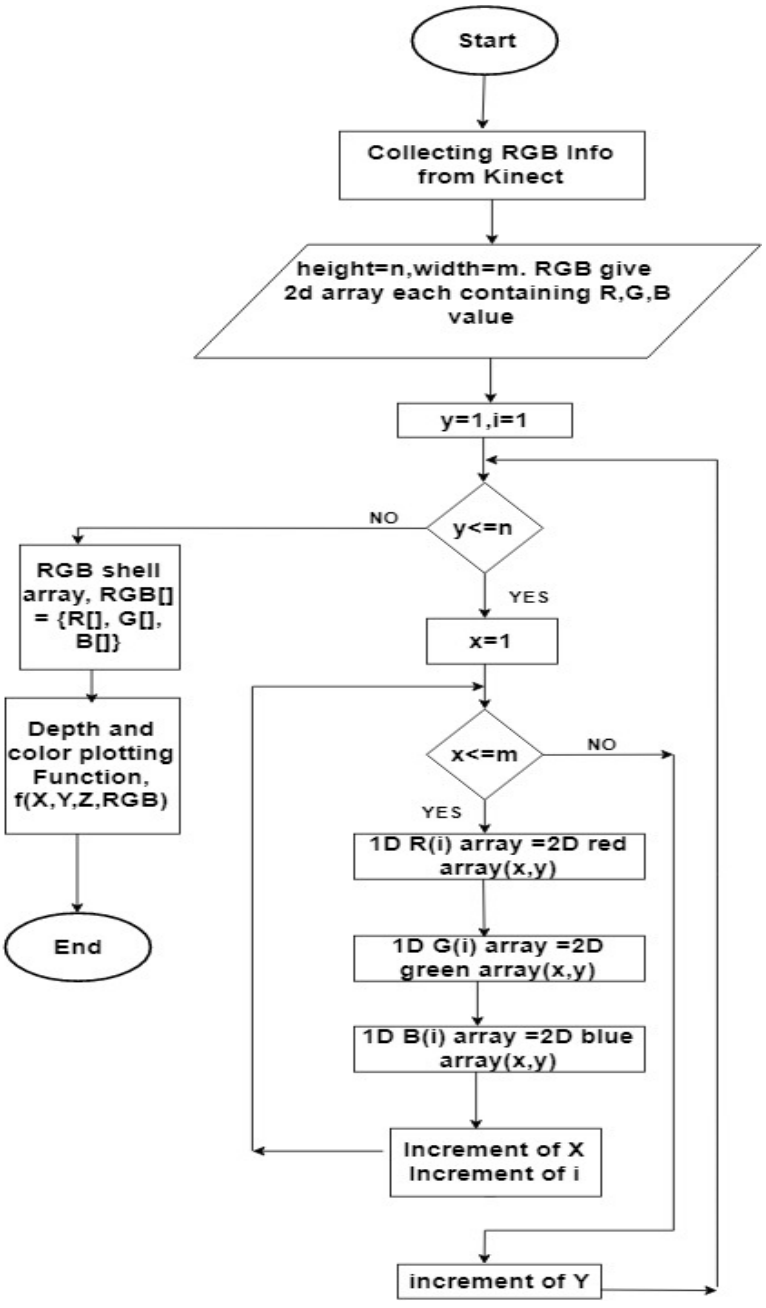

Fig. 3.6 Flow chart of the algorithm for adding RGB color.

To implement this method, introducing with shell array idea is needed. Shell array can hold shell values in one index of the array. One shell can hold multiple values in any sequence (for example, shell Array [1] = {10, 40, 20}).

Table. 3.3 Two dimensional array of color information (red and green and blue values).

Width ⟶

Height ↑

R&G&B(1,1)	R&G&B(2,1)	R&G&B(3,1)	R&G&B(4,1)	R&G&B(5,1)
R&G&B(1,2)	R&G&B(2,2)	R&G&B(3,2)	R&G&B(4,2)	R&G&B(5,2)
R&G&B(1,3)	R&G&B(2,3)	R&G&B(3,3)	R&G&B(4,3)	R&G&B(5,3)
R&G&B(1,4)	R&G&B(2,4)	R&G&B(3,4)	R&G&B(4,4)	R&G&B(5,4)

First we extract the red color values in a two dimensional array and then accordingly other colors (green and blue color values) are also to be moved to separate two dimensional arrays.

Table. 3.4 Two dimensional array of color information (only for red values).

Width ⟶

Height ↑

Red(1,1)	Red (2,1)	Red (3,1)	Red (4,1)	Red (5,1)
Red (1,2)	Red (2,2)	Red(3,2)	Red (4,2)	Red (5,2)
Red (1,3)	Red (2,3)	Red (3,3)	Red (4,3)	Red (5,3)
Red (1,4)	Red (2,4)	Red (3,4)	R&G&B(4,4)	Red (5,4)

From the extracted red colored data in different 2D array, along with other colored data in their separate allocated 2D array, total 3 sets of 2D array is constructed. Now, the process of one dimensional array construction can be proceeding (without separating the RGB values individually, our algorithm could not be implemented. For our algorithm 1D array is an essential concentration.). Table. 3.4 shows the 1D array where the 6^{th} index contains Red (5, 1) which is the 2^{nd} rows 1^{st} index's content of 2D array.

Table. 3.5 One dimensional array of color information (only for red values)

Red(1,1)	……	Red (5,1)	Red (1,2)	…….	Red (4,4)	Red f(5,4)

Finally from the 3 sets of 1D array each containing red, green, blue color's values of every pixel, our algorithm now focuses on making a shell array for adding color to the pre-described flowchart Fig. 3.5, sequentially shown in Fig. 3.6. The first index of the shell array holds 3 sets of values of the (1, 1, 1) co-ordinate and carries constructing the shell array till n (=height*width) times accordingly.

Table. 3.6 One dimensional array of color information (for red, green, blue values).

ShellArray({r(1),g(1),b(1) }	……	ShellArray({r(m),g(m),b(m) }	ShellArray({r(m+1),g(m+1),b(m+1) }	…….	ShellArray({r(n-1),g(n-1),b(n-1) }	ShellArray({r(n),g(n),b(n)) }

Now, the shell array provides with all the essential information to add color to the plotted model. Basically these 2 stages of plotting the 3D model methods are parallel process, but for better understanding of the 2 types of data handling systems to implement our algorithm, these are explained in two stages.

Chapter 04

Experimental setup and Result

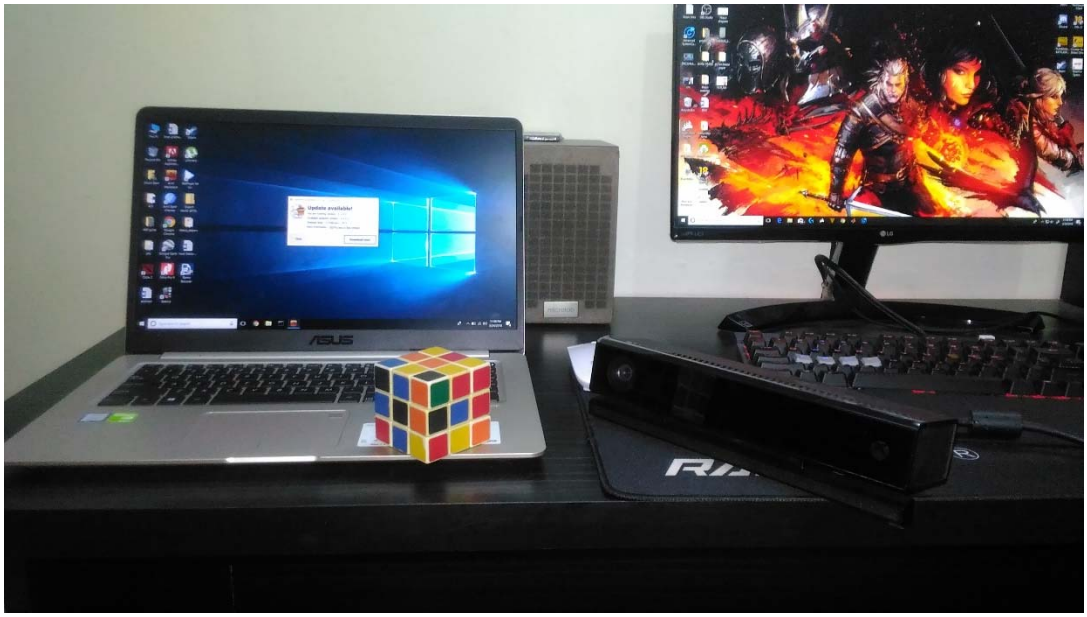

Fig. 4.1 PC, Kinect Xbox one and object.

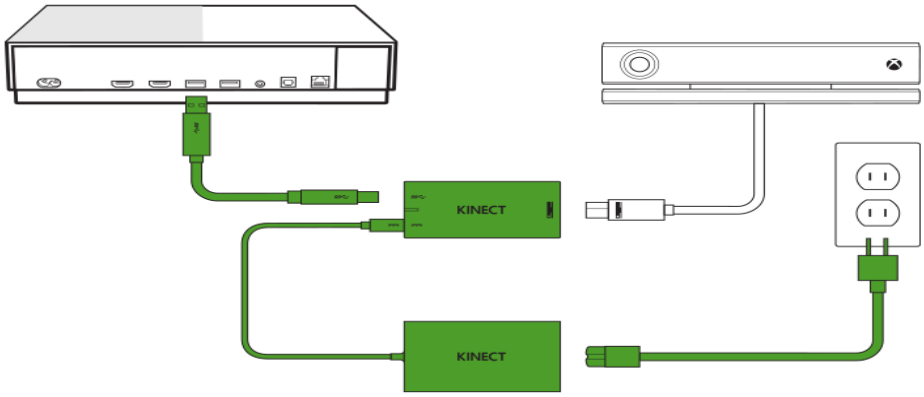

Fig. 4.2 Kinect sensor set up.

Kinect sensor must be purchased as a way so that both an Xbox Kinect sensor and the Kinect Adapter for Windows is surely collected. Xbox 360 Kinect has a lower resolution so we used Kinect Xbox one here for a higher resolution value of 680*480 when using depth camera.

We need to have a fixed computer where we saved our codes in MATLabR2016a. The Pc requirements for installing the MATLab16a and setting up the Kinect sensor are

> Windows 8 (64-bit) or Windows 8.1 (64-bit),Windows Embedded Standard 8 (64-bit), Windows Embedded Standard 8.1 (64-bit), 64-bit (x64) processor, Dual-core 3.2 GHz or faster processor, Dedicated USB 3.0 bus, 2 GB RAM, A Kinect for Windows v2 sensor

By putting our proposed method to use we were able to get our expected result. Details of our experiments are described below.

4.1 3D model Generation (Including background without color):

After getting all our data's by acquisition and processing we used the 1D full arrays X, Y and Z which have height*width number of values in each one, to create a 3D plot. X, Y, Z arrays have values in it row wise so the position -i value of each array maps a particular point on the 3D plot according to the real depth image. By plotting all the points we get a 3D representation of the whole depth image shown in figure fig 4.4.As the depth image is not clearly understandable as an image the RGB image is shown in fig 4.3 to clearly visualize the 3D simulation in black and white.

Fig. 4.3 Actual image taken by Kinect sensor.

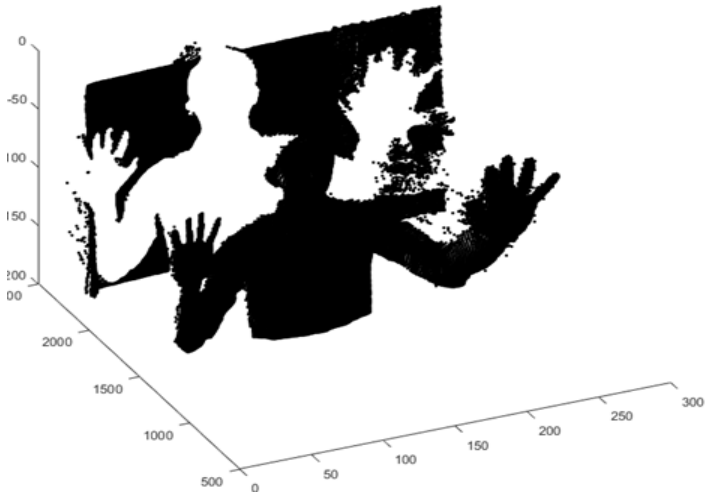

Fig.4.4 Depth image taken by Kinect sensor (left-front view).

4.2 3D model Generation (Object without color):

The model can be rotated at any angle to better visualize the 3D plot. We have shown here the visualization from a particular angle .From the fig 4.4 we can see there are background plotted along with the object (here the human body). To remove the background we edited the algorithm to filter out the points which has Z values more than the objects last maximum Z values and plotted the object only in 3D plane again shown in fig 4.5.

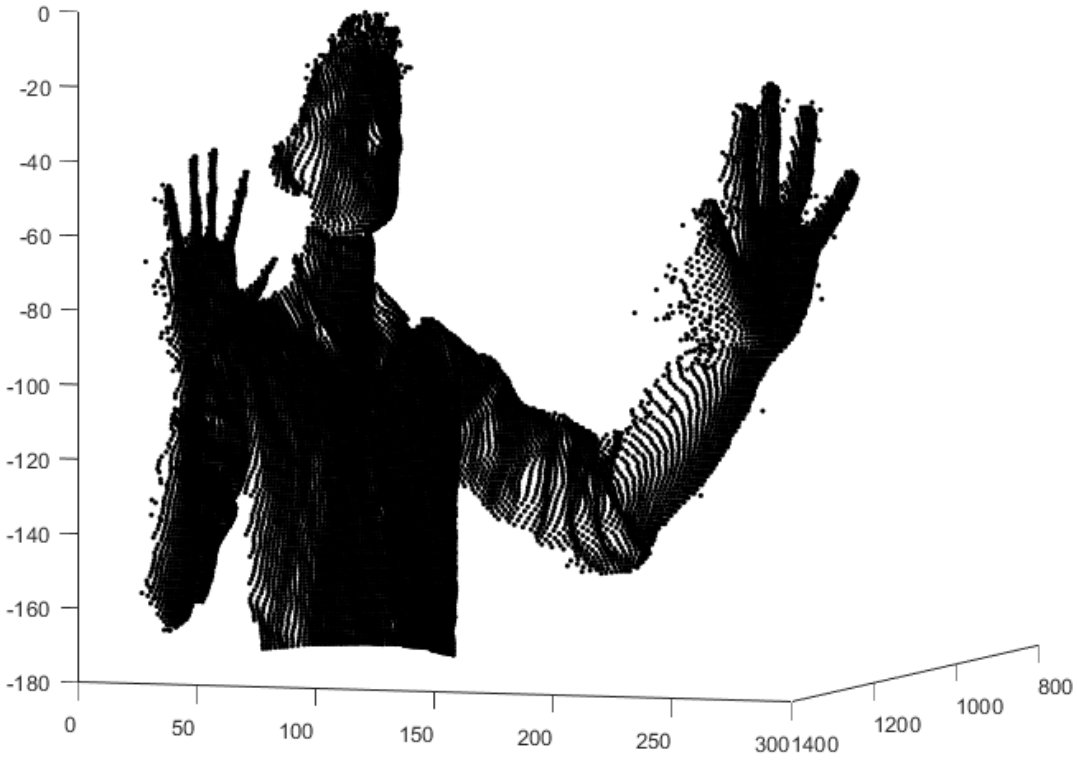

Fig. 4.5 Depth image of actual object (after image processing, front-right view).

4.3 3D model Generation (Object with color):

From the fig 4.5 we get the 3D model for the object .Now to add color we had to use our self-made pixel mapping algorithm which gives each point the exact color from the RGB image. We took the RED, GREEN and BLUE arrays similarly as the X, Y and Z arrays row by row to perfectly match each point's red, green and blue values. As we needed a full array with the red, green and blue values we had to use a shell array of named RGB which contains merged values of red, green, blue row wise in 1D array. Using this 1D array along with the previous plot we created the 3D model of the object with exact color in each point. The model can also be rotated for better visualization. We have shown the model From 2 particular angles in fig 4.6 and fig 4.7.

Fig. 4.6 Object's image after applying algorithm (front-right view).

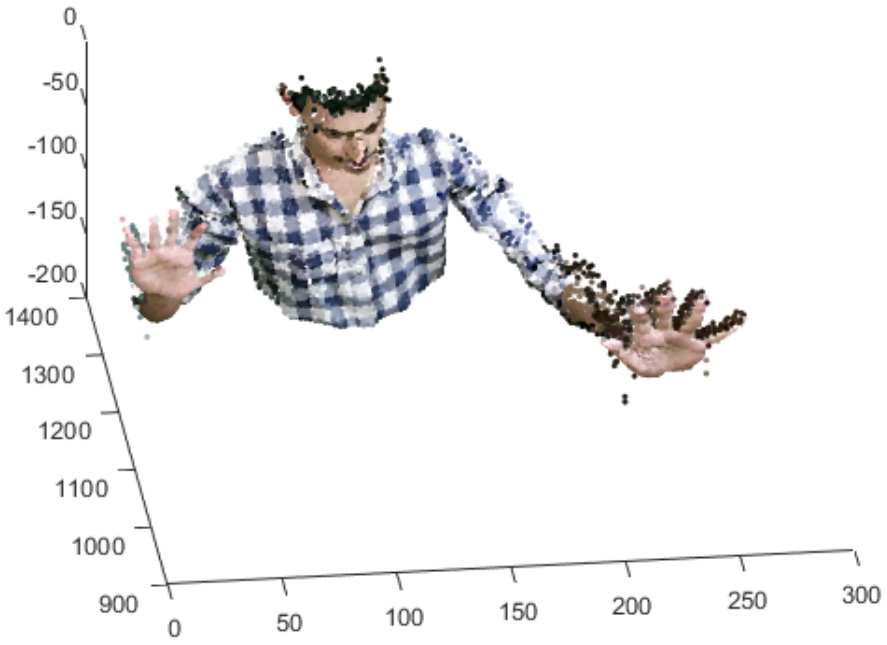

Fig.4.7 Object's image after applying algorithm (Top view).

Experiment using Rubik's cube:

Fig. 4.8(a) Actual image taken by Kinect sensor (rubric's cube).

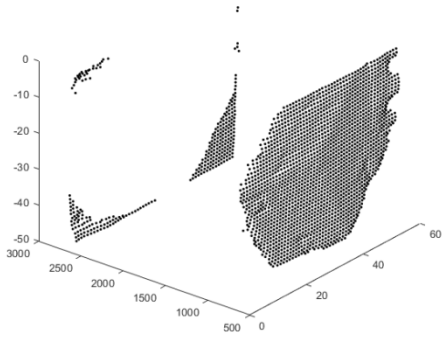

Fig. 4.8(b) Depth image taken by Kinect sensor (Object has been shown separately from background).

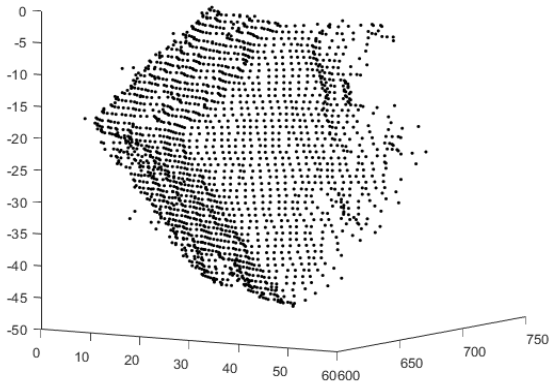

Fig. 4.8(c) Image of actual Object (after image processing, front-right view).

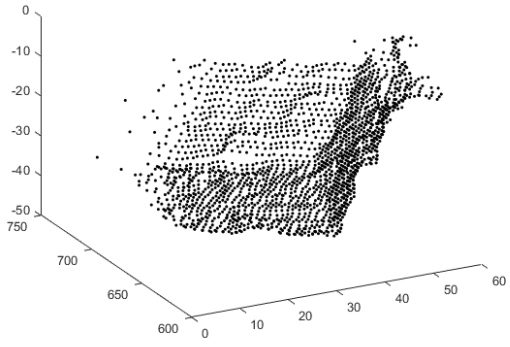

Fig. 4.8(d) Object's image before adding color (Top view).

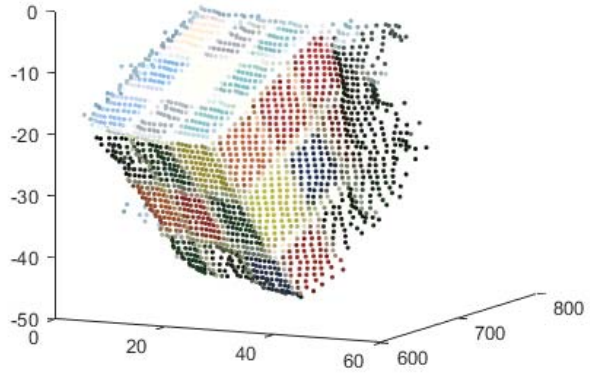

Fig. 4.8 (e) Image of actualObject (after adding color, front-right view).

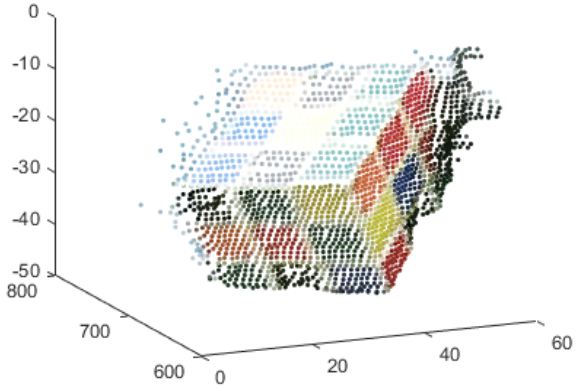

Fig. 4.8(f) Object's image after adding color (Top view).

Experiment using human figure:

Fig. 4.9(a) Actual image taken by Kinect sensor.

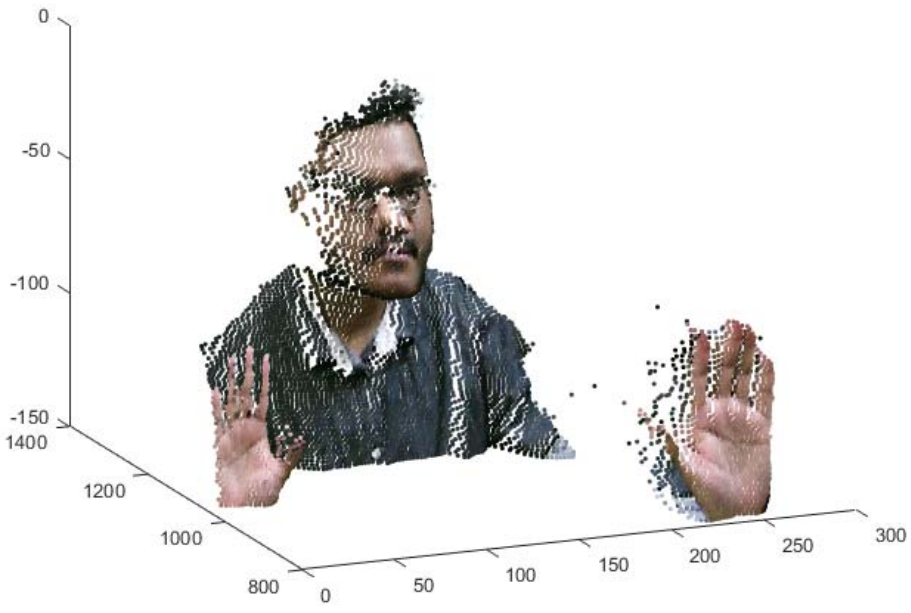

Fig. 4.9(b) Object's image after applying algorithm (left-front view).

Experiment using toy car:

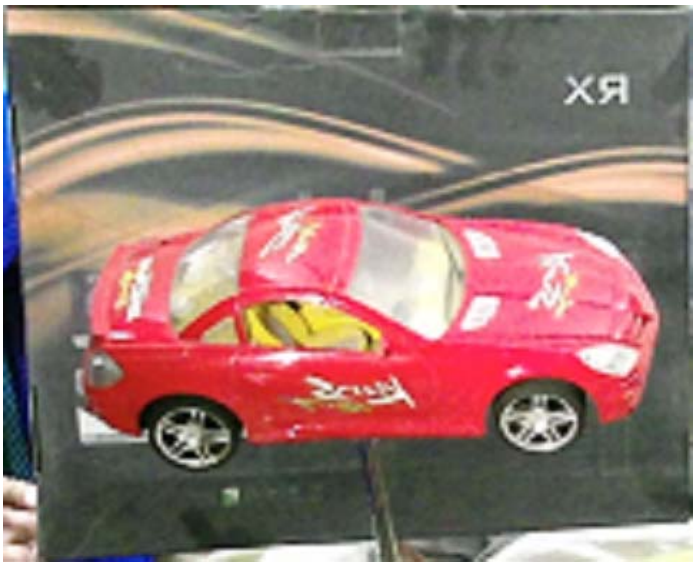

Fig. 4.10(a) Actual image taken by Kinect sensor.

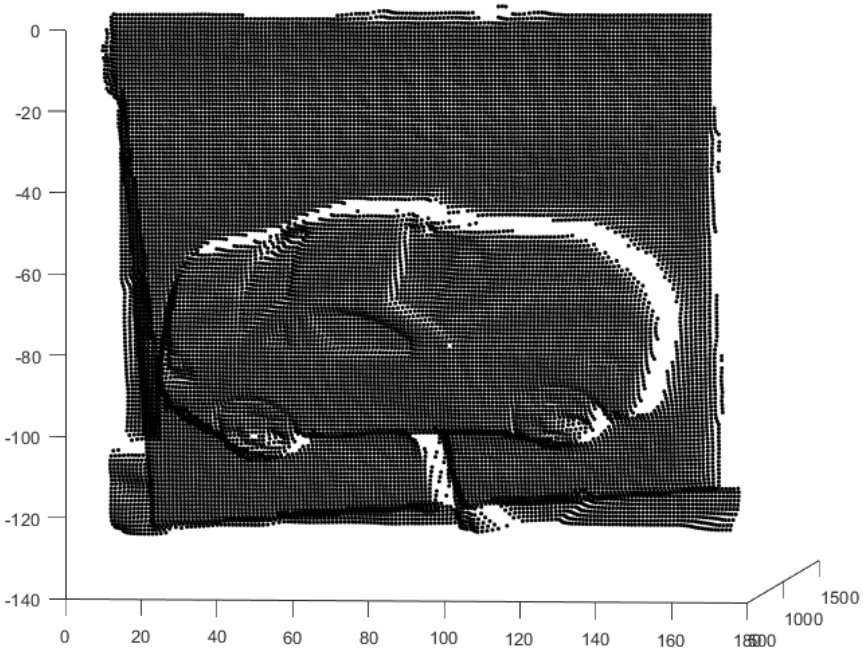

Fig. 4.10(b) Depth image taken by Kinect sensor.

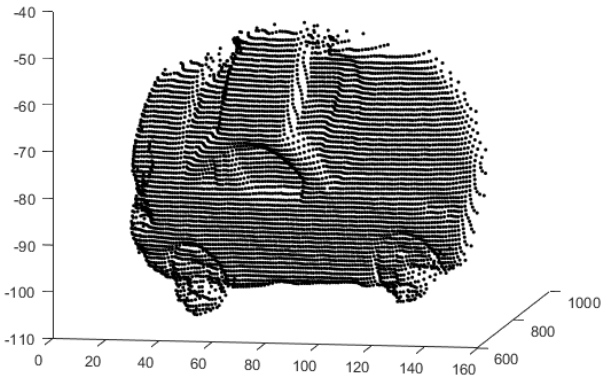

Fig. 4.10(c) Object's image before adding Color (left-front view).

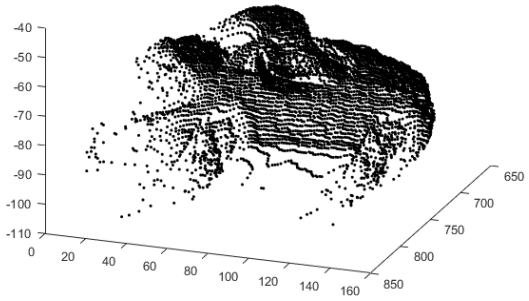

Fig. 4.10(d) Object's image before adding color (left-front view).

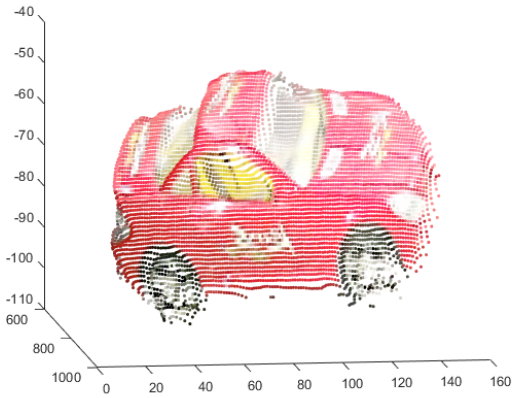

Fig. 4.10(e) Object's image after adding Color (left-front view).

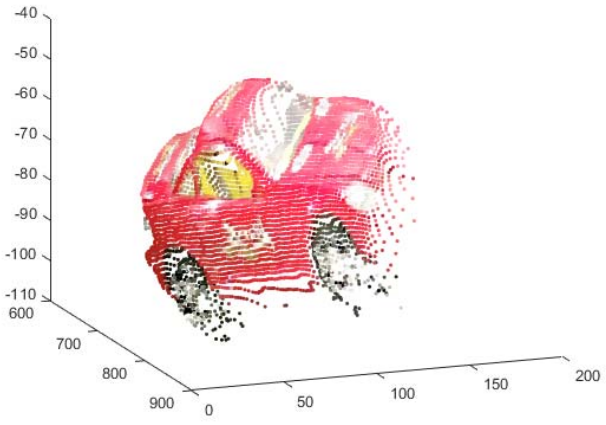

Fig. 4.10(f) Object's image after adding color (left-front view).

Accuracy rate

The comparison of a measurement with a known standard, used to determine whether the measurement is reliable. Estimation precision is recognized as the distinction between the estimation of a factor and the acknowledged an incentive for that factor from a trusted outside source, or the rate by which the two esteems vary. We tried to build a method, using some functional values. These values are calculated based on some functions that should be excluded from a standard acknowledged value that is exactly hundred percent or we can also say a whole percent.

So, our proposed method of measuring the accuracy rate depends on some functions. We have found out that these functions should be called Depth and RGB pixel difference, plotted pixel structure, blank spots, software incompetence, Motion limitations, Kinect sensor XBOX one camera placement etc. The first three functions are theoretically representational. Other functions contribute on accuracy lacking of the model but not predominantly like the first three functions. So these functions are,

Depth and RGB pixel difference:

The pixel values of original source depth picture and RGB picture are not the same. It's a major obstacle that is in between creating a perfect model. The depth picture's resolution is always less than the EGB valued pictured resolution. This problem occurs and we actually have to lessen the RGB valued pictures resolution to solve this problem which creates a pixel gap which has its effect on our pixel plotting algorithm. For better explanation our 3.7.1(a) figure's depth picture has a resolution of 56*47 and its RGB picture's resolution was 78*64. Here's their difference is 78-56 * 64-47 = 22 * 17 which is 38% less of the actual RGB value. To solve this problem we

made both the pixel values equal meaning we lessen the RGB values till depth values. So this way we have come up with a functional representation,

$$DRPD = \left(\sum_{i=1}^{n} ((Xi-Ai)/Xi)*100 + \sum_{i=1}^{n} ((Yi-Bi)/Yi)*100\right)/n \quad ....(1)$$

Here,

X, Y is the pixel values of depth picture and A, B is the RGB pixel values.

n is the total number of examples (for our thesis example case it was 5)

I is the sequence number.

DRPD will be a direct percentage value that contributes in decreasing the accuracy rate. For our case it was approximately 39%.

Plotted pixel structures:

We have implemented our algorithm in a pixel system where pixels are represented as circles, if it could be square pixel system our accuracy would be more close to perfection. This is a much easier calculation where we can clearly find out the main prob. Square pixels covers every way if a lot of pixels gets attached together, but for circular pixels there are tiny little areas that it cannot completely cover. So PPS (Plotted pixel structure) should be subtracted too.

$$PPS = \left(\left(\sum_{i=1}^{n} (Xi*Yi)\right)/n\right) (*(4 - \pi) * r^2) \quad .....(2)$$

Here, X, Y is the pixel values of depth picture.

r is the pixels radius, we and π stands for Pi value that is 3.1416

Blank spots:

From this step to further discussion of accuracy rate measuring, we didn't go to depth of the measurement, but these factors still affects our rate. In the 3D model there are some blank spots that Kinect sensor's camera couldn't catch. Almost every picture has these type of blank spots that gets covered by objects closer pixels, so depth camera cannot catch it and it stays there as blank.

Software incompetence:

We used matlab2016a software that is the updated version of matlab but there are others software's like visual studio that supports a lot of updated feature for plotting 3D graphical model. These updated technologies help rotating the model and view it more accurately. For matlab we could move our model flexibly, it could barely hold and was taking a very long time to execute one command. Moreover it doesn't support pictures with very large resolution.

Motion:

While taking the input depth and RGB pictures an object must stay still during the process, otherwise Kinect sensor cannot correctly catch a moving object. In one of our example, that is Fig 3.9(a), the object who was one of our member, was moving so the model could not be as perfect as the other models.

Kinect sensor XBOX one camera placement:

The two cameras (one is the RGB camera and the other is the depth camera) have a small portion of difference in between their centers. The accurate measurement of their difference is approximately (approximately because the camera cannot be seen from outside and Microsoft did not stated the accurate difference in between their centers) 5.4 mm. So if we fix our object in one place and take both the RGB and Depth pictures without moving the Kinect sensor, depth camera takes 5.4mm sided picture. So while merging the two of them in our matlab software plotting system with implementing our algorithm, it gets displaced. To solve this problem we have to exactly move the Kinect sensor 5.4mm that we cannot do perfectly.

So, these were the factors that contributes in lessening the accuracy rate and for our case we measured most of it from the first factor DRPD which was 39%. So our accuracy rate should be close to 1-.39 = .61 which means 61%. If we could eliminate this problems these accuracy rate would go high.

Conclusion

In this paper, we proposed a method which can generate 3D models using depth and color information of real objects. We were successful in developing 3D model from a 2D image. By using image acquisition we captured RGB and depth images and stored them. Furthermore, we create several 1D arrays and use then in our proposed pixel mapping algorithm and obtain our 3D model with background. After that, we remove the background and add RGB values in our model and obtain a 3D model of captured object. However, after achieving this there were still a few limitations, our proposed system can't make 3D in real time, it takes few moment for creating 3D model. In our proposed system we used Microsoft Kinect sensor, it switches between 15 and 30 FPS depending on the lighting conditions. For that hardware limitation our system can't generate 3D model in real time. Besides, there are many scopes for improving our method. Our proposed system can create a 3D model taking images just from one angle .As this can be achieved only by an angel the images taken only once for depth and RGB without having to move the sensors position which will greatly help making 3D model with more accuracy. Also, we can see now a day's console gamers are expanding step by step and they mostly have these Kinect sensors for playing. This thesis will inspire them to utilize those Kinect to some use and join the era of demonstrating 3D model easily and do further research on 3D models which can lead them to a considerably brighter future. . This thesis will also help in the field of animation and model designing as the methods of creating animation model now are extremely high costing. Moreover, by using our work we will try to develop it further by capturing and creating models from more angles makes and will also try to make whole 3D scenario much easier and cheaper with less effort.

Reference

[1] Mc.Danie, T.(1981).A technique for resolution amplification in three-dimensional field calculations for recording heads.*IEEE Transactions on Magnetics.* Vol. 17(6).

[2] Luan, X., Xie, Y., Ying, L., & Wu, L.(2008). Research and Development of 3D Modeling, 8(1), 49.

[3] U.Castellani, A.Fusiello, V.Murino, L.Papaleo, E.Puppo, M.Pittore. (2005). A complete system for on-line 3D modelling from acoustic images. Signal Processing: Image Communication, 20:832-852

[4] J.Vasky, Elias, M., Bezak, P., et al.(2011). 3D Model Generation from the Engineering Drawing. *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, 18(29), pp. 47-53. Doi: 10.2478/v10186-010-0025-z

[5] Zhu, Z., Yan, C., Li, L., Ren, Y., Luo, Q., & Li, J.(2017). Stereoscopic Visualization of 3D Model Using OpenGL. doi:10.1109/GlobalSIP.2017.8309183

[6] Choi, K. H.., Hwang, J. N. A real-time system for automatic creation of 3D face models from a video sequence.(2011, April 7). *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference*

[7] N.Pears, Y.Liu, and P.Bunting.(2014). 3D Imaging, Analysis and Applications, Springer London,

[8] E.Lachat, H.Macher, M.A.Mittet, T.Landes, and P.Grussen Meyer.(Feb.2015). "First Experiences with Kinect v2 Sensor for Close Range 3D modelling," ISPRS- International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp.93-100.

[9]S.Rusinkiewicz and M.Levoy.(June 2001). "Efficient variants of the ICP algorithm," in Third International Conference on 3D Digital Imaging and Modeling(3DIM).

[10] S.Babtobji, A.Omri, R.Bouallegue,K.Raoof," Modelling and Performance Analysis of MmWaves and Radio-frequency Based 3D Heterogeneous Networks", 12(3), 290-296. (2018). doi:10.1049/iet-com.2017.0777

[11]Li, Y., Li, H., & Lu, H. (2008).Fast 3D Model Construct Based on Free Sketching. doi:10.1109/CCPR.2008.29

[12] Shen, B.,Yin, F.,Chou, W. (2017). “A 3D Modeling Method of Indoor Objects Using Kinect Sensor “.*Computational Intelligence and Design (ISCID), 2017 10th International Symposium on*

[13] Sabale, A. S. &Vaidya, Y. M. *(*2016). “Accuracy measurement of depth using Kinect sensor”.*Advances in Signal Processing (CASP) Conference.*

[14] Aoki, R., Aoki, S., Ohtagaki, Y., & Miyamoto, R. (2017). Key Point Localization for 3d Model Generation from Facial Illustrations Using SURF and Color Features. doi:10.1109/ICCE-Berlin.2017.8210589

[15] Yaguchi, H., Takaoka, Y., Yamamoto, T., & Inaba, M. (2013). A Method of 3D Model Generation of Indoor Environment with Manhattan World Assumption Using 3D Camera. doi:10.1109/SII.2013.6776686