

3D Scanner for Packaging Solutions

Ameya Konkar¹, Omkar Ghuge²

¹Department of Electronics Engineering, KJ Somaiya College Of Engineering

²Department of Electronics Engineering, KJ Somaiya College Of Engineering

Abstract - This project describes the use of depth sensing technology and file format conversion to create computer aided designs (CAD) of packaging foam cavities. This product can be used by packaging industries to automate the process of foam cutting for use in transportation of delicate products. For depth sensing, the Xbox One Kinect depth sensor by Microsoft is used. The Kinect sensor is a Stereo camera and thus it returns RGB image as well as depth data of the RGB image pixels. The objects to be transported are scanned using the Kinect sensor. The scanned data is converted into CAD model of the foam having cavity of the dimensions of the objects. The proposed idea aims at reducing time for creating such CAD models as well as increasing its dimensional precision.

Key Words: FoV (Field of View), ToF(Time of Flight), CAD(computer aided designs), DXF, Potrace (Polygon Tracing), CCA(Connected Component Algorithm), GUI (Graphical User Interface).

1. INTRODUCTION

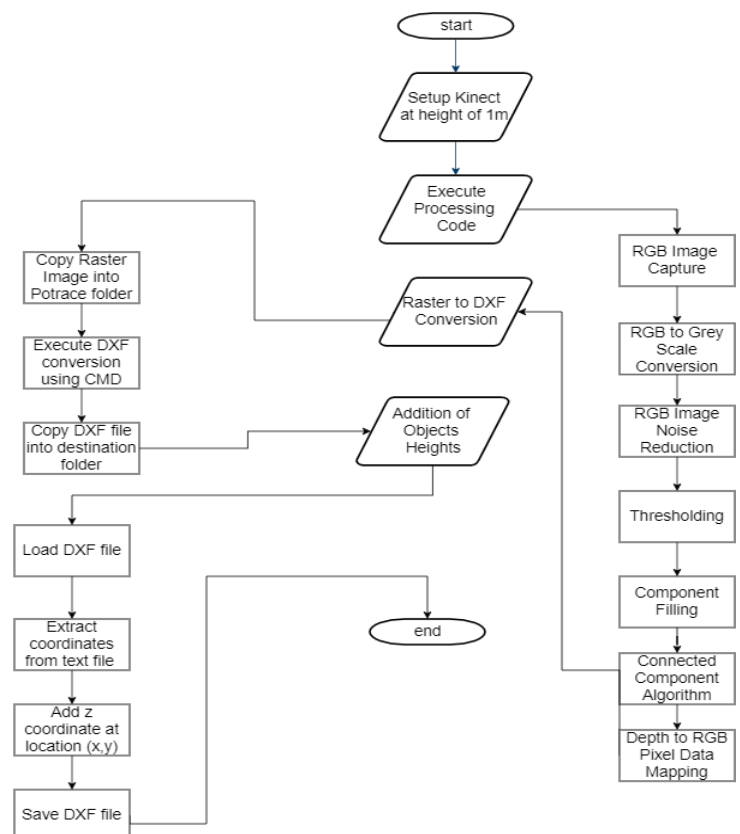
In Packaging Industry, it is required to create foam packaging of products. Cavities are created in the foam of the dimensions of the object required to be packaged. Nowadays, the process of measuring dimensions of objects is done by manually taking measurements. Using these measurements, a 3D file is generated of the foam containing cavities for the objects in software such as Solidworks. This file is then used by the CNC machines to create foams having cavity of the measured dimensions of the objects. Main drawback of this process is that it consumes much time and the time taken is proportional to the number of objects. Another problem is that the process can cause manual errors while taking measurements. In order to overcome these difficulties, 3D Scanning is the process used to find the dimensions of the objects. In our project, we have used Xbox One Kinect camera which is a ToF camera for acquiring 3D coordinates of the objects. The time required for scanning is very less as compared to measuring the dimensions manually and it remains constant irrespective of the number of objects scanned. The accuracy of the measurements has an error precision of ± 1 mm which makes it more reliable than manual measurement.

2. SOFTWARE AND SETUP

Processing is an open-source graphical library and coordinated improvement climate worked for the electronic expressions, to instruct non-developers the essentials of computer programming in a visual setting. It is used for Image Processing in this project. It is a Java based platform which has dependencies and libraries required to access RGB and Depth data from the Xbox One Kinect and to send commands in order to capture or save image in different formats.

The setup includes a Xbox One Kinect and a transparent table having white light projecting upwards through the transparent portion of the table. The Kinect sensor will be mounted vertically above the table at a distance of 1m. Objects to be scanned will be placed on the table with the white light projecting beneath it.

3. FLOW DIAGRAM



4. METHODOLOGY

4.1. Image Processing

The procedure includes processing an RGB image into an optimal binary image required in the formation of (.DXF) file. Processing an image divided into three parts:

1. Pre-processing.
2. Processing.
3. Post-processing.

4.1.1. Pre-processing.

The pre-processing is performed in order to remove noise and distortions present in the image and to reduce the complexity of the image without losing any vital data.



Fig -1: RGB image

- **RGB to Greyscale conversion:** After capturing an RGB image, it is converted into Grayscale image by averaging the values of RGB channels.

This enables to reduce the color parameters thus reducing the complexity of Image Processing Algorithms.



Fig -2: Grayscale image

- **Median Filter:** Noise reduction is required in order to remove speckles, salt and pepper noise and gaussian noise from the grey scaled image. This was accomplished by using Median filter Algorithm. The **median filter** is a non-linear digital filtering technique. Median filtering is used because it preserves edges while removing noise. The concept of the median filter is to run through an image pixel by pixel, replacing each pixel with the median of neighboring pixel values. The pattern of neighbours is called the "window", which slides, pixel by pixel, over the entire image.

4.1.2. Processing.

In the processing stage, the pre-processed image is converted into a binary image by thresholding it using an optimal threshold value.

- **Otsu Thresholding:** The grey scaled image is converted into binary image by segregating the pixels of the gray scale image into foreground (white pixels) and background (black pixels). The Foreground pixels consist of the objects and the Background pixels contain the background on which the objects are placed. The threshold value is calculated using Otsu Thresholding Algorithm. The calculation thoroughly looks for the limit that limits the intra-class variance, characterized as a weighted entirety of fluctuations of the two classes. The value corresponding to the maximum variance is considered as the threshold value of the gray scale image.



Fig -3: Binary Image

- **Component Filling:** In order to remove background pixels inside an object, a component filling algorithm is used. The image pixels are inverted at first. Flood fill algorithm is used to fill the background and is used to replace the boundary pixels by foreground pixels. Lastly adding the inverted image and the boundary filled image and then again inverting it to get the expected image.

- Flood Fill Algorithm:** The flood-fill calculation takes three boundaries: a beginning hub, an objective tone, and a substitution tone. The calculation searches for all hubs in the cluster that are associated with the beginning hub by a way of the objective tone and changes them to the substitution tone. There are numerous manners by which the flood-fill calculation can be organized, yet they all utilize a line or stack information structure, expressly or verifiably. Contingent upon whether we consider hubs contacting at the corners associated or not, we have two varieties: eight-way and four-way individually.



Fig -4: Component Filled Image

- Connected Component Algorithm:** Connected-component labeling (CCL) is used to label objects. The number of objects can be counted by counting the number of distinct labels present in the binary Component Filled image. Depending on the complexity of the objects, it can use two methods for labelling objects namely One Pass and Two Pass. In this application Two pass method is used to ensure proper labelling even in complex scenarios.
- Two Pass method:** The two-go calculation repeats through 2-dimensional double information. The calculation makes two disregards the image. The main pass to allocate impermanent labels and record equivalences and the subsequent pass to supplant every transitory label by the smallest label of its comparability class.

4.1.3. Post-Processing.

The DXF file contains boundaries of the black and white pixels. The binary image obtained from thresholding may contain background pixels inside the objects. While raster to DXF conversion, these background pixels will be considered as an object resulting in faulty conversion. In order to eliminate this, Component Filling Algorithm is used to remove the background pixels inside the objects (foreground pixels). Also, for the mapping of RGB to Depth data, number

of objects are counted and coordinates of the objects in the RGB space is measured.

- RGB to Depth data mapping:** Depth image of the Xbox One Kinect has a resolution of 512×424 pixels. The depth coordinates provided by Kinect are converted into real time depth by using a set of Camera Parameters. After this conversion, the maximum depth from the base is calculated from the Depth array. With respect to the Field of View (FOV) of RGB and Depth images, a superset of pixels of both the images for which depth for RGB pixels can be mapped is obtained. Based on the coordinates obtained from the Connected Component Labelling Algorithm, the depth of RGB coordinates in the image are obtained. Once the Coordinates of objects in the RGB image is obtained, the minimum depth of an object is subtracted from the maximum depth of the Depth array, thus calculating the depths of the individual object. The x, y, z coordinates of the objects are then stored in a text file which is then used to print the depth of individual objects in the DXF file

4.2. RASTER TO DXF CONVERSION

The Raster to DXF conversion is implemented using a software called PoTrace or Polygon Tracing. The maximum depth of each object obtained in the previous step is printed in the DXF file along with respective objects.

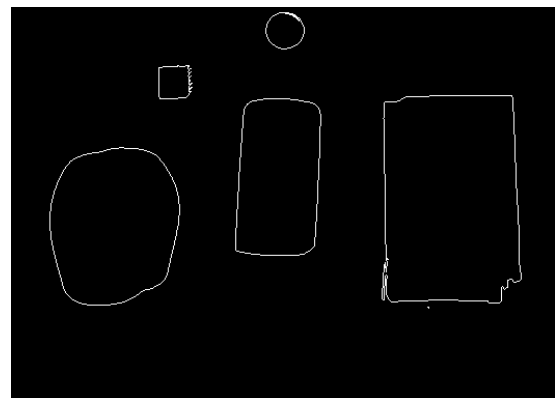


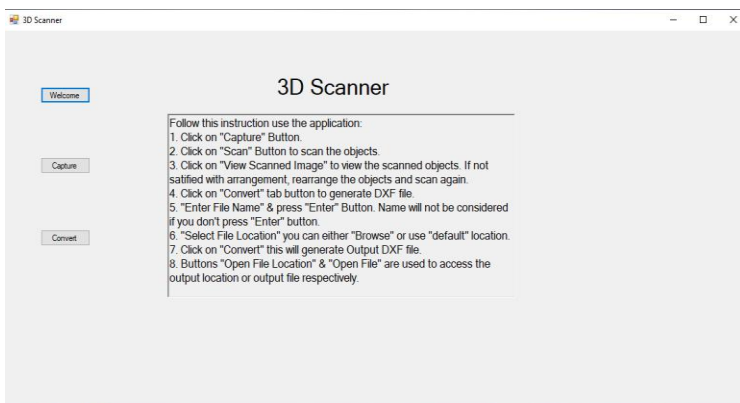
Fig -5: DXF image

4.3. GUI DEVELOPMENT

As the process of scanning the objects and converting them into a DXF file is complex it will require a highly educated person to operate the whole system. Also, it consumes a lot of time to carry out the whole process. Thus, we developed a graphical user interface for ease of end users and saving time. This GUI can be easily installed in any 64-bit windows operating system

This graphical user interface (GUI) is developed using Microsoft Visual Studio 2019. Considering the development of the GUI there are 4 important parts of the GUI namely:

- **Welcome Tab**
Used for instructing the user regarding the use of software.
- **Capture Tab**
Used for scanning of objects.
- **Convert Tab**
Used to convert the above captured data into a DXF file.



5. CONCLUSIONS

The aim of this project is to improve the process of creating CAD files used in the packaging industries. This project successfully scans the objects placed in the field of view and created precise DXF files along with third dimension (maximum height of the object) written inside the object.

We used XBOX one Kinect to capture RGB data and depth data of objects. This data is then processed using GUI which giving us precise DXF files. GUI is used to make the whole process user friendly, to make sure any man can use the product without special training. If we create appropriate lightning conditions using backlight table and black canopy to avoid external light interference, we will get better results than the normal environment.

ACKNOWLEDGEMENT

The outcome of this project required a lot of guidance and assistance from many people and we are extremely grateful to have it. We are grateful to our guide Prof. Lekha Das and our co-guide Prof. Sushma Kadge for providing us the necessary guidance to implement our project. We are very thankful to M/s Paxshell Pvt Ltd (packaging company from Mumbai) for sponsoring our project and guiding us in implementation of this project. We would like to express our

sincere thanks for giving us valuable guidance and we owe our deep gratitude to our project guides who took keen interest on our project work and guided us all along. We would also like to thank Prof. Bharathi Khedakar for constantly supporting and guiding us throughout the project.

REFERENCES

- [1] M.R. Andersen, T. Jensen, P. Lisouski, A.K. Mortensen, M.K. Hansen, T. Gregersen and P. Ahrendt: "Kinect Depth Sensor Evaluation for Computer Vision Applications," 2012. Department of Engineering, Aarhus University. Denmark. 37 pp. - Technical report ECE-TR-6M.
- [2] L. Cruz, D. Lucio and L. Velho, "Kinect and RGBD Images: Challenges and Applications," 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials, Ouro Preto, 2012, pp. 36-49, doi: 10.1109/SIBGRAPI-T.2012.13.
- [3] Lachat, E.; Macher, H.; Landes, T.; Grussenmeyer, P. Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potential Use for Close-Range 3D Modeling. *Remote Sens.* **2015**, *7*, 13070-13097.
- [4] Liu, Andy & li, Hengyu & Luo, Jun & Xie, Shaorong & Sun, Yu. (2019). Construction of All-in-Focus Images Assisted by Depth Sensing. *Sensors (Switzerland)*. 19. 10.3390/s19061409.
- [5] Casey Reas and Ben Fry. *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press, 2014.
- [6] Peter Selinger, "Potrace" [Online] Available: <http://potrace.sourceforge.net/> [Accessed Oct. 08, 2020].
- [7] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in ICCV'01: Proceedings of the 8th International Conference on Computer Vision, vol. 2, July 2001, pp. 416-423.
- [8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, second edition. Cambridge, MA, USA: MIT Press, 2001, ch. Chapter 22: Elementary Graph Algorithms, pp. 552-556.
- [9] A. Rosenfeld, "Connectivity in digital pictures," *Journal of the ACM*, vol. 17, no. 1, pp. 146-160, 1970.
- [10] K. Suzuki, I. Horiba, and N. Sugie, "Linear-time connected component labeling based on sequential local operations," *Computer Vision and Image Understanding*, vol. 89, no. 1, pp. 1-23, 2003.
- [11] W. D. Hillis and G. L. Steele, Jr., "Data parallel algorithms," *Communications of the ACM*, vol. 29, no. 12, pp. 1170-1183, 1986