

3D SCANNER USING WEB CAMERA AND LASER

Lekha Mirjankar¹, Alpesh Yadav², Ajesh DS³

¹Student, Dept. of Computer Engineering, Terna Engineering College, Maharashtra, India

²Student, Dept. of Computer Engineering, Terna Engineering College, Maharashtra, India

³Student, Dept. of Computer Engineering, Terna Engineering College, Maharashtra, India

Abstract - The purpose of this paper is to create a 3D scanner using a web camera and a laser line generator. The web camera will give the output image as the laser line generator moves, which will be used to estimate the depth of the object. The images will undergo image conversion techniques and noise removal to get the most efficient result. A collection of such images will be done with a motorized mount for scanning the surface and constructing a depth map of the object. This will be especially helpful in the fields of computer vision and robotics.

Key Words: 3D scanner, laser line generator, image conversion, depth map

1. INTRODUCTION

A depth map, like any other image relays information related to the object; however along with 2D information (x and y values), a depth map also gives the height (z value). It gives data about the distance of the object from a particular viewpoint, which is why depth maps are highly useful in 3D technology. In computer graphics, the z value i.e., the depth information is crucial in reconstructing surfaces and objects, in face detection, etc. A laser line generator helps to gather all the points of the object. Line generator can be used to display a uniform straight reference line for use in alignment, machine vision systems, construction and process control. The laser line generator serves as a contactless way to scan the surface and outputs images line by line as it illuminates then. All these images will be processed by applying image conversion techniques, histogram equalization and noise removal. After this, mathematical equation will be used to get 3 output points which will give us the depth.

Following are the objectives of this project:

1. To make a 3D scanner with the help of a laser line generator and webcam.
2. To apply image conversion techniques on the laser line output images like converting the bmp image to grayscale, h, s, v and monochrome.

3. To remove speckle noise caused due to scattering and make the project more efficient.
4. To derive a mathematical equation for calculation of x, y, z points.
5. To get the spatial points by applying the mathematical equation.
6. To generate a depth map for the object using the spatial points.

2. LITERATURE REVIEW

1. In the project titled '3D Photography on your desk' [1], a simple method for extracting 3D shape of objects has been demonstrated. It is based on 'weak structured lighting' and differs from more conventional approaches as it also requires very less hardware and the system is very straightforward. The object is illuminated by a desk-lamp and a camera faces it; the user moves a pencil in front of the desk-lamp to cast a shadow. From the location of the shadow, the shape of the image is extracted. They have presented experimental results on three different scenes and shown that the reconstruction error is less than 1%.
2. In [2], a depth image is acquired using Kinect camera frame-by-frame and then each frame is updated. After this, the output is further processed by using depth image smoothing algorithms and edge detection techniques. They have used the method of 'Median Filtering' for edge preservation and performed morphological operations like erosion and dilation for edge detection. This system aims at detecting edges of the depth image and help in the areas of object detection and object segmentation.
3. Indian Institute of sciences has presented a paper on 'High Quality Photometric Reconstruction using a Depth Camera' using the Kinect depth camera [3]. They have used IR depth camera system to obtain high resolution photometric information. This method fuses depth and normal information from multiple viewpoints so as to

reduce the problem of getting shadows and self-occlusions and build a complete, consistent and accurate 3D surface representation.

4. The proposed method [4] is comprised of a smoothing process for noise reduction, filling defects in the foreground area, and refining the filled depths. Depth sensors are widely used in such applications however there are certain limitations to it as the depth values of distal objects are not always captured. This is often seen in sensors in head-mounted displays; to avoid this a method has been proposed in this paper to conserve and reconstruct foreground shapes
5. In this study [5], to types of 3D image reconstruction methods are explored and then merged to acquire the most efficient results. High quality stereo images are taken using a webcam and the computation of depth information is done using Kinect to get an enhanced reconstruction.

The problem with using 'weak structured lighting' and a limited hardware is that it adds constraints to the system and makes it less scalable. Our aim is to create a system that focuses on the computation of depth-map while leveraging on the power of our hardware to reduce the environment constraints and make our system more flexible. In our system, the aim is not just object detection but to get the spatial points to generate depth map. Our system is not limited to HMD's and can be used in wide range of applications.

3. PROPOSED SYSTEM

3.1 ARCHITECTURE

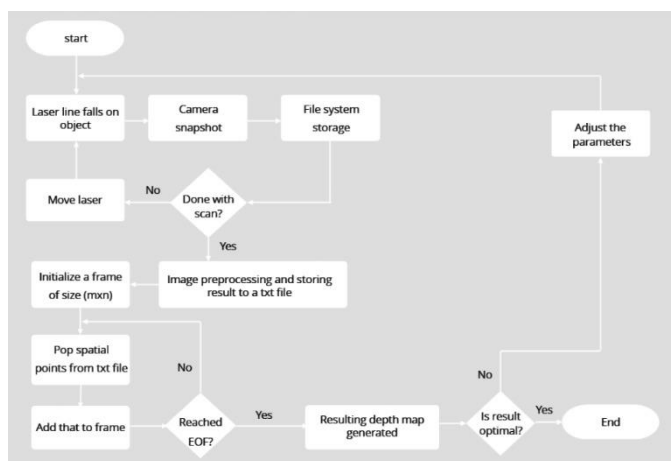


Fig -1: Architecture Diagram

3.2 COMMUNICATION PROTOCOL

SIZE -> a byte to represents the length of DATA

STATUS -> status code of the receiver that will help to know if data is received or not

DATA -> actual data carrying the command to be performed on each side

Table -1: Packet Structure

| Byte | Field | Definition |
|-----------|-----------|---|
| 0 | STX(0x02) | Start |
| 1 | STATUS | Status code of the receiver that will help to know if data is received or not |
| 2 | SIZE | A byte to represent the length of DATA |
| 3 to SIZE | DATA | Operations to be performed |
| SIZE + 3 | CHECKSUM | For error detection |
| SIZE + 4 | ETX(0x03) | End |

Set of Operations

1. Set Motor Direction (clockwise/anti-clockwise)
COMMAND: direction (0x0D)
VALUE: 0x00 (clockwise) / 0x01 (anti-clockwise)
2. Set Motor Speed
COMMAND: speed (0x0C)
VALUE: 0x00-0xFF (1 byte)
3. Turn on/off Laser Line Generator
COMMAND: Laser (0x0A)
VALUE: 0x00 (off) / 0x01 (on)

3.3 System

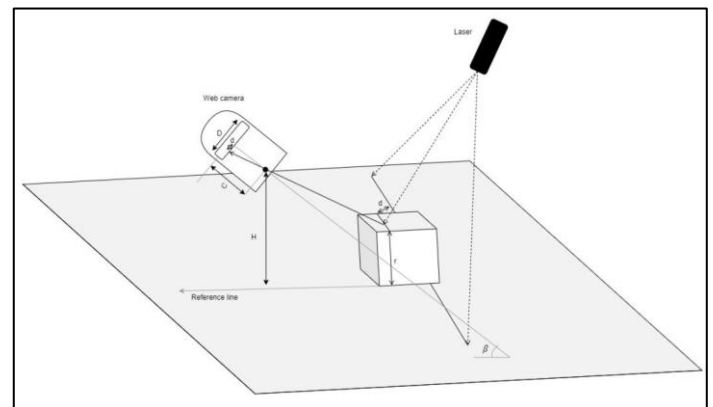


Fig -2: System Setup

The web camera is stationed at a particular position and it serves as our viewpoint. Consider, the cube as our object.

The laser moves slowly as the webcam captures the images one-by-one. Following are the parameters that can be seen in the above camera:

r -- height of object (z-value)

H -- height from reference to projection center

D -- physical size of image detector

β -- angle of impact with respect to reference level

d -- difference between position of laser line in background and position of laser line on object

C -- distance from projection center to image plane

After getting the images, we perform image conversion techniques on them which help us in further computation. On the grayscale image, histogram equalization and thresholding is done, after which we apply median filtering on it to reduce the noise. These processed images are then used to get the spatial points by applying the mathematical equation which we have derived.

4. IMAGE CONVERSION TECHNIQUES

Our webcam outputs images in the BMP format [6][7] - the first part is a header, followed by an information section, if the image is indexed color then the palette follows, and last of all is the pixel data. These RGB bitmap images are then converted to grayscale and HSV.

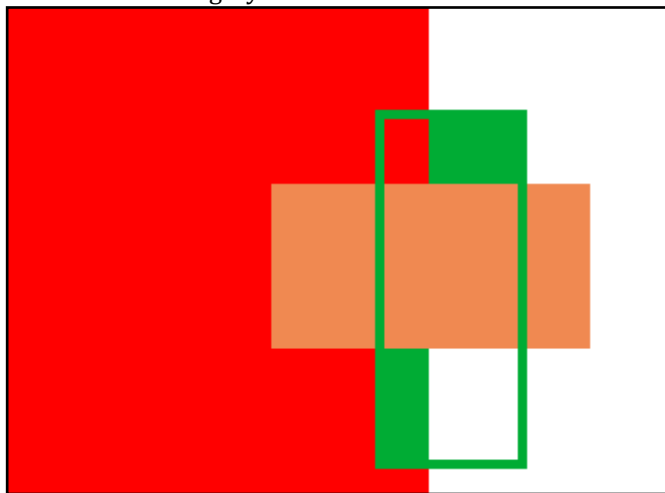


Fig -3: Input Image

RGB to grayscale

Grayscale image can be defined as an image having only range of monochromatic values from black to white. Each pixel represents information regarding only the intensity of light. For converting RGB image to grayscale, we have used linear approximation:

$$Y' = 0.299R' + 0.587G' + 0.114B'$$

This method is far better than just averaging the three-color channel values.

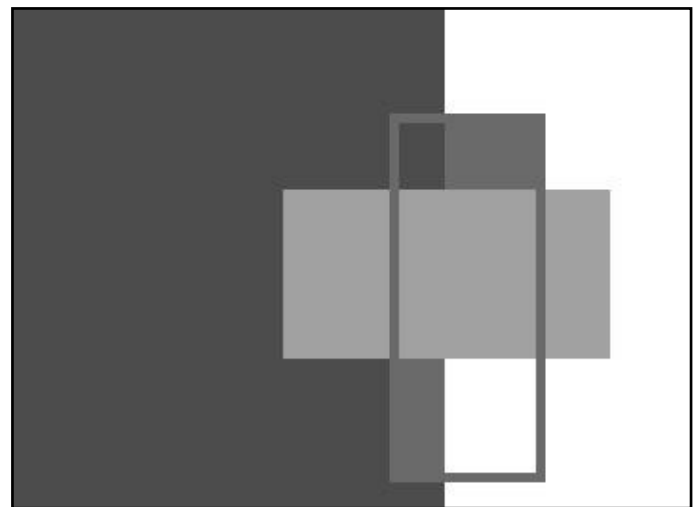


Fig -4: Grayscale Image

Grayscale to HSV

The Hue-Saturation-Value model is based on color characteristics as tint, shade and tone. The hue value H runs from 0 to 360°, the saturation S is the amount of color and ranges from 0 to 1, V is how bright the color is and it also ranges from 0 to 1. We have applied algorithms to get each channel (H, S, V) separately as output.



Fig -5: H Channel

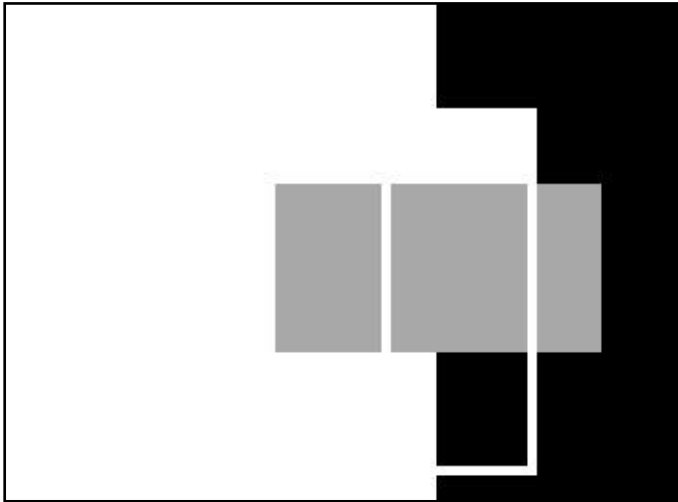


Fig -6: S Channel

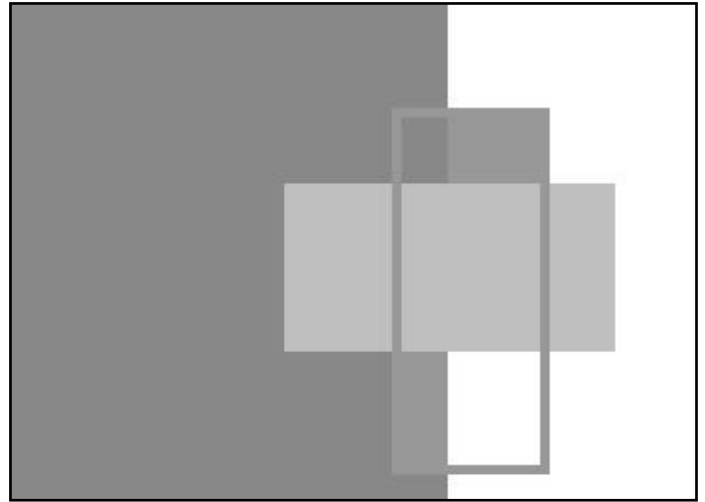


Fig -8: Histogram Equalization

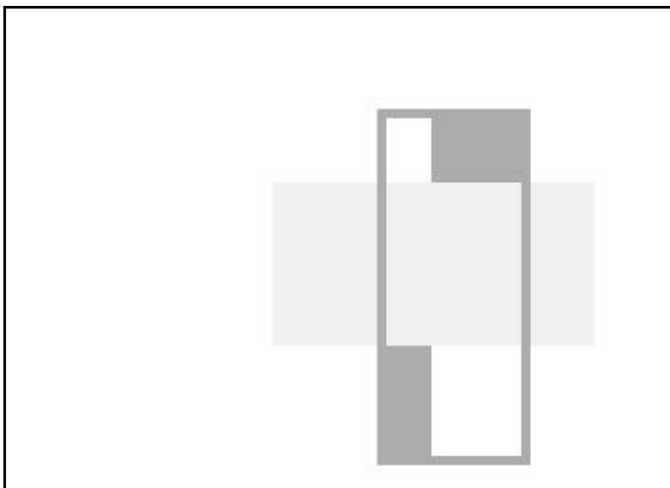


Fig -7: V Channel

Thresholding

Thresholding [8], [12] takes any pixel whose value is on the object side of the point and sets it to one; it sets all others to zero. It is a very simple method of image segmentation to get binary image i.e., either black or white; devoid of any gray values. We have used Otsu's algorithm [12]-[14] to select the threshold which is based on the principle "measure of region homogeneity is variance."

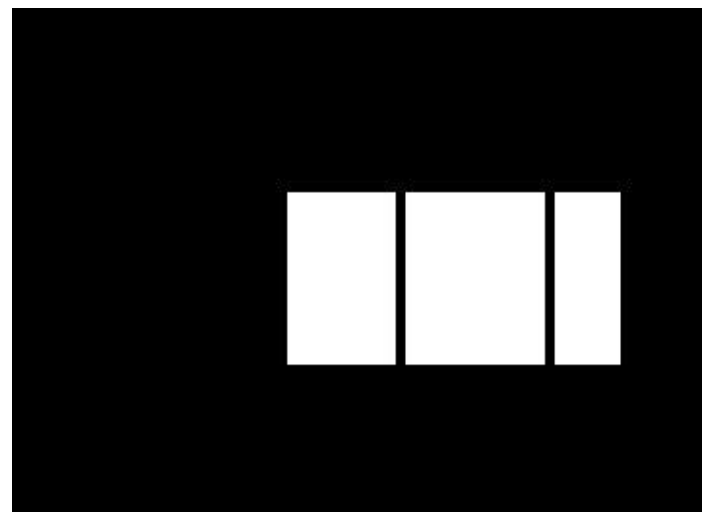


Fig -9: Thresholding

5. HISTOGRAM OPERATIONS

Histogram Equalization

In this method, the histogram of the image is processed to adjust the contrast. An area which grouped together is spread out into a flat histogram in this approach. Spreading or flattening the histogram makes the dark pixels appear darker and the light pixels appear lighter [8]-[11]. After implementing histogram equalization, the contrast appears to be enhanced and appears to be evenly distributed.

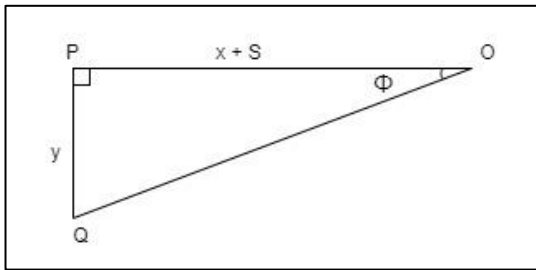


Fig -13

In ΔOPQ ,

$$\tan \phi = \frac{y}{S+x} \quad \dots (6)$$

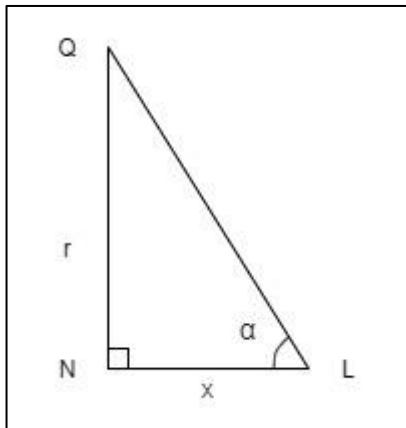


Fig -14

In ΔQNL ,

$$\tan \alpha = \frac{r}{x} \quad \dots (7)$$

$$H = r + y$$

$$H = r + (S+x)\tan \phi \quad \dots \text{from(6)}$$

$$H = r + \left(S + \frac{r}{\tan \alpha} \right) \tan \phi \quad \dots \text{from(7)}$$

$$r = H - \left(S + \frac{r}{\tan \alpha} \right) \tan \phi$$

$$r \tan \alpha = H \tan \alpha - S(\tan \phi \cdot \tan \alpha) - r \tan \phi$$

$$r = \frac{H \tan \alpha - S \tan \phi \tan \alpha}{\tan \alpha + \tan \phi} \quad \dots (8)$$

In our setup,

$$\alpha \rightarrow \frac{\pi}{2}$$

so,

$$r = \frac{H \tan \alpha - S \tan \phi \tan \alpha}{\tan \alpha + \tan \phi} \quad \dots (8)$$

$$r = \frac{H - S \tan \phi}{1 + \frac{\tan \phi}{\tan \alpha}}$$

since,

$$\lim_{\alpha \rightarrow \frac{\pi}{2}^-} \tan \alpha = +\infty$$

then,

$$\lim_{\alpha \rightarrow \frac{\pi}{2}^-} \frac{\tan \phi}{\tan \alpha} = 0$$

$$r = \frac{H - S \tan \phi}{1 + 0}$$

$$= H - S \tan \phi$$

$$\therefore = H - \frac{H(C - D \tan \beta)}{D + C \tan \beta} \cdot \frac{C \tan \beta - d}{C + d \tan \beta} \quad \dots \text{from(2)\&(5)}$$

$$r = H \cdot \left(1 - \frac{(C - D \tan(\beta)) \cdot (C \tan(\beta) - d)}{(C \tan(\beta) + D) \cdot (C + d \tan(\beta))} \right)$$

7. DETECTING LASER LINE PIXELS

Algorithm

- Make a monochrome equivalent of the BMP file
- Use median filtering [2], [15]-[17] to reduce noise. Median filtering replaces every external valued pixel by the median of its neighborhood pixels. It is less sensitive to outliers.

Steps:

1. Take each pixel
2. Sort neighbors according to intensity
3. Replace pixel value with median value

We find the mid value of the largest contiguous series of white pixels in both the first row and last row. This is because we need two points (start and end) to generate a profile line. We find the mid value using the following algorithm/pseudocode.

```
width := Image width
index := 0
midValue := 0
While index < width do
  pixel := Image[index]
  If pixel is White do //White = 1
    count := 1
    While pixel is not Black do //Black = 0
      pixel := Image[count+index]
      count := count + 1
    If midValue < count do
      midValue := count
      index := index + count
  Else do
    index := index + 1
  midValue = midValue / 2
```

Detecting the location of discontinuity

We use the laser projector on the surface without placing any object and get the reference line. Profile line is the actual line that we get when object is introduced. The start and end points of both these lines will be different. Introduction of object brings discontinuity in the reference line.

To detect the location of this discontinuity we follow the below steps:

1. After obtaining the reference start and end points, we will scan each row.
2. Find the mid-value of the largest contiguous series of white pixels on those rows, then calculate the perpendicular distance between that point and the reference line.
3. We'll skip any rows that don't have any white pixels and continue again to step 2.

8. RESULTS AND DISCUSSION

Our system is able to get the laser scan images, process them and compute the spatial points which finally give the depth map.

Errors

If the real system has an alignment issue, then the scan angle will vary from the ideal system and result in error. Most surfaces are rough and can produce unwanted noise due to scattering called as speckle noise when laser is used on them. This can also result in errors. Inaccuracy in position and distance of camera, laser, object can lead to positional errors and affect calculations [19].

$$\Delta r = \frac{\Delta d}{M \cdot \cos \beta}$$

Using the above relation, we can say that, the more the difference in d value, will directly affect the height value.

Limitations

The surface which is out of the laser's line of sight will not be measured, so complex geometry of an object can cause issues. Any form of external light will add unnecessary noise as we only want the system to read the laser light; because of which it is necessary to conduct scanning in a controlled environment. The material of the object will also affect the results as there will be more speckle noise in rough surfaces and also in highly transparent and shiny surfaces. In terms of concave and convex surfaces, because of reflection, a lot of noise will be registered [18].

9. CONCLUSIONS

Our project considers all the aspects required for capturing the images to processing them to getting the accurate spatial points required for depth-map creation. It is an affordable an easy to create method which promises optimal outcome. The setup for our system is simple, hassle free and feasible. In computer vision, depth maps are heavily relied on to get proper image information. In our system, we have minimized errors as much as possible to get an efficient result. Our project takes into account the

proper setup, image processing techniques, noise removal methods, error reduction methods and a comprehensive mathematical derivation to give the spatial points which make it possible to develop the depth map of the 3D object.

10. REFERENCES

- [1] "3D Photography on your desk - ICCV98 - Jean-Yves Bouguet - 3D scanner - 3D scanning - Shadow scanner - Shadow scanning," [Online]. Available: <http://www.vision.caltech.edu/bouguetj/ICCV98/>.
- [2] S. M. Abid Hasan and K. Ko, "Depth edge detection by image-based smoothing and morphological operations," *J. Comput. Des. Eng.*, vol. 3, no. 3, pp. 191–197, 2016, doi: 10.1016/j.jcde.2016.02.002.
- [3] S. M. Haque, A. Chatterjee, and V. M. Govindu, "High quality photometric reconstruction using a depth camera," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2283–2290, 2014, doi: 10.1109/CVPR.2014.292.
- [4] N. Awano, "Approximate depth shape reconstruction for RgB-D images captured from HMDs for mixed reality applications," *J. Imaging*, vol. 6, no. 3, 2020, doi: 10.3390/jimaging6030011.
- [5] W. Jia, W. J. Yi, J. Saniie, and E. Oruklu, "3D image reconstruction and human body tracking using stereo vision and Kinect technology," *IEEE Int. Conf. Electro Inf. Technol.*, 2012, doi: 10.1109/EIT.2012.6220732.
- [6] Lastnameholiu, "BITMAP (wingdi.h) - Win32 apps." <https://docs.microsoft.com/en-us/windows/win32/api/wingdi/ns-wingdi-bitmap>.
- [7] "BMP files." <http://paulbourke.net/dataformats/bmp/>.
- [8] P. Abbott, "Image Processing in C," *Assem. Autom.*, vol. 18, no. 3, p. aa.1998.03318cad.010, Sep. 1998, doi: 10.1108/aa.1998.03318cad.010.
- [9] Y. T. Kim, "Contrast enhancement using brightness preserving bi-histogram equalization," *IEEE Trans. Consum. Electron.*, vol. 43, no. 1, pp. 1–8, 1997, doi: 10.1109/30.580378.
- [10] J. A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 889–896, 2000, doi: 10.1109/83.841534.
- [11] and M. R. V. Premchandran, P. Logamurthy, M. Karthikkumar, V. Thamizharasan, G. Lingeswaran, "A Review on Image Enhancement Techniques using Modified Approach of Histogram Equalization," *Int. J. Res. Dev. Technol.*, [Online]. Available: https://www.researchgate.net/profile/Thamizharasan-Viswanathan/publication/350735568_A_Review_on_Image_Enhancement_Techniques_using_Modified_Approach_of_Histogram_Equalization/links/606f1ef192851c8a7bb2bcb8/A-Review-on-Image-Enhancement-Techniques-using-Modified-Approach-of-Histogram-Equalization.pdf.
- [12] P. et al. Jain et al., Thresholding. .
- [13] Y. Chen, D. R. Chen, Y. Li, and L. Chen, "Otsu's thresholding method based on gray level-gradient two-dimensional histogram," *CAR 2010 - 2010 2nd Int. Asia Conf. Informatics Control. Autom. Robot.*, vol. 3, pp. 282–285, 2010, doi: 10.1109/CAR.2010.5456687.
- [14] X. Yang, X. Shen, J. Long, and H. Chen, "An Improved Median-based Otsu Image Thresholding Algorithm," *AASRI Procedia*, vol. 3, pp. 468–473, 2012, doi: 10.1016/j.aasri.2012.11.074.
- [15] Y. Zhu and C. Huang, "An Improved Median Filtering Algorithm for Image Noise Reduction," *Phys. Procedia*, vol. 25, pp. 609–616, 2012, doi: 10.1016/j.phpro.2012.03.133.
- [16] and G. G. A. S. A. Villar, S. Torcida, "Median Filtering: A New Insight," *J. Math. Imaging Vis.*
- [17] U. Ali, I. H. Lee, and M. T. Mahmood, "Guided image filtering in shape-from-focus: A comparative analysis," *Pattern Recognit.*, vol. 111, 2021, doi: 10.1016/j.patcog.2020.107670.
- [18] B. Thörnberg, "Machine vision 3D imaging techniques," [Online]. Available: http://apachepersonal.miun.se/~bentho/machine_vision/download/f7.pdf.
- [19] Y. M. Amir and B. Thörnberg, "High Precision Laser Scanning of Metallic Surfaces," *Int. J. Opt.*, vol. 2017, 2017, doi: 10.1155/2017/4134205.