Real time object measurement using canny edge detection.

K. ChandraSekhar[1], M.S. N Abhiram[2], T. ChandraMouli[3], M. Roshan Raghunadh[4], B. Sandeep[5]

[1]Associate Professor, Department of Computer Science & Engineering, Anil Neerukonda Institute of Technology & Sciences, Visakhapatnam (Andhra Pradesh), INDIA

[2,3,4,5]B Tech Students, Department of Computer Science & Engineering, Anil Neerukonda Institute of Technology & Sciences, Visakhapatnam (Andhra Pradesh), INDIA

Abstract – **Measurement is perhaps one of the most fundamental concepts in science. Without the ability to measure, it would be difficult for scientists to conduct experiments or form theories. Measurement is perhaps one of the most fundamental concepts in science. Without the ability to measure, it would be difficult for scientists to conduct experiments or form theories. In this paper we will find the accuracy of the canny edge detection for different object and find the dimensions of the object. The measurement uses computer vision algorithms for scene reconstruction to obtain the object's point cloud. "Pixel per Metric" algorithm for processing the point cloud and for estimating the dimensions of the object.**

Keywords: **Canny edge algorithm, pixel per metric, Object measurement, Pre-processing, OpenCV, Computer Vision.**

1.Introduction:

Real time object measurement are very vital tasks in our day-to-day lifestyle. Measuring is about finding the dimensions of a specific object. Measuring is done with help of basic tools like tapes, ruler, or a protractor. When these tools are not in handy, we get real issue.

Our proposed system can deal with the current problem by providing a handy app in our mobile phones. There is a high chance that everyone has a mobile phone in their hands rather than having a tapes/ruler. To calculate the size of each object, firstly we need to determine the reference object. After that, the dimensions of the reference objects will be used to calculate the size of other objects.

Our phone captures the image using the phone camera which is inbuilt in our mobile phone. The image is stored in a local storage, and we consider the reference object and deduce the measurements of the object which is taken from the mobile. These measurements are displayed to the user.

1. Proposed System for object Measurement:

This system is divided into two steps a) Object detection b) Object measurement.

In the first part we capture the objects including reference and desired objects (whose dimensions to be find). We capture the image by using our mobile camera and this image is sent to custom AWS Lambda function which internally implements the canny edge detection algorithm to find the contours of the image and sends it back to the mobile. In the second part we find the dimensions of the desired object with help of referred object by using pixel per metric ratio. In this proposed system reference object is fixed before implementing the procedure. Figure 1, shows the architecture of the system where phone storage is used for temporary storage of the images and Amazon S3 is used for Lambda function storage.

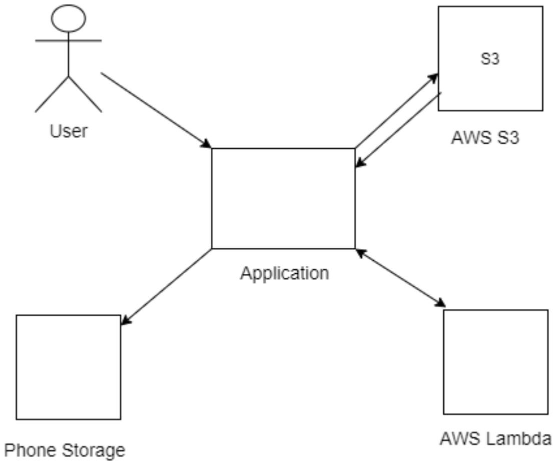

Figure 1. Shows the architecture of the system.

1.1 Object detection from the image

After capturing the image, we pre-process the image before applying canny edge detection algorithm. We apply the Gaussian blur to smooth the input image i.e., Figure 2 and output of the pre-processed image is on Figure 3.

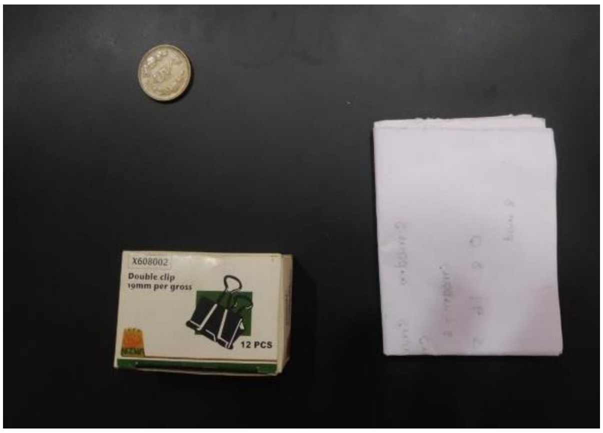

Figure 2. Captured input image.

After applying gray scale conversion and gaussian blur our pre- processed image will be Figure3.We convert image to gray scale because computation will be easier because no of pixel values reduces and gaussian blur is applied such that to reduce the noise of the captured image.

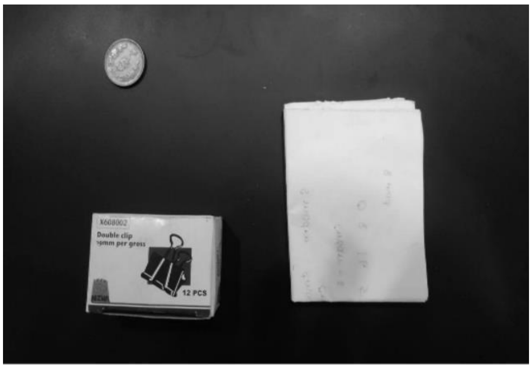

Figure 3. Gray Scale converted image.

In the next stage we compute gradient, we detect the edge gradient and direction for each pixel. For the define the gradient at every pixel of smoothened frame, Sobel operator utilized. Sobel operator gives us different gradient unwanted pixels that do not form the edge are eliminated as you can see in the figure 4 below.

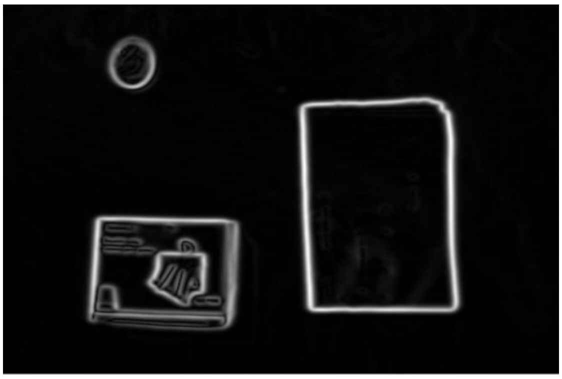

Figure 4. Gradient intensity

Smoothened image is then filtered with a Sobel kernel in both horizontal and vertical direction to get first derivative in horizontal direction (Gx) and vertical direction (Gy). From these two images, we can find edge gradient and direction for each pixel as follows:

$$\text{Edge Gradient}(G)=G_{2x}+G_{2y}$$

$$\text{Angle}(\theta)=\tan^{-1}(G_y/G_x)$$

For betterment of the results, we need thin edges rather than thick, so we include non-maximum suppression.

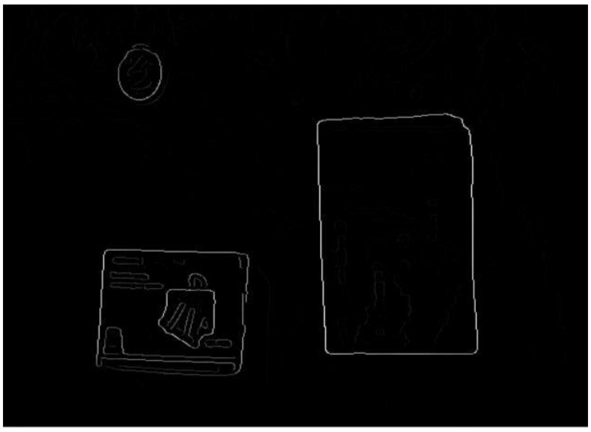

Figure 5. Result of Non-Maximum suppression

Form figure 5 though the edges are thin we can observe variation in edge intensity i.e., some pixels are brighter than rest of the pixels. This is eliminated in double threshold where only two pixels intensities are considered. In next step edge tracking by hysteresis, we convert the weak intensity pixel to high intensity pixels such that we get a well-defined edge of the objects in the image.

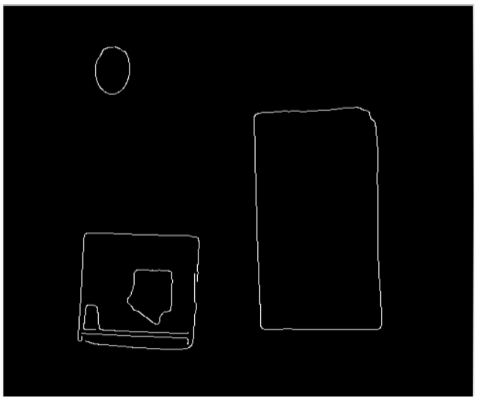

Figure 6. Hysteresis edge tracked.

Dilation and Erosion operations are also done to reduce the noises of the image. These two morphological processing operations that produce contrasting results when applied to Gray-scale images.

Figure 6 shows the final output of the image after the edges are tracked.

1.2 Object Size Measurement

In this step we will we find contours by using OpenCV cv2.findContours to find the shapes of the objects in the edge map. After finding the contours of the image we draw a rectangle box around the contours The reference object which is placed in the top left of the image (Pre calibrated). After we get mid points of the rectangle, we calculate pixel per metric variable for the reference object and calculated the length and width of each individual desired object in the image. Euclidean distance among sets of center points.

2. Results

Image is captured using mobile camera seeing angle perpendicular to the surface of the image as seen in Figure 7.

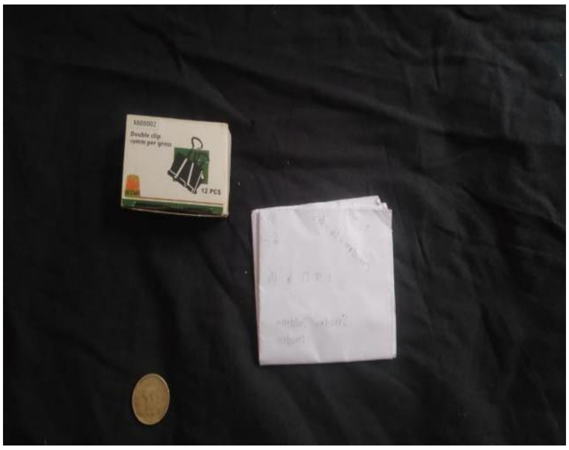

Figure 7. Input image

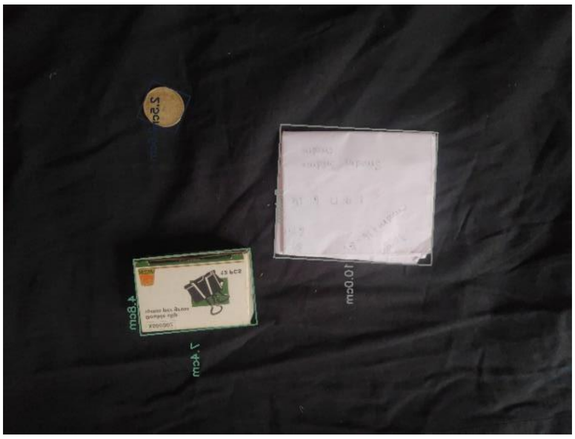

Figure 8. Output image

Figure 8 shows the output image where reference object is the coin and desired objects are clip box and White paper. Table I, shows the accuracy of proposed object measurement system for these objects. Abbreviations in the table are as follows; AM-H: Actual Measure-Height, PM-H: Proposed Measure- Height, AM-W: Actual Measure-Width, PM-W: Proposed Measure-Width.

TABLE Accuracy of different objects

NAME OF THE OBJECTS	AM-H (cm)	PM-H (cm)	AM-W (cm)	PM-W (cm)	ACCURACY (%)
Coin	2.5	2.5	2.5	2.6	96
Clip Box	4.3	4.8	6	7.4	83
White paper	7.7	8.1	8.9	10	91.5

Table 1 shows the data for the output of the image. The accuracy is high as it depends on the objects and background of the image, angle of the camera.

4.Conclusion

In this study, a powerful real time object measurement is proposed for industrial systems. Proposed systems can identify can detect and measure the size of the objects. We have enhanced the canny edge detection algorithm by using morphological operations to reduce the noises.

The proposed system works fast and accurate if all the above constraints are taken care of as the system run on cloud it does not take my time for computation.

5.References

[1] Muthu Krishnan. R and M. Radha “Edge Detection Techniques for image Segmentation” International Journal of Computer Science & Information Technology (IJCSIT) Vol3, No 6, Dec 2011.

[2] Moeslund T. Canny Edge detection. Denmark: Laboratory of Computer Vision and Media Technology, Aalborg University. ⟨http://www.cvmt.dk/ education/teaching/f09/VGIS8/AIP/canny_09 gr820.pdf⟩ [March 2009].

[3] Geng Xing, Chen ken, Hu Xiaoguang “An improved Canny edge detection algorithm for color image” IEEE TRANSATION ,2012 978-1- 4673-0311-8/12/$31.00 ©2012 IEEE.

[4] Chen.W, Yue.H, Wang.J, Wu.X. An improved edge detection algorithm for depth map in painting. Optics and Lasers in Engineering 2014;55: 69–77.

[5] OpenCV, 2021. http://opencv.org/ [Accessed June 9, 2021]

[6] AWS,2021AWS General Reference - AWS General Reference (amazon.com) [Accessed June 9,2021].

[7] Android,2020 Documentation | Android Developers [Accessed December,2020].

[8] Nashwan Adnan OTHMAN and Mehmet Umut SALUR “An Embedded Real-Time Object Detection and Measurement of its Size” International Conference on Artificial Intelligence and Data Processing (IDAP) 2018.

[9] G. Bradsky, A. Kaehler, “Learning OpenCV, Computer Vision with the OpenCV Library” (Book Style), O’Reilly Media, Sebastopol, CA, 2008.

[10] Z. Xu, X. Baojie and W. Guoxin, "Canny edge detection based on Open CV," *2017 13th* IEEE International Conference on Electronic Measurement Instruments (ICEMI*)*, Yangzhou,2017, pp.53-56.doi: 10.1109/ICEMI.2017.8265710.