

Real-Time Object Measurement Using Image Processing

Sukesh Jiwane^{#1}, Ashutosh Panda^{#2}, Yash Basutkar^{#3}, Avinash Golande^{#4}

sukeshjiwane30@gmail.com

Article Info

Page Number: 3260-3267

Publication Issue:

Vol. 71 No. 4 (2022)

Article History

Article Received: 25 March 2022

Revised: 30 April 2022

Accepted: 15 June 2022

Publication: 19 August 2022

Abstract

From an industrial standpoint, real-time object measurement and dimensioning is a crucial component of the requirements of the current and developing world. When dealing with computer vision challenges, these are the topics that must be tackled. This paper shows how an IoT video device, such as a camera, may also be used to recognize objects and compute their real-time measurements. Based on NumPy, OpenCV libraries, and cameras we proposed an IoT-based real-time object measuring technique for calculating measurements and dimensions of objects. Many modules and algorithms from OpenCV are used in this project. There are five stages to this technique, which are as follows. Capture a picture, Measurement of an object, Calculating the area, Saving the output, Displaying the Measurements.

Keywords: - Object measurement, Area Calculation, Image Processing, Canny edge detection.

Introduction

The real-time object measurement system is a program for detecting the dimensions of objects in real-time. There aren't many real-time object measuring models out there, so this method of measuring and dimensioning an object in real-time that we've proposed here has a lot of potentials. This is a critical characteristic of an issue in computer vision. As indicated, this project depicts a real-time approach for calculating measurements of the objects from images taken by the user. It identifies the object using a webcam and hovers its dimensions in real-time, including the area, in the appropriate units of measure. We created a system that uses the OpenCV software package to implement the proposed technique.

Automated technology relies largely on numerous sensors and cameras, and it also has gained a lot of traction in recent years. Running robots necessitates far too much automation. It's also important to figure precisely where an object is, how tall it is, and how wide it is. We learned how to measure the size of an object using IoT devices like a webcam and OpenCV in this project.

This system has various benefits, including being highly effective in the industrial field and simplifying human work.

1.1. Related Work:

Recently, much research has been done on this topic, and several researchers have published their findings.

The YOLO algorithm, created by Joseph Redmon et al., uses CNN-based visual processing to recognize real-time objects. Its method is based on a few variables and is extremely quick and easy to implement. [1]

M. Naveen Kumar et al. presented the OpenCV implementation including Python and MATLAB. This research makes use of a CUDA-based GPU interface. OpenCV is now available for Windows, Android, and a variety of other platforms. This study now introduces OpenCV Robotic methods. [2]

The LiDAR Imaging System, developed by Santiago Royo et al., works in both 2D and 3D settings. This research looked into the various LiDAR system configurations available for self-driving automobiles. The development of new imaging techniques, scanners, and detector arrays is the key strategy.[3]

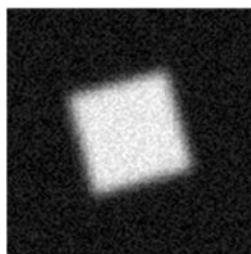
Zarine Abdelmoghith devised many real-time object detection methods. They used a stereo camera to accomplish their goal. The distance between two objects is detected and quantified using this method. [4]

Edge published Detection with Embedded Confidence, in which he elaborated on a core module in many computer vision operators that compute the heavy measure of the pixels in a frame. Meer, P., Georgescu, B.: The procedure is stated, in a linear space, emphasizing the importance of the many subspaces. [5]

A study on updated canny edge, presented by LiSheng Ding, looks over the theory behind the primitive canny edge thus, improves in location accuracy and has a more perceptible de-noising effect, according to the trial. [6]

2. Proposed Methodology

The system is primarily based on the use of Canny edge for detecting the edges and contouring part of an image given as an input, OpenCV for image processing and reading images from the system via an external webcam, and NumPy for rapidly and accurately formulating mathematical computations and determining the measurements of the desired objects placed within the constraints.



Noisy Image



Canny Filter $\sigma = 1$



Canny Filter $\sigma = 2$

Fig. 1 Canny Edge Filtration

2.1. Measurement and calculation of the dimensions:

The canny edge can precisely locate the edges and the vertices of the object placed within the periphery of the system. This provides the values for (X1, X2) and (Y1, Y2) respectively which are then taken as an input for implementing $\sqrt{(X2 - X1)^2 + (Y2 - Y1)^2}$. The system uses this formula for calculating the length and width of the object which are then used for calculating the area of the object.

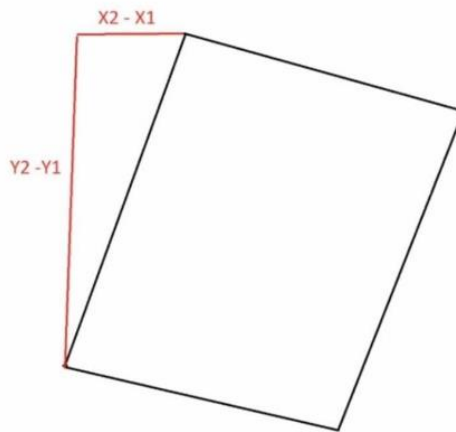


Fig.2 Diagram representing the formula used.

Object's length or width =

$$\sqrt{(X2 - X1)^2 + (Y2 - Y1)^2}$$

2.2. Implemented Algorithms

1. Noise reduction

Applying Gaussian blur to the image is one technique to smooth it out and get rid of the noise.

$$\frac{1}{2\pi\sigma^2} \exp \left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2} \right);$$

$$1 \leq i, j \leq (2k+1)$$

2. Gradient calculation

The Gradient calculation step analyses the edge strength using the edge-finding algorithms to calculate image grading.

$$|G| = \sqrt{I_x^2 + I_y^2}$$

$$(x, y) = \arctan \left(\frac{I_y}{I_x} \right)$$

3. Non-maximum suppression

This approach looks for pixels with the greatest value within the edge directions throughout all positions on the gradient intensity matrix.

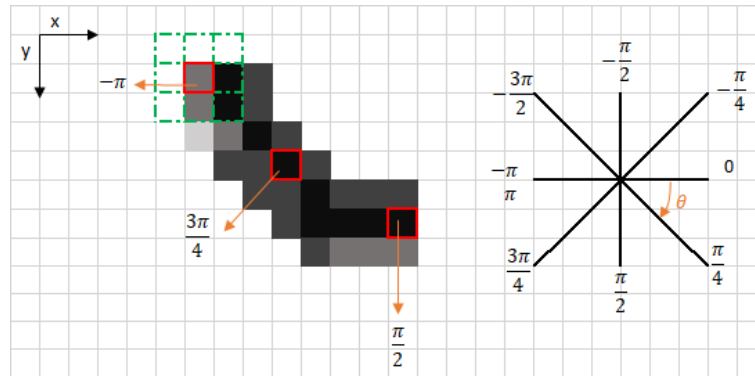


Fig.3 Non-maximum suppression

4. Double threshold

The double threshold approach attempts to distinguish between three main types of pixels: strong, weak, and unrelated.

5. Edge tracking with Hysteresis

Unless at least one pixel around the processor is strong, hysteresis converts weak pixels into solid ones based on the threshold of the results.

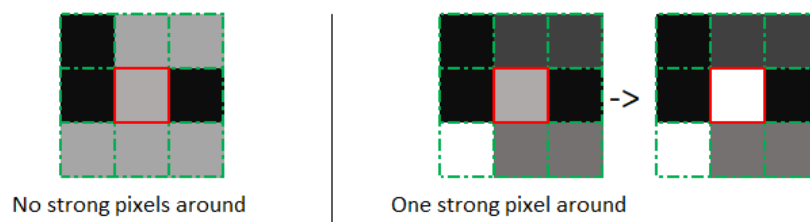


Fig. 4 Edge tracking with Hysteresis

3. Procedural Explanation

We require libraries available in python namely OpenCV and NumPy.

Further, a flag chunk of code to turn the webcam on or off which will contain a boolean value either “True” or “False”. "webcam = False," for example. Record the video. Set the Width, Height, and Parameters.

In PyCharm or any other IDE, download the OpenCV and NumPy requirements. Import NumPy as np and OpenCV as cv2.

As a result, the settings are complete. Then create the code that will be used for the measurement of the object in the image.

To obtain real-time object measurement, a python file (utlis.py) recognizes contours and applies methods to picture input as code.

The image is disturbed after dilation and deterioration.

- Generate a duplicate of the input image. To make the process of creating a canny image move as smoothly as possible, use dilation and erosion effects.
- The background provided to the system must be a white paper or it is equivalent and rectangular, the filter being applied should also be a rectangle similar to the white background.
- After finalizing the right contours with the Canny Edge Detector Algorithm along with the addition of the length and area to the objects within the resulting output, the white paper detection is specified.
- The system uses mathematical concepts and an illustration that measures the object that has been placed on white paper with the help of a mathematical formula and finds out the measurement of the object.
- It finds out the length, width, and area of the particular object which is present in the image and afterward denotes it over the image.
- The system uses arrowed lines and text to represent the object's dimensions, which are its length, width, and area over the image.

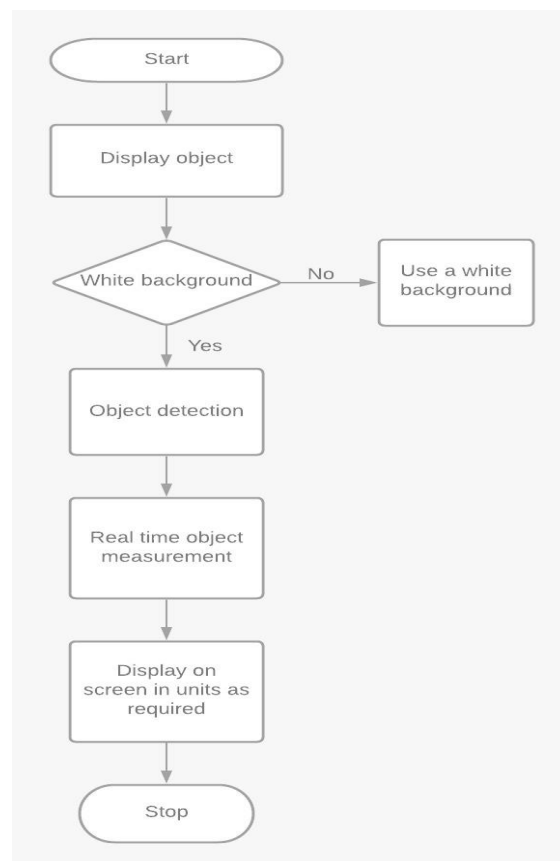


Fig. 5 Block drawing for Real-Time Object Measurement System

4. Experimental Analysis

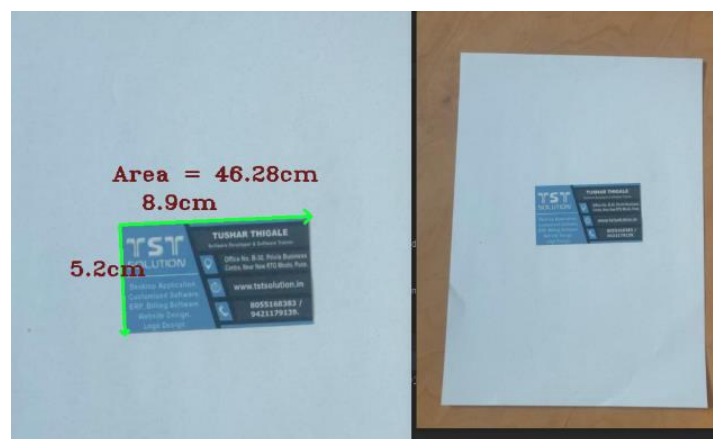
On conducting several tests on numerous objects our system was able to achieve an accuracy of 96.9% which is higher than any other object measurement and detection model previously developed.

Paper	Algorithm	Accuracy
Relative Performance Analysis of Edge Detection Techniques in Iris Recognition System	Canny for edge detection.	91.56%
Relative Performance Analysis of Edge Detection Techniques in Iris Recognition System	Sobel for edge detection.	79.40%
Relative Performance Analysis of Edge Detection Techniques in Iris Recognition System	Prewitt for edge detection.	76.10%
You Only Look Once: Unified, Real-Time Object Detection	Bounding Box Prediction.	78.60%
Real-Time Object Measurement Using Image Processing	Canny edge detection for object measurement.	96.90%

Table 1: Previously Tested Algorithms

5. Results

After running the system there comes two images as the output, with one being the original and the other with the measurements and dimensions displayed as an output. The original image is the image that the user has uploaded to the system. The result of the measurements of the objects is present below.



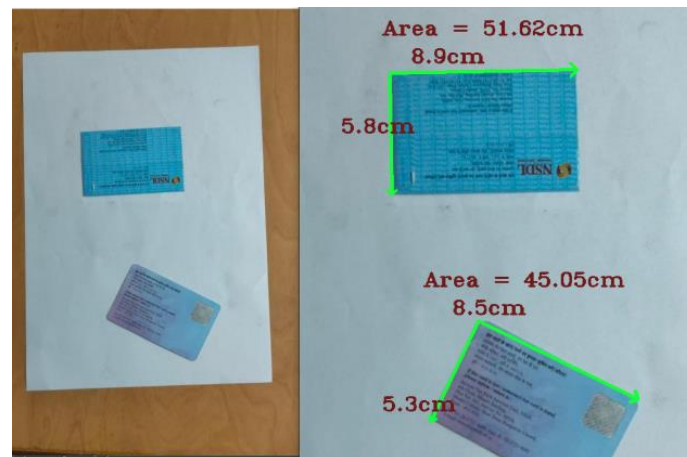


Fig. 6 Displayed Output of the System

6. Conclusion

This method can lead to many developments in various sectors prone to human errors caused during dimensioning and taking measurements. It depicts the proportions of objects within an image taken from a live video frame. Using a Canny edge detector, the dimensions are appropriately identified. This method is easy to use and has a variety of benefits and remarkable features that can be used in the real world.

7. References:

- [1] Erhan D, and C. Szegedy. Scalable object detection using deep neural networks. In CVPR, 2014 IEEE
- [2] R. Girshick, and J. Malik. In CVPR. 2014 IEEE Conference on, pages 580–587. IEEE, 2014.
- [3] J. Uijlings, and A. Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013.
- [4] Hariharan B, Arbelaez P, Girshick G, and J. Malik. Detection and segmentation pages 297–312. Springer, 2014.
- [5] J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In CVPR, 2014 IEEE Conference on, pages 2497–2504. IEEE, 2014.
- [6] C. L. Zitnick and P. Dollar. Edge boxes: In Proceedings of the CVPR pages 391–405. Springer, 2014.
- [7] Navneet B, Rama C, and Larry D. Soft-NMS–improving object detection with one line of code. In ICCV, pages 5561–5569, 2017.
- [8] Zhaowei C and Nuno V. Cascade R-CNN: Delving into high-quality object detection. In CVPR, pages 6154–6162, 2018.
- [9] Jiale C and Xuedong L. Hierarchical shot detector. In Proceedings of the ICCV, 2019.
- [10] K. Jeong and H. Moon, “Object Detection Using FAST Corner Detector Based on Smartphone Platforms”, First ACIS/JNU International Conference on Computers-Networks Systems and Industrial Engineering (CNSI), pp. 111 - 115, May 2011.

- [11] Liang C, George P, and Alan Y. In Proceedings of the ICCV, 40(4):834–848, 2017.
- [12] Terrance D and Graham T. Improved regularization of convolutional neural networks with CutOut. arXiv preprint arXiv:1708.04552, 2017.
- [13] Kaiwen D, Song B, and Q Tian. internet: Keypoint triplets for object detection. In Proceedings of the ICCV, pages 6569–6578, 2019.
- [14] Glanz Ghia, and Quoc V Le. In Proceedings of the CVPR, pages 7036– 7045, 2019.
- [15] Ross Girshick. Fast R-CNN. In Proceedings of the ICCV, pages 1440–1448, 2015.
- [16] Bharath Hariharan, and Jitendra Malik. Hyper columns for object segmentation and fine-grained localization. In Proceedings of the CVPR, pages 447–456, 2015.
- [17] G. Brodsky and A. Kaehler, “Learning OpenCV Computer Vision with the OpenCV Library” in, Sebastopol, O’Reilly Media, 2008.
- [18] Aydin and Othman N. In IDAP, pp. 1 - 6, 2017.
- [19] Othman N, and I. Aydin. In CICN, 108-112, 2017.
- [20] Abhinav Shrivastava, Abhinav Gupta. In Proceedings of the CVPR, pages 761–769, 2016.