Measurement of Objects' Size and Distance Between Them in an Image Using Perspective Transformation

Sourav Sarker
Computer Science Department
Memorial University of Newfoundland
NL, Canada
souravs@mun.ca

Abstract— **Object detection, dimensioning of the objects, and distance measurement plays a vital role in medical application, vehicle control, robotic movement control and many more. Undesirable measurement in objects' size and distance between them may occur as near things appear more prominent compared to those who are far away in a photograph. This study proposes a better method of measuring the objects' dimensions and distance between them based on perspective transformation followed by Gaussian filtering, canny edge detection technique and morphological operator with higher accuracy utilizing OpenCV libraries. With the goal of accurate measurement, this report compares the accuracy in dimensioning with and without perspective transformation on the images captures by smartphone.**

Index Terms— **Object detection, Perspective Transformation, Gaussian Filter, Canny Edge Detection, Morphological Operator, OpenCV.**

I. INTRODUCTION

In recent days, object tracking and dimensioning are fundamental tasks in computer vision-based applications such as surveillance and driving-assistance systems [2]. Nevertheless, in a captured image, during the conversion of the 3D world to a 2D image, undesirable changes in the objects' shape may occur from actual.

This paper aims to identify the sizes and degree of space between the objects in actual measurable units(cm) through a perspective transformation; to compare with the actual dimension of the detected object and to analyze the accuracy. The results are a reference for the vision-based detection system to determine how perspective transformation can reduce errors in measuring objects' dimensions in an image.

The proposed system has utilized the photos of targeted objects on an A4 size paper captured from an iPhone XR under a decent lighting condition to check the competence of the proposed system. Our model ideally performs well in applying the 4points perspective transformation, identifying targeted objects and separating from the background.

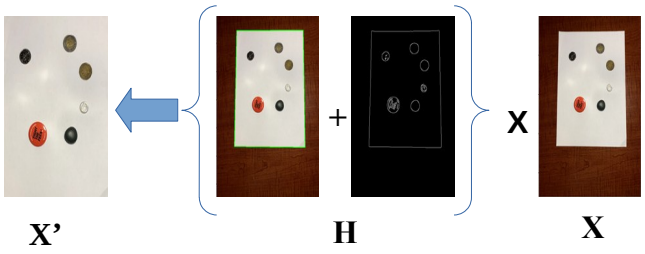

Figure 1: Perspective Transformation of the Image

For perspective transformation, we need to detect edges, utilize those edges to find the external contour of the A4 paper and apply a perspective transformation to get a top-down view of that paper. As shown in Figure 1, we can formulate the outcome of perspective projection problem as X' = H * X, where X is the input image, H is the perspective transformation, and X' is the output.

After the transformation, in order to calculate each object's dimension in an actual measurable unit (cm), we first need to perform a calibration using a reference object and a ratio of pixels captured by each object per given metric; in this case, it is cm [3]. We should know the measurement in cm of the object in terms of width or height and be able to locate this object based on its placement or via appearances as the left-most object is the reference object [1]. As shown in Figure 2, we will be using the Canadian quarter coin as the reference object.

We have organized the rest of the paper as follows. In section II, we discuss the one published approach for object detection and dimensioning. Section III presents our proposed methodology for object dimensioning and distance measurements. In section IV, we consider the feasibility of our model on images captured by smartphones.

II. RELATED WORK

The stated method on object detection and dimension measurement in [1] is the baseline for this report. In this section, we review that briefly.

A. *Object Detection ([1])*

Figure 2: Canadian Quarter as Reference Object

Firstly, after converting the captured frame through a raspberry pi camera into grayscale, a Gaussian filter filters out the noise by smoothing the frame. Secondly, to compute the gradient stage, the model uses a Sobel operator to detect the edge gradient and direction of each pixel, which helps eliminate any undesirable pixels that might not establish the edge. In the final stage, by hysteresis thresholding, two empirically selected threshold values help to recognize edges. After that, a combination of Morphological operations aids in obtaining a more accurate result in object detection. The Morphological initial operation takes place by erosion followed by dilation, and the operation closes by dilation followed by erosion.

B. Object Measurement ([1])

After detecting the targeted edges properly, an OpenCV function cv2.findContours maps the found contours and sorts them from left to right. Finally, the system determines the size of objects by calculating pixels per metric with the help of scanning every contour by performing calibration using a reference object, which is the leftmost contour in the contour list.

III. OBJECTS' DIMENSIONING AND DISTANCE MEASUREMENT USING PERSPECTIVE TRANSFORMATION METHOD

The proposed method consists of four parts, which are pre-processing the image for object segmentation and computation of results from a reference object. Figure 3 shows the recommended process of the system.

A. Image Pre-processing and Perspective Transformation

Firstly, the system reads a captured image through iPhone XR and resizes the image with a height of 650 pixels to speed up the image processing and make edge detection steps more accurate.

For a perspective transformation of the image, after RGB image to grayscale conversion, a 5x5 Gaussian filter that filters out the high-frequency noise. The image is now

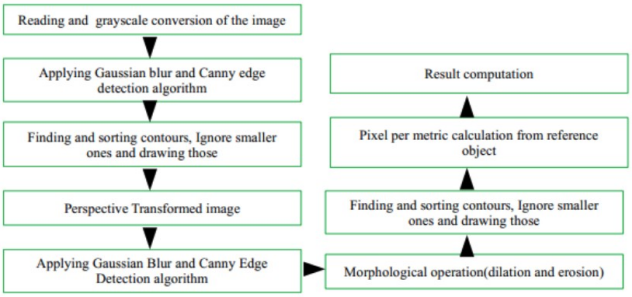

Figure 3: Processes of proposed system

ready for contour detection by the Canny edge detection algorithm with two threshold values of 75 and 200. In our inspection, edges marked as sturdy if edge pixels are more significant than the high threshold value 200 and suppress the edges if edge pixels are smaller than the small threshold value 75. As shown in Figure 4, on the left, we have our original image, and on the right side, we have the edges detected in that image. As mentioned earlier, all the targeted objects are on a piece of A4 size paper in the input image to find the contours, the system grabs the most significant contour, which is the piece of paper. Figure 1 shows the most substantial detected edge in the image illustrated by a green rectangle surrounding by the A4 paper (denoted as H). Lastly, the function cv2.getPerspectiveTransform helps to perform warping transformation, as shown in Figure 1 (denoted as X').

Afterwards, a Gaussian filter with 15x15 kernel blurs the perspective transformed image to pass through the Canny edge detector with threshold values of 50 and 70. The canny edge detector aids in recognizing the edges in the image and two morphological operators dilation and erosion with 5x5 square kernel and iteration of 1 perform to close any gaps in the edge map, respectively [5].

B. Object Segmentation

As our system only deals with objects' outlines, a call of cv2.findContours function with a parameter of cv2.RETR_EXTERNAL upholds to find the shapes of the objects in the edge map. Sorting the contours from left-to-right supports to find out the reference object, which is the left-most object in the image, will always be the first entry in the contour list. As shown in Figure 2, a Canadian quarter coin is the reference object.

C. Reference Object

The first object in the sorted contour list is the reference object that helps to calibrate the camera and describe the number of captured pixels. The pixel number provides the approximate number of pixels per every unit measurement

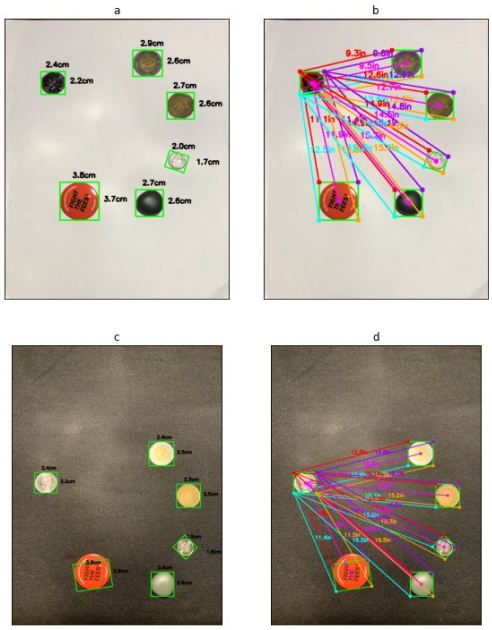

Figure 4: a and b display the measurement of objects and distances from the reference object with perspective transformation. In comparison, c and d illustrate the measurement of objects and distances from the reference object without a perspective transformation.

of its width, which is known as pixel per metric. We can express pixels per metric as follows:
pixel per metric = width of the object in pixel / actual width of the object.

D. Computation of Results

Next, our model scans through the contour list cancel out insignificant contour presuming to be noise and computes rotated bounding box by cv2.boxPoints method. After ordering the bounding box from top-left to bottom-left point, it calculates the set of midpoints of the objects using the following equation from those points [6].

$$M = \left(\frac{x_1 + x_1}{2}\right), \left(\frac{y_1 + y_1}{2}\right)$$

Then an object's height and width are calculated from the division of Euclidean distance between the set of midpoints of the object and pixels per metric parameter [7]. Figure 4 (a) illustrates the object detection and measurements in cm. Lastly, our model computes the distance for the bounding box coordinates and the centroid coordinates by dividing the Euclidean distance between the ordered bounding box points of the reference object and the targeted object by pixels per metric. Figure 4 (b) exhibits the distance measurement from the reference object to targeted objects.

IV. EVALUATION

For evaluation, a dataset has been created and available in GitHub with images captured by iPhone XR in different lighting conditions [4]. Python language with OpenCV libraries is the basis of the implementation of the proposed system.

Object Name	AH (cm)	PH (cm)	AW (cm)	PW (cm)	Avg. Accuracy (%)
Quarter Coin	2.4	2.2	2.4	2.4	95.83
Toonie	2.8	2.6	2.8	2.9	94.70
Dime	1.8	1.7	1.8	2	92.22
Loonie	2.7	2.6	2.7	2.7	98.15
Badge	3.7	3.7	3.7	3.8	98.68
Stone	2.5	2.5	2.6	2.7	98.15

Table I: Accuracy calculation of the object measurement system where AH: Actual Height, PH: Predicted Height, AW: Actual Width, PW: Predicted Width.

Object Name	AH (cm)	PH (cm)	AW (cm)	PW (cm)	Avg. Accuracy (%)
Quarter Coin	2.4	2.2	2.4	2.4	95.83
Toonie	2.8	2.5	2.8	2.5	89.29
Dime	1.8	1.8	1.8	1.9	97.37
Loonie	2.7	2.5	2.7	2.5	92.59
Badge	3.7	2.9	3.7	3.8	87.87
Stone	2.5	2.5	2.6	2.4	96.15

Table II: The accuracy calculation of objects' measurement without perspective transformation

We test the system on these images with and without perspective transformation. Figure 4 (a) and (b) display the measurement of objects and distances from the reference object with perspective transformation. In comparison, Figure 4 (c) and (d) illustrate the measurement of objects and distances from the reference object without a perspective transformation.

To determine the accuracy of our model, we measured the actual height and width of the objects such as quarter coin, pen, rubber eraser, petroleum jelly jar in the image and compared the measurement of our system. Table I presents the accuracy of the object measurement system where AH: Actual Height, PH: Predicted Height, AW: Actual Width, PW: Predicted Width. Table II represents the accuracy of objects' measurement without perspective transformation.

Nevertheless, the predicted results are not 100 percent accurate due to seeing angle and lens deformation. It is noticeable from both Table I and Table II that the proposed system gives better accuracy in predicting the dimension of the object in an image with perspective transformation in contrast to without the transformation.

V. CONCLUSION

In this project, we evaluated a way of measuring objects' dimensions and positional distance from a reference object in an image captured by typical smartphone cameras to get a better result. While the system performs well in predicting objects' dimensions, we found that many assumptions are not accurate, even in some cases, far behind the acceptance level. We proposed an objects' dimensioning and distance measurement system with a perspective transformation of the captured image.

We can conclude that the quest for powerful and robust object measurement methods for industrial systems is far from over, and many opportunities exist for further research.

REFERENCES

[1] Nashwan Adnan OTHMAN, Mehmet Umut SALUR, Mehmet KARAKOSE, Ilhan AYDIN "An Embedded Real-Time Object Detection and Measurement of its Size" Conference Paper · September 2018 DOI: 10.1109/IDAP.2018.8620812

[2] W.Hu,T.Tan,L.WangandS.Maybank,"A Surveyon Visual Surveillance of Object Motion and Behaviors",IEEETrans.Syst.,Man,Cybern.C,vol.34,pp.334-352,Aug.2004.

[3] Measuring size of objects in an image with OpenCV (by Adrian Rosebrock on March 28, 2016). Retrieved from:www.pyimagesearch.com/2016/03/28/measuring-size-of-objects-in-an-image-with-opencv/

[4] OpenCV, 2016. opencv.org/.[Accessed December 23, 2016]

[5] Morphological image analysis. Principles and applications by Pierre Soille. ISBN 3-540-65671-5 (1999), 2nd edition (2003)

[6] The Midpoint Formula. Retrieved from: www.purplemath.com/modules/midpoint.htm

[7] Meshram, Sameer. (2016). Re: Anybody is having an idea about how can we measure the real distance between the objects in the Image, if we know the height of the camera position?. Retrieved from: www.researchgate.net/post/Anybody_is_having_an_idea_about_how_can_we_measure_the_real_distance_between_the_objects_in_the_Image_if_we_know_the_height_of_the_camera_position/56fb8d8a404854eba107a363/citation/download.

[8] github.com/souravskr/image_processing_object_detection