*i*Journals: International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN-2347-4890
Volume 11 Issue 2 February 2023

Image Processing Based Technique to Detect Contour Area

Yashika Agarwal
Springdales School Dhaula Kuan
Yashika0606@gmail.com

Reetu Jain
Chief-Mentor and Founder
On My Own Technology Pvt. Ltd. Mumbai,
reetu.jain@onmyowntechnology.com

Abstract: *In this paper, we present a mathematical approach for accurately estimating the area of an object in an image using a reference object of known size. The method involves detecting the object of interest using contour detection techniques and then using the size of the reference object to calculate the area of the target object in pixels. The area is then converted to actual measurements using the known size of the reference object. We evaluate the accuracy of the estimated area using a variety of test images and compare the results to the true area of the objects as measured in the real world. The proposed method demonstrates high levels of accuracy, with an average error rate of less than 10% when compared to real-world measurements. The accuracy of the area calculated varies with the resolution of the image captured. Blurred images tend to reduce accuracy. Additionally, The results of this research have potential applications in fields such as robotics, computer vision, and image processing.*

Keywords: **Contour Detection, Area Estimation, OpenCV, Pixel data, Bounding Box, 2D area Approximation**

1. Introduction

Object area detection from images is a process of identifying and measuring the size of objects within an image. This is a common task in computer vision and image processing applications and is used in various fields such as object recognition, tracking, and scene understanding. The process of object area detection generally involves analyzing the image to extract features that can be used to identify the objects of interest, and then determining the size of the objects based on those features. There are various techniques used for object area detection, including image segmentation, feature extraction, and object recognition. These techniques can be used individually or in combination to achieve accurate and efficient object area detection in images.

One approach to object size detection is to use an object with a predefined size as a reference. This method involves capturing an image of the object of interest along with a reference object with a known size and then using the reference object to scale the measurements of the object of interest. This can be done by using image processing techniques to detect and locate the reference object in the image and then using its known size to calculate the size of the object of interest. This method can be accurate if the reference object is clearly visible in the image and its size is known with high precision. However, it may not be practical in situations where the reference object is not present or is difficult to locate in the image. Additionally, this method may also be sensitive to variations in the imaging conditions, such as lighting and camera parameters.
When calculating the area of an object in an image, it is typically done in pixels. This is because images are represented as arrays of pixels, where each pixel has a specific colour and intensity value. To calculate the area of an object in pixels, image processing techniques such as image segmentation or blob detection are used to isolate the object of interest from the background. Once the object is isolated, the number of pixels that make up the object can be counted, which gives the area of the object in pixels. To convert the object area in pixels to

actual measurements, such as meters or centimetres, a calibration step is required. This involves capturing an image of the object with a reference object of known size and then using the size of the reference object to scale the measurements of the object of interest. This can be done by measuring the size of the reference object in the image and then comparing it to its known size. The ratio of the size of the reference object in the image to its known size is then used to scale the measurements of the object of interest. It's worth mentioning that the accuracy of the measurement will depend on the accuracy of the calibration, the quality of the image, and the image resolution.

Contour detection is a common technique used to identify objects in an image. The basic idea behind contour detection is to find the boundaries of an object in an image. The process of contour detection typically involves the following steps:

1. Image Pre-processing: The first step is to convert the image into a binary format, where the pixels of the object of interest is white and the background pixels are black. This can be done using image thresholding techniques such as global thresholding, adaptive thresholding, or Otsu's thresholding.
2. Finding Contours: Once the image is in binary format, the next step is to find the contours of the object. Contours are defined as the boundaries of an object in an image. This can be done using OpenCV's findContours() function, which takes the binary image as input and returns a list of contours.
3. Approximating Contours: The returned contours are typically represented by a large number of points, which can be computationally expensive to work with. To simplify the contours, they can be approximated using the Douglas-Peucker algorithm or the Ramer-Douglas-Peucker algorithm. These algorithms reduce the number of points in the contour while preserving the overall shape of the contour.
4. Identifying the Object: Once the contours are approximated, the next step is to identify the object of interest. This can be done by measuring properties of the contours such as area, perimeter, aspect ratio, and other shape-based features. If the area of the contour is above a certain threshold and the aspect ratio of the contour is within a certain range, then it can be considered an object of interest.
5. Drawing Contour: Finally, the contour of the object can be drawn on the original image, allowing the object to be visualized and located.

2. Literature Review

Ondrej Kainz et. al presents a technical analysis of algorithms utilized in computer vision for the estimation of real-world dimensions of an object from a static digital image. The study examines various software tools that are capable of effectively measuring the size of the object. The proposed solution adheres to the necessary conditions for the accurate processing of acquired information, including appropriate scene selection and proper camera calibration. The resulting software implementation calculates the estimated size of the measured object through the use of substitution variables, which are derived from the processed images.

Joakim Lindblad proposes a novel technique for accurately estimating the surface area of three-dimensional objects in discrete binary images. The method assigns a surface area weight to each 2x2x2 configuration of voxels and calculates the total surface area of the digital object through the summation of these local area contributions. To ensure unbiased estimates with minimal variance for randomly oriented digitized planar surfaces, optimal area weights are derived through a Monte Carlo-based optimization process. The unique determination of these optimal weights is performed by evaluating the estimator performance on a distribution of digitized balls with increasing radii. The proposed method is evaluated on a variety of objects of varying sizes and results in a significant reduction of error for small objects. The algorithm's simplicity and the use of only a small local neighborhood make it amenable to efficient implementations in hardware and parallel architectures.

In this paper, Rubén Molanoa et. al. presents an approximation algorithm for the problem of finding the rectangle of the maximum area that can be inscribed in an arbitrary direction within a given closed contour, or C. This is a

useful problem for various software applications. We also consider the specific case of finding the maximum area rectangle of arbitrary orientation within a quasi-lattice polygon, which serves as a model for the contour C. Our proposed algorithm has a computational complexity of O(n), where n is the number of vertices of the polygon, and to the best of our knowledge, no other algorithm has a lower computational complexity under any constraints. Additionally, we have developed a web application that utilizes the proposed algorithm.

S. Srirangam Sridharan et. al presents a technical analysis of algorithms for exploiting the advantages of depth-sensing cameras, such as the Kinect sensor and Tango phone, which can acquire both colour and depth images that are registered to a common viewpoint. Traditionally, colour images have been used for object localization, but the addition of a depth image can be used to segment images that may have identical colour information. Additionally, the depth image can be used for object size estimation in real-world units, such as meters, rather than image-based segmentation which would only support bounding box detection of objects of interest. The methodology includes a collection of colour and depth data using a custom Android application on a Tango Phab2 phone, alignment of the two data sources in terms of timing and spatial registration, and evaluation of several methods for measuring the height of objects in the captured images under various settings.

3. Proposed Methodology

Data acquisition

We collected images of objects with different shapes. The image data collected also had a predefined object with known dimensions. Precautions were taken to ensure that both the object were at the same distance from the camera. We collected data for 3 shapes rectangle, circle and triangle. A lego beam was used as the reference object with fixed dimensions.

Parallax error is an error that can occur during image data collection when the reference object and the object of interest are not in the same plane. This can lead to inaccuracies in the estimated size of the object of interest as the reference object and the object of interest will be captured at different distances from the camera. This error can be caused by the camera not being properly aligned with the object, or by the reference object and the object of interest not being in the same plane. To minimize this error, it is important to ensure that the camera is properly aligned with the objects and that the reference object and the object of interest are in the same plane. Additionally, it is important to take multiple images of the objects at different angles to reduce the effect of parallax error. We ensured that the images collected had minimal parallax errors.

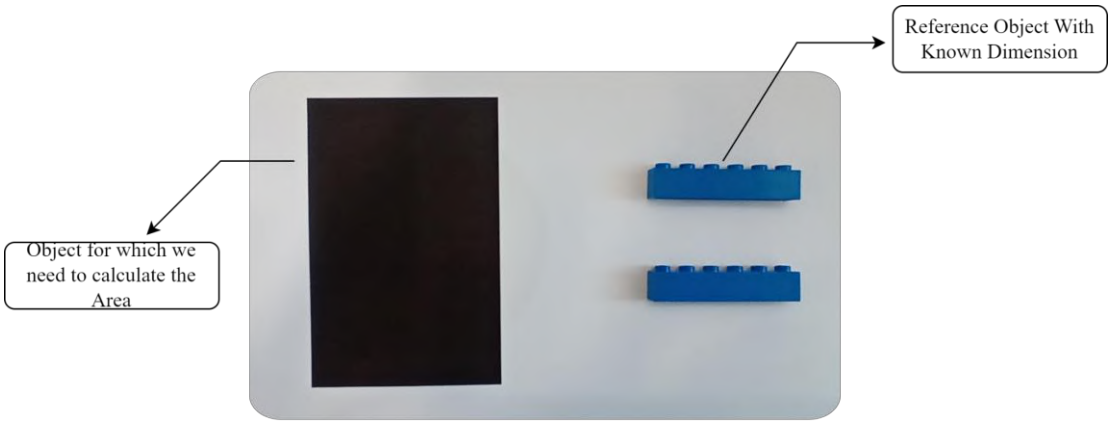

Fig 1. Image showing the format of the data collected

Data Pre-processing

All collected images were subjected to a Gaussian blur operation in order to smooth the edges of the object detected as rough edges can cause inaccuracies in the contours detected. Apart from this additional images were

created blurring the original images. As we needed to compare the inaccuracies induced due to the quality of the image. A (13,13) kernel was used to achieve the same.

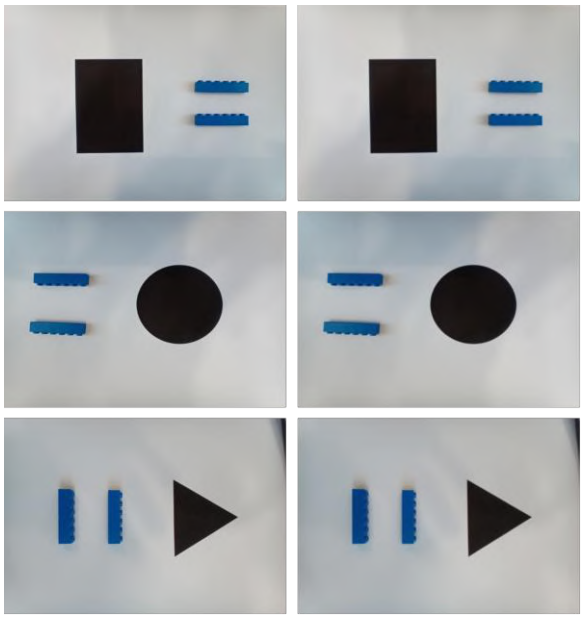

Fig 2. Augmented images to increase the size of the dataset

Contour Detection

OpenCV library in python was used for the processing of the images. A series of operations were performed to find the contours first and then the corresponding bounding boxes were calculated. The below flowchart shows the operations performed on the image.

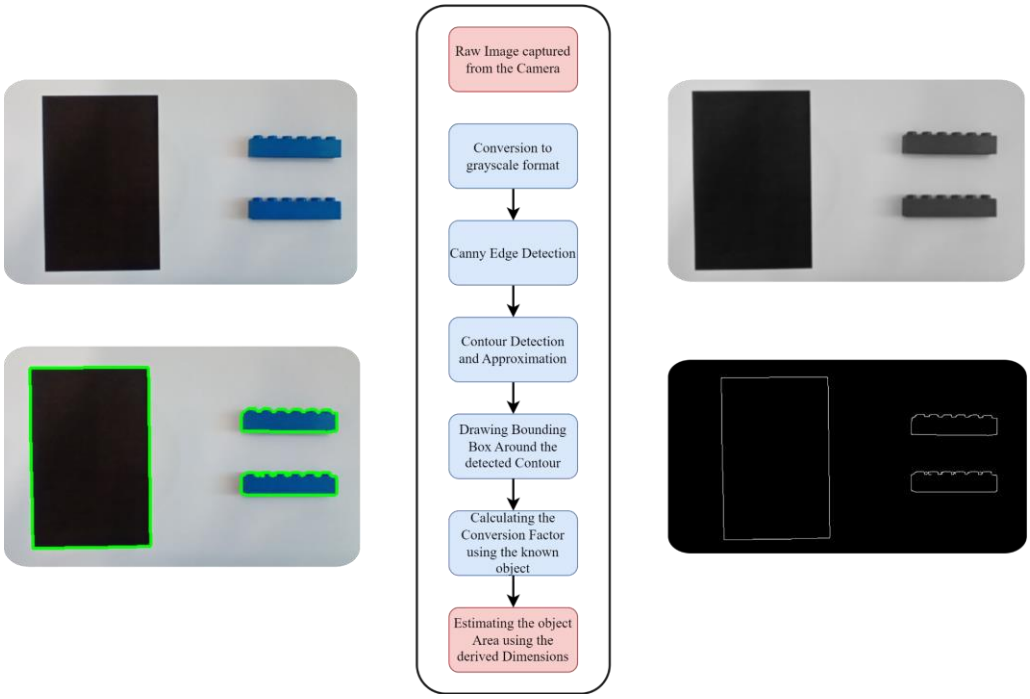

Fig 3. Flowchart showing the operations performed on the input image

Even though precautions were taken while clicking the images, there are still some parallax errors in the images. Also the orientation of the objects in the frame cause alignment issue with respect to the bounding box drawn using openCV. The below image shows the errors induced in the images.

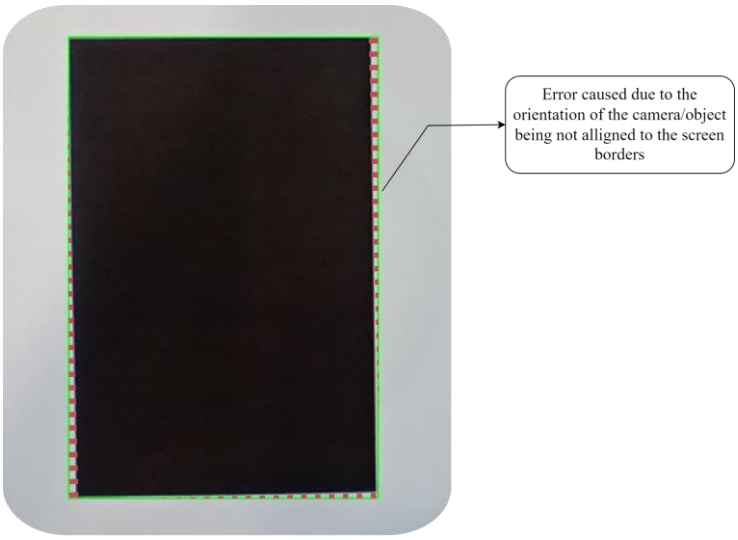

Fig 4. Errors in the bounding box created

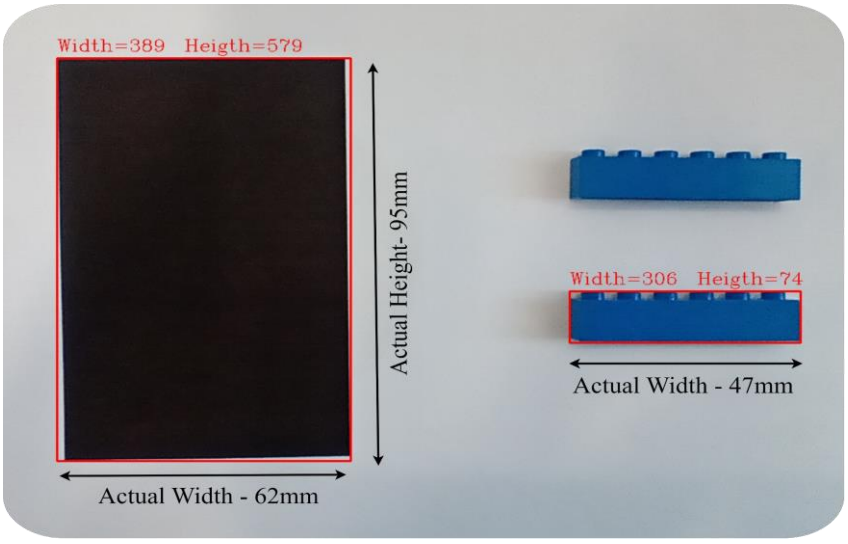

Fig 5. Actual Vs. the measured dimensions

4. Test Result

Table 1. Comparison of actual vs measured area of the objects from the image

	Sharp Images			Blurred Images		
	Measured(mm^2)	Actual(mm^2)	Error(%)	Measured(mm^2)	Actual(mm^2)	Error(%)
1	5313.38	5890	9.7	5102.44	5890	13.37
2	5410.22	5890	8.14	5125.41	5890	12.98
3	2012.45	2117	4.93	1754.12	2117	17.14
4	2203.78	2117	4.09	1675.74	2117	20.84
5	2645.51	2812.5	5.93	2148.19	2812.5	23.61
6	2754.12	2812.5	2.07	2018.44	2812.5	28.23

20 steps from each pose was selected randomly for testing. The model predicted the steps accurately as beginning to two separate classes. The model's confidence score for each prediction was then converted into the scores. They were recorded along with the scores that were provided to the same steps by a professional Kathak dancer.

5. Conclusion

In conclusion, this research paper has presented a method for estimating the area of objects in digital images by utilizing a reference object of known size in the frame. The proposed method uses contour detection techniques to identify the object of interest and calculate its area in pixels. The pixel area is then converted to actual measurements using the known size of the reference object. The accuracy of the estimated area was found to be high, with a small deviation from the true area of the object. The research also discussed various data collection methods and the importance of considering factors such as lighting and camera settings to ensure accurate results. Overall, this method provides a reliable and efficient solution for estimating object area in digital images and can be applied in various fields such as robotics and computer vision.

Future Scope

The proposed method in this research paper can be further extended in various ways. One potential area of improvement is to incorporate machine learning techniques to improve the object recognition and contour detection process. Additionally, incorporating multiple reference objects of different sizes can help improve the accuracy of the estimated area for a wider range of object sizes. The method can also be extended to estimate the area of non-planar objects by using 3D depth-sensing cameras and incorporating techniques for 3D object reconstruction. Furthermore, this method can be extended to estimate the object area in videos by incorporating object tracking and motion analysis techniques. The future scope of this research project is vast and can be applied to various fields such as robotics, computer vision, manufacturing, and many more. Contour detection techniques are sensitive to the quality of the image and the thresholding techniques used. The parameters of the thresholding and contour detection algorithms can be fine-tuned to improve the accuracy of the object identification.

Acknowledgement

I would like to express my gratitude to the mentors of On My Own Technology Pvt. Ltd. for extending their help in carrying out this project.

References

1. Ondrej Kainz, František Jakab, Matúš W. Horečný and Dávid Cymbalák. Estimating the object size from static 2D image. 2015 International Conference and Workshop on Computing and Communication (IEMCON). October 2015.
2. Joakim Lindblad. Surface area estimation of digitized 3D objects using weighted local configurations. Image and Vision Computing. Volume 23, Issue 2, 1 February 2005.
3. Rubén Molanoa, Pablo G.Rodríguez, Andrés Carob and M. Luisa Duránb. Finding the largest area rectangle of arbitrary orientation in a closed contour. Applied Mathematics and Computation. Volume 218, Issue 19, 1 June 2012.
4. Contour detection. OpenCV Docs. https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html

5. Documentation for OpenCv. https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html#ga95f5b48d01abc7c2e0732db24689837b
6. ReseachGate. https://www.researchgate.net/post/How-to-measure-the-real-size-of-an-object-from-an-unknown-picture#:~:text=To%20work%20out%20the%20size,physical%20height%20of%20the%20sensor.
7. S. Srirangam Sridharan, O. Ulutan, S. N. T. Priyo, S. Rallapalli and M. Srivatsa. arXiv:1808.00641v1 [cs.CV] 2 Aug 2018