VIRTUAL OBJECT DIMENSION MEASUREMENT SYSTEM

DESIGN PROJECT REPORT

Submitted by

RAJNANDAN KUMAR (730920201038)

VICKY KUMAR SINGH (730920201052)

RAHUL KUMAR (730920201035)

RAJ LAXMI (730920201036)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

EXCEL ENGINEERING COLLEGE
(AUTONOMOUS)

Komarapalayam - 637303

NOV 2023

EXCEL ENGINEERING COLLEGE(AUTONOMOUS), KOMARAPALAYAM

BONAFIDE CERTIFICATE

Certified that this Design Project report "**VIRTUAL OBJECT DIMENSION MEASUREMENT SYSTEM"** is the bonafide work of **"RAJNANADAN KUMAR (730920201038), VICKY KUMAR SINGH (730920201052), RAHUL KUMAR (730920201035), RAJ LAXMI(730920201036)"** who carried out the Design Project work under my supervision.

SIGNATURE
Prof. P. JAYAPRABHA, M.E.,
HEAD OF THE DEPARTAMENT,
Department of Artificial Intelligence and Data Science,
Excel Engineering College,
Komarapalayam, 637303.

SIGNATURE
Mrs. S.L. SWARNA, M.TECH.,
ASSISTANT PROFESSOR,
Department of Artificial Intelligence and Data Science,
Excel Engineering College,
Komarapalayam, 637303.

The Design Project report submitted for the viva voce held on …….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude in-depth to our institution "**EXCEL ENGINEERING COLLEGE"** for providing us the opportunity to do the Design Project.

We are greatly indebted to our Honorable Chairman **Dr. A.K. NATESAN, M.Com., MBA (NIT)., M.Phil., Ph.D., FTA,** for providing all the necessary facilities for successful completion for the Design Project.

We Express our heartiest gratitude and thanks to our respected Vice Chairman **Dr. N. MATHAN KARTHICK, M.B.B.S., MISTE., PHIE.,** of **EXCEL GROUP OF INSTITUTIONS** for providing all the facilities for successfully completing the Design Project.

We like to place on record to our beloved Principal and Executive Director **Dr. K. BOMMANNA RAJA, M.E., Ph.D.,** for his valuable suggestion in our entire endeavour.

We thank the Head of the Department **Prof. P. JAYAPRABHA, M.E.**, of Artificial Intelligence and Data Science for her guidance, constant inspiration and encouragement.

We wish to express our Heartfelt Thanks and sincere acknowledgment to our respected **Mrs. S.L. SWARNA, M.TECH,** for her encouragement and dedicated guidance.

We take the privilege to record our everlasting and loving thanks to our parents for their kind help and support which render in bringing our Project full manner.

ABSTRACT

The Virtual Object Dimension Measurement System is a cutting-edge technological solution designed to revolutionize the way we measure and analyse the dimensions of objects in a virtual environment. This innovative project harnesses the power of computer vision, augmented reality, and machine learning to provide precise and efficient measurements of real-world objects. The primary objective of this project is to create a system that can detect objects and accurately measure the dimensions of objects in a virtual space, allowing computer model to obtain critical information about the size, shape, and spatial relationships of objects.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

The Virtual Object Dimension Measurement system in an increasingly digital and interconnected world, the need for precise and efficient measurement of physical objects is paramount. Whether in fields like architecture, engineering, manufacturing, or everyday tasks like home improvement, accurate dimensions are crucial. The Virtual Object Dimension Measurement System project emerges as a groundbreaking solution at the intersection of technology and measurement, offering an innovative way to bridge the gap between the physical and digital realms.

Traditionally, measuring the dimensions of real-world objects has relied on physical tools such as rulers, tape measures, and calipers, often demanding a physical presence and time-consuming manual processes but at other side while the computer model or autonomous system have to know the dimension of object which are virtually present in image , it is critical to know about the dimesion of virtual object to a automatic system or robot . The Virtual Object Dimension Measurement System seeks to revolutionize this paradigm by leveraging cutting-edge technologies, including computer vision, augmented reality, and machine learning, to provide accurate and efficient measurements of objects in a virtual environment.

This project's primary goal is to create a seamless and user-friendly system that empowers individuals, professionals, and industries to obtain precise measurements without the need for physical contact with the objects being measured. By harnessing the capabilities of augmented reality, the system overlays virtual measurement markers on real-world objects in real time, providing users with an immersive and interactive measuring experience.

Key Points of the Virtual Object Dimension Measurement System:-

1. **Computer Vision and Augmented Reality**: The project relies on computer vision techniques to recognize and locate physical objects within the user's environment. Augmented reality then superimposes digital measurement tools on these objects, enabling users to measure dimensions and distances with unprecedented ease and accuracy.
2. **User-Friendly Interface**: Accessibility is a core focus of this system. It offers an intuitive interface through smartphones, smart glasses, or other AR-enabled devices, ensuring that users of varying backgrounds and expertise can interact with the technology effortlessly.
3. **Precise and Accurate Measurements**: The project integrates advanced algorithms and machine learning models to guarantee highly accurate measurements. Users can confidently measure dimensions of various element present in one image (it can be also improved for real time scenario).
4. **Scalability**: The Virtual Object Dimension Measurement System is designed to cater to a wide range of applications. It can be used for home improvement projects, architectural designs, engineering tasks, quality control in manufacturing, and various other scenarios, offering a versatile and adaptable solution.
5. **Data Visualization and Export**: Users can visualize and record measurement data in different formats, and the system provides the option to export results for further analysis or integration with other software tools.

1.1 Scope of project

The scope of the Virtual Object Dimension Measurement System project is comprehensive and encompasses various domains and applications. This innovative system aims to bridge the gap between the physical world and the digital realm, offering precise and efficient measurement capabilities. The project's scope includes, but is not limited to, the following aspects:-

Architecture and Engineering, Construction, Interior Design and Home Improvement, Manufacturing and Quality Control, Healthcare, Education and Training, Art and Design…etc.

1.2 Objective of the Project

The Primary objective of "Virtual Object Dimension Measurement Object" to detect the object in image and measure the size or dimension of the various object . This is mainly developed for virtual world where the computer model have to know about the dimension of objects present in image or it can be also built for real time scenario.

CHAPTER 2

LITERATURE SURVEY

2.1 Computer Vision and Augmented Reality

Literature in computer vision and augmented reality provides the foundational concepts and technologies required for object recognition and overlaying virtual measurements onto physical objects. Nadim Mahmud, Jonah Cohen, Kleovoulos Tsourides, Tyler M. Berzin. Gastroenterology Report, Volume 3, Issue 3, August 2015, Pages 179–184

2.2 Augmented Reality Measurement Apps

Research on augmented reality measurement apps for smartphones and smart glasses offers insights into existing applications that provide measurements using augmented reality.

2.3 Machine Learning and Object Recognition

Machine learning algorithms for object recognition and feature extraction are crucial for accurate measurements. Studies on object detection and pose estimation are pertinent. It's invented Y. KODRATOFF and S. MOSCATELLI.

2.4 3D Scanning and Modelling

Research on 3D scanning technologies and software tools for creating virtual models of physical objects is relevant for understanding the digitization process. Ertu Unver, Atkinson Paul &Tancock Dave , Pages 41-48 , Published on 05 Aug 2013.

2.5 Human-Computer Interaction (HCI) : HCI research that explores user interfaces and user experience design in augmented reality and virtual environments is essential for creating an intuitive system.

2.6 Spatial Computing and Mixed Reality

Literature on spatial computing and mixed reality provides insights into the seamless integration of physical and digital environments for measurement.

2.7 Applications in Architecture and Engineering

Case studies and research on the use of augmented reality and virtual measurement in architectural and engineering projects offer practical insights. Hung-Lin Chi , Shih-Chung Kang , Xiangyu Wang , Accepted 29 December 2012, Available online 22 January 2013.

2.8 Quality Control in Manufacturing

Studies on the application of augmented reality and computer vision in manufacturing processes for quality control and measurements.

2.9 Medical Augmented Reality

Research on medical applications of augmented reality, including surgical guidance and anatomical measurements, is relevant.

2.10 Educational Technology

Literature on the integration of augmented reality and virtual measurement tools in educational settings for STEM (Science, Technology, Engineering, and Mathematics) subjects.

2.11 Art and Design Tools

Studies on tools and technologies used by artists, designers, and sculptors that incorporate augmented reality and virtual measurements.

2.12 Data Visualization and Export

Research on data visualization techniques and methods for exporting measurement data for further analysis.

2.13 Accessibility Technologies

Studies on technologies designed to improve accessibility for individuals with disabilities, particularly in measurement and spatial understanding.

2.14 Inventory Management and Logistics

Research on using augmented reality for inventory management and logistics, including case studies on warehouse applications.

2.15 Remote Collaboration and Communication

Literature on technologies that facilitate remote collaboration and communication, especially in the context of augmented reality and measurements.

2.16 Security and Privacy in AR

Understanding the challenges and solutions related to security and privacy when using augmented reality in measurement applications.

2.17 Challenges and Future Directions

Reviewing the current challenges in virtual object dimension measurement and identifying potential research directions for further improvement.

2.18 Calculating the Width of the object

Camval is referred as the height from camera to image while capturing the picture and x is referred as the width parameter of an object (pixels). The x value indicates the operation given in the equ(1). By analysis ofdata, y and z values depends on the ratio camval parameter to the object's physical width. yvalue can be acquired from the equ(2), z value can be acquired from an equ(3).

$$X=y*K+z \quad (1)$$

$$Y=m1*\frac{camval}{widRc\ al} - b1 \quad (2)$$

$$z = m2 * \frac{camval}{widRc\ al\ -b} \quad (3)$$

For finding the estimated width value we have to find the values of the equ((1),(2),(3)). P is referred as the estimated width.

$$P = camval * \frac{x+b1*K+b2}{m1 * K+m2} \quad (4)$$

2.19 Calculating the Height of the object

Suppose consider a reference model camval value is as same as a theoretical model having the same height. Q is referred as the reference model and indicates the equ(5).

$$Q = m1 * K + m2 * camval + b2 \quad (5)$$

Herethe height parameter of the reference model can be obtained from equ(5) and indicated by Q. where K is the parameter for values of distance in pixels of the lower border of displayed area and displayed object, camval is parameter that can be obtained from equ(1) and b2 = 203.7, m2 = 0.06892, m1 = -1.028 are the constant values that are acquired from the data analysis.

M is referred as the the original height of the calculated object. The main goal is to acquire an estimated height parameter as equally close to the M value. Distance between real0 and real2 is referred as reference model's physical height. The distance is the cam value, i.e., equal as the height parameter of the camera. M is the parameter value indicates original object's displayed height(pixels) and N refers to reference model's displayed height. In case of Logitech C920 HD Pro Webcam camera, gamma value is the twice the value of vertical angle of view. The aim is to find estimated height value from the equ(6), that will be possibly equal to M value. Camvalvalue is the height from camera to picture while capturing the pictures, with possible values of K coefficient is -1 or 1. If Q, is greater than the pixHeig, then K value is -1, otherwise K value is 1. Based on basic fundamental trigonometry rules, one can find the value of distance between real1 and real2 from the equ(6) using basic trigonometry rules. The ρ

value can be to obtain from equ(7). y value is the substraction between angles values γ and δ, equ(8). δ value is possible to obtain from equ(9).

$$estHeig=camval+k*\tan(p)*D \quad (6)$$

$$p = \frac{pixHeig - Q}{Q} * y \quad (7)$$

$$y = \gamma - \delta \quad (8)$$

$$\delta = \frac{K}{K+ N} * \gamma \quad (9)$$

D is referred as the camera's horizontal distance and the computed object, that is possible to obtain from the equ(10). The equ(10) uses the α values acquired from previously used equations.

$$D =\cot(y)*camval \quad (10)$$

2.20 The filtering of shadows and background

The object detection system provides two differentoptions for background and shadow filtering of the object. The primary option is finding the object measurements using RGB model; this solution from the object will be calculted with its shadow. The another option is to remove the shadows formed from the object. The CIELab was selected for this work, after so many tests of different colormodels. By usingthe CIELab color model, it was found that it reduces the shadows formed by the objects in the photo.If the object is white, or in a shades of grey, or black, the CIELab model usage will be a drawback.

Fig. 2.19 Sample Picture

If the picture of an object is clicked in a situation where the shadows of the object will not affect the edge recognization in an object, so, the measurement of an object can be done without the need of shadows filtering. The detection is difficult when the object has a several areas with significantly various colors. In that type of cases, only any one of the areas with color is detected as the object.

CHAPTER 3

EXISTING SYSTEM

There are various system are available for object size measurement , that are use for realtime physical object size measurement. Some are :-

3.1 Augmented Reality (AR) Measurement Apps

Various mobile apps and software, such as AR Ruler, AR Measure, and ARKit/ARCore-based apps, leverage smartphone cameras and AR technology to measure distance and dimensions of real-world objects.

3.2 Laser Scanning and LiDAR

Laser scanning and LiDAR (Light Detection and Ranging) systems are used for 3D scanning and precise measurements in fields like surveying, construction, and archaeology. Teemu Hakala, Juha Suomalainen, Sanna Kaasalainen, and Yuwei Chen , 26 March , 2012 | PP. 6851-8218.

3.3 3D Scanning and Modeling Software

Software tools like Autodesk ReCap and Trimble RealWorks enable the creation of 3D models from scanned objects, which can be used for dimensional analysis.

3.4 Smart Glasses and Head-Mounted Displays (HMDs)

Devices like Microsoft HoloLens and Google Glass offer augmented reality capabilities, allowing users to view virtual measurements in their field of vision. Fabien Lareyre, Arindam Chaudhuri, August 2021, Pages 497-512.

3.5 Industrial Measurement Systems

Industries like manufacturing and quality control use specialized measurement systems that incorporate vision technology and sensors for precise measurements.

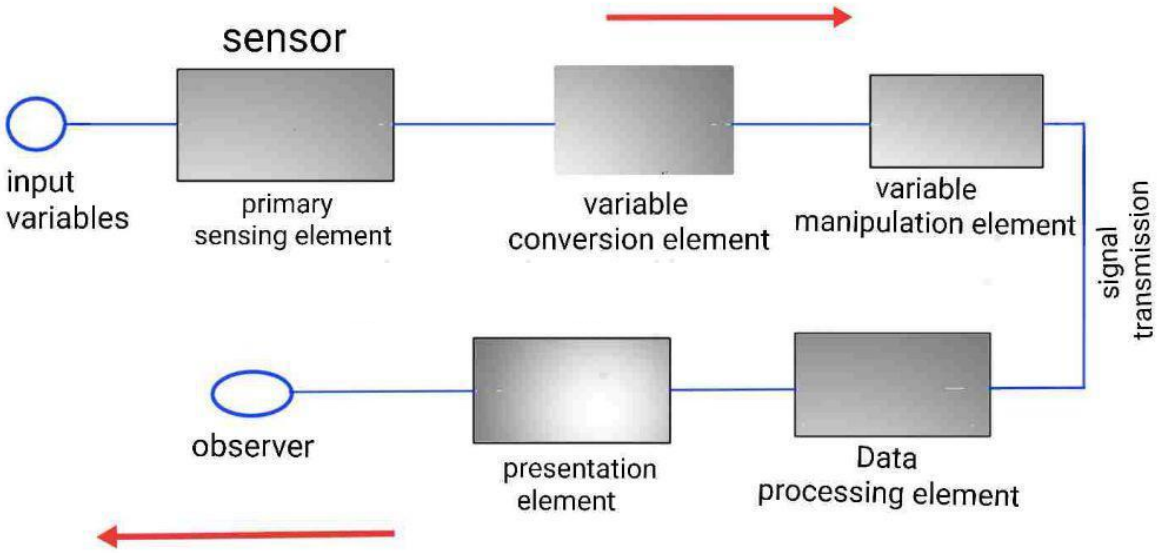


Fig. 3.5 : Industrial Measurement Systems diagram

3.6 Gaming and Entertainment

Virtual reality (VR) and augmented reality are utilized in gaming and entertainment for interactive experiences, offering insights into immersive digital interactions with objects. Eugene Martin Christiansen, Julie Brinkerhoff-Jacobs, April 1995, Pages 6, 79-94.

3.7 Remote Collaboration Tools

Collaboration platforms like Microsoft Teams and Zoom incorporate AR features that enable remote participants to share and interact with virtual content. Gheorghe H. Popescu, Florin Cristian Ciurlău, Cristian Ionuț Stan, Cecilia Băcănoiu (Văduva), Alina Tănase (Veisa), October 2022, page range : 21-34.

CHAPTER 4
PROPOSED WORK

4.1 Introduction

In this paper, we propose an approach that uses the video or image of the surroundings captured by the computer's webcam or external camera as an input to identify items, measure their dimensions, and identify them.With a stand, we are able to keep the camera at a specific distance while also adjusting the camera's height, width, and depth. With the use of this system, we are able to identify many objects at once, provide their dimensional measurements, and obtain other information such as the object's area of occupancy. The system is created using a python program which makes use of python libraries for queue, math, numpy, and computer vision (opencv-cv2). This system is composed of three modules(a,b and c) that act in accordance with various system-related tasks and are discussed below sequentially.

Module a - the first module is used for obtaining the video frames input, in this module we configure our environment by setting the camera width, height and also we set the frame rate of the video input.

Module b - the second module is used to detect object by capturing the objects boundaries. There are multiple functions performed in this module. Initially the input frame to gray scale to get better understanding of the details in input. Then we apply Gaussian blurring this methodology substitutes a Gaussian kernel for a box filter. The cv.GaussianBlur() function has completed its work.The kernel's width and height should be positive and odd, accordingly. Furthermore, we must supply sigmaX and sigmaY, which stand for the standard deviations in the X and Y axes, respectively. sigmaY is assumed to be the same as sigmaX if only sigmaX is supplied. If either or both are provided as zeros, values are chosen based on the kernel size. In order to exclude Gaussian noise from an image, Gaussian blurring

is highly effective.Then image thresholdinng cv.threshold() is applied, the thresholding is achieved through using procedure cv.threshold. The provided image, which must be in grayscale, is the first argument. The threshold value, which is used to classify the pixel values, is the second input. The following input determines the highest value that will be given to pixel values that are higher than the threshold. The function's final parameter, which is obtained using OpenCV, supports multiple types of thresholding. Then contours are detected three choices exist when employing the cv.findContours() function: source picture, contour retrieval mode, and contour approximation method. And the hierarchy and contours are produced. List of every contour in the image is identified as contours.The (x,y) coordinates of an object's border points are arrayed for each contour.

Module c - the third module is used to find the object dimensions by using the contour data we find the various lengths of the object from the math library we use hypot method which is used to calculate euclidean distance, there by allowing us to calculate length,breadth, and also also area of the object.

web sources, and for other things. We will send the request and through the API we will get the corresponding output.

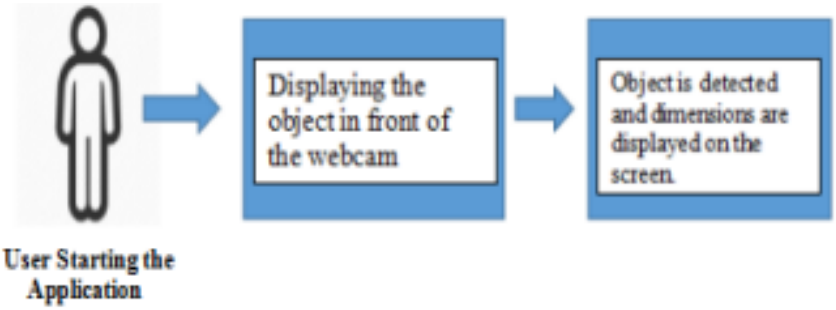


Fig. 4.1 : Block diagram of proposed approach

4.2 System Architecture Design

Define the overall architecture of the system, including the integration of computer vision, augmented reality, Machine learning, and data processing components.

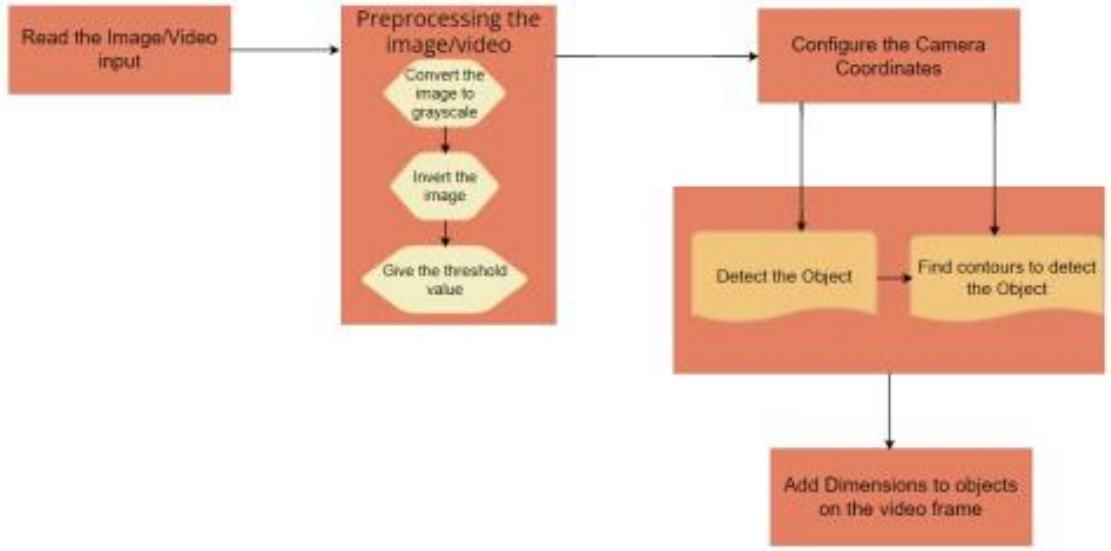


Fig. 4.2 : System Architecture Design

4.3 Object Recognition and Tracking

Develop algorithms for object recognition and tracking using computer vision to identify and locate physical objects within the user's environment**.**

4.4 Object detection and identification from stereo images

The proposed measurement system consists of object detection on the stereo images and blob extraction and distance and size calculation and object identification. The system also employs a fast algorithm so that the measurement can be done in real-time. The object measurement using stereo camera is better than object detection using a single camera that was proposed in many previous research works. It is much easier to calibrate and can produce a more result.

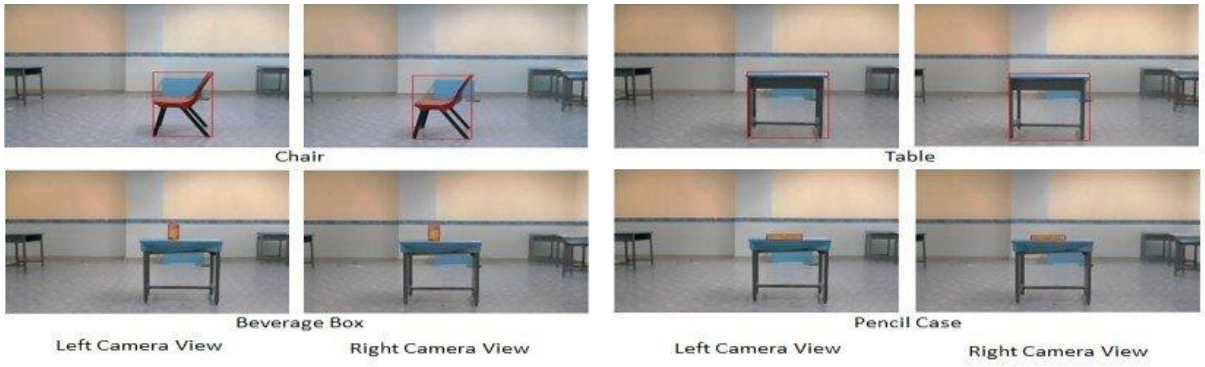


Fig. 4.4 : Object detection and identification from stereo images

4.5 Stereo Object Size and Distance Measurement

The flow of the object size measurement system that we proposed started with stereo vision image capture. Then, on both images, a pre-processing will be applied followed by object detection and segmentation. Finally, the object distance and size will be calculated. From the object size, object identification can be done.

Measured Distance (m)	Actual Distance (m)	Error (m)
3.101	3.000	-0.101
3.088		-0.088
2.901		0.099
3.112		-0.112
5.203	5.000	-0.203
4.880		0.120
5.204		-0.204
5.199		-0.199
7.016	7.000	-0.016
6.880		0.120
7.250		-0.250
6.810		0.190
9.917	10.000	0.083
9.780		0.220
10.211		-0.211
10.180		-0.180

Table 1 : Object Distance Measurement Result

4.6 Stereo Image Capture

Stereo image capture is done by using two video cameras which are aligned in parallel in fixed position. Both cameras are calibrated so that have matching image properties such as the size , colour space and lighting and corrected from distortion. The object of interest can be measured for its distance and size when it enters the overlapping view of the two cameras.

4.7 Pre-Processing

After the images are captured, the size of the image is down-scaled to quarter of its original size to improve the computation speed. For the proposed system , the original size which is 1280x720 pixels is downscaled to 640x360 pixels. This reduction of size does not affect the accuracy of the system since the size of the object of interest in original image is bigger than 2x2 pixels.

The downscaling applied earlier help to reduce the noise in the image since it averages a group of pixels together which inherently produce a smoothing effect. We further remove the souse by applying a median filtering on the images. Median filter is selected since it is fast and able to produce sufficient smoothing.

Object	Measured Width (cm)	Actual Width (cm)	Error (cm)
Chair	52.065	51.000	-1.065
	52.970		-2.970
	49.830		1.170
	49.571		1.429
Beverage Box	9.612	9.000	-0.612
	10.510		-1.510
	7.786		1.214
	8.901		0.099
Table	67.860	68.000	0.140
	69.015		-1.015
	69.652		-1.652
	66.083		1.917
Pencil Box	31.250	30.000	-1.250
	32.208		-2.208
	28.786		1.214
	29.550		0.450

Table 2 : Object width measurement results

4.8 Object Detection

Different most of the other stereo vision works the stereo matching process that we utilizes is only applied to the object of interest which can be detected using an object detection algorithm. This approach is much faster compared to the stereo matching using features similarity in the two images. In our proposed system, the object of interest is basically a foreign moving object that enters the view of the stereo vision system. We assume that the cameras of the system are always fixed to one position. Hence, the detection of the object of interest is done using two operations, the pixel to pixel background subtraction and thresholding.

The background subtraction is done by taking the difference of every pixel, Ir in the image to its respective reference pixels in the background model, IgG. The difference value is then thresholded with a value, TR to determine if the pixel belongs to the object of interest or not. If the pixel does not belong to the object of interest, it will be used to update the background model.

$$IT = \text{object if, } |IT - IBG| > TR$$

Background model is initialized by assuming that the first frame in the video is the background. The background model is updated by averaging the new intensity with the background model intensity value. An on-line cumulative average is computed to be a new background as:

$$\mu T = \alpha I + (1 - \alpha)\, \mu T\text{-}1$$

where T is the model intensity, I is the intensity and a is update weight. The value for a is set to be 0.5.

4.9 Object Segmentation

The binary image resulting from the background subtraction and thresholding stage is then processed for object of interest segmentation. Firstly, a quick morphology is applied on the binary image to improve the subtraction

result. A sequence of erode and dilate operation are involve in the morphology where the effect is to remove smaller detected regions usually due to noise and to enlarge the areas of object of interests and to close any holes within them We then applied a connected component analysis on the image to segment the object of interests on the image. Connected component analysis locates separated regions in the binary image and labels them as different objects. A one pass connected component analysis is applied to improve the speed of the system. From the connected component analysis results, blob extraction is done by drawing the bounding box around every object of interests.

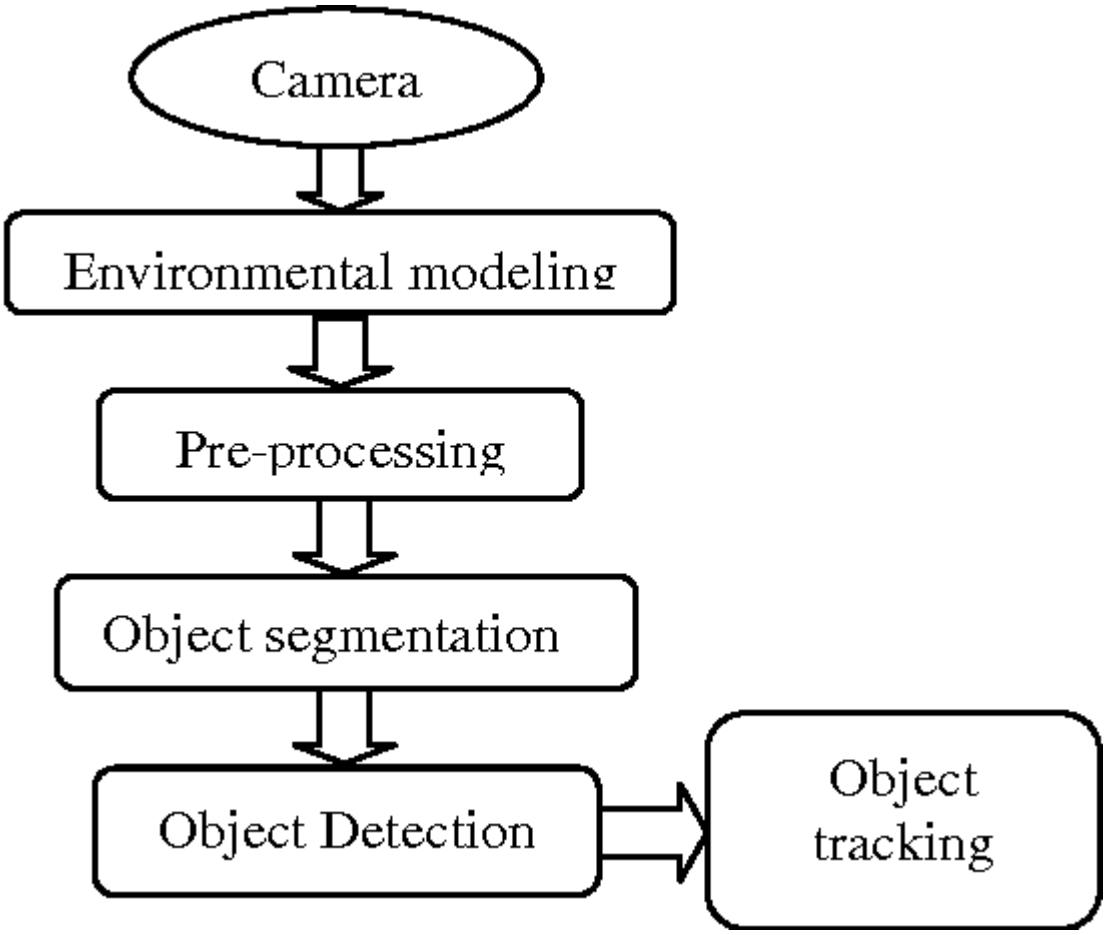


Fig. 4.9 : Object detection and recognition processing

4.10 Object Distance and Size Measurement

Up to this point, all the processing was done in parallel on two images captured from the two cameras. Distance measurement and size calculation use

the 1375 information extracted from the object of interest information on both images.

Distance measurement is done by using the using the disparity value of the object of interest in the two image, din pixel. Since the camera is aligned in parallel, we can simply take the pixel difference between the two width centre lines of the object of interest as the disparity.

The distance, D of the object can be calculated using the following equation:

$$D = \beta\, d\text{-}1 \quad \text{where} \quad \beta = b\, f$$

where b is the separation distance between two cameras and f is focal length of the cameras.

For size measurement, which consists of width and height, the calculations are done by using the disparity value in pixel, d and the pixel value of the width, w and the height, h of the blob. The calculations of actual object width, Wand height, Hare done using the relationship of between width per pixel, and height per pixel, 2 to the disparity. From our experiment we found that width per pixel and the height per pixel value of an object are linear and inversely proportional to its disparity. The equation for the width, W and height, H calculation is as the following:

$$W = \lambda w\, w \qquad H = \lambda h\, h$$

λw and λh are obtained from a linear equation of a graph of λw and λh against d as the following:

where m is the gradient of the plotted graph and c is the value of λ at d = 0.

The graphs of λw and λh against d are plotted by experimenting on several samples with known actual width and height. Different disparity value is obtained by varying the distance of the object from the camera.

Object	Measured Height (cm)	Actual Height (cm)	Error (cm)
Chair	80.784	80.600	-0.184
	78.645		1.955
	79.901		0.699
	82.020		-1.420
Beverage Box	20.196	19.000	-1.196
	19.780		-0.780
	17.661		1.339
	19.518		-0.518
Table	77.440	78.000	0.560
	77.480		0.520
	76.298		1.702
	79.519		-1.519
Pencil Box	5.852	5.000	-0.852
	5.198		-0.198
	3.986		1.014
	4.101		0.899

Table 3 : Object height measurement results

4.11 Object Identification

Different objects can be distinguished from each other depending on their sizes. For example, a car will have a different width compared to a pedestrian. For our system object is identified according to their height, h to width, w ratio and the height-width, hw product (blob area).

A database is utilized to store the information and the associated label of known objects. If an object with matching information is found, the object will be identified. When a new object with no matching height to width ratio and height-width product in found, a new label will be instantiated for that object in the database.

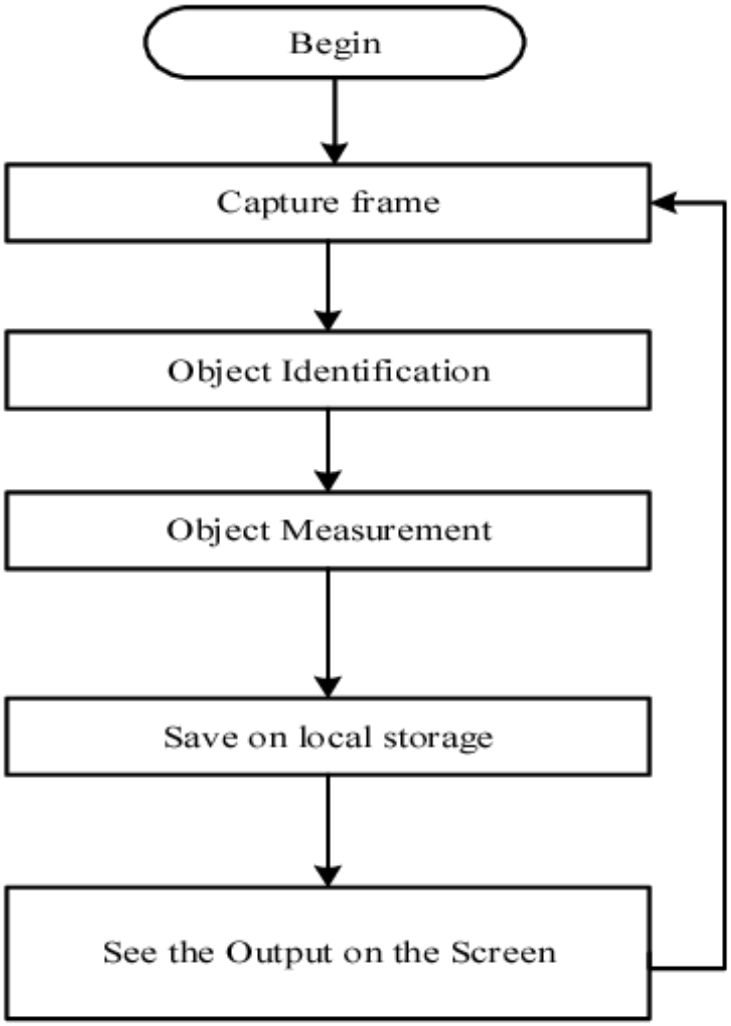
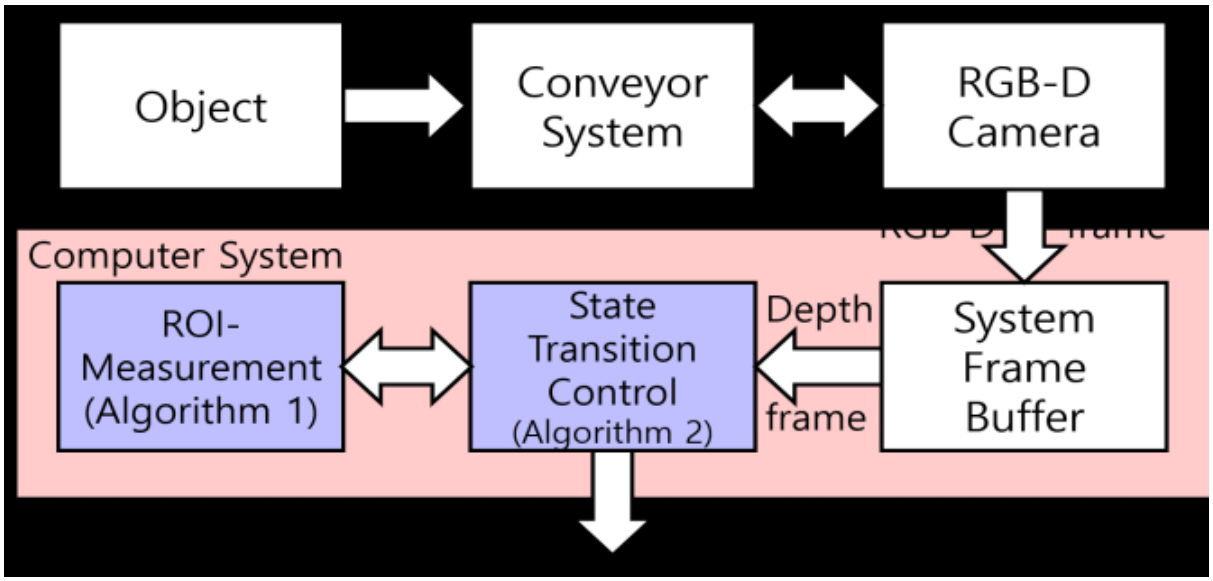


Fig. 4.11 : object detection and measurement of size

4.12 System Flow Diagram



CHAPTER 5

SYSTEM SPECIFICATION

5.1 Hardware Requirement

- **Camera:** The system should support one or more high-resolution cameras with the capability to capture clear images and videos of objects.
- **Processing Unit**: A powerful processor is required to handle image processing, object recognition, and measurement calculations efficiently.
- **Memory:** Sufficient RAM and storage space for storing captured data, processing images, and running the application smoothly.
- **Display:** A high-quality display is necessary for presenting measurement data, visual cues, and augmented reality overlays.
- **Sensors:** Depending on the application, additional sensors such as LiDAR, depth sensors, or accelerometers may be required to improve measurements and object recognition.

5.2 Software Requirements

- **Operating System:** Specify the compatible operating systems for the application, which might include Android, iOS, Windows, or others.
- **Computer Vision and Machine Learning Libraries:** Utilize computer vision and machine learning libraries such as OpenCV, TensorFlow, or PyTorch to implement object recognition and measurement algorithms.
- **Augmented Reality Framework:** Utilize AR frameworks like ARKit, ARCore, or other open-source alternatives to create the augmented reality user interface.

- **Measurement Algorithms:** Develop algorithms for measuring dimensions, distances, angles, and other parameters based on recognized objects. These algorithms should be highly accurate.
- **User Interface (UI):** Design an intuitive and user-friendly UI for easy interaction, object selection, and measurement visualization.

5.3 Functional Specifications

- **Object Recognition:** Specify the accuracy and speed of the object recognition process, including the types of objects that can be recognized.
- **Measurement Accuracy:** Define the level of measurement accuracy and precision the system is designed to achieve for different types of objects.
- **Augmented Reality Features:** Describe the augmented reality features, including the overlay of measurement markers, annotation tools, and interactive elements.
- **Data Export:** Specify the formats for exporting measurement data, including options for sharing or saving measurements.

5.4 Performance Requirements

- **Frame Rate:** Define the minimum acceptable frame rate for real-time object recognition and augmented reality overlays.
- **Latency:** Specify the maximum acceptable latency between object recognition and measurement display.
- **Scalability:** Ensure the system can handle a wide range of object sizes, from small objects to large structures.

5.5 Security and Privacy

- **User Data Protection:** Specify measures to protect user data, including images and measurements, and ensure compliance with privacy regulations.

5.6 Accessibility and Inclusivity

- **Accessibility Features:** Describe features that make the system accessible to individuals with disabilities, such as voice commands or screen reader support**.**

5.7 Testing and Validation

- **Testing Scenarios:** Define the testing scenarios, including controlled environments and real-world scenarios, to validate the system's accuracy and performance.

5.8 Documentation and User Support

- **User Manuals:** Provide comprehensive user documentation that explains how to use the system effectively.
- **User Support:** Describe the resources available for users to seek assistance and troubleshoot issues.

5.9 Customization and Integration

- **Customization Options:** Specify how the system can be customized to meet the specific needs of different industries or applications.
- **Integration with Other Systems:** Describe any available APIs or integration capabilities to link the system with other software or hardware.

5.10 Future Enhancements

- **Roadmap:** Outline potential future enhancements and features that can be added to the system in subsequent updates.

CHAPTER 6

IMPLEMENTATION AND RESULTS

6.1 Setup of proposed system

We proposed the system to measure objects in a real time video and pictures. We prepared a few experimental setups to test the correctness of the proposed method. The implement the proposed system has made by the help of Python language. Figure 6. show that the setup of the prepared system. Except the hardware formation, the software's required has installed.

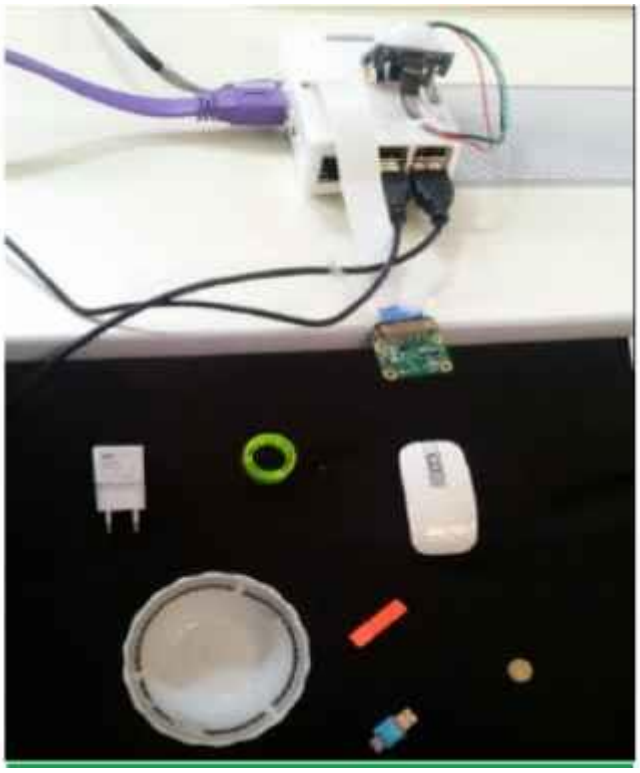


Fig. 6.1 : Setup of proposed system

6.2 Display output on the screen

For the experiment the camera has been effectively. capturing the pictures. The proposed system applies four operations such as record frames, find edges, find objects, and measure size for each object. When we run the application. the output screen displays on the PC screen as appear in below Figure.

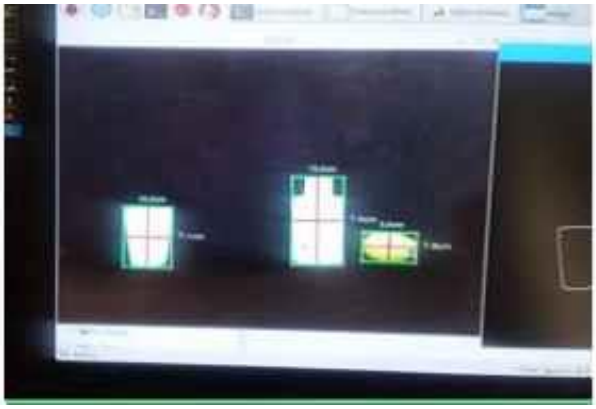


Fig. 6.2: Display output on the screen

6.3 Calculate the size of objects

In below Fig. illustrates the object detection and measurements. The size of each object in the frame which are calculated.

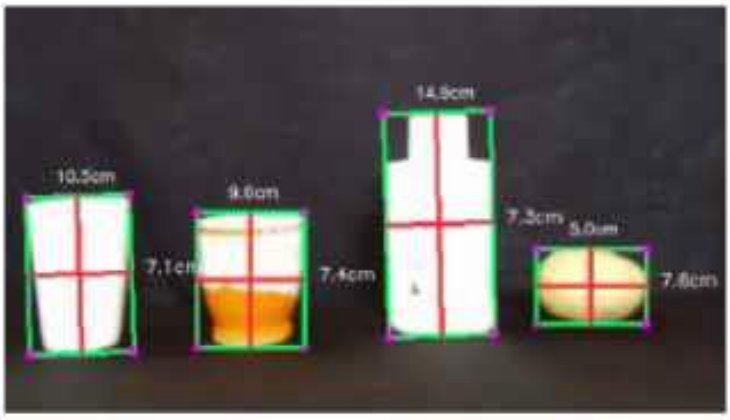


Fig. 6.3 : Calculate the size of objects

In the first experiment we measured size of objects such as white glasses, orange cup, bottle, potatoes. 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.

Name of objects	AM-H (cm)	PM-H (cm)	AM-W (cm)	PM-W (cm)	Accuracy (%)
White glasses	10.0	10.5	6.8	7.1	% 95.45
Orange cup	8.5	9.0	7.5	7.4	%97.56
Bottle	15.2	14.9	7.4	7.3	**% 98.23**
Potatoes	4.8	5.0	7.4	7.6	%96.82

Table 4 : Accuracy object measurement for one frame

6.4 Experimental result when camera above the objects

Nevertheless, not every result are perfect, since this is due to the seeing angle and lens deformation. By calibrate the camera and set good width parameter, accuracy will be increase.

In the second experiment, we set the camera above the objects. Figure 6.4 , shows that the results of the object detection and measurement for another objects. And the Table 6.4. demonstrations the accuracy values among actual measure and system measure.

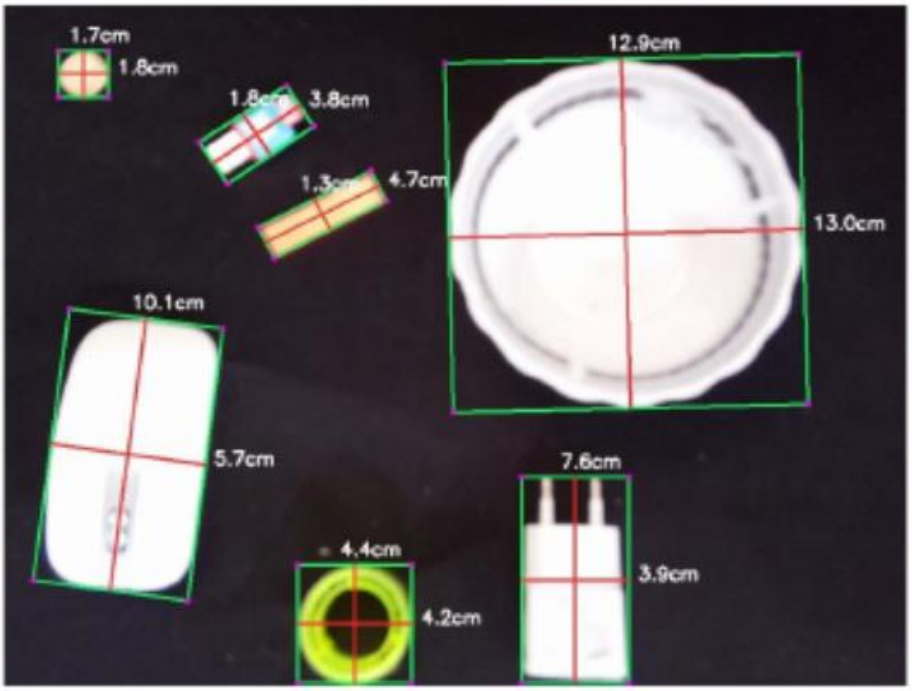


Fig. 6.4 : Experimental result when camera above the objects

Name of objects	AM-H (cm)	PM-H (cm)	AM-W (cm)	PM-W (cm)	Accuracy (%)
Turk kurus	1.7	1.7	1.7	1.8	%97.14
Mouse	10.0	10.1	5.2	5.7	%96.20
Card reader	3.8	3.8	1.7	1.8	%98.21
Peace of paper	4.6	4.7	1.2	1.3	%96.66
tray	12.6	12.9	12.6	13.0	%97.29
Charger	7.6	7.6	3.7	3.9	**%98.26**
plaster	4.4	4.4	4.4	4.2	%97.72

Table 5 : Accuracy object measurement when camera above the objects

The result for error column displays very low errors. The error rate is especially smaller when camera above the objects.

CHAPTER 7
CONCLUSION

In this project, a method for finding the object measurements present in an images using a camera was explained and related execution of software is explained. The identification of an object is based on taking two images in the same scenario without and with the object present in an image. The identified objects is distinguished by the size of object area displayed and the co-ordinates. The output obtained from these images are taken as input data for the equations to calculate the size. Based on the tests conducted, by considering various heights and distances of an object, and using various objects, it was found that the deviation of the measurement from original measurement is less than 7-10%.

APPENDIX – I

CODING

Program

```
# """
# Title: Virtual Object Dimension Measurement and Angle Finder System
# Author:Raj
# Date:11-11-2023
# Type: Design Project(year 2023-2024)
# Language:Python(version - 3.12.0)
# """
# # Import Required Module/Library
from scipy.spatial.distance import euclidean
from imutils import perspective
from imutils import contours
import numpy as np
import imutils
import cv2
from tkinter import filedialog
import tkinter as tk
from tkinter import *
from tkinter import Button
import math


def show_images(images):
    for i, img in enumerate(images):
        cv2.imshow("image_" + str(i), img)
    cv2.waitKey(0)
```

```
        cv2.destroyAllWindows()

# Select Image and Measure Dimension of Detected Object in image
def Measurement():
        img_path =filedialog.askopenfilename()
        # cap=cv2.VideoCapture(0)

        # Read image and preprocess
        image = cv2.imread(img_path)

        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        blur = cv2.GaussianBlur(gray, (9, 9), 0)

        edged = cv2.Canny(blur, 50, 100)
        edged = cv2.dilate(edged, None, iterations=1)
        edged = cv2.erode(edged, None, iterations=1)

        #show_images([blur, edged])

        # Find contours
        cnts = cv2.findContours(edged.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
        cnts = imutils.grab_contours(cnts)

        # Sort contours from left to right as leftmost contour is reference object
        (cnts, _) = contours.sort_contours(cnts)

        # Remove contours which are not large enough
```

```
cnts = [x for x in cnts if cv2.contourArea(x) > 100]

#cv2.drawContours(image, cnts, -1, (0,255,0), 3)

#show_images([image, edged])
#print(len(cnts))

# Reference object dimensions
# Here for reference I have used a 2cm x 2cm square
ref_object = cnts[0]
box = cv2.minAreaRect(ref_object)
box = cv2.boxPoints(box)
box = np.array(box, dtype="int")
box = perspective.order_points(box)
(tl, tr, br, bl) = box
dist_in_pixel = euclidean(tl, tr)
dist_in_cm = 2
pixel_per_cm = dist_in_pixel/dist_in_cm

# Draw remaining contours
for cnt in cnts:
        box = cv2.minAreaRect(cnt)
        box = cv2.boxPoints(box)
        box = np.array(box, dtype="int")
        box = perspective.order_points(box)
        (tl, tr, br, bl) = box
        cv2.drawContours(image, [box.astype("int")], -1, (0, 0, 255), 2)
```

```
            mid_pt_horizontal = (tl[0] + int(abs(tr[0] - tl[0])/2), tl[1] +
int(abs(tr[1] - tl[1])/2))
            mid_pt_verticle = (tr[0] + int(abs(tr[0] - br[0])/2), tr[1] +
int(abs(tr[1] - br[1])/2))
            wid = euclidean(tl, tr)/pixel_per_cm
            ht = euclidean(tr, br)/pixel_per_cm
            cv2.putText(image, "{:.1f}cm".format(wid),
(int(mid_pt_horizontal[0] - 15), int(mid_pt_horizontal[1] - 10)),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 0), 2)
            cv2.putText(image, "{:.1f}cm".format(ht), (int(mid_pt_verticle[0]
+ 10), int(mid_pt_verticle[1])),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 0), 2)

        show_images([image])

# Function for checking Example
# Function to show array of images (intermediate results)
def open_image():
    file_path = filedialog.askopenfilename()
    if file_path:
        img = Image.open(file_path)
        img.thumbnail((300, 300))  # Resize the image for display
        # photo = ImageTk.PhotoImage(img)/

        # label.config(image=photo)
        # label.image = photo

        # Predict what's happening in the image
        # predictions = predict_image(file_path)
```

```
        # result_text.set("Top Predictions:\n")
        # for _, label, probability in predictions:
        #     result_text.set(result_text.get() + f"{label}: {probability *
100:.2f}%\n")
def Existing():
        img_path ="car.png"
        # cap=cv2.VideoCapture(0)

        # Read image and preprocess
        image = cv2.imread(img_path)

        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        blur = cv2.GaussianBlur(gray, (9, 9), 0)

        edged = cv2.Canny(blur, 50, 100)
        edged = cv2.dilate(edged, None, iterations=1)
        edged = cv2.erode(edged, None, iterations=1)

        #show_images([blur, edged])

        # Find contours
        cnts = cv2.findContours(edged.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
        cnts = imutils.grab_contours(cnts)

        # Sort contours from left to right as leftmost contour is reference object
        (cnts, _) = contours.sort_contours(cnts)
```

```
# Remove contours which are not large enough
cnts = [x for x in cnts if cv2.contourArea(x) > 100]

#cv2.drawContours(image, cnts, -1, (0,255,0), 3)

#show_images([image, edged])
#print(len(cnts))

# Reference object dimensions
# Here for reference I have used a 2cm x 2cm square
ref_object = cnts[0]
box = cv2.minAreaRect(ref_object)
box = cv2.boxPoints(box)
box = np.array(box, dtype="int")
box = perspective.order_points(box)
(tl, tr, br, bl) = box
dist_in_pixel = euclidean(tl, tr)
dist_in_cm = 2
pixel_per_cm = dist_in_pixel/dist_in_cm

# Draw remaining contours
for cnt in cnts:
        box = cv2.minAreaRect(cnt)
        box = cv2.boxPoints(box)
        box = np.array(box, dtype="int")
        box = perspective.order_points(box)
        (tl, tr, br, bl) = box
```

```
            cv2.drawContours(image, [box.astype("int")], -1, (0, 0, 255), 2)
            mid_pt_horizontal = (tl[0] + int(abs(tr[0] - tl[0])/2), tl[1] +
int(abs(tr[1] - tl[1])/2))
            mid_pt_verticle = (tr[0] + int(abs(tr[0] - br[0])/2), tr[1] +
int(abs(tr[1] - br[1])/2))
            wid = euclidean(tl, tr)/pixel_per_cm
            ht = euclidean(tr, br)/pixel_per_cm
            cv2.putText(image, "{:.1f}cm".format(wid),
(int(mid_pt_horizontal[0] - 15), int(mid_pt_horizontal[1] - 10)),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 0), 2)
            cv2.putText(image, "{:.1f}cm".format(ht), (int(mid_pt_verticle[0]
+ 10), int(mid_pt_verticle[1])),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 0), 2)

    show_images([image])

# Angle Measurement
def Angle():
    path=filedialog.askopenfilename()
    img=cv2.imread(path)
    pointList=[]
    def mousePoints(event,x,y,flags,params):
        if event==cv2.EVENT_LBUTTONDOWN:
            cv2.circle(img, (x, y), 5, (0, 0, 255), cv2.FILLED)
            pointList.append(([x, y]))

    def gradients(pt1,pt2):
        return (pt2[1]-pt1[1])/(pt2[0]-pt1[0])
```

```
        def getAngle(pointList):
                pt1,pt2,pt3=pointList[-3:]
                n1=gradients(pt1,pt2)
                n2=gradients(pt1,pt3)
                angR=math.atan((n2-n1)/(1+(n2*n1)))
                angD=round(math.degrees(angR))
                # print(angD)
                cv2.putText(img,str(angD),(pt1[0]-40,pt1[1]-
20),cv2.FONT_HERSHEY_SIMPLEX,1.5,(0,0,255),2)
        # img_path = filedialog.askopenfilename()
        # img = cv2.imread(img_path)


        # def Finale():
                # img_path = filedialog.askopenfilename()
                # img = cv2.imread(img_path)
        while True:
                # img_path = filedialog.askopenfilename()
                # img= cv2.imread(img_path)
                if len(pointList)%3==0 and len(pointList)!=0:
                        getAngle(pointList)


                cv2.imshow('Image',img)
                cv2.setMouseCallback('Image',mousePoints)
                if cv2.waitKey(1)& 0xFF==ord('q'):
                        pointList=[]
                        img=cv2.imread(path)
# Initiating GUI
```

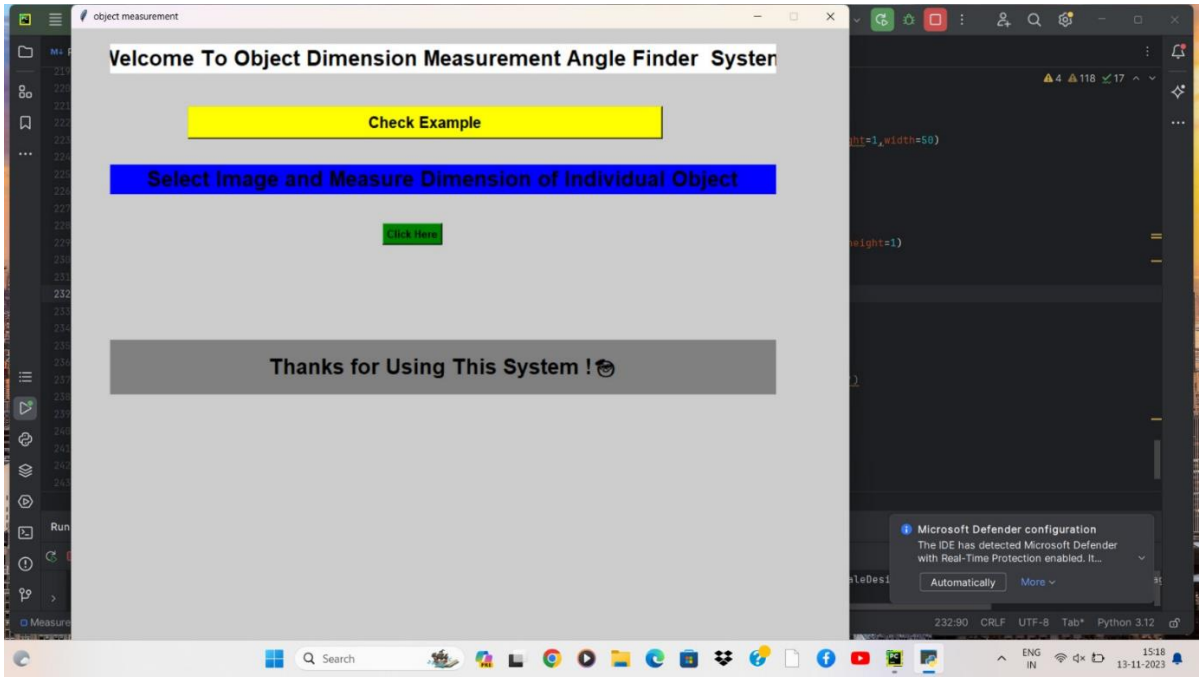
```
window=tk.Tk()

window.geometry("1000x1000")

window.resizable(False,False)

window.configure(background="#CDCDCD")

# bg=PhotoImage(file='example.png')

# canva=Canvas(window,width=1920,height=1080)

# canva.pack(fill="both",expand=True)

# canva.create_image(0,0,image=bg,anchor="nw")

# window.configure()

window.title("object measurement")

l1=Label(window,text="Welcome To Object Dimension Measurement Angle
Finder  System",font=("arial 20 bold"),bg="white",height=1,width=50)

l1.place(x=50,y=20)

# l1.pack()

b=Button(window,text="Check Example",font=("arial 15
bold"),bg="yellow",width=50,height=1,command=Existing)

b.place(x=150,y=100)

# b.pack()

l2=Label(window,text="Select Image and Measure Dimension of Individual
Object",font=("arial 20 bold"),width=50,bg="blue",height=1)

l2.place(x=50,y=175)

# l2.pack()

b1=Button(window,text="Click Here",font=('arial 10
bold'),bg='green',command=Measurement)

# b1.pack()

b1.place(x=400,y=250)

l4=Label(window,text="Thanks for Using This System !□",font=('arial 20
bold'),height=2,width=50,bg='grey')

l4.place(x=50,y=400)
```

```
# l3=Label(text="Click below for Angle Measurement Of Element in
image",fon=('arial 20 bold '),width=50,height=1,bg='blue')

# l3.place(x=50,y=300)

# b2=Button(text="click here",font=('arial 10
bold'),bg='green',command=Angle)

# b2.place(x=400,y=350)

# open_button = tk.Button(window, text="Open Image",
command=open_image)

# open_button.pack()

window.mainloop()
```

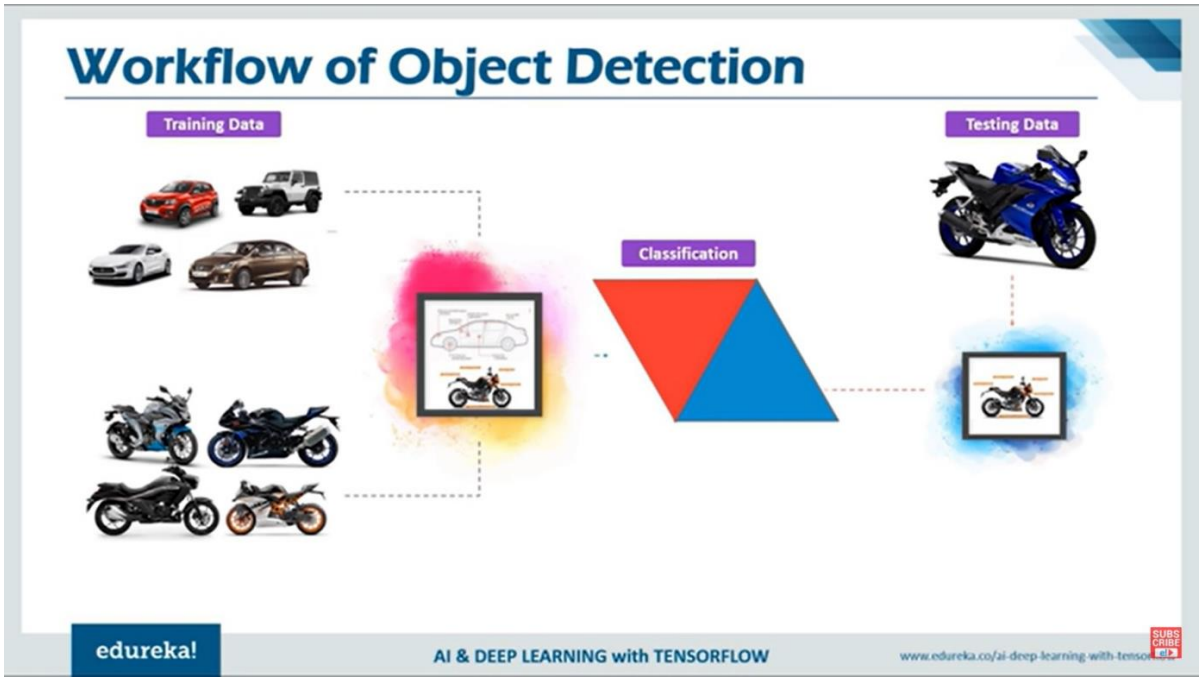
APPENDIX -II

OUTPUT

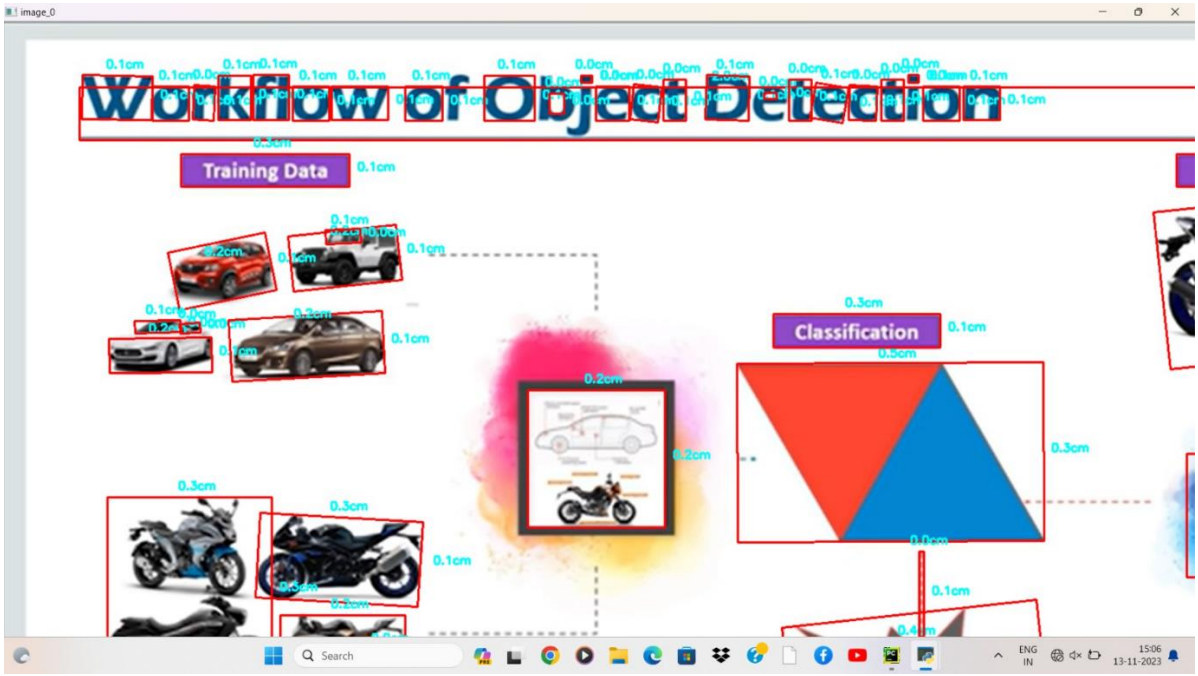
Interface



Input (a)

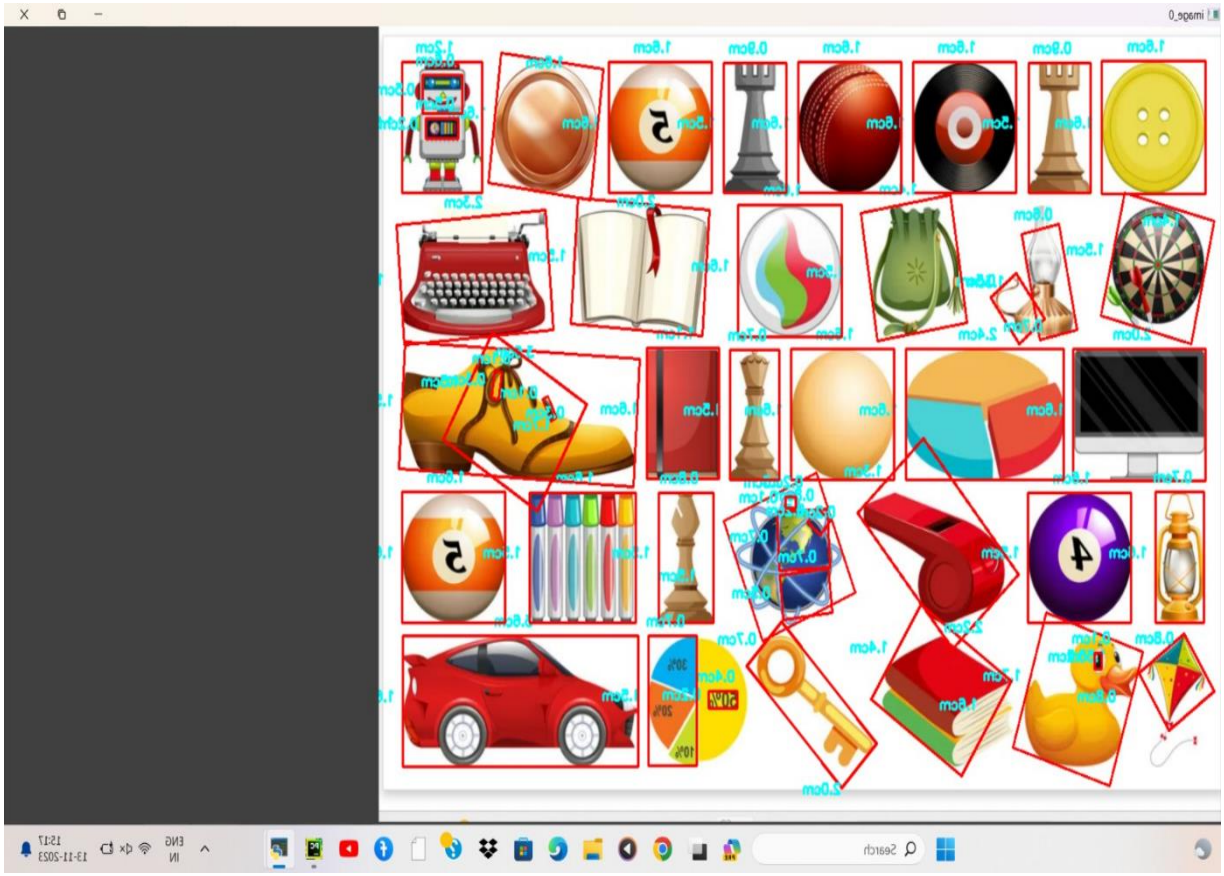


Output



Input (b)

Output



REFERENCE

- "Computer vision and augmented reality" , Nadim Mahmud, Jonah Cohen, Kleovoulos Tsourides, Tyler M. Berzin *Gastroenterology Report*, Volume 3, Issue 3, August 2015, Pages 179–184, https://doi.org/10.1093/gastro/gov027

- "Machine Learning and Object Recognition" , Y. KODRATOFF and S. MOSCATELLI , https://doi.org/10.1142/S0218001494000139

- "3D Scanning and Modelling" , Ertu Unver,Atkinson Paul &Tancock Dave , Pages 41-48 , Published on 05 Aug 2013 , https://doi.org/10.1080/16864360.2006.10738440

- "Applications in Architecture and Engineering" , Hung-Lin Chi , Shih-Chung Kang , Xiangyu Wang , Accepted 29 December 2012, Available online 22 January 2013 , https://doi.org/10.1016/j.autcon.2012.12.017

- "Laser Scanning and LiDAR" , Teemu Hakala, Juha Suomalainen, Sanna Kaasalainen, and Yuwei Chen , 26 March , 2012 | PP. 6851-8218, https://doi.org/10.1364/OE.20.007119

- "Smart Glasses and Head-Mounted Displays (HMDs)" , Fabien Lareyre, Arindam Chaudhuri, Cédric Adam, Marion Carrier, Claude Mialhe, Juliette Raffort, August 2021, Pages 497-512 , https://doi.org/10.1016/j.avsg.2021.02.033

- "Gaming and Entertainment" , Eugene Martin Christiansen, Julie Brinkerhoff-Jacobs, April 1995, Pages 6, 79-94, https://doi.org/10.1016/0010-8804(95)93848-O

- "Remote Collaboration Tools" , Gheorghe H. Popescu, Florin Cristian Ciurlău, Cristian Ionuț Stan, Cecilia Băcănoiu (Văduva), Alina Tănase (Veisa), October 2022, page range : 21-34.

- "Accessibility Technologies" , Mark Grechanik, Qing Xie, Chen Fu, 01-04 April 2009, added IEEE xplore : 26 May 2009.