JJMIE

Volume 17, Number 4, December. 2023
ISSN 1995-6665
Pages 521– 531
https://doi.org/10.59038/jjmie/170407
Editor-in-Chief: Prof. Ali M. Jawarneh

Jordan Journal of Mechanical and Industrial Engineering

Applying Image Processing to Implement Optimization Approaches

Luan Mulaku[a], Fitim Zeqiri[b]

[a]Faculty of Mechatronic Engineering, University for Business and Technology Pristina,10000, Kosovo
[b]Faculty of mechanical and computer engineering, Universityof Mitrovica "Isa Boletini", 40000, Kosovo

Received 25 May 2023 *Accepted 22 Aug 2023*

Abstract

Real-time plays a pivotal role in diverse scientific application, and optimizing the measurement process is essential to enhance efficiency. This study proposes the implementation of geometric methods to optimize real-time measurements through image processing. By evaluating the available space for a mobile robot to move, indirect measurement of the object as an obstacle in real time is achieved. This technique effectively reduces data transmission speed, a critical parameter for real-time measurements, with the potential to lower costs by necessitating smaller and simpler technical specifications for equipment. In order to enhance measurement accuracy, stereo cameras capturing images from various angles are employed simultaneously. However, this may introduce complexities in calculations, such as parallax errors. To address this issue, filters and restoration methods are applied, such as Gauss, to correct noise and accurately measure space. The study presents real-time results for a static object using the dynamic system, which demonstrated an accuracy exceeding 99%. By utilizing the same image for different calculations, the system complexity is minimized, also measurement time is reduced. While current optimization focuses on 2D, there is potential for extension in 3D. The methodologies presented in this study primarily involve analytical approaches such as Distance-Based Optimization and Multicriteria Decision Analytics.

The study's findings and proposed technology pave the way for exciting future research and practical implementations, potentially revolutionizing real-time measurements in numerous scientific and technology domains.

Keywords: capture image; dynamic system; optimization; mocrocontroller; real-time.

1. Introduction

The paper's organization reflects a meticulous adherence to scientific research conventions, providing a clear and logical progression of ideas. The introduction aptly establishes the importance of real-time measurements and the value of optimization in modern applications. It then navigates multifaceted challenges posed by image processing, emphasizing the dual demands of accuracy and optimization.

The methodology section intricately unfolds the proposed geometric optimization methods and elucidates the intricate data analysis technique employed. Parallelly, it provides a comprehensive view of the hardware system and software design concepts. This robust foundation serves as the bedrock for deriving accurate results that are meticulously showcased in the following section. The discussion thoughtfully encapsulates empirical findings and explores their implications, simultaneously outlining the roadmap for future research. Concluding the narrative, the paper underscores its contributions and achievements, foreseeing its potential to resonate across diverse scientific and technological domains. This paper's meticulous organization not only presents insights but also invites readers to embark on an intellectually stimulating journey through the complexities of real-time measurements and optimization.

The focal objectives of this study revolve around the comprehensive exploration and optimization of real-time measurements through the prism of image processing, harnessing an array of diverse methods and microprocessors. The research trajectory is meticulously designed to confront and surmount the intricate challenges embedded within image capture, information extraction, and noise mitigation, all with the overarching aspiration of elevating measurement precision. Delving deeper, this research especially delves into the utilization of parallax cameras as a potent avenue to extract finer nuances and engender heightened optimization. The ultimate aspiration encapsulates the development of an agile real-time detection system within the realm of machine vision, seamlessly geared toward impeccably measuring parameters while embodying attributes of exceptional precision, non-contact efficacy, and expeditiousness. In essence, this research manifests as a concerted endeavor to furnish tangible resolutions for real-time measurements, culminating in the establishment of an embedded system that exemplifies

* Corresponding author e-mail: luan.mulaku@ubt-uni.net.

reliability, efficiency, and Prowers within the sphere of image processing applications.

Measuring objects through real-time image processing using various methods and microprocessors represents a vibrant field of ongoing research. As emphasized by authors [1], they have elucidated several techniques and algorithms that facilitate the application of image capture[2], while also addressing the challenges that may arise during this process.

Optimization through image processing with the utilization of parallax cameras aims to leverage intricate details and employ diverse methods. Parallax cameras enable to capture of stereo images through the alignment of two video cameras in parallel, enhancing the optimization procedure [3]. The demand for heightened precision and visually captivating image has surged in tandem with the exponential growth of digital image capture on a daily basis. Regrettably, noise consistently impairs the quality of images captured by modern high-definition cameras, thereby diminishing their visual integrity. The predicament necessitates concerted endeavors to curtail noise while preserving image quality. Researchers have introduced various strategies for noise reduction, each carrying its own set of advantages and drawbacks. These diverse techniques encompassing image-denoising are methodically elucidated, commencing with the formulation of the image-denoising problem and culminating in the presentation of a spectrum of cogent recommendations [4].

Reflecting on the outcomes and the methodology employed for selectively acquiring pertinent data while circumventing extraneous elements, this study introduces a real-time machine vision detection system tailored to address parameter measurement challenges. The paper delves into the intricacies of the hardware systems components and the conceptual design of the software, spotlighting how geometry can be harnessed through digital image processing technologies. The operational sequence begins with the capture of an image utilizing the system's industrial camera. Subsequently, the captured image undergoes transmission to the processing unit for the application of image filtering and edge detection templates [5]. The trial results affirm the system's exceptional performance, exhibiting a measuring error of less than 0.5mm. The distinctive attributes of non-contact operation, rapid processing speed, heightened precision, and robust power capabilities underscore the system's potency, unveiling novel avenues for researchers to explore in their pursuit of faulty-free solutions[6].

By implementing such strategies, potential system failure and challenges can be preemptively averted. Tasks encompassing the disconnection of high bit-rate connections and the transmission of voluminous and intricate data loads warrant vigilant consideration. Even in scenarios demanding sophisticated serial buses with elevated data rates, the achievable frame rate in imaging systems, for instance, often encounters limitations owing to the serial connections between the camera and host system. The framework outlined here embodies adaptability, featuring standardized bandwidth and interfaces that can be tailored by the computer to suit specific variations. The integration of sparse data targets, augmented by access links for compression, accelerates data transmission. This comprehensive approach, inclusive of the compression technique, culminates in a comprehensive solution for optimal data transmission across diverse scenarios (Dinkar R. Patnaik Patnaikuni, 2017).

Traditional communication methods face limitations in achieving real-time and dependable transmission of time-sensitive data, particularly within contexts marked by diverse sensor information and the substantial data volumes required to transfer. Addressing this, authors [8] emphasize the amalgamation of a dynamic transmission of critical real-time data such as alarm notifications or production line [9].

Embedded systems exhibit the capability to undertake intricate tasks, necessitating an assessment of system "soft time" for determining its non-stochastic behavior when handling real-time data. The domain of image processing can be envisioned as signal analysis, a realm where inherent variations persist. This complexity, however, can be effectively navigated through the strategic applications of techniques, the establishment of connections, the allocation of technical time, and adaptable environment considerations.

Table 1. A comparioson Between Arduino, Raspberry and ESP node mcu[7]

	Arduino	Raspberry Pi	ESP8266 Node MCU
Developer	Arduino	Raspberry Pi Foundation	ESP8266 open-source community
Type	Si1ngle board microcontroller	Mini computer	Single board microcontroller
Operating System	None	Linux	XTOS
CPU	Atmel, ARM, Intel	ARM Cortex	LXT106
Clock Speed	16 MHz	1.2GHz	26 MHz – 52 MHz
Memory	32KB	1-4GB	Upto 128MB
Storage	1KB	MicroSDHC Slot	4MB
Power	USB, Battery, Power Supply	USB, Power Supply	USB
Operating Voltage	5V	5V	3.3V
I/O Connectivity	SPI I2C UART GPIO	SPI DSI UART SDIOCSI GPIO	UART, GPIO

2. Related work

Despite the fact that cameras have a specific focal length, it is difficult to find a mathematical model that can be adapted to suit different types of cameras, resolutions, object sizes, distances, and other parameters. Authors [10]attempted to compile a mathematical model using zoom measurements to observe changes, but they concluded that smaller distances increase error instead of reducing it. This differs from cameras used to determine the optimal distance to minimize measurement deviations. Due to the large number of parameters that must be considered to determine the most accurate distance, there is no exact relationship that can be adapted.

It is hard to envision any image processing application without using filters to eliminate unwanted signals or portions of images that waste time and serve no purpose. Image processing of noise measurement signals is necessary for accurate defect separation and detection in equipment. Sharp trend shifts in the signals are frequently used to detect problems, and during image processing, these trend shifts should be kept. Sharp trend shifts can be rounded off with linear filters while noise is eliminated. Yet, if the integer weights of nonlinear filters, such as weighted recursive median (WRM) filters, are chosen appropriately, they exhibit good noise reduction while keeping important image properties. For simulated noisy data used in this chapter, it is discovered to produce noise reduction of 52–64% for both sudden and progressive defects in gas turbine machines (Soni, Hetvi; Sankhe, Darshana; Student, 2019).

Several works use various filter types and sensors, and they all do so for essentially the same reasons: either to improve accuracy or to prevent signals that are not required to be processed, like the authors' examples (Md. Abdullah-Al-Noman and Anna Eva and Tabassum Binth Yeahyea and Riasat Khan, 2022) (Md. Abdullah-Al-Noman and Anna Eva and Tabassum Binth Yeahyea and Riasat Khan, 2022)[14][15], something that is also applied in my research because as emphasized, implemented HD cameras during the experiment, collect more data.

The application of parallel cameras to take images from various angles and distances has only one goal which is to improve the accuracy of object measurements, regardless of the field in which they are implemented, such as in industry, medicine, or the environment, and this is similar to what the authors [16][17] found. Cost-effective and compact cameras are able to capture up to 100 gigapixel-scale images thanks to parallel lens systems and parallel image signal processing. Authors [18] in this research, reviewed the context of such cameras in the emerging field of computational imaging, and discussed how the design of optical and electronic processing is affected by parallel architectures. Authors [19]discussed the efficiency of using a graphics processing unit (GPU) in large data processing in the context of processing digital images and how to avoid automatic obstacles (Yan Ren, Jiayong Liu, 2022)[21]. The quantity of simultaneous independent computations determines the level of parallelism and acceleration.

Almost all homes nowadays have IoT capabilities, and the availability of modern appliances has increased the demand for monitoring, which requires the usage of devices such as microcontrollers, similar to how the authors[22] use microcontroller ESP32, which is tested on neural networks. Authors[23] were able to adjust it to the specifications as they are as well as describing the proposed changes to the algorithms to meet the accuracy and performance requirements. The time limit will increase as the resolution quality rises, as is the case with the authors[24].

An effective and powerful minicomputer known as the Raspberry Pi, was developed by the United Kingdom Raspberry Pi Foundation with the intention of educating and enabling the present generation of students to be more creative and efficient. It has numerous applications[19], which are all unique from one another since it may be applied to secure motion capture cameras, AI assistants, low-power smartphones, home automation, or laboratory equipment, and teach bioinformatics[25]. It is frequently used for discovering objects[26], monitoring the environment, and speed, and designing paths[27][28][29][30].The Raspberry Pi model 4B, which is currently available and has a remote server display, can further expand the sophisticated requirements for transmission and monitoring in real-time, even through video with high resolution. Authors[31] have developed a brand-new haze metric called "SATVAL," which evaluates the ratio of an RGB image's maximum saturation to its maximum value when applied to an image scattering model and analyzed in a few moments of video. The authors (Md. Abdullah-Al-Noman and Anna Eva and Tabassum Binth Yeahyea and Riasat Khan, 2022)have also conducted a study in this area using a robotic arm based on computer vision to recognize the colors, shapes, and sizes of the object since the humanoid robot is ideal for applying it to make jobs more efficient.

In general, optimization refers to all techniques or approaches that should be used to increase a system's efficiency while incurring the least amount of expense possible while meeting certain requirements. In essence, it involves finding the best solution to a problem or making a system as efficient as feasible.

Real-time process optimization is critical, but there are several approaches to consider. For each process, several measurements must be made before any model that matches it can be offered. [33] came to the conclusion that a variety of optimization approaches are available now, making it possible to find problems using sophisticated algorithms and high performance or implementation of advanced numerical methods [34][35].

Real-time optimization provides technological excellence, which helps to maximize the plant's contribution to business profit by providing best-in-class performance, optimizing plant operation, and increasing safety and reliability[36], RTO (Recovery Time Objective) contributed to optimizing key process operating variables by shifting the unit margin toward the optimum, and operation was better placed to challenge targets and operating conditions. Due to the increased application of image processing, the increase in speed on the part of hardware components has enabled the development of artificial intelligence, which has also included IoT, where one application is the real-time detection of anomalies in the car registration table in traffic [37].However, just because many specialized architectures and analytics models have been developed for this field does not mean that they are applicable in every case; in these cases, time optimization is frequently done manually by noting errors and their permissions and putting them in the code. Real-time image processing differs from other types of processing in that its analysis and accuracy must be realized quickly.

Advanced technological advancements have made it possible to combine intricate algorithms with several built-in architecture systems, producing a new range of premium equipment at fair costs[38] or consumption time with a planning path and simulation [39][40], the application of real-time optimization in the economic aspect is the main focus of this work. While real-time optimization has advanced significantly, it also needs to build techniques that move us in the direction of cost-oriented and using more micromechatronic components [41].

Since applications can frequently be realized by enhancing the functions of already-existing systems and there is no possibility of changing the technology, which can also have an impact on the cost factor as mentioned above, it is crucial as a parameter in terms of optimizing technological requirements. The primary factors affecting hardware optimization are some of the components contained in execution situations, in the appropriate memory, in the appropriate consumption, and in the resources analysis by applying some algorithms [42][43].

The use of new efficient and effective problem-solving techniques and decision support tools, as well as the continual rise in processing power resources, is essential for addressing issues[44], also authors [45]have global optimization, particularly Bayesian optimization.

It was initially observed that the requirement for optimization has grown along with the development of the application of IoT, Machine learning, Real-time, etc. as a final conclusion from the literature viewed and the works that are similar to the one being given. It was discovered during the investigation that there are numerous parameters that can be optimized, including time, cost, technical requirements, speed of rate, adapting speed all of which are relevant and considered in this study.

Furthermore, it was noted that even though the creation of numerous algorithms, techniques, strategies, codings, etc., aids in our comprehension of the problem, it does not mean that they can be applied in the same way because the field of real-time applications is so vast and is still unexplored. The research outlined by comparable studies is viewed as optimization based on the collection of data from measurements in real-time, where it is then able to find optimal parameters during their analysis through diagrams, and can also apply mathematical techniques that help in the optimization of these data.

3. Data

After the embedded system begins to move within the path, it starts measuring when entering the 1 m region distance from the object, and has taken 10 image sequences while moving from two cameras, that have been analyzed for two objects of various sizes. Normally, the speed would have needed to be adjusted and based on the speed rate so that at least three images could be obtained for comparison in the decision-making area. But two cameras are also required, together with a memory card to store them until comparison and processing, in addition to the microcontroller. These gadgets were put together using the so-called S-car kit using.

Integrating more functions during real-time measurements initially increases the time of analysis and transmission. However, in many cases, it is better to use different methods to capture images for each function that the system needs to perform.

4. Methodology

This study uses mixed methods because there is no unique method between IoT and device models. Although the observable object is static, the system as a whole is in motion, suggesting that it is dynamic. Some sizes are pre-selected, such as the width of the path and the distance between the cameras, because the coding is based on them.

The system is realized in 2D, and it was conducted in a closed environment that had to be tested, with no other obstacles or unwanted noise. The methodology entails calculating the right side coordinate of an object as an obstacle positioned with a predetermined path, given the object's width, height, and x-coordinate of its bottom left corner. This is achieved by computing the horizontal distance of the difference between path width and object width. The calculated distance, representing the gap between the object and the path's right side, is then added to the bottom of the object's left x-coordinate[46]. Consequently, the right side coordinate can be precisely ascertained, allowing for the accurate positioning of objects within the confines of the path.

At the same time, four elements were measured directly, such as:

- The width of the object
- The height of the Object
- The distance from the Object
- The length from the left bottom coordinate object to the path in a horizontal line

and one element used as optimization is indirect:

- The length from the right bottom coordinate of the object to the path in a horizontal line

Conducting iterative experiments within the same scene was imperative to discern the slightest variations, as these disparities held the potential to inadvertently steer outcomes in an accurate direction. However, it is noteworthy that these distinct deviations, while perceptible, do not warrant detailed presentation in this context.

Since cameras support HD technology, they are more advanced, but at the same time, they capture unwanted images. For this reason, filters such as Gauss have been used in order to cancel them. Filters were used because the system did not work precisely from a distance. Decision is made, in cases when it captured any unwanted image.

The flowchart depicts a sequential process: it commences by initiating measurements on the object. Subsequently, measurements are taken for width, height, and distance. The process involves measuring from the left bottom coordinate to the left of the bar path. This leads to the condition IF statement if the right size is greater or equal to 25 cm, the flowchart takes the true path, bypassing the right side. In case that condition is not met, the flowchart analyses other alternatives, following which it repeats the process. Ultimately, the flowchart concludes by ending the process.

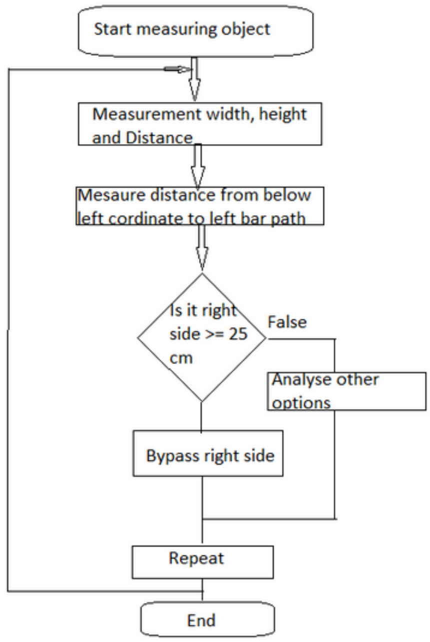

Figure 1. Flow Chart describing logic implementation [46]

The research aimed at the dimension of an object and no other image segmentation was used as is (Peijiang Chena and Yongjun Min, 2014). To determine accuracy, a statistical and comparative method was used, data from a generation at predetermined distances. The delay will be calculated in time that, with optimization on, a more satisfactory result is achieved.

Measurements will be carried out for two objects with different dimensions, the distance is farther than the optimal one to observe. The data collected and stored in the generated file will be structured and analyzed using comparative methods.

Data from a generation at predefined distances were used in a statistical and comparative technique to verify accuracy.

After the data from the experiment are presented in the table, an analysis in Excel and an optimization with "distance-based optimization" in Python will be performed, and finally, will use the optimization methods as "Distance-Based Optimization to justify this application in practice including these facts graphically as well.

4.1. What was the focus to solve out

Aligned with the trajectory or related research, this paper's core focus might encompass akin research methodologies. Preceding the inception of the study, an initial imperative entails grappling with the analysis of data procured via image processing. This endeavor promptly illuminated the potential for a consequential contribution to the optimization paradigm, driven by the escalating imperative for real-time dissemination that increasingly underscored contemporary contexts in recent years.

This work seeks to further examine the use of IoT technology in optimizing measures from real-time image processing based on experience and gaps found throughout the literature review. It can increase the precision and speed of the measurements as well as lower the technical requirements of the equipment needed for the measurement process by applying IoT and other technology advancements. Concentrated on using these data to make the case that it is advantageous to capture more data during the image capture rather than using these data to apply various methods in order to achieve efficiency, both in terms of time and cost, etc.

The phrase "A picture is worth a thousand words" is used frequently in ordinary speech, thus using this as an example, began this research project with the idea that may extract a lot of information (words) from a single image, understanding that after gathering all of this data, it would be simpler to identify relationships. After that, it becomes clearly obvious that by using indirect measurement, one might arrive at relativizing the requests and the task would be completed.

4.2. Designed system for optimized system

Initiating the research, a Smart Video Car kit was meticulously assembled in strict adherence to the manufactures stipulated instructions, featuring Raspberry Pi compatibility and a designated mounting point. Temporary image storage founds its ally in an 8GB micro memory card, judiciously selected to house the images until their impending analysis. To amplify outcome efficacy, HD technology-imbued cameras were strategically employed. Secured in place at a preordained distance, the cameras were affixed using a specialized adhesive and seamlessly interconnected via USB[46]. Noteworthy is the omission of battery supply, owing to the brief measurement distances, with the manufactured charger proving amply sufficient. Within the scope of this research, a consistent and unvarying path measuring 50 cm was employed, circumventing the need for further calibration. The coding architecture was meticulously engineered to undertake a trifold task: gauging the distance of the object from the device, ascertaining its width, and delineating its height.

Once the project was recently finished, it was important to consider every aspect of optimization, from the initial idea to the final system components, including the design and modeling. However, after analysis, nothing stands out as an issue, so they are adequate for this study. As the system uses a servo motor and is small, there may be a little application and testing issue where it dragged us during the return and causes problems with the space tolerance.

4.3. Pseudocode development approach

With an array of variables in play, encompassing rotations moment, driving speed, data transmission rate, and real-time analysis, applying a uniform distance metric based on pre-existing research posed a formidable challenge. This challenge was exacerbated by the nuanced interplay of object size and pre-measurement distance. In light of these complexities, the trajectory chosen diverged from a variable-based approach to embrace the standardized dimensions offered by cameras equipped with Hd technology, underscored by their fixed focal length. A meticulous series of measurements revealed an intriguing phenomenon, the system demonstrated optimal accuracy when situated at the distance in Table 2, pinpointed at 820mm. Remarkably, this optimal distance placement corresponded with the measurement value of 815mm as discerned from Figures 2 and 3.

The first column represents actual measurements with nearly 100% accuracy because they were taken manually, whereas the second and third columns are displayed by the system in real-time and show evident movement. can also figure out that during the testing, there was no correlation between how they were deviating from reality and that the only approximation is about the potentiated distance regardless of the size of the objects being examined.

Table 2. Gathering information to determine an object's real size and distance, comparingmeasurements in real time [46]

Real Distance (mm)	Width (mm)	Height (mm)	Distance showing cameras/(mm)
400	309	497	457
500	249	403	551
600	210	339	640
700	180	290	736
690	155	254	828
800	154	253	826
810	152	251.5	817
820	149.5	249	811
900	139	225	906
1000	126	203	991

The distance between 800 and 820 m, as depicted in Figures2 and 3, is real. Even the deviation of the distance between 817 and 811 millimeters was a fact, and as a result, the average distance of about 815 millimeters displayed by the cameras could be perceived and considered ideal. The range of 400 to 1000 millimeters has been studied, and it is clear that at the smallest distance, the object's width appears to be significantly larger than it actually is. Additionally, it has been observed that the system continues to produce errors as the distance increases, yet the object's width actually shrinks. With the exception of the starting point from the width becoming nearly twice as large at the extreme points, the analysis for Figure2 may be concluded in a manner similar to that for Figure3.

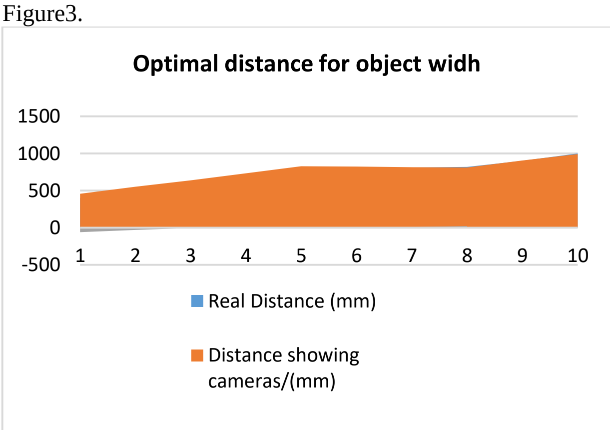

Figure 2. Data were examined to determine the object width

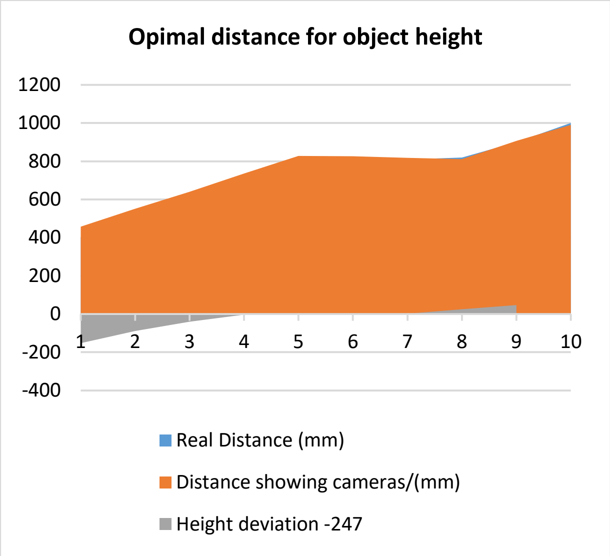

Figure 3. Data were examined to determine the object's height

This is observable even if the "distance-based optimization" method is used with the criteria of finding the ideal value of x4 that best fits x1, while also taking into account that x2 and x3 should be as near to 150mm and 250mm as possible but not necessary constant values. When x1 is the real distance, x2 the width, x3 the height, and x4 the camera's distance as indicated in Table 2, the index of the x mean column is used.

Calculate the distances of x2 and x3 from their respective targets of 150 mm and 250 mm:

$dis\_X2 = abs(x2 - 150)$

$dis\_X3 = abs(x3 - 250)$

For each value of x4, calculate the distance to x1:

$dis\_X1 = abs(x1 - x4)$

Calculate the overall distance as the sum of the distances to x1, x2, and x3:

$dis = dis_{X1} + dis\_X2 + dist\_X3$

To find the value of x4 that minimizes the overall distance are created a code with data from Table 2 using distance-based optimization and it is 802mm from appendix1.Even after numerous tests in the space, it is clear that this is the optimal distance, which has also been confirmed by Figures 2 and 3 which is approximately the same between 800mm and 820mm.

The measurement accuracy is fairly good since when the same image is taken from several angles, it can only deviate if proper approaches are not used. With the appropriate technologies, such as OpenCV and Raspberry Pi, approximately 98% accuracy can be attained[48]. The technology can identify and measure things in a real-time motion picture. After an object is found using a cunny edge detector, its size is calculated using OpenCV methods. Knowing more about the measurement process through using open CV and computer vision, along with various methods is essential, as is the case with the authors[49], where they have suggested several methods as well as devices to achieve accuracy and they convert pixel via multiplying with relevant factor to find the size after measurements.

The uploaded code as Appendix 2 and Appendix 3 performs object analysis within the predefined path. It measures object size and calculates its position relative to the left border. The calculated free space on the right side is evaluated, and a decision is made for movement based on whether the free space exceeds the minimum required of 250 mm. Runtime variations occur due to factors such as hardware differences and concurrent processes. The code aims to ensure safe movements by avoiding obstacles and maintaining adequate space.

5. Resultsand discussions

Figure 4 aptly depicts the bottom coordinate situated on the left side of the object depicted in blue. This illustration harmonizes with the Python code detailed in Appendix 4, encapsulating the culmination of the segment measurement extending up to the designated path. Subsequently, computation seamlessly yields the measurements on the left side. The synergy of utilizing twin parallel cameras, coupled with the crafting of a standardized shape of an upper segment akin to a rectangle, materializes as a visual testament. The envisioned outcomes materialize in the form of two juxtaposed figures that embrace comparability, in resonance with the intended design execution.

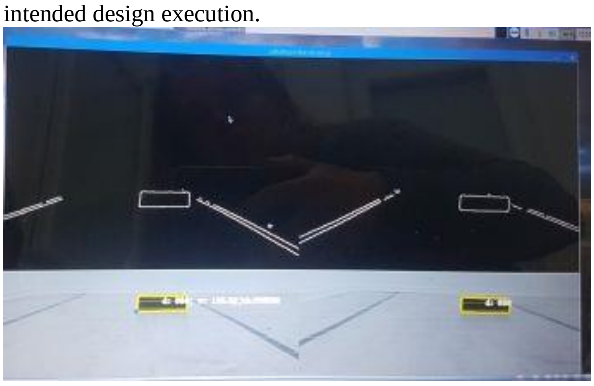

Figure 4. Real-time displaying measurements [46]

Figure 5 serves as an illustrative representation of the practical application of elementary geometric expression for distance determination. Within this construct, the width of the path (L) having been established, the interplay between the object's side measurement from the camera (b) and the segment extending from the left side (a) offers a straightforward avenue for calculating the distance (c).

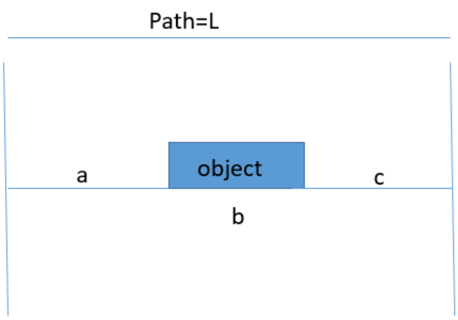

Figure 5. Logic implementation of measurement [46]

Accuracy can be calculated in two different objects, data are collected from Appendix 2 and 3, after applying the equation:

L=a+b+c (1)

Table 3. Calculation of accuracy for both objects

Objects	a/mm	b/mm	c/mm	L/mm	Accuracy/%
1	91	149.9	250.6	491	98.3
2	68	147	285	500	100

The executed code presents a successful endeavor in optimizing object localization within a predefined path by utilizing indirect measures. The approach proves effective in estimating objects' right side while circumventing the computational load of intricate image processing. Through a blend of known distance, calculated geometry, and measurements, the code achieves commendable precision as an average of 99.15%in assessing object position.

The realized optimization not only expedites decision-making for object navigation but also introduces a pragmatic compromise between computational efficiency and accuracy. This outcome serves as the cornerstone for facilitating real-time response in dynamic scenarios, promising a pragmatic solution for efficient object localization within the predefined path.

The method used to compare and evaluate different systems based on multiple criteria and priorities is called multi-criteria decision analysis (MCDA) to find the optimal system or case [50][51][52]. It employs systematic approach that take into account various factors and assigns weights to arrive at a decision that is well-rounded and well-informed. For instance, the cost should vary up to 10%, the processing time for capturing information by up to 25% (given there are four scenarios with the focus solely on indirect measurements), accuracy might experience a decline 2%, and the capacity to observe changes and pinpoint the optimal function value using table may be included. Additionally, there might be potential 10% reduction in technical performance (including aspects like resolutions, memory, and other microcontroller considerations).

In light of this, a method termed multi-criteria analysis (MCDA) can be employed. The systematic approach takes into consideration a spectrum of factors and assigns them appropriate weights, facilitating a more balanced and well-informed decision-making process. Specifically, it operates with a focus on prioritizing both capture time and technical requisites.

To calculate the total weighted score, multiply each criterion by its weight and then add up the results. From the table, we can see that System 1 is more optimized than System 2, with a total weighted score of -15.6% compared to 13.6% for System 2. This means that System 1 is expected to have a greater overall benefit compared to System 2.

Table 4. Applying the MCDA method to analyzethe system

Criteria	Weight	System 1	System 2
Capture Time	0.30	-25%	25%
Accuracy	0.20	-2%	2%
Technical Performance	0.30	-10%	10%
Cost	0.20	-10%	10%
Total Weighted	1.00	-15.6%	13.6%

The main reason for System 1 being more optimized is the larger decrease in capture time, which is given the highest weight. The negative impact on accuracy and technical performance is relatively small and is offset by the decrease in cost. On the other hand, System 2 has a higher increase in accuracy and technical performance, but this is outweighed by the increase in capture time and cost.

According to the title of the article, optimization with the application of a technique that increased data analysis, lead to accurate measurements, reducing cost and time, and increasing speed, while roughly maintaining accuracy even if they were made for each situation.

6. Conclusion

Translating optimization's success into concrete navigation decisions reveals a multifaceted landscape. While leveraging unique methods and the camera's focal length could yield benefits, the intricacies of real-time decision-making reflect a complex interplay of variables. The speed of mobile robot cars, speed of rate, and object size add layers of complexity that necessitate adaptive algorithms.

The gap between optimization and practical applications underscores the critical role of adaptive decision-making strategies that comprehensively address the aforementioned variables. This vision holds the potential to bridge the gap between optimization outcomes and on-ground navigation decisions. Further investigations into adaptive mechanisms present an exciting frontier that could yield breakthroughs in navigation accuracy and responsiveness in rapidly changing environments.

Implementing precise measurements from parallax cameras and optimizing them within industrial settings offers enhanced efficiency and accuracy. By integrating real-time object measurements and sophisticated algorithms. Industries such as manufacturing, logistics, and robotics can achieve advanced automation and quality control.

The utilization of 3D data acquired through parallax imaging allows accurate object recognition sizing, and positioning, leading to optimized processes and reduced errors. This technology can streamline assembly lines, enable robotic systems to navigate complex environments, and improve quality assessment, ultimately contributing to higher productivity, cost saving, and improved operational reliability within industrial operations

The optimization framework's success elucidates a promising trajectory for future research. Specifically, the exploration of adaptive decision-making algorithms emerges as a compelling avenue. The integration of system speed, speed of rate, and object dimensions as dynamic inputs could pave the way for algorithms that autonomously adjust navigation strategies based on real-time conditions.

Acknowledgments

This article would not have been published without the valuable assistance of my colleagues.

There was no specific grant from any funding body in the public, private, or nonprofit sectors for this research.

References

[1] Q. Yunliang; Y. Zhen; S. Wenhao,L. Meng, L. Jing, Zh. Wenwei, D. Xiangyu, M. Yide, "A Comprehensive Overview of Image Enhancement Techniques," *Archives of Computational Methods in Engineering*, vol. 29, No. 1, 2022, 583-607.

[2] N. Alaa, "Image Processing with Python: An Introduction, Kalaa of Sraghna, Essaouira and Safi" LAMAI Laboratory FST Marrakech, 2021, 10.13140/RG.2.2.18870.37444.

[3] T. Hsu., T.Wang, "An Improvement Stereo Vision Images Processing for Object Distance Measurement," *International Journal Of Automation And Smart Technology*, vol. 5, no. 2,2015, pp. 85-90,

[4] F. Linwe, Zh. Fan, F. Hui, Zh. Caiming, "Brief review of image denoising techniques," *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, no. 1,2019, p. 7.

[5] Ch. Kuang, X. Zheng, "Space Trajectory Planning of Electric Robot Based on Unscented Kalmar FIlter," *Jordan Journal of Mechanical and Industrial Engineering*, vol. 15, no. 1, 2021 pp. 29-38.

[6] W. Wencheng, X. Cui, "An Image-Based System for Measuring Workpieces," *Measurement and Control*, vol. 47, no. 9,

2014, pp. 283-287.

[7] R. Dinkar, P. Patnaikuni, "A Comparative Study of Arduino, Raspberry Pi and ESP8266 as IoT Development Board," *International Journal of Advanced Research in Computer Science,* vol. 8, 2017, pp. 2350-2352.

[8] R.B. Mohammed, R.V. Silfhout, "High bandwidth data and image transmission using a scalable link model with integrated real-time data compression,," *Advances in Electrical Engineering, Electronics and Energy,* vol. 1,2021, p. 100017.

[9] *"Research on Multi-priority Real-time Data Transmission Scheduling Method for Intelligent Production Line". in 13th International Conference on Intelligent Computation Technology and Automation (ICICTA), . A. Chai, Y. Ma, Z. Yin, M. Li, Z. Bi, L. Li, Z. Huang,* Intelligent Computation Technology and Automation, *Xi'an, China, 2020*

[10] J. Huda, H. Tahseen, "Measuring Object Dimensions and its Distances Based on Image Processing Technique by Analysis the Image Using Sony Camera," *Eurasian Journal of Science and Engineering,* vol. 3, no. 2,2017, pp. 100-110.

[11] R. Ganguli. Image Processing Filters for Machine Fault Detection and Isolation. In:*O. Sergiyenko,W. F.Fuentes, P. Mercorelli, editors,Machine Vision and Navigation,* Springer International Publishing, Cham: h*ttps://doi.org/10.1007/978-3-030-22587-2_23;* 2020, pp. 733-749.

[12] C. M. Maheshan., P. Kumar, "Performance of image pre-processing filters for noise removal in transformer oil images at different temperatures," *Applied Science,* vol. 2, no. 1, 2019, p. 67.

[13] S. Hetvi, S. Darshana, student, "Image Restoration using Adaptive Median Filtering," *International Research Journal of Engineering IT & Scientific Research,2019,*pp. 2395-0056.

[14] Gan, W.S.. Digital Image Filtering. In: Signal Processing and Image Processing for Acoustical Imaging. Springer, Singapore.2020 https://doi.org/10.1007/978-981-10-5550-8_11

[15] C. Kuang, X. Zheng, "Space Trajectory Planning of Electric Robot Based on Unscented," *Jordan Journal of Mechanical and Industrial Engineering,* vol. 5, no. 1, 2021, pp. 29-38.

[16] M. Janne, K. Juh, H. Janne, "Parallax correction via disparity estimation in a multi-aperture camera," *Machine Vision and Applications,* vol. 27, no. 8,2016, pp. 1313-1323.

[17] "Embedded Real-Time Object Detection and Measurement of its Size". O. Nashwan, S. Mehmet, K. Mehmet, A. Ilhan, in *International Conference on Artificial Intelligence and Data Processing*, Turkey, 2018.

[18] B. David, P. Wubin, L. Han, M. Zhan, T. Yue, C. Xun, "Parallel cameras," *Optica,* vol. 5, 2018, p. 127.

[19] "Digital image processing using parallel computing based on CUDA technology". International Conference on Information Technologies in Business and Industry S. Igor, Pustovit, A, A. Mariya, *Journal of Physics, Tomsk, Russian Federation, 2016*

[20] Y.Ren, J.Liu, "Automatic Obstacle Avoidance Path Planning Method for," *Jordan Journal of Mechanical and Industrial Engineering,* vol. 16, no. 1, 2022, pp. 11-18.

[21] P. K. Dan, "Obstacle Avoidance and Travel Path Determination in Facility Location Planning," *Jordan Journal of Mechanical and Industrial Engineering* , vol. 3, no. 1, 2009, pp. 37-46.

[22] O. Maksym, H. Volodymyr, "IMAGE PROCESSING ON ESP32 MICROCONTROLLERS BASED ON MOBILENET CONVOLUTIONAL NEURAL NETWORK," *May 20, 2022 • Cambridge, UK • 159 . SECTION XIV. AUTOMATION AND APPLIANCES MAKING,* vol. 8, 2022.

[23] M. Chamin, K.Michihiro, S. Masak, K.Yasuhiro, I. Teppei, M. Hitoshi, M.Takeshi, H. Masaki, "Image and video processing on mobile devices: a survey," *the visual Computer,* vol. 37, no. 12, 2021, pp. 2931-2949.

[24] E. B.-Sterp, R. Brad, R, Brad, "A Comparative Study of Stereovision Algorithms," *International Journal of Advanced Computer Science and Applications,* vol. 8, no. 11, 2017.

[25] H. Ghael, "A Review Paper on Raspberry Pi and its Applications," *International Journal of Advances in Engineering and Management,* vol. 2, no. 12, 2020, pp. 225-227.

[26] A.. C. Fletcher,C. Mura,, "Ten quick tips for using a Raspberry Piv," *PLoS Comput Biol,* vol. 15, no. 5, 2019, p. e1006959.

[27] N. I. Binti Rasidi, O. I. Al-Sanjary, M. Y. Kashmola, K. Loo Teow Aik, "Development on Autonomous Object Tracker Robot using Raspberry Pi," in *IEEE 10th Conference on Systems, Process & Control (ICSPC)*, Malacca, Malaysia, 2022.

[28] Li, Y.P., "Design of path tracking control system for UAV," *Jordan Journal ofMechanical and Industrial Engineering,* vol. 14, no. 1,2020, pp. 101-108.

[29] P. Sanjana, Mahalakshmi, P., John, A., S. Clement, Swathi, R., "Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor," in *International Journal of Computer Science and Information Technologies,* Vol 5, no 6,2014, p. 7107-7109

[30] Y. Tang, "Monitoring Algorithm for Speed Information of Autonomous," *Jordan Journal of Mechanical and Industrial Engineering,* vol. 14, no. 1, 2020, pp. 43-52.

[31] G. Avra, A. Asfak, R. Sangita, C. S. Sinha, "Novel parametric based time efficient portable real-time dehazing system," *Journal of Real-Time Image Processing,* vol. 20, no. 2 2023, p. 23.

[32] Md. Abdullah-Al-Noman and Anna Eva and Tabassum Binth Yeahyea and Riasat Khan, "Computer Vision-based Robotic Arm for Object Color, Shape, and Size Detection," *Journal of Robotics and Control (JRC), Vol. 3, 2022, no. 2.*

[33] "Optimization Techniques in Modern Times and Their Applications,". in *International Electrical Engineering Congress (iEECON*B. Somiya, S. K. Bhushan, S. R.,, IEEE Xplore, Krabi, Thailand, 2019.

[34] Y. Wang, .o Yu, F. Berto, W. Cai, K. Bao, "Modern numerical methods and their applications in mechanical engineering," *Advances in Mechanical Engineering,* vol. 11, no. 11, 2019, p. 1687814019887255.

[35] J. Agushaka, E. Ezugwu, "Advanced arithmetic optimization algorithm for solving mechanical engineering design problems.," *PLoS ONE,* vol. 16, no. 8, 2021, p. e0255703., 2021.

[36] R. M. Ansari, H. Imran, A. S. Hunaidy, "Integration of Real-Time Optimization and Model-Predictive Control: Application to Refinery Processes,," *International Journal of Control Science and Engineering,* vol. 10, no. 1, 2020, pp. 1-10.

[37] M. Chen, H. Te-Wu, "Real-time intelligent image processing for the internet of things," *Journal of Real-Time Image Processing,* vol. 18, no. 4, 2021, pp. 997-998.

[38] E. Oliveira-Silva, C. de Prada, D.Navia, "Economic MPC with Modifier Adaptation using Transient Measurements,"Computer Aided Chemical Engineering vol. 50, 2023, p. 1253-1258.

[39] M. Samer, H. Mohammad, B. Suleiman, Q. Mohammad,"Simulation of Trajectory Tracking and Motion Coordination for Heterogeneous Multi-Robots System," *Jordan Journal of Mechanical and Industrial Engineering,* vol. 15, no. 4, 2021, pp. 337-345.

[40] H. Zhaoxi, Y. Feng, L. Zhongkai, T. Guangdong, T. Jianrong, "Reliability-Based and Cost-Oriented Product Optimization Integrating Fuzzy Reasoning Petri Nets, Interval Expert Evaluation and Cultural-Based DMOPSO Using Crowding

Distance Sorting," *Applied Sciences,* vol. 7,2017, p. 791.

[41] Q. Wang, Y. Zhao, H. Shi, "Design of Visual Monitoring Software for Micro-Drive Debugging in Electromechanical System," *Jordan Journal of Mechanical and Industrial Engineering* , vol. 16, no. 1,2022, pp. 133-139.

[42] "Hardware Resource Optimization for Embedded System Design," in *3rd International Conference for Convergence in Technology (I2CT*, Y. J. Pavitra, S. Jamuna, J. Manikandan, Pune, India, 2018.

[43] R. V. Rao, N. Zinzuvadia, "Design Optimization of Selected Mechanical Engineering Components using Variants of Rao Algorithms," *Jordan Journal of Mechanical and Industrial Engineering* , vol. 16, no. 5, pp. 835-863, 2022.

[44] "Hardware Resource Optimization for Embedded System Design," in *3rd International Conference for Convergence in Technology (I2CT*, Y. J. Pavitra, S. Jamuna, J. Manikandan, Pune, India, 2018.

[45] C. Antonio, A. Francesco, "Global optimization in machine learning: the design of a predictive analytics application," *Soft Computing,* vol. 23, no. 9, 2019, pp. 2969-2977.

[46] Mulaku, L.. Cost-oriented image processing for mobile robots [Dissertation, Technische Universität Wien]. reposiTUm. http://hdl.handle.net/20.500.12708/79847, (2020).

[52] A. F. A. Moneim, "Fuzzy Genetic Prioritization in Multi-Criteria Decision Problems," *Jordan Journal of Mechanical and Industrial Engineering* , vol. 2, no. 4, pp. 175-182, 2008.

[47] P. Chena, Y. Min, "Automobile Longitudinal Axis Detection Method Based on Image Segmentation and Preliminary Results," *Jordan Journal of Mechanical and Industrial Engineering* , vol. 8, no. 5, 2014, pp. 297-303.

[48] P. Et, G. Sachin, "Realtime Object's Size Measurement from Distance using OpenCV and LiDAR," *Turkish Journal of Computer and Mathematics Education,* vol. 12, 2021, pp. 1044-1047.

[49] G. Anand, A. K. Kumawat, "Object detection and position tracking in real time using Raspberry Pi,," *Materials Today: Proceedings,* vol. 47, no. 11, 2021, pp. 3221-3226..

[50] N. Godwin, R.Ebenezer, S. Ramabalan , "Advanced Multi Criteria Optimal Design of Spiral Bevel Gear," *Jordan Journal of Mechanical and Industrial Engineering,* vol. 16, no. 2, 2022, pp. 185-193..

[51] L. Ocampoa, E. Clark, "A Sustainable Manufacturing Strategy Decision Framework in the Context of Multi-Criteria Decision-Making," *Jordan Journal of Mechanical and Industrial Engineering* , vol. 9, no. 3, 2015, pp. 177-186..

APPENDIX 1

Algorithm: Finding optimal distance for the decision to measure in real-time based on Table 1 data

```
import pandas as pd

# Create a DataFrame with the table data
table = pd.DataFrame({
'X1/mm': [400, 500, 600, 700, 690, 800, 810, 820, 900, 1000],
'X2/mm': [309, 249, 210, 180, 155, 154, 152, 149.5, 139, 126],
'X3/mm': [497, 403, 339, 290, 254, 253, 251.5, 249, 225, 203],
'x4/mm': [457, 551, 640, 736, 828, 826, 817, 811, 906, 991]
})

# Find the optimal value of x4
min_dist = float('inf')
optimal_x4 = None

for x4 in table['x4/mm']:
dist_x2 = abs(table.loc[table['x4/mm'] == x4, 'X2/mm'].values[0] - 150)
dist_x3 = abs(table.loc[table['x4/mm'] == x4, 'X3/mm'].values[0] - 250)
dist_x1 = abs(table.loc[table['x4/mm'] == x4, 'X1/mm'].values[0] - x4)
dist = dist_x1 + dist_x2 + dist_x3
if dist < min_dist:
min_dist = dist
optimal_x4 = x4

print('The optimal value of x4 is:', optimal_x4, 'mm')
```

APPENDIX 2

Algorithm: After execution, optimization of the second object is performed[46].

1. **Input** Analyze Shapes Distance from object: 815
2. Measure object **sizeanalyses** _shapes Distance, width and high at 815, width 147.00, height 224.48
3. Measure **Distance difference** analyse_shape Distance: -5
4. Measure distance from **left border** analyse_shapes Distance from left border:68
5. Time of runtime analyse_shapes <module> Run time: 0.4155s
6. Measure **Distance difference** <module> Distance diff: -5
7. Measure free space on right side and **decision** for movement: free space right side: 285.0, pass right side

APPENDIX 3

Algorithm: After execution, optimization of object one is performed.

1. **Input** Analyze Shapes Distance from object: 815
2. Measure object **size** analyses _shapes Distance, width and high at 815, width 149.94, height 42.32
3. Measure **Distance difference** analyse_shape Distance: -5
4. Measure distance from **left border** analyse_shapes Distance from left border:91
5. Time of runtime analyse_shapes <module> Run time: 0.318s
6. Measure **Distance difference** <module> Distance diff: -5
7. Measure free space on right side and **decision** for movement: free space right side: 250.06, pass right side

APPENDIX 4

Algorithm : Logic to implement for optimization[46]

1. Part of **Defination**:
   def distance_from_left_border (image, x, y):
2. Crop a single subject
   roi = (0, y, x, 20)
3. Crop from **Python Imaging Library**
   imCrop = image [int (roi (1):int(roi(1) * roi (3)), int (roi(0)): int (roi(0) * roi(2))]
4. Add image to window adjust size
   cv2.imshow (‘imCrop’, imCrop)
5. **Colours from Library**
   gray = cv2.cvtColor (imCrop, cv2, COLOR_BGR2GRAY)
6. Detection **Edge**
   edges = cv2.Canny (gray, 150, 200, apertureSize = 3
7. Fit image Size
   cv2.imshow (‘imCrop gray’, edges)
8. String minimum lenght
   miniLineLength = 20
9. Single line allow
   maxlineGap =5
10. Threshold edge detection, applying hough transform
    lines = cv2. HoughlinesP (edges, 1, np.pi/100, 20, minLineLength, m
11. Using **For** Loop
    for line in lines:
    for x1, y1, x2, y2 in line

```
    #cv2.circle (img_1, (int(x), int(y)), 5, (255, 0, 0), -1)
#cv2.line (image, (x1, y1+y), (x2, y2+y), (255, 0, 0), 2
    #
```

If loop under For

```
    if y2<5:
    #cv2.line(image, (x,y), (x2,y2+y), )255,255,0), 2
    print (‘Distance from left border: {}’, format (x-x2))
    return x-x2
```