ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

THE LUCAS-KANADE METHOD IMPLEMENTATION FOR ESTIMATING THE OBJECTS MOVEMENT IN THE MOBILE ROBOT'S WORKSPACE

Svitlana Maksymova[1], Vladyslav Yevsieiev[1], Ahmad Alkhalaileh[2]

[1]Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

[2]Senior Developer Electronic Health Solution, Amman, Jordan

Abstract:

This article presents the Lucas-Kanade method implementation for estimating the objects movement in the mobile robot's workspace using the Python programming language. The Lucas-Kanade method is used to calculate optical flow from sequential images and allows the motion of objects to be estimated. The necessary mathematical expressions are considered. As part of the study, experiments were conducted with different lighting levels to evaluate the robotic ability of the method under changing lighting conditions.

Key words: Industry 5.0, Computer Vision Systems, Mobile Robots, Manufacturing Innovation, Industrial Innovation

INTRODUCTION

In today's Industry 5.0 context, where automation and digitalization play a key role in production processes, robotics is becoming an integral part of the work environment [1]-[11]. However, effective operation of mobile robots in work areas requires reliable detection and tracking of objects to avoid collisions and ensure process safety [12]-[29]. Therefore, various methods and approaches can be used here [30]-[37]. In this context, the Lucas-Kanade method for estimating the motion of objects in the mobile robot's workspace becomes an important tool for computer vision systems [38]-[47]. This method, based on computational optics, provides the ability to analyze streaming video and detect the movement of objects in real time. The use of the Lucas-Kanade method in computer vision systems for mobile robots provides the ability to accurately and quickly identify moving objects in the work area, which increases the efficiency of robots and the safety of production processes. This article will discuss the

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

implementation of the Lucas-Kanade method for estimating the movement of objects in the work area of a mobile robot using modern computational methods and technologies. The focus will be on the algorithmic and software aspects of the implementation, as well as investigating its effectiveness and applicability in the context of Industry 5.0 and modern requirements for computer vision systems for mobile robots.

Related works

The Lucas-Kanade algorithm is widely used to construct a motion trajectory for a mobile robot. Let us look at some recent research in this field.

Let us begin with the research [38]. It presents an efficient control structure of two mobile robots based-visual navigation methods in an indoor environment.Here the proposed navigators are based on decision systems employed the necessary values estimated by a Lukas-Kanade algorithm of optical flow approach.

The paper [39] is devoted to solving the problem for computer vision systems, that lies in estimating homography to align image pairs captured by different sensors or image pairs with large appearance changes. A generic solution to pixel-wise align multimodal image pairs by extending the traditional Lucas-Kanade algorithm with networks is proposed in this work.

The authors in [40] investigate two discriminant algorithms to analyze visual cues in unfamiliar static environments. They propose control systems that rely on estimations of movement and decision-making mechanisms that use the measured data calculated by the optical inflow algorithms to direct the robot autonomously within its working area.

Xue, Z. in [41] note, that SLAM technology is also developing rapidly as an essential part of robot perception modality. The basic architecture of SLAM based on binocular vision is developed. The author separate the front-end and back-end and achieve good tracking and map-building results using a multi-threaded approach.

The study [42] presents multi-robot vision collaborative SLAM system/ It is provided to respond to the needs of large error and poor sensitivity in the multi-robot vision collaborative SLAM system. Here Lucas-Kanade (LK) sparse optical flow is employed to enhance the ORB algorithm’s feature matching process.

Bahnam, S., and co-authors [43] used the Inverse Lucas-Kanade algorithm for feature tracking and stereo matching.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

Scientists [44] build and evaluate a system for tracking a person's movement and motion under various lighting conditions. The system uses Lucas-Kanade Optical Flow and Canny Edge Detector.

Research [45] assesses the performance of a newly developed algorithm for visual-inertial navigation of robotic systems in GNSS-denied environments.

Two gradient-based algorithms—Lucas–Kanade and Horn–Schunck—were implemented on a ZCU 104 platform with Xilinx Zynq UltraScale+ MPSoC FPGA in [46]. The algorithms realized in this work can be a component of a larger vision system in advanced surveillance systems or autonomous vehicles.

Hannum, C., & et al. [47] use the Lucas-Kanade optical flow algorithm for the robot. They evaluate trust level dynamically throughout the collaborative task that allows the trust level to change if the human performs false positive actions, which can help the robot avoid making unpredictable movements and causing injury to the human.

The Lucas-Kanade optical method implementation for detecting the contour of an object in real time

The Lucas-Kanade method is a classic algorithm for optical flow in computer vision. It is used to estimate the motion of objects in video sequences based on the movement of bright points (or features) between frames. Lucas-Kanade method is one of the simple and effective optical flow methods, and it is widely used in many applications such as object tracking, video stabilization, motion speed estimation and others. However, it has its limitations, such as the assumption of local motion and inapplicability to large displacements of objects.

Let's assume that we have an image $I(x, y, t)$, where (x, y) are the coordinates of the pixel in the image, and t is the time. We want to determine how each pixel moves between two frames t and $t+\Delta t$ may be approximated by a linear function:

$$I(x, y, t+\Delta t) - I(x, y, t) = \nabla I \cdot \Delta P, \qquad (1)$$

∇I – pixel intensity gradient (vector-gradient), which is determined by the partial derivatives of intensity with respect to coordinates x and y;

ΔP – pixel offset vector.

Minimizing errors. The goal is to find a bias vector ΔP that minimizes the error between the actual and predicted intensity change. To do this, it is proposed to use the

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

least squares method. This will allow us to minimize the sum of squared errors E for all pixels in the image:

$$E = \sum_{(x,y)} [I(x,y,t+\Delta t) - I(x,y,t) - \nabla I \cdot \Delta P]^2 . \qquad (2)$$

Since the model is linear and may be a rough approximation, the Lucas-Kanade method uses an iterative approach to refine the optical flow estimate. At each iteration k, the bias vector ΔP_k is calculated using the current estimate ΔP_{k-1} and adjusted for the residual error.

From a program development perspective, Python implementations of the Lucas-Kanade method need to consider the following parameters, which can be tuned to optimize performance and accuracy:

winSize – window size for optical flow calculation. It determines the size of the region in which the optical flow search occurs. A larger window size may improve accuracy, but may increase the computational load;

maxLevel – maximum pyramid level for multi-level optical flow calculation. This allows you to increase the search area and improve the stability of the algorithm to large movements of objects;

criteria – criterion for stopping iterations. This option allows you to control the number of iterations and stop the process when a certain condition is reached.

The Lucas-Kanade method allows you to estimate the movement of objects in a video stream using a linear approximation of the change in pixel intensity between frames and an iterative approach to refine the optical flow estimate. It is based on minimizing the error between the actual and predicted pixel intensity changes using the least squares method.

Software implementation and experiments

To check the correctness of the reasoning, we will develop a program in Python in the development environment PyCharm 2022.2.3 (Professional Edition). Let us give an example of software implementation of the above described mathematical expressions.

```
# Draw lines between old and new points
for i, (new, old) in enumerate(zip(good_new, good_old)):
a, b = new.ravel()
```

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

```
c, d = old.ravel()
frame = cv2.line(frame, (int(a), int(b)), (int(c), int(d)), (0, 255, 0), 2)
```

This code snippet draws lines between the old and new points that were found by the Lucas-Kanade method. Let's break it down:

for i, (new, old) in enumerate(zip(good_new, good_old)):: This loop goes through each pair of new and old points. The zip() function pairs good_new and good_old to iterate over them simultaneously. The enumerate() function adds an index to each pair so that we can keep track of the number of the current point pair;

a, b = new.ravel() and c, d = old.ravel(): These lines unpack the coordinates of the new and old points from their arrays and assign them to the variables a, b and c, d respectively. The ravel() method is used to convert a two-dimensional array of points into a one-dimensional array of coordinates;

frame = cv2.line(frame, (int(a), int(b)), (int(c), int(d)), (0, 255, 0), 2): This line draws a line between the old point (c, d) and a new point (a, b) on frame. The line color is specified by the tuple (0, 255, 0), which represents the color green in BGR format. Line thickness is set to 2.

This code snippet draws lines between each pair of old and new points found by the Lucas-Kanade method. This allows you to visualize the optical flow and track the movement of objects on the video stream.

```
# Update previous frame
    old_gray = gray_frame.copy()
```

This code snippet updates the previous frame, old_gray, to the current gray image, gray_frame.

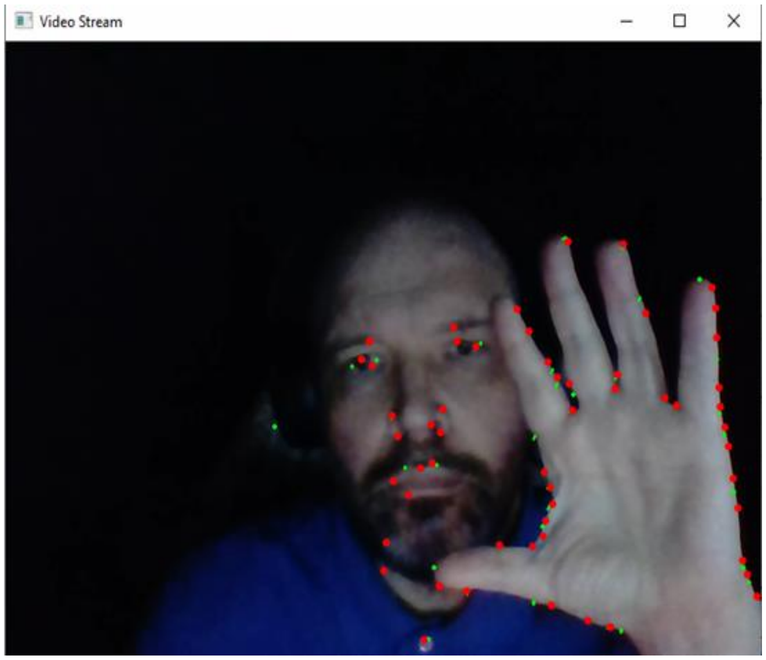
Let's look at what's going on:

old_gray = gray_frame.copy(): This line creates a copy of the current gray image gray_frame and assigns it to the old_gray variable. This is necessary to subsequently use a copy of the current frame as the previous frame in the next processing step. Using a copy allows you to save the previous frame state and use it to compare with the current frame when calculating optical flow using the Lucas-Kanade method.

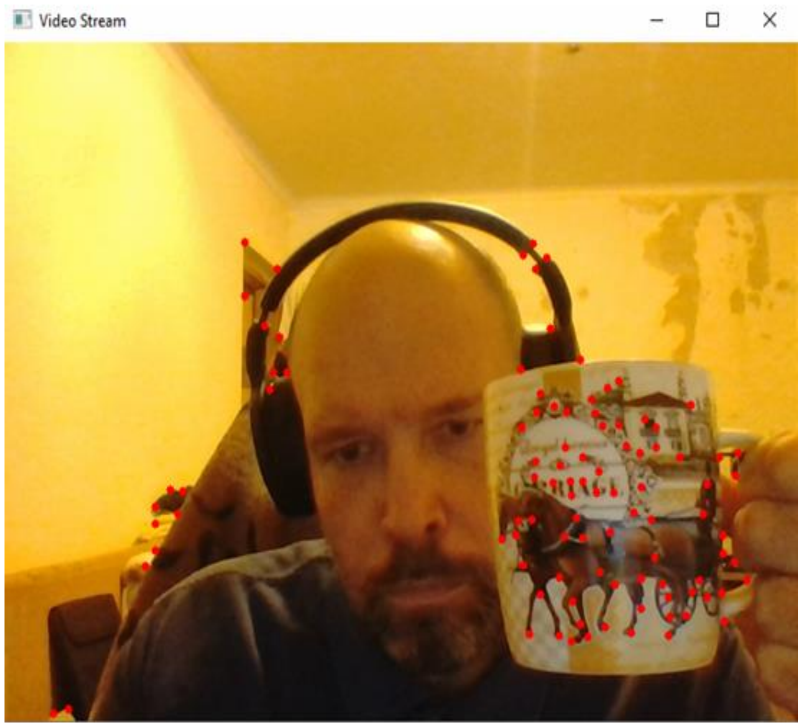
The following hardware was used for research: CPU Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz, RAM 16 Gb, GPU NVideo GeForce GTX 1660Ti (Ram 8Gb), Web-camera HD WebCam, OS Windows 10 Pro (Version 22H2). A program

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

for implementing the Lucas-Kanade optical method for detecting the contour of an object in real time was developed in the PyCharm 2022.2.3 (Professional Edition) environment in Python. The results of the program are presented in Figure 1.



a)



ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

b)

a) – in the dark (minimal lighting); b) – under artificial lighting

Figure 1: Results of implementations of the Lucas-Kanade method for detecting the contour of an object in real time.

Conclusion

The Lucas-Kanade method is an effective method for real-time object contour detection for mobile robots. It is based on the calculation of optical flow, which is a vector field that reflects the movement of points in the image. The advantage of this method is its relative simplicity and speed of operation, which makes it suitable for use on mobile robots.

To successfully implement the Lucas-Kanade method for real-time edge detection, it is recommended to consider the following aspects. First, it is necessary to correctly select the method parameters, such as window size and threshold for identifying motion points. Secondly, it is important to consider real-time image processing to minimize latency and ensure smooth execution.

It is also recommended to use optimizations such as parallel computing and the use of hardware acceleration to improve the performance of the method on mobile robots. It is also important to consider the environment and lighting conditions when processing images for more accurate edge detection.

Overall, the Lucas-Kanade method provides an efficient and fast way to detect object contours in real time for mobile robots. Its ease of implementation and high speed make it an excellent choice for computer vision tasks on mobile platforms.

REFERENCES:

1. Baker, J. H., Laariedh, F., Ahmad, M. A., Lyashenko, V., Sotnik, S., & Mustafa, S. K. (2021). Some interesting features of semantic model in Robotic Science. SSRG International Journal of Engineering Trends and Technology, 69(7), 38-44.

2. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2021). Neural networks as a tool for pattern recognition of fasteners. International Journal of Engineering Trends and Technology, 69(10), 151-160.

3. Sotnik, S., Matarneh, R., & Lyashenko, V. (2017). System model tooling for injection molding. International Journal of Mechanical Engineering and Technology, 8(9), 378-390.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

4. Nevliudov, I., & et al.. (2020). Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis. International Journal of Emerging Trends in Engineering Research, 8(10), 7465-7473.

5. Maksymova, S., Matarneh, R., Lyashenko, V. V., & Belova, N. V. (2017). Voice Control for an Industrial Robot as a Combination of Various Robotic Assembly Process Models. Journal of Computer and Communications, 5, 1-15.

6. Sotnik, S., Mustafa, S. K., Ahmad, M. A., Lyashenko, V., & Zeleniy, O. (2020). Some features of route planning as the basis in a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(5), 2074-2079.

7. Babker, A. M., Abd Elgadir, A. A., Tvoroshenko, I., & Lyashenko, V. (2019). Information technologies of the processing of the spaces of the states of a complex biophysical object in the intellectual medical system health. International Journal of Advanced Trends in Computer Science and Engineering, 8(6), 3221-3227.

8. Nevliudov, I., Yevsieiev, V., Lyashenko, V., & Ahmad, M. A. (2021). GUI Elements and Windows Form Formalization Parameters and Events Method to Automate the Process of Additive Cyber-Design CPPS Development. Advances in Dynamical Systems and Applications, 16(2), 441-455.

9. Mustafa, S. K., Yevsieiev, V., Nevliudov, I., & Lyashenko, V. (2022). HMI Development Automation with GUI Elements for Object-Oriented Programming Languages Implementation. SSRG International Journal of Engineering Trends and Technology, 70(1), 139-145.

10. Sotnik, S., & Lyashenko, V. (2022). Prospects for Introduction of Robotics in Service. Prospects, 6(5), 4-9.

11. Ahmad, M. A., Sinelnikova, T., Lyashenko, V., & Mustafa, S. K. (2020). Features of the construction and control of the navigation system of a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(4), 1445-1449.

12. Abu-Jassar, A., & et al. (2024). The Optical Flow Method and Graham's Algorithm Implementation Features for Searching for the Object Contour in the Mobile Robot's Workspace. Journal of Universal Science Research, 2(3), 64-75.

13. Al-Sharo Y., & et al. (2023). A Robo-hand prototype design gripping device within the framework of sustainable development. Indian Journal of Engineering, 20, e37ije1673.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

14. Yevsieiev, V., & et al. (2024). Object Recognition and Tracking Method in the Mobile Robot's Workspace in Real Time. Technical Science Research In Uzbekistan, 2(2), 115-124.

15. Basiuk, V., & et al. (2023). Mobile Robot Position Determining Using Odometry Method. Multidisciplinary Journal of Science and Technology, 3(3), 227-234.

16. Abu-Jassar, A., & et al. (2023). Obstacle Avoidance Sensors: A Brief Overview. Multidisciplinary Journal of Science and Technology, 3(5), 4-10.

17. Yevsieiev, V., & et al. (2024). Active Contours Method Implementation for Objects Selection in the Mobile Robot's Workspace. Journal of Universal Science Research, 2(2), 135-145.

18. Igor, N., & et al. (2023). Using Mecanum Wheels for Radio Shuttle. Multidisciplinary Journal of Science and Technology, 3(3), 182-187.

19. Akopov, M., & et al. (2023). Choosing a Camera for 3D Mapping. Journal of Universal Science Research, 1(11), 28-38.

20. Stetsenko, K., & et al. (2023). Exploring BEAM Robotics for Adaptive and Energy-Efficient Solutions. Multidisciplinary Journal of Science and Technology, 3(4), 193-199.

21. Yevsieiev, V., & et al. (2024). Using Contouring Algorithms to Select Objects in the Robots' Workspace. Technical Science Research In Uzbekistan, 2(2), 32-42.

22. Nevliudov, I., & et al. (2023). Mobile Robot Navigation System Based on Ultrasonic Sensors. In 2023 IEEE XXVIII International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), IEEE, 1, 247-251.

23. Kuzomin, O., Ahmad, M. A., Kots, H., Lyashenko, V., & Tkachenko, M. (2016). Preventing of technogenic risks in the functioning of an industrial enterprise. International Journal of Civil Engineering and Technology, 7(3), 262-270.

24. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2023). Generalized Procedure for Determining the Collision-Free Trajectory for a Robotic Arm. Tikrit Journal of Engineering Sciences, 30(2), 142-151.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

25. Lyashenko, V., Laariedh, F., Ayaz, A. M., & Sotnik, S. (2021). Recognition of Voice Commands Based on Neural Network. TEM Journal: Technology, Education, Management, Informatics, 10(2), 583-591.

26. Attar, H., & et al.. (2022). Zoomorphic Mobile Robot Development for Vertical Movement Based on the Geometrical Family Caterpillar. Computational Intelligence and Neuroscience, 2022, Article ID 3046116.

27. Sotnik, S., & et al.. (2022). Agricultural Robotic Platforms. International Journal of Academic Engineering Research, 6(4), 14-21.

28. Lyashenko, V., & et al.. (2021). Modern Walking Robots: A Brief Overview. International Journal of Recent Technology and Applied Science, 3(2), 32-39.

29. Sotnik, S., & et al.. (2022). Overview of Innovative Walking Robots. International Journal of Academic Engineering Research, 6(4), 3-7.

30. Lyubchenko, V., & et al.. (2016). Digital image processing techniques for detection and diagnosis of fish diseases. International Journal of Advanced Research in Computer Science and Software Engineering, 6(7), 79-83.

31. Vasiurenko, O., Lyashenko, V., Baranova, V., & Deineko, Z. (2020). Spatial-Temporal Analysis the Dynamics of Changes on the Foreign Exchange Market: an Empirical Estimates from Ukraine. Journal of Asian Multicultural Research for Economy and Management Study, 1(2), 1-6.

32. Lyashenko, V., Matarneh, R., & Kobylin, O. (2016). Contrast modification as a tool to study the structure of blood components. Journal of Environmental Science, Computer Science and Engineering & Technology, 5(3), 150-160.

33. Lyashenko, V. V., Matarneh, R., Kobylin, O., & Putyatin, Y. P. (2016). Contour Detection and Allocation for Cytological Images Using Wavelet Analysis Methodology. International Journal, 4(1), 85-94.

34. Babker, A., & Lyashenko, V. (2018). Identification of megaloblastic anemia cells through the use of image processing techniques. Int J Clin Biomed Res, 4, 1-5.

35. Babker, A., Sotnik, S., & Lyashenko, V. (2018). Polymeric Materials in Medicine. Sch. J. Appl. Med Sci, 6, 148-153.

36. Kuzomin, O., Lyashenko, V., Dudka, O., Radchenko, V., & Vasylenko, O. (2020). Using of ontologies for building databases and knowledge bases for consequences management of emergency. International Journal of Advanced Trends in Computer Science and Engineering, 9(4), 5040-5045.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-3

37. Khan, A., Ahmad, M., Joshi, S., & Lyashenko, V. (2016). Synthesis of Alumina Fibre by Annealing Method using Coir Fibre. American Chemical Science Journal, 15(2), 1-7.

38. Elasri, A., & et al. (2021). Multi-robot Visual Navigation Structure Based on Lukas-Kanade Algorithm. In International Conference on Artificial Intelligence and its Applications, Cham: Springer International Publishing 534-547.

39. Zhao, Y., & et al. (2021). Deep lucas-kanade homography for multimodal image alignment. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 15950-15959).

40. Elasri, A., & et al. (2023). Mobile Robot Visual Navigation Systems using Optical Flow. In 2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM), IEEE, 1, pp. 1-6.

41. Xue, Z. (2023). Analysis of simultaneous localization and mapping technology for mobile robot based on binocular vision. In 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), IEEE, 1735-1739.

42. Wu, L., & et al. (2022). Multi-robot collaborative SLAM Based on optimized ORB. In 2022 China Automation Congress (CAC), IEEE, 3578-3583.

43. Bahnam, S., & et al. (2021). Stereo visual inertial odometry for robots with limited computational resources. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 9154-9159.

44. Leonida, K. L., & et al. (2022). A Motion-Based Tracking System Using the Lucas-Kanade Optical Flow Method. In 2022 14th International Conference on Computer and Automation Engineering (ICCAE), IEEE, 86-90.

45. Abdelaziz, S. I. K., & et al. (2020). Low-cost indoor vision-based navigation for mobile robots. In Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020), 2560-2568.

46. Blachut, K., & Kryjak, T. (2022). Real-time efficient fpga implementation of the multi-scale lucas-kanade and horn-schunck optical flow algorithms for a 4k video stream. Sensors, *22*(13), 5017.

47. Hannum, C., & et al. (2020). Trust or Not?: A Computational Robot-Trusting-Human Model for Human-Robot Collaborative Tasks. In 2020 IEEE International Conference on Big Data (Big Data), IEEE, 5689-5691.