First-Person Vision Robot Guidance Using Pointing Gesture

Alex Iliarski
iliar004@umn.edu

Isaac Berlin
berli113@umn.edu

***Abstract*—With the recent technological developments of Augmented Reality (AR) headset devices, such as the Apple Vision Pro and Meta Orion, there is increasing applicability and interest in first-person vision (FPV) techniques. These innovations open the door to extended human-robot interaction, particularly using human perspectives to guide robots through dynamic environments. FPV-based navigation can have applications in a wide range of industrial or household robotics applications, with robots navigating primarily on human visual input. We propose a novel methodology to control a robot based entirely on human FPV input. Our work focuses on guidance to a goal object, particularly using a pointing gesture to direct the robot's target destination. The goal of this work is to develop a more natural robot control system that fits seamlessly into AR technology by leveraging real-time human visual input.**

***Index Terms*—First-person vision, robot guidance, pointing gestures, human-robot interaction, computer vision.**

I. Introduction

First-person vision (FPV) is quickly becoming an integral part of emerging and innovative human-computer interaction technologies. With the advancement of augmented reality (AR) technology, such as the Apple Vision Pro and Meta Orion, the potential for human FPV data to be used in a variety of applications has exploded. In particular, FPV could be used to provide a more natural and intuitive mechanism for human-robot interaction. This provides an opportunity to develop robotic systems that rely entirely on human visual input. These methods could have a bevy of possible applications, particularly in assistive household robotics.

Despite the rapid advancement of AR and gesture recognition technologies, the application of FPV for guiding robots remains unexplored. Most existing systems instead rely upon fixed external cameras or robot-mounted cameras, along with potentially additional sensors or prior knowledge of the environment. These approaches are effective but can have limited applicability in some real-world scenarios. Additionally, although gesture recognition and pose recognition have been extensively studied, there is much less research into this recognition from an FPV perspective in the field of egocentric vision. These technologies have yet to be combined into a single application for FPV robotic control, which is the gap to be addressed here. This presents some unique challenges, such as camera instability, real-time processing, and FPV gesture recognition.

In this work, we develop a novel approach for robot guidance that utilizes solely FPV input, along with a pointing gesture, to guide a robot to a goal object. This seems to be a natural starting point for FPV-based human-robot interaction. For example, take the scenario in a household where a human wants to instruct a robot to pick up and clean a particular piece of trash on the floor. The robot could then move to the item and retrieve it.

Our methodology can be split into three main problems. First, we must identify the robot and its location within the camera input. Second, we must determine the orientation of the robot within the frame, since this will be a key piece of information when wanting to move the robot to the goal object. Finally, we must be able to "select" the goal, or target, object that we desire the robot to move towards. We combine existing methods and techniques to accomplish these tasks and create a successful, but slightly inconsistent, mechanism for robot guidance to a goal object.

This paper makes three key contributions and insights. First, a limited framework for using exclusively FPV and pointing gestures for robot control is developed that can be expanded upon. Second, we highlight key challenges posed by this problem, as well as some insights into how to address them. Third, we provide an analysis of the feasibility and effectiveness of this approach in a simplified and obstacle-free environment, which can be useful information when attempting to expand this work to a more complex situation. By developing this system and analyzing its results, we aim to advance the field of human-robot interaction and pave the way for FPV-based robotic systems with AR technologies.

The rest of this paper is organized as follows. Section 2, the problem description, describes the problem, experimental setup and its constraints, and notes a few of the inherent challenges presented by this problem. Section 3, the related works, discusses the related research on this topic, as well as the methodologies and technologies that are leveraged in our implementation of this program. Section 4. the results and insights section, discusses the methodology of our implementation, how our implementation performed, and how it affected the results. Section 5 details the lessons learned and main takeaways from this report. Section 6 discusses future work and the potential ways that our work here can be expanded upon. Section 7, the problems encountered section, describes the main challenges encountered in this process. Finally, section 8, concludes this report by review the purpose and results of our study.

Fig. 1. Apparatus for capturing FPV footage.

II. Problem Description

With the advancement of AR technology, such as the Apple Vision Pro and the Meta Orion, there is an increasing availability of human point of view (POV) data and increasing applicability of FPV-based technologies and methods. This can provide for a more natural mechanism for human-robot interaction, as opposed to traditional methods like mouse, keyboard, and controllers. The field of human-computer interaction has started to look at humans themselves as the source of input. We, in particular, will explore how human POV input alone can be used to conduct simple robot guidance. We propose an experimental setup as follows: a robot without any sensory equipment and a target object both on a level plane with no obstacles, as well as a camera attached to a human's head, with both the robot and the target destination object in the frame. The human controller will point at the target object to identify the robot's intended destination and drop their hand from the camera's frame, simulating a natural situation where a human operator instructs a robot where to go. Then, the robot navigates to the target destination using only the input from the FPV camera. Our method for capturing human POV footage is seen in Figure 1, using a simple webcam attached to a pair of glasses. This will be a challenging task since there is minimal research surrounding robot guidance from an external first-person perspective. Additionally, with a camera attached to a human perspective, there will naturally be camera instability. However, we believe that this is a useful technology to explore and develop for the bevy of potential uses it could have when integrating with AR headset technology.

We apply some additional constraints to the experiment to keep the project scope simple and manageable. Notably, we ensure a level plane for the robot and goal objects to operate in without any obstacles. Next, we ensure that the environment is static, so the goal objects are not moving, and the only moving object is the robot itself. Additionally, we ensure that all potential goal objects and the robot can be captured within the camera's view, which limits the distances in the environment. As such, we kept all objects within a 5-meter radius of the robot's initial position. Another constraint is that no other humans should be visible in the frame. This is important as it relates to tracking the tip of the index finger for the pointing gesture. Lastly, we ensure that all potential goal objects in the camera's view are unique objects. This is important due to our usage of a YOLO (You Only Look Once) model for goal detection.

Lastly, we note that the robot used for this project is the Turtlebot3. This allows for simple control of the robot through ROS and the ability to direct its movement. Note that we do not use any of the onboard sensors on the Turtlebot3, since we deliberately intend to use solely the FPV footage to guide the robot to its destination.

III. Related Work

Our study involves both human-computer interaction using hand gestures and overhead robotic control. To inform our design choices we conducted a review of literature from both fields. In the field of human-computer interaction, pose estimation[1][2] and hand gesture recognition[3] are a well-established topic supported by extensive research with a wide range of solutions[4][5]. Real-time implementation of these tasks has improved in recent years due to efficiency and accuracy increases in state-of-the-art models. Notable examples of these include Google's MediaPipe Hands[6] and Carnegie Mellon University's OpenPose[7], both of which are pipelines for advanced pose estimation and gesture recognition. These technologies allow the accurate mapping of human hands within an image, providing a foundation for inferring valuable information. This information enables a more intuitive approach to human interaction with computers and robots. Similarly, Raheja et al[8] demonstrated the ability to control a robot arm using only hand gestures and Mondal et al[9] showed human hand gestures being used to control a robot navigating through an unstructured environment. Pointing as a goal-selection method has also been explored in previous studies. Tölgyessy et al[10] and Abidi et al[11] both demonstrated the ability to use a pointing gesture to choose an arbitrary target and navigate to said goal with a mobile robot. However, in both of these approaches, a camera system is placed on the robot, not in a human first-person point of view.

Following the selection of the goal object, the subsequent task is navigation using the FPV perspective. Much of our knowledge about robot navigation and control from an FPV view builds off the more heavily researched subject of robotic control from an overhead camera. Our assumption that the FPV view behaves similarly to the overhead view relies on a few factors. Overhead control algorithms often rely on camera calibration and frame transformations to plot the path of a mobile robot[12], but due to the inherent instability of an FPV view, we must use an uncalibrated camera. This problem of using an uncalibrated camera often leads to a wide array of possible solutions. Rau et al[13] propose using strategies that rely on partial path planning strategies with a feedback loop to reach a desired position. They combined this approach with the strategy of computing a homography, or image transformation, that projects the uncalibrated overhead image onto the ground plane. They then completed pathfinding based on their transformed image's differences between frames. Liang

et al[14] proposed a strategy of displaying points of interest on top of a mobile robot. They then used the location of all three of these points of interest within an uncalibrated image to map the kinematics of their mobile robot and implement path planning and trajectory generation. By using a combination of information from the above strategies, particularly for first-person point of view gesturing for goal selection and uncalibrated overhead camera control of a mobile robot, we developed a novel solution to our problem.

IV. Results and Insights

The methodology with which we implemented our FPV-guided robot consisted of three main subproblems: determining the Turtlebot3's position, determining the Turtlebot3's orientation, and creating a mechanism for goal selection and tracking the position of the goal. Once these three key pieces of information were able to be tracked for each frame of camera input, a simple algorithm was developed to navigate the Turtlebot3 to its destination. Note that the implementation of our FPV guidance was done in ROS and Python, making use of libraries and packages such as opencv, mediapipe, and ultralytics.

A. Turtlebot3 Location Tracking

First, a mechanism was developed to track the Turtlebot3's location in a frame of camera input. For this, we made use of a You Only Look Once (YOLO) model, which is able to predict a bounding box surrounding the Turtlebot3 in an image. First proposed by Redmon, Divvala, Girshick, and Farhadi in 2016, YOLO is an especially useful and fast general-purpose object detector [15]. By training this model on a dataset of images with bounding boxes, we are able to achieve extremely strong results. Additionally, YOLO is especially useful for object detection in a real-time processing environment, since an entire image is processed in a single forward pass. This makes YOLO an excellent choice of model for this scenario, considering the substantial computational challenge of determining where the Turtlebot3 is continuously for a stream of image input. We thus used a YoloV8 Nano model with 3.2 million parameters that was previously trained on the COCO dataset. We then fine-tuned this model further on the Turtlebot3 object detection dataset [16]. An example of a training sample can be seen in Figure 2. We obtained a mean average precision (mAP) of 0.956 and a mean average recall (mAR) of 0.981, which indicates that our model was successfully able to learn to detect a Turtlebot3. We even saw good results when parts of the Turtlebot3 were occluded due to our method of orientation estimation.

Fig. 2. Example image used to train the TurtleBot3 detection YOLO model.

B. Turtlebot3 Orientation Estimation

Next, we developed an approach to determining the Turtlebot3's orientation. This is a key task for our problem and is quite a challenging task in computer vision. Some existing methodologies include feature-based methods, relying on extracting a particular feature of the object from an image and geometric techniques to determine its orientation[17], as well as various deep learning methods and pose estimation techniques[18]. We proceeded with a simple version of the first approach. As such, we added a strip of blue cardboard along the forward-backward axis of the Turtlebot3. Thus, when a human looks at the Turtlebot3 from above, the long axis of this cardboard is assumed to be the forward-backward orientation of the robot.

Specifically, we applied techniques from computer vision to convert the raw image of the Turtlebot3 with a blue strip on top into a piece of particularly useful orientation information. First, we cropped the full image to include just 1.5 times the Turtlebot3's bounding box, as determined by the YOLO model in the previous step. Next, we masked this image for blue pixels. Then, morphological transformation methods were used to reduce noise, namely erosion and dilation[19]. Next, we drew a contour around the largest area of blue pixels, assumed it to be roughly rectangular, and extracted the largest edge. This largest edge was then used as our orientation line. The full approach is seen in Figure 3. This method worked somewhat well but had limitations in low-light environments since it is difficult to detect blue pixels accurately. Additionally, this approach experienced some instability, with other lines along the cardboard strip being detected, instead of the longest line, especially when looking at the Turtlebot3 from certain angles. Despite this, our approach ended up being robust enough to accomplish our task at hand.

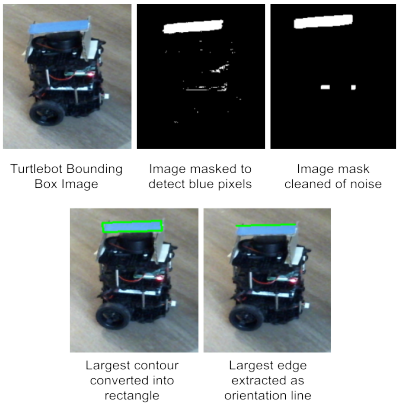

Fig. 3. Five steps for Turtlebot3 orientation detection.

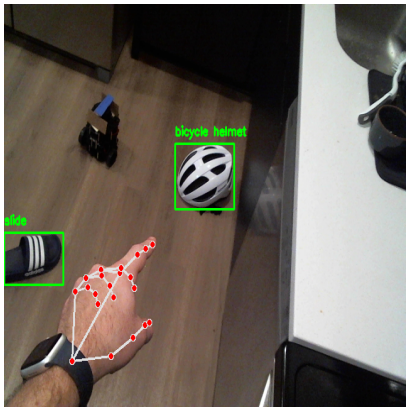

Fig. 4. Image demonstrating pointing mechanism to select goal object.

C. Goal Object Selection using Pointing Gesture

Lastly, we developed a mechanism for the user to select a goal object using a pointing gesture. This included training a YOLO model on five potential goal objects: a bicycle helmet, a water bottle, a shaker bottle, a flip-flop, and a basketball. This is why we included the constraint in the experiment that all potential goal objects in the human's POV should be distinct since the model will only predict at most one bounding box for each object type. We attempted to develop a generic object detection methodology that draws bounding boxes around all objects by using Canny edge detection but found it to be wildly inconsistent. Another possible approach would be to leverage deep learning for generic object detection, as described in [20], but we decided to limit the scope of our study by instead only allowing the previously mentioned potential goal objects.

Once we developed the YOLO model to identify all potential goal objects within an image, we still needed to implement the capability to select a particular object as the goal object using a pointing gesture. We made use of the previously mentioned MediaPipe Hands library, developed by Google, to conduct human pose detection and tracking[21]. In particular, we tracked the hand of the user in order to find the point representing the tip of the user's index finger. Once this point is tracked successfully, we determine a goal object to select if the index finger is closest to that object's bounding box for a period of at least two seconds. An example of this process is seen in Figure 4. Note how well this image is able to capture all the joint points on a human hand, even capturing the approximate location of the joints on the fingers curled beneath the hand, which are not even visible to the camera.

This ended up being a particularly effective approach that seldom failed, especially when first showing a fully extended hand to the camera so that it is easier to establish a track of all of its points, and then begin to point to the desired object. One limitation of this approach is that the MediaPipe library will attempt to determine the pose of any human within the frame. Thus, if another hand is visible in the frame, other than the user's hand, it could track the wrong tip of the index finger. Otherwise, keeping a simple environment with no other humans, as established by the experimental constraints, the pointing gesture to select a goal object is highly successful.

D. Turtlebot3 Guidance Algorithm

Now that these three key pieces of information have been established, the location of the Turtlebot3, the location of the goal, and the orientation of the Turtlebot3, we developed a simple algorithm for the robot to navigate to the goal. The first step is to build a triangle between the goal, the Turtlebot3, and the end of the line representing the Turtlebot3's orientation. We then rotated the Turtlebot3 until the angle of this triangle fell below 5 degrees, which is our angle tolerance. This process is seen in Figure 5 and Figure 6. Note how the angle of the triangle is reduced from 59.93 degrees to 3.74 degrees.

Fig. 5. Initial position of Turtlebot3 not in line with goal.

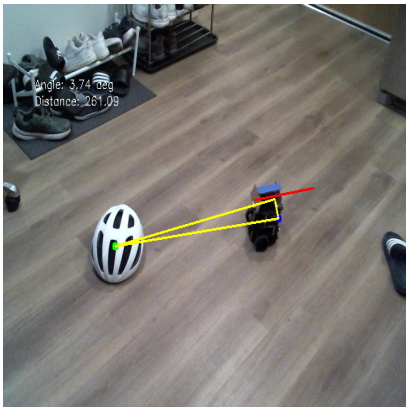

Fig. 6. Reduced angle after Turtlebot3 rotates to be in line with goal.

Now, we know that the goal is directly in front of the Turtlebot3 or directly behind the Turtlebot3. Note that we do not care what direction the Turtlebot3 faces when it reaches the goal object, as long as it makes it there. Hence, our next step was to conduct a calibration step to determine whether the goal is forward or backward. When determining the distance from one object to another we used the euclidean distance as our measurement of measurement. We briefly move the Turtlebot3 forward, determine if the center of the Turtlebot3 bounding box has gotten closer to the center of the goal bounding box, and use this information to appropriately assign the direction the Turtlebot3 should travel to reach the goal. After this calibration step, the Turtlebot3 is free to move to the goal in the appropriate direction. If, along the route, the angle is greater than five degrees, this will be recognized and adjusted appropriately.

E. Results

With this methodology, the Turtlebot3 reaches the goal object in 12 out of 20 trials. In the successful trials, it took approximately 20-30 seconds to reach the goal object situated about 1 meter away from the Turtlebot3's starting position. These results are promising, but do highlight some key shortcomings. Mainly, our implementation struggled in conducting the forward-backward calibration step. It seems the bounding boxes tracking the Turtlebot3 and the goal object have some instability to them from frame to frame and this can distort the resulting distance calculation between their centers. Additionally, the other main source of failure was instability in the orientation detection. Although most frames would have the orientation facing in the correct direction, there are frames in which the orientation line is drawn incorrectly. This caused instability in the running of the program and ultimately led to some failures.

An extremely promising result is the usage of the pointing gesture to select the goal object. This succeeded on the first attempt in 18 out of 20 trials and succeeded on the second attempt in the two failures. The MediaPipe Hands library is very successful at tracking points on the human hand, especially one that is so close to the camera. By leveraging the approach of tracking the tip of the index finger and identifying its closest goal bounding box, we were able to achieve highly promising results. We believe that pointing, and other forms of gesturing with hands, have significant potential for applications in FPV systems.

V. Lessons Learned

This project provided many valuable insights into the challenges and opportunities in FPV-based robot control using human gesturing. The main takeaway is that this implementation demonstrates that guiding a robot using solely FPV input and pointing gestures is a feasible approach, although it carries certain limitations due to the inherent instability of the camera. Namely, inaccuracies in bounding box detection and orientation estimation cause some inconsistencies and failures in performance. This suggests that the development of more stable mechanisms and algorithms to handle these objectives is the main obstacle to the practical deployment of FPV-based robot control systems.

Real-time processing also emerged as a limitation of our implementation. Running three models, the Turtlebot3 detection, the goal detection, and the hand detection, proved to be a bottleneck in computation. The need for continuous image processing, analysis, and decision-making places substantial real-time computation demands on the system. Thus, computation must be run on a high-performance computer, rather than on the robot's onboard computer. Additionally, this highlights the importance of optimizing algorithms such that performance can be accelerated in real-world situations.

Orientation detection also posed a significant challenge. Our approach, which relied on a distinct feature, the blue strip, worked but had inaccuracy and instability. In particular, it struggled in low-light environments and when viewed from specific angles. This suggests that a more sophisticated approach is needed, such as utilizing deep learning techniques for pose estimation, to improve the reliability and accuracy of orientation detection in FPV systems.

One of the most promising results of this project was the success of FPV gesturing. The use of MediaPipe Hands to track the tip of the index finger to select goal objects proved to be both intuitive and effective. This usage of gesturing in FPV applications validates the applicability of its potential use as an intuitive interface for human-robot interaction. These results could be expanded to include other forms of gesture recognition, such as sign language recognition in FPV, or other ways to issue commands. This takeaway goes beyond just robot applications, as it could be used for control in a variety of AR applications where FPV input is used. Gesturing with a hand provides a natural and intuitive method for a wide range of applications, from industrial to household environments. This reinforces the idea that FPV and AR technologies can seamlessly integrate to create a natural method for human-computer and human-robot interaction.

It is worth noting that applying significant constraints to the environment and keeping a small area with no obstacles, greatly enhances the performance of the system. These conditions were vital for the success of the methodology

proposed here. However, improvements to the methodology could certainly allow for application to more complex, real-world scenarios. In particular, this demonstrates the importance of developing systems that can dynamically generalize to environments of varying complexities.

Finally, the findings of this project highlight the need for a more robust goal object selection system. While our YOLO model implementation worked extremely well, it only allows the selection of objects that the YOLO model has been trained on. Thus, it is not very robust. An accurate generic object detection system would allow for any object that is pointed at to be selected as a goal. Then, there remains the issue of how to continue to track this particular generic object once it is selected since our approach simply recalls the type of object that was selected as the goal. This highlights an existing gap in computer vision that could have significant implications for this sort of application.

These lessons highlight the significant potential of FPV-based systems for human-robot interaction while also identifying some key areas for improvement and further research. These insights lay the foundation for a more robust, stable, and adaptable FPV robotic guidance system.

VI. Future Work

The results and lessons learned from this project indicate several paths for future work to improve the performance, robustness, reliability, and applicability of FPV-based robot guidance systems.

One critical area for improvement is orientation detection. Future work should be done to explore sophisticated orientation estimation techniques. This could include deep learning pose estimation or a different physical method for feature recognition. One proposed approach we make is to use a UV light or laser that points forward from the robot. This is something that would not disturb a human, since it is outside the visible light spectrum, but it could be easily picked up with more sophisticated cameras. This light would provide a simple method to estimate the robot's forward direction. Additionally, since the light would only be visible on one side of the robot, we can conclusively use this line of light to distinguish between the forward and backward directions of the robot.

Another improvement could be made by making use of a depth camera, instead of a traditional one. This is not an unreasonable expectation, since many AR headsets already use depth cameras to place virtual objects in the scene. The additional depth data could help to address many issues encountered in the current approach. For example, depth data could help to determine robot orientation. They can also make detection of generic objects easier since these objects would clearly stand out from the floor. Additionally, depth data could help with obstacle avoidance in a scenario more complex than the one discussed here.

A more robust goal selection would also be an area for significant potential enhancement. Our approach only allows for the selection of one of five possible goal objects, due to our YOLO model approach. Our methodology then remembers this label for the goal object (bicycle helmet, water bottle, etc), and then searches for such an object in all future frames of image input, rather than tracking a particular instance of an object. A more robust and generalized approach would allow for any object to be selected as a goal. Thus, a different approach, instead of a YOLO model, would likely provide more generalizable results. One proposed solution is the above-mentioned depth camera. Alternatively, there is substantial research being done using deep-learning approaches for generic object detection. In particular, Fast R-CNNs, SPP-net, R-FCN, or FPN models have been successful for generic object detection[22]. Integrating these deep learning techniques into our system to build bounding boxes surrounding all objects in an image would allow for a much more robust selection of potential goal objects. Then, of course, this particular object would need to be tracked and remembered, once it is selected. This presents another source of potential research, as this object can be "remembered" in a few different ways, including, but not limited to, instantly training a YOLO model on images of this object or recalling its location in 2D/3D space relative to other objects.

Complex environments are certainly more realistic than the simple environment discussed here, which was obstacle-free and static. Expanding the system's capabilities to handle complex environments by implementing obstacle avoidance and handling potentially moving objects would give the system much more flexibility and applicability. Additionally, incorporating SLAM (Simultaneous Localization and Mapping) techniques or other environmental mapping methods through depth cameras or AR would certainly enhance the system's versatility.

Another source of future work can be developing additional FPV-based gesture recognition methods. This task should not prove to be beyond current capabilities, given the success of current third-person gesture recognition and pose estimation, particularly with MediaPipe. Specifically, usage of hand gestures or sign language from an FPV perspective can be a useful source of user input to a system, particularly in AR applications. Additionally, the development of this technology could improve quality of life for deaf, non-verbal, or hard-of-hearing users, as they may prefer using hand gestures to provide input rather than voice commands.

Lastly, future work should explore integrating with an AR headset. This could enhance the intuitive experience for the user, especially with the ability to provide visual feedback to the user through visual output on the headset. An advanced user interface can be developed to further realize the benefits of robot guidance from FPV, including highlighting goal objects in green and charting the robot's projected path. Such a progression feels like a natural next step in human-robot interaction, leveraging the rapid advancement of AR technology.

In summary, there is a bevy of potential future work to be done in order to create a sophisticated and reliable methodology for FPV-based robot guidance, especially given

the minimal research into FPV-based techniques in computer vision and robotics. However, advancements in this field will ultimately be extremely rewarding, given the significant expected increase in users of AR headsets and availability of FPV data.

VII. Problems Encountered

One of the main challenges we faced was developing a mechanism for orientation estimation and detection. We were hesitant to augment the Turtlebot3 in any way, to keep a "pure" implementation of our system. Thus, originally, we attempted to use the features of the Turtlebot3, such as the wheels being in the front, to determine its orientation. However, this proved to be too difficult, and the camera often couldn't find the wheels with much accuracy, especially at a distance. So, we moved on from this approach and attempted approaches that augmented the Turtlebot3's features in some way. We first tried an approach similar to what is described in [14], where three differently colored dots were used to determine the robot's orientation in an overhead camera situation. We slightly augmented their approach, instead using four differently colored strips of cardboard, each facing in one of the Turtlebot3's cardinal directions of forward, backward, left, and right. This approach ended up being too unstable, since it was trying to find all four colored strips at all times, and often predicted incorrect lines. Thus, we simplified this approach heavily, and instead used a single blue strip to demarcate the forward-backward direction of the Turtlebot3. This new approach was much more successful than the ones we had tried previously but was still prone to some instability and error.

Another challenge we faced was attempting to develop our system in a simulation environment. We were hoping to develop some proof-of-concept confidence by simulating our FPV robot guidance system in Gazebo, before beginning to test on an actual Turtlebot3. However, there were substantial limitations to the simulation environment that ultimately made this unfeasible. Firstly, it is difficult to model the human head's instability very well in simulation. Mainly, though, using the simulation failed because the YOLO model we had trained to identify the Turtlebot3 simply did not work in the simulation environment. This made it entirely impossible for us to test our implementation and served as a significant time-sink as we attempted to resolve this issue in various ways.

VIII. Conclusion

In this paper, we introduced a novel methodology for mobile robot guidance from a first-person perspective using human pointing gestures. By using a human's first-person perspective, a more intuitive approach to robot control is presented, particularly in the context of emerging AR technologies. Our approach makes use of real-time image processing and computer vision techniques to interpret the user's point gesture and create a plan for the robot to reach its intended target. By implementing a prototype system using a Turtlebot3 and solely FPV input, we have demonstrated the feasibility of an FPV-based paradigm for robot control.

The results of our approach demonstrated that the pointing gesture to select a goal object was very successful, allowing for a simple and reliable method for human-computer interaction. However, other parts of our methodology were inconsistent, highlighting some of the key challenges of using exclusively FPV imagery for robot guidance. In particular, we faced challenges in robot orientation estimation, real-time processing constraints, and camera instability. Despite the challenges, our approach proves the potential for FPV-guided robot systems as a more natural, intuitive, and human-centered approach to robot control.

Our study also highlights key areas for future research. In particular, more work should be done to improve orientation estimation, develop algorithms that are optimized for real-time processing, and create generalized approaches to goal selection and object detection in complex environments. Additionally, we believe that integrating depth-sensing technologies or AR headsets is a natural next step to providing a more robust, accurate, and immersive user experience in FPV-based robot guidance systems. We also see significant potential in gesture recognition from an FPV perspective, even excluding a robotics context. This is a significant frontier in human-computer interaction, and an expanded framework that could recognize sign language and other gestures would allow for a robust method for human input.

Ultimately, this study contributes to the rapidly growing field of FPV and AR-driven human-robot and human-computer interaction, offering a groundwork for the future development of intuitive robot and computer systems. By refining the methodology proposed here and conducting further research to address its current limitations, we believe that FPV-based systems will become a valuable tool for a wide range of applications, creating a more intuitive form of human interaction.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[2] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[3] Y. Fang, K. Wang, J. Cheng, and H. Lu, "A real-time hand gesture recognition method," in *2007 IEEE International Conference on Multimedia and Expo*, 2007, pp. 995–998.

[4] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *ACM Comput. Surv.*, vol. 56, no. 1, Aug. 2023. [Online]. Available: https://doi.org/10.1145/3603618

[5] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," in *2012 IEEE RO-MAN:*

The 21st IEEE International Symposium on Robot and Human Interactive Communication, 2012, pp. 411–417.

[6] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," 2020. [Online]. Available: https://arxiv.org/abs/2006.10214

[7] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," 2019. [Online]. Available: https://arxiv.org/abs/1812.08008

[8] J. Raheja, R. Shyam, A. Rajsekhar, and P. Prasad, "Real-time robotic hand control using hand gestures," 02 2012.

[9] A. Mondal, S. Paul, S. Ghosh, and A. Chatterjee, "Hand gesture controlled mobile robot navigation guidance using constant-q transform," in *2022 IEEE 6th International Conference on Condition Assessment Techniques in Electrical Systems (CATCON)*, 2022, pp. 328–333.

[10] M. Tölgyessy, M. Dekan, F. Duchoň, J. Rodina, P. Hubinský, and L. Chovanec, "Foundations of visual linear human–robot interaction via pointing gesture navigation," *International Journal of Social Robotics*, vol. 9, pp. 1–15, 09 2017.

[11] S. Abidi, M. Williams, and B. Johnston, "Human pointing as a robot directive," pp. 67–68, 03 2013.

[12] Z. Ziaei, R. Oftadeh, and J. Mattila, "Global path planning with obstacle avoidance for omnidirectional mobile robot using overhead camera," in *2014 IEEE International Conference on Mechatronics and Automation*, 2014, pp. 697–704.

[13] R. Rao, C. Taylor, and V. Kumar, "Experiments in robot control from uncalibrated overhead imagery," in *Experimental Robotics IX*, M. H. Ang and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 491–500.

[14] X. Liang, H. Wang, Y.-H. Liu, B. You, Z. Liu, and W. Chen, "Calibration-free image-based trajectory tracking control of mobile robots with an overhead camera," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2, pp. 933–946, 2020.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[16] TurtleBot3, "Turtlebot3 dataset," https://universe.roboflow.com/turtlebot3-o3l1b/turtlebot3-6fviu, jun 2024, visited on 2024-12-10. [Online]. Available: https://universe.roboflow.com/turtlebot3-o3l1b/turtlebot3-6fviu

[17] N. Stojanović, V. Pantić, V. Damjanović, and S. Vukmirović, "3d vehicle pose estimation from an image using geometry," in *2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2022, pp. 1–6.

[18] J. Liu, Y. Gu, and S. Kamijo, "Joint customer pose and orientation estimation using deep neural network from surveillance camera," in *2016 IEEE International Symposium on Multimedia (ISM)*, 2016, pp. 216–221.

[19] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, pp. 532–550, 1987.

[20] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," 2019. [Online]. Available: https://arxiv.org/abs/1809.02165

[21] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for building perception pipelines," 2019. [Online]. Available: https://arxiv.org/abs/1906.08172

[22] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.