From equations to actions: A system-level design research experience of an undergraduate student

Vincenzo A. Macri, Dr. Daniel Raviv and Juan D. Yepes
Florida Atlantic University / Florida Atlantic University / Florida Atlantic University

Abstract

With the trend of autonomous vehicles and world sensing devices becoming more widespread, the importance of learning computer vision is becoming clearer. In the standard academic environment, the average undergraduate student is deprived of the opportunity to engage in research and experience real world problem solving specifically on a hands-on, system level within the computer vision space. This paper outlines the learning journey of an undergraduate student studying the basics and practical applications of a novel computer vision algorithm called visual looming, highlighting the challenges faced along the way and the ways those challenges were overcome. In this paper we share the learning approach, skills acquired, and knowledge gained.

Entering this study with no prior knowledge of visual looming and only basic computer vision knowledge, the student was given the opportunity to discover unique and creative approaches to solving an open-ended problem without the bias of past literature. Using this self-guided approach to learning, the student was able to produce several divergent solutions using different computer vision techniques to implement the looming algorithm on a NVIDIA processor located on a motorized 3-wheeled rover. Through the constant prototyping of these different solutions, the student ran into many challenges on both the hardware and software level which required out-of-the-box thinking and real-world problem-solving skills to overcome. With the guidance of a professor and a doctoral student, each with over 25 years of computer vision experience, the student learned how to use available resources to navigate through hurdles in the design process and eventually converge on a final design solution.

By the end of the process, the student was able to successfully develop a system-level solution to demonstrate the visual looming algorithm in practice by having the 3-wheeled rover keep a constant distance away from a moving object, using a feedback control loop to govern the rover's movement based on the calculated looming values. This system-level, end-to-end project required a plethora of skills including the ability to look at the bigger picture, the understanding of hardware and software interactions, the clever use of tools like GitHub and Discord for collaboration, the ability to receive and use constructive comments to enhance the final design and all around creative, and out-of-the-box divergent thinking.

We will explore in great depth many of these skills the student learned through this experience that may be beneficial to other instructors looking to bridge the gap between the academic classroom and real-world problem solving. Altogether, this paper will shed light on the inner details of a self-guided learning experience and demonstrate the benefits of creative exploration and hands-on experimentation when learning and applying a new topic.

Keywords

Computer Vision, Visual Looming, Divergent Thinking, Exploratory Learning, System Design

Introduction

The art of robust computer vision algorithm development is rapidly becoming a more valuable and needed skill in both research and production environments. From autonomous vehicle control systems to facial detection security applications, computer vision has promise in many fields. Despite the growth in the demand of computer vision, there is little to no noticeable growth in the magnitude of curriculum being taught in the classroom on production level computer vision techniques and applications.

In researching for literature review, it was difficult to find many prior works that were directly relevant to our study, given the unique computer vision problem domain of our research and the individualized undergraduate student design experience described in this paper. In (Raviv and Radzins, 2014) a successful design experience is shared of an undergraduate student creating novel speed bump alternatives, highlighting the multistep design process used and encouraging innovation and creativity in the academic environment. Similarly, (Sawatzki, 2021) shares the benefits of a hands-on engagement that comes with the inclusion of project based design curriculum in the engineering classroom, with a focus on low-cost kit based projects. Like our research, (Raviv and Radzins, 2014) describe the learning process and outcomes of an undergraduate research experience and (Sawatzki, 2021) shows the importance of design integration into undergraduate curriculum. However, unlike the subjects in both studies, in this paper the student tackles a highly technical problem and produces not only a single component, but a full system-level design through the convergence of a single idea from divergent prototyping. Additionally, this paper is unique in our approach to the design process, intentionally encouraging the student to avoid initial studying of past literature when brainstorming solutions to the problem, allowing for creativity and innovation unbounded by the bias of past findings. We will show an unconventional design project that uses low-cost high-tech hardware from NVIDIA, production grade computer vision techniques and complex control behaviors together to create a complete system.

The average undergraduate student suffers from the lack of exposure to basic computer vision concepts leaving them unprepared for potential opportunities following their graduation. Furthermore, even if the basic concepts of computer vision are taught academically, there is still the intense learning curve of taking the theoretical concepts and applying them to real world problems that is difficult to teach in the classroom setting. We will share our experience to show that an effective way to teach students to solve real world computer vision problems and accelerate this learning curve, is to present open-ended problems with corresponding fundamental equations for them to use their own intuition and skills to develop novel solutions. In this paper we will document the learning journey of an undergraduate student who was selected based on exceptional work ethic and a motivation to create. We will share the student's experience exploring an unorthodox computer vision technique called visual looming through self-guided learning with a goal of creating a system-level solution on production grade hardware. Through this process we saw firsthand how the open-ended problem approach fostered:

- ***Divergent to convergent thinking*** – Initially attacking the problem with many abstract approaches and from those options converging on a final design.

- ***Exploratory learning*** – Encouraging the student to search on their own for possible solutions either within their own skill toolbox or by utilizing university and online resources.

- ***System-level thinking*** – Motivating the student to look at the bigger picture and develop multiple components that work collaboratively to achieve a desired system-level behavior.

- ***Collaboration*** – Allowing the student to bounce ideas off a professor and a doctoral student and learn from their working experience.

We will discuss these concepts in more depth and shed light on other byproducts of this open-ended problem-solving approach. Through the process we will document and though the outcomes we will share, we hope to encourage other educators to include open-ended problems into their curriculum to give their students the opportunity to put their creativity to the test and strengthen their real-world thinking skills.

Motivation and Desired Takeaways

In this section we will discuss our main objectives going into this research from both educational and technical points of view.

1. Educational Perspective

From an educational perspective, the main motivation of this research is to see how an undergraduate student approaches learning a completely new, high-level topic and converges on a final system design from the divergence of exploratory learning. Additionally, we gave the student the opportunity to improve on both technical skills and general problem-solving skills through this research project that would be hard to obtain exposure to in the traditional engineering classroom environment. These skills include:

- ***Hands-on system development skills***: Assembling not just an algorithm or program, but an entire input/output system with a camera as a primary input, computer vision algorithms to process data, and a closed loop feedback control system to govern output motors.

- ***Innovation***: Using an active imagination to brainstorm different ideas and solutions to the problem no matter how unfeasible they may seem, with the goal of exploring all possible solutions.

- ***Divergence to convergence***: Taking many divergent potential solutions to a problem that have been brainstormed and converging to a single final optimized design solution.

- ***Trial and error***: Testing and implementing designs and having the ability to observe an error, formulate a potential solution and utilize available resources to overcome the issue.

- ***Theory to action***: Examining real world problems and noticing where theoretical principles and equations can be applied to uncover solutions.

There are also some additional skills the student was given the opportunity to improve upon and practice that are more common but still not generally taught in the classroom environment. These skills include:

- ***Communication***: Proactively communicating project progress to a mentoring professor and doctoral student through weekly meetings.

- ***Collaboration and documentation***: Using GitHub and Discord to upload and share files between the mentors and the student with the secondary purpose of keeping record of each iteration of project files and progress. Furthermore, publicly uploading all code and files to GitHub meant that others in the community could learn from and build off our work.

- ***Time management***: Utilizing tools like online calendars and to-do lists to prioritize tasks and make sure proper research progress is made each week without sacrificing coursework.

- ***Receiving constructive criticism***: Having an open mind and being flexible to mentor feedback. This goes along with having a growth mindset and a willingness to learn from others.

2. Technical Perspective

From a hands-on technical perspective, the main goal of this research is to develop a system which uses the visual looming algorithm to keep a Jetbot (*Figure 1*) a certain distance away from a given object of interest. The Jetbot is an affordable, 3 wheeled rover that is controlled by an onboard Jetson Nano board, a full self-contained computer developed by the graphics card company NVIDIA. For context and clarity, the visual looming algorithm is derived from the visual looming cue which is described by (Raviv et al., 2000) as the relative expansion of an object's area in the observer's retina as it gets closer. This natural phenomenon is further detailed by (Joarder, 1995) as the negative value of the derivative with respect to time of the relative range between the observer and the object divided by the relative range:

$$L = -(dR/dt)/R$$

Where L is the looming value measured in time^{-1} units, R is the relative range between the observer and the object and d/dt is the first derivative with respect to time. The use of the objects area in an image to calculate looming is constructed from the definition above by (Raviv et al., 2000) and is approximated by the relative temporal change in image area:

$$L \cong (dA/dt)/2A$$

Where A is the objects area as seen from the lens of the camera and d/dt is the first derivative with respect to time. In our code we calculate looming according to this formula during each frame update using a previously recorded area value, and the current area. To make the problem slightly more challenging and encourage the student to think outside of the box, only one camera was given as the input device and the use of machine learning methods for object detection was initially discouraged. This eliminated the possibility for depth estimation through stereo vision techniques and ensured the creation of a well thought out and programmed computer vision-based system as opposed to the "black box" that comes with using a machine learning based technique.

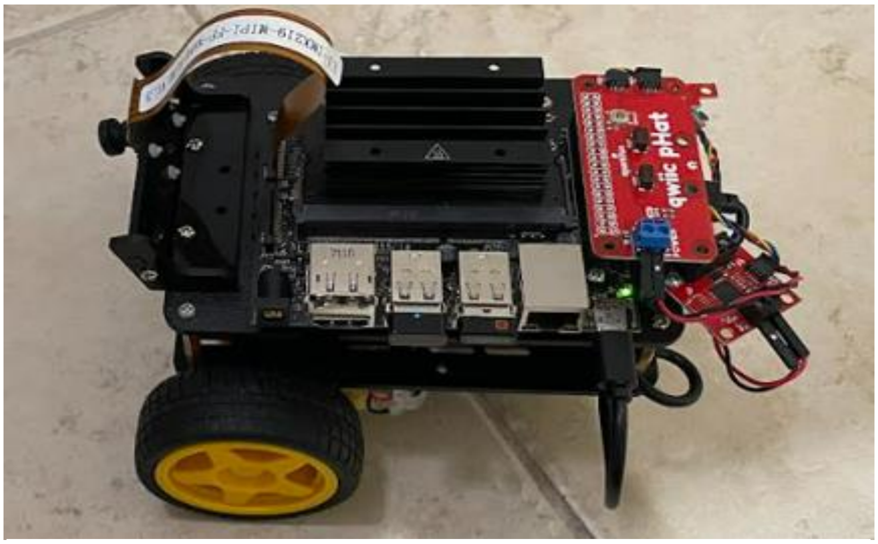

Figure 1: NVIDIA Jetson Nano "Jetbot" Robot

Design Process

The engineering design process is a very active area of research where there have been many different multi-step methods proposed on how to arrive at a final working solution for a given problem. From the 10 step process proposed by (Eide, 1997) which emphasized getting a clear understanding of problem constraints and criteria before prototyping design solutions and selecting a final design, to the multistep process proposed by (Guzey, 2017) which emphasized the importance of building, testing, rebuilding, and retesting, it is evident that the design process is fluid and can be represented in many ways depending on the problem type and solution goals. The method used to arrive at a final system-level design from an initial open-ended problem description, specifically for this research, can be summarized by the following design process, loosely based on the "*creativity and innovation process*" described by (Raviv and Radzins, 2014), which we have broken down into 6 main steps.

1. **Understanding the problem**: Observing the problem from many different angles to obtain a deeper understanding of the problems constraints and a general bigger picture view of the project.

2. **Creative Brainstorming**: Trying to use your own intuition, skills, and past knowledge to creatively solve the problem before doing any heavy research on the topic.

3. **Self-Guided Research**: Turning to past literature to see what has already been done after there are many solid potential original solutions brainstormed for the problem. This step ensures that the original ideas aren't influenced by past works but allows for the knowledge share of past mistakes and solutions.

4. **Divergent Thinking and Prototyping**: Taking only the most feasible ideas from the previous step and beginning to rapidly prototype many alternate candidates for a final design. This step will naturally come with many challenges which will help highlight the pros and cons of different designs and ultimately result in the student learning valuable design lessons.

5. **Converging on a Final Design**: Discovering the strengths and weaknesses of the prototype designs and converging on a single system design to focus on.

6. **Testing and Evaluation**: Using mentor feedback and firsthand observation to ensure that the final design is working as planned and meeting all project requirements.

We will now share an in-depth breakdown for the reader on how we used the 6-step process noted above to go from our problem description to a system-level final design.

1. Understanding the problem:

The problem solving process began with understanding the core problem and its importance. Without an understanding of the problem, it is impossible to start formulating solutions and without a clear understanding of its importance, the motivation behind finding a solution can become unclear. Given the open-ended problem of using the visual looming algorithm to keep a wheeled robot a given distance away from an object, the student immediately observed the domain the problem was given in and realized that a final design would require a full system solution rather than just a single component. Furthermore, with no prior knowledge of visual looming or control theory, the student would be tasked with learning many new skills to successfully complete this design task. The next realization was the intensity of this project, with less than two semesters to complete a final system, the student would need to find a way to balance this ambitious task with fulltime coursework and a part-time job. The student noted that they would only have a single camera to work with, immediately ruling out the use of stereo vision and early on realized they would have to construct a type of control system to constantly check the robot's relative position and orientation and adjust them based on the camera's input. Aside from the initial physical observations about the hardware given, the student also took a step back and observed the bigger picture of why this problem was being given, and its overall importance on a larger scale to autonomous vehicles and object avoidance applications.

2. Creative Brainstorming:

Once the student had a clear idea of the problem they were trying to solve, rather than immediately turning to Google or past literature written in the space, the student made it a point to try and use their own intuition and skills to start thinking about possible solutions. Through the intentional absence of past literature, the student unlocked the full potential of their creativity and was able to brainstorm without the bias of previous findings. Through this process and using past knowledge and skills from coursework, the student was able to come up with many solutions to the individual components of the system, both feasible and unrealistic such as:

a. Using a reinforcement learning model to control the motors of the Jetbot.

b. Developing a machine learning model to detect the position of the object.

c. Using template matching to highlight an objects position and size.

d. Using color segmentation to extract the position and size of the object.

e. Developing a control feedback loop to control the motors of the Jetbot.

3. Self-Guided Research:

After the student unlocked the full potential of their creativity without bias from past literature or internet resources, they began sifting through their initial ideas and turning to past works to see what has been done, and what is doable. After looking more closely at previous works done on the Jetbot hardware, the student found that using machine learning for object detection had already been overdone and had many documented resources. Additionally, the student noticed that using machine learning could over complexify the problem and lead to a black box in the middle of the system flow which we wanted to avoid. For this reason and the yearn to develop a system that was grounded in a completely novel analytic approach, the student decided to scrap any machine learning based ideas that were brainstormed and focused their attention to more basic and intuitive computer vision methods. During this period, the student also assembled the Jetbot using online available instructions and did initial tests on the hardware to ensure everything was working properly before moving on. In doing these tests, the student faced the first major challenge while trying to access the onboard camera through a Python 3 script. The way the camera pipeline was setup, it was only compatible with Python 2 however, all the libraries for controlling the Jetbot's motors were written for Python 3. This presented a major compatibility problem where the student was unable to access both the input and output components from one script. The student would either be required to rewrite all the libraries that control the Jetbot's motors into Python 2 or they would need to figure out a way to access the Jetbot's camera pipeline using Python 3. After turning to Google and hours of sifting through community forums, the student was finally able to find others that were having the same dilemma and discovered a software image for the Jetbot that provided the Python 3 camera pipeline compatibility that was needed. Excited, the student flashed the Jetbot with the image and successfully was able to access the camera from a Python 3 script, checking the last box of the initial tests and allowing for the progression to the next step.

4. Divergent Thinking and Prototyping:

Following the student's self-guided research, it was time for the fun part; prototyping the designs that were brainstormed and diverging into many possible final solutions. In this section we will give a few of the designs the student ended up prototyping, challenges that were faced along the way, and how the student overcame them. To help organize these designs we have separated them into 2 groups of solutions: computer vision designs and control system designs.

a) Computer Vision Designs

***K-means clustering*:** K-means clustering was the first approach the student explored to segment the object and involved clustering similarly colored pixels into groups using built in OpenCV functions in Python 3. This approach ended up failing due to color noise in the background of the video stream leading to unclear cluster groups.

***Template Matching*:** In template matching, an example of the object needing to be tracked can be used to create a filter which when run over the input image as a 2d convolution can be used to generate a feature map that highlights where that filter's feature can be found in the image. The student soon realized however when implementing this technique that the template matching detection relied heavily on finding an exact match of the template filter and was unsuccessful at detecting the object if its size changed leading to the idea being scrapped.

***Greyscale Segmentation*:** For this technique the student first used Python 3 and the OpenCV library to collapse the cameras RGB input into a greyscale image input and then segmented the image based on a range of (0-255) values in the greyscale spectrum. The number of pixels in the image that fit within the defined range were used to define the objects area and from that, calculate looming. When using an object with high contrast to the rest of the environment, this method worked very well at accurately tracking the objects area; however, when tracking objects with less color contrast relative to the background, errors from false positives led to segmentation inaccuracies. For this reason, the student moved on to the next method.

***Color HSV Segmentation*:** Next, the student used OpenCV and Python 3 to perform segmentation based on pixel's values in the HSV color space. The HSV color space is a method of defining a color based on hue, saturation, and lightness value, hence the abbreviation. This approach segmented the image based on 3 ranges of (0-255) values for each channel in the HSV color space to define the objects area. This method segmented objects with less color contrast relative to the background much better than the previous method but there was still a considerable amount of noise affecting the area count.

***Refining the Segmentation*:** In attempts to reduce the area bias caused by false positives during segmentation, the student used functions in the OpenCV library to cluster the detected pixels into separate disconnected groups. Only the largest cluster of pixels was then used to define the objects area, assuming that the objects cluster would be larger than any noise. By doing this, the student not only increased the accuracy of the objects pixel area but also provided a method to determine the horizontal position of the object within the frame. Using simple geometry, the center of the object was calculated and used to define the position of the object within the frame.

b) Control System Designs

***Control using regions*:** Since the horizontal position of the object was known, the student started to think of ways they could turn the Jetbot to follow the object. The control method the student formulated involved splitting the frame up into 5 equal regions labeled as the left, mid-left, middle, mid-right, and right regions. From these regions, the algorithm determined what section the objects center point was in for each frame and sent corresponding commands to each of the motors scaled by the calculated looming value.

***Control using error*:** Rather than using regions, next the student implemented an analog control method based on the distance of the center point to the middle of the frame. By using this error of how far away the center point is from the center of the frame, a dynamic value for each of the wheels could be calculated. These values become more extreme when the error is large and less extreme when the error is small, making the Jetbot's movements more responsive.

Without realizing it, and with no prior knowledge of control theory, through the process of exploratory learning and divergent thinking, the student developed a working feedback control loop from the ground up. It should be emphasized that the student never took a control class or learned control theory concepts from other course work prior to this project and essentially "re-invented" this control system through the design freedoms presented in this research.

5. Converging on a Final Design:

Once the student had prototyped and tested many divergent designs, it was time to converge on to a single system-level approach. The first implementations that could be weeded out when picking a final design candidate were K-means clustering and template matching due to their lack of success in the prototyping phase. Through the natural evolution of ideas in the previous step of the design process, certain implementations became outdated and could also be eliminated as possible final design choices such as greyscale segmentation and control through regions. In the end after ideas were discussed in the weekly meetings between the undergraduate student and the mentors, the final design the student converged on was a system that collected input using HSV color segmentation and controlled the movement of the Jetbot using a feedback control loop. We will now give a full overview of the final system design.

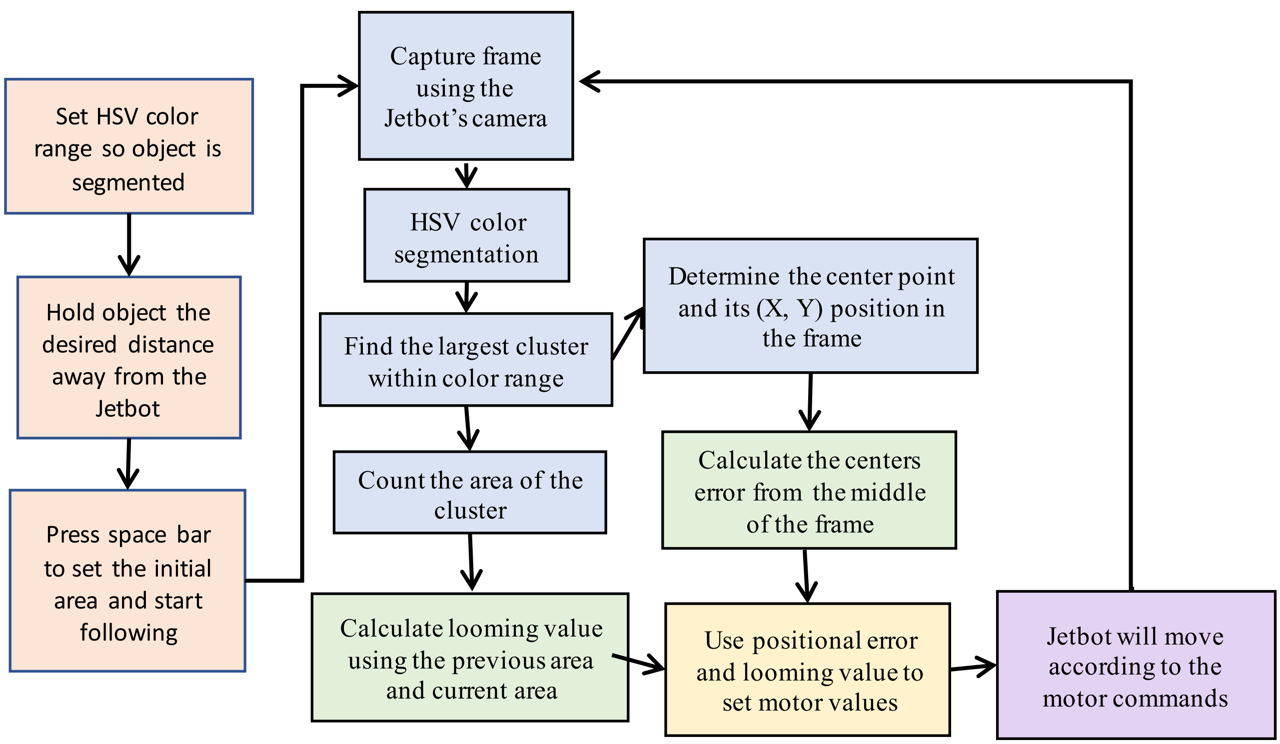

Figure 2: Main system diagram

Figure 2 diagrams the main flow of logic and operations in the final system starting with the initial calibration steps of setting the HSV color range using sliders and positioning the object the desired distance away from the Jetbot. Once the color range is set and the object is the desired "follow distance" away from the Jetbot, the user can press the spacebar on a Bluetooth connected keyboard to begin the main system loop. This loop begins with the camera capturing a frame and applying the HSV color segmentation algorithm, described in step 4 of the design process "Divergent Thinking and Prototyping". This will produce a mask of all the pixels that fit within the given range illustrated by the white clusters in the right window of Figure 3. From these white clusters the largest one is then chosen and assumed to be the object of interest. Once the object of interest's cluster is determined the program uses basic geometry to calculate the regions centroid and draw both the center point and an outline around the object for visualization purposes on the image, as shown in the left window of Figure 3. Counting the pixels within the selected cluster gives the perceived area of the object on the cameras lens and can be saved as the initial starting area.

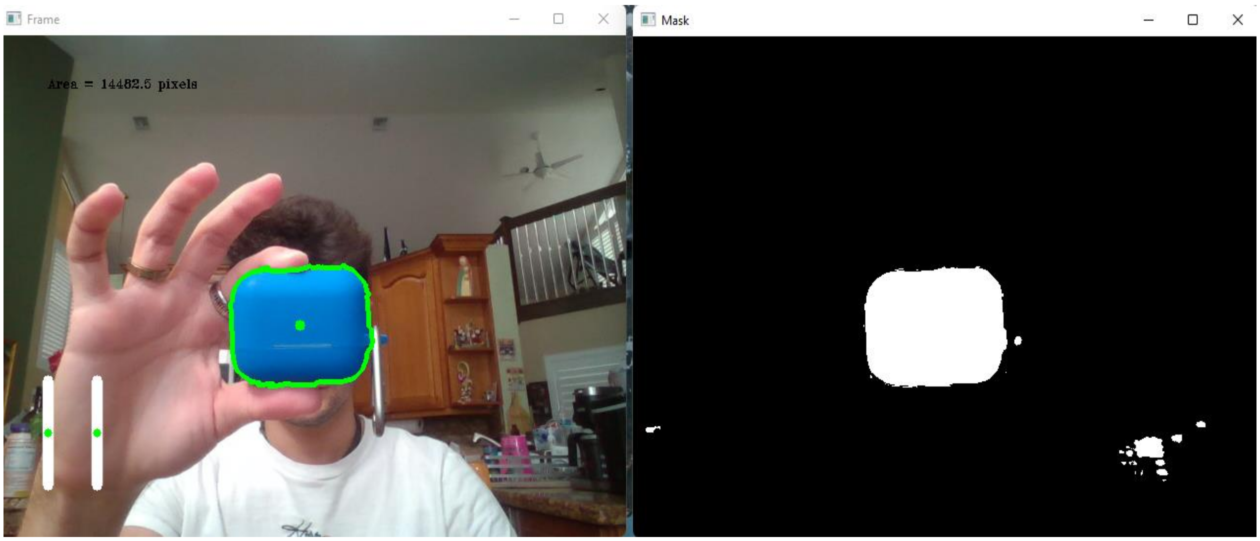

Figure 3: Frame with object outlined in green in the left window, and HSV color segmentation mask in the right window

For each frame a looming value is produced using the current and previous computed area of the object. Once the looming value is calculated, the position of the object's centroid will be used to determine an error value for how far the object is from the middle of the screen. Both the looming value and error measure are then used to send commands to the left and right motors. These commands are sent with the objective of keeping both the center point as close as possible to the middle of the frame and the current area as close as possible to the previous recorded area. Once the Jetbot moves according to the motor commands, a new frame is captured by the camera and the loop repeats. This closed loop system constantly updates the motors with the most accurate commands to keep the object at the desired distance away from the Jetbot resulting in a dynamic, real-time, following behavior.

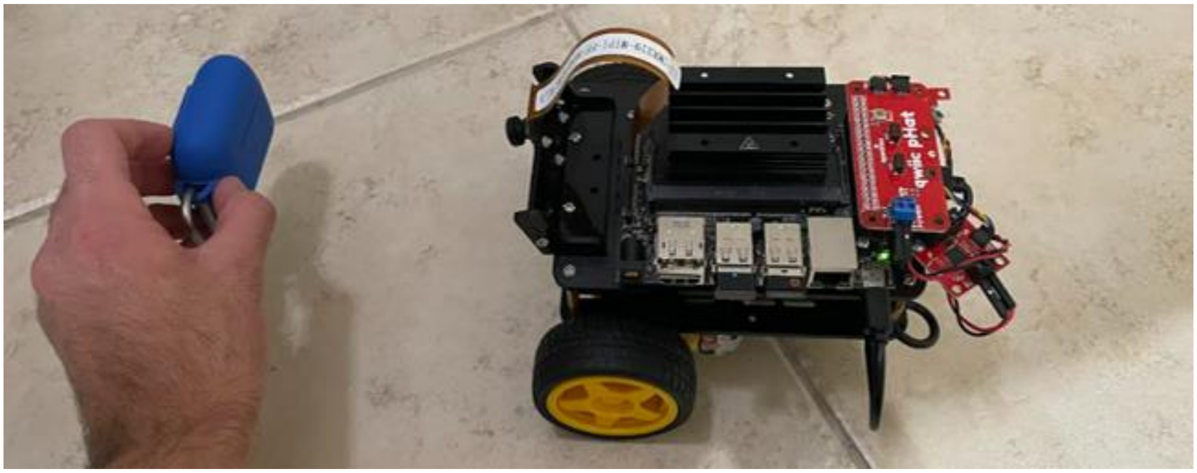

Figure 4: Jetbot keeping a 10 centimeter distance away from the object during testing and evaluation

6. Testing and Evaluation:

With a final design selected and built, it was time to test the system and evaluate it based on the project requirements. To test the functionality of the system, the student used a blue AirPods case as the object and had the Jetbot follow it, keeping about a 10-centimeter distance away as demonstrated in Figure 4. During the many tests of the Jetbot following this object, we found that it was very responsive with its forward and backward motions as well as its turns, rarely loosing track of the object or making unexpected movements. We further tested this system under different lighting conditions and found that it worked best when the object was well lit, allowing for the best color detection. This testing in different lighting conditions led us to brainstorm on how we could preserve the detection accuracy of the object, regardless of environmental conditions like light or shadows. The obvious solution to this problem that we came up with was to attach a small light at the head of the Jetbot near the camera so that the object would always be illuminated in a common way. The light we purchased however was too strong and lit only a small region of the cameras field of view leading to no detection benefits and in some cases causing more detection problems. In attempts to fix this issue, we used multiple layers of saran wrap to try and diffuse the light, so it was not as intense and had a wider spread. This temporarily solved the problem but only showed strong benefits when running the system in a poorly lit space and could probably be improved with a different kind of light in the future. We plan to quantitatively measure, present, and analyze these exact testing results in future work.

As a final test of the system and to push the limits of our design, we acquired another Jetbot and cloned it to be an exact replica of the original one. Running the exact same software on both devices, we had one Jetbot, with a marker attached to the back of it, follow an object in our hand and had the other Jetbot follow the marker on the rear of the first one as visualized in Figure 5. This showed the communication of movement between 2 devices running the same software using nothing but computer vision and the visual looming algorithm and resulted in a visually impressive demo which can be found in our GitHub, linked at the end of the paper.

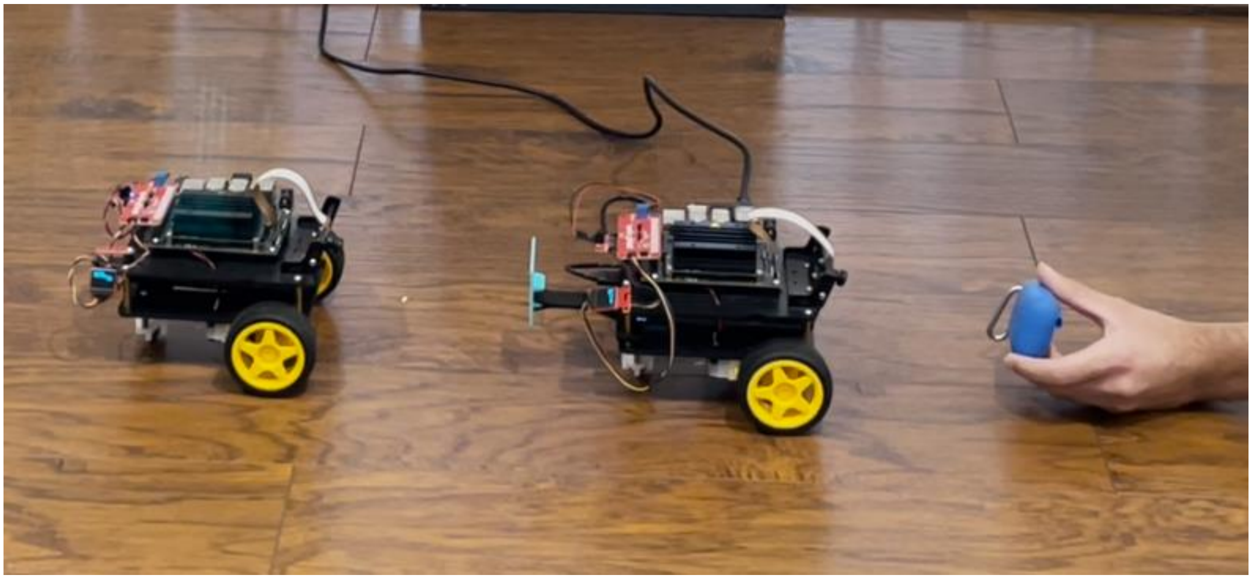

Figure 5: A Jetbot following another Jetbot which is following an object, both running the same software.

Conclusion and Future work

In this paper we outlined the learning journey of an undergraduate student studying the basics and practical applications of the visual looming computer vision algorithm in a system-level design, highlighting the challenges faced along the way and the ways those challenges were overcome. We showed the benefits of exploratory learning and how many divergent ideas, through the 6-step design process we outlined, converge into a single system-level final design. Through this design process we also shared firsthand accounts of the student's learning process, exploring how the student came up with different design ideas, faced challenges, learned from their shortcomings, and eventually emerged with a working system. If the reader were to ask, "what were we supposed to take away from this paper?", the answer would be, to encourage instructors to bridge the gap between the academic classroom and real-world vision based problem solving through the addition of open-ended, system design projects in coursework.

Since this research is still being conducted, exact performance measurements and figures along with corresponding data analyses are not present in this paper, however, as this research concludes, we look forward to documenting the exact experimental measurements of the Jetbot's performance under different conditions in future work to give a more complete design evaluation. Furthermore, in the near future we are excited to continue using the design process outlined in this paper to develop a system that uses neural networks to approximate visual looming and push the boundaries of what is currently possible in the computer vision space.

We hope that through this design experience, we can inspire students to learn with a growth mindset and attempt to first solve problems with their own intuition without the bias of past literature, allowing for out-of-the-box thinking and unbridled creativity.

Code and demos can be found at our GitHub: github.com/vmacri7/looming-research

References

1. Raviv, D., & Radzins, A., From Idea to Prototyping a Hands-on, Engaging Undergraduate Design Experience Paper presented the 2014 ASEE Annual Conference & Exposition, Indianapolis, Indiana, 2014
2. Sawatzki, C. G., & Muraleedharan, R. (2021, July), *Work in Progress: Using Cost-effective Educational Robotics Kits in Engineering Education* Paper presented at 2021 ASEE Virtual Annual Conference Content Access, Virtual Conference.
3. Kunal, J. *The visual looming navigation cue: A unified approach,* Florida Atlantic University, Boca Raton, Fla. 1995
4. D. Raviv, K. Joarder, *Visual looming navigation cue: A unified approach*, Computer Vision and Image Understanding, 79 (2000), pp. 331-363, 10.1006/CVIU.2000.0862
5. Arvid R. Eide, Roland D. Jenison, Lane H. Mashaw, Larry L. Northup, *Engineering Fundamentals and Problem Solving*, Hardcover, 416 Pages, Published 1997 by Mcgraw-Hill College
6. Guzey, S. S., Ring-Whalen, E. A., Harwell, M., & Peralta, Y. (2017). Life STEM: *A case study of life science learning through engineering design.* International Journal of Science and Mathematics Education, 17(1), 23-42.

Vincenzo A. Macri

Vincenzo Macri is an undergraduate student at Florida Atlantic University (FAU) pursuing a double major in Computer Science and Computer Engineering. He is also dual enrolled in FAU's Master of Science program with a Major in Artificial Intelligence. Vincenzo began his studies at FAU in the summer of 2017 in FAU's unique high school program which allowed him to begin taking course work for his bachelor's degrees in ninth grade. As a dual enrolled high school student, he joined FAU's Machine Perception and Cognitive Robotics Lab where he studied the latest developments in deep learning research. In the first year following his high school graduation, Vincenzo was recognized with the 2022 University Scholar Award for the College of Engineering & Computer Science, an award given to one student from each of FAU's 8 undergraduate Colleges in recognition of outstanding academic achievement. vmacri.com

Daniel Raviv

Daniel Raviv received his B.Sc. and M.Sc. degrees from the Technion, and his Ph.D. from Case Western Reserve University in Cleveland, Ohio. He is a professor at Florida Atlantic University (FAU) where he is the Director of the Innovation and Entrepreneurship Lab. In the past he served as the assistant provost for innovation. Dr. Raviv taught at Johns Hopkins University, the Technion, and the University of Maryland, and was a visiting researcher at the National Institute of Standards and Technology (NIST) as part of a group that developed a vision-based driverless vehicle for the US Army (HUMVEE; 65 mph). His related research work includes exploration of visual invariants that exist only during motion and can be used for real-time closed-loop control systems of cars and drones. He is also interested in teaching and learning innovative thinking, and how to teach innovatively. He is the author of five books: three on learning innovative thinking and two on teaching in visual, intuitive, and engaging ways.

Juan D. Yepes

With 18 years of experience in the telecommunications industry, Juan Yepes is a seasoned professional with a diverse background in technology management and leadership. He served as a manager in the telecom industry until 2017, and then went on to become the CEO of a technology company specializing in industrial equipment. In 2020, he began a new chapter in his career by starting his PhD program at Florida Atlantic University. His expertise spans a wide range of networking platforms, including IP, MetroE, ATM, wireless, and fiber and he has a deep understanding of the regional operations and deployment of complex networks throughout Latin America. In addition to his professional experience, he is also a researcher and developer in cutting-edge fields such as AI, computer vision, autonomous flying vehicles (drones), the Internet of Things, and digital fabrication.