Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #47510

From Mathematical Theory to Engineering Application: An Undergraduate Student's Research Journey

Tony Malayil, Florida Atlantic University
Dr. Daniel Raviv, Florida Atlantic University

Dr. Raviv is a Professor of Computer & Electrical Engineering and Computer Science at Florida Atlantic University. In December 2009 he was named Assistant Provost for Innovation and Entrepreneurship.

With more than 30 years of combined experience in th

Juan David Yepes, Florida Atlantic University

Assistant Professor in Teaching

From Mathematical Theory to Engineering Application: An Undergraduate Student's Research Journey

Abstract

With the rapid development of autonomous vehicles and advanced sensing technologies, the demand for expertise in computer vision has surged. However, many undergraduate students have limited or no exposure to this growing field. This paper documents an undergraduate student's journey in learning and implementing a Time-to-Contact (TTC) algorithm—a critical tool that estimates the time until a moving observer collides with an object— thus expanding the range of vehicles that can be driven autonomously. By sharing this experience, the paper provides a roadmap for other instructors to guide their students in acquiring essential knowledge and practical experience in computer vision.

The student began with minimal knowledge of computer vision and no prior experience with optical flow algorithms. By studying scholarly papers on TTC and visual looming and engaging with visual demonstrations, they developed a foundational understanding of the concepts. The next step was to create a vision-based 3D Python simulation to calculate TTC for a moving object relative to the camera, which yielded positive and verifiable results. Weekly team meetings, where professors and students reviewed progress, fostered a collaborative learning environment. Peer teaching methods were employed, with the student explaining learned concepts and outlining future steps. This ensured clarity, reinforced understanding, and strengthened the student's communication skills—an essential aspect of research success. Discussions with professors encouraged divergent thinking, enabling the student to explore multiple approaches. This process ultimately led to the development of a threat detection algorithm using YOLO3D to estimate TTC in real-world vehicles.

Instructors can replicate this learning experience by incorporating hands-on activities, such as modifying the object position matrix in the Python simulation. This allows students to observe changes in TTC as an object moves relative to the camera. With minimal guidance and clear code documentation, students can quickly grasp the core principles of optical flow and computer vision.

This scalable learning approach provides students with a strong foundation in applying mathematical concepts to real-world scenarios. As they progress, students can take on more advanced challenges, such as modifying object properties, further deepening their understanding of computer vision algorithms like OpenCV. By combining hands-on experience with effective teaching strategies, this approach accelerates learning and prepares students for higher-level opportunities in computer vision research. By sharing both technical insights and teaching methodologies, this paper empowers instructors to introduce undergraduates to computer vision, paving the way for impactful contributions to autonomous technologies.

1. Introduction

Along with the rise of natural language processing and large language models, computer vision has emerged as a rapidly growing field within artificial intelligence. Computer vision involves the use of algorithms to analyze visual stimuli, mimicking our ability to perceive the environment around us through vision. This technology has driven advancements across multiple industries, including applications in the medical field, agricultural production, and autonomous vehicles [1]. Its broad range of applications has significantly increased demand, positioning the field for substantial projected growth. However, undergraduate students in college and university institutions nationwide lack the adequate experience and skills needed to fill the labor demand.

Upon entering university, the student desired to explore and gain proficiency in computer vision and related technologies. Through undergraduate research programs such as OURI, which will be discussed later, the student began to be actively involved in undergraduate research, working on visual looming and TTC. This research experience provided the student with foundational and advanced skills in computer vision that will extend beyond their academic career, fostering both technical expertise and professional growth. Key competencies developed during this project include:

- ***System-level understanding*** – Comprehending how individual components, from visual concepts to complex code structures, interconnect to achieve the overarching project goals.
- ***Innovative mindset*** – Constantly seeking creative solutions and identifying ways to improve the efficiency of existing methods.
- ***Divergent and Convergent thinking*** - Generating multiple potential solutions to a problem and converging them to an optimal solution.
- ***Soft skills*** – Developing various essential interpersonal skills, such as problem-solving, time management, and team collaboration.
- ***Technical skills*** – Gaining hands-on experience with tools and technologies such as Python, Matplotlib, OpenCV, source control through GitHub, and frameworks like YOLO (You Only Look Once).

Over the course of the project, a variety of instructional strategies were employed to guide the student. Along with giving them these skills, these strategies allowed them to overcome various obstacles that came up during research. The following sections will provide an in-depth analysis of these techniques as well as how they positively impact the student’s progress. By sharing this developmental process, we hope to show educators practical methods that can be used to mentor students in acquiring these vital skills, gain real experience, and stay engaged in meaningful research, preparing them to meet the needs of this growing field.

2. Barriers to Research

Prior to entering the research, the student faced various challenges that discouraged him from pursuing research opportunities. Many students enter university with the goal of getting practical experience and understanding in their field. However, barriers such as lack of knowledge and skills, motivation, and structural blocks can hinder them from gaining experience through academic research [2]. This section highlights challenges faced by the student and the strategies that helped overcome them, offering possible solutions to broader undergraduate populations.

- **"Fear" of research**: The student doubted their ability to conduct research or to work on a project alongside more experienced students and faculty. They had the belief that most opportunities would be limited to highly experienced students.
- **Time constraints**: Students fear that their class schedules and other commitments will prevent them from dedicating time to research. The struggle to balance these commitments could lead to a decline in quality of life and poor work output.
- **Lack of interest**: The student sought to work on meaningful, impactful projects rather than grunt work. They felt uninspired by research opportunities that lacked clear significance or tangible results, leading to hesitancy surrounding their role and involvement.

Despite these considerable barriers, the student was able to overcome them with support from the university and professors:

- **Undergraduate research program**: The student joined a program run by the Office of Undergraduate Research through which students are introduced and guided on research practices. The program has two components: a class in which students learn about academic writing and out-of-classroom research with a professor. As a result, the program significantly prepared the student prior to joining a research group, helping eliminate his "fear" of research. Universities that have and encourage students to join such programs can strengthen the abilities of their students [3].
- **Active recruitment by peers**: The student was recruited into the lab by one of its undergraduate researchers. The researcher recognized their interest in computer vision and believed they would be a strong candidate to join.
- **Professor mentorship**: Once accepted into the lab, the professor explained computer vision using one of the field's strengths: visually demonstrating the foundational concepts. Along with this explanation, the professor gave the student the overall goal of the lab and how the student would fit in it.

By addressing these obstacles through structured programs, peer recruitment, and mentorship, the student overcame the challenges that initially discouraged him from pursuing academic research. The common themes of these solutions are reaching out to students and easing their concerns about entering academic research. Initially, students may require the extra push to get out of their comfort zone to engage and have their questions about the process answered. However, after this first stage, students would be able to take initiative on their own with proper guidance, as seen in the student during the later stages of this project. Educators looking to grow undergraduate research in their labs or universities should propose programs engaging students in research, preparing them and creating a pool of students who are capable. Actively searching for and recruiting qualified candidates is vital, as professors and fellow researchers would best be able to identify interested students, whether it be from classes or other extracurricular activities. By encouraging undergraduate research, faculty not only strengthen the abilities and skills of the student population but also their academic research and professional network, and foster innovation grounded in the scientific method.

3. Approach

The student began the project in his second semester in university with no prior knowledge of computer vision and some experience with Python and fundamental computer science concepts. As mentioned earlier, the student received instruction from the professor, beginning to build foundational knowledge. In addition to the professor's visual demonstration of the computer vision concepts, the student also received handwritten notes and drawings of them to recall in the future. To deepen their theoretical understanding, the student was then provided research papers such as David Lee's *General Tau Theory* [4].

The educational approach implemented to guide the student can be broken up into three core components: educational dynamics and techniques, design process, and skills acquired.

3A. Educational Dynamics and Techniques

The student benefited from intentional mentoring and collaborative dynamics with peers to bridge the gaps in the semi-directed self learning process.

- **Professor-Student Dynamics**: On top of the weekly update meeting, the student was constantly communicating with the mentoring professor. From questions and updates in progress to scheduling issues, the student was encouraged to share what was going on openly. This not only increased the student's confidence in their mostly self-guided project but gave the professor a better understanding of their trajectory. Along with responding to the student in a timely manner, the professor regularly sent real-world examples demonstrating the visual concepts using videos and images.

- **Peer Dynamics**: Communication with peers within the lab was vital to the student's growth. As lab members would work on similar but separate projects, the student was able to get different perspectives on approaching a problem and grow in understanding of visual looming. This also allowed for a supportive environment where peers would assist each other rather than compete.

Along with these dynamics, a form of the Feynman Technique and a focus on foundational concepts ensured the student stayed on course. The Feynman technique, created by the Nobel Laureate physicist Richard Feynman, involves four steps: learn about the content, teach it as if to a child, fill the gaps in understanding or explanation, and further simplify [5]. This technique was loosely adapted to evaluate the student's understanding of instructions and concepts. The new steps were the following:

a. Study research material and instructions related to the upcoming week's tasks.
b. Reteach material and instructions back to the group in your own words, signaling comprehension.
c. Pose questions and incorporate feedback to fill gaps in student's understanding.

d. Develop and solidify a plan for the upcoming week based on refined understanding.

The student was also encouraged to focus on the foundational concepts from the initial papers and constantly realign his plan with the goal of the project, developing a system to detect threats using TTC. This prevented the student from getting caught up in the depth and complexity of the field and to focus on the goal. These strategies, in tandem, ensured the student stayed on track for the duration of the project. The combination of collaborative dynamics, the adapted Feynman technique, and a focus on foundational concepts ensured that the student made steady progress while developing a robust understanding of computer vision. These strategies promoted self-confidence, enabling the student to navigate challenges independently while benefiting from mentorship and peer support.

3B. Design Process

While implementing techniques to encourage learning, students must also adopt methods that will encourage progress in reaching results. The process of taking a concept to application is the most laborious of most projects; thus, planning the steps that must be taken so that the student sees consistent progress and satisfaction from it. In this project, the student used a design process known as the "*creativity and innovation process*" discussed by [6] and adapted by [7]. This process consists of six steps:

1. **Understanding the problem**: Analyzing the problem and evaluating current solutions.
2. **Creative Brainstorming**: Suggesting a creative approach to the problem without any prior research and acquiring a better understanding through testing.
3. **Reevaluate and Redefine**: Adjusting the student's understanding of the problem after gaining a better understanding. This may involve splitting the problem into smaller problems or changing the original question to be more specific.
4. **Ideation / Divergent Thinking**: Generating multiple divergent solutions to the problem.
5. **Convergent Thinking**: Synthesizing multiple ideas into a cohesive and efficient solution through refinement.
6. **Evaluation**: Testing the final solution in a controlled real-world setting, evaluating its accuracy and speed.

3C. Skills

Through the process of exposing them to high-level topics to converge on a real-world application, students are empowered to grow at their own pace and apply the skills they acquire to a tangible product. Through this process, students gain a diverse set of skills, including but not limited to:

Soft Skills

- *Problem-Solving Skills*: Developing critical thinking through exploration, divergent and convergent thinking, and iterative trial-and-error approaches.
- *Understanding the Bigger Picture*: Understanding high-level concepts and effectively applying knowledge to real-world scenarios.

- *Time management*: Effectively managing time to meet project milestones and balance schoolwork.
- *Team collaboration*: Leveraging interdisciplinary team diversity, with members from various levels of experience (undergraduate to PhD) and engineering disciplines, to spur innovation through shared knowledge.
- *Communication*: Articulating ideas clearly, sharing progress, and receiving constructive criticism.
- *Proactive mindset*: Working with a “can-do” attitude when faced with challenges.

Technical Skills

- *Python Programming*: Proficiency in Python and relevant libraries, including Matplotlib for visualization and OpenCV for computer vision.
- *GitHub*: The student became familiar with using GitHub for source control and to share his work with others.
- *YOLO framework*: Experience with YOLO (You Only Look Once) and its advanced versions for object detection.

4. Technical Approach

A. The Problem

The main goal of this research is to develop a system that uses time-to-contact visual invariant to determine potential obstacles to vehicles using a single camera. Prior to implementing this system, a simulation was developed to visualize and test these concepts. The simulation, created using Python and Matplotlib, analyzed TTC and the angles created by obstacles, ensuring a thorough understanding of the underlying principles before implementing them in a real-world application. For context, one must first understand optical flow and visual looming.

As in many fields, inspiration for this concept came from humans and animals. Humans and animals can visually detect threats with minimal other sensory support. This allows us to respond to these threats or obstacles, predicting their movement and countering them. Seeing how this is an instinctive process, our brains can process frames and determine the relative velocity of objects. Just as animals and humans unknowingly utilize the mathematical relationships present in the world, these relationships can be used to determine threats and avoid them.

This pattern of motion within an image or frame is known as optical flow. In *General Tau Theory*, optical flow and other information can be categorized as tau, which is the information required for movement control [4]. *General Tau Theory* determines that visual factors can solely be used to create optical flow, which can predict an object’s velocity with respect to the viewer.

The visual concept used in this paper is time-to-contact (TTC), which measures how long it takes for an object to reach the same plane as the viewer. TTC is calculated from the angle made by the object to the vanishing point and the rate of change of this angle over time.

Expanding on this, TTC could be used to determine which objects are moving closer to the vehicle's path, leading to an impending collision. As found in [8], the equation for TTC is:

$$1 / \text{TTC} = 2\dot{\theta} / \sin 2\theta$$

As Time-To-Contact can be used to find the collision time of an object in a certain plane, it can be used to determine the time to impact across multiple planes. By evaluating the TTC on the X-plane (horizontal) and Z-plane (depth), one can estimate the possible collision time by determining when and if the object will intersect the vehicle's path. The images below demonstrate this effect using a corridor. As the viewer moves laterally from Figure 1A to Figure 1B, theta (θ), the angle made by the top and bottom of the pillars and highlighted by the red lines, increases. This angle progressively approaches 180° as the viewer gets closer to the pillars. Using theta, we can use this effect to calculate the time to reach contact with this plane, assessing spatial relationships in real-time and finding TTC in the Z-plane. Similarly, we can use alpha (α) —the angle created by the lines extending from the vanishing point to the viewer and from the viewer to the bottom left corner of the vehicle— to calculate TTC in the X-plane.

Figure 1A and 1B: The image on the left (Fig. 1A) shows the corridor while the viewer is at T_1, in the center. The image on the right (Fig. 1B) shows the corridor while the viewer is at T_2, when the viewer is closer to the side.

B. Detailed Design Process

The following sections will outline how the "*creativity and innovation process*" facilitated the student's transition from understanding mathematical theory to applying it to real life.

1. Understanding the problem

As outlined in the technical objectives of this project, the student was assigned the task of creating a system that could determine the Time-To-Contact of approaching objects using a single camera. Collision detection and objection avoidance systems are extensively employed in modern vehicles, whether they support autonomous driving or not. However, such systems are often reliant on a multitude of sensors, rendering them inaccessible for smaller vehicles or older car models. In contrast, Time-To-Contact can be determined using a single camera, making it more accessible and applicable.

To achieve this, the program would have to be able to identify vehicles and the angles they make relative to the vanishing point. It must track these angles across multiple frames to find its time derivative, adding to the complexity of object-tracking algorithms. Another key limitation of the overarching project would be that it would not support rotation by the camera. Rotation of the camera would change the location of the vanishing point in the frame, resulting in inaccurate TTC calculations.

This project proved to be a significant challenge for the student and pushed them to work outside of their comfort zone, i.e. a more traditional classroom structure. Without strict deadlines, the student could easily fall into a pattern of procrastination, lowering the quality of work. To circumvent the problem early on, the student allotted time throughout the work to focus on the project, adding more when necessary. To grow their understanding, the student was also encouraged to digest the material themselves and, if needed, to reach out to other lab members such as professors and partners.

2. Creative Brainstorming

With an understanding of the problem at hand, the student chose to start by exploring a possible solution with the meager knowledge of computer vision they had. The first possibility that came to mind was to use OpenCV to find the lines created to the vanishing point by the roof and floor of the car. The student would do this by using the Canny transformation on the frame first, which would detect the edges within it and remove noise, as shown in Fig. 2B. After doing so, the lines created by the edges would be found using the Probabilistic Hough Transform. To find the angle of the lines to the vanishing point, the student created a module in Python containing two functions that would apply the theory of the project. The first function finds the angle created by two vectors, which in this situation would be the roof and floor of the car, using the dot product of the vectors. The second function calculates the Time-To-Contact of an object when given the previous angle, current angle, and time interval between, using the TTC equation found in *Technical Goals*.

In theory, this would allow the student to find the TTC of vehicles. However, it had many major errors in planning and implementation, which the student failed to address. While applying Canny and Hough Transforms would find the lines within the frame, the student would not be able to attribute which lines belonged to the car as seen in Fig. 2C. On top of that, the abundance of noise was still an issue even after applying the Canny Transform, making this method unusable. Seeing that the initial attempt at an intuitive approach had failed, the professors advised the student to explore other methodologies and to realign his focus.

Figure 2: Image of the train created in Unity. Fig. 2A (left) shows the train prior to applying transforms. Fig. 2B (middle) shows the train in Fig. 2A after applying Canny Transform. Fig. 2C (right) shows the train after applying both Canny Transform and Hough Transform.

3. Research and Redefine

After the student's initial failure, they were able to divide the project into two problems: identifying cars within the frame and extracting the lines created by the roof and floor of each car. The biggest concern of these two would be identifying vehicles in the frame, as this suggests using machine learning through algorithms such as Convolutional Neural Networks (CNNs). Only once the vehicle was identified could the angle it forms be found and used to find TTC. Now that the student had redefined the problem they faced, they also chose to simplify it as well, deciding to focus on finding the TTC for a single vehicle in an environment free of other cars. Once this solution was achieved, they would scale it up for more complex, busier environments.

4. Ideation / Divergent Thinking

Now that the student had a clearer view of the problem, they began to offer possible pathways to creating a system capable of collision detection using TTC. They were the following:

a. Python 3D Motion Simulation: To test the mathematical theory of TTC, the student proposed to simulate a 3D environment in Python. In this environment, the student would be able to create obstacles that could approach or move away from the viewer at varying speeds while calculating and displaying the TTC.
b. Unity Simulation: Unity is a 3D engine that would allow the student to create a complex environment in which the mathematical theory could be further tested. Like the Python simulation, this simulation would have obstacles approach or move away from the viewer, calculate the TTC of the object, and compare it to the real-time of collision.
c. OpenCV: This path chooses to improve on the initial prototyping done by the student. After using a model to identify the vehicle, the student would use OpenCV to extract the vehicle's angles to use for calculation.
d. YOLO3D: YOLO is a computer vision framework used for object identification, specifically for vehicles and pedestrians. By using a specific version of it known as YOLO3D where 3D bounding boxes surround the car, the student would be able to extract the angles needed from the bounding boxes.

After evaluation, the student and professor decided to develop the Python 3D motion simulation to test the theory and implement YOLO3D for the real-world collision detection system. Although Unity would have allowed for creating complex environments for testing, its steep learning curve and extensive development time would make it inefficient for simply testing the theory. In contrast, the student already had experience in Python and thus would be able to start coding the simulation with little preparation time. YOLO3D was also favored over OpenCV as it would be able to solve both challenges that the student had identified during the reevaluation process: identifying vehicles and extracting the lines using the bounding boxes.

5. Converging on Final Product

The first step of the student was to develop the Python simulation. By being able to visually test the theory in a controlled environment, the student would be able to grow his own understanding and demonstrate his research to others. The student began by creating a short Python module containing two essential functions that could be used both in the simulation and in the YOLO3D system: function *findAngle* that would take in the endpoints of the two lines and return the angle they make using dot product, and function *TTC* which calculated the time-to-contact using the previous angle, current angle, and time between. With the theoretical function complete, the student began to build the simulation from scratch; however, they ran into an issue. Creating a 3D simulation requires knowledge of projection matrices, which would map the 2D points into 3D depending on the viewer's position. The student had very limited knowledge of the topic and was encouraged to research methods for resolving the problem. While researching projection matrices, the student found an open-source 3D cube simulation using projection matrices. Rather than building from scratch, the student adapted the cube simulation code to give the cube motion, allowing the student to control its velocity. Afterward, the Python module was imported and used to calculate the angle and TTC of the cube in live time.

Images of the simulation are shown below. The frames on the left side of Figures 3 and 4 show what the viewer would see as the object approaches them, while the frames on the right display the bird's eye view of the object when the viewer, signified by the blue triangle, is at the origin. As the vehicle moves closer as time passes, the vehicle does not collide with the viewer but passes to their right.

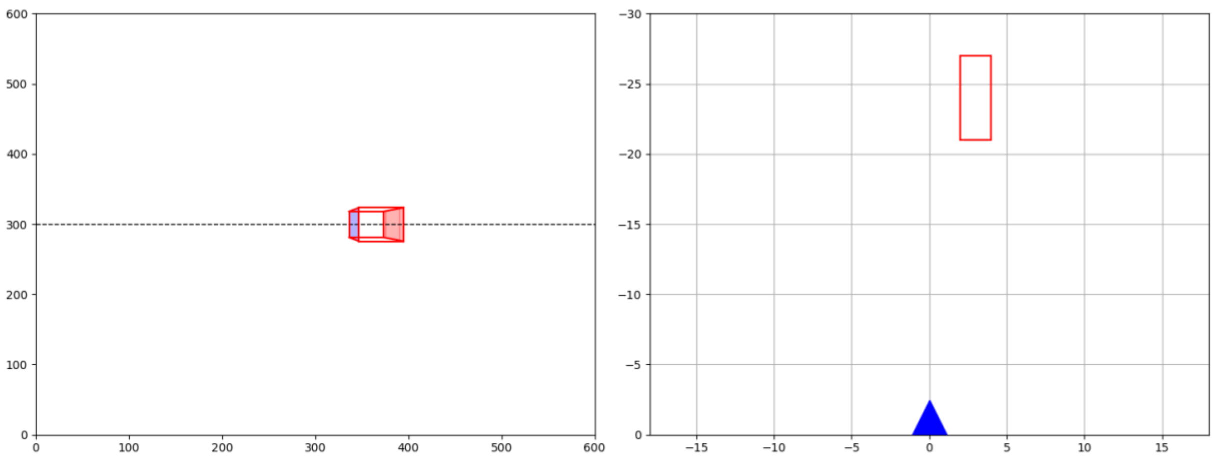

Figure 3: Python simulation at T_1 (frame #2)

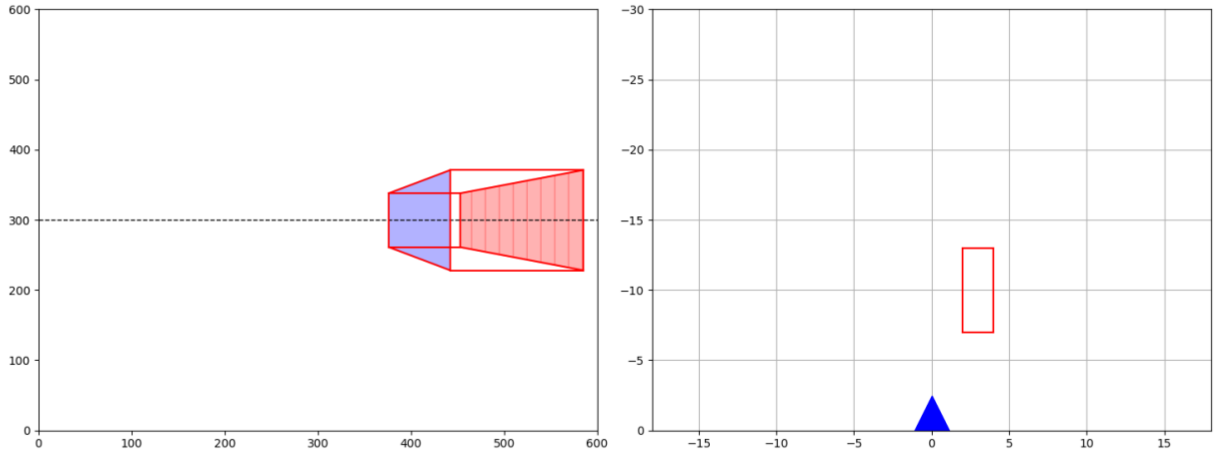

Figure 4: Python simulation at T_2 (frame #11)

Once the Python simulation was completed, the student transitioned to implementing the YOLO framework. YOLO is an object detection system known for its real-time detection. By utilizing this framework, the collision detection system could swiftly assess the threat before it gets too close.

As mentioned earlier, this system used YOLO3D, a specialized version of YOLO where the bounding boxes around the vehicles are 3D, allowing us to detect the angles it creates. The code environment was set up on Google Colab, a Jupyter Notebook service that would provide free access to GPUs, accelerating inference. However, the student encountered a significant challenge: YOLO3D, being a somewhat dated open-source software, had multiple failing dependencies and several errors. After several hours of troubleshooting and updating the code, the pre-trained model could run inference on data. Unexpectedly, the model struggled to detect vehicles, making it unusable. Believing they had reached a dead-end, the student began researching other methods for vehicle detection. They were able to find an improved version of YOLO3D, known as YOLO3D-Lightning, that was currently under development. After another round of troubleshooting, the student was able to use YOLO3D-Lightning to run inferences on footage. The student then integrated the Python module into YOLO3D-Lightning, allowing them to find the TTC of vehicles.

6. Testing and Evaluation

This project is currently evaluating the YOLO collision detection system in a real-world environment. As a result, the results and findings will be shared later. Figures 5 and 6 below show the preliminary testing outcomes.

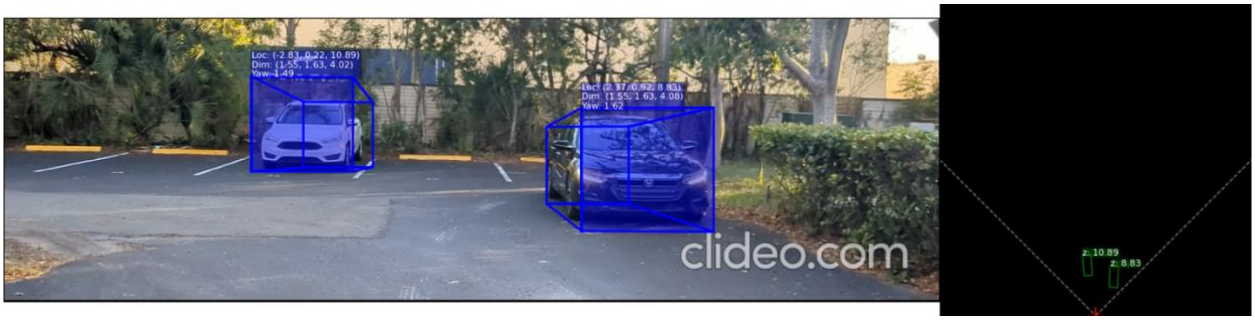

Figure 5: Frame #3 of preliminary YOLO3D results

Figure 6: Frame #4 of preliminary YOLO3D results

These preliminary results were taken in a controlled environment using a phone camera. In this footage, the vehicle, starting at rest, passes the viewer at a constant speed, similar to the simulation shown in Figures 3 and 4. Similarly, the results should yield a decreasing TTC in the Z-plane and no value for TTC in the X-plane. These results will be compared to the actual TTC values found by measuring the time it takes for the vehicle to pass the viewer. After initial testing, the evaluation will be expanded to a dataset of real road footage.

Scaling for the Classroom

Although originally designed to support an undergraduate student in a research setting, the strategies applied in this project demonstrate significant potential for broader application in the classroom. Many of the techniques empower the students to elevate their passion through self-directed learning with minimal supervision, fostering critical thinking and a proactive mindset. These strategies would be ideally suited for use in an Engineering Design class, where students design innovative solutions involving semester-long assignments. Upon selecting a problem -computer vision related or not- students should be given the "*creativity and innovation process*" as a roadmap to developing and refining their projects. This design process provides clear but flexible instructions that guide students through the various stages of brainstorming, prototyping, and testing. As students journey through this process, they should participate in biweekly meetings and have open paths of communication with mentors, allowing them to seek guidance when needed. This structure ensures that instructors prioritize students who need help rather than uniform guidance for the whole class. When teaching students who need help, instructors should apply the Feynman Technique to ensure students understand how to overcome obstacles. This approach mutually benefits both instructors and students, allowing instructors to dedicate more time to supporting students in need and guiding students to acquire skills and experience through a clear process.

While ideally suited for an engineering background, these strategies could be adapted for various project-based learning environments. As remote learning continues to grow in prevalence, the clear guidelines that the "*creativity and innovation process*" sets could be used so educators no longer need to constantly supervise students through creative projects, which is especially more difficult virtually. In smaller classes, educators could incorporate the Feynman Technique or adapt it to better fit their situation. For example, students would be taught a new concept and then split into small groups. In their groups, they attempt to reteach the concept and correct each other on any mistakes. To ensure comprehension, the instructor should quickly check for understanding before moving on to the next concept.

The best way for students to learn and build experience is through hands-on practice. To encourage other students to grow their skills in computer vision, we have provided the code for the Python 3D simulation. As part of this activity, students should modify the object position matrix to change the object's location. After changing its position, the student should view how it changed TTC in both dimensions. While this activity should not replace a self-directed project, it provides students the opportunity to familiarize themselves with computer vision and visual looming concepts, hopefully igniting a passion that they will pursue further.

The code and instructions can be found on our GitHub:
https://github.com/TonyMalayil/Looming-Python-Simulation

Conclusion

This paper documents the research journey of an undergraduate student as they acquire knowledge and experience in the field of computer vision. We discussed the initial struggles that they faced prior to joining research and how these obstacles were overcome through support from the university and professors. We showed the educational approach used, consisting of educational dynamics and techniques, the design process, and skills acquired. Through the design process, we shared the student's process of brainstorming, ideation, and the development of a collision detection system based on Time-to-Contact. We hope that through this progression, instructors receive inspiration to apply these techniques in their classroom and in their research.

As the system is still undergoing testing and development, test results from real data are not shown in this paper. In the future, we hope to share results from real-world testing. Future models would aim to be developed to allow for rotation of the camera and for live application. Once complete, the system will be able to identify threats and their TTC, allowing measures to avoid collision.

Acknowledgments

We extend our gratitude to Ayush Gowda for his mentorship and assistance throughout this project. We thank Vincenzo Macri for their essential contributions, being a vital part of the project. Lastly, we are extremely grateful to Michael Levine for his unwavering support of our work.

References

1. A. I. Khan and S. Al-Habsi, Machine Learning in Computer Vision, Procedia Computer Science, Volume 167, 2020, Pages 1444-1451, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.355.

2. Y. A. Adebisi, Undergraduate students' involvement in research: Values, benefits, barriers and recommendations, Annals of Medicine and Surgery, Volume 81, 2022, 104384, ISSN 2049-0801, https://doi.org/10.1016/j.amsu.2022.104384.

3. H. K. Beasley, Z. Vue, M. R. McReynolds, K. Neikirk, E. Garza-Lopez, M. Mungai, A. G. Marshall, B. Shao, J. Benjamin, C. N. Wanjalla, C. R. Williams, S. A. Murray, H. D. Shuler, A. Kirabo, A. Hinton, A review of undergraduate research programs aimed at underrepresented students, STAR Protocols, Volume 5, Issue 2, 2024, 102997, ISSN 2666-1667, https://doi.org/10.1016/j.xpro.2024.102997.

4. Lee, D. N., Bootsma, R. J., Land, M., Regan, D., & Gray, R. (2009). A theory of visual control of braking based on information about time-to-collision. Perception, 38(6), 837–858. https://doi.org/10.1068/pmklee

5. Reyes, Englevert & Blanco, Ron & Doroon, Defanee & Limana, Jay & Torcende, Ana. (2021). Feynman Technique as a Heutagogical Learning Strategy for Independent and Remote Learning. Recoletos Multidisciplinary Research Journal. 9. 1-13. 10.32871/rmrj2109.02.06.

6. Raviv, D. & Radzins, Anthony & Rahmani, Morad. (2017). From Observation to Prototyping: A Hands-On Undergraduate Research and Design Experience. 10.18687/LACCEI2017.1.1.403.

7. Macri, V. A., Raviv, D., & Yepes, J. D. (2023, March). *From equations to actions: A system level design research experience of an undergraduate student* Paper presented at ASEE Southeast Section Conference, Arlington, Virginia. https://doi.org/10.18260/1-2--45012

8. Raviv, D. (1991), Invariants in Visual Motion, NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=820367