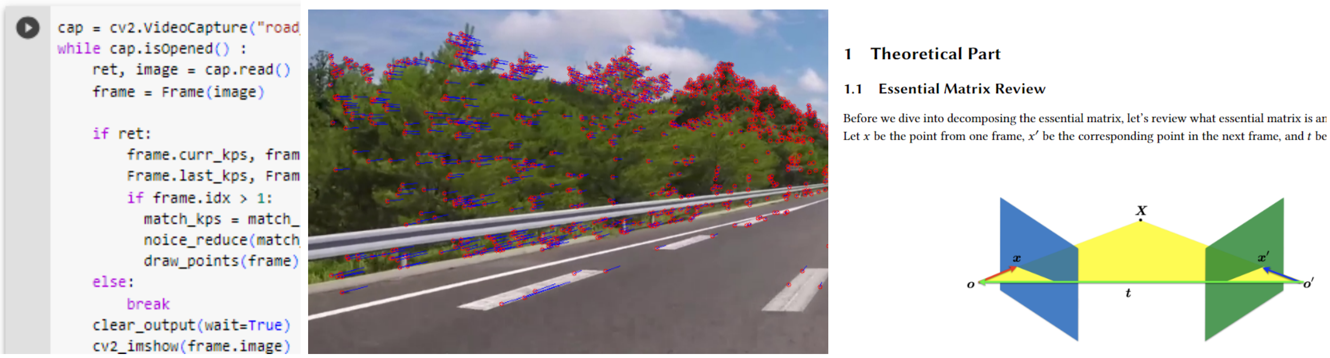
A proposal for homework 7 that further extends course content to learn SLAM systems

Yueqian Zhang
University of Washington
yueqianz@cs.washington.edu

Evan Zhao
University of Washington
hzhao5@cs.washington.edu



ABSTRACT

We aim to further extend the course content to study SLAM based inside-out tracking systems by mimicking the assignments in this class. We have designed theoretical questions related to basic introduction to SLAM. Then we built the starter code to put the theory into practice. We designed a basic 2D SLAM program that guides students through the pipeline of MonoSLAM (Real-Time Single Camera SLAM). The concepts and techniques we ask students to implement here are not only fundamental to SLAM but also applicable to other domains, such as multi-view geometry.

1.INTRODUCTION

With the rapid development in the Computer Vision field, software level solutions for the tracking problem started showing up. Many Modern VR headsets shifted from outside-in tracking to in-sideout tracking due to the breakthroughs in SLAM. Even controllers such as Meta Quest Pro's are using on-device cameras to track its pose now. Thus we feel the need to take a step to explore the technologies the knowledge behind it and even better, we decided to mimic the style of previous assignments and design a new one with the intent to possibly educate students with necessary information to have a basic understanding of what is SLAM, why use SLAM, and how does SLAM work.

Building a SLAM solver from scratch can be extremely challenging. However, someone has invented the wheels. Our work is based on the powerful OpenCV library and scikit-image[4]. We utilize the essential and well-optimized functions to complete many complicated tasks in computer vision. Yet we also deliberately chose to manually code up some of the algorithms, hoping that it could give students a both high level, and low level understanding of how SLAM works.

1.1 Contributions

- We designed a gentle yet informative introduction to how some of the key operations in SLAM work, along with the theoretical questions that help students fully digest the content just like previous assignments.
- We explored many approaches and greatly abstracted the code to run a simulated version of the MonoSLAM into a single Colab file that students can easily run on any platform.
- We put together a program that runs and provides visual feedback for students.
- We carefully picked and cut part of the code in our solution file in order to make it an actual assignment.

2. RELATED WORK

With the rapid development in the field of Computer Vision, Simultaneous Localization and Mapping (SLAM) has emerged as a powerful technique that combines data from multiple sensors, including cameras and inertial measurement units (IMUs), to estimate the position of a device while simultaneously mapping the surrounding environment. Different types of SLAM approaches, such as those discussed in [5], [6], and [7], have been developed, each with its own focus on achieving portability and high performance, which also inspired different kinds of SLAM courses like [1], [3].

In addition to the advancements in SLAM research, there is a growing discussion around making computing education more equitable and accessible. Several resources, as mentioned in [8] and [9], have been dedicated to building an inclusive computing education experience. These efforts align with our goal of constructing a project that introduces fundamental SLAM concepts in a manner that is accessible and usable for all students.

3. METHOD

By incorporating monocular SLAM and leveraging the progress made in computing education accessibility, we aim to create a learning resource that not only teaches SLAM principles but also contributes to fostering a more inclusive educational experience.

Among the various SLAM techniques, we specifically focus on monocular SLAM in this project. Monocular SLAM is chosen for its accessibility to a wider range of students, considering the complexity that can arise from working with robotics systems. As discussed in [5], this methodology enables localization using a single uncontrolled camera, relying solely on visual information.

We have drawn inspiration from the successful structure and philosophy of previous assignments in CSE 493v, incorporating both conceptual and programming components. In line with the approach followed in CSE 493v, where webGL and Chrome were employed to create effective visualizations and ensure accessibility for all students across different operating systems, we have opted for Colab as the platform for this project, capitalizing on its interactive nature and extensive collection of pre-installed Python libraries.

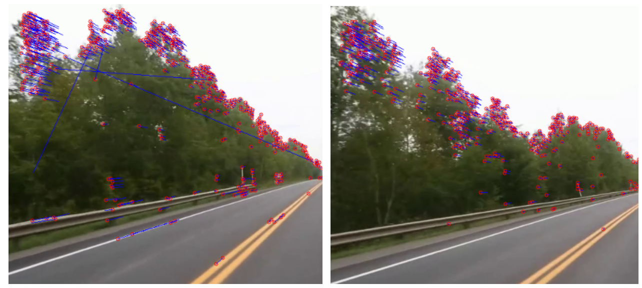
This choice not only facilitates an immersive learning experience but also eliminates setup and compatibility concerns. Students can seamlessly dive into the project, focusing on grasping the fundamental concepts and pipeline of SLAM without being hindered by technical obstacles.

4. IMPLEMENTATION DETAILS

The monocular SLAM pipeline is divided into two distinct parts for this project: the programming part and the conceptual part.

In the programming part, the first section of code involves extracting keypoints from the input images and subsequently matching these keypoints between frames. This step is crucial for establishing correspondences and tracking features across consecutive frames.

Following the keypoint extraction and matching, the second part of the code aims to reduce noise in the matching process. This is achieved by implementing a simple RANSAC (Random Sample Consensus) algorithm. RANSAC helps to eliminate outliers and improve the accuracy of the matched keypoints, thereby enhancing the overall performance of the SLAM system.



Matching before RANSAC Matching after RANSAC
(red dots are extracted keypoints and blue lines shows the movements of keypoints between frames)

Moving on to the conceptual questions part, we have carefully designed the structure following an actual MonoSLAM pipeline. We do not assume that students should have any prior knowledge on SLAM more than what was required in this class. Instead, we added detailed helper information along with each step in the derivation to walk students through step by step. Complicated mathematical calculations are omitted from the questions to ensure a reasonable difficulty level. Yet key derivations in the algorithm are presented and we believe students will benefit from them in many ways. For more information about the theoretical questions part, please refer to the PDF file in either the student folder or the solutions folder along with this file.

Similar to previous assignments in CSE 493v, the project is hosted on GitLab. This enables easy access and adoption of the homework. Interested individuals can access the project via the provided GitLab link, allowing them to explore and utilize the resources for their own learning purposes.

By splitting the SLAM pipeline into programming and conceptual parts and making the homework available on GitLab, we aim to provide a comprehensive learning experience of monocular SLAM.

5.DISCUSSION OF BENEFITS AND LIMITATIONS

Our work greatly leverages the power of some of the existing libraries, giving users the fastest way to code a working SLAM program. It also benefits from the online collaboration tool Google Colab. With its built-in environment, students are able to run the program anywhere and anytime. The integration of both conceptual understanding and programming components complements and reinforces each other, enhancing the overall learning experience, and providing students the whole picture of a simple SLAM in a greatly accessible way.

However, we are also aware of some of the limitations that our solution has. Firstly we explored the possibility of tracking using one camera, but most VR headsets use multiple cameras for better tracking. This may result in having a completely different style of SLAM. Second we didn't fully integrate the solver with existing headsets. The performance and accuracy of it remains to be tested on a real headset.

6.FUTURE WORK

Functionally, we aim to further extend the program to support reconstruction in 3D, namely the triangulation and extraction of camera's poses.

From an educational perspective, we would conduct a user study among students. The study will allow us to gather valuable feedback on the assignment's accessibility, usability, and effectiveness in teaching SLAM concepts. By engaging with students and addressing potential challenges or difficulties of the project, we can refine the assignment's content and structure to make it more engaging, intuitive, and inclusive.

7. CONCLUSION

Our project, inspired by the well-designed assignments in this class and the emerging new headsets such as Meta Quest Pro, aims to further extend the course content to cover the topics of SLAM based inside-out tracking systems. In the report, we first discussed the methodologies we used to create and design the theoretical questions and coding the MonoSLAM. Second, we dived deeper into the implementation details, along with its benefits and limitations. The modular, lightweight program and the corresponding written questions provide a good practice for students to have a quick preview of this field. Last but not least, we believe that there are a lot more we can improve due to the time limitations. Many interesting research questions within this topic are still left unanswered. We believe that more and more computer vision based software solutions will emerge and it will be very essential to get a touch on the basic education in this field.

REFERENCES

[1] Simulating slam from scratch using Python | introduction (2021) YouTube. Available at: https://www.youtube.com/watch?v=2GJuEIh4xGoamp;list=PL9RPomGb9IpRJLw5UTdSy4eJeoLrwNcfC (Accessed: 09 May 2023).

[2] (No date) 1 simultaneous localisation and mapping (SLAM): Part I the ... - people. Available at: https://people.eecs.berkeley.edu/pabbeel/cs287-fa09/readings/Durrant-Whyte$BaileySLAM-tutorial-I.pdf$ $(Accessed:10May$2023)

[3] Robotics: Perception (no date) Coursera. Available at: https://www.coursera.org/learn/roboticsperception (Accessed: 09 May 2023).

[4] Van der Walt, Stefan, et al. "scikit-image: image processing in Python." PeerJ 2 (2014): e453.

[5] Davison, Andrew J., et al. "MonoSLAM: Real-time single camera SLAM." IEEE transactions on pattern analysis and machine intelligence 29.6 (2007): 1052-1067.

[6] Thrun, Sebastian, and Michael Montemerlo. "The graph SLAM algorithm with applications to large-scale mapping of urban structures." The International Journal of Robotics Research 25.5-6 (2006): 403-429.

[7] Li, Ao, et al. "DP-SLAM: A visual SLAM with moving probability towards dynamic environments." Information Sciences 556 (2021): 128-142.

[8] Waller, Annalu, Vicki L. Hanson, and David Sloan. "Including accessibility within and beyond undergraduate computing courses." Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility. 2009.

[9] Lin, Kevin. "CS Education for the Socially-Just Worlds We Need: The Case for Justice-Centered Approaches to CS in Higher Education." Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1. 2022.