Development and Implementation of a Hands-On Robotic Course on Computer Vision for Autonomous Vehicles

Golnaz Habibi, Samuel Billerbeck, Daniel Vargas, Evan McCulley, Daniel Perez
University of Oklahoma
{golnaz, billerbeck, dvargas88, evan.l.mcculley-1, danielperez}@ou.edu

***Abstract*—This paper introduces the development of a robotic course, Computer Vision for Autonomous Vehicles. Derived from the VNAV course offered at MIT, this course is designed for senior and graduate students at the University of Oklahoma, where few courses focus on robotics. Our course offers hands-on experience with low-cost robotic cars, allowing students to implement fundamental computer vision techniques in a physical mini-city environment. We integrated the Kimera visual SLAM algorithm, providing extensive experience in visual odometry and mapping. The course material, including a Docker implementation of VNAV, is open-source and accessible to students and educators worldwide.**

***Index Terms*—Robotics, Software, Docker, SLAM, Kimera, Multi-view Geometry, Mini-city.**

I. Introduction

Robotics is an interdisciplinary field integrating engineering, computer science, and AI. Recognizing its significance, universities have begun incorporating robotics into curricula. At the University of Oklahoma, students face challenges accessing robotics education due to high equipment costs and technical complexity. To address this, we developed a new course inspired by MIT's VNAV, emphasizing practical computer vision applications.

The course provides a controlled mini-city environment and robotic platforms to experiment with SLAM algorithms. A Docker-based setup minimizes prerequisites, making it accessible to students across disciplines. The final project involves running Kimera VIO in real-time, enabling students to analyze SLAM performance under different conditions.

II. Course Infrastructure

A. Hardware

The course utilizes custom-built ARCPro robotic cars with Intel NUC i7 processors, RealSense D435i cameras, and a 1/10th scale mini-city for real-world testing. The hardware setup ensures consistency in experimental conditions.

B. Software

We provide a Docker-based ROS environment supporting ROS1 and ROS2, simplifying installation and compatibility. Students engage with feature extraction, object recognition, and pose estimation. The course also employs remote desktop utilities for controlling the robots.

III. Course Curriculum

The course consists of lab assignments and a final project focusing on visual odometry and SLAM techniques:

- Introduction to C++, Docker, GitHub
- ROS Basics
- Feature Matching and 3D Pose Estimation
- Loop Closure Detection in Mini-city
- Object Detection using YOLO
- Final Project: Real-time Kimera VIO on ARCPro Robots

IV. Course Assessment and Student Feedback

Two student surveys were conducted, and results showed that 80% of students improved their problem-solving and research skills. Additionally, 83% of students reported increased interest in computer vision, with 86.6% of graduate students finding the final project rewarding.

V. Conclusion

This course bridges the robotics education gap at the University of Oklahoma by offering hands-on experience with computer vision algorithms. The use of Docker enhances accessibility, while real-world experiments reinforce theoretical knowledge. Future iterations will refine the curriculum to further improve student engagement.

Acknowledgments

We thank Phong Nguyen, Alexander Cruz, Luyu Niu, and Christian Sink for their contributions. This work was supported by the Gallogly College of Engineering.

References