Visual Navigation for Autonomous Aerial Vehicles (VNA2V) - AEROSP 740
Syllabus

[picture courtesy of NASA, Urban Air Mobility Program]

Lecture, Lab, and Office Hours

- Online Lectures:† M/W/F 1-2pm EST (on Zoom): https://tinyurl.com/VNA2V21-lectures
- Online Labs:† TBD (on Zoom): https://tinyurl.com/VNA2V21-labs
    - Lab handouts: https://tinyurl.com/VNA2V2021-handouts
- Office Hours: TBD (on Zoom): https://tinyurl.com/VNA2V2021-office-hours

†Lecture and Lab Hours will also be made **available offline**, to assist **asynchronous learning**.

Websites

- *Canvas*: https://tinyurl.com/VNA2V2021-canvas (announcements, material)
- *Piazza*: https://tinyurl.com/VNA2V2021-piazza (Q&A with TAs and Instructors)
- Link to this syllabus: https://tinyurl.com/VNA2V2021-syllabus

Class Staff

- *Instructors*: Vasileios Tzoumas, Assistant Professor, Aerospace Engineering (he/him/his; vtzoumas@umich.edu; https://vasileiostzoumas.com/)
- *Teaching Assistants*: TBD

Course description

Visual Navigation for Autonomous Aerial Vehicles (VNA2V) covers theoretical foundations of vision-based navigation as well as implementation and testing of algorithms in a photo-realistic Unity-based simulator. Lectures will explore fundamental tools and results from a wide spectrum of disciplines (non-convex and combinatorial optimization, reinforcement learning, geometric control, non-linear estimation, geometry) that underlie modern techniques for real-time control and trajectory optimization, robot perception, and 3D computer vision (including visual-inertial navigation and SLAM), and machine learning. Implementation and testing will be based on C++ and the Robot Operating System (ROS) (introduction to both will be given). Students will be given access to an advanced drone simulator, and will be able to implement and test state-of-the-art algorithms and learn about the cutting edge of autonomous navigation. The final portion of the class includes an individual or team-based project that has the goal of advancing the state-of-the-art in vision-based navigation, according to students' interests.

Prerequisites

Basic knowledge real-time embedded programming (EECS 461 or similar), optimal estimation and control (AEROSP 550 or AEROSP 575 or similar), linear algebra (MATH 214 or ROB 501 or similar), or permission of the instructors. Coding assignments are based on ROS and openCV, and are in C++ (we will provide the necessary introduction to C++, ROS, and openCV).

Lectures and Laboratory Schedule

Week	Dates	Lecture topic	Lab	Platform
1	Jan 20, 22	Introduction	Lab 1: Linux, C++, Git	-
2	Feb 1, 3	3D Geometry	Lab 2: ROS	-
3	Feb 5, 8, 10	Geometric Control	Lab 3: 3D trajectory following	Drone simulator
4	Feb 19, 22, 24	Trajectory Optimization	Lab 4: 3D trajectory optimization	Drone simulator
5	Feb 26, Mar 1, Mar 3	2D Computer Vision	Lab 5: feature detection	Drone simulator
6	Mar 5, 8, 10	2-view Geometry and Minimal Solvers	Lab 6: object localization	Datasets
7	Mar 12, 15	Non-Minimal Solvers and Visual Odometry	Lab 7: GTSAM	Drone simulator
8	Mar 17, 19, 22	Place Recognition	Lab 8: ML for robotics	Drone simulator
9	Mar 24, 26, 29	SLAM and Visual-Inertial Navigation	Lab 9: SLAM	Drone and datasets
10	Mar 31, Apr 2, 5	Outlier-Robust Estimation	Final project	Depending on project
11	Apr 7, 9	**Advanced Topics**:† Combinatorial Methods for Robust Feature Tracking	Final project	Depending on project
12	Apr 12, 14	**Advanced Topics**:† Metric-Semantic Understanding	Final project	Depending on project
14	Apr 16, 19, 20	Guest Lectures and Students Presentations	Final project	Depending on project

†**Advanced Topics** lectures will be decided depending on interest and guest lecturers.

Assignments and Grading

Graded **assignments** include lab exercises and a final project. Lab exercises include a set of questions covering both theoretical aspects (discussed during the lectures) and the result of implementation and testing of algorithms (assigned during the labs). The code written in some of the labs will also be subject to evaluation. Assignments for Lab 1 and 2 are individual. Labs 3-9 include a mix of team assignments (we expect teams of 2-3 students) and individual assignments. Lab exercises are typically announced on Wednesdays at 3pm (EST), and the corresponding assignments are due the following Wednesdays at 11:59pm (EST).

The final project will be completed in teams formed according to the students' interests. The grading for the final project will be based on:

1) Technical report (formatted according to ICRA guidelines).
2) Final demo, showcasing the outcome of the project.
3) Team presentation, including videos describing the project and its implementation.

List of dates for the assignments

Week	assignment	Individual/ Team	Announced (3pm)	Due (11:59pm)
1	Lab 1: Linux, C++, Git	I	W Jan 20	W Jan 27
2	Lab 2: ROS	I	W Jan 27	W Feb 3
3	Lab 3: 3D trajectory following	I/T	W Feb 3	W Feb 10
4	Lab 4: 3D trajectory optimization	I/T	W Feb 10	W Feb 17
5	Lab 5: Feature detection	I/T	W Feb 17	W Feb 24
6	Lab 6: Object localization	I/T	W Feb 24	W Mar 3
7	Lab 7: GTSAM	I/T	W Mar 3	W Mar 10
8	Lab 8: Place recognition	I/T	W Mar 10	W Mar 17
9	Lab 9: SLAM	I/T	W Mar 17	W Mar 24
10-14	Final project (report, demo, presentation)	T	W Mar 24	W Apr 19, 20

The grading policy is as follows

- **[60%] Lab exercises:** Set of questions covering both theoretical aspects (discussed during the lectures) and the result of implementation and testing of algorithms (assigned during the labs). You will upload your solutions and code produced during the labs to a github.umich.edu repository.
- **[25%] Final project report, demo, presentation:** A final demo, together with a technical report (formatted according to ICRA guidelines) detailing and motivating the choice of final project and your approach to address and solve the corresponding problem. A final presentation reporting on the outcome of your implementation, including experimental results and videos describing the overall project.
- **[10%] Participation, TA evaluation:** Your participation in the lectures and labs, including attendance and evaluation of your performance in the labs by the VNA2V staff.
- **[5%] Team members' assessment:** Assessment of your performance by your teammates.

Resources

There is **no required textbook (lecture notes will be provided)**. The following is a **recommended** book (also available online):

- "*An Invitation to 3-D Vision: From Images to Geometric Models*" by Yi Ma, Jana Kosecká, Jana, Shankar Sastry, and Stefano Soatto, Springer (ISBN: 9780387008936).

Pointes to other relevant references (tutorial/survey papers as well as technical papers) will be provided during each lecture.

Additional Policies

- **Late work** will receive reduced credit, according to the following policy: if the grade achieved by a submission is X, the grade resulting from a late submission is computed as: X_delayed = X * max(1 - 0.10 * d, 0), where d is the number of days of delay. For instance, if X = 10 and there is no late submission, then X_delayed = 10, but if the assignment is submitted 2 days after the deadline, then X_delayed = 10 * (1 - 0.2) = 8.
- **Collaboration is essential** in the field, and therefore is essential for all assignments. Within teams, teamwork is an absolute necessity, and we expect that teams will work together to generate the technical content and solve the exercises for each lab.
- Across teams, we encourage collaboration and discussion. At the same time, we stand by the core values of engineering ---please refer to U-M College of Engineering's Statement of Values and Honor Code--- which ask for (i) *explicitly crediting any collaborators* (see acknowledgement below for an example), (ii) *forbidding the appropriation of code, data, plots, or writing across teams, even with modifications or paraphrasin*g. That is, any writing included in a lab assignment must be authored by your team; any data or plots must come from your team.
- For the **final project**, full collaboration within the team on all aspects of the challenge is encouraged. On the values of fairness and hard-work, every member of the team is expected to contribute a roughly equal share to the design, implementation, and presentation of the challenge.

Please contact a member of the course staff for any clarification about the policies above.

Acknowledgements

The course is based on material developed at the Massachusetts Institute of Technology (MIT) by Luca Carlone, Kasra Khosoussi, Markus Ryll, Golnaz Habibi, Vasileios Tzoumas, and numerous TAs (individually acknowledged at each lab assignment for their contributions in developing the assignment), for the MIT course 16.485, Visual Navigation for Autonomous Vehicles.