3D Computer Vision - Practical Project - 2025

From 2D images to 3D scene reconstruction

Overview

The Course Project is an opportunity for you to apply what you have learned in class. You can use existing models (algorithms) and apply them to tackle this project. You can also build new models (algorithms), or a new variant of existing models, and apply it to tackle this project.

You might look at recent deep learning publications from top-tier computer vision conferences, as well as other resources.

You are welcome to come to the reserved meeting room during the time slots scheduled for this project to brainstorm and discuss your project ideas.

Learning outcomes

The objective of this project is to front your knowledge in computer vision (and understanding of fundamentals and theory) with practice, rather than only using an available network architecture (as a black-box based on AI) without understanding and mastering what are the main steps involved in 3D reconstruction.

Important Dates

Project presentation: April 1
Midterm progress report: due April 23
Final course project: due June 2
Final project presentations: May 28

Grading Policy: The Course Project contributes to 50% of your final grade. Specifically, it is broken down into a few main parts:

- Midterm Project Report: 10%
- Final Project Report: 30%
- Final Project Presentation: 10%

Project description

- Title: From 2D images to 3D scene reconstruction.
- Problem statement: reconstruct in 3D an indoor scene with rigid objects from two or more 2D views of this scene.

  > *3D Reconstruction is the task of creating a 3D model or representation of an object or scene from 2D images or other data sources. The goal of 3D reconstruction is to create a virtual representation of an object or scene that can be used for a variety of purposes, such as visualization, animation, simulation, and analysis. It can be used in fields such as computer vision, robotics, and virtual reality [1].*
  >
  > *[1] https://paperswithcode.com/task/3d-reconstruction*

- Dataset: You can either use an existing dataset (such as ETH3D) or build your own dataset for tests, experimentations, evaluations and comparisons.

  > *ETHD is a multi-view stereo benchmark / 3D reconstruction benchmark that covers a variety of indoor and outdoor scenes. Ground truth geometry has been obtained using a high-precision laser scanner. A DSLR camera as well as a synchronized multi-camera rig with varying field-of-view was used to capture images [2].*

[2] https://paperswithcode.com/dataset/eth3d

In your first report you'll have to explain: What data will you use? If you are collecting new datasets, how do you plan to collect them? How will you evaluate your results? Qualitatively, what kind of results do you expect (e.g. plots or figures)? Quantitatively, what kind of analysis will you use to evaluate and/or compare your results (e.g. what performance metrics or statistical tests)? All these questions must be anticipated before choosing and implementing a method.

- Technical approach: You are free to use any existing method[1], or to propose a variant of existing methods. In your first report you'll have to explain:

  What method(s) or algorithm(s) are you proposing? If there are existing implementations, will you use them and how? How do you plan to improve or modify such implementations?

Midterm Progress Report

Your midterm progress report (PDF file) should be at most 4 pages. The following is a suggested structure for your report:

- Author(s)
- Introduction: this section introduces the problem, and the overall plan for approaching the problem;
- Problem statement: Describe the problem precisely specifying the dataset to be used, expected results and evaluation;
- Technical Approach: Describe the methods you intend to apply to solve the given problem;
- Intermediate/Preliminary Results: State and evaluate your results up to the current date.

Submission by email to alain.tremeau@univ-st-etienne.fr**.**

Final Submission

Your final write-up should be between **6 - 8** pages. You will submit the following:

1. A PDF file of your final report (to be sent by email to alain.tremeau@univ-st-etienne.fr)
2. A link to your Git repository. You can put this somewhere else, for example a private repository.

Report. The following is a suggested structure for the report:

- Author(s)
- Abstract: It should not be more than 300 words
- Introduction: this section introduces the problem, and the overall plan for approaching the problem
- Background/Related Work: This section discusses relevant literature for this project
- Approach: This section details the framework of this project. Be specific, which means you might want to include equations, figures, plots, etc.
- Experiments/Analysis: This section begins with what kind of experiments you're doing, what kind of dataset(s) you're using, and what is the way you measure or evaluate your results. It then shows in details the results of your experiments. By details, we mean both quantitative evaluations (show numbers, figures, tables, etc.) as well as qualitative results (show images, example results, etc.).
- Conclusion: What have you learned? Suggest future ideas.
- References: This is absolutely necessary.

Supplementary Material is not counted toward your 6-8 page limit. Examples of things to put in your supplementary material:

- Source code (if your project proposed an algorithm, or code that is relevant and important for your project.).

[1] As example https://dust3r.europe.naverlabs.com/, see also Tutorial — COLMAP 3.8 documentation or https://docs.opencv.org/4.x/d4/d18/tutorial_sfm_scene_reconstruction.html

- Cool videos, interactive visualizations, demos, etc.

Examples of things to not put in your supplementary material:

- All of your source code. Instead give a link to a Git repository
- Various ordinary data preprocessing scripts.
- Any code that is larger than 1MB.
- Model checkpoints.
- A computer virus.

Examples of reports

- https://web.stanford.edu/class/cs231a/prev_projects_2022/Project_final_report_2.pdf
- web.stanford.edu/class/cs231a/prev_projects_2022/1Final_Project_Report.pdf
- web.stanford.edu/class/cs231a/prev_projects_2022/CS231A_Final_Project_Write-up.pdf
- web.stanford.edu/class/cs231a/prev_projects_2022/FinalReport_cs231a.pdf

Project Presentations

The contents of the oral presentation should be similar to the contents of the final project report, with more emphasis on your approach and results. A rough suggested allocation of time is the following:

- Introduction: Introduce the motivation and your problem, and then relevant prior work and approaches for this problem (2 minute)
- Approach: Provide an overview of your approach and highlight the key technical aspects you worked on. If you have not fully finished implementing your approach, highlight what parts are done and which are still planned for the final report. (5 minutes)
- Experiments and evaluation: Explain the experimental setup and summarize the quantitative results (numbers, figures, tables, etc.) and qualitative results (images, example results, etc.). If the results are not as expected, explain what the challenges are and how you plan to improve the results in the final report. (5 minutes)

Grading

We will be grading for completeness and clarity more than the quality of your results. The rubric will be split into the following categories:

- 20% for problem statement, motivation, and background;
- 30% for technical approach;
- 30% for sufficient and informative quantitative and qualitative results;
- 10% for visual style;
- 10% for addressing questions raised during Q&A.

Grades for each category will be assigned on a standard scale out of 5.

Collaboration Policy

You can work in teams of up to **3** people. We do expect that projects done with 3 people have more impressive writeup and results than projects done with 2 people. The contribution of each team member should be clearly stated and well balanced between team members.

Author contributions can be defined as follow: Conceptualization; Methodology; Software; Validation; Formal analysis; Investigation; Writing report(s); Supervision.

Honor Code

You may consult any papers, books, online references, or publicly available implementations for ideas and code that you may want to incorporate into your strategy or algorithm, so long as you clearly cite your sources in your code and your writeup. However, under no circumstances may you look at another group's code or incorporate their code into your project.