L-Università ta' Malta
Faculty of Information &
Communication Technology

Department
of Artificial
Intelligence

Group Project

ARI 2129 - Principles of Computer Vision for AI

Project Title: 3D Scene Reconstruction from Video Using Structure from Motion

Deadlines: 9th June 2025
Group Size: Maximum of 5 students (Equal effort will be assumed)
Register here: https://forms.gle/ehfdR3CuwrYuKzzv7
Maximum Marks: 100 (50% of the entire study unit)

Plagiarism and Authenticity:

- No plagiarism will be tolerated. All sources must be adequately referenced in the code and documentation.
- Generative AI (GenAI) is allowed as long it is documented and in line with the instructions in this documentation and reported in the Generative AI Journal.
- The use of generative AI needs to be declared and documented.
- Students may be asked for impromptu interviews to check their knowledge of the submitted content. Any information in the submitted documentation not being adequately explained by students and without its respective sources will be immediately reported as plagiarism.

Marking Scheme:
The project will be graded based on the following criteria:

- **Jupyter Notebook (70%):**
  - Criteria:
    - Clarity, organisation, and readability of the code.
    - Completeness of the implementation.
    - Correctness of the results.
    - Quality of visualisations.
    - Explanation of the code and algorithms.
  - Marking Breakdown:
    - Video Acquisition (10%)
    - Frame Extraction (5%)
    - Feature Detection and Description (10%)
    - Feature Matching and Outlier Rejection (10%)
    - Essential/Fundamental Matrix Computation (5%)
    - Camera Pose Estimation (10%)
    - 3D Point Triangulation and Scene Visualisation (10%)
    - Evaluation and Analysis (10%)
- **Project Documentation (20%):**
  - Depth and breadth of the literature review.
  - Thoroughness and clarity of the description of the method.
  - Justification of implementation decisions.

  - Quality and rigour of the evaluation.
  - Clarity and completeness of the GenAI usage description (if applicable).
  - Clarity of writing, organisation, and formatting.
  - Completeness and accuracy of the team contributions section.
- **AI Journal (10%)**

Project Description

Teams will develop a software system to reconstruct a 3D model of a scene or object from a short video sequence captured using a mobile phone or webcam. Using classical computer vision techniques, the system will implement a simplified Structure from Motion (SfM) pipeline. Teams are encouraged to explore different aspects of the SfM process and experiment with various algorithms and parameters. Generative AI tools can assist in the development process, but their use must be documented in the AI Journal Section.

Deliverables:

1. **Jupyter Notebook:** A well-structured Jupyter Notebook containing all code, visualisations, and explanations.
2. **Detailed Project Documentation:** A comprehensive document (PDF format) covering the following:
   - **Literature Review:** A review of relevant academic literature on Structure from Motion, including the algorithms and techniques used in the project.
   - **Methods:** A detailed description of the SfM pipeline implemented, including:
     - Video capture and frame extraction process (explaining how you capture video).
     - Feature detection and description algorithms (SIFT, SURF, ORB, or others).
     - Feature matching algorithm(s) and outlier rejection method (RANSAC).
     - Essential/Fundamental matrix computation.
     - Camera pose estimation.
     - 3D point triangulation.
   - **Implementation Decisions:** Justification of design choices, algorithm selection, parameter tuning, and any modifications made to the standard algorithms.
   - **Evaluation:** A thorough evaluation of the reconstructed 3D model, including:
     - Qualitative assessment (visual inspection of the reconstructed scene).
     - Quantitative assessment (if possible, e.g., reprojection error).
     - Discussion of limitations, error sources, and potential improvements.
   - **Use of Generative AI:** A detailed description of how Generative AI tools were used in the project (if applicable), including:
     - Specific tools used (e.g., ChatGPT, Bard, etc.).
     - Tasks for which GenAI was employed (e.g., code generation, debugging, documentation).
     - Prompts used.
     - Evaluation of the GenAI output (accuracy, correctness, efficiency).
     - Discussion of any issues encountered and how they were resolved.

- **Team Contributions:** A clear statement of each team member's contributions to the project.

Project Stages

1. **Video Acquisition (10%):**
    - Teams will capture a short video sequence (5-10 seconds) of a static object or scene using a mobile phone or webcam.
    - Teams can capture video from locations such as the Faculty of ICT or the University of Malta. Teams should not use OpenCV's VideoCapture as the primary method for video capture but can use other methods, such as platform-specific APIs or browser APIs. The chosen method must be explained and justified.
    - The video should have sufficient overlap between frames.
        - To achieve good results, consecutive frames should overlap by approximately 70-80%. This means that a feature visible in one frame should still be visible in the next few frames.
        - **Tips for capturing a good video:**
            - **Camera Motion:** Use smooth, continuous camera motion. Avoid abrupt movements or rotations. A circular or lateral motion around the object is often effective.
            - **Focus:** Ensure the object or scene is in focus throughout the video.
            - **Lighting:** Use consistent and diffuse lighting to avoid shadows and overexposed areas.
            - **Object/Scene Selection:** Choose a static object or scene with sufficient texture and distinct features. Avoid featureless surfaces (e.g., a blank white wall).
        - Teams should document what video capture techniques worked best and why in their project report.
    - Teams should consider and document the lighting conditions and camera motion during capture.

2. **Frame Extraction (5%):**
    - Teams will extract individual frames from the captured video and are to choose a feasible interval.

3. **Feature Detection and Description (10%):**
    - Teams will implement feature detection and description using OpenCV.
    - Teams will select and implement at least two feature detection/description algorithms (SIFT, SURF, ORB, or others).
    - A comparison of the chosen algorithms' performance is required in the evaluation.
        - **Hints for comparison:**
            - **Number of detected features:** How many keypoints does each algorithm detect in the same image? More keypoints are not always better, but a sufficient number is required for reliable matching. Students could create a plot of the number of features detected vs frame number.

    - **Matching performance:** How many correct matches are found by each algorithm? This can be evaluated by visually inspecting a subset of the matches. Students could visualise the matches using a small number of image pairs (e.g., 5-10 pairs).
    - **Computational speed:** How long does each algorithm take to detect and describe features? Students can use Python's time module to measure the execution time for this stage.
    - **Robustness to image transformations:** How well do the algorithms perform under different conditions, such as changes in scale, rotation, illumination, and viewpoint? This can be done qualitatively by selecting image pairs with significant viewpoint changes and observing the number of matches.
    - **Qualitative assessment:** How well do the features visually correspond between frames? Students can create a montage of a few frames, and the feature matches.
  - Teams must explain the algorithms' parameters and justify their values.

4. **Feature Matching and Outlier Rejection (10%):**
   - Teams will implement a feature-matching algorithm (e.g., Brute-Force, FLANN).
   - Teams will implement RANSAC to remove outliers.
   - Visualisation of feature matches before and after RANSAC is required in the Jupyter Notebook.

5. **Essential/Fundamental Matrix Computation (5%):**
   - Teams will compute the essential matrix (for calibrated cameras) or the fundamental matrix (for uncalibrated cameras).
   - Teams will use OpenCV's findEssentialMat or findFundamentalMat functions, but they must demonstrate an understanding of the underlying calculations (e.g., by providing a clear explanation in the documentation).
   - The report must explain the difference between calibrated and uncalibrated cameras.

6. **Camera Pose Estimation (10%):**
   - Teams will decompose the essential or fundamental matrix to obtain the rotation and translation matrices.
   - Teams will use OpenCV's recoverPose function.
   - The report must explain the chirality problem and how the team resolved it.

7. **3D Point Triangulation and Scene Visualisation (10%):**
   - Teams will triangulate the 3D positions of the matched feature points.
   - Teams will use OpenCV's triangulatePoints function.
   - The Jupyter Notebook should explain the triangulation process.

- Teams will visualise the reconstructed 3D scene using a suitable library (Matplotlib, Open3D, CloudCompare, or others).
- The visualisation should be interactive.

8. **Evaluation and Analysis (10%):**
   - Teams will evaluate the quality of the reconstruction, considering both qualitative and quantitative aspects.
   - Teams will compare the results obtained with different feature detection and matching methods.
   - Teams will discuss the limitations of their implementation, potential sources of error, and possible improvements.

Generative AI Journal Guidelines

The objective of this document is to critically examine and reflect on how generative AI models, like Gemini, Claude, Grok or ChatGPT, were used in this project. This documentation will focus on ethical considerations, the methodology employed, and the specific contributions of generative AI to improving your work. You must cite any references or resources you consulted about generative AI. The document, which is separate from the technical paper, has a page limit of 10 pages, including references and should contain the following sections:

1. **Introduction**
   a. Briefly describe the generative AI models you used and the rationale behind that choice. (Maximum of 1 page)

2. **Ethical Considerations**
   a. Discuss the ethical aspects of using generative AI in your project. This should include issues like data bias and privacy. (Maximum of 1 page)

3. **Methodology**
   a. Outline the methods and steps to integrate the generative AI model into your work. Which tools did you use, and in which sequence? Did you create your pipeline, or did you just use one tool? There is no wrong answer, and this journal aims for transparency and accountability.

4. **Prompts and Responses**
   a. List down the specific prompts used with the generative AI model that you found noteworthy. For each prompt, include the generated response and explain how it improved your project. It is advisable to use screenshots for this part.

5. **Improvements, Errors and Contributions**
   a. Discuss the areas where generative AI-enhanced your work or instances where the output contained errors. This can include but is not limited to data

analysis, formulation of ethical considerations, literature review enhancement, or idea generation. Highlight specific cases where this happened.

6. **Individual Reflection**
    a. Reflect on your personal experience using generative AI in your project. Discuss what you learned, what surprised you, and how your perspective on using AI in academic projects has changed, if at all. Did Gen AI help you be more efficient? Do you feel you wasted more time when you used it? For which part was it most helpful? Literature review, debugging?

7. **References and List of Resources Used**

End of Project Specifications