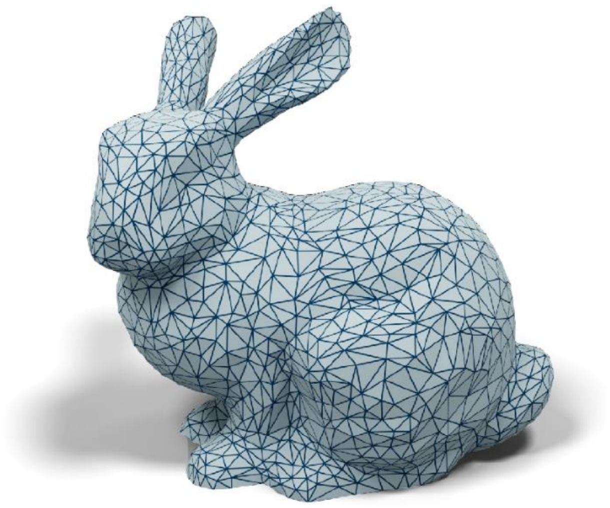

3D Computer Vision

Professor Ahad Harati

Homework 1:

Introduction to feature matching and tracking and Transformations

Designed By:

Mahdieh Alizadeh - Ali Goli

Mahdieh20201@gmail.com

Spring 2025

Preface

Welcome to your first homework!

- Feature extraction and transformation are fundamental concepts in computer vision and image processing, playing a crucial role in tasks such as object recognition, image registration, and 3D reconstruction. Feature-based techniques allow us to identify key structures in an image, establish correspondences across different views, and apply geometric transformations for spatial alignment in both **2D and 3D spaces**.
- This assignment is designed to provide hands-on experience with **feature detection, matching, and geometric transformations**.Through this homework, students will explore the principles of **feature-based vision systems**, implement algorithms from scratch, and analyze transformation properties in **image processing and 3D vision**.

The homework is structured into the following sections**:**

- **Feature Manager (Feature Extraction & Matching):**
  - ✓ Implementing feature detection algorithms such as **SIFT, ORB, or Harris Corner Detector**.
  - ✓ Performing feature matching using techniques such as **Brute Force Matching and FLANN-based Matching**.
  - ✓ Evaluating feature robustness under different lighting and perspective changes.
- **Transformations in 2D and 3D:**
  - ✓ Understanding and applying **affine, projective transformations**.
  - ✓ Computing **2D transformations** for image alignment and registration.

The goal of this assignment is not only to implement feature-based vision techniques but also to develop an **intuitive and mathematical understanding** of how feature extraction, matching, and geometric transformations contribute to real-world applications. By the end of this homework, students should be able to:

- Detect and extract **key features** from images using different algorithms.
- Implement and analyze **feature matching techniques** for image correspondence.
- Compute and apply **2D and 3D transformations** for spatial alignment.

We hope this assignment enhances your understanding of **feature extraction, matching, and transformations in 2D and 3D** and encourages further exploration into advanced topics such as **epipolar geometry, structure-from-motion, and 3D vision**.

Note:

• Include well-commented code and relevant plots in your notebook.

• Clearly present all comparisons and analyses in your report.

• Ensure reproducibility by specifying all dependencies and configurations.

Submission:

The deadline for this homework is 1403/12/29 (March 19th 2025) at 11:59 PM

Please submit your work by following the instructions below:

- Place your solution alongside the Jupyter notebook(s)
  - Your Written Solution must be a single PDF file named *HW1_Solution.pdf*
  - If there is more than one Jupyter notebook, put them in a folder named Notebooks
- Zip all files together with the following naming format: *C3V_HW1_[Fullname].zip*
- If you have any questions about this homework, please ask them in our Telegram Group.
- If you are using any references to write your answers, **consulting anyone or using AI, please mention them in the appropriate section.**

Grading:

The grading will be based on the following criteria, with a total of 100 points:

Tasks	points
Task 1: Implemetation and Results	60
Task 2: Implementation and Results	30
Clarity and Quality of Code	5
Clarity and Quality of Report	5

Keep up the great work and best of luck with your submission! ☺

Task 1.

Feature Extraction, Matching, and Landmark Analysis

In this task, you will implement **three feature extraction methods** (SIFT, ORB, and Harris Corner Detector) and **two feature matching techniques** (Brute Force Matching and FLANN). You will analyze the efficiency and effectiveness of each method by evaluating the runtime, number of detected candidate points, and number of matched landmarks across multiple frames. Also the **output visualization videos over frames should be attached to your report** (three of them)

A **candidate point** is any feature point detected in a single frame. A **landmark** is a candidate point that appears in at least **five consecutive frames**, making it a stable feature across a sequence of images.

Step 1: Implement Feature Extraction Methods (Write three separate functions for SIFT-ORB-Harris Corner Detection)

Step 2: Implement Feature Matching Methods

Step 3: Track Landmarks Over Frames (Keep track of **candidates (detected keypoints in a single frame)** and **landmarks (features seen in at least five consecutive frames)**.

Step 4: **Visualize and Compare** Performance in a table (**Mention the specific parameter of algorithm**).

Note:

To evaluate the performance of your feature extraction and matching algorithms, you will work with **a subset of the KITTI Vision Benchmark Dataset** which is provided in *data* directory.

Figure 1: Matching features from the first two frames in the KITTI dataset.

Task 2.

Projective Transformation and Vanishing Points in R^2

In this task, you will explore the effects of projective transformations on geometric shapes and parallelism in R^2. A projective transformation alters the perspective of an image, often making parallel lines converge at a vanishing point.

You will start by drawing a square in 2D space, apply a projective transformation, and analyze the impact on parallel lines by computing their vanishing points before and after the transformation.

Step 1: Draw a Square in $\boldsymbol{R^2}$

- ✓ Define the four corner points of a square in a 2D coordinate space.
- ✓ Plot the square using Matplotlib or another visualization tool.

Step 2: Apply a Projective Transformation

- ✓ Define a projective matrix (3×3 projective transformation matrix).
- ✓ Apply the transformation to each vertex of the square, creating a new quadrilateral.
- ✓ Plot the transformed shape to visualize how the square has changed.

Step 3: Compute the Vanishing Point for Two Parallel Lines in the Original Shape

- ✓ Select two parallel lines from the original square.
- ✓ Represent each line using two points and compute their intersection in homogeneous coordinates.
- ✓ Convert the intersection to Cartesian coordinates and print the vanishing point.

Step 4: Compute the Vanishing Point in the Transformed Shape

- ✓ Select the corresponding two lines in the transformed shape.
- ✓ Compute their vanishing point using the same method as in Step 3.
- ✓ Print and visualize the vanishing points on the transformed shape.