

# Project 4: Camera Calibration and Object Projection

**Name:** Priyanshu Ranka (NUID- 002305396)

**Subject:** PRCV

**Professor:** Prof. Bruce Maxwell

**Semester:** Spring 2025

**3 Time-Travel days were utilized to complete this project.**

**OS used:** Windows 11, Visual Studio 2022

## Overview

This project involves camera calibration, feature detection, and object projection using C++. The provided code and data facilitate the implementation and evaluation of these tasks.

## File Structure

Submission Files/

```
| -- calibration.csv  
| -- cameraCalibration.cpp  
| -- featureDetection.cpp  
| -- objProjection.cpp  
| -- Project 4 Report.pdf
```

## File Descriptions

- **calibration.csv:** This file contains precomputed camera calibration data, including intrinsic and extrinsic parameters. It is used by the cameraCalibration.cpp program to calibrate the camera model before applying projection techniques.
- **cameraCalibration.cpp:** This C++ program performs camera calibration using a set of input images. It utilizes OpenCV functions to estimate intrinsic parameters (focal length, principal point, and distortion coefficients) and saves the calibration results for later use.
  - Input: A set of calibration images.
  - Output: Camera matrix and distortion coefficients stored in calibration.csv.
  - Task covered: 1 to 4

- **objProjection.cpp:** This program projects a 3D object onto a 2D image using the camera calibration parameters. It reads the calibration.csv file, uses the computed camera matrix and distortion coefficients, and applies projection transformations.
  - Input: 3D object coordinates and an image.
  - Output: Image with the projected object.
  - Tasks Covered: 4 to 6
- **featureDetection.cpp:** This C++ program detects key features in images using feature extraction techniques like SIFT, ORB, or SURF. It finds and matches key points between images, which can be useful for tasks like structure from motion and object recognition.
  - Input: An image or a set of images.
  - Output: Image with detected keypoints visualized.
  - Tasks Covered: Task 7
- **Project 4 Report.pdf:** This document provides a detailed explanation of the project, including methodology, algorithms, experimental results, and conclusions. It serves as a reference for understanding the implementation details.

## How to Run

1. **Compile the code** using a C++ compiler with OpenCV support:
2. `g++ cameraCalibration.cpp -o cameraCalibration `pkg-config --cflags --libs opencv4``
3. `g++ featureDetection.cpp -o featureDetection `pkg-config --cflags --libs opencv4``
4. `g++ objProjection.cpp -o objProjection `pkg-config --cflags --libs opencv4``
5. **Run each executable** with appropriate input data:
6. `./cameraCalibration`
7. `./featureDetection`
8. `./objProjection`

## Requirements

- C++ compiler (GCC or Clang)
- OpenCV library
- CMake (optional for building the project)