|  |  |  |  |
| --- | --- | --- | --- |
| **Machine Problem No. 3** | | | |
| **Topic:** | **Module 2.0: Feature Extraction and Object Detection** | **Week No.** | 6-7 |
| **Course Code:** | **CSST106** | **Term:** | 1st Semester |
| **Course Title:** | **Perception and Computer Vision** | **Academic Year:** | 2024-2025 |
| **Student Name** | **Lesly-Ann B. Victoria** | **Section** | BSCS-4B |
| **Due date** |  | **Points** |  |

**Machine Problem No. 3: Feature Extraction and Object Detection**

**Objective:**

The objective of this machine problem is to implement and compare the three feature extraction methods (**SIFT**, **SURF**, and **ORB**) in a single task. You will use these methods for feature matching between two images, then perform image alignment using **Homography** to warp one image onto the other.

**Problem Description:**

You are tasked with loading two images and performing the following steps:

1. Extract keypoints and descriptors from both images using **SIFT**, **SURF**, and **ORB**.
2. Perform feature matching between the two images using both **Brute-Force Matcher** and **FLANN Matcher**.
3. Use the matched keypoints to calculate a **Homography matrix** and align the two images.
4. Compare the performance of SIFT, SURF, and ORB in terms of feature matching accuracy and speed.

You will submit your code, processed images, and a short report comparing the results.

**Task Breakdown:**

**Step 1: Load Images**

* Load two images of your choice that depict the same scene or object but from different angles.

**Step 2: Extract Keypoints and Descriptors Using SIFT, SURF, and ORB**

* Apply the **SIFT** algorithm to detect keypoints and compute descriptors for both images.
* Apply the **SURF** algorithm to do the same.
* Finally, apply **ORB** to extract keypoints and descriptors.

**Submission**:

* Python code (feature\_extraction.py)
* Processed images showing keypoints for SIFT, SURF, and ORB (e.g., sift\_keypoints.jpg, surf\_keypoints.jpg, orb\_keypoints.jpg).

**Step 3: Feature Matching with Brute-Force and FLANN**

* Match the descriptors between the two images using **Brute-Force Matcher**.
* Repeat the process using the **FLANN Matcher**.
* For each matching method, display the matches with lines connecting corresponding keypoints between the two images.

**Submission**:

* Python code (feature\_matching.py)
* Processed images showing matches for Brute-Force and FLANN for each algorithm (e.g., sift\_bf\_match.jpg, sift\_flann\_match.jpg).

**Step 4: Image Alignment Using Homography**

* Use the matched keypoints from **SIFT** (or any other method) to compute a **homography matrix**.
* Use this matrix to warp one image onto the other.
* Display and save the aligned and warped images.

**Submission**:

* Python code (image\_alignment.py)
* Aligned and warped images (e.g., aligned\_image.jpg, warped\_image.jpg).

**Step 5: Performance Analysis**

1. **Compare the Results**:
   * Analyze the performance of **SIFT**, **SURF**, and **ORB** in terms of keypoint detection accuracy, number of keypoints detected, and speed.
   * Comment on the effectiveness of **Brute-Force Matcher** versus **FLANN Matcher** for feature matching.
2. **Write a Short Report**:
   * Include your observations and conclusions on the best feature extraction and matching technique for the given images.

**Submission**:

* A PDF or markdown document (performance\_analysis.pdf or performance\_analysis.md).

**Submission Guidelines:**

* **GitHub Repository**:
  + Create a folder in your CSST106-Perception and Computer Vision repository named Feature-Extraction-Machine-Problem.
  + Upload all code, images, and reports to this folder.
* **File Naming Format**: **[SECTION-LASTNAME-MP3]** 4D-LASTNAME-MP3
  + 4D-BERNARDINO-SIFT.py
  + 4D-BERNARDINO-Matching.jpg

**Additional Penalties:**

* **Incorrect Filename Format**: -5 points
* **Late Submission**: -5 points per day
* **Cheating/Plagiarism**: Zero points for the entire task

**Rubric for Feature Extraction and Object Detection Machine Problem**

| **Criteria** | **Excellent**  **(90-100%)** | **Good**  **(75-89%)** | **Satisfactory**  **(60-74%)** | **Needs Improvement**  **(0-59%)** |
| --- | --- | --- | --- | --- |
| **Step 2: Feature Extraction (SIFT, SURF, ORB)** | All feature extraction methods (SIFT, SURF, ORB) are implemented correctly. The extracted keypoints are clearly visualized and well explained. The code is well-commented, and outputs are saved properly. | Feature extraction is implemented correctly, but there may be minor visualization issues or explanations lacking depth. | At least two methods are implemented correctly, with basic explanations and some issues with visualization or code. | Feature extraction methods are incomplete, implemented incorrectly, or not explained well. Poor or no visualization provided. |
| **Step 3: Feature Matching (Brute-Force and FLANN)** | Both Brute-Force and FLANN matchers are implemented correctly, and keypoint matches are clearly visualized with detailed explanations. The matching performance for each method is analyzed. | Both matchers are implemented correctly, but there may be minor issues with the visualization, or the explanation lacks depth. | At least one matcher is implemented correctly, with basic explanations and minimal analysis of matching performance. | Feature matching methods are incomplete, implemented incorrectly, or poorly explained. Matches are not visualized, or results are unclear. |
| **Step 4: Image Alignment Using Homography** | The Homography matrix is computed correctly using matched keypoints, and the image is aligned and warped successfully. The output is visually accurate, and the process is well explained. | The Homography matrix is computed correctly, but the alignment has minor issues, or the explanation lacks depth. | The Homography matrix is computed, but there are significant alignment issues, or the explanation is basic. | Homography computation is incorrect or incomplete. Image alignment does not work as expected, or no explanation is provided. |
| **Step 5: Performance Analysis** | The performance analysis is thorough, comparing the accuracy and speed of SIFT, SURF, and ORB, and evaluating the effectiveness of Brute-Force and FLANN. The conclusion is insightful and well-supported. | The performance analysis is good but lacks some depth in comparing the methods or has minor gaps in the evaluation of the matchers. | The performance analysis is basic, with minimal comparison or weak conclusions. Some methods or matchers are not evaluated. | The performance analysis is incomplete or missing. Little to no comparison or evaluation of methods and matchers is provided. |