Name: Verdad, Jane Benneth Dione
Year & Section: BSCS IS 4B
Date: September 23, 2024
Instructor: Sir Mark Bernardino

Feature Extraction and Object Detection

The initial step is uninstalling installed OpenCV and a package manager (apt-get) is used to install necessary dependencies like CMake and other libraries for compiling OpenCV with extra modules. The cloning of OpenCV and OpenCV contrib repositories is attempted, but it fails because the directories already exist. A build directory is then created, and CMake is configured to enable non-free modules, followed by building and installing OpenCV. After the installation, OpenCV packages (opencv-python-headless and opencv-contrib-python) are installed using pip to finalize the setup.

Task 1: SIFT Feature Extraction

The SIFT (Scale-Invariant Feature Transform) algorithm is used to detect and compute keypoints and descriptors from an image. After importing the required libraries, an image is loaded and converted to grayscale. The SIFT detector is initialized, and keypoints and descriptors are extracted from the grayscale image. These keypoints are then drawn onto the original image to visualize feature points, and the result is displayed using Matplotlib, showing the image with highlighted key points in various locations. Observations show that key points represent distinct image regions where SIFT identifies features invariant to scale and rotation.

Task 2: SURF Feature Extraction

The SURF (Speeded-Up Robust Feature) feature detection process begins by loading an image and converting it to grayscale. The SURF detector is initialized and applied to compute keypoints and descriptors. Keypoints are visually drawn on the image, which is displayed with matplotlib, highlighting the detected key points in a blue color, showing successful SURF feature extraction.

Task 3: ORB Feature Extraction

The ORB (Oriented FAST and Rotated BRIEF) algorithm is used to detect and compute keypoints and descriptors from the same image. The ORB detector is initialized with adjusted parameters to enhance feature detection. Keypoints are extracted and drawn onto the image with size and orientation information included. The result is displayed, marking the detected ORB keypoints using green circles, emphasizing distinct features within the image with a high density of feature points. Based on observations, ORB is a more efficient alternative to SIFT, focusing on speed, and it is ideal for tasks that require many features with less computation.

Task 4: Feature Matching using SIFT

This task shows feature matching between two images using the SIFT algorithm and a Brute-Force matcher. Both images are loaded and converted to grayscale. SIFT keypoints and descriptors are detected in both images. Using the BFMatcher with L2 norm, descriptors are matched and sorted by distance. The top matches are drawn on the image pairs, illustrating corresponding features between the two images. The result is displayed with lines connecting matching keypoints between and showing similarities between the both images. Matches reveal similar regions between the two images, which could indicate similarity in content or orientation.

Task 5: Applications for Feature Matching

Feature matching is applied to align two images of the same scene taken from different perspectives. After detecting and matching features using SIFT, the matched keypoints are filtered based on distance to identify the best matches. Using these matches, a homography matrix is computed, and one image is warped to align with the perspective of the other. Based on observations, using ORB with homography is efficient for image alignment and feature matching, although SIFT generally provides more precise feature matches.

Task 6: Combining SIFT and ORB

For the last task, both SIFT and ORB algorithms are combined to extract features and match them between two images. Keypoints and descriptors are detected separately with SIFT and ORB, and both feature sets are matched independently using a Brute-Force matcher. The combined method illustrates the differences in matching accuracy and keypoint detection between SIFT and ORB that highlights the trade-offs between SIFT's precision and ORB's efficiency.

The matches from both SIFT and ORB are visualized side by side for comparison. The combined image is displayed with both SIFT and ORB matches, showing the differences in feature matching between these two methods.