VNUHCM - UNIVERSITY OF SCIENCE

FACULTY OF INFORMATION TECHNOLOGY

LAB 03 REPORT

COMPUTER VISION

Lâm Thanh Ngọc - 21127118

Class: 21TGMT

Lecturers:

Phạm Minh Hoàng

Nguyễn Trọng Việt

Võ Hoài Việt

18th March 2024

Contents

1 Evaluation 2
2 Users guide 2
3 Function description 2
4 Proof images 4
5 References 5

1 Evaluation

Manipulation	Level of completion	Evaluation
Object detection using Local Feature	100%	Completed

2 Users guide

2.1. Open the terminal and navigate to the folder containing the source code.

2.2. Run the command 21127118.exe to execute the program.

2.3. Then, the program will show the valid command:
<Executable file> –sift <TemplateImagePath> <SceneImagePath> <OutputFilePath>

- <Executable file> is 21127118.exe.
- TemplateImagePath and SceneImagePath are the paths to the template and scene images.
- OutputFilePath is the path to the output image.

2.4. For example:
21127118.exe -sift Input\temp.jpg Input\scene.jpg Output\Result.jpg

3 Function description

In the code, there are 3 supporting functions:

- Mat read_image(string path) is used to read the image from the given path with the imread(path, IMREAD_COLOR) function of OpenCV.
- void show_and_save(string name, Mat image, string save_path) is used to show the image and save it to the given path with the imshow() and imwrite() functions of OpenCV.
- void draw_bb(Mat& scene, Mat& temp, Mat H) is used to draw the bounding box around the object in the scene image with the perspectiveTransform() function of OpenCV.

SIFT algorithm

The SIFT algorithm is used widely in computer vision for object recognition, image stitching, and 3D reconstruction.

The algorithm follows these steps:

1. Scale-space peak selection: Keypoints (or interest points) in an image are detected using the difference of Gaussian (DoG) function to identify locations where the DoG is maximized.
2. Keypoint Localization: Detected keypoints are localized to reduce unnescary points by using Taylor series expansion and checking the intensity at the detected point. If the intensity is less than a threshold, the point is discarded.
3. Orientation Assignment: For the purpose of stability, the orientation of the keypoint is assigned by creating a histogram of the gradient directions of the pixels in a region around the keypoint. The highest peak in the histogram is considered as the orientation of the keypoint.
4. Keypoint Descriptior: A descriptor is computed by creating a 16x16 grid around the keypoint which is divided into 16 sub-blocks of 4x4 pixels. The gradient magnitude and orientation of each pixel in the sub-block are used to create a 128-dimensional vector.
5. Keypoint Matching: Once descriptors are computed for all keypoints in the image, the nearest neighbor algorithm is used to match the keypoints between the template and the scene image.

The algorithm is implemented with the following functions:

void detect_and_compute(Mat img, vector<KeyPoint>& kps, Mat& desc)

Description: The function is used to detect keypoints and compute the descriptors for the given image.

Implementation: From the initialized SIFT Ptr<Feature2D> object by SIFT::create(), the function detectAndCompute() is used to detect keypoints and compute the descriptors for the given image with Mat() as the mask.

vector<DMatch> match_descriptors(Mat descriptors1, Mat descriptors2)

Description: The function is used to match the descriptors between the template and the scene image using the FLANNBASED matcher and the KNN match.

Implementation: After initializing the FLANNBASED object with DescriptorMatcher::create(DescriptorMatcher::FLANNBASED), the function knnMatch() is called in order to match the descriptors between the template and the scene image.

The return value is a DMatch vector which contains the index of the best match for each descriptor in the template and the scene image calculated with the condition that the distance of the best match is less than 0.75 times the distance of the second best match.

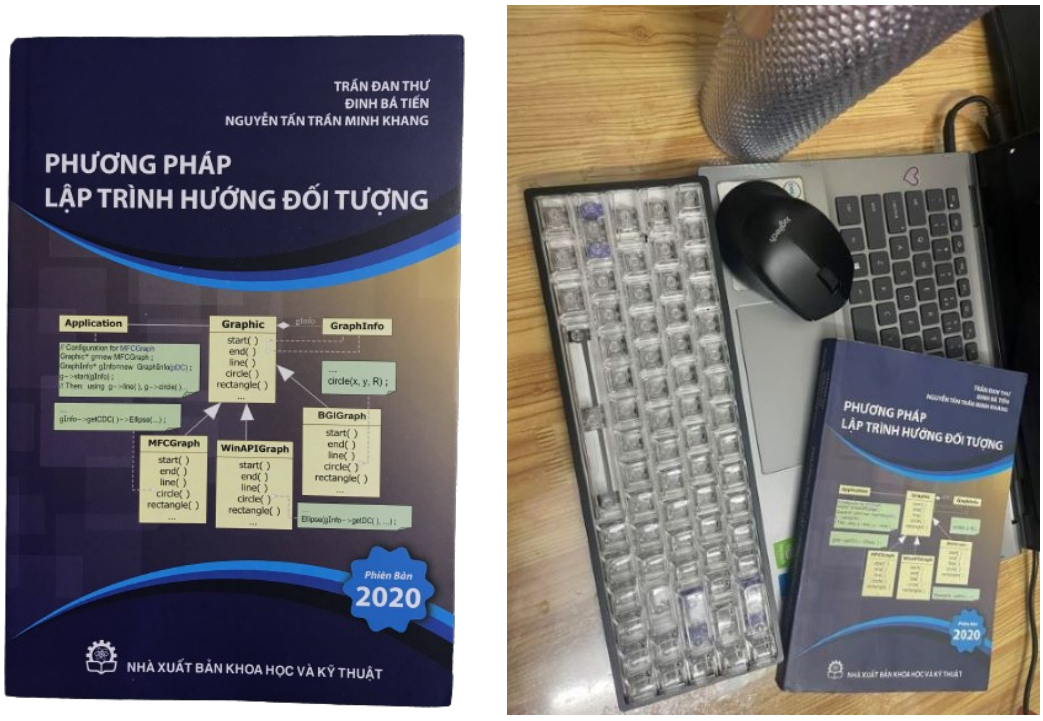
Mat Homography(vector<KeyPoint> keypoints1, vector<KeyPoint> keypoints2, vector <DMatch> good_matches)

Description: The function is used to find the homography matrix between the template and the scene image using the RANSAC algorithm.

Implementation: The 2 vectors of keypoints are initialized by getting the coordination of query index and train index from the good matches among the 2 vectors of keypoints in the template and the scene image.

The function then uses findHomography() is used to find the homography matrix between the 2 keypoints vector with the RANSAC algorithm and the RANSAC threshold is set to 3.

4 Proof images



(a) Template image (b) Scene image

Figure 1: Template and Scene images

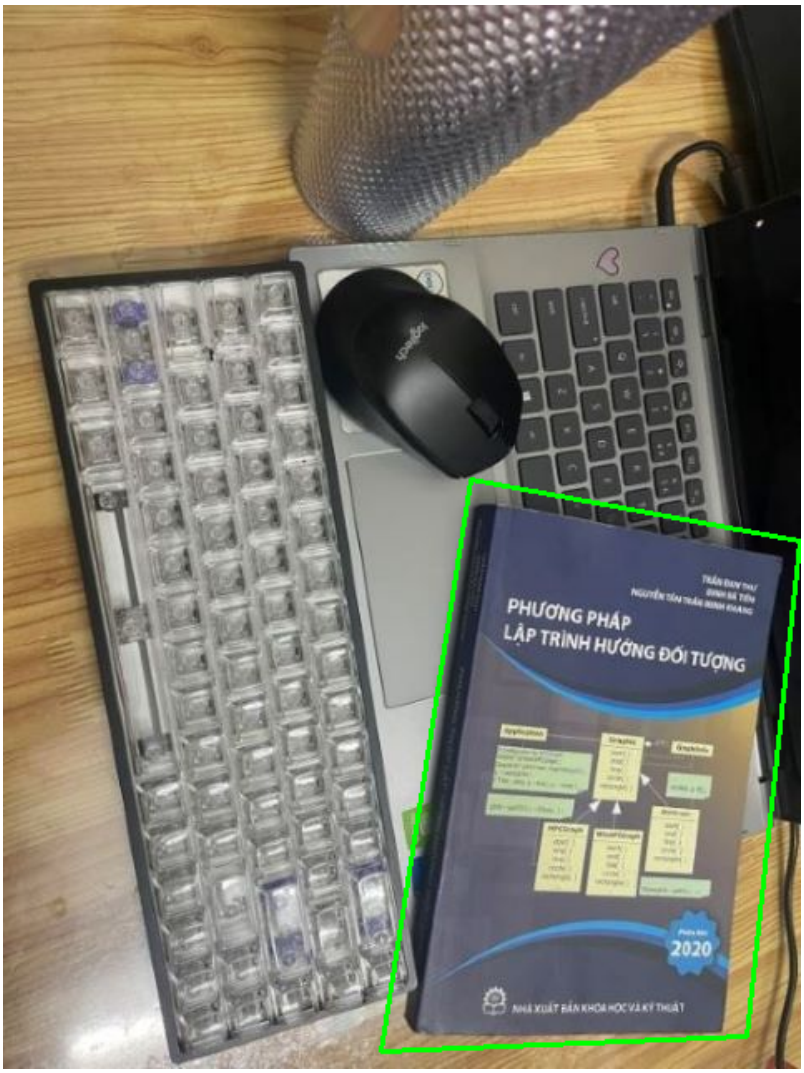


Figure 2: Result image

5 References

Lab02 document on moodle.

SIFT algorithm.

Object detection using SIFT.