Sift Feature Detection

Prepared by
Sergio Torres

Computer Vision
CAP 4410

3/29/20

Florida Polytechnic University

Table of Contents

List of Figures

1. Report

1.1 Read and Load

I first load the image that was included for the assignment and then another image to compare it with. After the images are read, a mask is made to convert the images to CV_8U since I kept getting an error saying that the mask type wasn't CV_8U. I then apply the mask to the images. Finally, I convert the images to grayscale and then show the images.

```
//read image
Mat image1 = imread("blocks_L-150x150.png");
Mat image2 = imread("cat.jpg");

Mat mask1 = Mat::zeros(image1.size(), CV_8U);
Mat mask2 = Mat::zeros(image2.size(), CV_8U);

Mat roi1(mask1, cv::Rect(10, 10, 100, 100));
Mat roi2(mask2, cv::Rect(10, 10, 100, 100));
roi1 = Scalar(255);
roi2 = Scalar(255);

imshow("Image 1", image1);
imshow("Image 2", image2);
waitKey();

//convert image to grayscale
cvtColor(image1, image1, COLOR_BGR2GRAY);
cvtColor(image2, image2, COLOR_BGR2GRAY);


imshow("Image 1", image1);
imshow("Image 2", image2);
waitKey();
```

1. Load Image

1.2 Feature Extraction

I first create a Sift Feature Detector and then two vector key points. I use the detect and compute functions in order to get the features on the images. After that I draw the key points of both images and then show them.

```
//create sift and vector keypoint
Ptr<xfeatures2d::SiftFeatureDetector> detector = xfeatures2d::SiftFeatureDetector::create();
std::vector<cv::KeyPoint> keypoint1;
std::vector<cv::KeyPoint> keypoint2;
printf("Vector created");
waitKey();

//detect keypoints
Mat descriptor1, descriptor2;
detector->detectAndCompute(image1, NULL, keypoint1, roi1);
detector->detectAndCompute(image2, NULL, keypoint2, roi2);
printf("detected");
waitKey();



//draw keypoints
//drawKeypoints(image1, keypoint1, descriptor1, Scalar::all(-1), DrawMatchesFlags::DRAW_RICH_KEYPOINTS);
//drawKeypoints(image2, keypoint2, descriptor2, Scalar::all(-1), DrawMatchesFlags::DRAW_RICH_KEYPOINTS);
imshow("SIFT Image 1 KeyPoints", descriptor1);
imshow("SIFT Image 2 KeyPoints", descriptor2);
printf("keypoint");
waitKey();
```

2. Feature Extraction

1.3 Feature Description

I first create a Brute Force Matcher in order to see the matches for both pictures. I then create the vector for the brute force matcher and then do a knn Matcher. I look through the pictures in order to see if there are any good matches between them and push them onto the gMatches vector. After that I draw the matches of both the images and print them out.

```
//create brute force matcher
BFMatcher bf = BFMatcher();

//create vector matches for bf matcher
vector<vector<DMatch>> matches;
bf.knnMatch(descriptor1, descriptor2, matches, 2);

vector<DMatch> gMatches;
for (size_t i = 0; i < matches.size(); i++)
{
	if (matches[i][0].distance < 0.7 * matches[i][1].distance)
	{
		gMatches.push_back(matches[i][0]);
	}
}

//draw matches
Mat imMatches;
drawMatches(descriptor1, keypoint1, descriptor2, keypoint2, gMatches, imMatches, Scalar::all(-1), Scalar::all(-1), vector<char>(), DrawMatchesFlags::DRAW_OVER_OUTIMG);
imshow("Matches", imMatches);
waitKey();
return 0;
```

3. Feature Description

1.4 Product Scope

This assignment requires me to implement SIFT method of feature extraction and feature description for some simple input images. You are provided a set of experimental data. Do your own research to achieve final set of features and find a way to handle those data.

1.5 Challenges

I couldn't get the program to show any of the feature description, extraction and then the matches. I couldn't get past errors I was having where the vector iterator insert was out of range and then where the mask type wasn't correct, so I wasn't able to test anything out.

References

[1] Introduction to SIFT (Scale-Invariant Feature Transform) (n.d.). Retrieved from https://docs.opencv.org/trunk/da/df5/tutorial_py_sift_intro.html

[2] How to use SIFT in opencv. Retrieved from https://stackoverflow.com/questions/22722772/how-to-use-sift-in-opencv

[3] Implementing SIFT in OpenCV. Retrieved from https://aishack.in/tutorials/implementing-sift-opencv/