Local Feature Matching

Algorithm Description

Harris Corner Detector

The Harris Corner Detector follows a standard format for detecting points of interest. It converts the image to grayscale, before applying a Gaussian blur with a kernel size given by the user. The default value of 5 was found to give reasonable results.

The image is then filtered with the standard Sobel matrices. Then the Harris criteria are calculated using the gradient covariances.

Non-Maximum Suppression

The non-maximum suppression is done via the algorithm outlined in the given paper from Microsoft. First, a threshold is applied to our cornerness matrix. The points are then sorted in descending order of cornerness. We then find the suppression radii of the keypoints by using the following formula:

$$r_i = \min_j |\boldsymbol{x}_i - \boldsymbol{x}_j|, \;\; s.t. \;\; f(\boldsymbol{x}_j), \;\; \boldsymbol{x}_j \in \mathcal{I}$$

We then take corners with the largest suppression radii as our corners.

SIFT Features

We apply a Sobel filter to our image, followed by creating an array of angles and magnitudes which we use later on. We iterate through our keypoints and create a histogram of their neighbourhood as defined by the feature size. To create this, we use the angles and magnitudes that are relevant to that neighbourhood. We then add this to our feature vector.

Lowe's Ratio Test

We calculate the distances between the keypoints of each image. For each keypoint we keep the 2 nearest matches in the other image. We then check if the distances are significantly different, if they are not we eliminate the keypoint from consideration.

Adjustable Parameters

- **feature_width** : Size of features
- **R_threshold** : The threshold for considering an interest point in R

- **bins** : The number of bins when creating a histogram in SIFT
- **k_size** : the kernel size when applying our blur
- **num_corners** : the number of corners we are considering
- **threshold** : the ratio threshold for considering matches

Testing

We will vary the following parameters:

- feature_width
- bins
- k_size
- num_corners
- threshold

We do not change the R_threshold as if we reach maximum corners then the threshold does not affect the results.

Results

All results have the parameter set written as (feature_width, bins, k_size, num_corners, threshold).

Notre Dame

Parameter Set	Accuracy	Number of Matches
(16, 8, 5, 2000, 0.75)	0.05	9
(16, 8, 5, 4000, 0.85)	0.37	86
(16, 16, 5, 4000, 0.85)	0.23	51
(16, 8, 9, 4000, 0.85)	0.32	95
(16, 8, 17, 4000, 0.85)	0.42	114
(32, 8, 17, 4000, 0.85)	0.66	91

Mount Rushmore

Parameter Set	Accuracy	Number of Matches
(32, 8, 17, 4000, 0.85)	0.08	21
(16, 8, 5, 4000, 0.85)	0.02	30
(64, 8, 5, 4000, 0.85)	0.11	19
(64, 8, 5, 8000, 0.85)	0.15	28
(64, 8, 17, 8000, 0.85)	0.49	49
(64, 8, 17, 8000, 0.90)	0.64	126

Episcopal Gaudi

Parameter Set	Accuracy	Number of Matches
(64, 8, 17, 8000, 0.90)	0.12	68

(64, 8, 17, 16000, 0.90)	0	135

In general, it was found that increasing the number of corners was helpful to a point. This is to be expected as with more relevant points to work from, we can increase our accuracy. But there is a limit where superfluous points begin to be added and this breaks down.

Increasing the kernel size also seemed to help with matching features. This is because it negated some of the problems with erroneous matchings where keypoints had more in common than anticipated.

Increasing the feature size also appeared to help. Mount Rushmore especially, with all its stone surfaces, benefited from large features and an increased threshold when measuring for matchings.

Unfortunately, the results for Episcopal Gaudi were below what was hoped for. Upon inspection, the two images are to different scales. Scale Invariance was not implemented as part of this coursework, thus the results.

Better results could likely be obtained from all of the images, with enough tuning of the parameters. The results above serve to show the “highlights” of the testing, as many tests were run to try and obtain better accuracy. But, unfortunately, none were found.