Machine Learning for Computer Vision: Coursework 2 - Image Matching

Tormento, Marion
Imperial College London
marion.tormento17@imperial.ac.uk

McLaughlin, Edward
Imperial College London
edward.mclaughlin17@imperial.ac.uk

Abstract

The functions created are available at https://github.com/MarionTormento/MLCV for your information.

1 Matching

In this section, techniques for finding and matching corresponding interest points in two different images are explored. Firstly, the interest points are found (either manually or automatically). Subsequently, descriptors of these interest points are constructed and compared to determine the corresponding points. From these pairs of matching points the homography and fundamental matrices and their accuracies are calculated for the transformation from one image to the next.

1.1 Finding points manually

Interest points were found manually by clicking the cursor on the interest points in sequence. This was repeated for the second image. These two sets of interest points were matched using the techniques described later in this section.

1.2 Finding points automatically

In order to find interest points automatically, several corner detectors were implemented and their results compared. The performance of each algorithm was tested against the same toolbox corner detectors available on OpenCV. As SURF and SIFT corner detectors are not freely available on OpenCV, they were not implemented. Once relevant interest points had been located, several descriptors were used to characterise the corner points. The performance of each descriptor was assessed on its ability to correctly match corresponding interest points using a nearest neighbour algorithm.

1.2.1 Interest points and descriptors:

Initially, Harris [1] and Shi-Tomasi [3] interest point detectors were implemented. These detectors differ only in their evaluation of R - the determination of whether a given point is considered a corner. The input parameters for these detectors are detailled in appendix 3. In both cases, the threshold of the R value was set to the 99-th percentile value: this way, enough interest points were obtained in a reasonable computational time.

Subsequently, a Features from Accelerated Segment Test (FAST) [4] feature detector was implemented. The input parameters for the FAST interest point detector can be found in appendix 3. In addition a buffer zone at the sides of the image was created. All interest points within this buffer zone were deleted. This was to ensure the window size of the descriptor did not exceed the boundaries of the image. It is assumed the a sufficient number of interest points are detected outside of this buffer region such that the buffer would not be detrimental to the performance of the algorithm.

The corner points detected from these three detectors is shown in Figure 1. The comparison of the corner pointed found by the detectors implemented by the authors and the toolboxes available on OpenCV is shown in Figure 2.

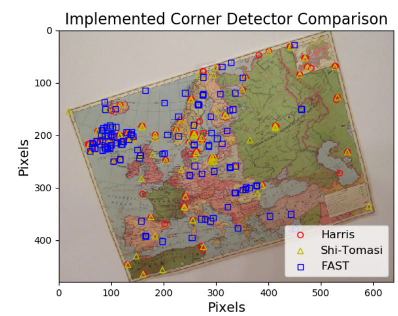
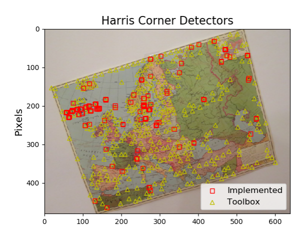
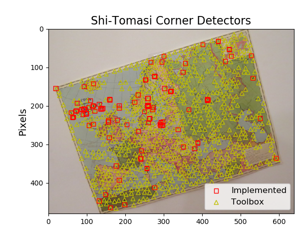


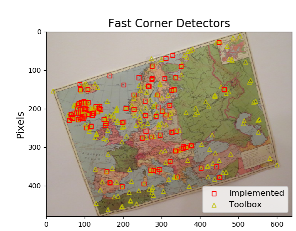
Figure 1. Comparison of Corner Points detected by implemented solutions. Harris Implemented - 138, Shi-Tomasi - 121, FAST Implemented - 125



(a) Harris: Implemented - 138, Toolbox - 380



(b) Shi-Tomasi: Implemented - 121, Toolbox - 768



(c) FAST: Implemented - 125, Toolbox - 297

Figure 2. Comparison of Corner Points detected by implemented solutions and openCV algorithm.

Three types of descriptor were implemented for this coursework : simple coulour detector (RGB), histogram of oriented gradients

(HoG) and RGBoG. While the first two are classic techniques, the third was designed for this coursework: for each color an histogram of gradient is performed. The parameters used for each are detailled in appendix 3.

Comparison of descriptor:

- pros and cons of each
- in what case do they work better

Matching interest points across images

To match interest points, a simple nearest neighbour search algorithm was implemented: for each features of the test image, the distance between its descriptor and all the descriptors of the base image features is computed, and only the index returning the minimal distance is stored. To maximize the chances of returning true positive matches, three simple strategies have been implemented:

- the best matches are returned first - the quality of a match depending on its distance
- the ratio technique, first presented by D. Lowe [2], was implemented: in addition to the nearest neighbour, the second nearest neighbour is searched, and the ratio of the two distances is computed. Unfortunately, this technique did not returned interesting results, and was latter deleted from the code.

Additionally, to avoid clustering of interest point which negatively affects the matching performance, only the best match of each cluster is saved as an interest points - the maximum size of the cluster being set to one twentieth of the minimum between the width and height of the image. RANSAC was later implemented to eliminate outliers and will be presented in section 2. When testing the descriptor is was found that RGB are the best suited when picture are rotated, and HOH are best suited when the picture are zoomed in or shifted.

Figure 3. Matching interesting points across two images, TP : FP :

Transformation estimation

Homography matrix, projection and accuracy

few lines on the method used to compute Homography

Fundamental matrix, epipolar lines and accuracy

few lines on the method used to compute Homography

2 Image Geometry

Homography

Part A

An image of the rotated map set was used for this question. The detector used is the implemented Harris Corner detector and it is compared to the openCV version. All types of descriptor are tested. The homography are each time computed between the original image and one of the reduced size. The homography accuracy is computed both in pixel and in percentage by diving the absolute distance in pixel by the minimum of the dimension of the reduced image - ie, for the implemented Detector, RGB descriptor, and reduction factor of 2, in average the distance between the real interest points and the estimated one is 56Px, which represent 23% of the image.

In order to get consistent results, even though the size of the image was radically decreasing, a dynamic window size, depending on the size of the image, have been implemented for the descriptor. In addition every descriptor value have been normalized. Nevertheless the homography is not correct with an average percentage error of 23%, resp. 20% for the openCV, for a reduction factor of 2, and 31%, resp. 22%, for a reduction factor of 3. The smaller the image, the harder it is to get relevant interest points as the information is compressed in less pixels.

Table 1. Homography Accuracy in Pixel and Percentage between Original Image and its reduction by a factor of 2 (RF2) or 3 (RF3)

		RF2		**RF3**	
Detector	**Descriptor**	**Px**	%	**Px**	%
Implemented	RGB	56	23	42	26
	HOG	59	25	58	36
	RGBHOG	36	15	43	27
OpenCV	RGB	27	11	24	15
	HOG	55	23	38	24
	RGBHOG	72	30	33	21

Part B

The homography matrix is computed for manually and automatically picked interest points. When enough interest points are carefully manually picked (25 or more), the value returned is closed to the automatic one, with a euclidian distance of 12.0 between the two matrices. Figure 4 presents a comparison between the location of the real interest points (in blue and green - which are the one used to compute the homography matrix) and the estimated one (in red in the top right image). The bottom images are the projection of each image in the other image plan.
From the two sets of matching points, the geometric transformation parameters could be estimated. The real transformation was estimated from image measures on Photoshop.

$$\vec{T_{manual}} = \begin{pmatrix}29 & -50 & 0\end{pmatrix}^T, R_{manual} = 12.7^o \tag{1}$$

$$\vec{T_{auto}} = \begin{pmatrix}36 & -42 & 0\end{pmatrix}^T, R_{auto} = 10.1^o \tag{2}$$

$$\vec{T_{real}} = \begin{pmatrix}19 & 20 & 0\end{pmatrix}^T, R_{real} = 11^o \tag{3}$$

Part C

Table of homography depending on the number of points Show a graph of false positive compared to true positives depending on the number of points?

Stereo Vision

Part A

Use manual and automatic feature detection (just use one OF OUR OWN from the ones presented in section) and compare and discuss the fundamental matrices and their accuracies.

Part B

Calculate the epipolar points and the epipolar lines and in images A and B. Display the disparity map (for our own images AND IN APPENDIX from sample image sets). Describe the two ways in which the disparity maps could be calculated (Ixy^2 and intensity)

Part C

Depth map is 1/disparity

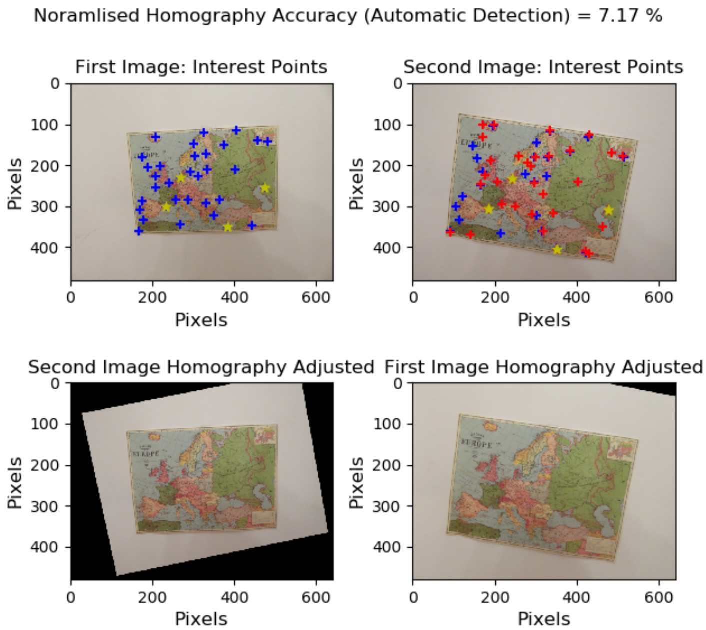


Figure 4. Interest points: used (yellow ⋆) and not used (blue +) for homography calculation. The points from image A projected by the homography matrix to image B are shown red +. Bottom two images are the images transformed to the plane of their respective counterpart.

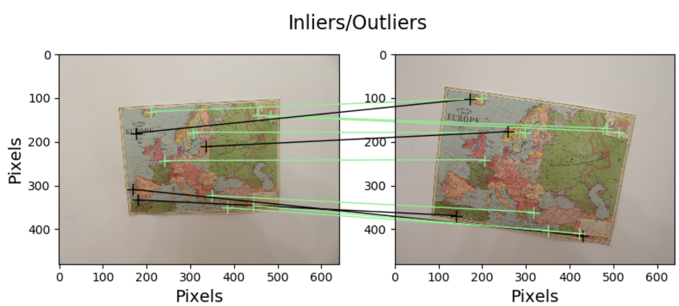


Figure 5. A subset of the matched points having had the homography applied to them. Outliers (black) are defined as correspondences with greater than 5 pixels distance between the actual and projected points.

Part D

Change the focal length and add a small gaussian noise (max 2 pixel). Repeat disparity map. (and its corresponding depth map?)

Part E

STEREO RECTIFIED IMAGE YAY.

References

[1] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Alvey vision conference*, pages 147–151, 1988.

[2] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[3] C. Tomasi and J. Shi. Good Features. *Image (Rochester, N.Y.)*, pages 593–600, 1994.

[4] M. Trajkovii, M. Hedley, M. Trajkovic, and M. Hedley. Fast corner detection. *Image and Vision Computing*, 16(2):75–87, 1998.

3 Appendix

Table 2. Number of interest points detected by each detector.

	Harris		Shi-Tomasi		FAST	
	# Corner Points Detected					
Image (size)	**Implemented**	**ToolBox**	**Implemented**	**ToolBox**	**Implemented**	**ToolBox**
Map 1 (640x840)	129	164	136	407	55	112
Map 2 (640x840)	138	380	121	768	125	297
Computer Room (640x840)	113	94	106	93	187	364
Living Room (640x840)	78	87	94	183	62	131
Tsukuba (384x288)	51	47	52	87	192	353
Art Work (1390x1110)	720	458	672	985	278	969
	# Corner Points Detected per 10,000 pixels					
Image (size)	**Implemented**	**ToolBox**	**Implemented**	**ToolBox**	**Implemented**	**ToolBox**
Map 1 (640x840)	2.4	3.1	2.5	7.6	1.0	2.1
Map 2 (640x840)	2.6	7.1	2.3	14.3	2.3	5.5
Computer Room (640x840)	2.1	1.7	2.0	1.7	3.5	6.8
Living Room (640x840)	1.5	1.6	1.7	3.4	1.2	2.4
Tsukuba (384x288)	4.6	4.2	4.7	7.9	17.4	31.9
Art Work (1390x1110)	4.7	3.0	4.4	6.4	1.8	6.3
Mean	3.0	3.5	2.9	6.9	4.5	9.2
Standard Deviation	1.2	1.8	1.2	4.0	5.8	10.3

ADD PARAMETERS DESCRIPTORS

Parameters

- σ_I: automatically vary with the size of the image;
- σ_D: set to $1.6\sigma_I$ according to the lecture slides;
- α: set to 0.04 - should be between 0.04 and 0.06 according to online documentation REFERENCE;
- $maxima_{NN}$: number of nearest neighbours when performing non maxima suppression, set to 50 to reduce clustering;
- $maxima_{perc}$: the percentile threshold for the corner point's R value used to qualify corner points for maxima suppression.
- radius: chosen as 3 which optimised the performance and computation time;
- threshold: found empirically for each image as it depends on the distribution of intensities in the image;
- S: varied between 7 and 10 depending on the image.