Practical Session: Multiview Analysis

Object Recognition

University of Barcelona

April 19, 2016

Authors:
Camps, Julià

Contents

1 Introduction

In this practical exercise we will deal with experimenting with several of the characteristics of multiview analysis on pairs of images.

The aim for this exercise is to implement implement and test a method for computing the projective transformation that maps one image to an other from the same scene but from a different point of view. This method will be launched over the same set of images by using manual and automatic (using SIFTLowe [1]) keypoints correlations.

The structure of this report is:

1 Description of the method for performing the projective transformation given two images.

2 Discussion on some topics required by the exercise description.

3 Description of the method used for the manual keypoints correlation.

4 Description of the method used for the automatic keypoints correlation.

5 Tests of the methods over the selected set of images.

6 Discussion on the results.

2 The method for performing the projective transformation

The projective transformation is given by

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \underbrace{\begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{pmatrix}}_{\mathbf{H}} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{1}$$

having the homogeneous coordinates $(x, y) = (x_1/x_3, x_2/x_3)$ representation of the points in an image. So that the coordinates named x' compose the points of the transformed image, while the ones named x are form the original one.

After performing some mathematical transformations (*see the lab assignment description for further details on the maths*) we derive the following equation:

$$\begin{pmatrix} 0 & 0 & 0 & -x'_3x_1 & -x'_3x_3 & x'_2x_1 & x'_2x_2 & & \\ x'_3x_1 & x'_3x_2 & x'_3x_2 & x'_3x_3 & 0 & 0 & 0 & -x'_1x_1 & -x'_1x_2 \end{pmatrix} = \begin{pmatrix} -x'_2x_3 \\ x'_1x_3 \end{pmatrix} \tag{2}$$

Notice that in (2) is composed by the 8 first elements of H in (2), since in order to obtain the (2) we had to perform an assumption on the element h_{33} of (2), stating that $h_{33} = 1$.

Pleas notice that in variable the elements of H are sorted in a row-wise notion, rather than a column-wise order.

By means of implementing the (2) algorithm in Matlab we were able to by having 4 point correspondences between two images, build a system of equations and solve all the unknown variables obtaining the from which could be easily deduced H (just by adding the $h_{33} = 1$ removed element and reformatting to the original rows-columns shape of the matrix.

Once we have the H matrix, the rest of the procedure is just computing for every point of the original image, which is its projection in the transformed on (i.e. $x' = \mathbf{H}x$). However, we are recommended to use the inverse transformation, $x = \mathbf{H}^{-1}x'$. This is due to some pixels

of the original image could correspond to more than one pixel in the transformed one, and then if we do the inverse transformation we ensure that we obtain all the information needed for rebuilding the full image transformed, without loosing any part that could be provided by the transformation procedure. It may also be the case that some pixels of the original image have no projection on the transformed one. From this two facts we can deduce that using the inverse transformation is ensuring to obtain good results, while can't be stated if we used the direct transformation.

3 Correspondence methodologies

In this section we will review the methodologies implemented for the correspondence procedure.

3.1 Manual correspondence

For the manual correspondence we have followed the guiding instructions provided with the description of the exercise, which were:

1 Select 4 points significant in the original image.

2 Select the projected points of the original image into the transformed image.

3 Associate the points from both images in order to have pairs of [original/transformed] points. *Note that for the points extraction from the images we have to swap the X's with the Y's, since for images in Matlab the axes are inverted.*

3.2 Automatic correspondence

For the manual correspondence we have followed the guiding instructions provided with the description of the exercise, which were:

1 Extract the SIFT descriptors of both images.

2 Match each descriptor with the closest one on the other image using the L2 norm of the difference among them.

3 Using information from the L2 norm select a subset of best keypoints matching pairs candidates for computing the **H** transformation matrix.

4 At this point we should find best matching pairs of keypoints and that give the most representative changes for computing the projective transformation matrix. However, we don't know which are this "best" matching points, in fact we don't even know if given a pair of keypoints if they really are one the projection of the other, or if they are very similar, but from different parts of the image.
Taquing these facts into account, we decided implement two possibilities:

- Using the matches with lower differences among themselves: the motivation for choosing this subset of keypoints is that since they are really similar, its less provable that they have been wrongly matched. So we can ensure in a more likely notion to be working among true labelled data. However, they won't be able to represent the view transformation in a proper way, since that part of the image will be almost the same in order to be giving such low L2 norm values. We should also consider that they are more likely to come from the same section of the image.
- On the other hand, if we take the ones with higher L2 norm values, they are the points more likely to be wrongly matched from the whole image.

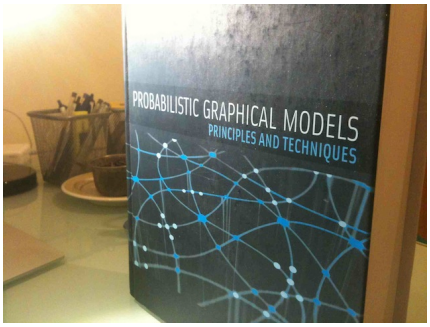
Note that for the points extraction from the images we have to swap the X's with the Y's, since for images in Matlab the axes are inverted.

4 Testing the implemented methods

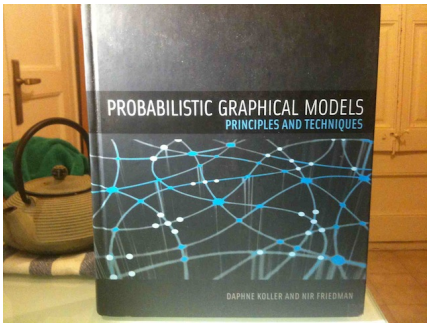
For the experiments, we will use the set of images provided with the description of the exercise.

4.1 Test for the image: *Llibre*

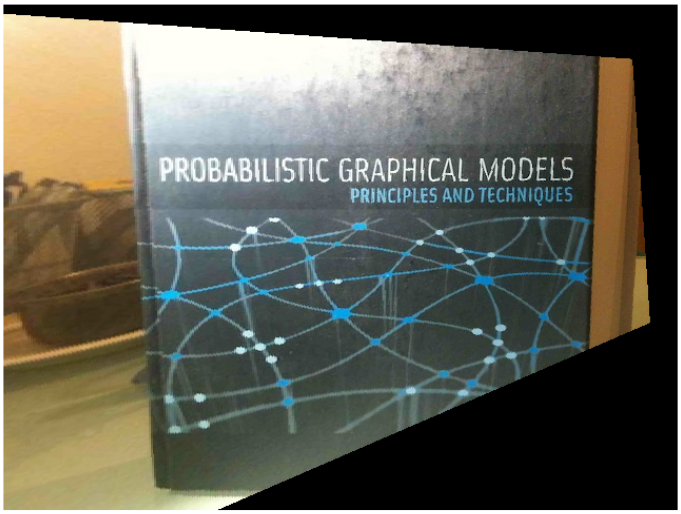
In this section we can see the results for the tests performed with the *Llibre1.jpg*1a image as the original image and the *Llibre2.jpg*1b image as its objective projection.



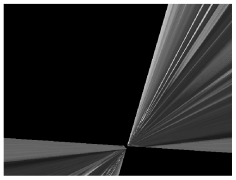
(a) Original image.



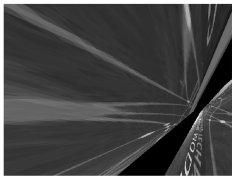
(b) Target transformation image.



(c) Transformation using the manual keypoints correspondence.



(d) Transformation using the automatic SIFT descriptors correspondence (from matches with lower scores).



(e) Transformation using the automatic SIFT descriptors correspondence (from matches with higher scores).

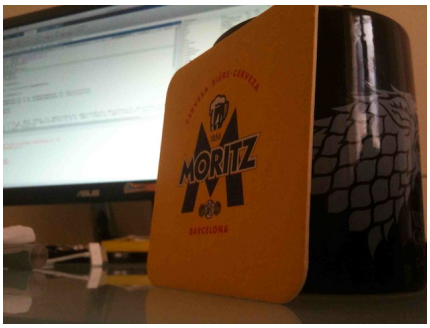
Figure 1: Here we can appreciate the result of applying the method when using manually set keypoints correlations vs automatically detected keypoints correspondences, by means of using SIFT descriptors.

In figure 1 we can observe that the implemented method is working properly and achieves its objective to compute a matrix $\mathbf{H}$, which is clearly encoding the mapping relation between an given original image and its transformation, from a set of 4 given keypoints correspondences. However, observe that the automatic correspondence between SIFT descriptors is facing some problems, and it's not able to return such an accurate result as when performing the manual correspondences.

4.2 Test for the image: *moritz*

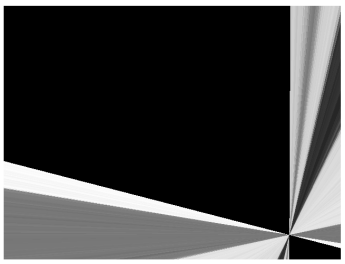
In this section we can see the results for the tests performed with the *moritz1.jpg*2a image as the original image and the *moritz2.jpg*2b image as its objective projection.

(a) Original image.

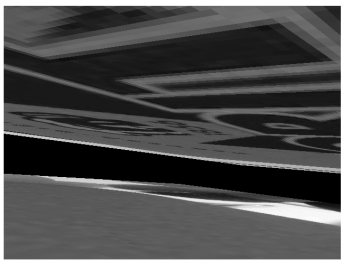


(b) Target transformation image.

(c) Transformation using the manual keypoints correspondence.



(d) Transformation using the automatic SIFT descriptors correspondence (from matches with lower scores).



(e) Transformation using the automatic SIFT descriptors correspondence (from matches with higher scores).

Figure 2: Here we can appreciate the result of applying the method when using manually set keypoints correlations vs automatically detected keypoints correspondences, by means of using SIFT descriptors.

In figure 2 we can observe that the implemented method is working properly and achieves its objective to compute a matrix **H**, which is clearly encoding the mapping relation between an given original image and its transformation, from a set of 4 given keypoints correspondences. However, observe that the automatic correspondence between SIFT descriptors is facing some problems, and it's not able to return such an accurate result as when performing the manual correspondences.

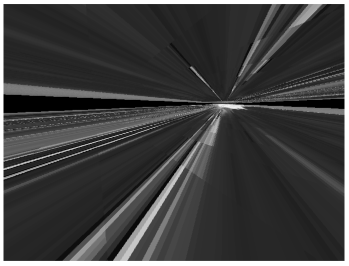
4.3 Test for the image: *picture*

In this section we can see the results for the tests performed with the *picture1.jpg*3a image as the original image and the *picture2.jpg*3b image as its objective projection.

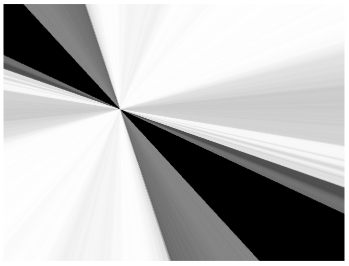
(a) Original image.

(b) Target transformation image.

(c) Transformation using the manual keypoints correspondence.



(d) Transformation using the automatic SIFT descriptors correspondence (from matches with lower scores).



(e) Transformation using the automatic SIFT descriptors correspondence (from matches with higher scores).

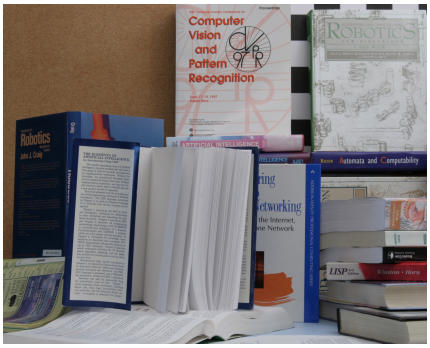
Figure 3: Here we can appreciate the result of applying the method when using manually set keypoints correlations vs automatically detected keypoints correspondences, by means of using SIFT descriptors.

In figure 3 we can observe that the implemented method is working properly and achieves its objective to compute a matrix **H**, which is clearly encoding the mapping relation between an given original image and its transformation, from a set of 4 given keypoints correspondences. However, observe that the automatic correspondence between SIFT descriptors is

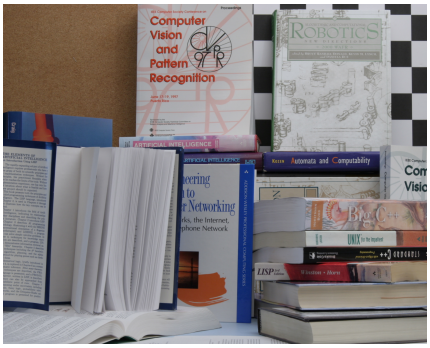
facing some problems, and it's not able to return such an accurate result as when performing the manual correspondences.

4.4 Test for the image: *books*

In this section we can see the results for the tests performed with the *books1.png*4a image as the original image and the *books2.png*4b image as its objective projection.

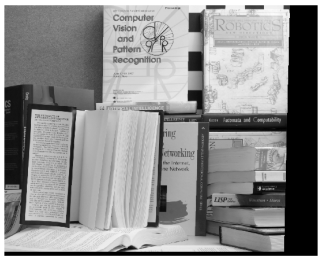


(a) Original image.



(b) Target transformation image.

(c) Transformation using the manual keypoints correspondence.



(d) Transformation using the automatic SIFT descriptors correspondence (from matches with lower scores).

(e) Transformation using the automatic SIFT descriptors correspondence (from matches with higher scores).

Figure 4: Here we can appreciate the result of applying the method when using manually set keypoints correlations vs automatically detected keypoints correspondences, by means of using SIFT descriptors.

In figure 4 we can observe that the implemented method is working properly and achieves its objective to compute a matrix **H**, which is clearly encoding the mapping relation between

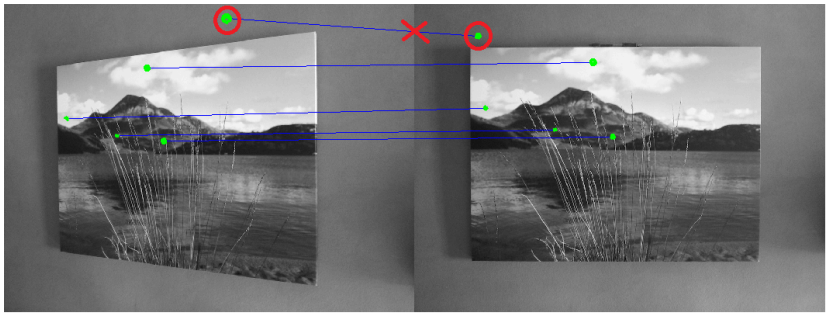
an given original image and its transformation, from a set of 4 given keypoints correspondences. However, observe that the automatic correspondence between SIFT descriptors is facing some problems, and it's not able to return such an accurate result as when performing the manual correspondences.

4.5 Discussion on the results

From the experiments we can conclude that the method for performing the projective transformation using the calculated matrix **H** is working properly, since that when we use manual correspondence retrieves a correct result for the projective transformation.
Although, as we already suspected from the information given by the lecturer, the method showed very good results as far as, the main transformed object of the scene was "flat" (e.g. a book, a picture, a screen, etc.), since with no such flat objects (i.e. complex 3D shapes) such as scenes with many objects, the transformation did not perform that well. This can be seen in test4.4.

However, it can be appreciated that the automatic correspondence using L2 norm on the difference of the found SIFT descriptors in both images is not performing perfectly, although some times it does as can be observed in 4d, most of times we are facing incorrect results when applying it. This is due to the following facts:

- If we choose the keypoints by means of selecting the matches with smaller score values, since the score value represents the L2 norm between both descriptors, if the L2 norm is very small means that the difference is very small, and though the points are not representing the transformation in a highly expressive way. Moreover, I would like to point out, that this points perhaps may not be the best candidates for computing the projective transformation matrix, but surely are crucial for deducing that two images are different views from the same scene, which could not be so well deduced using the matches with higher L2 norm, since they are expressing differences rather than resemblances.
- On the other hand, if we use the keypoints corresponding to the matches with higher score, means that the L2 norm is very high, so the differences are being maximised (which could fix our previous problem). However, since the differences are so high, we might find most of times that the differences are so high because the matching process failed and one point does not correspond to the other. From this we can deduce that the transformation matrix **H** will be computed from wrong assumptions and, though, will return wrong projection results.
- In order to demonstrate that the problems faced in the automatic approach were due to having incorrect matches, when using the matches of maximum score values (the ones that explain better the transformation, since their projection differences are maximised). I selected, instead of the first 4 matches retrieved by the algorithm, the first 4 ones checked by myself to be correct matches, by checking on the matches correspondence plot. As it can be seen in figure 5, which corresponds to experiment 3, from the results reviewed, the algorithm is **H** is calculated properly if the matches are correct and enough significant, for explaining the transformation.



(a) Matches selected, corresponding to $indexes = [1, 2, 4, 5]$, instead of $indexes = [1, 2, 3, 4]$, when sorted from highest to lowest scores. Since the match corresponding to $index = 3$ was wrong.

(b) Transformation using the automatic SIFT descriptors correspondence, manually checked to be correctly matched.

Figure 5: Here we can appreciate the result of applying the method when using automatic "checked manually" set keypoints correlations.

- For giving more support to the deductions extracted from the analysis of the results when using the automatic approach. The correspondences of the points that where being used in each case for computing the **H** matrix, have been added to the A section at the end of the report. There, in figure 6, it can be clearly seen all the facts that have been described during this section.

5 Conclusions and Future work

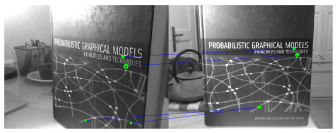
From this exercise we can conclude that the multiview analysis is of highly importance, when referring to the object detection problem, since it can not be expected to have different object classes for the same object from different points of view if the information is easily extractable just by transforming the image.

Although, we haven't studied yet any algorithm that can deal with this problem in a straightforward notion. The reviewed techniques in this exercise showed very good results, it's just the automatic correspondence missing some details for differing among well matched descriptors from wrong ones. I'm 100% sure that the solution to this problem is already well known, since most of the matches are performed correctly, there is sure some relation among them that the wrong matches do not follow, and from voting can easily be discarded.

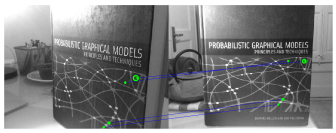
References

[1] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110. ISSN 1573-1405. doi: 10.1023/B:VISI.0000029664.99615.94. URL `http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94`.

A Matches being selected in the automatic approach

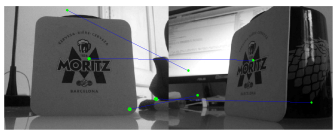


(a) Matches selected for 1d

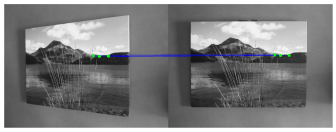


(b) Matches selected for 1e

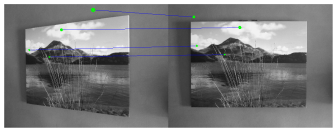
(c) Matches selected for 2d



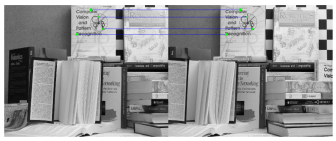
(d) Matches selected for 2e



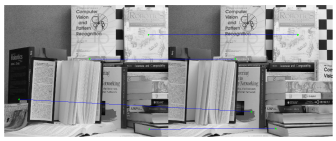
(e) Matches selected for 3d



(f) Matches selected for 3e



(g) Matches selected for 4d



(h) Matches selected for 5b

Figure 6: Here we can appreciate the matches being selected in every automatic test performed, in order to better understand the results achieved.