Computer Vision: Lab
Playing with homographies

Francesca Canale 4113133
Filippo Gandolfi 4112879
Marco Giordano 4034043

23/05/2019

1 Introduction

The goal of this lab is playing with homographies to address some typical problems. We approach three different problems:

- Correcting the perspective distortion of an image.
- Estimating the homography between a pair of images.
- Building a mosaic from a sequence of images.

2 Correcting the perspective distortion of an image.

In this section we estimate the homography that maps a real distorted shape into virtual-synthetic non-distorted one; to do this we select in the image four real points and then another four points that represent the corresponding virtual one.

Figure 1: Point selection

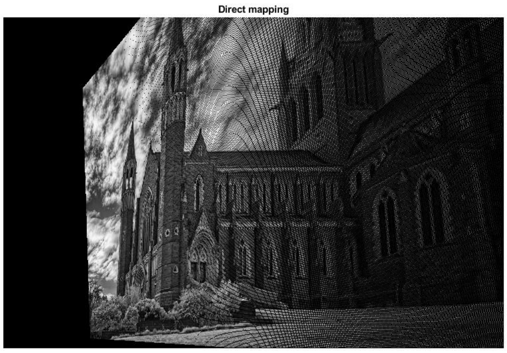

Figure 2: Direct mapping

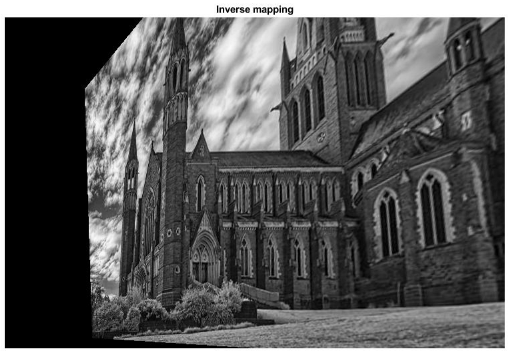

Figure 3: Inverse mapping

In the direct mapping we can see a lot of black lines; this is due to the fact that it is a direct transformation which means that for each pixel in the original image we find a corresponding pixel in the transformed one, so there are some pixels in the transformed image that will have no value. The density of the black lines is higher in the areas where the pixels are shifted more. Instead in the inverse mapping each point has a correspondence because we start computing the result from the pixels of the transformed image.

2.1 Bilinear interpolation

In the code relative to the inverse mapping there is a bilinear interpolation that makes the images more uniform making its pixels have a colour more similar to the near ones. Indeed if we discard the interpolation the resulting image is more grainy as we can see in the picture below.

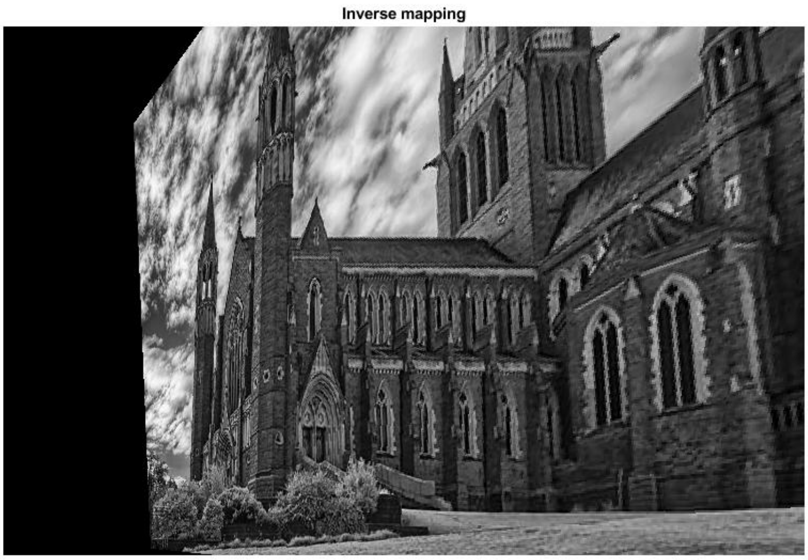

Figure 4: Inverse mapping without bilinear interpolation

3 Estimating the homography between a pair of images

Our aim is to estimate the homography H between a pair of images, again manually selecting the corresponding points in different stereo images. The code estimates the homography matrix H and tries to modify the first image to make it equal to the second one, through an inverse mapping (same inverse mapping function as in the previous case).

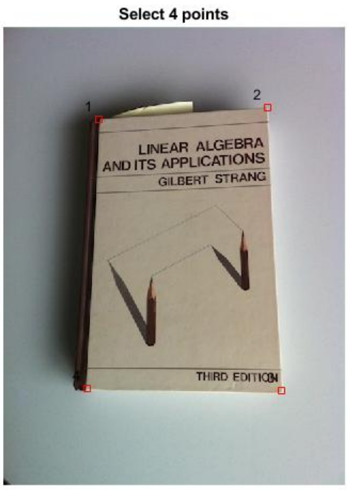

Figure 5: Point selection number one

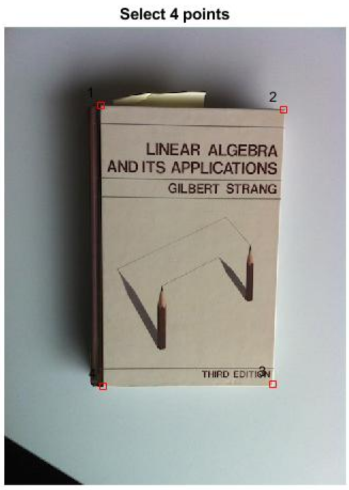

Figure 6: Point selection number two

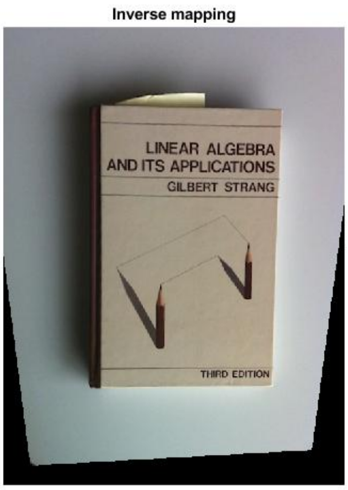

Figure 7: Inverse mapping

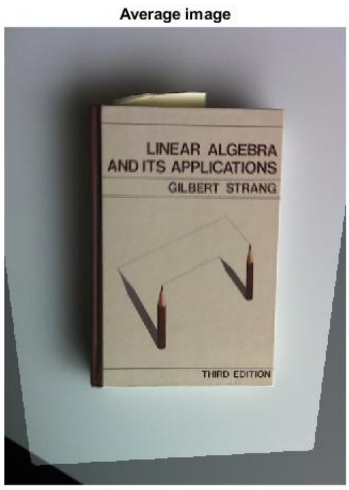

Figure 8: Average image

We tried the algorithm for different image pairs and we noticed that the homography fails in mapping one image into the other when the scene is not planar, as we can see in the picture below.

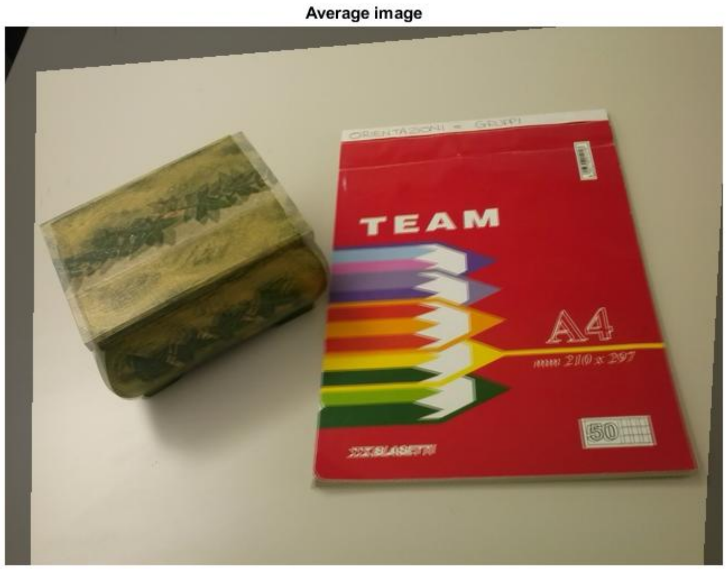

Figure 9: Average image

The result doesn't get any better if we use the RANSAC method.

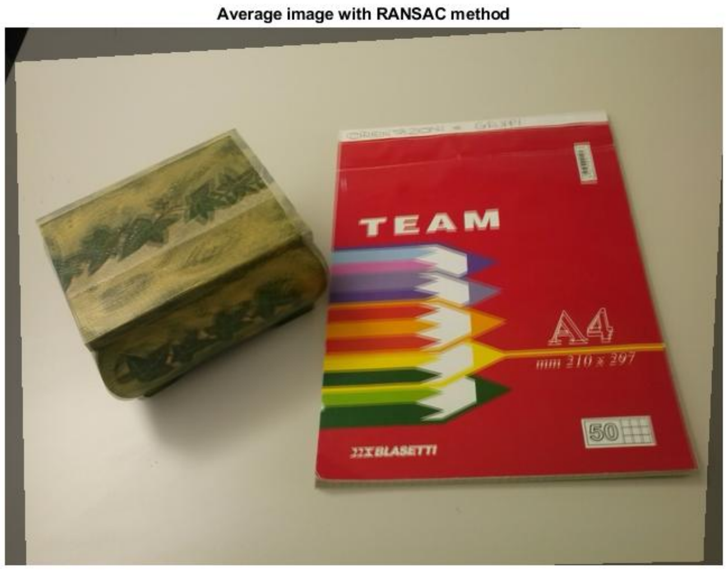

Figure 10: Average image with RANSAC

3.1 RANSAC method

The RANSAC is a statistic method which computes the H matrix different times with different correspondences (each time it takes 4 couples because 4 are the minimum to compute the homography matrix H, because H has 4 degrees of freedom) and chooses the most suitable H according to a given threshold. Obviously the number of points chosen must be greater than 4, otherwise the choice of which couples to select is unique and we can't see the differences from the normal method. In the pictures below we run the code selecting 8 points, two of them with a bad correspondence (points number 5 and 7).

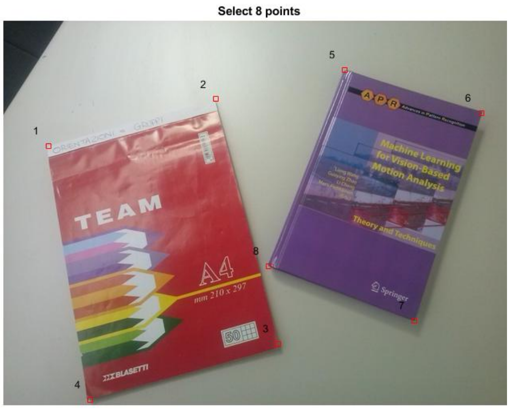

Figure 11: Points selection number one

Figure 12: Point selection number two

In the results without RANSAC the bad correspondences are taken into account and so the output is not good. Instead the results with RANSAC function are very good.

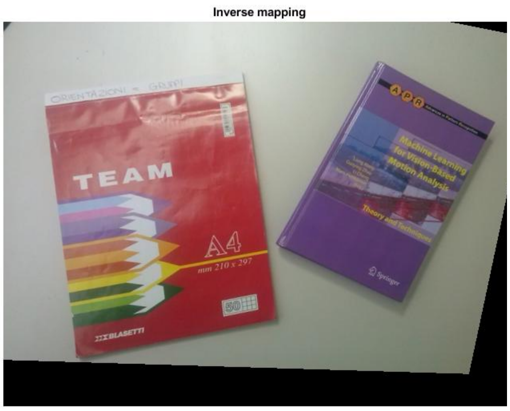

Figure 13: Inverse mapping without RANSAC

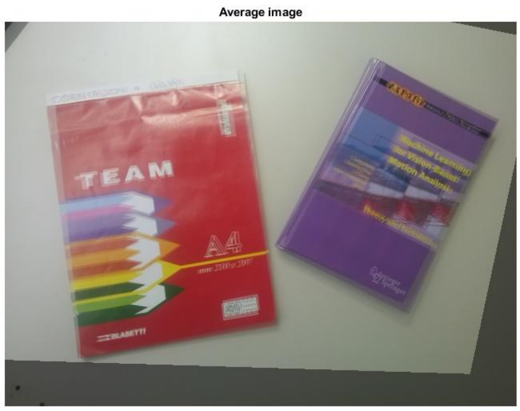

Figure 14: Average image without RANSAC

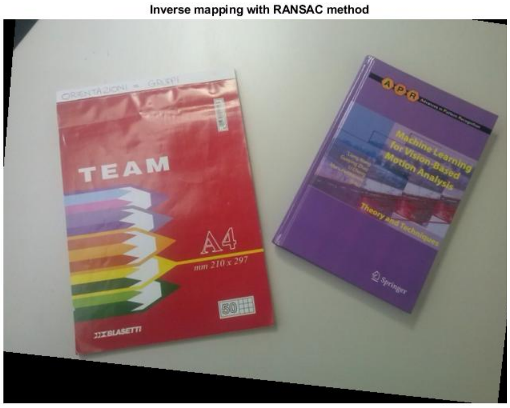

Figure 15: Inverse mapping with RANSAC

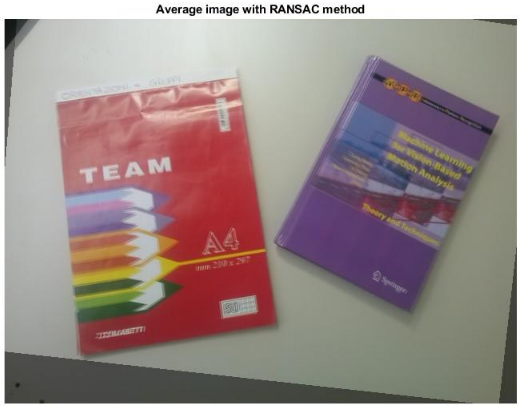

Figure 16: Average image with RANSAC

3.2 Normalized method

We tried to modify the code using normalized points to estimate the H matrix. In the pictures below we run the code selecting 6 points, three of them with a bad correspondence (points number 3, 5 and 6).

Figure 17: Point selection number one

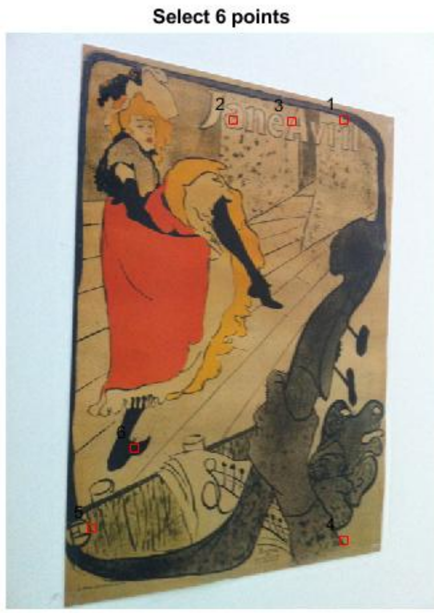

Figure 18: Point selection number two

As we can see from the result in this case the normal version is very bad while the normalized one works good.

Figure 19: Inverse mapping no normalization

Figure 20: Average image no normalization

Figure 21: Inverse mapping with normalization

Figure 22: Average image with normalization

4 Building a mosaic from a sequence of images

In this section we build images mosaic from a sequence of images acquired by purely rotating the camera exploiting homographies. This is similar to the second problem but now it is more difficult because we need 8 correspondence points: in fact the images are not 2D, and so the homography matrix has 8 degrees of freedom. The correspondence is done selecting the corresponding points between all the images and one reference frame.

Figure 23: Mosaic