

## Exercises in 3D Computer Vision

### Exercise 11 (P) Homography computation

- a) Given 4 corresponding points  $\mathbf{p}_i = [u_i, v_i, 1]^\top$  and  $\mathbf{p}'_i = [u'_i, v'_i, 1]^\top$  where  $i \in \{1, 2, 3, 4\}$ , give the expression of the entries of the homography  $\mathbf{G}$  such that:

$$\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & 1 \end{bmatrix}$$

where  $g_{jk} \in \mathbb{R}$  linking the corresponding points as follows

$$\mathbf{p}'_i \propto \mathbf{G}\mathbf{p}_i$$

- b) What are the necessary conditions to solve the problem?
- c) Give a method to solve the problem if we have more than 4 points.
- d) Write a Matlab script that makes it possible to do the following steps:
- Harris corner point detection in 2 different images.
  - Manual selection ('mouse selection') of  $N > 4$  corresponding corner points that belong to a certain plane in the image.
  - Computation of the homography that transforms the points (in the plane) from the first image to their corresponding points in the second image.

### Exercise 12 (P) Matching with SVD

In order to perform a matching between two images, we need to define a similarity and proximity measure between two detected feature points.

A simple measure or "score" between two feature points  $\mathbf{p}_i$  in the first image and  $\mathbf{p}'_j$  in the second image can be defined as follows:

$$\mathbf{G}(i, j) = \frac{C_{ij} + 1}{2} \exp\left(\frac{-d_{ij}^2}{2\sigma^2}\right)$$

where  $C_{ij}$  is the normalized cross-correlation measure computed with two windows centered around the feature points  $\mathbf{p}_i$  and  $\mathbf{p}'_j$ ,  $d_{ij} = \|\mathbf{p}_i - \mathbf{p}'_j\|$  and the parameter  $\sigma$  controls the degree of interaction between the two sets of features: a small value of  $\sigma$  enforces local interactions, while a larger value permits more global interactions.

- a) Build the matrix  $\mathbf{G}$  that gives the matching score between two Harris corner points detected in 2 images.

- b) Perform the SVD decomposition of the matrix  $\mathbf{G} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ .
- c) Compute the matrix  $\mathbf{N} = \mathbf{U}\mathbf{I}\mathbf{V}^\top$ , where  $\mathbf{I}$  is a zero matrix that has the entries on the diagonal equal to 1 (obtained by replacing every diagonal element  $\mathbf{D}_{kk}$  with 1).
- d) Two points  $\mathbf{p}_i$  and  $\mathbf{p}_j$  are matched if the entry  $\mathbf{N}(i, j)$  is the maximum value of the column  $j$  and the maximum value of the line  $i$ .
- e) Compute the homography between 2 images of a planar patch given the corresponding points. The planar region should be selected manually (using the mouse) in one of the two images. Use RANSAC to remove outliers.
- f) Warp the second image using the homography obtained and compute the absolute difference between the warped image and the first image. Plot the difference image (the region that belongs to the plane should be dark!).

### **Exercise 13 (P) Assignment: Your first special effect**

The special effect consists in inserting a “color” movie of your choice (yourself smiling or your pet running,...) into a 2nd “color” movie scene that contains a planar object moving.

Write a Matlab program that performs feature-based tracking of the planar structure, compute the homography to warp the 2nd video into every individual frame of the 1st video.

The only manual intervention in the Matlab program should be the fact of clicking on some points in the 1st frame of the 1st video in order to select the position of the plane. The output video should be in color. However, it is allowed to process one channel of the color images “internally”, i.e. into the Matlab code, to perform the tracking for example. This should be transparent for the user. The user should see only color images and should not see any black and white images.

The Matlab program does the following steps:

- a) Click some points ( $\geq 4$ ) on the 1st frame of the video to define the polygon corresponding to the plane.
- b) Track the Harris corner points detected inside the polygon along the sequence.
- c) Compute the “incremental” homography between two consecutive images of the plane.
- d) Insert a frame of the 2nd movie at the polygon position in the frames of the 1st movie. The 2nd movie images should be transformed using the previously computed homography.
- e) Save your movie as an .avi file.

Your program should work on any movies and the result should be as realistic as possible (without jittering). Please provide also all included material.

The interface for your program will be `Assignment (movie1, movie2, fileNameToSave)` and you should provide a script `showAssignment` which loads the video and tests the function.

**Remark 1** : Tracking using interframe matching drifts very quickly. That is why the matching should always be done with respect to the 1st image.

**Remark 2** : The point tracking should be based on the method above. This method can not be used if there is an important rotation between the images. To solve this problem, the image of the frame  $i$  should be warped back using the estimated homography at the frame  $i - 1$  to create the image  $warped_i$ . Then, the image  $warped_i$  should be matched with the image 0.