Computer Vision, fall 2020

Exercise 3, return latest on Sunday 27.9.2020 at 23.59 via Moodle

Ex 3.1 Line detection

Detect edges in the image Corridor1.jpg using Matlab's Canny edge detector and find as many lines as possible among the edges by implementing yourself a Hough line detector based on the method presented at the lectures.

Ex 3.2 Implement the normalized DLT (Direct Linear Transformation) algorithm

1) Take two images e.g. with your smartphone
2) Detect features from images using Matlab's SURF detector and Matlab's matching algorithm
3) Compute the 3x3 homography matrix H procedure such that xi'=Hxi (in homogenous coordinates) using DLT and at least 4-point correspondences.

To get improved performance normalize the points using the following algorithm

(i) Normalize points xi: First move their average to (0,0) by substracting their mean. Then scale them to average length of 2 . The final transformation is: T= Tscale*Ttranslate

(ii) Do the same to points xi', and obtain the transformation T'

For each correspondence xi ↔xi' compute the matrix Ai (using the transformed points); see explanation and definitions from lecture slides

Assemble the n 2x9 matrices Ai into a single 2nx9 matrix A

Obtain the SVD of A. Here you can use Matlab's SVD function

The unit singular vector corresponding to the smallest singular value is the solution h. Specifically, if A=UDVT with D diagonal with positive diagonal entries, arranged in descending order down the diagonal, then h is the last column of V

Determine the homography H from h using the following Matlab commands:

```
[U,S,V] = svd(A);
H=V(:,end);
H=reshape(H,3,3);
```

Denormalize the solution: inv(T')*H*T

("Multiple View Geometry in computer vision", by R. Hartley and A. Zisserman, p. 92)