

Send your solution to:

Andrew Comport

[Andrew.Comport@cnrs.fr](mailto:Andrew.Comport@cnrs.fr)

SUBJECT: Visual SLAM TP 2022

## Planar Homography Tracking

General objective:

Write some functions for a **planar Homography-tracking algorithm** that allows to sequentially determine the location of a planar patch throughout an image sequence via a **non-linear iterative minimisation procedure**.

The direct planar tracking algorithm functions as follows:

1. Select a patch in the first image as a reference patch.
2. Initialise the first estimate of the Homography  $\mathbf{H} = \mathbf{I}$  (identity).
3. Warp the current image using the current estimate of  $\mathbf{H}$ .
4. Compute the error between the reference patch and the warped patch.
5. Compute the update of  $\mathbf{H}$  according to the **pseudo inverse** of the Jacobian times the error.
6. If the update is still large then repeat to 3.

Code and Sequence:

Obtain a copy of the « Cyclopes » source code and the stereo sequence Versialles\_canyon. The **main file** to run is:

*cyclopes/examples/TrackImageSL3/mainTrackImageSL3.m*

Inside this file is a function *test()* at the end of the file which contains all the parameters for this program. Modify the paths of the sequence and the source code to suit your installation.

### Question 1 :

At the core of a planar Homography tracker is the **geometric warping function**:

$$\mathbf{p}_2 = \mathbf{H}\mathbf{p}_1$$

Implement this function in the directory: cyclopes/warp

The file `WarpSL3.m` should perform the geometric warping of 2D points from the Reference Image to the Current Image according to the current estimate of  $H$ . Note that the image coordinates of the Reference patch are stored in :

`ReferenceImage.P.U(ReferenceImage.index)`

`ReferenceImage.P.V(ReferenceImage.index)`

where `ReferenceImage.index` is an index of the chosen pixels.

multiply the 3(u,v, 1) coordinates by w

Don't forget to normalise the homogeneous warped point so that  $z = 1$ .

Note also that the warping function may take some points outside the image.

The results should be stored in :

`WarpedImage.P.U(WarpedImage.index)`

`WarpedImage.P.V(WarpedImage.index)`

where `WarpedImage.index` is an index of the pixels within the target region and which are inside the image.

For the program to function correctly, it is necessary to have a binary Mask image that shows which pixels are being used. This code is included at the end of the file.

## Question 2 :

The planar Homography tracker also requires an intensity warping function:

$$I_2 = I_1(p_2)$$

Implement this function in the directory: `cyclopes/warp`

The file `WarpImageSL3.m` should take the warped 2D points from question 1 and interpolate the current image at those locations. Hint, Matlab has the `interp2()` inbuilt function. Note that the Matlab `interp2()` function doesn't interpolate on the border pixels in the image.

Add the border pixels to the index: `WarpedImage.visibility_index`

Test that the warping function works when  $H=I$ .

## Question 3 :

Implement the stopping criterion for the non-linear iterative minimisation. This requires modifying the file `track/TrackImageSL3.m`

It is also possible to use several stopping criterion simultaneously.

#### Question 4 :

Test the tracking of a patch up to the end of the sequence using different options.

- Try tracking using **CurrentJacobian**, **ReferenceJacobian** and **ESM algorithm**. What differences can you note?
- Try tracking using the **M-estimator**. How does this affect the minimisation? What does the choice of the **Tukey or Huber weighting functions** do?

#### Question 5 :

Track the same patch in both **left and right images** and "calibrate" the planar Homography between left and right cameras.

- Is this Homography the same for each pair of images?
- Devise a technique to **obtain a more optimal estimate** of this calibration Homography.

#### Question 6 :

Investigate the effect of **changing** the **reference patch** each image or keeping it for as long as possible

- Plot the two graphs showing the estimated parameters using the two techniques.
- Comment on their difference.
- Devise a method for detecting when to change the reference patch.

#### Question 7: Challenge

Deviser une implementation basé sur les avancées précédents qui permet de faire le suivi d'un patch plan sur le sequence sous marin : IMAGES\_smallRGB

L'objectif sera de suivre le patch le plus longtemps possible dans le sequence.

Faire un video du resultat avec un descriptif de l'approche implementé et l'envoyer à [Andrew.Comport@cnrs.fr](mailto:Andrew.Comport@cnrs.fr) pour evaluation.

#### Question 8 :

(optional)

Augmented Reality: use the estimated Homographies to render a **virtual object** in the scene.