Master in
Computer Vision
Barcelona

Module: 3D Vision
Project: 3D recovery of urban scenes
Session 2

Gloria Haro

Session 2

Goal: compute the homography that relates to images

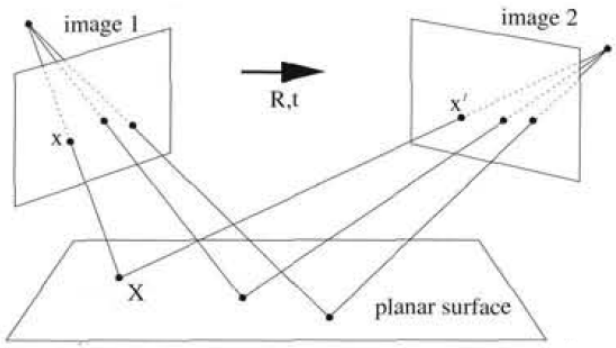
Algorithms:

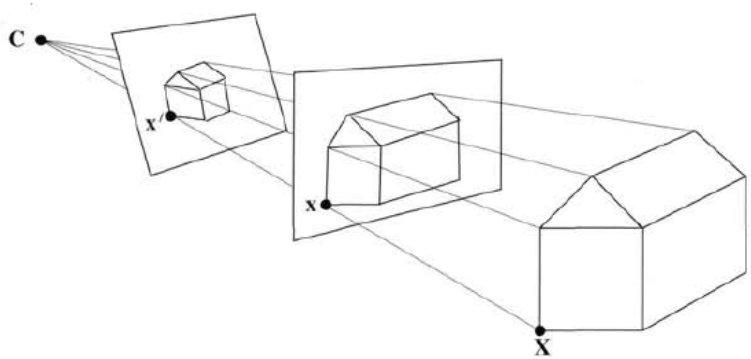
- Robust normalized DLT algorithm (algebraic method).
- Gold-Standard algorithm (geometric method).
- Camera calibration using a planar pattern.

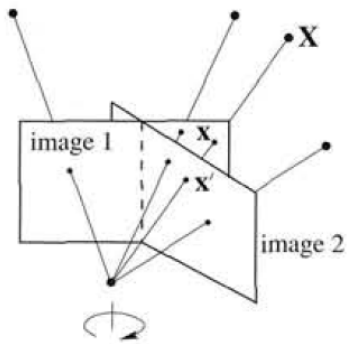
Applications:

- Image mosaics (panoramas).
- Camera calibration.
- Augmented reality.
- Logo detection in an image.
- Logo insertion in an image.

Homographies







A homography relates two images:

- of the same plane in the 3D scene;
- taken with a camera rotating about its centre;
- taken with the same static camera varying its focal length;
- the whole scene is far away from the camera.

Image mosaics

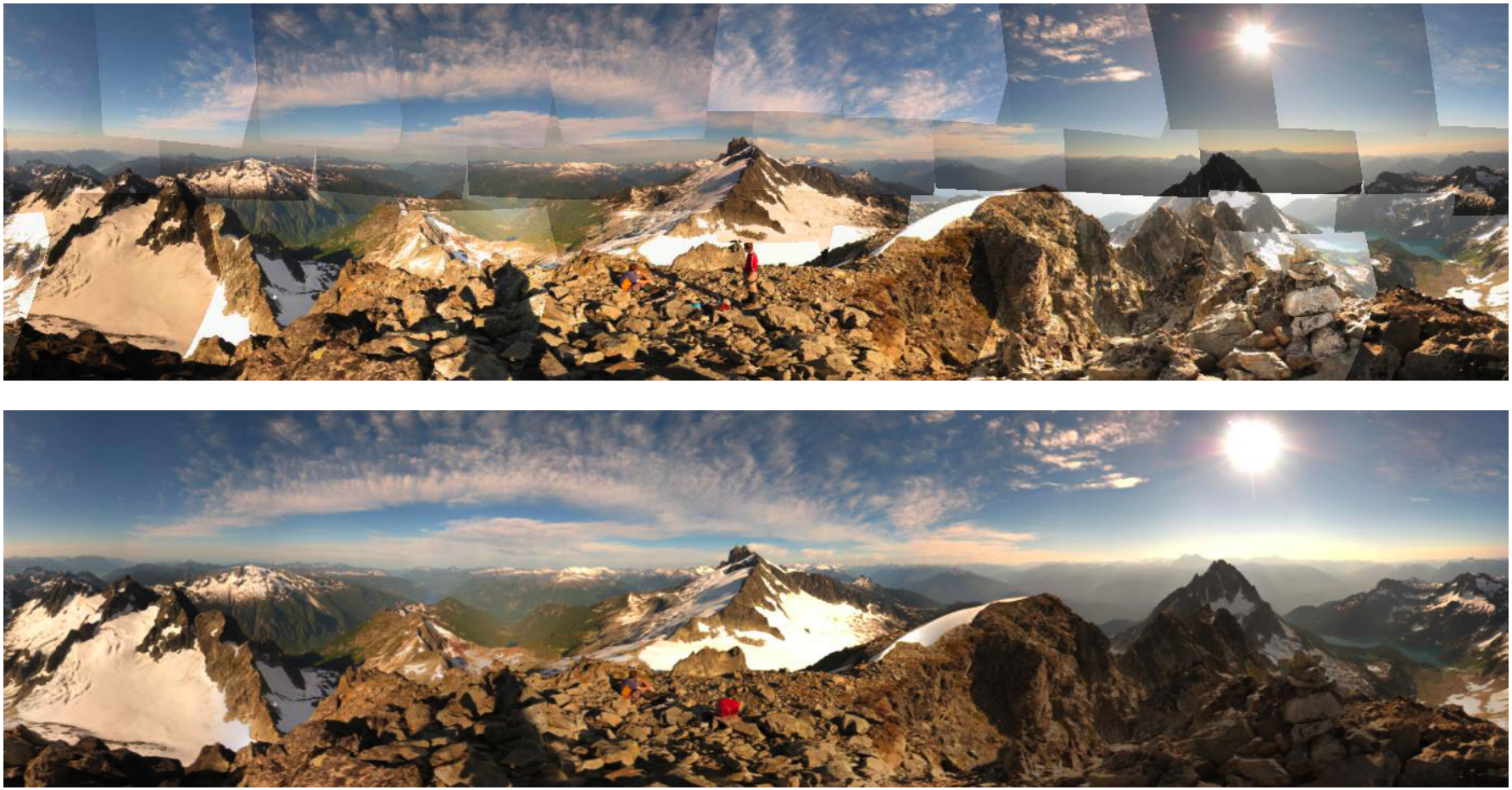


Image mosaics

Setting the canvas




Gold-Standard algorithm

Geometric algorithm: It minimizes the reprojection error

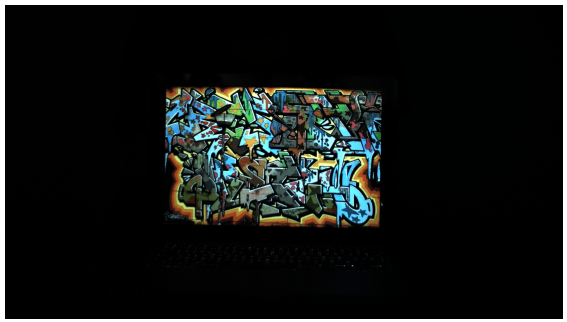
$$\min_{\hat{H},\hat{x}_i,\hat{x}'_i} \sum_i d([x_i],[\hat{x}_i])^2 + d([x'_i],[\hat{x}'_i])^2 \text{ s. t. } \hat{x}'_i = \hat{H}\hat{x}_i \forall i$$

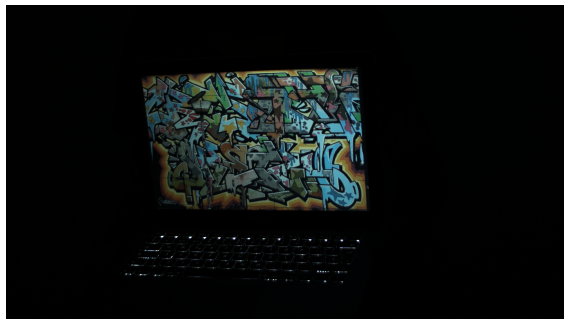
where different matchings $x_i \longleftrightarrow x'_i$ are the available data, [.] is the projection operator to Euclidean coordinates.

The above problem simplifies to the non-constrained minimization problem:

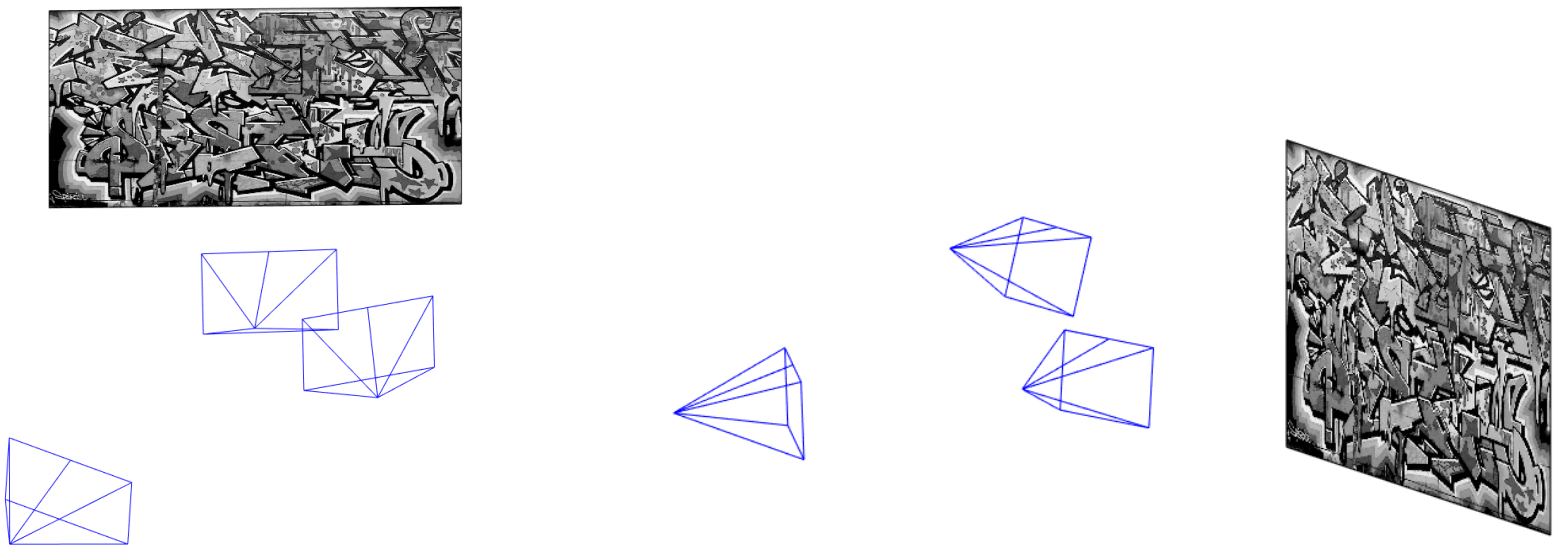
$$\min_{\hat{H},\hat{x}_i} \sum_i d([x_i],[\hat{x}_i])^2 + d([x'_i],[\hat{H}\hat{x}_i])^2$$

Camera calibration with a planar pattern





Camera calibration with a planar pattern



Augmented reality

Logo insertion

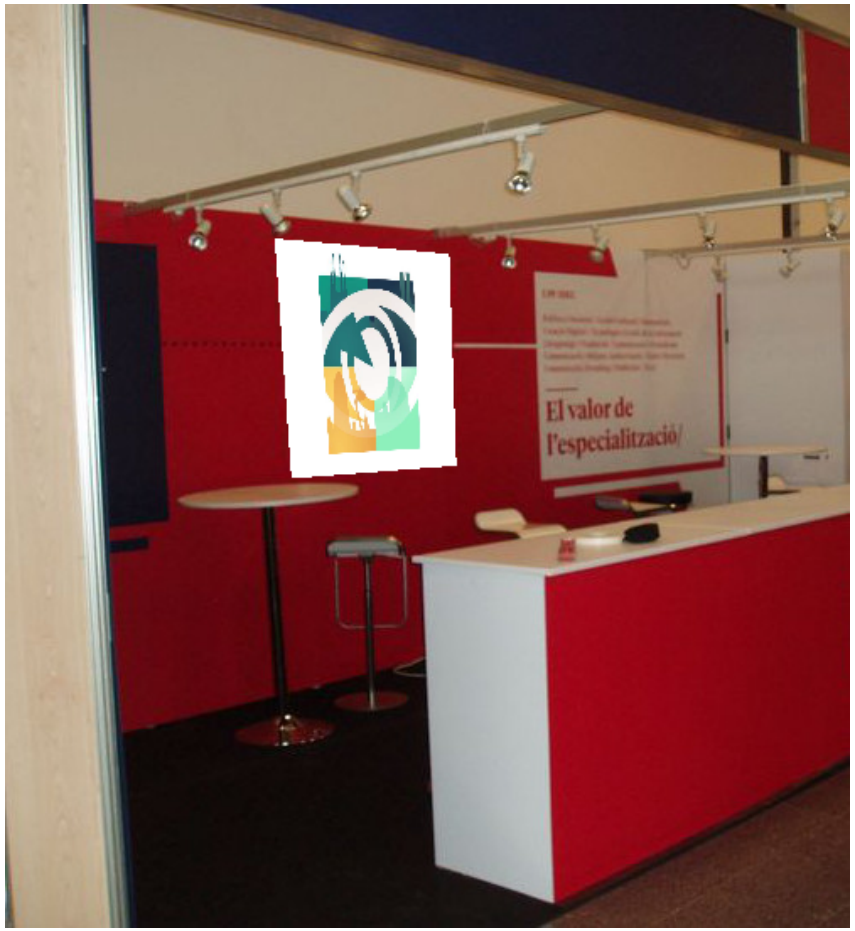
Manually selecting the four corners in the target image

Logo detection

Fully automatic

Logo replacement

Fully automatic



Session 2

Mandatory tasks:

- Function that estimates the homography with the normalized DLT algorithm given $n \geq 4$ image correspondences.
- Complete the RANSAC function.
- Compute an image mosaic with four different sets of data; compare and comment the results (why it works or it does not work in the different cases).
- Estimation of the homogaphy with the Gold-Standard algorithm.

Optional tasks:

- Complete the code on camera calibration using a planar pattern and answer (in the report) two questions raised in the file lab2.ipynb.
- Detect a logo in an image.
- Replace a logo in an image.

Session 2

Language: PYTHON

To Do:

- Complete the code in lab2.ipynb as indicated in the same file
- Write the function 'DLT_homography' (DLT algorithm)
- Complete the function 'Inliers' in 'Ransac_DLT_homography'
- Prepare the input variables and function to call the 'least_squares' function (Gold-Standard algorithm)
- (Complete the code on camera calibration)
- (Detect a logo in an image using the DLT algorithm)
- (Replace a logo in an image using the DLT algorithm)

Evaluation

To deliver **before 9am of the day before** the next lab session:

- **Code deliverable:**
  - READY TO BE LAUNCHED on the provided images
- **Short document:**
  - Results
  - Problems and comments
  - Conclusions

Evaluation

Grading:

- Report(including answers to questions): **2.75 points**
- DLT function: **2.5 points**
- RANSAC: **1.25 points**
- 4 mosaics: **1 point**
- Gold-Standard algorithm: **2.5 points**
- Optional calibration: **+ 1 points**
- Optional detect logo in an image: **+ 0.25 points**
- Optional replace logo in an image: **+ 0.25 points**

Lab 5

Installation instructions for lab 5: Colmap

In lab 5 we will use Colmap, a state-of-the-art open source solution for rigid Structure from Motion. You can install it before the session to make that sure that you can run the lab. Please contact me at marc.perez@upf.edu if you have any issues.

You can find the code here: https://github.com/colmap/colmap
Please, install from source as explained here: https://colmap.github.io/install.html#build-from-source

We will also use the python bindings: https://github.com/colmap/pycolmap
Please, install them from source as well: https://github.com/colmap/pycolmap#building-from-source

Ubuntu with CUDA installed is recommended.