# CVIS Course Assignment

Javier Morlana Jesús Bermúdez Tomás Berriel J.M.M Montiel

## I. Visual place recognition and localisation

Visual *place recognition* and *visual localisation* are two intertwined problems in computer vision. Place recognition aims to determine if two images are observing the same 3D scene. The two images might differ in illumination or camera location. Also, if the images were taken at different times, the scene might have changed. Fig. 1 shows two images of 'El Pilar' from different points of view, with different cameras and widely separated in time, 150 years. Visual localization aims to determine the pose of a camera that is observing a scene, for example in Fig. 1, to estimate where the photographer that took the old photo was localized. Fig. 1 also shows another example in which the images are frames extracted from two colonoscopies of the same patient separated by two weeks.

## II. Objective

The main goals of the assignment are visual localisation and visual place recognition from real images. More specifically:

1) Computing a precise localisation of the camera that took the old photo. You have to describe the world frame used for the localisation and the localisation itself.
2) Determination of which regions of the old photo correspond to areas of the scene that have been modified since the image was taken and which correspond to unchanged areas.

## III. Methodology

Each student **has to select one of the two use cases** of place recognition and visual localisation. The first one, *buildings use case*, aims for two widely separated in time images of a rigid scene, typically urban scenes containing buildings or monuments. The second, *colonoscopy use case*, aims two separated in time images of a real human colon using images from the EndoMapper dataset [2].

This section describes the data and the software to be used.

### A. Image acquisition

*1) Buildings use case:* You have to select an old picture, a photo taken before 2000. Additionally, you have to take several new images (3-10) of the same scene with your mobile phone. Fig. 3 shows an example of an old picture and the new images corresponding to "El Pilar".

It is recommended that at least one of the new images is taken from a similar point of view to the old picture to ease the matching. Scene points at different depths but overlapped in the images allow to verify if the two points of view are coincident (see Fig. 2). Additionally, from these points and an aereal view, the old camera pose can be localised.

In any case, consider selecting a place where taking images is easy at any time because you might have to make several attempts before having the definitive set of images.

*2) Colonoscopy use case:* The old picture corresponds to a frame of a colonoscopy, `Seq_027`, imaging the rectum colon of a patient. The set of new images are extracted from a second colonoscopy, `Seq_035`, of the same patient performed two weeks afterwards. See the images in Fig. 4. These images are provided as data for the assignment.

### B. Camera calibration

Calibration is needed for 3D modelling and camera pose estimation. The old and the new cameras are different and might have different calibration parameters. Next, it is detailed the estimation of the calibration parameters.

*1) Buildings use case:* You have to take a set of images of a planar calibration pattern and estimate the calibration of the new camera, Fig. 5 displays an example of calibration images. You have to write a program to estimate the calibration of the old camera from the camera pose.

*2) Colonoscopy use case:* Both colonoscopies were taken with the same colonoscope, hence they have the same calibration. The endoscope has a fish-eye lens already calibrated with the Kannala-Brand model as part of the EndoMapper dataset. The calibration parameters are provided as data for the assignment.

### C. Discrete point matching

Any point detection/description/matching combination can be used. It might be worth testing the new lightglue matcher [4].

### D. Your SfM software

During the labs you have developed your own SfM libraries. It is compulsory to use your libraries to compute the old camera pose. You can use COLMAP to have a reference solution to compare with, which very likely will help you in the debugging and tuning of your SfM libraries.

### E. COLMAP

Colmap is an open-source library for SfM `https://colmap.github.io/index.html` [5], [6]. Using COLMAP you can process a discrete number of images even a video shot to estimate both the scene structure and the camera poses. Additionally, it can produce dense 3D models using multiview stereo. COLMAP operates from raw images using SIFT detector and descriptor but you can also

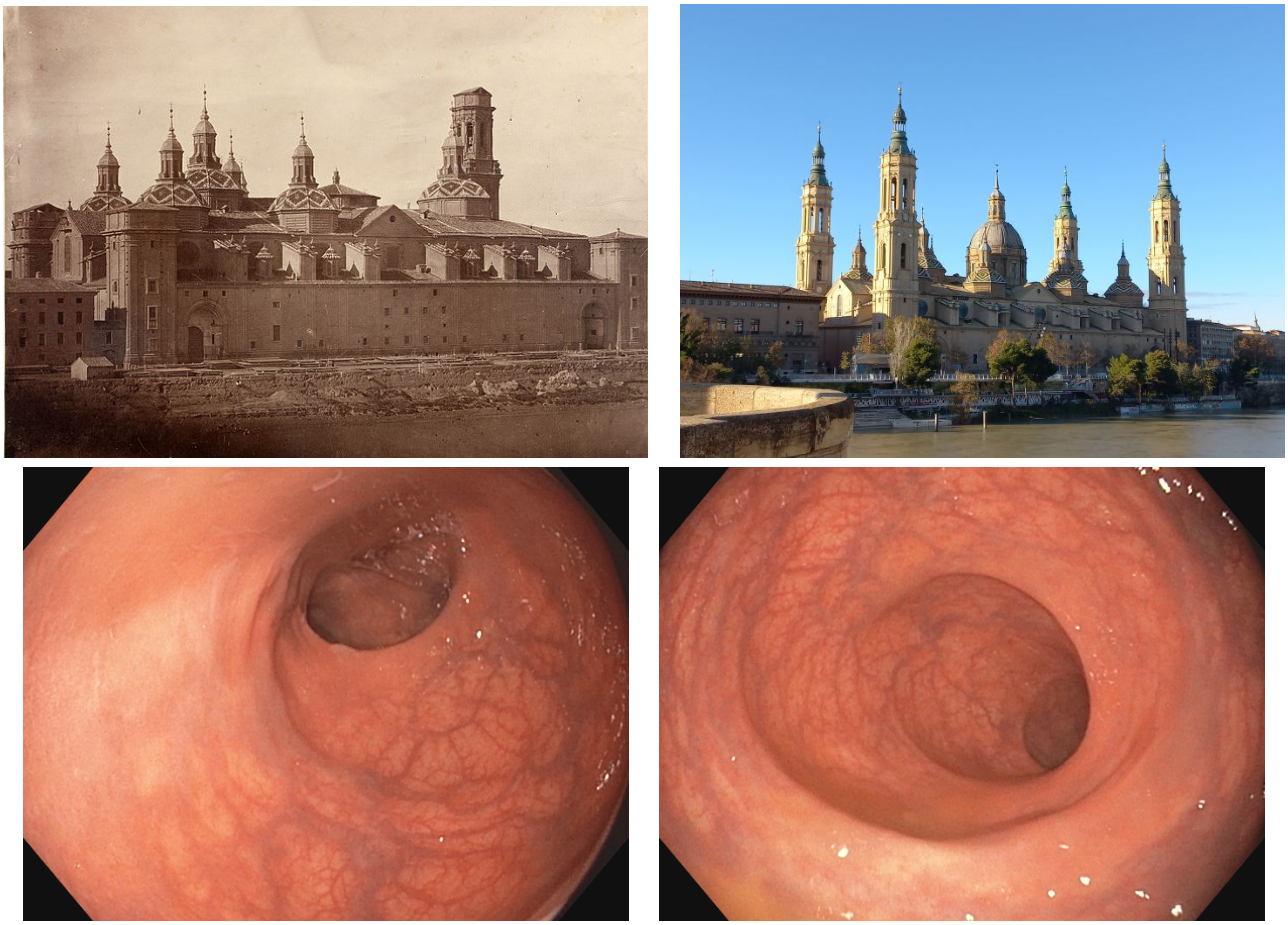

Fig. 1. Visual place recognition examples. (top left) 'El Pilar' old picture dated 1860 [1]. (top right), 'El Pilar' new image dated 2023. (bottom left), old image from a first colonoscopy (`Seq_027`). (bottom right), new image from a second colonoscopy of the same patient (`Seq_035`). [2]

import pairwise matches coming from any other detector, for example from SuperPoint + SuperGlue + RANSAC.

See the Section V for hints and an example of how to import matches in COLMAP.

### *F. Change detection*

Change detection is an open problem. We recommend exploiting the fact that you can establish geometrical relationships between the old photo and the new images. The geometrical relationships can be computed either using your SfM software or COLMAP.

## IV. Assessment

E02 -Course assignment, a project which globally combines the contents of the course (70 %). Based on a per-student per-student 15 minute oral talk. For the 15 minute talk, it is recommended 10 slides using Powerpoint or Google slides. You can include short videos in the talk.

All the texts and figures used in the slides have to be original work from the author, otherwise, it has to be explicitly stated in the slides, providing reference to the source.

### *A. Deliverables*

Each individual student *has to select one use case* and fulfil the next deliverables before the deadline detailed in add:

1) Uplodad in add the slides for the talk.
2) Make oral talk. Reserve the slot in add.

### *B. Criteria*

1) Slides & Oral talk 10 %.
2) Quantitative information 10 %. Tables reporting, for example, RMSE errors, computing time or number of matches.
3) Computing the old photo camera localisation 50 %
    a) Proposed method 10 %
    b) Experimental performance 15 %.
    c) Comparison wrt COLMAP 15 %
    d) COLMAP using 10 or more new images 10 %
4) Determining the changes 30 %
    a) Proposed method 15 %
    b) Experimental performance 15 %

## V. COLMAP usage

Basic COLMAP usage can be found at `https://colmap.github.io/tutorial.html`. You can try to

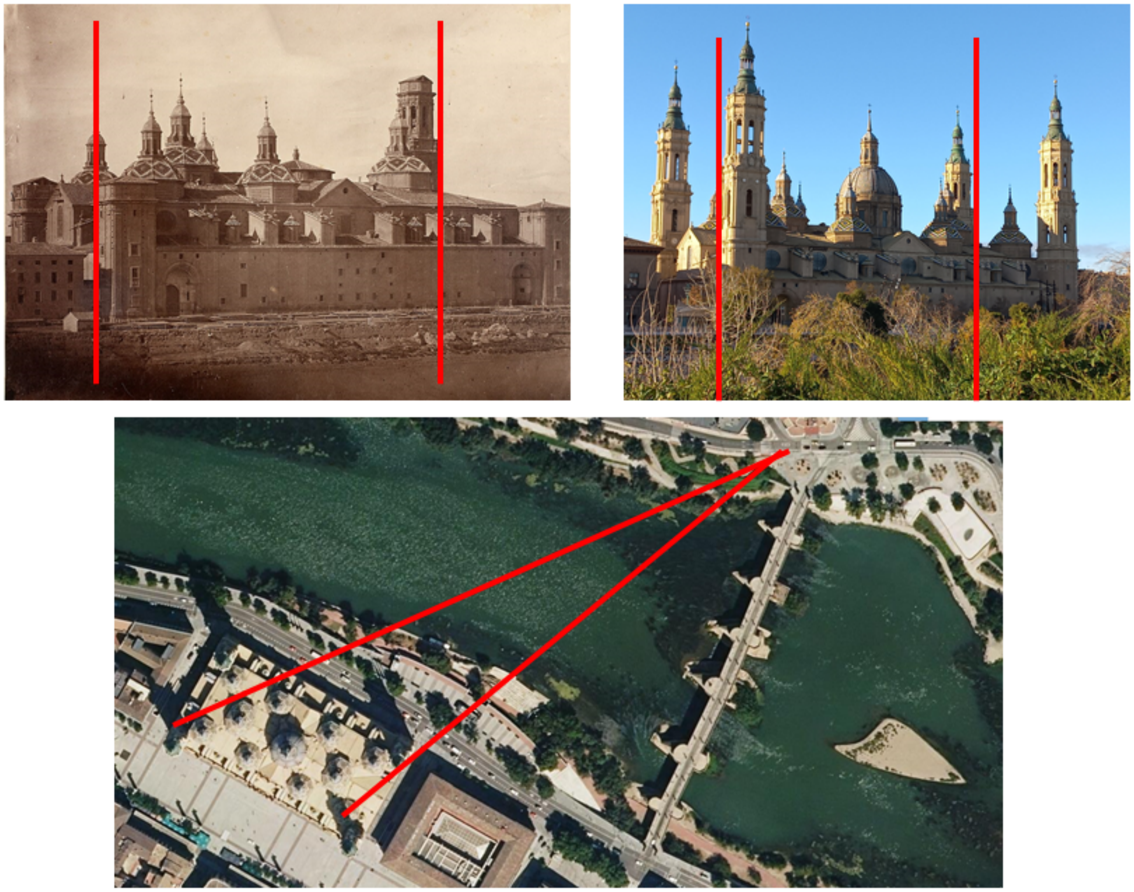

Fig. 2. Approximate estimation and verification of similar point of view. Scene points at different depths but overlapped in the image are highlighted in red in the old picture (top left), and in the new image (top right). (bottom) top view on aereal image of the projection rays and old camera pose. [1] [3].

Fig. 3. Example images of 'El Pilar' for the building use case. Top-leftmost, the old picture dated 1860 [1]. The rest are new images taken in December 2023. The new image with a similar point of view to the old is shown in first row second column.

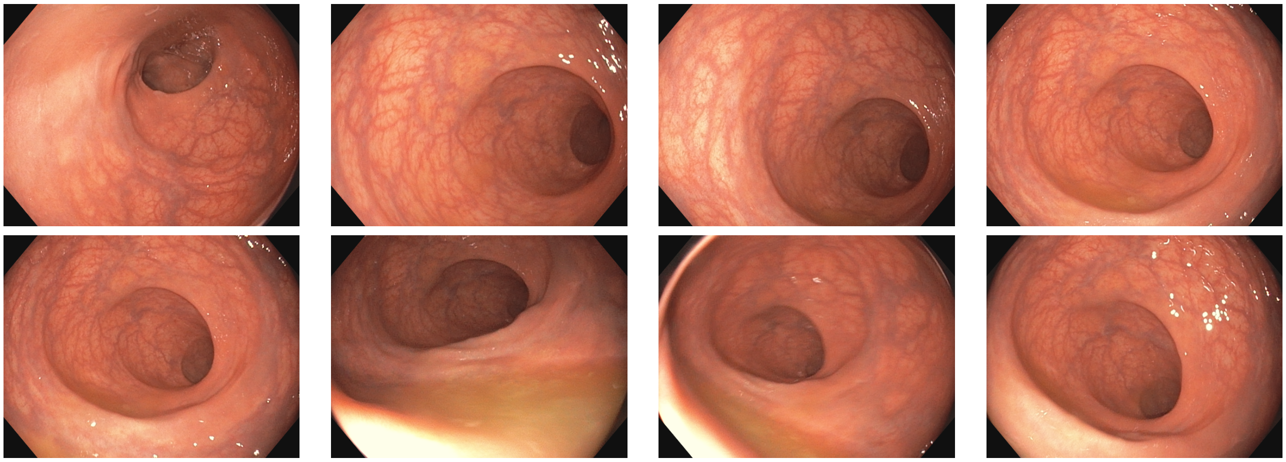

Fig. 4. Colonoscopy use case images. Top-lefmost image is the old image from `Seq_027`. The rest are the new images extracted from `Seq_035`

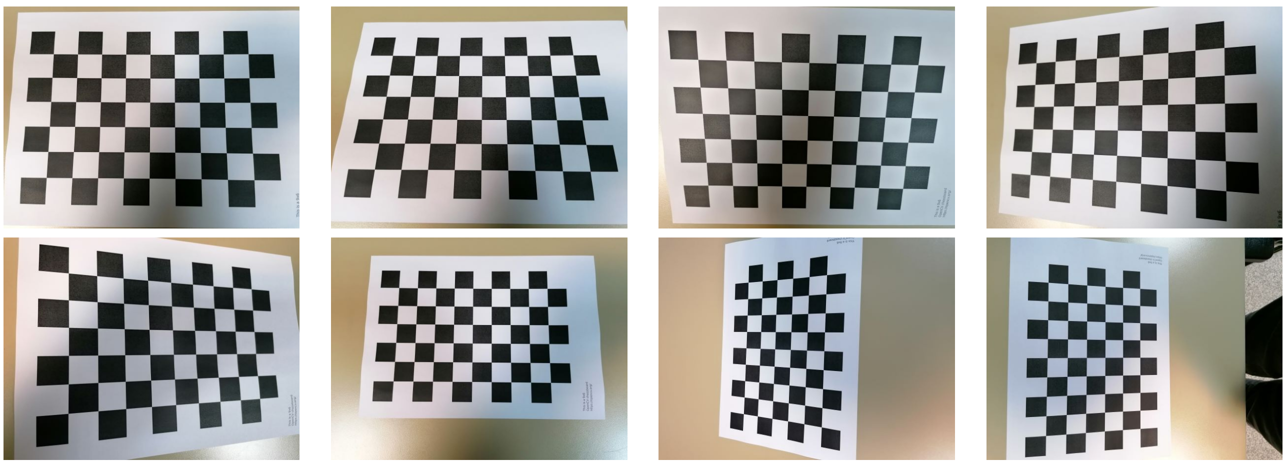

Fig. 5. Calibration images.

reconstruct your images with default parameters. COLMAP will extract and match SIFT features, applying two-view robust RANSAC geometry refinement, triangulation and Bundle Adjustment.

Here we explain how to use COLMAP with custom features and matchers i.e. SuperPoint + SuperGlue. You will need to give the files in the format specified in feature detection and feature matching.

We exemplify this with the use of Superglue, which can be found in this repository. Specifically, the script `match_pairs.py` can read a `.txt` file where you specify every pair of images that should be matched with SuperPoint and SuperGlue. Be careful with default values such as the image resolution and the max number of keypoints, as they will impact performance. By default, the script will export an `.npz` file with the matches that are above a confidence threshold. You should modify the script to apply RANSAC two-view geometry verification to filter inliers/outliers. After that, export all the raw keypoints extracted in both images and the inlier matches. To use these keypoints and matches in the COLMAP pipeline, you will need to create two types of files. For keypoints, you will create a `.txt` file for every image (See Fig. 6).

Attached you will find one `.txt` example showing the content of each of these files. The format is the one defined in feature detection, but you will need to sum 0.5 to each of the $(x, y)$ keypoint coordinates (COLMAP uses a different image origin). For scale and orientation write 0.0 and for the 128 entries of the descriptor vector write 0's:

```
NUM_FEATURES 128
x+0.5 y+0.5 0.0 0.0 0 0 0 ... 0
```

For the matches, you will create a `.txt` file with the format defined in feature matching. For each pair of images, you will specify the id of the keypoints matched. For example,

```
image1.jpg image2.jpg
0 1
1 2
3 4
```

Considering kpts1 and kpts2 the arrays containing the kyepoints for the image1 and image2, this means that kpts1[0] matches with kpts2[1], kpts1[1] with kpts2[2], and so on.


Fig. 6. Defining key points for the images

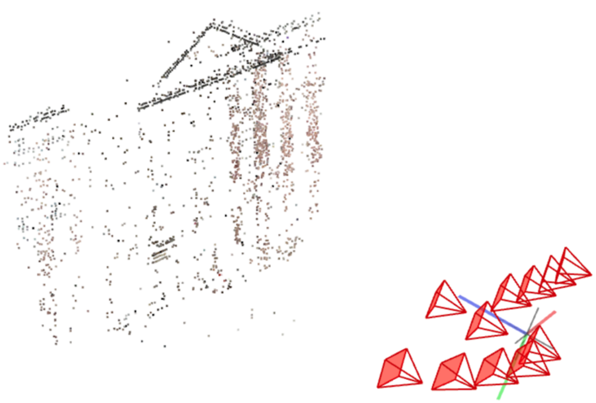

Fig. 7. Example of a 3D model and camera poses

Attached you will find an example of `matches.txt`.

Finally, you can use these matches to obtain a COLMAP reconstruction. You can either use the GUI application or the command-line. For the command-line, importing the features and matches is done with the following commands:

```
colmap feature_importer --database_path
path/to/database.db --image_path
path/to/images --import_path
path/to/images

colmap matches_importer --database_path
path/to/database.db --match_list_path
path/to/matches.txt --match_type
'inliers'
```

Then, you can open the database and reconstruct the model from the GUI to visualize the 3D model (See Fig. 7).

## References

[1] Unknown Author, "Vista posterior de la Basílica del Pilar de Zaragoza," https://commons.wikimedia.org/w/index.php?curid=24970366, Public Domain. Accessed: 2023-20-12.

[2] P. Azagra, C. Sostres, Á. Ferrández, L. Riazuelo, C. Tomasini, O. L. Barbed, J. Morlana, D. Recasens, V. M. Batlle, J. J. Gómez-Rodríguez *et al.*, "Endomapper dataset of complete calibrated endoscopy procedures," *Scientific Data*, vol. 10, 2023, article number: 671.

[3] Unknown Author, "Plaza Pilar (del) Zaragoza ," https://fototeca.cnig.es/fototeca/, BY CC. Accessed: 2023-20-12.

[4] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local Feature Matching at Light Speed," in *ICCV*, 2023.

[5] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016, pp. 4104–4113.

[6] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *ECCV 2016*, 2016, pp. 501–518.