M4 - 3D Vision

3D recovery of Urban Scenes

Ana Caballero - ana.caballeroc@e-campus.uab.cat
Arnau Vallvé - arnau.vallve@e-campus.uab.cat
Claudia Baca - claudiabaca.perez@e-campus.uab.cat
Joaquim Comas - joaquim.comas@e-campus.uab.cat

INTRODUCTION

The goal of this project is to learn the basic concepts and techniques to reconstruct a real world scene given several images (points of view) of it, not necessarily previously calibrated. In this project we focus on 3D recovery of Urban Scenes using images of different datasets, namely images of facades and aerial images of cities.

WEEK 3

Goal: compute the fundamental matrix that relates two images

Mandatory:

- Function that estimates the fundamental matrix F with the normalized 8-point algorithm.
- Compute the theoretical fundamental matrix that relates two images with corresponding camera matrices P = [I|0], and P'= [R|t].
- Function that robustly estimates F using the previous function and RANSAC.
- Compute the epipolar lines of the matching points in both images.
- Apply the theoretical concepts to do photo-sequencing.

Optional:

- Photo-sequencing with your own images

1. Normalized 8-point algorithm

In this first part, we were are asked to estimate the fundamental matrix by the 8-point algorithm, the fundamental matrix is a 3x3 matrix, so we have 9 unknowns, and we can get rid of one of them by the scale, then we will need 8 equations as minimum, in other words, we will need 8 set of points p and p' correspondences between two images. As we have seen in class, the relationship between F an a set of points in correspondence can be expressed:

$$\tilde{p}'^{T} F \tilde{p} = 0$$

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

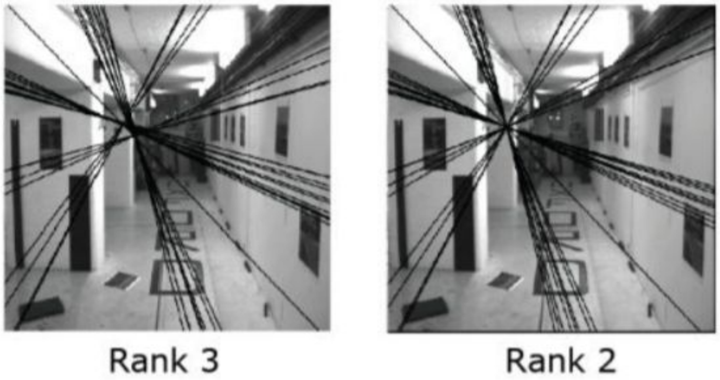
And that can be express as this system of equations:

$$W\,f = 0$$

$$\begin{bmatrix} uu' & vu' & u' & uv' & vv' & v' & u & v & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

So we compute W, that it is formed by the 8 set of points, this points have been previously normalized in order to stabilize the estimation of the F, we can say that they are transformed in a new normalized system. Once we have the W computed, we can solve the system of equations by using SVD, in the last column of V (null vector)we will find f, our solution. Then we can compose F as a 3x3 matrix. However, the rank of the F should be 2 , in other the matrix F should be singular, in order to make the epipolar lines coincide in only one point, the epipolar.

At the image below we can see the difference between having and F with rank 3 and 2:



Rank 3 Rank 2

So, to ensure that the F has rank 2 , it was done a new svd decomposition of the resulting F and then the recomposing in the next way:

$$F_{RANK\_3} = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T$$

$$F_{RANK\_2} = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

With the previous part we have already the F estimated and the only step missing is the denormalization of it, in order to have the F and the image coordinates in the same space. Below we can see how the denormalization is done:

$$F = H'^T F_q H$$

Where:

- H is the homography that relates the space of the image coordinates and the normalized space.
- F_q is the Fundamental matrix normalized.

2. Compute the theoretical fundamental matrix

In this next section, we have computed the theoretical fundamental matrix where we assume that the world reference system coincides with the reference system of the left camera. Then:

$$P = k[I \mid 0] \qquad\qquad P' = k'[R \mid t]$$

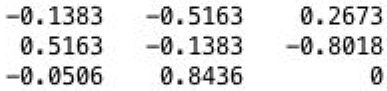
where:

- K, K' are the calibration matrices in both cameras.
- R and t express the relative pose of the reference systems of both cameras.

Taking in account this assumption then we have that the theoretical fundamental matrix is:

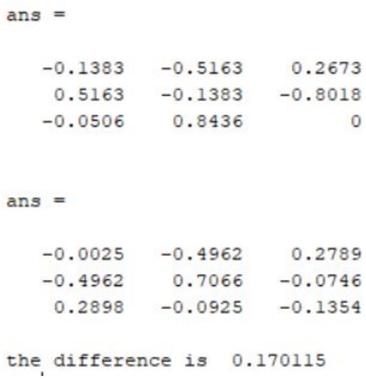
$$F = K'^{-T}[T'_x]RK^{-1}$$

In our case, K1 and K2 are equal to the identity matrix because P1 and P2 are a Rotation and translation. The resulting theoretical fundamental matrix using the camera parameters and after normalize the two matrices we have:



```
   -0.1383   -0.5163    0.2673
    0.5163   -0.1383   -0.8018
   -0.0506    0.8436         0
```

After that, we have compared the resulting matrices between the theoretical fundamental matrix (F_gt) and the estimated fundamental matrix (F_es) which was computed using the normalise 8-point algorithm. In the following figures we have two examples where we can see that the theoretical and the estimate fundamental matrix are very similar (absolute difference around 0.18):



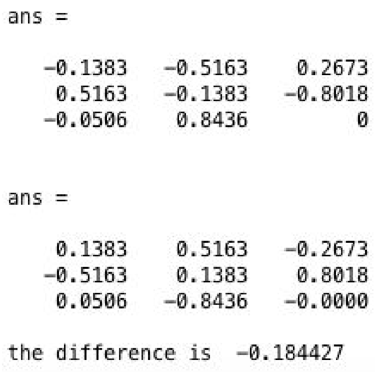
```
ans =

   -0.1383   -0.5163    0.2673
    0.5163   -0.1383   -0.8018
   -0.0506    0.8436         0


ans =

   -0.0025   -0.4962    0.2789
   -0.4962    0.7066   -0.0746
    0.2898   -0.0925   -0.1354

the difference is  0.170115
```



```
ans =

   -0.1383   -0.5163    0.2673
    0.5163   -0.1383   -0.8018
   -0.0506    0.8436         0


ans =

    0.1383    0.5163   -0.2673
   -0.5163    0.1383    0.8018
    0.0506   -0.8436   -0.0000

the difference is  -0.184427
```

We will get different values for the estimated fundamental matrix in each execution we pick random values for the World Point (X) coordinates.

3. Robust normalized 8-point algorithm

In previous section we have explained the normalized 8-point algorithm to estimate the fundamental matrix. One of the disadvantages of this method is the sensitivity to false matches (or correspondences). To solve this problem what we have implemented is a combination between Ransac and the normalized 8-point algorithm.

The process of this robust method was the following:

- RANSAC loop:
  - Select 8 points randomly
  - Estimate F from selected points
  - Determine the number of inliers which comply a threshold
  - Choose F with the largest number of inliers.
- Re-estimate F using all inliers

An important part in this method is the choice of the geometric distance that it is used in the computation of the inliers. In our case, we have used the Sampson distance which has the advantage that the resulting cost function only involves the parameters of F without introducing new variables such as the Gold Standard.

Therefore, our computation of the inliers is based on the equation which defines the Fundamental matrix:

$$x'^{T} \cdot F \cdot x = 0$$

The idea is that two correspondences x and x' in two images, I and I', must comply the previous equation (they should be the same point). Taking in account this concept we have computed this expression for each pair of correspondences and their respective epipolar lines.

With all this information we can compute the Sampson distance which has the following expression:

$$\sum_i \frac{(x_i'^T F x)^2}{(Fx_i)_1^2 + (Fx_i)_2^2 + (F^T x_i')_1^2 + (F^T x_i')_2^2}$$

Using the Sampson distance and a threshold which we initialize to 2.0 we have achieved a better performance of the normalise 8-point algorithm removing more outliers.

During this third lab session we have used the following two images which are from the same building.

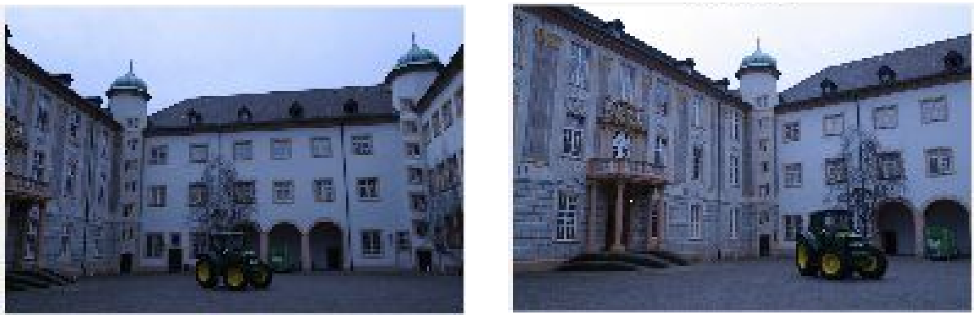


Image 1, 2: Right and left facade of the building

Then we have used SIFT to find the correspondences between the two facades:

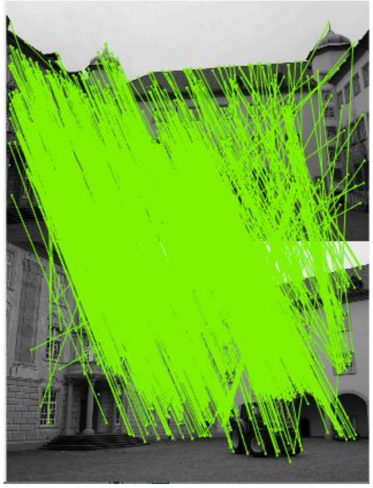


Image 3: Correspondences between image 1 and 2

Finally, after computing the robust method using Ransac and normalise 8-point algorithm we can appreciate in the next figure that we can find each point in one image in the second image using the epipolar lines:

Image 4: Example of point correspondences between image pairs that match, as they are present on both images (i.e. overlapping regions)

4. Compute the epipolar lines

In this next section, we have selected 3 points in a overlapping region which they represent the same point in the real world. In our case, these 3 points are located in the left facade as we can notice in our two images:

Images 5,6: Location of the 3 pairs of correspondences

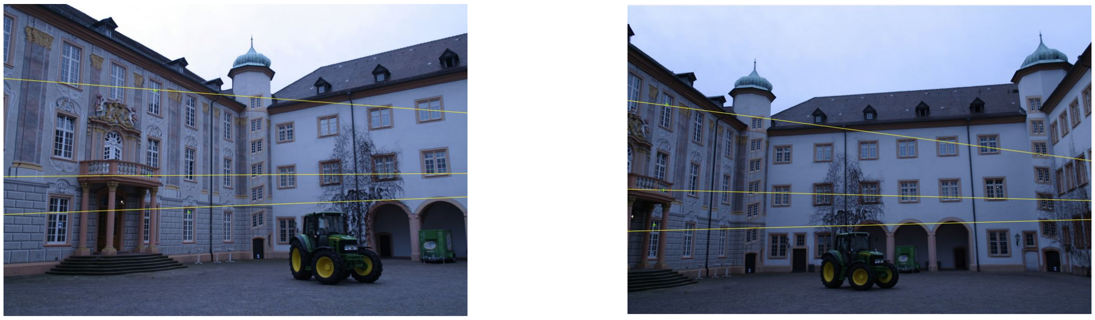
To match these selected points we have computed the epipolar for each image. On one hand we have computed the epipolar lines in the second image corresponding to the points, in the first image.

$$l_2 = F \cdot [x_1\, y_1 1]^T$$

On the other hand, we have computed the epipolar lines of the first image corresponding to the points, in the second image.

$$l_1 = F^T \cdot [x_2\, y_2 1]^T$$

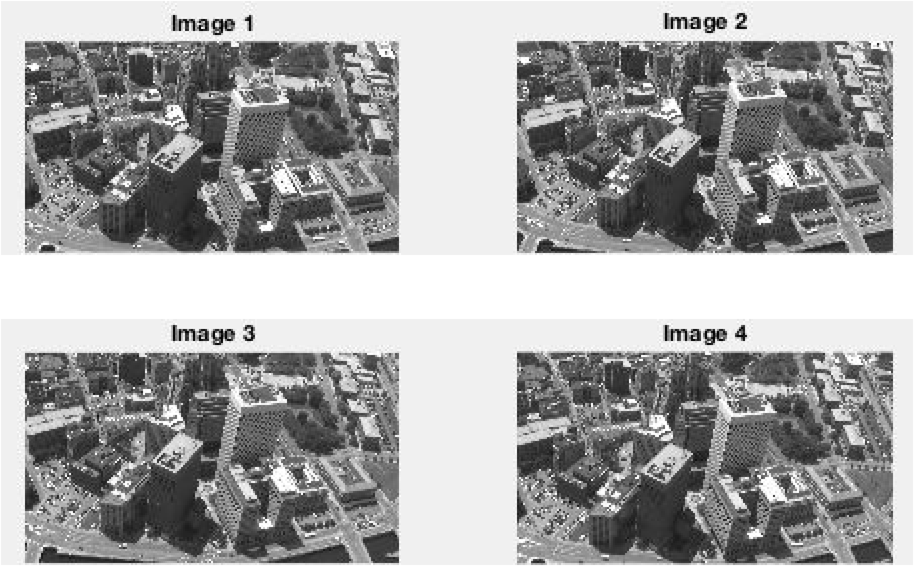
Finally, in the following images we can see the results of our epipolar lines for each image which they intersect quite good the 3 points used:



Images 7, 8: A pair of images with superimposed corresponding points and their epipolar lines

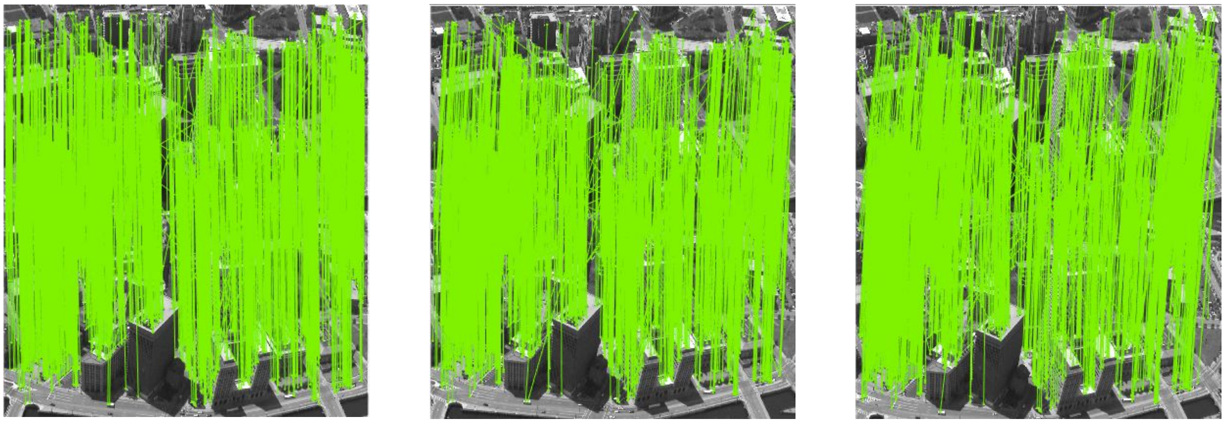
5. Photo-sequencing

For this task we will work with the set of 4 aerial images taken at a small time interval difference from slightly different points of view (approximately the same position) we can see below. The goal is to recover the temporal ordering of the provided images given a manually picked dynamic point (a van in the road) which is moving roughly following a straight line.



Images 9, 10, 11, 12: Aerial images used for the simplified photo-sequencing algorithm

From each of the 4 images in the set, using the Robust normalized 8-point algorithm using RANSAC seen in previous tasks, we compute the SIFT keypoints which will be used to obtain the fundamental matrices between each of the images and the reference image (image 1). With this, we obtain the correspondences between the reference and the rest of the aerial images.



Images 13, 14, 15: Correspondences between reference and the rest of aerial images obtained with Robust normalized 8-point algorithm.
From left to right: [ref, img2], [ref, img3], [ref, img 4]

In order to be able to retrieve the temporal ordering we would like to have the points corresponding to the van (dynamic feature) from all the aerial images projected into the same reference image so that we can have a way to establish the ordering in time of the images.

First step is to obtain the line l_i joining the two points at different times which are close from each other (t1, t2); one that defines the coordinates of the van in the reference image 1 and another point from another image that is put in reference image coordinates (which was given data, in our case).

$$\ell^i = p_1^i(t_1) \times p_2^i(t_2)$$

We also need to obtain the epipolar lines of each of the k points by using the previously obtained corresponding fundamental matrices F which corresponded points from the reference image to the rest. The epipolar lines l_{ki} will be obtained then computing:

$$\ell_k^i = F_k p_k^i(t_k)$$

Finally we find the points that intersect each of the obtained epipolar lines *lki* with the previously computed line *li*, so that we obtain the projections of each of the “van” points from the different images in the reference image, so that we can establish the temporal ordering.

$$p_1^i(t_k) = \ell^i \times \ell_k{}^i = (p_1^i(t_1) \times p_2^i(t_2)) \times \left(F_k\ p_k^i(t_k)\right)$$

From the paper *Dekel, T., Moses, Y., & Avidan, S. (2014). Photo sequencing* from which this algorithm has been extracted we can also show the diagram to make this equations make sense in a graphical manner, and the results easier to follow.

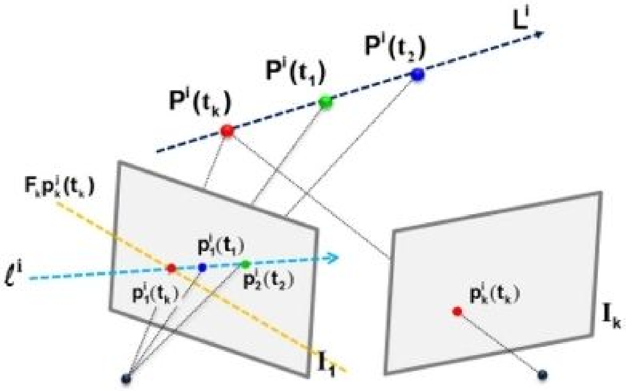


Image 16: Diagram representing the geometric basics of how to perform photo-sequencing from k images.

By looking at the points obtained, just to notice it, we can see how the yellow and cyan lines corresponding to points 1 and 2 respectively are the closest together, as they were the ones used to compute the line *li* from which we wanted points roughly in the same position. We recall that one of the hypothesis for the photo-sequencing algorithm is that “Two of the images are taken from approximately the same position”, those being image 1 and 2 in our case.

Qualitatively, the results make sense as the points from the intersection of the epipolars and *li* fall inside the road, as we would expect the path of the van to be on. Here the effect of the second hypothesis of the photo-sequencing algorithm can be observed. We assume that the trajectories can be approximated by a straight line, assumption which can be seen at effect by looking at the yellow line.

Color to image correspondences are as follows: [Yellow, img 1], [Cyan, img 2], [Blue, img 3], [Green, img 4]. From the spatial ordering of the intersecting points we can get the temporal ordering of the images. So looking at the results below, with the intersection points computed and shown on the reference image we can set the temporal ordering, assuming the van goes from left to right, as [blue, yellow, cyan, green] which would correspond to [img 3, img 1, img 2, img 4]. If we visualize the original images and focus on the van position we can see, however, that even though the temporal ordering obtained make sense for images 1, 2 and 4, they do not make sense for image 3. The reason seems to be in the fact that the blue epipolar line from img 3 is close to following the trajectory of the van, which can produce noisy results.

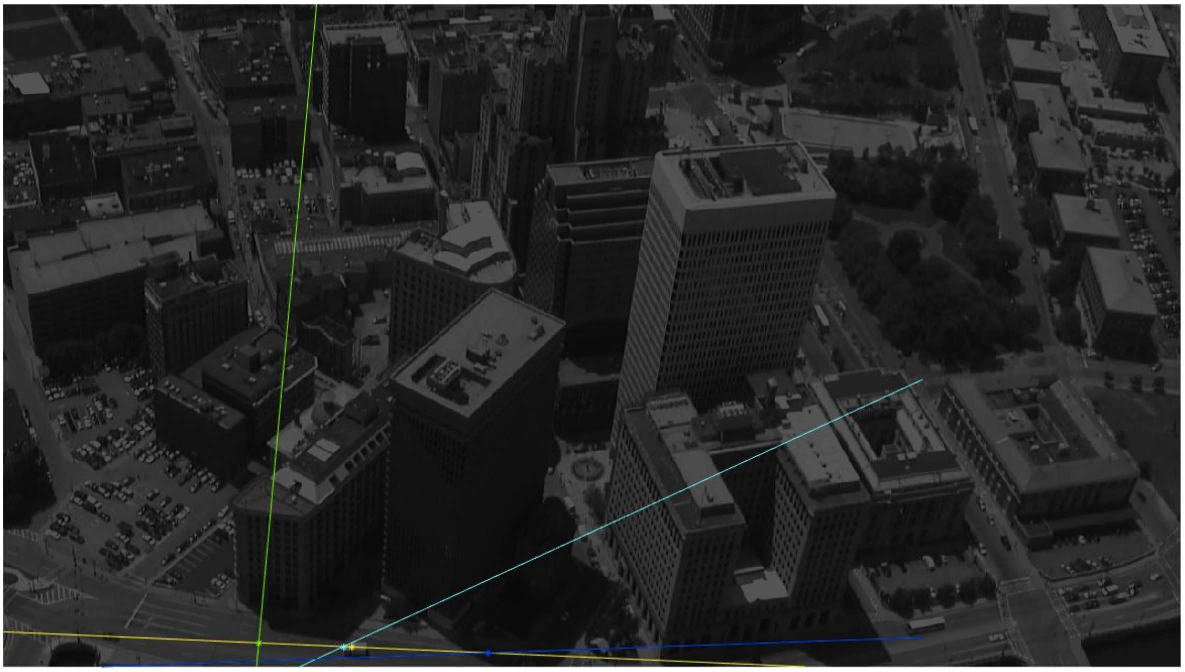
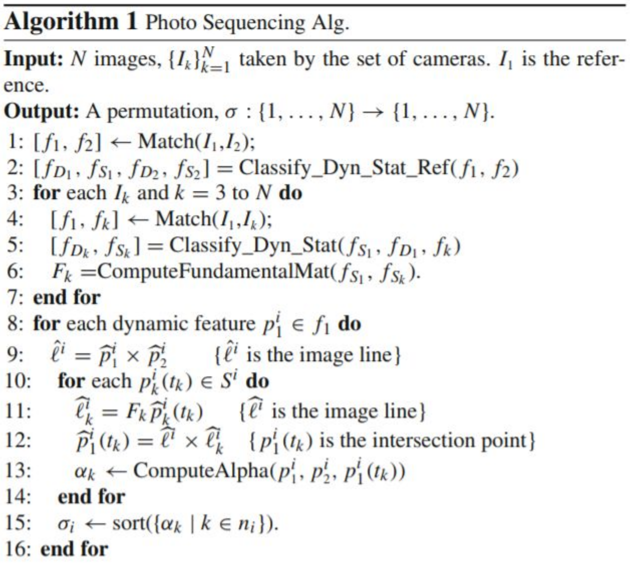


Image 17: Van trajectory plot in reference image 1. Color to image correspondences: [Yellow, img 1], [Cyan, img 2], [Blue, img 3], [Green, img 4].

6. OPTIONAL: Photo-sequencing with your own images

We have schematically developed the complete algorithm of Photo Sequencing, which is the following:



Algorithm 1 Photo Sequencing Alg.

Input: N images, $\{I_k\}_{k=1}^{N}$ taken by the set of cameras. I_1 is the reference.

Output: A permutation, $\sigma : \{1, \dots, N\} \rightarrow \{1, \dots, N\}$.

```
$[f_1, f_2] \leftarrow$ Match($I_1$,$I_2$);
$[f_{D_1}, f_{S_1}, f_{D_2}, f_{S_2}] =$ Classify_Dyn_Stat_Ref($f_1, f_2$)
for each $I_k$ and $k = 3$ to $N$ do
    $[f_1, f_k] \leftarrow$ Match($I_1$,$I_k$);
    $[f_{D_k}, f_{S_k}] =$ Classify_Dyn_Stat($f_{S_1}, f_{D_1}, f_k$)
    $F_k =$ComputeFundamentalMat($f_{S_1}, f_{S_k}$).
end for
for each dynamic feature $p_1^i \in f_1$ do
    $\hat{\ell}^i = \widehat{p}_1^i \times \widehat{p}_2^i$        {$\hat{\ell}^i$ is the image line}
    for each $p_k^i(t_k) \in S^i$ do
        $\widehat{\ell}_k^i = F_k \widehat{p}_k^i(t_k)$      {$\widehat{\ell}^i$ is the image line}
        $\widehat{p}_1^i(t_k) = \widehat{\ell}^i \times \widehat{\ell}_k^i$     {$p_1^i(t_k)$ is the intersection point}
        $\alpha_k \leftarrow$ ComputeAlpha($p_1^i, p_2^i, p_1^i(t_k)$)
    end for
    $\sigma_i \leftarrow$ sort($\{\alpha_k \mid k \in n_i\}$).
end for
```

The idea is to match all images with the first one, and then classify the Dynamic and Static matching points (Classify_Dyn_Stat_Ref) and compute the fundamental matrix (ComputeFundamentalMat) as we did in previous exercices.

To discriminate between Static and Dynamic is quite easy. Static features are easily detected by thresholding the Euclidean distance between matched features, and the dynamic features are the remaining ones.

The second part of the algorithm is the same as previous exercices to obtain the ordering of the images, including a computation and sorting of the alpha to automatically obtain the temporal ordering.

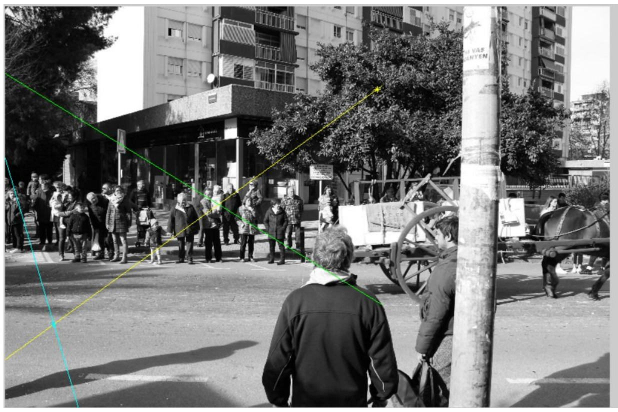


Image 18 shows an approximation of the solution, without classifying the static and dynamic points. Results are not as we would expect from the algorithm, but we should try with different sets of images and investigate further the algorithm to see where it could fail

Image 18: Photo-sequencing own images

Conclusions:

- The estimation of the Fundamental matrix by the 8-point algorithm is quite good, but it depends a lot of the matches of the points, so sometimes if the matches are not really accurate the resulting F is more differente of the real one.

- On the other hand, the previous instability in the estimation of the Fundamental matrix is resolved by the next algorithm computed, the robust normalized 8-point algorithm, in where RANSAC give us the accuracy that we had missing in the sift-matches.

- Using Sampson distance we have the advantage that the resulting geometric cost function only involves the parameters of the fundamental matrix.

- Using the learnt methods from Fundamental matrix estimation applied for a photo-sequencing algorithm showed us ways to apply the basics of the implemented methods for interesting applications. We saw some good results while also showing the limitations the algorithm can have.

- Optional was not finished because we were not able to fully understand how to compute alpha and the threshold to use when differentiating between static and dynamic points. Given more time we could experiment with more customized images and try different approaches to solve it.