Module: **M4. 3D Vision** **Final exam**
Date: February 21, 2019
Teachers: Antonio Agudo, Coloma Ballester, Josep Ramon Casas, Gloria Haro, Javier Ruiz
Time: 2h

- Books, lecture notes, calculators, phones, etc. are not allowed.
- All sheets of paper should have your name.
- Answer each problem in a separate sheet of paper.
- All results should be demonstrated or justified.

Problem 1 *0.75 Points*

(a) *(0.25p)* Let us denote by H a homography in the 2D projective space. What is the size of the matrix H and which kind of matrix is it? How many degrees of freedom does it have (justify it)?

(b) *(0.25p)* Enumerate the different situations where two images may be related by a homography.

(c) *(0.25p)* Mention two different problems in computer vision whose solution involves the estimation of a homography. Justify why the use of a homography is reasonable.

Problem 2 *0.75 Points*

Consider the problem of computing a 2D homography H between two image views of a plane object. Let $\mathbf{x}_i \in \mathbb{P}^2$ and $\mathbf{x}'_i \in \mathbb{P}^2$, $i = 1, \ldots, n$, be a set of points on the first image and the second image, respectively, such as, in pairs, they correspond: $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$, $\forall i = 1, \ldots, n$.

(a) *(0.25 points)* What is the minimum value of n? More precisely, how many corresponding points in general position do you need to compute H such that $\mathbf{x}'_i = H\mathbf{x}_i$, $\forall i = 1, \ldots, n$?
(Recall that general position means that no three points are collinear).

(b) *(0.5 points)* Describe the Normalized Direct Linear Transformation (Normalized-DLT) algorithm to compute H.

Problem 3 *0.75 Points*

What is the general form of a finite projective camera matrix P? Describe in detail its internal and external parameters.

Problem 4 *1.25 Points*

Camera calibration

(a) *(0.75p)* Explain the fundamental idea and the main steps of Zhang's calibration method studied in class (you can explain it in words, there is no need to include formulas but you can add them if you prefer).

(b) *(0.25p)* Define the image of the absolute conic. How many degrees of freedom does it have?

(c) *(0.25p)* Define the Perspective-n-Point (PnP) problem and specify which are unknown variables and the available/known data.

Problem 5 *2 Points*

Consider two images I and I' taken by the same camera (with intrinsic matrix $K = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$) and the estimated Fundamental matrix between them $F = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$. Answer the following questions:

a) Find the two epipolar lines in image I' corresponding to points $p_1 = (1, 0)$ and $p_2 = (10, 10)$ in image I.

b) Find the epipole e' in image I'.

c) Are the two images I and I' rectified?

d) If the correspondence point to p_1 is $p'_1 = (2, 1)$ and to p_2 is $p'_2 = (11, 10)$. Would you say any of them is an outlier?

e) Find the essential matrix E of the system.

f) What is the main difference between the fundamental matrix F and the Essential matrix E?

g) Would you be able to reconstruct the structure of the camera configuration (rotation, translation and scale) in this system?

f) What would you need to reconstruct everything (rotation, translation and scale)?

Problem 6 *1 Point*

Triangulation methods.

(a) *(0.75p)* Derive the expression of the matrix A that describes the linear system of equations in the linear triangulation methods.

(b) *(0.25p)* State the advantages and disadvantages of using a pair of views with a small angle between the visual rays.

Problem 7 *0.5 Points*

Assume that we have N different calibrated views of a 3D object and their corresponding depth maps d_i, $i = 1, ..., N$. Describe the main steps of the depth map fusion algorithm that reconstructs the object surface.

Problem 8 *0.5 Points*

Formulate the projection equation (indicating the corresponding size of every matrix) in terms of a measurement matrix, assuming a perspective camera for: 1) a rigid shape, 2) a non-rigid one. To compute shape and motion by rigid factorization, which rank do we have to enforce in every case and how can this be done?

Problem 9 *1.0 Points*

Let us assume a collection of I image frames with extrinsic parameters $\mathbf{P}_i$ with $i = \{1, \ldots, I\}$, where a 3D rigid object composed of P points is observed. Due to lack of visibility and outliers, a few points are not viewed in some frames. Particularly, the corresponding visibility vectors contain 12, 10, 14, and 14 components for every image, respectively. Assuming $P = 14$, for this particular case we always observe the points with smaller indexes $p = \{1, \ldots, P\}$. We want to simultaneously estimate 3D shape $\mathbf{X} \equiv [\mathbf{x}_1, \ldots, \mathbf{x}_P]$ (every $\mathbf{x}_p$ contains the 3D coordinates of the p-th point) and motion solely from 2D annotations by sparse bundle adjustment. For this toy example, represent the corresponding structure of the Jacobian matrix to code the problem and indicate the final matrix size. The intrinsic parameters of the camera can be assumed to be known. (0.7 points)

If the four image frames are a part of a monocular video, could we impose more constraints to sort out the problem? If so, describe them and represent this type of priors in the previous pattern. (0.3 points)

Problem 10 *0.75 Points*

3D data captured by a 3D sensor has a double nature (photometry + geometry) which is, let's say, more balanced than for 2D data captured by regular camera.

(a) Describe the two different natures of 3D SENSOR data

(b) The data we capture with a 3D sensor (3D vision), is it equivalent to the 3D geometry in a blueprint/CAD?

Problem 11 *0.75 Points*

(a) In J. Papon, et al (2018), "Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds," CVPR 2013, the following explanation introduces what is a superpixel approach:

> *Segmentation algorithms aim to group pixels in images into perceptually meaningful regions that conform to object boundaries. Graph-based approaches, such as Markov Random Field (MRF) and Conditional Random Field (CRF), have become popular, as they merge relational low-level context within the image with object-level class knowledge. The cost of solving pixel-level graphs led to the development of mid-level inference schemes which do not use pixels directly, but rather use groupings of pixels, known as superpixels, as the base level for nodes. Superpixels are formed by over-segmenting the image into small regions based on local low-level features, reducing the number of nodes that must be considered for inference. Due to their strong impact on the quality of the eventual segmentation, it is important that superpixels have certain characteristics. Of these, avoiding violating object boundaries is the most vital, as failing to do so will decrease the accuracy of classifiers used later - since they will be forced to consider pixels that belong to more than one class. Additionally, even if the classifier does manage a correct output, the final pixel level segmentation will necessarily contain errors. Another useful quality is regular distribution over the area being segmented, as this will produce a simpler graph for later steps.*

Note: Old image segmentation techniques using quadtrees may be seen as a predecessor of modern superpixel approaches.

Papon used supervoxels for the segmentation of point cloud data. Could you extend the explanation above to the concept of supervoxels? Derive your explanation from what you know about 2D segmentation and the nature of point cloud data, and use the following terms in your answer (over-segmentation, object boundaries, octree, even distribution, 26-adjacency).

(b) Supervoxel connectivity graphs can be used at the basis of a hierarchical tree segmentation structure in several levels, varying from coarse to fine. Such structures represent object segmentation at different scales of object-connectivity, from the super-voxel graph up to the scene level. Could you explain the advantage of using graphs based on over-segmentation in fine elements (super-voxels) for video object segmentation and tracking in stream data (RGBD + time).