Module: **M4. 3D Vision** **Final exam**
Date: February 20, 2020
Teachers: Antonio Agudo, Josep Ramon Casas, Pedro Cavestany, Gloria Haro, David Reixach, Javier Ruiz, Federico Sukno
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 *1.8 Points*

a) (0.1p) In 2D geometry, name two invariants under similarity, affine and projective transformations, in this order.

b) (0.2p) What makes the points at ℓ_∞ under affine transformation different from the points at ℓ_∞ under projective transformation?

c) (0.5p) Decompose a general 3D homography as a product of three transformations: from projective space to affine space, from affine space to similarity space and from similarity space to euclidean space. Tipify the submatrices that comprise these three transformations. In the context of 3D reconstruction, what is the meaning of each one of these submatrices?

d) (1p) Describe in your own words the Gold Standard Algorithm for 3D homography estimation.

a) Similarity: angles, absolute conic; affine: parallelism, ℓ_∞; projective: collinearity, concurrency.

b) Under affine transformation, points at ℓ_∞ are fixed, whereas under projective transformation they may move along ℓ_∞.

c) H can be decomposed as $H = H_{e\leftarrow s} H_{s\leftarrow a} H_{a\leftarrow p}$, with

$$H_{a\leftarrow p} = \begin{pmatrix} I & \vec{0} \\ \vec{v}^T & 1 \end{pmatrix}, \qquad H_{s\leftarrow a} = \begin{pmatrix} K & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}, \qquad H_{e\leftarrow s} = \begin{pmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix},$$

where R is a rotation, K upper-triangular matrix, $s > 0$, $\vec{t} \in \mathbb{R}^2$, $\vec{v}^T = (v_1, v_2)$.

In the context of 3D reconstruction, R and t define the pose of a camera, K is the calibration matrix and $\vec{v}$ is related to $\mathbf{\Pi}_\infty$ under projective transformation by the expression

$$\mathbf{\Pi}_{\infty,p} = \begin{pmatrix} -K^{-T}\vec{v} \\ 1 \end{pmatrix}$$

d) The Gold Standard Algorithm estimates the Maximum Likelihood Estimate $\hat{H}$ of the homography mapping between two images for $n > 4$ image point correspondences $\{x_i \leftrightarrow x_i'\}$. The MLE involves also solving for a set of subsidiary points $\{\hat{x}_i\}$, which minimise

$$\sum d(x_i, \hat{x}_i)^2 + d(x_i', \hat{x}_i')^2$$

where $\hat{x}_i' = \hat{H}\hat{x}_i$

The main steps of the algorithm are:

(i) **Initialisation**: Compute an initial estimate of $\hat{H}$ to provide a starting point for the geometric minimisation. Two good options are the linear normalised DLT algorithm or RANSAC

(ii) **Geometric minimisation**:

- Compute an initial estimate of the subsidiary variables $\hat{x}_i$ using the measured points $\{x_i\}$ (or the Sampson correction to these points given by $\delta_x = -J^T \left(JJ^T\right)^{-1} \epsilon$).
- Minimize the cost

$$\sum d(x_i, \hat{x}_i)^2 + d(x_i', \hat{x}_i')^2$$

over $\hat{H}$ and $\{\hat{x}_i\}$, by using the Levenberg-Marquardt algorithm over $2n + 9$ variables: $2n$ for the n 2D points $\{\hat{x}_i\}$, and 9 for the homography matrix $\hat{H}$.

Problem 2 *1.75 Points*

a) (0.25p) Explain how a conic is represented in projective geometry. What is the conic equation?

b) (0.25p) Explain the concept of the image of the absolute conic and how it is related to the internal parameters of the camera.

c) (0.25p) Enumerate three different practical applications that use the image of the absolute conic.

d) (0.75p) Consider the algebraic method for camera resectioning seen in class that assumes some 3D to 2D point correspondences are known. Explain the resectioning problem, derive the optimization problem we need to solve and justify the minimum amount of correspondences we need to solve it.

e) (0.25p) State the differences between the camera resectioning problem and the Perspective-n-Point problem in terms of unknown and available data.

a) A conic is represented by a 3×3 symmetric matrix. Let $\mathbf{x} \in \mathbb{P}^2$ be a 2D point in homogeneous coordinates, and C a matrix representing a conic, then the conic equation writes: $\mathbf{x}^T C \mathbf{x} = 0$

b) The image of the absolute conic, ω, is the projection to the image plane of the absolute conic, a conic which lives at infinity and is defined by the 3×3 identity matrix. Its relation to the matrix of internal of the camera, K is the following: $\omega = (KK^T)^{-1}$.

c) Some applications are:

- Image rectification (removal of projective distortion)
- Camera calibration
- Auto-calibration
- Measuring real angles from image projections

d) The goal of the camera resectioning is to estimate the camera projection matrix. It assumes some 3D to 2D correspondes are known, i.e. $\mathbf{X}_i \longleftrightarrow \mathbf{x}_i$ where $\mathbf{X}_i \in \mathbb{P}^3$, $\mathbf{x}_i \in \mathbb{P}^2$, $i = 1, ..., N$. It uses the fact that the 3D point $\mathbf{X}_i$ and its 2D projection $\mathbf{x}_i$ are related by the equation:

$$\lambda \mathbf{x}_i = P\mathbf{X}_i, \text{ where } \lambda \in \mathbb{R}, \lambda \neq 0 \tag{1}$$

We define $\mathbf{x}_i = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ and $P = \begin{pmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{pmatrix}_{3\times 4}$, develop the expression in (1) and obtain the following equations:

$$x\mathbf{p}_3^T\mathbf{X}_i - \mathbf{p}_1^T\mathbf{X}_i = 0$$
$$y\mathbf{p}_3^T\mathbf{X}_i - \mathbf{p}_2^T\mathbf{X}_i = 0$$

which may be written in algebraic form as:

$$\begin{pmatrix} \mathbf{0}^T & -w_i\mathbf{X}_i^T & y_i\mathbf{X}_i^T \\ w_i\mathbf{X}_i^T & \mathbf{0}^T & -x_i\mathbf{X}_i^T \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

Or, in short,

$$\mathbf{A}_i\mathbf{p} = \mathbf{0}$$

where $\mathbf{p}$ is the 12×1 vector of unknowns.

Each correspondence $\mathbf{X}_i \longleftrightarrow \mathbf{x}_i$ produces two equations for the 12 unknowns of $\mathbf{p}$: $\mathbf{A}_i\mathbf{p} = \mathbf{0}$. As P has 11 degrees of freedom ($3 \times 4 - 1$ scale factor), we need $N \geq 6$ correspondences $\mathbf{X}_i \longleftrightarrow \mathbf{x}_i$.

If $\mathbf{A}$ denotes the matrix obtained by

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \dots \\ \mathbf{A}_N \end{pmatrix}_{2N\times 12}$$

then the system of equations that considers the N correspondences is

$$\mathbf{A}\mathbf{p} = \mathbf{0} \in \mathbb{R}^{2N}$$

The vector $\mathbf{p}$ is in the null space of $\mathbf{A}$. In practice, the correspondences have noise, so the identity is not satisfied and we look for $\mathbf{p}$ by solving the following constrained optimization problem:

$$\begin{aligned} &\min \ \|\mathbf{A}\mathbf{p}\| \\ &\text{such that } \|\mathbf{p}\|_2 = 1 \end{aligned}$$

e) The Perspective-n-Point (PnP) is the problem of estimating the pose (external parameters: rotation and translation, 6 degrees of freedom) of a calibrated camera (known internal parameteres) given a set of n 3D points in the world and their corresponding 2D projections in the image. On the other hand, in camera resectioning both the internal and external parameters (11 degrees of freedom) are estimated from n 3D to 2D point correspondences.

Problem 3

1.80 Points

We want to compute the Fundamental Matrix F between two images I and I' of 100×100 pixels capturing the same secene from different viewpoints taken by the same camera. We compute two possible correspondences between the images as $p_1 = [2, 0]^T$, $p_1' = [2, 8]^T$ and $p_2 = [10, 1]^T$, $p_2' = [12, 2]^T$ and two epipolar lines in image U' as $l_1' \equiv 2x - 7y + 52 = 0$ and $l_2' \equiv 5x - 7y + 4 = 0$ (corresponding to point p_1 and p_2 respectively). Answer the following questions (consider the image coordinates origin at the bottom-left and positive going up and right):

a) Enumerate briefly the steps to compute matrix F using the 8-point algorithm.

b) Write the first two rows of matrix W that allows us to estimate the Fundamental Matrix F (expressed as a vector f) with a homogeneous system $Wf = 0$.

c) Propose a transformation to the pixel coordinates of p_1,p_2 and another for p'_1,p'_2 to reduce the numerical unstability of the 8-point algorithm.

d) Can any of the two correspondences be considered as an outlier?

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

f) Justify if the epipole is inside or outside of image I'.

g) Are the two images I and I' rectified? Why?

h) Would it be possible to compute the Essential Matrix E in this configuration?

a)
1. Create matrix W from normalized correspondences
2. Compute the SVD of matrix $W = UDV^T$
3. Create vector f from last column of V
4. Compose matrix F'
5. Compute the SVD of $F' = UDV^T$
6. Remove last singular value of D to create D'
7. Re-compute matrix $F = UD'V^T$
8. Un-normalize

b) $W = \begin{bmatrix} 2*2 & 0*2 & 2 & 2*8 & 0*8 & 8 & 2 & 0 & 1 \\ 10*12 & 1*12 & 12 & 10*2 & 1*2 & 30 & 10 & 1 & 1 \end{bmatrix}$

c) We need to normalize the coordinates, for instance with 0 mean and maximum 1.

$$H = \begin{bmatrix} 1/10 & 0 & -6/10 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1 \end{bmatrix} \quad H' = \begin{bmatrix} 1/12 & 0 & -7/12 \\ 0 & 1/8 & -5/8 \\ 0 & 0 & 1 \end{bmatrix}$$

d) Correspondence p_2,p'_2 is an outlier as it does not lies in the epipolar line l'_2

e) Epipole corresponds to the crossing of l'_1 and l'_2: $e' = [16, 12]^T$

f) Inside of image as is of size 100×100

g) No, for them to be rectified epipolar lines should be parallel to each other and X axes. Also epipoles should be at infinity.

h) No, we need 8 correspondences and intrinsic matrix K to compute E.

Problem 4 *1 Point*

a) (0.25p) Describe the triangulation problem, what are the unknowns and the available data.

b) (0.25p) How does the angle between visual rays affect the uncertainty in the estimation of the 3D point by triangulation given two views?

c) (0.5p) Enumerate the main steps of the view morphing technique proposed by Seitz and Dyer in 1996 in order to generate a new intermediate view given two images from two general points of views. Is it possible to generate a new view at any location of the virtual camera? Do the cameras need to be calibrated? Justify your answers.

a) Given two corresponding points $\mathbf{x}, \mathbf{x}' \in \mathbb{P}^2$, the problem is to estimate a point $\hat{\mathbf{X}} \in \mathbb{P}^3$ that satisfies $\hat{\mathbf{x}} = P\hat{\mathbf{X}}, \hat{\mathbf{x}}' = P'\hat{\mathbf{X}}$, for some points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ in the images near the corresponding points $\mathbf{x}, \mathbf{x}'$. The cameras are calibrated, so both P and P' are known.

b) 3D points are less precisely localized along the ray as the angle between rays becomes smaller.

c) The main steps of the algorithm are the following:

(i) Prewarp: apply a proper homography to each view so that views become parallel.

(ii) Morph: create a new intermediate image by linear interpolation of position and colors of corresponding points in the two available images.

(iii) Postwarp: apply a proper homography to the new image in order to get the desired orientation and zoom factor.

The position of the virtual camera is restricted to any point in the baseline of the two initial views. The cameras don't need to be calibrated because we can transform their corresponding images to parallel views with a stereo rectification method after having estimated their fundamental matrix.

Problem 5 *0.90 Points*

Regarding the 3D shape recovery from a set of images:

a) (0.2p) Using voxel-based methods the problem is generally ill-posed. Explain how this affects these methods by specifying the assumptions or limitations on **two** of them.

b) (0.5p) Formulate the factorization method for projective reconstruction from two or more views given by the paper: *Sturm, P., & Triggs, B. (1996, April). A factorization based algorithm for multi-image projective structure and motion. In European conference on computer vision (pp. 709-720). Springer, Berlin, Heidelberg.* **Detail each step of the algorithm.**

c) (0.2p) Explain the projective ambiguity inherent to image calibration and briefly describe a solution to this problem.

a) Shape from silhouette: Requires consistent silhouettes and does not recover concavities.

Voxel coloring: (Ambiguities are avoided by performing an spatial ordered-layer exploration). Then, the layer-order needs to be accessed from all the views, imposing the following constraint: **compatible scenes do not contain points within the convex hull of the camera centers**.

Space carving: Recovers the union of all possible photo-consistent shapes, not necessarily being the true 3D shape.

b) 1) Determine a feasible subset of scene points and cameras.

2) Normalize input data by applying the following transformation to each image i:

$$H_s^i = \begin{bmatrix} s^i & 0 & -s^i c_x^i \\ 0 & s^i & -s^i c_y^i \\ 0 & 0 & 1 \end{bmatrix}$$

where (with N being the total number of points in the image),

$$c^i = \begin{bmatrix} c_x^i \\ c_y^i \end{bmatrix}, \quad s^i = \frac{\sqrt{2}}{\frac{1}{N}\sum_n (\sqrt{Q^T Q})}, \quad Q = \begin{bmatrix} x_n \\ y_n \end{bmatrix} - c^i$$

3) Initialize scalar factors, $\lambda_j^i = 1 \quad \forall i, j$

4) Given

$$\Lambda = \begin{bmatrix} \lambda_1^1 & \lambda_2^1 & \cdots & \lambda_n^1 \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^m & \lambda_2^m & \cdots & \lambda_n^m \end{bmatrix},$$

reescale its rows and columns to have unit norm in an alternated manner. Stop when variation drops. (Usually two loops).

5) Build the measurement matrix

$$M = \begin{bmatrix} \lambda_1^1 x_1^1 & \lambda_2^1 x_2^1 & \cdots & \lambda_n^1 x_n^1 \\ \lambda_1^2 x_1^2 & \lambda_2^2 x_2^2 & \cdots & \lambda_n^2 x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^m x_1^m & \lambda_2^m x_2^m & \cdots & \lambda_n^m x_n^m \end{bmatrix},$$

where $x_j^i \quad \forall i, j$ refer to the normalized input data.

6) Determine the SVD of $M = UDV^T$.

7) Performing a rank-4 subselection:

$$\begin{bmatrix} P^1 \\ P^2 \\ \cdots \\ P^m \end{bmatrix} = UD_4, \quad \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_1 & \cdots & \mathbf{X}_n \end{bmatrix} = V_4^T$$

8) Stop if $\sum_i \sum_j d(x_j^i, P^i \mathbf{X}_j)^2$ converges. Otherwise set $\lambda_j^i = (P^i \mathbf{X}_j)_3$ and go to step 4.

9) Unnormalize the camera matrices $(H_s^i)^{-1} P^i$.

10) (Triangulate and resection the non-nucleus scene points and cameras).

c) It is possible to apply a projective transformation H such that $x = (PH)^{-1}(H\mathbf{X})$. A solution to the projective ambiguity problem is the stratified reconstruction, allthought it requires more information either about the scene, the motion or the camera calibration. Its main steps are:

1) Estimate a projective reconstruction

2) (Upgrade the previous estimate to an affine reconstruction)

3) Upgrade the previous estimate to a metric reconstruction

Problem 6 *0.45 Points*

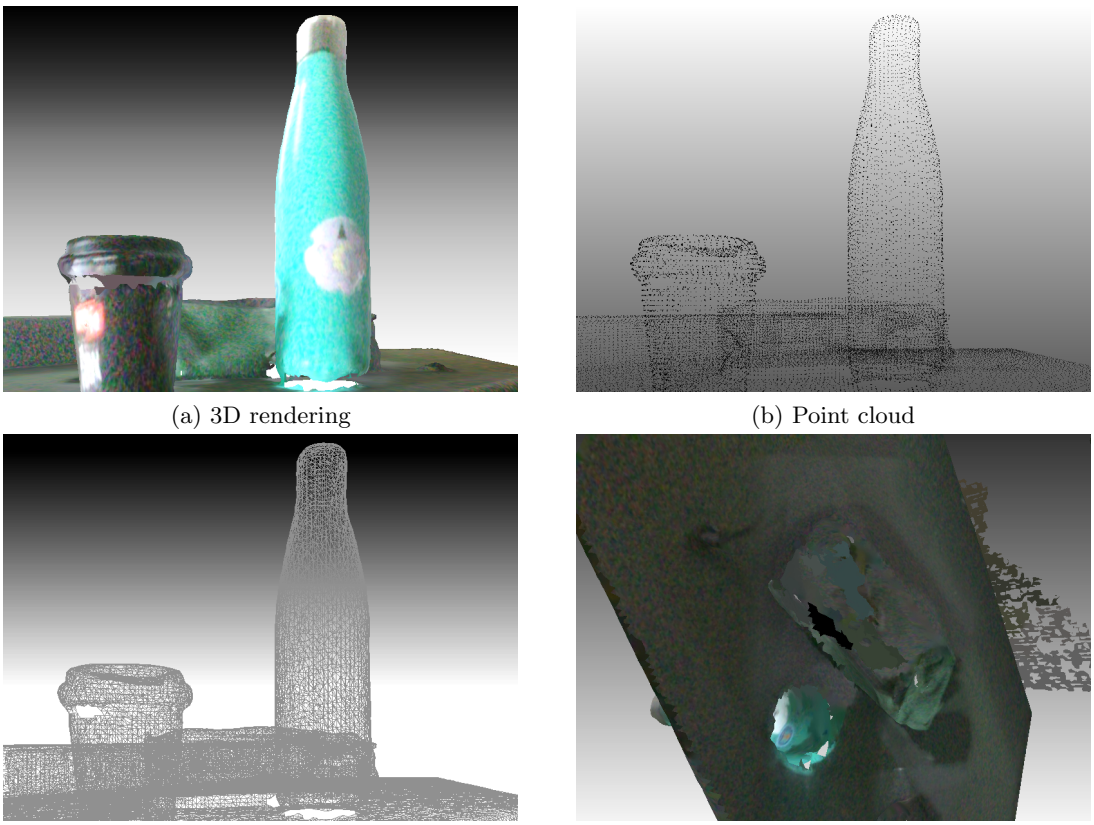
Formulate the structure-from-motion problem from 2D point tracks in a monocular video (indicating the corresponding size of every matrix) in terms of a measurement matrix $\mathbf{M}$, assuming an orthographic camera for: 1) a rigid shape, 2) a non-rigid one. To compute shape and motion by factorization, which rank do we have to enforce in every case and how can this be done?

Considering m the number of images and n the number of points to be observed, the projection system can be written as $\mathbf{M} = \mathbf{PX}$:

In both cases: $\mathbf{M}$ is a $2m \times n$ measurement matrix, $\mathbf{P}$ and $\mathbf{X}$ are a $2m \times 3$ motion and $3 \times n$ shape matrix components, respectively. We will use a SVD factorization, imposing a rank-3 and 3K decomposition, respectively, where K is the rank of a linear subspace.

Problem 7 *1 Point*

We captured a 360º 3D scene using an RGBD sensor and turning the sensor around some objects laying on a table. The double nature (photometry+geometry) of the captured data is shown in the images of Fig. 1.



(a) 3D rendering (b) Point cloud

(c) Mesh (d) Bottom view of the 3D scene

Figure 1: RGBD capture of a 3D scene

a) Images in Fig. 1 (a), (c), and (d) show some parts of the geometry not being captured. Which are those parts? And, what may be the reasons why they have not been correctly captured?

b) Why should we avoid having any motion in the objects of a scene captured this way?

c) Does this mean that we cannot capture dynamic 360º 3D scenes (i.e. RGBXYZT)? How can we capture them?

d) Even if we avoid the problems mentioned in the previous questions to capture a 3D dynamic scene, what would we always miss from a dynamic scene capture with an RGBD sensor?

e) If we want to get the "whole geometry" of a dynamic scene, what solution would you propose?

a) The 3D reconstruction missed some portions of the geometry in the lower part of the bottle. This is most probably due to an excess of reflectance of the convexity in the bottom of the bottle, which reflects away the light of the active sensor, which the sensor cannot recover in order to obtain the depth of these points.

b) We should avoid motion because we reconstruct the 3D shapes by stitching together a sequence of RGBD frames from a sequence of points of view captured 360º around the scene. Any motion during the capture sequence will result in a deformation of the reconstructed shape.

c) No. We may capture a dynamic scene in 360º geometry using multiple sensors placed around the scene, which would capture all the frames (points of view) simultaneously. If these sensors are able to capture video, then we may build a dynamic 360º scene by performing 3D reconstruction at each time instant with the frames from all the sensors captured at that time.

d) RGBD sensors usually assume one (or several) points of view from which 3D geometry is reconstructed, but this leaves part of the scene geometry unavailable due to occlusions and auto-occlusions from these points of view. Occlusions will hide parts of the scene from the RGBD

sensor (this is the case of the portion of the table in Fig. 1 occluded by the bottom of the objects; also for the inner volumes of the opaque objects in the scene)

e) We would need a 3D scan system that works through opaque surfaces (e.g. MRI, TMI, PET...)

Problem 8 *0.40 Points*

Organized data structures may help in vision analysis tasks for pointclouds. Graphs and trees have been proposed to analyze (segment, detect, classify) point clouds.

a) Explain how adjacency, hierachy, captured primitives (visual or geometric features) and visual boundaries play a role for analysis tasks within the edge and node elements of graphs and trees.

b) What is the reason why graph and tree structures, specially for point clouds, are expected to perform better than raw data?

a) The graph representation simplifies the sparse and inhomogeneous input data by grouping homogeneous points on of the point cloud into nodes, while preserving the boundary information. Such representations are appropriate for building a hierarchical description and for performing scene analysis and segmentation. Supervoxel connectivity graphs, for instance, add adjacency relationships and homogenize the raw data in pointclouds. This can be used at the basis of a hierarchical tree segmentation structure in several levels, varying from coarse to fine. Such structures can represent, for instance, object segmentation at different scales of object-connectivity, from the super-voxel graph up to the scene level. We can use Graphs or Trees to define meaningful primitives for processing, detection and classification in the nodes, whereas edges will represent relationships and hierarchies. Data pooling procedures are then able to enrich edges and nodes with features and/or classes. Additionally, trees and graphs use to be well suited for parallel computing.

b) Pointclouds are a relatively sparse kind of data, unstructured and based on simple and numerous primitives (the points in the cloud). This is why graph structures have become an essential tool for modeling point clouds obtained from RGB-D sensors.

Problem 9 *0.90 Points*

Let $\mathcal{G} = \{V, E\}$ be a graph with vertices V and edges E. To facilitate the implementaiton of convolutions on the graph, we can use the spectrum of the graph, which can be obtained by eigendecomposition of the Laplacian operator Δ such that $\Delta = \Phi^T \Lambda \Phi$. Assuming that we wish to convolve a function $f : V \to \mathbb{R}$ (e.g. a scalar function over the graph vertices) with a filter g (also defined on the graph domain).

a) Indicate how can we map function f onto the spectral domain (i.e. in analogy to the Fourier Transform).

b) Idem (a) for the inverse mapping (back from the spectrum to the graph domain, in analogy to the inverse Fourier Transform).

c) Provide the mathematical expression to compute $f * g$ (convolution) using the spectral domain.

d) Indicate one dissadvantage of this straight-forward use of the spectral domain as building block for a graph convolutional network.

a) We can map onto the spectral domain using the (transposed) eigenvector matrix: $F = \Phi^T f$

b) We can map back onto the graph domain using the eigenvector matrix: $f = \Phi F$.

c) $f * g = \Phi\big((\Phi^T f)(\Phi^T g)\big)$

d) Firstly, this procedure is very computationally expensive: $O(n^2)$ to transform between domains and $O(n^3)$ for the eigendecomposition (n is the number of vertices). Secondly, recalling that the filters are learned directly on the spectral domain, there is no guarantee that they end up being spatially localized (it will be considered OK if students can name at least one dissadvantage).