Master in Computer Vision Barcelona

UAB UOC UPC upf.

Module: **M6. 3D Vision** **Final exam**
Date: May 4, 2017
Teachers: 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 *1.25 Points*

(a) *(0.5 points)* What is the general form of a matrix that represents a similarity transformation in the 2D projective space? How many degrees of freedom does it have?

A similarity transformation is represented by a 3×3 non-singular matrix of the following form:

$$H_s = \begin{pmatrix} s\cos(\theta) & -s\sin(\theta) & t_x \\ s\sin(\theta) & s\cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix}.$$

where $s \in \mathbb{R}$, $s \neq 0$ is a scaling factor, θ is a rotation angle, and $\mathbf{t} = (t_x, t_y)^T$ is a translation vector.

It has 4 degrees of freedom: scaling factor s, rotation angle θ, and translation vector $\mathbf{t} = (t_x, t_y)^T$.

(b) *(0.5 points)* Show how a similarity transformation acts on conics.

Let H be a 2D similarity transformation .

A point $\mathbf{x} \in \mathbb{P}^2$ is transformed by $\mathbf{x}' = H\mathbf{x}$, then

$$\mathbf{x} = H^{-1}\mathbf{x}'. \qquad (1)$$

On the other hand, the conic equation is

$$\mathbf{x}^T C \mathbf{x} = 0, \qquad (2)$$

where C is a 3×3 symmetric matrix.

Using (1) in (2) we get:

$$\mathbf{x}'^T H^{-T} C H^{-1} \mathbf{x}' = 0$$

and identifying this equation with the conic equation on the transformed points $\mathbf{x}'^T C' \mathbf{x}' = 0$ and transformed conic C' we get:

$$C' = H^{-T} C H^{-1}.$$

(c) *(0.25 points)* Which are the geometric invariants for a similarity transformation?

Angles, ratio of lengths, ratio of areas.

Problem 2 *0.75 Points*

Consider an image representing a plane in the 3D world. Assume that the image has been affinely rectified so that the line at infinity in the image is given by $\ell_\infty = (0, 0, 1)$. Explain the method of metric rectification via orthogonal lines.

We would like to compute a planar projective transformation that metrically rectifies our image. In general, we know that if H is any planar projective transformation, H can be written as $\begin{pmatrix} A & \vec{t} \\ \vec{v}^T & 1 \end{pmatrix}$ and, moreover, H can be decomposed as $H = H_{e\leftarrow s} H_{s\leftarrow a} H_{a\leftarrow p}$, where each matrix is of the form

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

and $A = sRK + \vec{t}\vec{v}^T$, R rotation, K upper-triangular matrix. In our case, our image has been affinely rectified. Therefore, we only need to compute a planar projective transformation of the form $H_{s\leftarrow a}$, that is, mapping affine coordinates to metric ones.

To compute $H_{s\leftarrow a}$, we use that we know that, if $\mathbf{l} = (l_1, l_2, l_3)$ and $\mathbf{m} = (m_1, m_2, m_3)$ are the image of two lines that are orthogonal in the world, then $\mathbf{l}^T M \mathbf{m} = 0$, where $M = \begin{pmatrix} S & \vec{0} \\ \vec{0}^T & 0 \end{pmatrix}$ and where S is a 2×2 symmetric and positive-definite matrix of the form $S = \tilde{A}\tilde{A}^T = \begin{pmatrix} s_1 & s_2 \\ s_2 & s_3 \end{pmatrix}$, with $\det \tilde{A} \neq 0$.

Then, taking such a $\mathbf{l} = (l_1, l_2, l_3)$ and $\mathbf{m} = (m_1, m_2, m_3)$ and imposing $\mathbf{l}^T M \mathbf{m} = 0$, we obtain

$$(l_1 m_1, l_1 m_2 + l_2 m_1, l_2 m_2)\vec{s} = 0,$$

where $\vec{s}^T = (s_1, s_2, s_3)^T$ is the vector with the entries of S. If now we take an extra pair of orthogonal lines (that is, two pairs in total), we will be able to compute S. Indeed, the image of two such orthogonal pairs provide two equations and they permit to compute S solving the homogeneous system of two equations with three unknowns (s_1, s_2, s_3).

The Cholesky decomposition of S allows to compute an upper triangular matrix K such that $S = KK^T$, with $\det K \neq 0$. Then, it suffices to define $H_{a\leftarrow s} = \begin{pmatrix} K & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}$ and

$$H_{\mathrm{a}\leftarrow\mathrm{s}}^{-1} = H_{\mathrm{s}\leftarrow\mathrm{a}} = \begin{pmatrix} K^{-1} & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}$$

The rectified image can be defined by $u_{\text{metrect}}(\vec{x}_s) = u_{\text{affrect}}([H_{\mathrm{a}\leftarrow\mathrm{s}}\mathbf{x}_s])$, where $\mathbf{x}_s = (\vec{x}_s, 1)$ and $[(p_1, p_2, p_3)] = (p_1/p_3, p_2/p_3)$.

Problem 3 *0.25 Points*

Consider the problem of computing (with the DLT algorithm, for instance) a 2D homography H between two image views of a plane objet. 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: $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$, $\forall i = 1, \ldots, n$.

What is the minimum value of n?

The minimum number n of corresponding points in general position is four because the 2D homography H has nine entries to compute, minus one scale factor. That is, eight unknowns. On the other hand, each pair of corresponding points provides two equations.

Problem 4 *0.75 Points*

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

P decomposes in $P = K[R|\mathbf{t}]$, where K and R are 3×3 matrices and $\mathbf{t}$ is a 3×1 vector. K is the calibration matrix containing the internals parameters, and R, $\mathbf{t}$ represent the external parameters of the camera. In particular,

$$K = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

and R and $\mathbf{t}$ give the position and orientation of the camera in the world coordinate system. See lecture3.pdf for the description of the internal parameters $x_0, y_0, \alpha_x, \alpha_y, s$.

Problem 5 *0.75 Points*

Explain (briefly) the camera calibration method (of Zhang) using a planar pattern and several images of it.

The summary of the answer is in page 18 of the course notes “lecture 4.pdf”. The details are in pages 8-16.

Problem 6 *2 points*

Consider two images I and I' of 80×80 pixels capturing the same scene with the same camera from different viewpoints and two epipolar lines in image I such as: $\ell_1 \equiv 2x - 4y = 0$ and $\ell_2 \equiv x + 10y - 400 = 0$ Consider the coordinates origin on the bottom right of the images (positive going up and right). Answer the following questions:

a) What are the pixel coordinates of the epipole in image I?

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

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

d) Compute the last row of the fundamental matrix F if the epipolar line ℓ_1 corresponds to the point $x' = (0, 0)^T$ in image I'.

Let's assume now that the fundamental matrix F between the two images I and I' is known.

e) Would you be able to reconstruct the structure of the camera configuration (rotation, translation and scale)? Why?

f) If the answer to the previous is negative, What extra information would you need to obtain the structure of the camera configuration? If positive, what steps would you do to obtain the structure?

a) Epipole corresponds to the crossing of ℓ_1 and ℓ_2 $e = (200/3, 100/3)^T$.

b) Outside as image is of size 80×80.

c) No, for them to be rectified epipolar lines should be parallel to each other and x axes. Also epipole should be at infinity.

d) We know that therefore last row should be equal to ℓ_1 in homogeneous coordinates: $(2, -4, 0)^T$

e) No, if the camera is uncalibrated then F defines the structure up to a projective transformation.

f) We would need to calibrate the camera and obtain the intrinsic parameters to obtain R and T (normalized). The scale is not possible to obtain unless there is an object in the scene with known length.

Problem 7 *1 Point*

Depth and disparity estimation.

(a) *(0.5 points)* Describe how the disparity is estimated by the local and global methods and what is the main difference between them.

Local methods for stereo matching search for the right disparity by sliding a window along the same line in the right image and comparing its content to that of the reference window in the left image. The estimated disparity is given by the offset position ot the window wich gives a minimum matching cost (or a maximum similarity). In other words, the disparity is found by minimizing a matching cost (defined on a window) for every pixel.
Global methods are based on an energy minimization; the energy has a data term that measures the matching score and a regularity term that imposes certain type of regularity on the solution. The main difference between them is that global methods include a regularization in the estimation process.

(b) *(0.5 points)* Describe the main steps of the plane sweep method.

For each sampling depth:

- Compute the fronto-parallel plane at the corresponding depth.
- Compute the image to image homography given that plane.
- Warp the second image according to the homography.
- Compute the matching score.
- For each pixel keep the depth with best score.

Problem 8 *1 Point*

Factorization method.

(a) *(0.25 points)* Describe what are the unknowns and the available data in the factorization method.

The data available are a set of images (views) and a set of point correspondences across the different images; these points correspond to the 2D projections of a set of unknown 3D points. The goal is to find the 3D points that project to the given image points in the different images and the camera projection matrices associated to each view.

(b) *(0.5 points)* What is the algebraic system of equations we form in the factorization method? Describe the different elements involved and the size of the matrices.

Let us denote a 3D point in homogeneous coordinates by $\mathbf{X}_j$, we have n 3D points, $j = 1, ...n$. On the other hand we have m cameras with projection matrices P^i, $i = 1, ...m$. We get the 2D homogeneous coordinates of the projected points $\mathbf{x}_j^i$ in the different views thanks to the projective equations:

$$\mathbf{x}_j^i \equiv P^i \mathbf{X}_j \qquad \longrightarrow \qquad \lambda_j^i \mathbf{x}_j^i = P^i \mathbf{X}_j$$

where λ_j^i are unknown scalar factors, called *projective depths*. We can collect all projective eq's into a matrix equation:

$$\begin{bmatrix} \lambda_1^1 \mathbf{x}_1^1 & \lambda_2^1 \mathbf{x}_2^1 & ... & \lambda_n^1 \mathbf{x}_n^1 \\ \lambda_1^2 \mathbf{x}_1^2 & \lambda_2^2 \mathbf{x}_2^2 & ... & \lambda_n^2 \mathbf{x}_n^2 \\ ... & ... & ... & ... \\ \lambda_1^m \mathbf{x}_1^m & \lambda_2^m \mathbf{x}_2^m & ... & \lambda_n^m \mathbf{x}_n^m \end{bmatrix} = M = \begin{bmatrix} P^1 \\ P^2 \\ ... \\ P^m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & ... & \mathbf{X}_n \end{bmatrix}$$

We have

$$\underbrace{M}_{3m \times n} = \underbrace{\begin{bmatrix} P^1 \\ P^2 \\ ... \\ P^m \end{bmatrix}}_{3m \times 4} \underbrace{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & ... & \mathbf{X}_n \end{bmatrix}}_{4 \times n}.$$

(c) *(0.25 points)* How do we use the prvious system of equations in order to find an estimate of the solution?

M has at most rank 4 and a possible solution for the camera projection matrices and the 3D points is obtained by the SVD decomposition of M, i.e. $M = UDV^T$, thus

$$\begin{bmatrix} P^1 \\ P^2 \\ \dots \\ P^m \end{bmatrix} = UD_4 \qquad \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_n \end{bmatrix} = V_4^T$$

where D_4 is the $n \times 4$ submatrix of D, and V_4^T is the $4 \times n$ submatrix of V^T.

Problem 9 *0.75 Points*

In the metric reconstruction step of a stratified structure from motion approach the essence is to find an estimate of the image of the absolute conic.

(a) *(0.25 points)* How does the image of the absolute conic relates to the matrix of i nternal parameters of the camera?

$\omega = K^{-T}K^{-1}$, where ω is the image of the absolute conic and K is the matrix of internal parameters.

(b) *(0.5 points)* What are the different constraints on the image of the absolute conic we use to estimate it with the method studied in the course? How many constraints do we need? Why?

We need 5 constraints since ω is a 3×3 symmetric matrix that we need to estimate up to a scale factor.

We use two different types of constraints:

- Constraints coming from scene orthogonality.
  If $\mathbf{v}_1$ and $\mathbf{v}_2$ is a pair of vanishing points arising from orthogonal scene lines, then we have a linear constraint on ω:
  $$\mathbf{v}_1^T \omega \mathbf{v}_2 = 0.$$
  We use three pairs of different orthogonal lines and thus we have three equations (constraints).
- Constraints coming from known internal parameters.
  We assume the camera has zero skew, then $\omega_{12} = 0$.
  We also assume pixels are square, then: $\omega_{11} = \omega_{22}$.

Problem 10 *0.5 Points*

Depth sensors

(a) State the advantage of active depth sensors over depth measurement with passive stereo sensors. What are the main features of commercial depth sensors regarding range, noise, and resolution compared to pre-existing industrial grade scanners (such as FARO)? What are the advantages and disadvantages of both?
Why have commercial depth sensors experienced such a success since 2010?

Active depth sensors project their own light (usually in well-defined patterns) into the scene, and do not depend on scene illumination. This eases correspondence matching for the projector-camera stereo pair.
Commercial depth sensors use to have reduced range, larger noise factors and reduced resolution

compared to industrial grade scanners. Commercial depth sensors also work like matricial systems (like cameras, without point or line scanning) and this does not restrict their use to static scenes (can capture video in moving scenes). Furthermore, they are far cheaper than industrial scanners. This explains the success of commercial depth sensors, particularly for research purposes.

(b) RGBD data is usually captured as color+depth, but it can also be represented in terms of point clouds (RGBXYZ).

Explicit the operation performed for the conversion from RGBD to RGBXYZ. Which parameters are required for the conversion? State advantages and disadvantages of each format.

Back-projection, i.e. computing the line from the optical center, through the pixel and finding the XYZ 3D point on this line at distance D from the sensor. For this, we need camera intrinsics (focal distance) and camera distortion parameters if we want to correct for distortion.
RGBD format is more compact: 3+2 bytes per pixel for RGBD vs 3+3x4 (or 3+3x8) for RGBXYZ, and makes neighbor finding easier, whereas RGBXYZ can be readily rendered and manipulated (rotated) as a 3D graphic.

Problem 11 *0.5 Points*

3D data and Point clouds

(a) Point clouds can be ”unorganized”, what does this mean? What is the disadvantage of unorganized point clouds for processing purposes? Can an unorganized point cloud be converted into RGBD?

Unorganized point clouds are datasets representing a non-regular sampling of 3D space. Unorganized point clouds are not projectable, i.e. there is no correlation according to a pinhole camera model between the (u,v) index of a projected pixel, like for organized point clouds, and the actual 3D values.

(b) Explain the main problem for processing unorganized point clouds for local feature extraction or smoothing (i.e. filtering based on spatial or temporal coherence)

Neighborhood operations, such as those needed for local feature extraction or smoothing, require k-d tree search operations, for finding neighboring points in 3D. A k-d tree, or k-dimensional tree, is a data structure for organizing some number of points in a space with k dimensions. It is a binary search tree with other constraints imposed on it. K-d trees are very useful for range and nearest neighbor searches, but still costly in terms of computation compared to finding nearest neighbors in organized/indexed datasets such as 2D images or voxelized 3D volumes.

Problem 12 *0.5 Points*

Depth scans and meshing

(a) What are the principles of a good point feature representation for point cloud data according to R.B. Rusu?

A good point feature representation distinguishes itself from a bad one, by being able to capture the same local surface characteristics in the presence of:

- rigid transformations - 3D rotations and translations should not influence the resultant feature vector F estimation.
- varying sampling density - a local surface patch sampled more or less densely should have the same feature vector signature.

- noise - the point feature representation must retain the same or very similar values in the presence of mild noise in the data.

(b) How can we compute a tangent plane on a local neighborhood of the point cloud?

Estimating a plane tangent to the surface (i.e. the local surface normal) is usually posed as a least-square plane fitting estimation problem, and reduced to an analysis of eigenvectors and eigenvalues (or PCA Principal Component Analysis) of a covariance matrix created from the N-nearest neighbors of the query point (the point for which we want to estimate the surface normal).