Master in Computer Vision
Barcelona

UAB UOC UPC upf.

Module: **M4. 3D Vision** **Final exam**
Date: February 15, 2018
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*

Consider an affine transformation in the 2D projective space.

(a) *(0.25p)* Write the general form of a matrix that represents it. How many degrees of freedom does it have?

$$H_a = \begin{pmatrix} A & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix},$$

where A is a non-singular 2×2 matrix and $\vec{t}$, $\vec{0}$ are 2×1 vectors. It has 6 degrees of freedom which are the 6 elements of A and $\vec{t}$.

(b) *(0.3p)* Show how to decompose it in fundamental transformations such as rotations, scalings and translations. Specify the proper order of these transformations.

The SVD decomposition of A gives:

$$A = UDV^T = UV^T(VDV^T) = R_\theta R_{-\phi} D R_\phi$$

where D is a diagonal matrix that represent an anisotropic change of scale, R_θ and R_ϕ rotations of angles θ and ϕ respectively. On the other hand, $\vec{t}$ is a translation vector.

Then we can decompose the transformation H_a as the composition of the following fundamental transformations:

$$H_a = \underbrace{\begin{pmatrix} I & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix}}_{\text{translation}} \underbrace{\begin{pmatrix} R_\theta & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}}_{\text{rotation}} \underbrace{\begin{pmatrix} R_{-\phi} & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}}_{\text{rotation}} \underbrace{\begin{pmatrix} D & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}}_{\text{anisotropic scaling}} \underbrace{\begin{pmatrix} R_\phi & \vec{0} \\ \vec{0}^T & 1 \end{pmatrix}}_{\text{rotation}}$$

(c) *(0.2p)* Enumerate the different invariant properties that are preserved after this type of transformation.

Parallel lines, ratio of parallel lengths, ratio of two areas, line at infinity.

Problem 2 *0.75 Points*

Consider the task of removing projective distortions of flat objects on an image. To this goal, consider an image containing a plane in the 3D world.

(a) *(0.25p)* What is the vanishing line of that plane? What is a vanishing point of a world line?

The vanishing line is the image (the projection) of the line at infinity onto the image plane. A vanishing point of a world line is the intersection of that line onto the image plane with the line at infinity, onto the image plane.

(b) *(0.5p)* Explain the method of affine rectification of the image via the vanishing line.

First, we compute the line at infinity ℓ (the vanishing line) on the image which has a projective distortion. To this goal, we take two sets of two parallel lines, be it $\ell^a, \ell^b, \ell^c, \ell^d$, with ℓ^a parallel to ℓ^b, and ℓ^c parallel to ℓ^d.

Now, we compute the vanishing point of each pair of parallel lines, $v^{ab} = \ell^a \times \ell^b$ and $v^{cd} = \ell^c \times \ell^d$.

Then, from these two points, which should be on the vanishing line, compute the vanishing line as $\ell = v^{ab} \times v^{cd}$. Let us denote this line as $\ell = (l_1, l_2, l_3)^T$.

Finally, the projective transformation (let us denote it by $H_{a\leftarrow p}$) of $\mathbb{P}^2$ given by $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{pmatrix}$ affinely rectifies the image. Other possible transformations are

$$H_{a\leftarrow p} = H_a \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{pmatrix},$$

where H_a is any affine transformation. We can affinely rectify the image u which contains the line ℓ by defining

$$u_{\text{affrect}}(\vec{x}) = u([H_{a\leftarrow p}^{-1}\mathbf{x}]),$$

where $\mathbf{x} = (\vec{x}, 1)^T$ and $[\cdot]$ denotes$[(p_1, p_2, p_3)] = (p_1/p_3, p_2/p_3)$.

Problem 3 *0.75 Points*

(a) *(0.25p)* What is or how is defined the plane at infinity $\mathbf{\Pi}_\infty$ in the 3D projective space $\mathbb{P}^3$?

The plane at infinity $\mathbf{\Pi}_\infty$ is defined by the points $\{\mathbf{X} = (x_1, x_2, x_3, 0)^T : (x_1, x_2, x_3)^T \in \mathbb{R}^3 \setminus \{(0,0,0)\}\}$ modulus the equivalence relation $\mathbf{X} \equiv \mathbf{X}'$ if there exists $\lambda \neq 0$ such that $\mathbf{X} = \lambda \mathbf{X}'$.

Those points $(x_1, x_2, x_3, 0) \in \mathbf{\Pi}_\infty$ are called ideal points or points at infinity. They form a plane:

$$x_4 = 0$$

with equation $< (0,0,0,1)^T, (x_1, x_2, x_3, x_4)^T >= 0$, or $(0,0,0,1)(x_1, x_2, x_3, x_4)^T = 0$, or $\mathbf{\Pi}_\infty^T \mathbf{X} = 0$, where $\mathbf{\Pi}_\infty = (0,0,0,1)^T$.

(b) *(0.5p)* What is the general form of a finite projective camera matrix P? Describe its internal and external parameters.

A general projective camera P maps world points $\mathbf{X} \in \mathbb{P}^3$ to image points $\mathbf{x} \in \mathbb{P}^2$ according to $\mathbf{x} = P\mathbf{X}$. A general finite projective camera P can be decomposed as $P = K[R|\mathbf{t}]$, where K and R are 3×3 matrices and $\mathbf{t}$ is a 3×1 vector. K is the calibration matrix containing the internal parameters, and R, $\mathbf{t}$ represent the external parameters of the camera. In particular,

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

where $x_0 = m_x p_x, y_0 = m_y p_y$, with f the focal length of the camera, (p_x, p_y) the coordinates of the principal point in a reference system in the image where the origin of coordinates in the image plane is situated at a corner of the image plane, $\alpha_x = m_x f, \alpha_y = m_y f$ are parameters allowing the possibility of having non-square pixels for the image coordinates (of units m_x, m_y) and s is the skew parameter.

Finally R and $\mathbf{t}$ represent the external parameters of the camera and give the position and orientation of the camera in the world coordinate system (in particular, R and $\mathbf{t}$ relate the inhomogeneous 3-vector $\widetilde{\mathbf{X}}$ representing the coordinates of a point in the world reference system with the same point in the camera coordinate frame, $\widetilde{\mathbf{X}}_{\text{cam}}$, that is, $\widetilde{\mathbf{X}}_{\text{cam}} = R(\widetilde{\mathbf{X}}_{\text{w}} - \widetilde{\mathbf{C}})$, and $\mathbf{t} = -R\widetilde{\mathbf{C}}$, where $\widetilde{\mathbf{C}}$ are the inhomogeneous coordinates of the camera centre $\mathbf{C}$ in the world coordinate frame.

Problem 4 *1.25 Points*

Calibration with a planar pattern.

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

The image of the absolute conic is the projection of the absolute conic to the image plane. It has the following expression $\omega = (KK^T)^{-1} = K^{-T}K^{-1}$ where K is the matrix of the internal parameters of the camera. It is a symmetric matrix and thus has 6 different elements that reduce to 5 degrees of freedom because of the scaling invariance.

(b) *(0.6p)* Derive the equations that relate the image of the absolute conic and the columns of the homography that relates the planar pattern with an image of it taken from a certain point of view.

First we consider the relation, through a planar homography, of the coordinates of points on a flat object and their projection to the image plane

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [\mathbf{h}_1\, \mathbf{h}_2\, \mathbf{h}_3] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Now we consider that the flat object is located at plane $Z = 0$ and project it to the image

$$\begin{bmatrix} \alpha u \\ \alpha v \\ \alpha \end{bmatrix} = K \Big[\underbrace{\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{r}_3}_{R}\ \mathbf{t}\Big] \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = [K\mathbf{r}_1\, K\mathbf{r}_2\, K\mathbf{t}] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

We then deduce:

$$[K\mathbf{r}_1\, K\mathbf{r}_2\, K\mathbf{t}] \sim [\mathbf{h}_1\, \mathbf{h}_2\, \mathbf{h}_3]\,. \tag{1}$$

Since $\mathbf{r}_1$ and $\mathbf{r}_2$ are columns of an orthogonal matrix they satisfy

$$\left.\begin{array}{c} \mathbf{r}_1^T \mathbf{r}_2 = 0 \\ \mathbf{r}_1^T \mathbf{r}_1 = \mathbf{r}_2^T \mathbf{r}_2 = 1 \end{array}\right\}. \tag{2}$$

Isolating the external parameters of the camera in (1)

$$[\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{t}] \sim \left[K^{-1}\mathbf{h}_1\, K^{-1}\mathbf{h}_2\, K^{-1}\mathbf{h}_3\right].$$

Applying (2) to the previous expression

$$\left.\begin{array}{c} \mathbf{h}_1^T K^{-T} K^{-1} \mathbf{h}_2 = 0 \\ \mathbf{h}_1^T K^{-T} K^{-1} \mathbf{h}_1 = \mathbf{h}_2^T K^{-T} K^{-1} \mathbf{h}_2 \end{array}\right\},$$

where we identify the image of the absolute conic and rewrite it as

$$\left.\begin{array}{c} \mathbf{h}_1^T\omega\,\mathbf{h}_2 = 0 \\ \mathbf{h}_1^T\omega\,\mathbf{h}_1 = \mathbf{h}_2^T\omega\,\mathbf{h}_2 \end{array}\right\}.$$

(c) *(0.2p)* Assume the camera parameters are completely unkonwn. How many views of the planar pattern do we need to calibrate the camera? Justify your answer.

Since the image of the absolute conic has 5 unknowns and every view provides two equations we need three different views.

(d) *(0.2p)* Under which hypothesis is possible to calibrate the camera just by using a single view. Justify your answer.

If some assumptions of the internal parameters can be established we can reduce the number of required views. For example, if we assume zero skew and known principal point then we need just one view of the planar pattern.

Problem 5 *2 Points*

Consider two images I and I' taken by the same camera (with intrinsic matrix $K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$) where the motion between them consists in a rotation along the Y axes (the axes defined by the optical center of the camera) and a translation of d_x meters along the X' axes (see corresponding Figure). Remember that a 3D point in the world P (expressed in camera O_c coordinate system) corresponds to point $P' = RP + T'$ (expressed in camera O'_c coordinate system). Answer the following questions:

a) Compute the corresponding translation vector T'.

b) Compute the corresponding rotation matrix R.

c) Compute the essential matrix E.

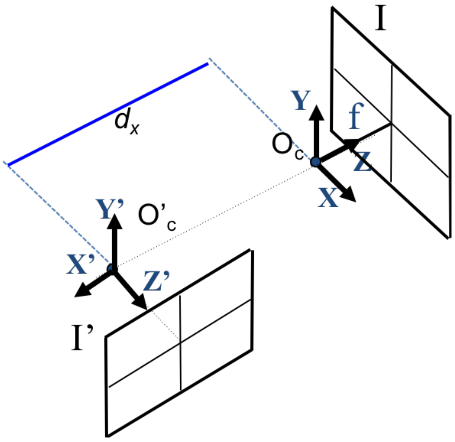
d) Compute the fundamental matrix F.

e) Where is the epipole e in image I?

f) Where is the epipole e' in image I'?

g) What is the main difference between the essential matrix E and fundamental matrix F?

f) Is it possible to recover R and T' from E? with any restriction?



a) $T' = \begin{bmatrix} -d_x \\ 0 \\ 0 \end{bmatrix}$

b) $R = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

c) $E = [T'_\times]\,R = \begin{bmatrix} 0 & 0 & 0 \\ d_x & 0 & 0 \\ 0 & -d_x & 0 \end{bmatrix}$

d) $F = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -f & 0 \end{bmatrix}$

e) At image center: $e = (0, 0)$

f) At infinity.

g) F expresses the relation in uncalibrated cameras (in pixel coordinates) while E expresses the relation in the calibrate cased (camera coordinates).

f) Yes, you can obtain R and T' from E but with normalized T'. The scale is not possible to obtain unless there is an object in the scene with known length.

Problem 6 *1.5 Points*

Disparity estimation with local methods.

(a) *(0.25p)* Describe the main steps for estimating the disparity of a pair of stereo rectified images with a local method.

The local methods work independently for every pixel and they need a matching cost. The basic steps are described below.

For every pixel in the left image:

(i) Slide a window along the same line in the right image and compare its content to that of the reference window in the left image.

(ii) Pick the disparity with minimum matching cost (WTA: Winner-Takes-All approach).

(b) *(0.5p)* Define the Sum of Squares Differences, Sum of Absolute Differences, Normalized Cross Correlation and comment their advantages/disadvantages.

- SSD: Sum of Squared Differences ($m = 2$)
- SAD: Sum of Absolute Differences ($m = 1$)

$$C(\mathbf{p}, \mathbf{d}) = \sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q}) \, |I_1(\mathbf{q}) - I_2(\mathbf{q} + \mathbf{d})|^m, \qquad \sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q}) = 1$$

$N_\mathbf{p} = \{\mathbf{q} = (q_1, q_2)^T \,|\, p_1 - \frac{n}{2} \le q_1 \le p_1 + \frac{n}{2}, \, p_2 - \frac{n}{2} \le q_2 \le p_2 + \frac{n}{2}\}$

- NCC: Normalized Cross Correlation

$$NCC(\mathbf{p}, \mathbf{d}) = \frac{\sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q})(I_1(\mathbf{q}) - \bar{I}_1)(I_2(\mathbf{q} + \mathbf{d}) - \bar{I}_2)}{\sigma_{I_1} \sigma_{I_2}}$$

$\bar{I}_1 = \sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q}) I_1(\mathbf{q}); \quad \bar{I}_2 = \sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q}) I_2(\mathbf{q} + \mathbf{d});$

$\sigma_{I_1} = \sqrt{\sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q})(I_1(\mathbf{q}) - \bar{I}_1)^2}; \; \sigma_{I_2} = \sqrt{\sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q})(I_2(\mathbf{q} + \mathbf{d}) - \bar{I}_2)^2}$

The SSD and SAD are faster to compute than the NCC. The SAD is more robust to outliers. The NCC is more robust to illumination changes but looses discriminatory power.

(c) *(0.25p)* Consider a window with uniform weights. Describe the positive and negative effects of the window size.

A smaller window gives more details but at the expense of noisier disparities. On the other hand, a larger window gives smoother results but looses detail at object boundaries and small objects.

(d) *(0.25p)* Write the expression of the bilateral weights and comment its benefits.

$$w(\mathbf{p},\mathbf{q}) = w_{col}(\mathbf{p},\mathbf{q})\, w_{pos}(\mathbf{p},\mathbf{q}) = exp\left(-\frac{\Delta c(\mathbf{p},\mathbf{q})}{\gamma_{col}}\right)\, exp\left(-\frac{\Delta g(\mathbf{p},\mathbf{q})}{\gamma_{pos}}\right)$$

$$\Delta c(\mathbf{p},\mathbf{q}) = \frac{1}{3}\|I(\mathbf{p}) - I(\mathbf{q})\|_1 = \frac{1}{3}\sum_{c\in\{r,g,b\}} |I_c(\mathbf{p}) - I_c(\mathbf{q})|$$

$$\Delta g(\mathbf{p},\mathbf{q}) = \|\mathbf{p}-\mathbf{q}\|_2 = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2}$$

These weights allow to work with large window sizes for getting smooth results without loosing accuracy at disparity edges.

(e) *(0.25p)* Describe different possible failure cases when estimating the disparity through local methods and two views.

- Textureless areas
- Occlusions
- Repetitions (self-similarity)
- Non-Lambertian surfaces, specularities

Problem 7 *0.4 Points*

Formulate the projection equation (indicating the corresponding size of every matrix) in terms of a measurement matrix, assuming: 1) a perspective camera, 2) an orthographic camera. To compute shape and motion by rigid 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, the projection equation is $\mathbf{M} = \mathbf{PX}$:

1) Perspective case: $\mathbf{M}$ is a $3m \times n$ measurement matrix, $\mathbf{P}$ and $\mathbf{X}$ are a $3m \times 4$ motion and $4 \times n$ shape matrix components, respectively.

2) Orthographic case: $\mathbf{M}$ 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.

In both cases, we will use a SVD factorization, imposing a rank-4 and rank-3 decomposition for perspective and orthographic cases, respectively.

Problem 8 *1.1 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 16, 10, 18, and 18 components for every image, respectively. Assuming $P = 18$, 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. (0.8 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. Could the designed pattern be used to handle continuous non-rigid objects? Explain why, and provide some ideas to modify it if needed. (0.3 points)

The rows correspond to the number of equations, and the columns with the number of parameters to be estimated. Thus, the number of rows is $2 * (16 + 10 + 18 + 18)$, and the number of columns is $(I * e + 3 * P)$, where e represents the number of extrinsic parameters, i.e., 3 translations and 3 rotations (or 4 using quaternions). The corresponding pattern is displayed in Fig.1.

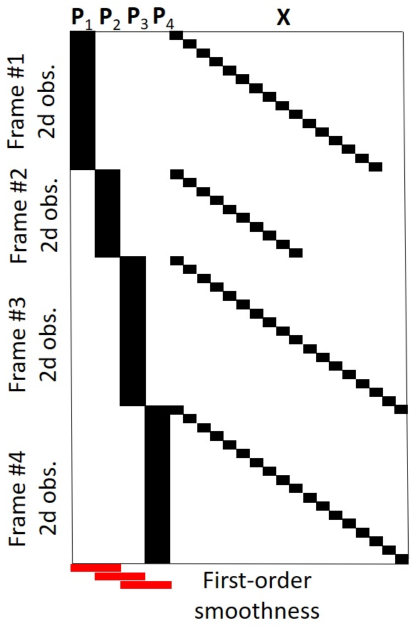


Figure 1: Jacobian pattern.

For monocular video, we may add temporal smoothness priors to estimate the camera motion. To this end, first- (pure sequential) or high-order approximations (such as a sliding-window with several frames) can be used. For simplicity, we incorporate and represent a pure sequential case (adding 3 constraints), i.e., $\mathbf{P}_i \approx \mathbf{P}_{i+1}$.

The pattern is not available for the non-rigid case, due to in this case the p-th 3D location is not independent of the rest of points. For this case, a different 3D shape configuration $\mathbf{X}_i$ is needed to be estimated, where we could also include temporal smoothness priors on the deformation.

Problem 9 *0.5 Points*

We set out with four questions for the 3D sensors/ 3D processing part of this module. Can you provide a short answer for each one?

1) Is "projective vision" a natural way to capture the 3D world?
2) Do we need photometry to get geometry?
3) Does 3D vision mean the same than 3D geometry?
4) Does 2D/3D matter for "Teaching computers to see"?

1) Humans and most living beings are equipped with a pair of projective (passive) sensors (eyes) performing stereoscopic vision to compute distantces. However, capturing 3D geometry can be better done with active sensors probing the actual distance to scene surfaces.

2) For 2D projective imaging, the answer is yes: we do need photometry. And so we need computing disparities from stereo sensors or structure from motion to get the scene geometry. But Lidar or TOF sensors, for instance, can compute scene geometry by radar principles, and without resorting to disparity in photometric data.

3) Absolutely not. 3D vision usually assumes one (or several) points of view from which 3D geometry is computed. This leaves part of the scene geometry unavailable due to occlusions!
On the contrary, 3D geometry uses to be a complete representation of the scene (such as in Graphics or in CAD design) both if such geometry is occluded or not.

4) Reconstructing scenes and objects in the 3D world for analysis and recognition can help avoiding large variability in appearances and missing information due to occlusions, reasoning about events and understanding complex visual scenes. Yes, 3D geometry may definitely help computers to see and "understand" scenes.

Problem 10 *1.0 Points*

1) Point cloud data can be ”organized” or ”unorganized”. Define both types.

2) Which type of point cloud data can be straightforwardly converted into RGBD and how? Which type has a disadvantage for processing purposes and why?

3) A convenient feature extracted from pointcloud data should be able to capture the same local surface characteristics in the presence three main variations: transformations, data density and noise. Describe these three variations and state why pointcloud features should be invariant to those

4) Mention two point feature representations and describe what do they aim to capture from pointcloud data characteristics

1) Organized point-cloud data is arranged in matrix-like structure. It is projectable and facilitates nearest neighbor operations, which are more efficient, and speed up computation times. Unorganized point-clouds are a non-regular sampling of 3D space, in which simple neighborhood operations may require a KD tree search.

2) An organized point-cloud is projectable, i.e. has a correlation according to a pinhole camera model between the 2D (u,v) index of a point in the organized point cloud and the actual 3D positions, which eases finding neighbor points. Unorganized point clouds are not projectable, i.e. there is no correlation according with a 2D projected pixel, like for organized point clouds. This makes neighborhood operations, such as those needed for local feature extraction or smoothing, re- quire k-d tree search, for finding neighboring points in 3D which is costly in terms of computation compared to finding nearest neighbors in organized/indexed datasets such as 2D images or voxelized 3D volumes.

3) A good point feature representation is able to capture the same (invariant) local surface characteristics in the presence of a) rigid transformations (3D rotations and translations should not influence the resulting feature vector estimation), b) varying sampling density (a local surface patch sampled more or less densely should have the same feature vector signature), c) noise (the point feature representation must retain the same or very similar values in the presence of mild noise in the data)

4) Any of the following ones: Signature of Histograms of OrienTations (SHOT), VFH signatures, Point Feature Histograms (PFH), Fast Point Feature Histograms (FPFH). Descriptions available in PCL (pointclouds.org/documentation/tutorials)