Lecture 03: Image Formation Part II

1 Image Formation Continued

In this segment we are going to introduce linear and non-linear methods for determining the pose of the camera in the world.

1.1 Nonlinear Algorithms

1.1.1 Pose determination from n Points (PnP) Problem

Problem Description

Given the realitve spatial locations of n control points and given the angle to every pair of control points from an additional point called the Center of Perspective C_P, find the lengths of the line segments joining C_P to each of the control points.
We assume we know the camera intrinsic parameters. Given known 3D landmarks in the world and their image correspondence in the camera frame, determine the 6DOF pose of the camera in the world frame. **Where is the camera?**

Behaviour of the Solutions

- Given 1 point: ∞ solutions.
- Given 2 points: ∞ *bounded* solutions.
- Given 3 **non collinear** points: finitely many (up to 4) solutions.
- Given 4 points: **unique** solution.

P3P: Solution for 3 Points

The 3-points case is depicted in Figure 1. In order to solve this instance of the problem,

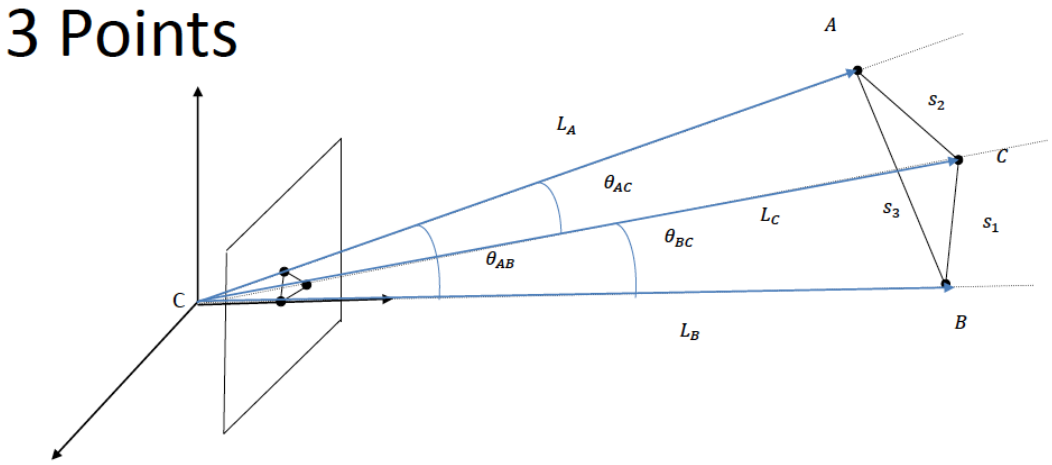

Figure 1: PnP for 3 points.

one can use the fact that the angles inscribed in the triangle are the same: the Carnot's theorem for them reads

$$\begin{aligned} s_1^2 &= L_B^2 + L_C^2 - 2L_B L_C \cos(\theta_{BC}) \\ s_2^2 &= L_A^2 + L_C^2 - 2L_A L_C \cos(\theta_{AC}) \\ s_3^2 &= L_A^2 + L_B^2 - 2L_A L_B \cos(\theta_{AB}) \end{aligned} \tag{1.1}$$

In general, n independent polynomials with n unknowns, can have no more solutions than the **product** of their degrees: here 8.
The fourth point is needed to **disambiguate** the solutions! By defining

$$x = \frac{L_B}{L_A}, \tag{1.2}$$

we can reduce the system to the 4^{th} order equation

$$G_0 + G_1 x + G_2 x^2 + G_3 x^3 + G_4 x^4 = 0. \tag{1.3}$$

This applies to camera pose estimation from known $3D - 2D$ correspondences (e.g. hololens).

1.2 Linear Algorithms

1.2.1 Camera Calibration

Camera calibration represents a procedure to determine **intrinsic** and **extrinsic** parameters of the camera model.

Tsai Method

Tsai proposed in 1987 a procedure consisting in measuring the 3D position of more than 6 points (also known as *control points*) on a 3D calibration target and the 2D coordinates of their projection in the image. This problem is known as *resection* or *perspective from n points* and is extremely widely used. This algorithm can be written, by recalling the perspective projection equation and by neglecting the radial sensor distortion.

Direct Linear Transform (DLT)

The goal of this procedure is to determine matrices K, R, and T which satisfy the perspective projection equation. Let's recall the representation of an image point:

$$\begin{aligned}
\text{Image point} = \tilde{p} = \begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} &= \lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K[R|T] \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \\
\text{Assuming independent elements} \qquad &= \underbrace{\begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix}}_{M} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} m_1^\intercal \\ m_2^\intercal \\ m_3^\intercal \end{pmatrix} \cdot \underbrace{\begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}}_{P},
\end{aligned} \tag{1.4}$$

where $m_i^\intercal$ represents the i-th row of the unknown matrix M.
One can now exploit the conversion from homogeneous coordinates to pixel coordinates and gets:

$$\begin{aligned}
u = \frac{\tilde{u}}{\tilde{w}} = \frac{m_1^\intercal \cdot P}{m_3^\intercal \cdot P}, \\
v = \frac{\tilde{v}}{\tilde{w}} = \frac{m_2^\intercal \cdot P}{m_3^\intercal \cdot P},
\end{aligned} \tag{1.5}$$

which can be rewritten as

$$\begin{aligned}
(m_1^\intercal - u_i m_3^\intercal) \cdot P_i = 0, \\
(m_2^\intercal - v_i m_3^\intercal) \cdot P_i = 0,
\end{aligned} \tag{1.6}$$

for all points P_i. Rearranging the terms for one point results in the compact matrix equation

$$\begin{pmatrix} P_1^\intercal & 0^\intercal & -u_1 P_1^\intercal \\ 0^\intercal & P_1^\intercal & -v_1 P_1^\intercal \end{pmatrix} \cdot \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{1.7}$$

Generalizing this structure for n points, one gets a $2n \times 12$ matrix Q, such that the problem can be written as

$$Q \cdot M = 0, \tag{1.8}$$

where Q is known and M is unknown.

Minimal Solution:

In order for the system to have a unique (up to scale) non-trivial (different from 0) solution M, the $2n \times 12$ matrix Q should have rank 11 (i.e. at most rank deficient by 1). Since each 3D-to-2D point correspondence provides 2 *independent* equations, a total of $\frac{11}{2} = 5 + \frac{1}{2}$ point correspondences are needed. Clearly, in practice 6 point correspondences are needed.

Overdetermined Solution:

As soon as one has more that 6 points, the equations will overdetermine the solution and a minimization approach will be needed. One of the possible approaches is to minimize the euclidean norm

$$\|Q \cdot M\|^2, \tag{1.9}$$

subject to the constraint

$$\|M\|^2 = 1, \tag{1.10}$$

i.e., normed solution. This can be solved using Singular Value Decomposition (SVD). The solution is then represented by the eigenvector corresponding to the smallest eigenvalue of the matrix $Q^\intercal Q$. In fact, this vector is the unit vector which minimizes the expression

$$\|Qx\|^2 = x^\intercal Q^\intercal Q x. \tag{1.11}$$

In Matlab, this translates into

```
[U,S,V] = svd(Q);
 M = V(:,12);
```

Degenerated Configurations:

Are all points useful for this procedure? In the following, we provide a list of situations where the points don't provide enough information for the calibration to be feasible.

- Points lying on a plane and/ or along a line passing through the projection center.
- Camera and points on a twisted cubic (degree 3).

Once we have $M = K(R|T)$, we have a matrix which encodes the camera intrinsics and extrinsics. Recalling

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}, \tag{1.12}$$

one can use the QR-factorization of M to separate intrinsics and extrinsics. In fact, the factorization decomposes M into an orthogonal matrix R,T and an upper triangular matri K.

Remark. Notice that we are not enforcing orthogonality of R.

Tsai Method 1987

Tsai method can be essentially described through the following blocks:

1. Edge detection.
2. Straight line fitting to the detected edges.
3. Intersecting the lines to obtain the image corners (<0.1 pixels accuracy).
4. Use more than 6 points (more than 20) and **not** all on the same plane.

Originally pixels were not considered to be squared (i.e., the two focal lengths were different). Furthermore, a skew factor ($K_{12} \neq 0$) was considered and the pixels were parallelogramms instead of rectangles. Most cameras today are well manufactured and have hence

$$\frac{\alpha_u}{\alpha_v} = 1, \quad K_{12} = 0. \tag{1.13}$$

Residual: With the term residual, one refers to the average reprojection error, computed as the distance (in pixels) between the observed point and the camera-reprojected 3D point. This measure gives an intuition on the accuracy of the calibration.

What if K is known? Nothing changes!

Calibration from Planar Grids (Homographies)

Tsai calibration required observed points not to lie on the same plane. An alternative method (today's standard camera calibration method) consists of using a planar grid (e.g. a chessboard) and a few images of it shown at different orientations. This method was invented by Zhang, now at Microsoft Research. Given that all the points on the chessboard lie on a plane, we can set $Z_w = 0$. It holds

$$\begin{aligned}
\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} &= \begin{pmatrix} \alpha_u & 0 & v_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} \alpha_u & 0 & 0 \\ 0 & \alpha_v & \nu_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ 1 \end{pmatrix} \\
&= H \cdot \begin{pmatrix} X_w \\ Y_w \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} h_1^\intercal \\ h_2^\intercal \\ h_3^\intercal \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ 1 \end{pmatrix}.
\end{aligned} \tag{1.14}$$

Matrix H is called **Homography**. One can exploit once more the conversion from homogeneous coordinates to pixel coordinates and gets

$$\begin{aligned}
u &= \frac{\tilde{u}}{\tilde{w}} = \frac{h_1^\intercal \cdot P}{h_3^\intercal \cdot P}, \\
v &= \frac{\tilde{v}}{\tilde{w}} = \frac{h_2^\intercal \cdot P}{h_3^\intercal \cdot P},
\end{aligned} \tag{1.15}$$

and hence

$$\begin{aligned}(h_1^\intercal - u_i h_3^\intercal) \cdot P_i = 0 \\ (h_2^\intercal - v_i h_3^\intercal) \cdot P_i = 0,\end{aligned} \tag{1.16}$$

for all points P_i. Rearranging the terms, one has

$$\begin{aligned}Q \cdot H = 0 \\ \begin{pmatrix} P_1^\intercal & 0^\intercal & -u_1 P_1^\intercal \\ 0^\intercal & P_1^\intercal & -v_1 P_1^\intercal \end{pmatrix} \cdot \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\end{aligned} \tag{1.17}$$

where Q is known and H is unknown. Since we dropped Z_w from the unknowns, the generalized structure for n points makes Q a $2n \times 9$ matrix.

Minimal Solution:

In order for the system to have a unique (up to scale) non-trivial (different from 0) solution H, the $2n \times 9$ matrix Q should have rank 8 (i.e. at most rank deficient by 1). Since each 3D-to-2D point correspondence provides 2 *independent* equations, a total of $\frac{8}{2} = 4$ point correspondences (with non-collinear points) are needed.

Overdetermined Solution:

As soon as one has more that 4 points, the equations will overdetermine the solution and a minimization approach will be needed. One of the possible approaches is to minimize the euclidean norm

$$\|Q \cdot M\|^2, \tag{1.18}$$

subject to the constraint

$$\|M\|^2 = 1, \tag{1.19}$$

i.e., normed solution, as previously explained.

Applications of homographies are

- Augmented reality
- Beacon-based localization.

Remark. If the camera is calibrated, only R and T need to be determined. Pnp leads to smaller error than DLT.

1.3 Non Conventional Camera Models

1.3.1 Omnidirectional Cameras

Omnidirectional sensors come in many varieties, but by definition must have a wide field-of-view (FOV). We can find:

- Wide FOV dioptric cameras (e.g. fisheye (180)).
- Catadioptric cameras (e.g. mirrors (>180)). Combine a standard camera with a shaped mirror.

  - Mirror: **central**, mirror (surface of revolution of a conic), single effective view point.
  - Perspetive: hyperbola+perspective / parabola+orthographic lens.

- Polydioptric cameras (e.g. multiple overlapping cameras) ≈ 360.

Definition 1. A vision system is said to be **central**, when the optical rays to the viewed objects intersect into a single point in 3D called *projection center* or *single effective viewpoint*. For hyperbolic and elliptical mirrors, the single viewpoint property is achieved by ensuring that the camera center coincides with one of the foci of the hyperbola (ellipse). For this, refer to Figure 2

In general, mirrors which ensure centrality of the camera are *rotated (swept) conic shapes* (hyperbolical, parabolical and elliptical mirrors).
Why is it important for the camera to be central? If the camera is central, we can unwarp parts of the omnidirectional image into perspective. We can transform image points in the unit sphere. We can apply algorithms for perspective geometry. Perspective and omnidirectional model are equal!

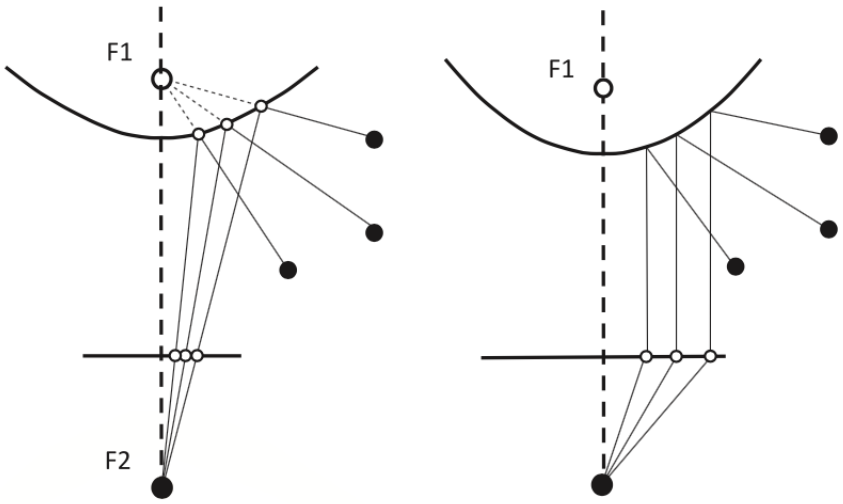

Fig. 3. Central catadioptric cameras can be built by using hyperbolic and parabolic mirrors. The parabolic mirror requires the use of an orthographic lens.

Figure 2: How should the mirrors be?

1.4 Understanding Check

Are you able to:

- *Describe the general PnP problem and derive the behaviour of its solutions?*
- *Explain the working principle of the P3P algorithm?*
- *Explain DLT? What is the minimum number of point correspondences it requires?*
- *Define centeral and non central omnidirectional cameras?*
- *What kind of mirrors ensure central projection?*