CSE-483: Mobile Robotics **Monsoon 2019**

Camera Calibration and DLT

Prepared by: Yash Agarwal, Anshul Lahoti

1 The Inner Parameters

In this section we will introduce the inner parameters of the cameras. Recall from the camera equations:

$$\lambda \mathbf{x} = P\mathbf{X} \tag{1}$$

where P=K[R t], K is a 3 × 3 matrix R is a 3 × 3 rotation matrix and t is a 3 × 1 vector. The 3 × 4 matrix [R t] encodes the orientation and position of the camera with respect to a reference coordinate system. Given a 3D point in homogeneous coordinates X, the product [R t]X can be interpreted as the 3D coordinates of the scene point in the camera coordinate system. Note that alternatively we can interpret the result as the homogeneous coordinates of the projection of X in to the image plane embedded in $\mathbb{R}^3$, since the projection in the camera coordinate system is computed by division with the third coordinate. The 3 × 3 matrix K transforms the image plane in $\mathbb{R}^3$ to the real image coordinate system (with unit pixels), see Figure 1.

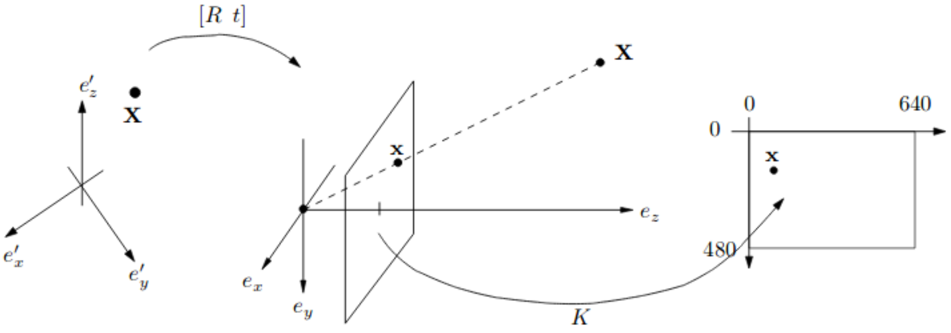

Figure 1: The different coordinate systems and mappings

The matrix K is an upper triangular matrix with the following shape:

$$K = \begin{pmatrix} \gamma f & sf & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

The parameter f is called the focal length. This parameter re-scales the image coordinates into pixels. The point (x0,y0) is called the principal point. For many cameras it is enough to use the focal length and principal point. In this case, the K matrix transforms the image points according to:

$$\begin{pmatrix} fx + x_0 \\ fy + y_0 \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{3}$$

that is, the coordinates are scaled by the focal length and translated by the principal point. Note that the center point (0,0,1) of the image in $\mathbb{R}^3$ is transformed to the principal point (x0,y0).
The parameter γ is called the aspect ratio. For cameras where the pixels are not square the re-scaling needs to be done differently in the x-direction and the y-direction. In such cases the aspect ratio γ will take a value different from 1.

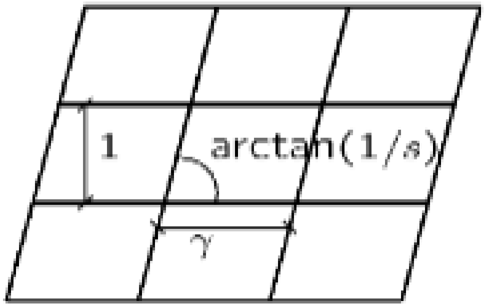

Figure 2: The skew parameters corrects for non-rectangular pixels.

The final parameters is called the skew. This parameter corrects for tilted pixels, see Figure 2, and is typically zero.
A camera P=K[R t] is called calibrated if the inner parameters K are known. For such cameras we can eliminate the K matrix from the camera equations by multiplying both sides of (1) with K^{-1} from the left. If we let

$$\tilde{\mathbf{x}} = K^{-1}\mathbf{x}$$

we get:

$$\lambda\tilde{\mathbf{x}} = K^{-1}K[R \quad t]\mathbf{X} = [R \quad t]\mathbf{X} \tag{4}$$

The new camera matrix [R t] is called the normalized(calibrated) camera and the new image points x are called the normalized image points. (Note that later in this lecture there is a different concept of normalization that is used for improving stability of computations. However in the context of calibrated cameras, normalization always means multiplication with K^{-1}.)
The calibration model presented in this section is limited in the sense that all the normalization is limited to applying the homography K to the image coordinates. For some cameras this is not sufficient. For example, the image in Figure 3 lines that are straight in 3D do not appear as straight lines in the image. Such distortion is common in cameras with wide field of view and can not be removed with a homograpy.

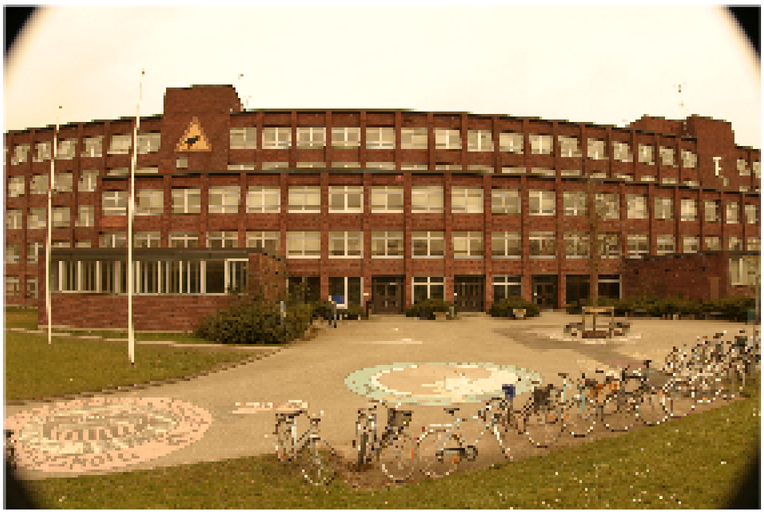

Figure 3: Radial distortion can not be handled with the K matrix. This requires a more complicated model.

2 Projective vs. Euclidean Reconstruction

The main problem of interest in this course is the Structure from Motion problem, that is, given image projections X_{ij} (of scene point i in image j) determine both 3D point coordinates X_j and camera matrices P_i such that

$$\lambda_{ij}\mathbf{x}_{ij} = P_i\mathbf{X}_j, \quad \forall i,j \tag{5}$$

Note that the depths λ_{ij} are also unknown and need to be determined. However, primarily we are interested in the scene points and cameras, the depths are a bi-product of this formulation. If the calibration is unknown, that is P_i can be any non-zero 3×4 matrix then the solution to this problem is called a projective reconstruction. Such a solution can only be uniquely determined up to a projective transformation.To see this suppose that we have found cameras P_i and 3D-points X_j such that

$$\lambda_{ij}\mathbf{x}_{ij} = P_i\mathbf{X}_j \tag{6}$$

To construct a different solution we can take an unknown projective transformation H $\left(\mathbb{P}^3 \mapsto \mathbb{P}^3\right)$ and let $\tilde{P}_i = P_iH$ and $\tilde{\mathbf{X}}_j = H^{-1}\mathbf{X}_j$. The new cameras and scene points also solve the problem since

$$\lambda_{ij}\mathbf{x}_{ij} = P_i\mathbf{X}_j = P_iHH^{-1}\mathbf{X}_j = \tilde{P}_i\tilde{\mathbf{X}}_j \tag{7}$$

This means that given a solution we can apply any projective transformation to the 3D points and obtain a new solution. Since projective transformations do not necessarily preserve angles or parallel lines projective reconstructions can look distorted even though the projections they give match the measured image points. To the left in Figure 4 a projective reconstruction of the Arch of Triumph in Paris is displayed.

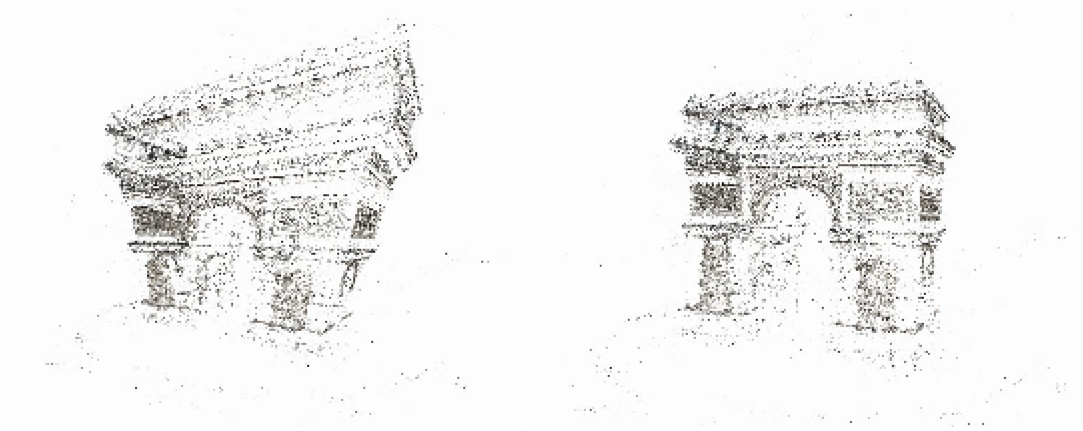

Figure 4: Reconstructions of the Arch of Triumph in Paris. Left: Projective reconstruction. Right: Euclidean reconstruction(known camera calibration). Both reconstructions provide the same projections.

One way to remove the projective ambiguity is to use calibrated cameras. If we normalize the image coordinates using $\tilde{\mathbf{x}} = K^{-1}\mathbf{x}$ then the structure form motion problem becomes that of finding normalized (calibrated) cameras $[R_i \quad t_i]$ and scene points X_j such that

$$\lambda_{ij}\tilde{\mathbf{x}}_{ij} = [R_i \quad t_i]\,\mathbf{X}_j \tag{8}$$

where the first 3×3 block R_i is a rotation matrix. The solution of this problem is called a Euclidean Reconstruction. Given a solution $[R_i \quad t_i]$ and X_j we can try to do the same trick as in the projective case. However when multiplying $[R_i \quad t_i]$ with H, the result does not necessarily have a rotation matrix in the first 3×3 block.To achieve a valid solution we need H to be a similarity transformation,

$$H = \begin{bmatrix} sQ & v \\ 0 & 1 \end{bmatrix} \tag{9}$$

where Q is a rotation. We then get

$$\frac{\lambda_{ij}}{s}\mathbf{x}_{ij} = \begin{bmatrix} R_i & t_i \end{bmatrix} \begin{bmatrix} Q & \frac{1}{s}v \\ 0 & \frac{1}{s} \end{bmatrix} H^{-1}\mathbf{X}_j = \begin{bmatrix} R_iQ & \frac{1}{s}\left(R_iv + t_i\right) \end{bmatrix} \tilde{\mathbf{X}}_j \tag{10}$$

which is a valid solution since R_i^1Q is a rotation. Hence, in the case of Euclidean reconstruction we do not have the same distortion since similarity transformations preserve angles and parallel lines. Note that there is still anambiguity here. The entire reconstruction can be re-scaled, rotated and translated without changing the image projections.

3 Finding the Inner Parameters

In this section we will present a simple method for finding the camera parameters. We will do it in two steps:
1. First, we will compute a camera matrix P . To make sure that there is not projective ambiguity present (as in Section 2) we will assume that the scene point coordinates are known. This can for example be achieved by using an image of a known object where we have measured all the points by hand.
2. Secondly, once the camera matrix P is known we can factorize it into K[R t] where K is triangular and R is a rotation. This can be done using the so called RQ-factorization.

3.1 Finding P : The Resection Problem

In this section we will outline a method for finding the camera matrix P . We are assuming that the scene points $\mathbf{X}_i$ and their projections $\mathbf{x}_i$ are known. The goal is to solve the equations

$$\lambda_i\mathbf{x}_i = P\mathbf{X}_i, \quad i = 1, \ldots, N \tag{11}$$

where the λ_i and P are the unknowns. This problem, determining the camera matrix from know scene points and projections is called the resection problem. The 3×4 matrix P has 12 elements but the scale is arbitrary and therefore 11 degrees of freedom. There are 3N equations (3 for each point projection), but each new projection introduces one additional unknown λ_i. Therefore we need

$$3N \geq 11 + N \Rightarrow N \geq 6 \tag{12}$$

points in order for the problem to be well defined. To solve the problem we will use a simple approach called Direct Linear Transformation (DLT). This method formulates a homogeneous linear system of equations and solves this by finding an approximate null space of the system matrix. If we let $\mathbf{p}_i$, i = 1, 2, 3 be 4×1 vectors containing the rows of P , that is,

$$P = \begin{bmatrix} p_1^T \\ p_2^T \\ p_3^T \end{bmatrix} \tag{13}$$

then we can write (11) as

$$\begin{gathered} \mathbf{X}_i^T p_1 - \lambda_i x_i = 0 \\ \mathbf{X}_i^T p_2 - \lambda_i y_i = 0 \\ \mathbf{X}_i^T p_3 - \lambda_i = 0 \end{gathered} \tag{14}$$

where $\mathbf{x}_i$=($\mathbf{x}_i$, $\mathbf{y}_i$, 1) . In matrix form this can be written

$$\begin{bmatrix} \mathbf{X}_i^T & 0 & 0 & -x_i \\ 0 & \mathbf{X}_i^T & 0 & -y_i \\ 0 & 0 & \mathbf{X}_i^T & -1 \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ \lambda_i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{15}$$

Note that since $\mathbf{X}_i$ is a 4×1 vector each 0 on the left hand side actually represents a 1×4 block of zeros. Thus the left hand side is a 3×13 matrix multiplied with a 13×1 vector. If we include all the projection equations in one matrix we get a system of the form

$$\underbrace{\begin{bmatrix} \mathbf{X}_1^T & 0 & 0 & -x_1 & 0 & 0 & \dots \\ 0 & \mathbf{X}_1^T & 0 & -y_1 & 0 & 0 & \dots \\ 0 & 0 & \mathbf{X}_1^T & -1 & 0 & 0 & \dots \\ \mathbf{X}_2^T & 0 & 0 & 0 & -x_2 & 0 & \dots \\ 0 & \mathbf{X}_2^T & 0 & 0 & -y_2 & 0 & \dots \\ 0 & 0 & \mathbf{X}_2^T & 0 & -1 & 0 & \dots \\ \mathbf{X}_3^T & 0 & 0 & 0 & 0 & -x_3 & \dots \\ 0 & \mathbf{X}_3^T & 0 & 0 & 0 & -y_3 & \dots \\ 0 & 0 & \mathbf{X}_3^T & 0 & 0 & -1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{=M} \underbrace{\begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \end{pmatrix}}_{=v} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}.$$

Figure 5:

Here we are interested in finding a non-zero vector in the nullspace of M . Since the scale is arbitrary we can add the constraint $\|v\|^2 = 1$. In most cases the system Mv = 0 will not have any exact solution due to noise in the measurements. Therefore we will search for a solution to

$$\min_{\|v\|^2=1} \|Mv\|^2 \tag{16}$$

We refer to this type of problem as a homogeneous least squares problem. Note that there are always at least two solutions to (16) since $\|Mv\| = \|M(-v)\| \quad and \quad \|v\| = \|-v\|$. These solutions give the same projections however for one of them the camera faces away from the scene points thereby giving negative depths. If the homogeneous representative for the scene points have positive fourth coordinate then we should select the solution where the λ_i are all positive.
An alternative formulation with only the p-variables can be found by noting that (11) means that the vectors $\mathbf{x}_i$ and $\mathbf{PX}_i$ are be parallel. This can alternatively be expressed using the vector product

$$\mathbf{x}_i \times P\mathbf{X}_i = 0, \quad i = 1, \dots, N \tag{17}$$

These equations are also linear in $\mathbf{p}_1$,$\mathbf{p}_2$, $\mathbf{p}_3$ and we can therefore set up a similar homogeneous least squares system but without the λ_i .

3.1.1 Solving the Homogeneous System

The solution to (16) can be found by eigenvalue computations. If we let $f(v) = v^T M^T M v$ and $g(v) = v^T v$. We can write the problem as

$$\min_{g(v)=1} f(v) \tag{18}$$

Therefore the solution of (16) has to be an eigenvector of the matrix $M^T M$. Suppose v_* is an eigenvector with eigenvalue γ_* . If we insert into the objective function we get

$$f\left(v_*\right) = v_*^T M^T M v_* = \gamma_* v_*^T v_* \tag{19}$$

Since $\|v_*\| = 1$ we see that in order to minimize f we should select the eigenvector with the smallest eigenvalue.Because of the special shape of $M^T M$ we compute the eigenvectors efficiently using the so called Singular Value Decomposition (SVD).

Theorem 1. Each m × n matrix M (with real coefficients) can be factorized into

$$M = USV^T \tag{20}$$

where U and V are orthogonal (m × m and n × n respectively),

$$S = \begin{bmatrix} \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) & 0 \\ 0 & 0 \end{bmatrix} \tag{21}$$

$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ and r is the rank of the matrix. If M has the SVD (20) then

$$M^T M = \left(USV^T\right)^T USV^T = VS^T U^T USV^T = VS^T SV^T \tag{22}$$

Since S T S is a diagonal matrix this means that V diagonalizes $M^T M$ and therefore $S^T S$ contains the eigenvalues and V the eigenvectors of $M^T M$. The diagonal elements of $S^T S$ are ordered decreasingly $\sigma_1^2, \sigma_2^2, \ldots, \sigma_r^2, 0, \ldots, 0$. Thus, to find an eigenvector corresponding to the smallest eigenvalue we should select the last column of V . Note that if r n, that is, the matrix M does not have full rank then the eigenvalue we select will be zero which means that there is an exact nonzero solution to Mv = 0. In most cases however r = n due to noise.

3.1.2 Normalization.

The matrix M will contain entries $\mathbf{x}_i$, $\mathbf{y}_i$ and ones. Since since the x i and y i are measured in pixels the values can be in the thousands. In contrast the third homogeneous coordinate is 1 and therefore the matrix M contains coefficient of highly varying magnitude. This can make the matrix $M^T M$ poorly conditioned resulting buildup of numerical errors.
The numerics can often be greatly improved by translating the coordinates such that their center of mass is zero and then rescaling the coordinates to be roughly 1.
Suppose that we want to solve

$$\lambda_i \mathbf{x} = P\mathbf{X}_i, \quad i = 1, \ldots, N \tag{23}$$

as outlined in the previous sections.
We can change the coordinates of the image points by applying the normalization mapping

$$N = \begin{bmatrix} s & 0 & -s\overline{x} \\ 0 & s & -s\overline{y} \\ 0 & 0 & 1 \end{bmatrix} \tag{24}$$

This mapping will first translate the coordinates by $(-\overline{x}, \overline{y})$ and then re-scale the result with the factor s. If for example $(\overline{x}, \overline{y}))$ is the mean point then the transformation

$$\tilde{\mathbf{x}} = N\mathbf{x} \tag{25}$$

gives re-scaled coordinates with "center of mass" in the origin.
We can now solve the modified problem

$$\gamma_i \tilde{\mathbf{x}} = \tilde{P}\mathbf{X}_i \tag{26}$$

by forming the system matrix M and computing its singular value decomposition. A solution to the original "un-normalized" problem (23) can now easily be found from

$$\gamma_i N\mathbf{x} = \tilde{P}\mathbf{X}_i \tag{27}$$

3.2 Computing the Inner Parameters from P

When the camera matrix has been computed we want to find the inner parameters K by factorizing P into

$$P = K[R \quad t] \tag{28}$$

where K is a right triangular and R is a rotation matrix. We this can be done using the RQ-factorization.

Theorem 2: *If A is an n × n matrix then there is an orthogonal matrix Q and a right triangular matrix R such that*

$$A = RQ\ .$$

(If A is invertible and the diagonal elements are chosen positive then the factorization is unique.)

In order to be consistent with the notation in the rest of the lecture we will use K for the right triangular matrix and R for the orthogonal matrix. Given a camera matrix P= [A a] we want to use RQ-factorization to find K and R such that A=KR. If

$$K = \begin{pmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{pmatrix}, \quad A = \begin{bmatrix} A_1^T \\ A_2^T \\ A_3^T \end{bmatrix} \text{ and } R = \begin{bmatrix} R_1^T \\ R_2^T \\ R_3^T \end{bmatrix} \tag{29}$$

that is, R1, R2, R3 and A1, A2, A3 are 31 vectors containing the rows of R and A respectively, then we get

$$\begin{bmatrix} A_1^T \\ A_2^T \\ A_3^T \end{bmatrix} = \begin{pmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{pmatrix} \begin{bmatrix} R_1^T \\ R_2^T \\ R_3^T \end{bmatrix} = \begin{bmatrix} aR_1^T + bR_2^T + cR_3^T \\ dR_2^T + eR_3^T \\ fR_3^T \end{bmatrix} \tag{30}$$

From the third row of (30) we see that A_3=fR_3. Since the matrix R is orthogonal R_3 has to have the length 1. We therefore see that need to select

$$f = \|A_3\| \quad \text{and} \quad R_3 = \frac{1}{\|A_3\|} A_3 \tag{31}$$

to get a positive coefficient f. When R_3 is known we can proceed to the second row of (30). The equation $A_2 = dR_2 + eR_3$ tells us that A_2 is a linear combination of two orthogonal vectors (both of length one). Hence,the coefficient e can be computed from the scalar product

$$e = A_2^T R_3 \tag{32}$$

When e is known we can compute R_2 and d from

$$dR_2 = A_2 - eR_3 \tag{33}$$

similar to what we did for f and R_3 in (31). When R_2 and R_3 is known we use the first row of (30)

$$A_1 = aR_1 + bR_2 + cR_3 \tag{34}$$

to compute b and c. Finally we can compute a and R_1 from

$$A_1 - bR_2 - cR_3 = aR_1 \tag{35}$$

The resulting matrix K is not necessarily of the form (2) since element (3,3) might not be one. To determine the individual parameters, focal length, principal point etc. we therefore need to divide the matrix with element(3,3). Note however, that this does not modify the camera in any way since the scale is arbitrary.

References

[1] *What Is Camera Calibration?* Link

[2] *Robotic Vision*, Aalto University Link

[3] *Direct Linear Transformation(DLT)* Link

[4] *Camera Calibration and DLT, Young-Hoo Kwon* Link

[5] *Calibration and Reconstruction*, University of Toronto Link

[6] *Camera Calibration*, University of Maryland Link

[7] *DLT method for Stereo Camera Calibration*, P. Morasso and V. Mohan Link