Gebze Institute of Technology
Department of Computer Engineering

# Computer Vision

# Geometric Camera Calibration

# Geometric Camera Calibration

- Geometric camera calibration is the estimation of camera parameters (intrinsic, extrinsic, or both)
- Why do we need camera calibration?
- Most vision techniques need camera parameters but there are a few that don't need the parameters.

$$p = M P$$

$$K = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$p = K [R \ T] P$$

$$( \ ) \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & & & \\ 0 & & & \end{bmatrix}$$

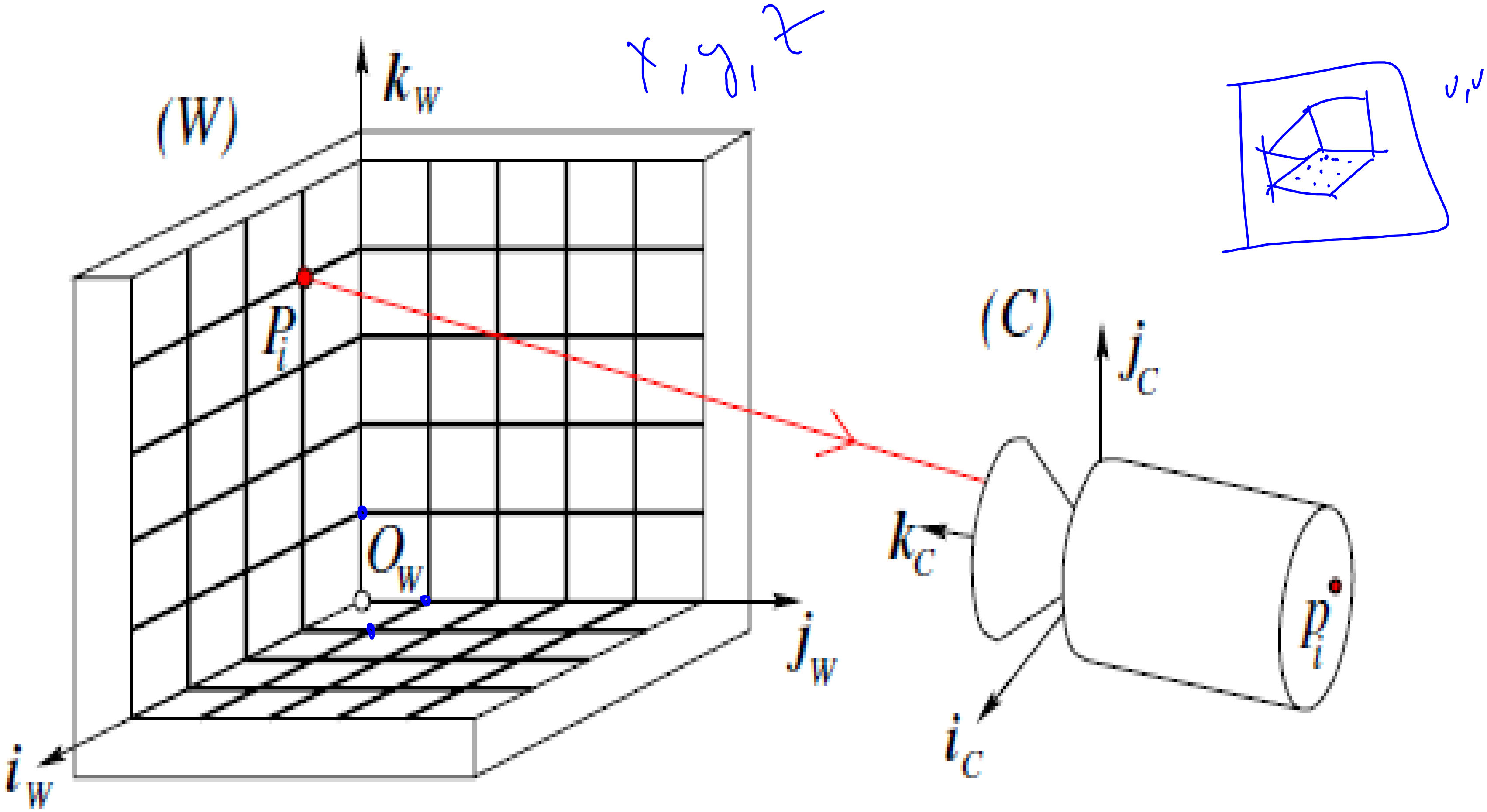



**Figure 6.1.** Camera calibration setup: in this example, the calibration rig is formed by three grids drawn in orthogonal planes. Other patterns could be used as well, and they may involve lines or other geometric figures.

# General Process

$(a\ b\ c)(a\ b\ c)$

- Assume we have n image points (ui, vi) and corresponding n world points Pi in homogeneous coordinates.
- We first find the projection Matrix

$RR^T = I$

$R^T = R^{-1}$

$$p = \frac{1}{z} \mathcal{M} P, \quad \text{where} \quad \mathcal{M} = \mathcal{K} ( \mathcal{R} \quad t ),$$

- Then estimate the intrinsic and extrinsic parameters.

# Estimation of Projection Matrix

$$\mathcal{M} = \begin{pmatrix} \alpha r_1^T - \alpha\cot\theta r_2^T + u_0 r_3^T & \alpha t_x - \alpha\cot\theta t_y + u_0 t_z \\ \frac{\beta}{\sin\theta} r_2^T + v_0 r_3^T & \boxed{\frac{\beta}{\sin\theta} t_y + v_0 t_z} \\ \boxed{r_3^T} & t_z \end{pmatrix},$$

$$K = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M = K\,[R\ \ t]$$

# Estimation of the projection Matrix

- Collect n points from the scene with the image correspondences.
- $$\boldsymbol{p} = \frac{1}{z}\mathcal{M}\boldsymbol{P}, \quad \text{where} \quad \mathcal{M} = \mathcal{K}\begin{pmatrix}\mathcal{R} & \boldsymbol{t}\end{pmatrix}$$

$\mathcal{M}$ and $\mathcal{P}$ since, if $\boldsymbol{m}_1^T$, $\boldsymbol{m}_2^T$ and $\boldsymbol{m}_3^T$ denote the three rows of $\mathcal{M}$, it follows directly from (5.2.7) that $z = \boldsymbol{m}_3 \cdot \boldsymbol{P}$. In fact, it is sometimes convenient to rewrite (5.2.7) in the equivalent form:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = MP$$

$$\begin{cases} u = \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}, \\ v = \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}. \end{cases} \tag{5.2.8}$$

$$p = \frac{1}{z} MP$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{z} MP$$

$$\begin{cases} (\boldsymbol{m}_1 - u_i \boldsymbol{m}_3) \cdot \boldsymbol{P}_i = 0, \\ (\boldsymbol{m}_2 - v_i \boldsymbol{m}_3) \cdot \boldsymbol{P}_i = 0. \end{cases}$$

$$\underline{Bm = a}$$

$$\begin{cases} (\boldsymbol{m}_1 - u_i \boldsymbol{m}_3) \cdot \boldsymbol{P}_i = 0, \\ (\boldsymbol{m}_2 - v_i \boldsymbol{m}_3) \cdot \boldsymbol{P}_i = 0. \end{cases}$$

Collecting the constraints associated with all points yields a system of $2n$ homogeneous linear equations in the twelve coefficients of the matrix $\mathcal{M}$, namely,

$$\mathcal{P}\boldsymbol{m} = 0, \tag{6.2.1}$$

$m = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$ $\quad 2n = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

where

$$\mathcal{P} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{P}_1^T & \boldsymbol{0}^T & -u_1 \boldsymbol{P}_1^T \\ \boldsymbol{0}^T & \boldsymbol{P}_1^T & -v_1 \boldsymbol{P}_1^T \\ \dots & \dots & \dots \\ \boldsymbol{P}_n^T & \boldsymbol{0}^T & -u_n \boldsymbol{P}_n^T \\ \boldsymbol{0}^T & \boldsymbol{P}_n^T & -v_n \boldsymbol{P}_n^T \end{pmatrix} \quad \text{and} \quad \boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{pmatrix} = 0.$$

SVD

# Least Squares Methods

Let us first consider a system of $p$ linear equations in $q$ unknowns:

$$\begin{cases} u_{11}x_1 + u_{12}x_2 + \ldots + u_{1q}x_q = y_1 \\ u_{21}x_1 + u_{22}x_2 + \ldots + u_{2q}x_q = y_2 \\ \ldots \\ u_{p1}x_1 + u_{p2}x_2 + \ldots + u_{pq}x_q = y_p \end{cases} \iff \mathcal{U}\boldsymbol{x} = \boldsymbol{y},$$

where

$$\mathcal{U} \stackrel{\text{def}}{=} \begin{pmatrix} u_{11} & u_{12} & \ldots & u_{1q} \\ u_{21} & u_{22} & \ldots & u_{2q} \\ \ldots & \ldots & \ldots & \ldots \\ u_{p1} & u_{p2} & \ldots & u_{pq} \end{pmatrix}, \quad \boldsymbol{x} \stackrel{\text{def}}{=} \begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_q \end{pmatrix} \quad \text{and} \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_p \end{pmatrix}$$

Consider the over-constrained case $p > q$

# Least Squares Methods

- Minimize an error measure to find the optimal x values.

$$E \stackrel{\text{def}}{=} \sum_{i=1}^{p} (u_{i1}x_1 + \ldots + u_{iq}x_q - y_i)^2 = |\mathcal{U}\boldsymbol{x} - \boldsymbol{y}|^2.$$

$$E = \boldsymbol{e}\cdot\boldsymbol{e}, \text{ where } \boldsymbol{e} \stackrel{\text{def}}{=} \mathcal{U}\boldsymbol{x} - \boldsymbol{y}.$$

$$\frac{\partial E}{\partial x_i} = 2\frac{\partial \boldsymbol{e}}{\partial x_i}\cdot\boldsymbol{e} = 0 \quad \text{for} \quad i = 1, \ldots, q.$$

But if the columns of $\mathcal{U}$ are the vectors $\mathbf{c}_j = (u_{1j}, \ldots, u_{mj})^T$ $(j = 1, \ldots, q)$, we have

$$\frac{\partial \boldsymbol{e}}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\begin{pmatrix} \mathbf{c}_1 & \ldots & \mathbf{c}_q \end{pmatrix}\begin{pmatrix} x_1 \\ \ldots \\ x_q \end{pmatrix} - \boldsymbol{y}\right] = \frac{\partial}{\partial x_i}(x_1\mathbf{c}_1 + \ldots + x_q\mathbf{c}_q - \boldsymbol{y}) = \mathbf{c}_i.$$

# Least Squares Methods

In particular, writing that $\partial E/\partial x_i = 0$ implies that $\boldsymbol{c}_i^T(\mathcal{U}\boldsymbol{x} - \boldsymbol{y}) = 0$, and stacking the constraints associated with the $q$ coordinates of $\boldsymbol{x}$ yields the *normal equations* associated with out least-squares problem, i.e.,

$$\mathbf{0} = \begin{pmatrix} \boldsymbol{c}_1^T \\ \dots \\ \boldsymbol{c}_q^T \end{pmatrix} (\mathcal{U}\boldsymbol{x} - \boldsymbol{y}) = \mathcal{U}^T(\mathcal{U}\boldsymbol{x} - \boldsymbol{y}) \Longleftrightarrow \mathcal{U}^T\mathcal{U}\boldsymbol{x} = \mathcal{U}^T\boldsymbol{y}.$$

When $\mathcal{U}$ has maximal rank $q$, the matrix $\mathcal{U}^T\mathcal{U}$ is easily shown to be invertible, and the solution of the normal equations is $\boldsymbol{x} = \mathcal{U}^\dagger\boldsymbol{y}$ with $\mathcal{U}^\dagger \stackrel{\text{def}}{=} [(\mathcal{U}^T\mathcal{U})^{-1}\mathcal{U}^T]$. The $q \times q$ matrix $\mathcal{U}^\dagger$ is called the *pseudoinverse* of $\mathcal{U}$. It coincides with $\mathcal{U}^{-1}$ when the matrix $\mathcal{U}$ is square and non-singular. Linear least-squares problems can be solved without explicitly computing the pseudoinverse, using for example QR decomposition or singular value decomposition (more on the latter in Chapter 14), which are known to be better behaved numerically.

- # Singular Value Decomposition:
  - Any mxn matrix can be written as the product of three matrices $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$

$\mathbf{U}_1$ $\mathbf{V}_1$

$$\begin{bmatrix} a_{11} & a_{12} & & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ & & & \\ a_{m1} & a_{m2} & & a_{mn} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & & u_{1m} \\ u_{21} & u_{22} & & u_{2m} \\ & & & \\ u_{m1} & u_{m2} & & u_{mm} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & & 0 \\ 0 & \sigma_2 & & \\ 0 & & & \sigma_n \\ 0 & & & 0 \end{bmatrix} \begin{bmatrix} v_{11} & v_{21} & & v_{n1} \\ v_{12} & v_{22} & & v_{n2} \\ & & & \\ v_{1n} & v_{2n} & & v_{nn} \end{bmatrix}$$

- Singular values $\sigma_i$ are fully determined by A
  - D is diagonal: $d_{ij} = 0$ if $i \neq j$; $d_{ii} = \sigma_i$ (i=1,2,…,n)
  - $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_N \geq 0$
- Both U and V are not unique
  - Columns of each are mutual orthogonal vectors

$Ax = \lambda x$

# Singular Value Decomposition

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

- 1. Singularity and Condition Number
  - nxn A is nonsingular IFF all singular values are nonzero
  - Condition number : degree of singularity of A $\quad C = \sigma_1 / \sigma_n$
    - A is ill-conditioned if 1/C is comparable to the arithmetic precision of your machine; almost singular

$$\begin{bmatrix} & A & \end{bmatrix} x = b$$

$$x = A^{-1} b$$

- 2. Rank of a square matrix A
  - Rank (A) = number of nonzero singular values
- 3. Inverse of a square Matrix

$$A A^{-1} = I$$

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \quad \begin{bmatrix} \frac{1}{a} & 0 & 0 \\ 0 & \frac{1}{b} & 0 \\ 0 & 0 & \frac{1}{c} \end{bmatrix}$$

  - If A is nonsingular $\quad \mathbf{A}^{-1} = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T$
  - In general, the pseudo-inverse of A $\quad \underline{\mathbf{A}^{+} = \mathbf{V}\mathbf{D}_0^{-1}\mathbf{U}^T}$

$$A x = b$$

$$x = A^{+} b$$

- 4. Eigenvalues and Eigenvectors
  - Eigenvalues of both $A^TA$ and $AA^T$ are $si^2$ ($si > 0$)
  - The columns of U are the eigenvectors of $AA^T$ (mxm)
  - The columns of V are the eigenvectors of $A^TA$ (nxn)

$$\mathbf{A}\mathbf{A}^T\mathbf{u}_i = \sigma_i^2\mathbf{u}_i$$

$$\mathbf{A}^T\mathbf{A}\mathbf{v}_i = \sigma_i^2\mathbf{v}_i$$

# Singular Value Decomposition

$$\mathbf{Ax} = \cancel{\mathbf{b}}\;0$$

- Least Squares
  - Solve a system of m equations for n unknowns **x**(m >= n)
  - A is a mxn matrix of the coefficients
  - b ($\neq 0$) is the m-D vector of the data
  - Solution:

$$\underbrace{\mathbf{A}^T\mathbf{A}}_{\text{nxn matrix}}\mathbf{x} = \mathbf{A}^T\mathbf{b} \quad \Rightarrow \quad \mathbf{x} = \underbrace{(\mathbf{A}^T\mathbf{A})^+}_{\text{Pseudo-inverse}}\mathbf{A}^T\mathbf{b}$$

  - How to solve: compute the pseudo-inverse of $A^TA$ by SVD
    - $(A^TA)^+$ is more likely to coincide with $(A^TA)^{-1}$ given m > n
    - Always a good idea to look at the condition number of $A^TA$

# Singular Value Decomposition

- Homogeneous System $\mathbf{Ax} = \mathbf{0}$
  - m equations for n unknowns $\mathbf{x}$(m >= n-1)
  - Rank (A) = n-1 (by looking at the SVD of A)
  - A non-trivial solution (up to a arbitrary scale) by SVD:
  - Simply proportional to the eigenvector corresponding to the only zero eigenvalue of $A^T A$ (nxn matrix)
- Note: $\mathbf{A}^T \mathbf{A} \mathbf{v}_i = \sigma_i^2 \mathbf{v}_i$
  - All the other eigenvalues are positive because Rank (A)=n-1
  - In practice, the eigenvector (i.e. $v_n$) corresponding to the minimum eigenvalue of $A^T A$, i.e. $\sigma_n^2$

# Singular Value Decomposition

- Problem Statements
  - Numerical estimate of a matrix A whose entries are not independent
  - Errors introduced by noise alter the estimate to Â
- Enforcing Constraints by SVD
  - Take orthogonal matrix A as an example
  - Find the closest matrix to Â, which satisfies the constraints exactly
    - SVD of Â $$\hat{\mathbf{A}} = \mathbf{UDV}^T$$
    - Observation: D = I (all the singular values are 1) if A is orthogonal
    - Solution: changing the singular values to those expected

# Estimation of the Intrinsic and Extrinsic Params

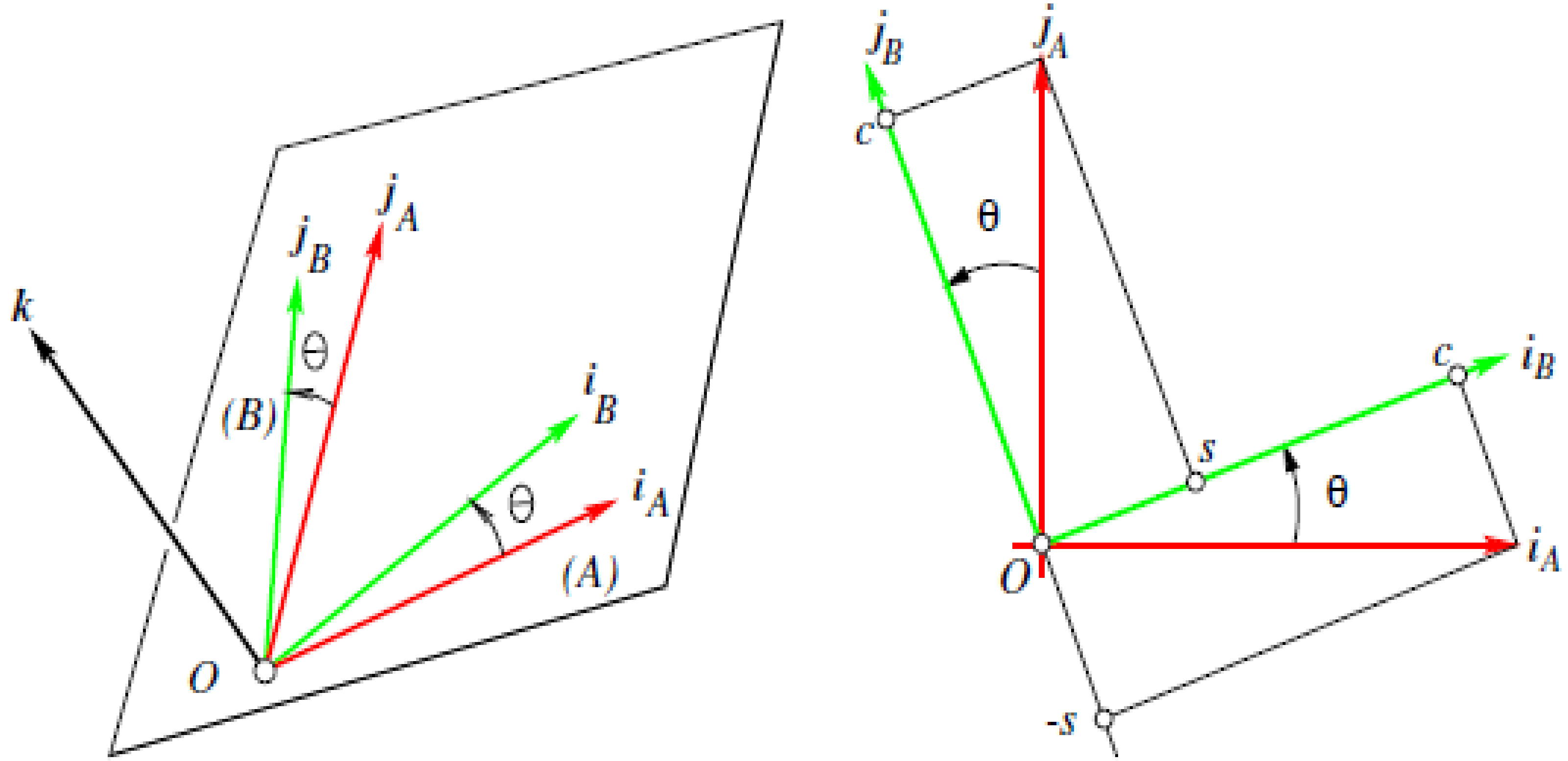
Once we recover M = (A b), then we can define

$\boldsymbol{a}_1^T$, $\boldsymbol{a}_2^T$ and $\boldsymbol{a}_3^T$ denoting the rows of $\mathcal{A}$, and obtain

$$\rho\,(\mathcal{A} \quad \boldsymbol{b}) = \mathcal{K}\,(\mathcal{R} \quad \boldsymbol{t}) \iff \rho \begin{pmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \boldsymbol{a}_3^T \end{pmatrix} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T \\ \dfrac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T \\ \boldsymbol{r}_3^T \end{pmatrix},$$

where $\rho$ is an unknown scale factor,

Remember R is orthogonal $R^T R = I$




$$
{}^{B}_{A}\mathcal{R} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.
$$

$$\cdot \rho \begin{pmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \boldsymbol{a}_3^T \end{pmatrix} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T \\ \dfrac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T \\ \boldsymbol{r}_3^T \end{pmatrix},$$

$$\begin{cases} \rho = \varepsilon / |\boldsymbol{a}_3|, \\ \boldsymbol{r}_3 = \rho \boldsymbol{a}_3, \\ u_0 = \rho^2 (\boldsymbol{a}_1 \cdot \boldsymbol{a}_3), \\ v_0 = \rho^2 (\boldsymbol{a}_2 \cdot \boldsymbol{a}_3), \end{cases} \qquad \text{where } \varepsilon = \mp 1.$$

$$\cdot \rho \begin{pmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \boldsymbol{a}_3^T \end{pmatrix} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T \\ \dfrac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T \\ \boldsymbol{r}_3^T \end{pmatrix},$$

$$\begin{cases} \rho^2 (\boldsymbol{a}_1 \times \boldsymbol{a}_3) = -\alpha \boldsymbol{r}_2 - \alpha \cot\theta \boldsymbol{r}_1, \\ \rho^2 (\boldsymbol{a}_2 \times \boldsymbol{a}_3) = \dfrac{\beta}{\sin\theta} \boldsymbol{r}_1, \end{cases} \quad \text{and} \quad \begin{cases} \rho^2 |\boldsymbol{a}_1 \times \boldsymbol{a}_3| = \dfrac{|\alpha|}{\sin\theta}, \\ \rho^2 |\boldsymbol{a}_2 \times \boldsymbol{a}_3| = \dfrac{|\beta|}{\sin\theta}, \end{cases}$$

thus:

$$\begin{cases} \cos\theta = -\dfrac{(\boldsymbol{a}_1 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3)}{|\boldsymbol{a}_1 \times \boldsymbol{a}_3| |\boldsymbol{a}_2 \times \boldsymbol{a}_3|}, \\ \alpha = \rho^2 |\boldsymbol{a}_1 \times \boldsymbol{a}_3| \sin\theta, \\ \beta = \rho^2 |\boldsymbol{a}_2 \times \boldsymbol{a}_3| \sin\theta, \end{cases}$$

$$\cdot \rho \begin{pmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \boldsymbol{a}_3^T \end{pmatrix} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T \\ \dfrac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T \\ \boldsymbol{r}_3^T \end{pmatrix},$$

$$\begin{cases} \rho^2(\boldsymbol{a}_1 \times \boldsymbol{a}_3) = -\alpha \boldsymbol{r}_2 - \alpha\cot\theta \boldsymbol{r}_1, \\ \rho^2(\boldsymbol{a}_2 \times \boldsymbol{a}_3) = \dfrac{\beta}{\sin\theta}\boldsymbol{r}_1, \end{cases} \quad \text{and} \quad \begin{cases} \rho^2|\boldsymbol{a}_1 \times \boldsymbol{a}_3| = \dfrac{|\alpha|}{\sin\theta}, \\ \rho^2|\boldsymbol{a}_2 \times \boldsymbol{a}_3| = \dfrac{|\beta|}{\sin\theta}, \end{cases}$$

$$\begin{cases} \boldsymbol{r}_1 = \dfrac{\rho^2 \sin\theta}{\beta}(\boldsymbol{a}_2 \times \boldsymbol{a}_3) = \dfrac{1}{|\boldsymbol{a}_2 \times \boldsymbol{a}_3|}(\boldsymbol{a}_2 \times \boldsymbol{a}_3), \\ \boldsymbol{r}_2 = \boldsymbol{r}_3 \times \boldsymbol{r}_1. \end{cases}$$

$$\rho\begin{pmatrix}\mathcal{A} & \boldsymbol{b}\end{pmatrix} = \mathcal{K}\begin{pmatrix}\mathcal{R} & \boldsymbol{t}\end{pmatrix} \Longleftrightarrow \rho\begin{pmatrix}\boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \boldsymbol{a}_3^T\end{pmatrix} = \begin{pmatrix}\alpha\boldsymbol{r}_1^T - \alpha\cot\theta\boldsymbol{r}_2^T + u_0\boldsymbol{r}_3^T \\ \dfrac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0\boldsymbol{r}_3^T \\ \boldsymbol{r}_3^T\end{pmatrix},$$

$$\mathcal{K}\boldsymbol{t} = \rho\boldsymbol{b} \qquad\qquad \boldsymbol{t} = \rho\mathcal{K}^{-1}\boldsymbol{b}.$$

The sign for epsilon can be calculated using the sign of tz.

# Rotation R : Orthogonality

$$\hat{\mathbf{R}}^T \hat{\mathbf{R}} = \mathbf{I} \, ?$$

$$\hat{\mathbf{R}} = \mathbf{UDV^T} \Rightarrow \mathbf{R} = \mathbf{UIV^T}$$

Replace the diagonal matrix D with the 3x3 identity matrix