Lecture 16

Camera Calibration

+

Homographies 1

Mon. Nov. 2, 2020

Recall lecture 14: Camera Model

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{12} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

We would like to estimate ***P***.

Recall lecture 14: Camera Model

$$\mathbf{P} = \underbrace{\mathbf{K}}_{\text{intrinsic}} \overbrace{\mathbf{R}\,[\,\mathbf{I}\,|\,-\mathbf{C}\,]}^{\text{extrinsic}}$$

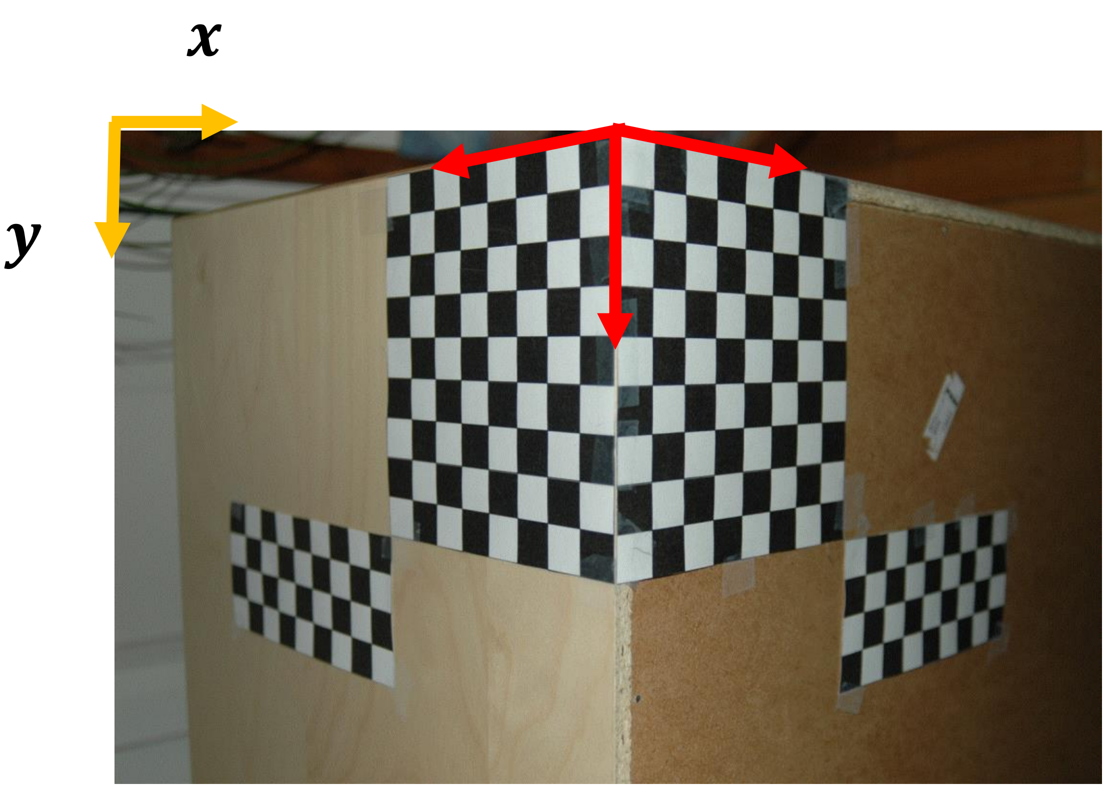
3x4 3x3 3x3 3x4

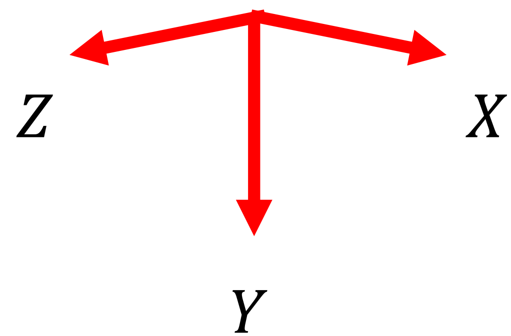
$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

We would like to estimate $\boldsymbol{P}$, and then factor it.

Suppose we have an object with *identifiable points* measured at 3D positions (X_i, Y_i, Z_i) in some scene coordinate system and labelled corresponding pixels (x_i, y_i) for i=1,…,N in an image.
e.g. corner points of the squares in the checkerboard below.

Compute a projection matrix $\boldsymbol{P}$ that best fits these data $\{X_i, Y_i, Z_i, x_i, y_i\}$.
This problem is called *camera calibration*.





Compute a projection matrix $\boldsymbol{P}$ that best fits these data $\{X_i, Y_i, Z_i\ , x_i, y_i\}$.

$$\begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i \end{bmatrix} \approx \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

We will solve it using least squares.   How ?

Compute a projection matrix $\boldsymbol{P}$ that best fits these data $\{X_i, Y_i, Z_i\ , x_i, y_i\}$.

$$\begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i \end{bmatrix} \approx \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$x_i = \frac{w_i x_i}{w_i} \approx \frac{P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}}$$

$$y_i = \frac{w_i y_i}{w_i} \approx \frac{P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}}$$

Compute a projection matrix $\boldsymbol{P}$ that best fits these data $\{X_i, Y_i, Z_i\ , x_i, y_i\}$.

$$\begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i \end{bmatrix} \approx \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$x_i = \frac{\cancel{w_i} x_i}{\cancel{w_i}} \approx \frac{P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}}$$

$$y_i = \frac{\cancel{w_i} y_i}{\cancel{w_i}} \approx \frac{P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}}$$

$$x_i(P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}) \approx P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}$$
$$y_i(P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}) \approx P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}.$$

From previous slide:

$$x_i(P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}) \approx P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}$$
$$y_i(P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}) \approx P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}.$$

Stack the N pairs of equations (corresponding to the N points) :

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N & -x_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N & -y_N \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ \vdots \\ P_{31} \\ P_{32} \\ P_{33} \\ P_{34} \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$2N \times 12$

$$
\begin{bmatrix}
X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\
0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N & -x_N \\
0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N & -y_N
\end{bmatrix}
\begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ \vdots \\ P_{31} \\ P_{32} \\ P_{33} \\ P_{34} \end{bmatrix}
\approx
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

This is of the familiar form $\boldsymbol{A}\boldsymbol{x} \approx 0$, where $\boldsymbol{A}$ is an $2N \times 12$ data matrix, and $\boldsymbol{x}$ is a vector of P_{ij} values.

Solution: ?

$$
\begin{bmatrix}
X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\
0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N & -x_N \\
0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N & -y_N
\end{bmatrix}
\begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ \vdots \\ P_{31} \\ P_{32} \\ P_{33} \\ P_{34} \end{bmatrix}
\approx
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

This is of the familiar form $\boldsymbol{Ax} \approx 0$, where $\boldsymbol{A}$ is an $2N \times 12$ data matrix, and $\boldsymbol{x}$ is a vector of P_{ij} values.

Choose the $\boldsymbol{x}$ vector that minimizes $\|\boldsymbol{Ax}\|$ subject to $\|\boldsymbol{x}\| = 1$.

Solution: take the eigenvector of $\boldsymbol{A^TA}$ with the smallest eigenvalue.

$$
\begin{bmatrix}
X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\
0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N & -x_N \\
0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N & -y_N
\end{bmatrix}
\begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ \vdots \\ P_{31} \\ P_{32} \\ P_{33} \\ P_{34} \end{bmatrix}
\approx
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

$$
\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix}
$$

ASIDE: $\| \mathbf{P} \| = \sqrt{\sum_{ij} {P_{ij}}^2}$ is called the *Frobenius norm* of matrix $\mathbf{P}$.

How to improve the estimate of $\mathbf{P}$?

$$
\begin{bmatrix}
X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\
0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N & -x_N \\
0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N & -y_N
\end{bmatrix}
\begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ \vdots \\ P_{31} \\ P_{32} \\ P_{33} \\ P_{34} \end{bmatrix}
\approx
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

What are the units of $(X_i, Y_i, Z_i\ )$ and (x_i, y_i) ?

Image and scene position values have noise.

The magnitudes of values in different columns can vary *a lot*!
Columns with largest values will dominate the estimate.

Data Normalization

$$(\bar{x}, \bar{y}) = \frac{1}{N}\sum_{i=1}^{N}(x_i, y_i)$$

$$(\bar{X}, \bar{Y}, \bar{Z}) = \frac{1}{N}\sum_{i=1}^{N}(X_i, Y_i, Z_i)$$

$$\sigma_1^2 = \frac{1}{2N}\sum_{i=1}^{N}(x_i - \bar{x})^2 + (y_i - \bar{y})^2$$

$$\sigma_2^2 = \frac{1}{3N}\sum_{i=1}^{N}(X_i - \bar{X})^2 + (Y_i - \bar{Y})^2 + (Z_i - \bar{Z})^2$$

Normalize so that the mean and standard deviation are 0 and 1, respectively.

$$(x_i, y_i) \rightarrow (\frac{x_i - \bar{x}}{\sigma_1}, \frac{y_i - \bar{y}}{\sigma_1})$$

$$(X_i, Y_i, Z_i) \rightarrow (\frac{X_i - \bar{X}}{\sigma_2}, \frac{Y_i - \bar{Y}}{\sigma_2}, \frac{Z_i - \bar{Z}}{\sigma_2})$$

Data Normalization

$$(x_i, y_i) \rightarrow (\frac{x_i - \bar{x}}{\sigma_1}, \frac{y_i - \bar{y}}{\sigma_1})$$

$$\mathbf{M_1} = \underbrace{\begin{bmatrix} 1/\sigma_1 & 0 & 0 \\ 0 & 1/\sigma_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\bar{x} \\ 0 & 1 & -\bar{y} \\ 0 & 0 & 1 \end{bmatrix}}$$

$$(X_i, Y_i, Z_i) \rightarrow (\frac{X_i - \bar{X}}{\sigma_2}, \frac{Y_i - \bar{Y}}{\sigma_2}, \frac{Z_i - \bar{Z}}{\sigma_2})$$

$$\mathbf{M_2} = \underbrace{\begin{bmatrix} 1/\sigma_2 & 0 & 0 & 0 \\ 0 & 1/\sigma_2 & 0 & 0 \\ 0 & 0 & 1/\sigma_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\bar{X} \\ 0 & 0 & 0 & -\bar{Y} \\ 0 & 0 & 1 & -\bar{Z} \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

Data Normalization

$$(x_i, y_i) \rightarrow (\frac{x_i - \bar{x}}{\sigma_1}, \frac{y_i - \bar{y}}{\sigma_1})$$

$$(X_i, Y_i, Z_i) \rightarrow (\frac{X_i - \bar{X}}{\sigma_2}, \frac{Y_i - \bar{Y}}{\sigma_2}, \frac{Z_i - \bar{Z}}{\sigma_2})$$

$$\underbrace{\begin{bmatrix} 1/\sigma_1 & 0 & 0 \\ 0 & 1/\sigma_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\bar{x} \\ 0 & 1 & -\bar{y} \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{M_1}}$$

$$\underbrace{\begin{bmatrix} 1/\sigma_2 & 0 & 0 & 0 \\ 0 & 1/\sigma_2 & 0 & 0 \\ 0 & 0 & 1/\sigma_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\bar{X} \\ 0 & 0 & 0 & -\bar{Y} \\ 0 & 0 & 1 & -\bar{Z} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{M_2}}$$

Solve the least squares problem using these normalized coordinates, namely estimate $\boldsymbol{P}_{normalized}$.

$$\mathbf{M_1} \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i \end{bmatrix} \approx \mathbf{P}_{normalized} \mathbf{M_2} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$\mathbf{M_1}\begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i \end{bmatrix} \approx \mathbf{P}_{normalized}\mathbf{M_2}\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Finally, we want a matrix a matrix **P** that is in terms of the original data. Left multiplying by $\mathbf{M_1}^{-1}$, we see what we want:

$$\mathbf{P} \equiv \mathbf{M_1}^{-1}\mathbf{P}_{normalized}\mathbf{M_2}$$

Experiments have shown that this normalization technique can give much better results *in practice*.

ASIDE: Another least squares formulation…

$$\begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i \end{bmatrix} \approx \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

The above model gave us:

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N & -x_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N & -y_N \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ \vdots \\ P_{31} \\ P_{32} \\ P_{33} \\ P_{34} \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

One issue with problem formulation is that it is unclear how to *interpret* the error in terms of the geometry of the situation, namely the 3D points (X_i, Y_i, Z_i) and the image pixel positions (x_i, y_i).

What are we minimizing here ?!

(ASIDE: continued) A more geometrically *meaningful* way to set up the problem is to minimize:

$$error = \sum_i (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2$$

plug in

where (x_i, y_i) are the measured image positions and $(\hat{x}_i, \hat{y}_i)$ are the *predicted positions* for any $\boldsymbol{P}$ and scene point (X_i, Y_i, Z_i).

$$\begin{bmatrix} \hat{w}_i \hat{x}_i \\ \hat{w}_i \hat{y}_i \\ \hat{w}_i \end{bmatrix} \equiv \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \longrightarrow \begin{aligned} \hat{x}_i &= \frac{P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} \\ \hat{y}_i &= \frac{P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} \end{aligned}$$

This defines a *non-linear* least squares problem. How to solve it? One can make a linear approximation of this *error* (linear in the P_{ij}) which is of the form $\| \mathbf{A}\,\mathbf{u} - \mathbf{b} \|^2$ and then apply a least squares method, and iterate similar to Lucas-Kanade.

Finally…

Having solved for a matrix $\boldsymbol{P}$ one can then factor it using linear algebra techniques.

(In an Appendix in the lecture notes, I have provided details on how to compute this factorization, for those who are interested.)

$$\mathbf{P} \;=\; \mathbf{K}\,\mathbf{R}\,[\,\mathbf{I}\,|\,-\mathbf{C}\,]$$

3x4 3x3 3x3 3x4

Summary of Camera Calibration

- For each scene point and corresponding image point, rewrite the following as two equations:

$$\begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i \end{bmatrix} \approx \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

- Stacking these 2N equations into a matrix and solve using least squares.

- Normalization of the data points gives better performance in practice.

- There are other ways to set up and solve the problem, e.g. non-linear least squares.

Lecture 16

Camera Calibration

+

Homographies 1

Mon. Nov. 2, 2020

Recall (lecture 13): Homogeneous Coordinates in 2D

Translation:

$$\begin{bmatrix} x+T_x \\ y+T_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotation:

$$\begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Recall (lecture 13): Homogeneous Coordinates in 2D

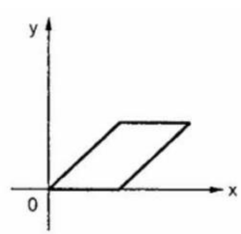
scaling

$$\begin{bmatrix} \sigma_x x \\ \sigma_y y \\ 1 \end{bmatrix} = \begin{bmatrix} \sigma_X & 0 & 0 \\ 0 & \sigma_Y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

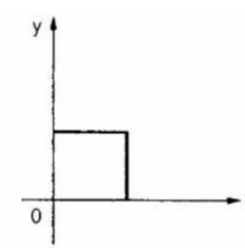
shear

$$\begin{bmatrix} x + sy \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Recall motion field of ground plane from last lecture



(b) Object after x shear



(a) Original object

Affine (2D) transformation

$$\begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

translation scaling rotation reflection shear

If we multiply such matrices, we always obtain a matrix of the form:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}$$

Note the two elements in bottom row that have value 0.

The resulting mapping is called a 2D *affine* transformation.
It is invertible. (if scaling coefficients are non-zero 0).

Homography

Any 3×3 *invertible* matrix that maps between 2D “homogeneous points” is called a *homography*.

$$\boldsymbol{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

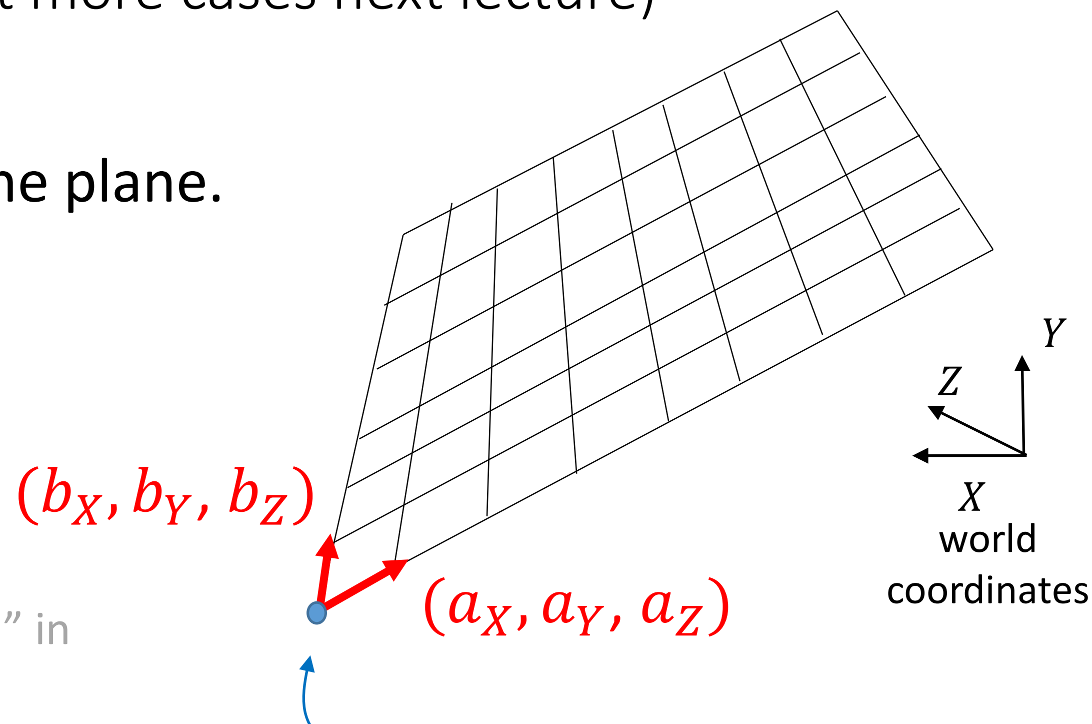
$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \boldsymbol{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

A *homography* is more general than an affine transform.
In particular, it can capture perspective effects. (Very cool.)

Case 1

(we'll look at more cases next lecture)

Suppose we have a scene plane.



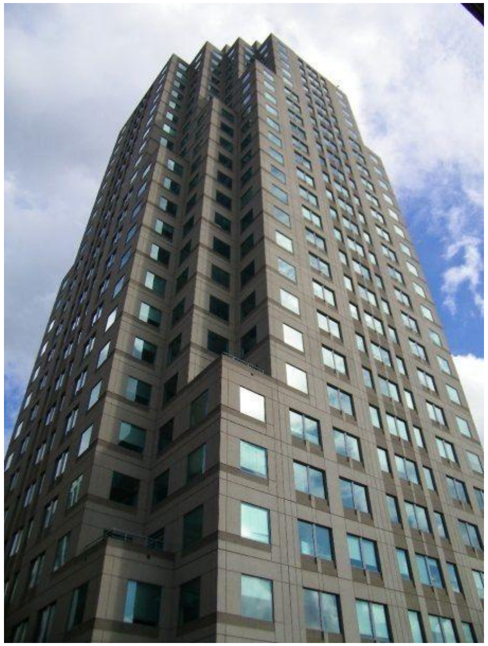
(This is used in "texture mapping" in computer graphics)

3D origin (X_0, Y_0, Z_0) corresponds to (s, t)= (0,0).

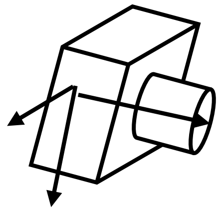
$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} a_X & b_X & X_0 \\ a_Y & b_Y & Y_0 \\ a_Z & b_Z & Z_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}$$

Examples

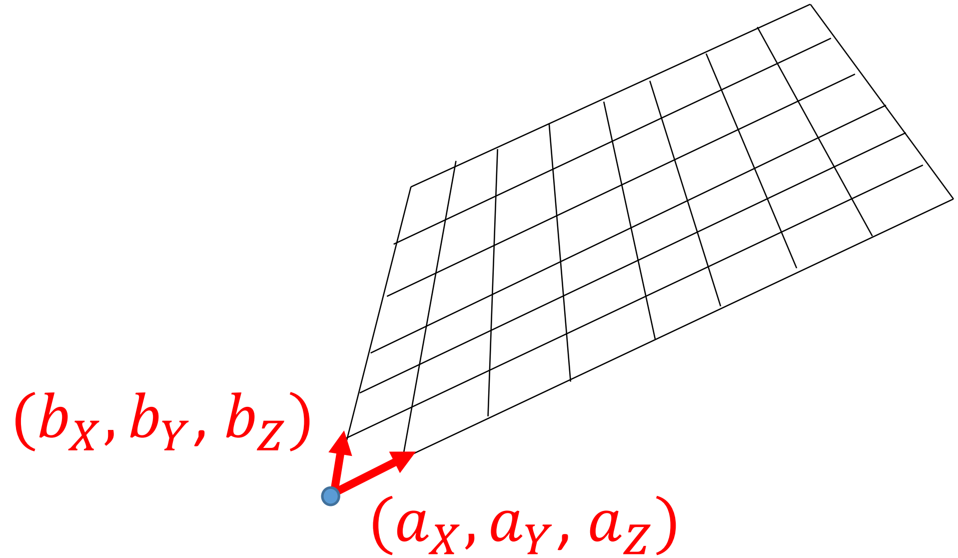
Ground plane + horizon



3 point perspective

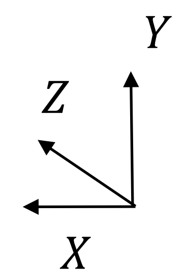


Camera with projection matrix $\boldsymbol{P}$

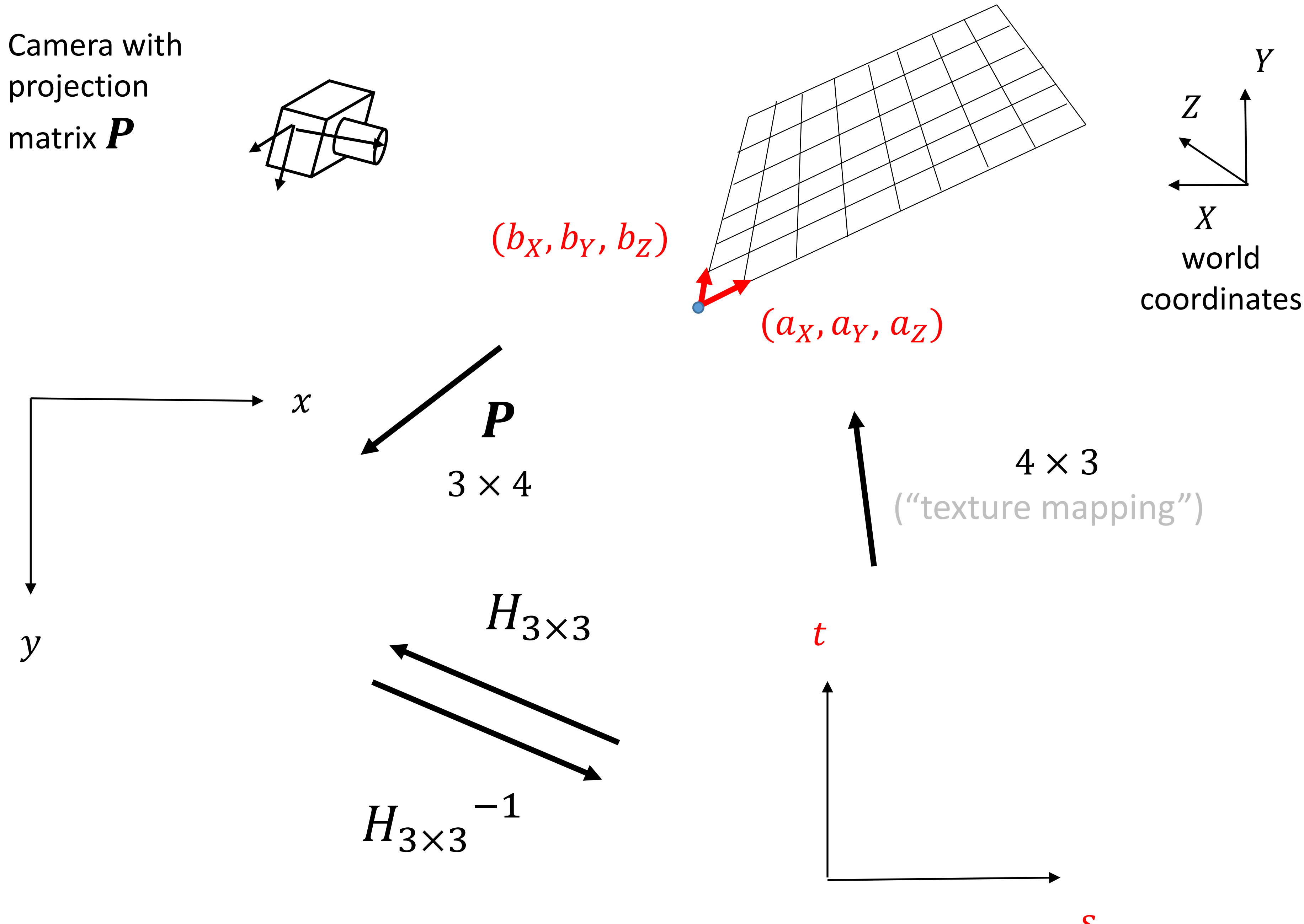


$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \underbrace{\boldsymbol{P} \begin{bmatrix} a_X & b_X & X_0 \\ a_Y & b_Y & Y_0 \\ a_Z & b_Z & Z_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{H}_{3\times 3}} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}$$

3x1 3x4 4x3 3x1



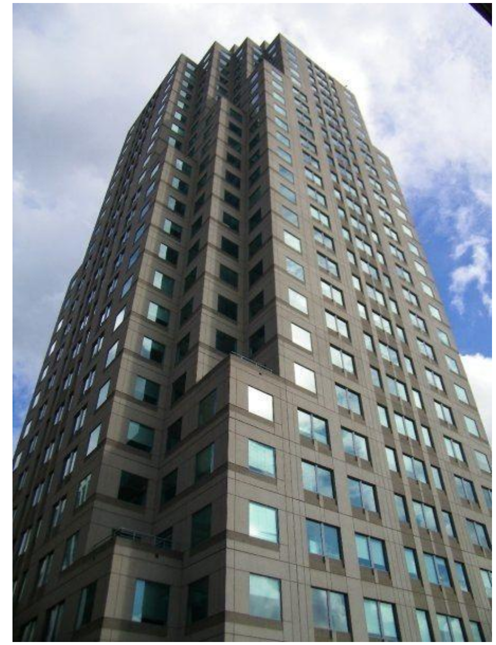
world coordinates



When is H not invertible?

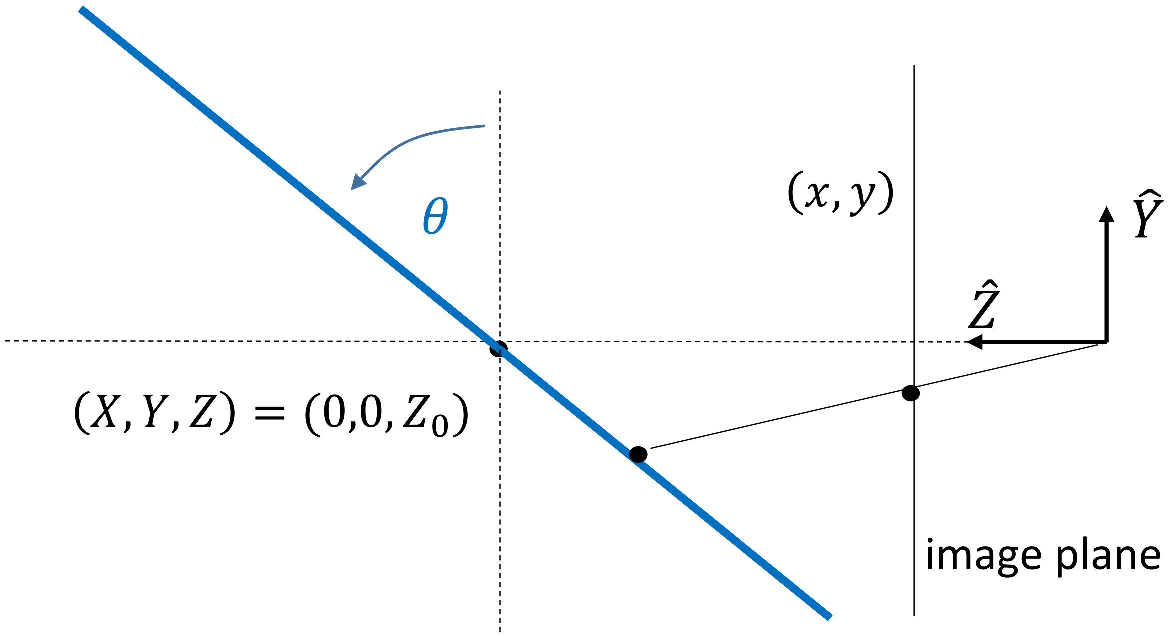
Examples

What is the homography that maps positions on the ground plane to positions in the image (or vice-versa) ?



What is the homography that maps points on *one plane* of the building façade to positions in the image (or vice-versa) ?

Exercise (see PDF)



Suppose a plane $Z = 0$ is rotated by θ degrees about the X axis, and the origin is then translated to $(0,0, Z_0)$.

What is the homography that maps 3D points $\boldsymbol{X}(s, t)$ on this scene plane to points (x, y) in the image plane and vice-versa ?

Next lecture: more on homographies

- Case 2 : Suppose we have two different cameras looking at the same plane in the scene. What is the homography mapping pixels in one image to the other image?

- Case 3 : Suppose we have one camera and we rotate it to obtain a second image. What is the homography needed to distort one image so that it aligns with the other?

- Case 4 : Suppose we have two cameras looking at a non-planar 3D scene. How can we “rectify” the cameras ?

- How to find matching points in two images ?