Tracking landmarks

points → how landmark move

⇓

how the robot move

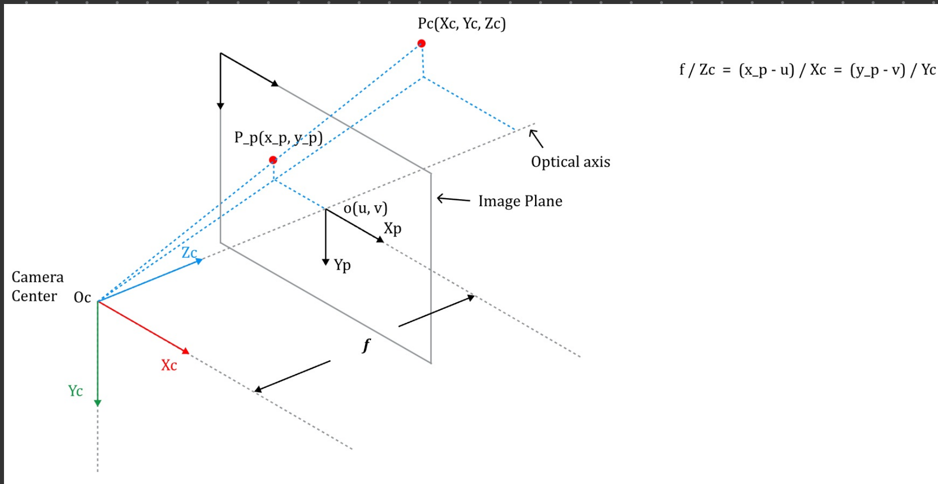
- Camera as a sensor
- Geometry of 2 camera views.
- Geometric vision algorithm
- Multi-view reconstruction
- learning-based techniques.

How to render 3D ^ into 2D ?

space

Perspective projection . lines come from a single point

vanishing point.



$$\frac{f}{Z_c} = \frac{x_p - u}{X_c} = \frac{y_p - v}{Y_c}$$

$$x_p = f\,\frac{X_c}{Z_c} + u$$

$$y_p = f\cdot\frac{Y_c}{Z_c} + v$$

f_{x_p} & f_{y_p} are not always same.

skew : distortion ⇐ undo by camera calibration

⇒ straight lines may be curve

highway X parallel

$$\underbrace{\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}}_{\text{image coordinate pixel point}} = \underbrace{\begin{bmatrix} f_{x_p} & 0 & u \\ 0 & f_{y_p} & v \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsic camera matrix}} \underbrace{\begin{bmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \\ 1 \end{bmatrix}}_{\text{original point p in real world divided by } Z_c}$$

$\vec{x}$ $\qquad$ K $\qquad$ $\frac{P}{Z_c}$

↓ property of the camera Calibration

Rearrange Z_c to left.

$$\underbrace{\boxed{\lambda \vec{x} = KP}}_{\text{depth of the point.}}$$

In robotics, we care about sensors & measurement

Some → directly measure

Other → estimate from

$K^{-1}\tilde{x} = \tilde{x}_n$ the ray from the ~~image~~ camera frame

convert pixels to rays

↓

equivalent to image coordinates

when $f = 1$, $u = v = 0$

ie. $K = I$

same image size, ↑ FOV , uncertainty of 1 pixel ↑

Field of View

estimation & limit based on resolution

Many images of the same scene, how could do better?

e.g. take images of moon from diff places

⇒ end up in the same spot ⇒ rays parallel

⇒ moon is very far.

objects that are closer, more move when go from one image to next

Two view Reconstruction

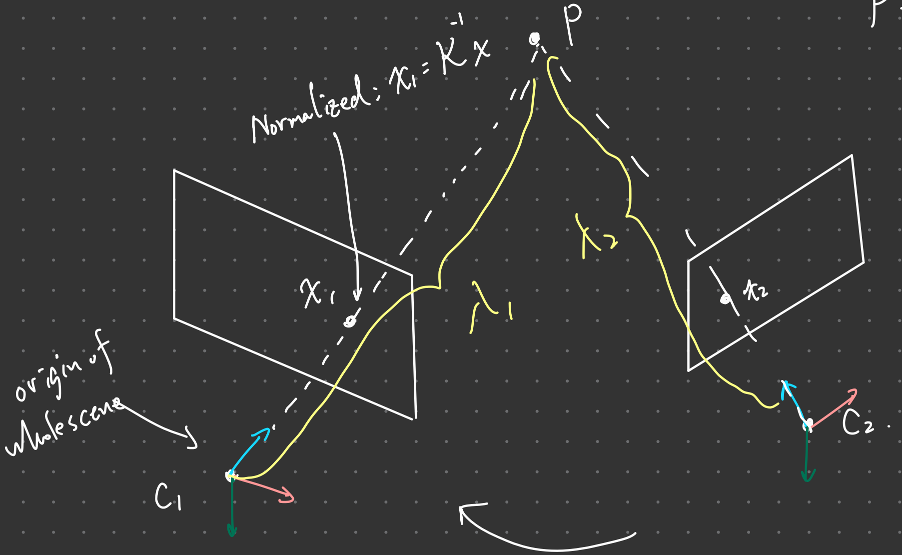
Given: ① 2 images of the same scene

② n corresponding points $(x_1^{(1)}, x_2^{(1)}), \dots, (x_1^{(n)}, x_2^{(n)})$

Find: ① $G_{21} = (R_{21}, T_{21}) \in SE(3)$ betwen 2 cameras. (motion)

② 3D locations of the n-points. (structure)

$$P = \lambda_1 \vec{x_1} = \lambda_2 \vec{x_2}$$



$$G_{21} = (R_{21}, T_{21}) \in SE(3)$$

moving the frame to the frame 2.

↓

everything defined in

1. If R_{21}, T_{21} are known, how to find 3D coords of P?

* The object shouldn't be too far compared to the distance between cameras.

Angles between the ray x too small $\Rightarrow$ $\uparrow$ uncertainty

$$G_{21} = (R_{21}, T_{21}) \in SE(3)$$

defined in frame 1

$$\lambda_1 \vec{x}_1 = P, \quad \lambda_2 \vec{x}_2 = \underline{R_{21}} P + \underline{T_{21}}$$

$$\lambda_2 \vec{x}_2 = R \lambda_1 \vec{x}_1 + T$$

$$\lambda_2 \vec{x}_2 - \lambda_1 R \vec{x}_1 = T$$

Triangulation.

$$\underset{3\times 2}{\begin{pmatrix} \vec{x}_2 & -R\vec{x}_1 \end{pmatrix}} \underset{2\times 1}{\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}} = \underset{3\times 1}{T}.$$

Overdetermined: number of equations > number of unknowns

$\Downarrow$

least square

$Ax = b$ instead solve $\min \|Ax - b\|_2^2$

Solution:

Gradient $= 0$

$\Rightarrow$ Derivative: $2A^T(Ax - b) = 0$

$\Rightarrow A^T A x = A^T b.$

$$x^* = (A^T A)^{-1} A^T b.$$

$$x^* = (A^T A)^{-1} A^T b$$