Multiple views + SLAM

two important camera models

orthographic

$$\begin{matrix} x \\ y \\ z \end{matrix} \rightarrow \begin{matrix} x \\ y \end{matrix}$$

in camera frame

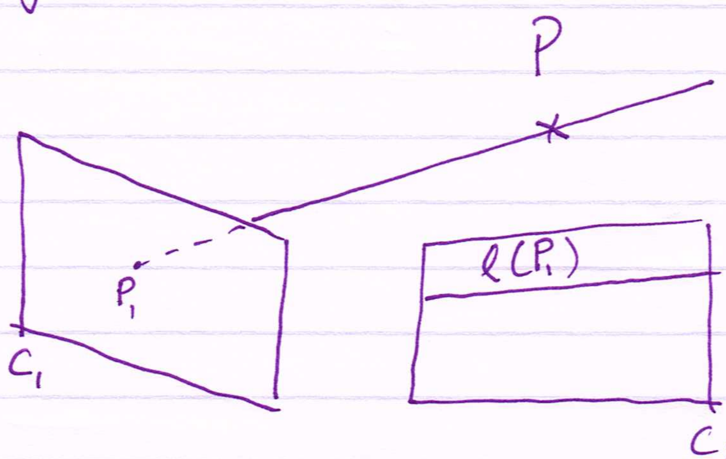
perspective

$$\begin{matrix} x \\ y \\ z \end{matrix} \rightarrow \begin{matrix} f\frac{x}{z} \\ f\frac{y}{z} \end{matrix}$$

in camera frame

Perspective cameras are often represented in <u>Homogeneous</u> <u>Coordinates</u> : (Q.V. $\xrightarrow[\text{web page}]{}$)

<u>Simplest</u> <u>geometry</u> of 2 views



P could project anywhere on this line in C_2

<u>C_1, C_2 orthographic</u> :

1) $l(P_1) \parallel l(P_2)$ if C_2 is <u>rot</u> wrt C_1

2) & if <u>no rot</u> can't recover depth

3) Many P's in C_1 csp to the same line in C_2

4) there is a map

ϕ : pts in C_1 $\longrightarrow$ lines in C_2

5) this map :

- is rank deficient (P_1, Q_1 can give same line in C_2)
- is NOT onto
  (- there are lines in C_2 not in image(ϕ) .
- follows from camera configuration

This map is useful

- constrains correspondence.
- follows from camera move

• easily estimated.

ϕ: for 2 orthographic cameras.

$$C_1: \quad \begin{matrix} x \\ y \\ z \end{matrix} \rightarrow \begin{matrix} x \\ y \end{matrix}$$

$$C_2: \quad \begin{matrix} u \\ v \\ w \end{matrix} \rightarrow \begin{matrix} u \\ v \end{matrix}$$ ← in Camera coords

and $$\begin{matrix} u \\ v \\ w \end{matrix} = R\begin{pmatrix} x \\ y \\ z \end{pmatrix} + t$$

~~the in~~ line in C_1's frame:

$$\begin{matrix} p_1 \\ 0 \end{matrix} + \lambda \begin{matrix} 0 \\ 0 \\ 1 \end{matrix}$$

in C_2's frame

$$\begin{pmatrix} r_{11}p_x + r_{12}p_y + t_x \\ r_{21}p_x + r_{22}p_y + t_y \\ r_{31}p_x + r_{32}p_y + t_z \end{pmatrix} + \lambda \begin{pmatrix} r_{13} + t_x \\ r_{23} + t_y \\ r_{33} + t_z \end{pmatrix}$$

so on C_2's image plane.

$$\begin{pmatrix} r_{11}p_x + r_{12}p_y + t_x \\ r_{21}p_x + r_{22}p_y + t_y \end{pmatrix} + \lambda \begin{pmatrix} r_{13} + t_x \\ r_{23} + t_y \end{pmatrix}$$

<u>OR</u>

$$\left[u - (r_{11}p_x + r_{12}p_y + t_x)\right](r_{23} + t_y) =$$

$$\left[v - (r_{21}p_x + r_{22}p_y + t_y)\right](r_{13} + t_x)$$

OR

$$\underset{\text{indep } P_x, P_y}{\alpha u} + \overset{\text{indep } P_x, P_y}{\beta v} + \underset{\text{linear function of } P_x, P_y}{\gamma} = 0$$

Implies we can recover ϕ from point correspondences (without reasoning about rotation, translation).

<u>have</u>

$$\alpha p^i_{2x} + \beta p^i_{2y} + \gamma_0 + \gamma_1 p^i_{1x} + \gamma_2 p^i_{2x} = 0$$

5 unknowns <u>BUT</u> homogeneous

1 eqn per correspondence

$\therefore$ 4 points should do it.

Idea:

4 point correspondences constrain all others (if correct). But we don't know which ones are right!

Strategy

- Select 4 csps A.R. from pool of plausible csps
- Get φ
- Count inliers

- Choose the best.

- <u>plausible</u>:
  - (often)
    - haven't moved too much
    - similar local features in pt rep'n

- <u>inliers</u>:
  "plausible" csps where p_2^i is "close" to line.

- <u>How many iterations</u>:
  - easy sum in binomial probs.

Process: few hyp csps + search

↓

multiple confirmed csps

↓

3D reconstruction.

3D reconstruction from multiple orth cameras

- Simplest:

  assume:
  - every camera sees every point
  - csps are known
  - cameras are orthographic

- Assume C.O.G of points is at origin
- notice that orth. proj. of COG is COG in camera frame.
- So translate origin in camera frame to COG
- We now need to estimate <u>only</u> camera orientations
- equiv

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_i = R_{w \to i} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

point in world C.S

No Translation!

coords in i'th camera frame.

so j'th pt in i'th camera:

$$r_{11_i} x_j + r_{12_i} y_j + r_{13_i} z_j$$

$$r_{21_i} x_j + r_{22_i} y_j + r_{23_i} z_j$$

usually, there is a scale so.

~~set~~ write $\underline{a}_i$, $\underline{b}_i$ vectors

where
$$\underline{a}_i^T \underline{a}_i = \underline{b}_i^T \underline{b}_i \quad \text{and}$$

$$\underline{a}_i^T \underline{b}_i = 0$$

then
$$\underline{q}_{ij} = \begin{pmatrix} \underline{a}_i^T \underline{P}_j \\ \underline{b}_i^T \underline{P}_j \end{pmatrix}$$

point j in world coords

coords of pt j in ith camera.

Construct

$$D = \begin{bmatrix} \text{Pt 1 } x \text{ in } C_1 & \cdots & 2\,x\,C_1 & \cdots \\ 1\ \ y\ \ 1 & & 2\,y\,C_1 & \\ x\ \ 2 & & 2\,x\,C_2 & \cdots \\ y\ \ 2 & & & \\ & & & \end{bmatrix}$$

But

$$D = \begin{bmatrix} a_1^T \\ b_1^T \\ a_2^T \\ b_2^T \\ \vdots \\ \vdots \end{bmatrix} \begin{bmatrix} P_1 & P_2 & \cdots \end{bmatrix}$$

$\Rightarrow$ SVD D ! (to 3d in this case!)

Notice important ambiguity

$$SVD(D) = U \Sigma V^T$$

→ go to 3D

$$= U_3 \Sigma_3 V_3^T$$

write $A = U_3 \Sigma_3^{1/2}$ $B = \Sigma_3^{1/2} V_3^T$

Now $AB \approx D$

but $A W W^{-1} B \approx D$ ← same error.

But $a_i^T b_i = 0$, $a_i^T a_i = b^T b_i$

⇒ search for W st AW

has this property.

This is the root of a very rich enterprise; <u>issues</u>

- What if matches don't appear in all frames?
- What if cameras are not orthographic
- What if views are far apart?
- What if you don't have points?
- What if cameras aren't calibrated?

- Good answers available to all - we will explore some.