Multiple views + SLAM

two important camera models

orthographic

\[
x \rightarrow x
\]
\[
y \rightarrow y
\]
\[
z
\]

in camera frame

perspective

\[
x \rightarrow \frac{fx}{z}
\]
\[
y \rightarrow \frac{fy}{z}
\]

in camera frame
Perspective cameras are often represented in 
**Homogeneous Coordinates**: \((Q:Y \rightarrow \text{web page})\)

**Simplest geometry of 2 views**

![Diagram](image)

- \(C_1, C_2\) orthographic:
  1. \(l(P_1) \parallel l(P_2)\) if \(C_2\) is not \(\text{wrt}\) \(C_1\)
  2. If no rot can’t recover depth
  3. Many \(P\)'s in \(C_1\) CSP to the same line in \(C_2\)
4) there is a map

\[ \phi : \text{pts in } C_1 \rightarrow \text{lines in } C_2 \]

5) this map:
- is rank deficient (\( p_i, q_i \) can give same line in \( C_2 \))
- is NOT onto
  - there are lines in \( C_2 \) not in image(\( \phi \))
- follows from camera configuration

This map is useful
- constrains correspondence
- follows from camera move
easily estimated.

\[ \phi: \text{ for 2 orthographic cameras.} \]

\[ C_1: x \rightarrow x \]
\[ y \rightarrow y \]
\[ z \rightarrow z \]

\[ C_2: u \rightarrow u \]
\[ v \rightarrow v \]
\[ w \rightarrow w \]

\[ v = R \begin{pmatrix} x \\ y \\ z \end{pmatrix} + t \]

\[ \text{the line in } C_1\text{'s frame:} \]
\[ p_i + \lambda o \]
\[ o \]
\[ 0 \]
\[ 1 \]
in \( C_2 \)'s frame

\[
\begin{pmatrix}
\gamma_{11} p_x + \gamma_{12} p_y + t_x \\
\gamma_{21} p_x + \gamma_{22} p_y + t_y \\
\gamma_{31} p_x + \gamma_{32} p_y + t_z
\end{pmatrix}
\quad + \quad \lambda
\begin{pmatrix}
\gamma_{13} + t_x \\
\gamma_{23} + t_y \\
\gamma_{33} + t_z
\end{pmatrix}
\]

so on \( C_2 \)'s image plane.

\[
\begin{pmatrix}
\gamma_{11} p_x + \gamma_{12} p_y + t_x \\
\gamma_{21} p_x + \gamma_{22} p_y + t_y \\
\gamma_{31} p_x + \gamma_{32} p_y + t_z
\end{pmatrix}
\quad + \quad \lambda
\begin{pmatrix}
\gamma_{13} + t_x \\
\gamma_{23} + t_y \\
\gamma_{33} + t_z
\end{pmatrix}
\]

or

\[
\begin{bmatrix}
U - (\gamma_{11} p_x + \gamma_{12} p_y + t_x) \\
\cdot (\gamma_{22} + t_y)
\end{bmatrix}
= \\
\begin{bmatrix}
V - (\gamma_{21} p_x + \gamma_{22} p_y + t_y) \\
\cdot (\gamma_{11} + t_x)
\end{bmatrix}
\]
OR

\[ \alpha u + \beta v + \gamma = 0 \]

\[ \text{indep } p_x, p_y \]

\[ \text{indep } p_x, p_y \]

linear function of

\[ \text{indep } p_x, p_y \]

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

\[ \alpha p_{1x}^i + \beta p_{1y}^i + \gamma p_{1z}^i + \gamma p_{2x}^i + \gamma p_{2y}^i = 0 \]

5 unknowns BUT homogeneous

1 eqn per correspondence

\[ \therefore 4 \text{ points should do it.} \]
Idea:

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

Strategy

- Select 4 csps AR from pool of plausible csps
  - Get P
  - Count outliers
- Choose the best.
plausible:
- (often)
- haven't moved too much
- similar local features in pt neph

inliers:
"plausible" csps where $p_j$ is "close to line".

How many iterations:
- easy sum in binomial prob.
Process: few hyp csps + search
  ↓
multiple confirmed csps
  ↓
3D reconstruction.

3D reconstruction from multiple ortho 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 C.O.G is C.O.G in camera frame.
- So translate origin in camera frame to C.O.G.
- We now need to estimate only camera orientations.

\[
\begin{pmatrix}
\mathbf{t}_d \\
\mathbf{y} \\
\mathbf{z}
\end{pmatrix}
= R_{w \rightarrow i}
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
\]

point in world c.s.

No Translational coords in i'th camera frame.
so \( n \)th pt in \( i \)th camera:

\[
\begin{align*}
\gamma_{11i} x_j + \gamma_{12i} y_j + \gamma_{13i} z_j \\
\gamma_{21i} x_j + \gamma_{22i} y_j + \gamma_{23i} z_j
\end{align*}
\]

usually, there is a scale so we write \( a_i, b_i \) vectors

where

\[ a_i^T a_i = b_i^T b_i \quad \text{and} \quad a_i^T b_i = 0 \]

then

\[
\begin{pmatrix} a_i^T P_j \\ b_i^T P_j \end{pmatrix}
\]

coords of pt \( j \) in world coords

\[ \text{in } i \text{'th camera.} \]
Construct

\[ D = \begin{bmatrix}
    P_k^{+} & \mathbf{x} & \mathbf{w} & \mathbf{c}_1 \\
    y^1 & 2 \times \mathbf{c}_1 \\
    x^2 & 2 \times \mathbf{c}_2 \\
    y^2 & \mathbf{c}_2
\end{bmatrix} \]

But

\[ D = \begin{bmatrix}
    \mathbf{a}_1^\top \\
    \mathbf{b}_1^\top \\
    \mathbf{a}_2^\top \\
    \mathbf{b}_2^\top \\
    \vdots
\end{bmatrix} \begin{bmatrix}
    \mathbf{p}_1 \\
    \mathbf{p}_2 \\
    \mathbf{p}_3 \\
    \vdots
\end{bmatrix} \]

\[ \implies \text{SVD} \quad D ! \quad (\text{to 3d in this case!}) \]
Notice important ambiguity

\[ \text{SVD}(D) = U \Sigma \sqrt{\Sigma} \]

- go to 3D

\[ = U_3 \Sigma_3 \sqrt{\Sigma_3} \]

write \[ A = U_3 \Sigma_3 \]

\[ B = \Sigma_3 \sqrt{\Sigma_3} \]

Now \[ AB \approx D \]

but \[ AWW^{-1}B \not\approx D \] \leftarrow same error.

But \[ a_i^T b_i = 0 \quad a_i^T a_i = b_i^T b_i \]

\[ \Rightarrow \text{search for } W \text{ st } AW \]

has this property.
This is the root of a very rich enterprise; issues

- 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.
Perspective cameras:

- we cannot recover all camera translation for ortho. cameras (think about sliding along rays)
- but we can for persp. cameras

Epipoles, epipolar structure and the fundamental matrix:

Notice: as \( x \) moves along ray through \( x_L \), \( x_L \) fixed, \( x_R \) along line (as in orthographic case)
\((x_L, x, x_R)\) form a plane.

\((f_L, x, f_R)\)'s plane.

\[ x_L \rightarrow \text{(line in } \mathbb{R}, \text{ given by plane } \cap \pi_R) \]

\(\rightarrow\) epipolar line
now

\[ x_{l1} \rightarrow \ell_{l1} \cap \ell_{l2} \text{ line} \]

\[ x_{l2} \rightarrow \ell_{l2} \cap \ell_{l2} \text{ line} \]

\[ \ell_{l1} \cap \ell_{l2} \text{ at } \ell_{l} \]

for any \( x_{l1}, x_{l2} \)

\[ \ell_{l} \cap \ell_{l} \]

\( \ell_{l} \) is the epipole (in right image)

where line \((f_{l}, f_{r})\) intersects

right image plane.
Notice \( l_k \) can be written as

\[
\alpha x + by + c = 0
\]

It's a line.

Because of how plane eq.

is formed.

So there is a matrix \( F \)

\[
\begin{align*}
\mathbf{x}_L^T F \mathbf{x}_R &= 0 \\
\text{where } \mathbf{x}_L &= \begin{pmatrix} x_{L_1} \\ x_{L_2} \\ 1 \end{pmatrix} \text{ etc} \\
\text{or Homogeneous coords.}
\end{align*}
\]
Why?

It maps a 2D family of pts to a 1D family of lines.