The simplest Mapping

- Cases of interest:
  - weak motion model, strong sensing
    = our car
  - fair motion model, landmarks
    = various SLAM algs
  - visual sensing, no motion model
    = S.F.M.

First case: weak motion model, strong sensing
landmarks

these could be beacons, etc:

Q: - in frame 1, I have landmark locations $x_1 \cdots x_N$

- in frame 2, I $\cdots$ $\cdots$ $y_1 \cdots y_N$

What happened to car?

Rotation + translation

so we have

known

$$R \underline{x}_i + t = \underline{y}_i$$

known.

or rather

$$\underset{R,t}{\operatorname{argmin}} \left(\sum_i (R x_i + t - y_i)^2 \right)$$

you can solve this in closed form by moment matching.

in particular

$$t = \frac{1}{N}\sum_i y_i - \frac{1}{N}\sum_i x_i = \mu_y - \mu_x$$

Rotation:

write $X = \begin{matrix} x_1^T - \mu_x^T \\ \vdots \\ x_N^T - \mu_x^T \end{matrix}$ $\quad Y = \begin{matrix} y_1^T - \mu_y^T \\ \vdots \\ y_N^T - \mu_y^T \end{matrix}$

now consider

$X^T X$ $\qquad Y^T Y$

$\llcorner_{3\times 3}$, second moments.

we must have $\| R X^T X R^T - Y^T Y \|^2$ is min

$\therefore$ ~~R~~

Recall

$$X = U \Sigma V^T$$

orthonormal

diagonal.

orthonormal

so $X^T X = V_x \Sigma^2 V_x^T$ etc.

so $\| R V_x \Sigma_x^2 V_x^T R^T - V_y \Sigma_y^2 V_y^T \|^2$

$\|$

so $\| V_y^T R V_x \Sigma_x^2 V_x^T R^T V_y - \Sigma_y^2 \|$

So $V_y^T R V_x = Id$, so $\underline{R = V_y V_x^T}$

Actually, you don't <u>need</u> 2 SVDs

consider $X^T Y = V_x \Sigma_x U_x^T \cancel{\Sigma_y} U_y \Sigma_y V_y^T$

If the points correspond, these should cancel to $I_{3\times 3}$

$= V_x \Sigma_x \Sigma_y V_y^T$

So $SVD(X^T Y)$ is good enough.

<u>Issue</u>: we don't usually have corresponding landmarks.

<u>ICP</u> = <u>iterated closest points</u>

$u_i = x_i$

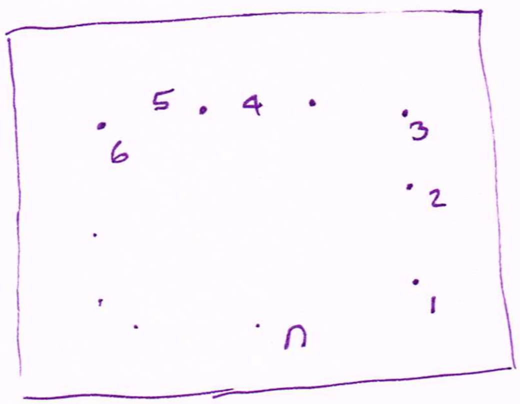
for each u_i, find closest y_i

- compute R, t from u_i, y_i
- $u_i \to R x_i + t$.

Simple, popular, effective IF point sets are close enough.

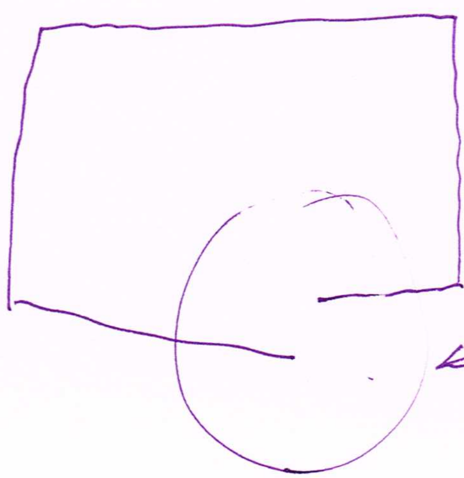
Variants from slides

Bundle adjustment



Get lidar at 1
⋮ 2
now reg to 1
⋮
now reg to 1, 2

Q: does loop close? (NO!)



aaargh!

Q: Why? – small errors accumulate.

Options: ease the R_i, t_i by:

- repeatedly taking obs. pts. from one location at random, re-est. R, t

Massive non-linear least squares

- we have correspondences for each pair of obs. locations.
- Fix one set of obs pts (i.e one location)
- register all others to
  - this
  - one another
- by least squares on $R_{u \to v}$, $t_{u \to v}$

Simple Bundle adjustment

+ assume point csp's are **known**

$$\underset{\text{coord frame}}{{}^{k}}\underline{x}_{\underset{\text{point identity}}{i}}$$

coord frame

point identity

(so we see point i in several frames)

+ we want best est. of ${}^{w}\underline{x}_i$ ← world

+ **write** $\underline{m}_i$ ← for ${}^{w}\underline{x}_i$

+ **assume** frame 1 is world

then

$$\sum_{k \in \text{frames}} \sum_{i \in \text{pts. vis in } k} \left[\mathcal{R}(\theta_k)\, {}^{k}\underline{x}_i + {}^{k}t - \underline{m}_i \right]^2 = F(\theta_k, {}^{k}t, \underline{m}_i)$$

minimize this **WRT** θ_k, ${}^{k}t$, $\underline{m}$.

iterate two phases

$$\hat{\theta}_k, \hat{t}^k = \operatorname{argmin} F(\theta_k, t^k; \hat{m}_i)$$

here we fix world points, adjust θ, t;
notice this decomposes but isn't linear

$$\hat{m}_i = \operatorname{argmin} F(\hat{\theta}_k, \hat{t}^k; m_i)$$

least squares

You can extend this to ICP if you choose

- a fixed number of points in world coords
- initialize with map constructed above
- complicate w/ IRLS,

Notice :

- the solution for R, t works if ~~points~~ distances are weighted (easy avg – think of weighted moments of inertia)

- weighted case:

  ~~μ~~ · $\mu_x = \frac{1}{N}\sum_i \sqrt{w_i}\, x_i$ · $\mu_y = \frac{1}{N}\sum_i \sqrt{w_i}\, y_i$

  · $\| R^T X^T W X R - Y^T W Y \|^2$

  where $W = \text{diag}[w_i]$

etc.

[If we care about $\sum_i w_i (R x_i + t - y_i)^2$]

Robustness:

- the square of a large number is very large.
- this means occasional large errors can dominate least squares fits
- hard to spot.

Strategy: M- estimator and IRLS.

currently:

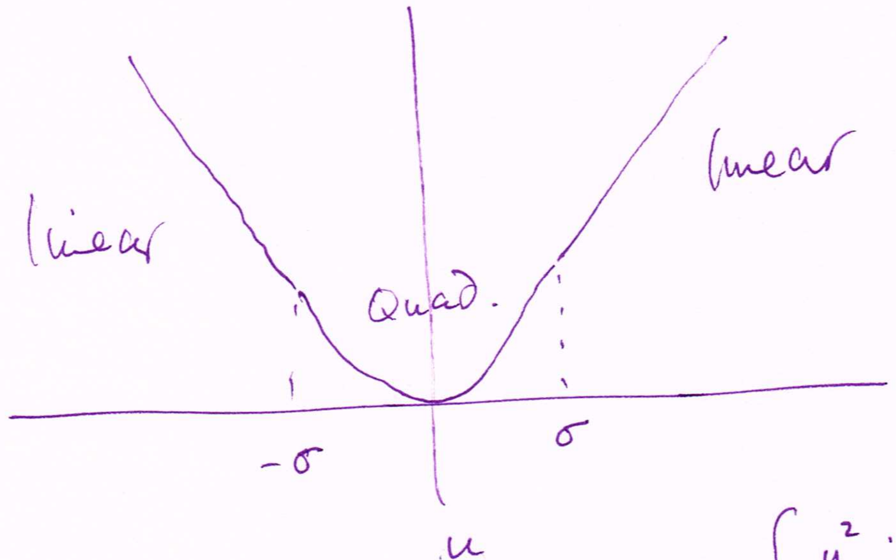
$$\min_{R,t} \frac{1}{2}\sum_i \|(Rx_i + t - y_i)\|^2 .$$

$$= \sum_i c(Rx_i + t - y_i) .$$

where $c(u) = \frac{\|u\|^2}{2}$

alternative $c(u)$ = fn of $\|u\|$ that looks like $\frac{\|u\|^2}{2}$ around 0, linear away.

Huber loss:



$$h(u) = \begin{cases} \dfrac{u^2}{2} ; & -\sigma \leq u \leq \sigma \\ \sigma |u| - \dfrac{\sigma^2}{2} & \text{otherwise} \end{cases}$$

use:

$$c(\underline{r}) = h(\|\underline{r}\|)$$

optimization:

- Iteratively reweighted least squares

~~Inner loop~~

IRLS for translation

$$\min_t \frac{1}{2}\sum_i w_i \|(y_i - x_i - t)\|^2 = \frac{1}{2}\sum_i w_i \|\delta_i - t\|^2 = F(t; w)$$

$$\text{w/ m est} \quad \min_t \sum_i h(|\delta_i - t|) = G(t).$$

now want $\nabla_t F = 0 \iff \nabla_t G = 0$

$$\nabla_t F = \sum_i w_i (t - \delta_i)$$

$$\nabla_t G = \sum_i \frac{\partial h}{\partial u} \cdot \frac{1}{2|\delta_i - t|} \cdot (t - \delta_i)$$

so $$w_i = \frac{\partial h}{\partial u} \cdot \frac{1}{2|\delta_i - t|}$$

Alg:

→ Compute w_i from t, x_i, y_i

Compute t from w_i, x_i, y_i

This is harder for rotation, because a rotation matrix <u>must satisfy constraints</u>

· <u>one strategy</u>:

- $R(\text{small angles}) \approx I + A + \frac{1}{2} A^2 \ldots$

for A <u>antisymmetric</u>.

- find an initial rotation $R^{(0)}$

now $R^{(i+1)} = R^{(i)} + R^{(i)} A$

unknown, antisymmet

now do IRLS to get A

(~~member~~ linear constraints OK.

14