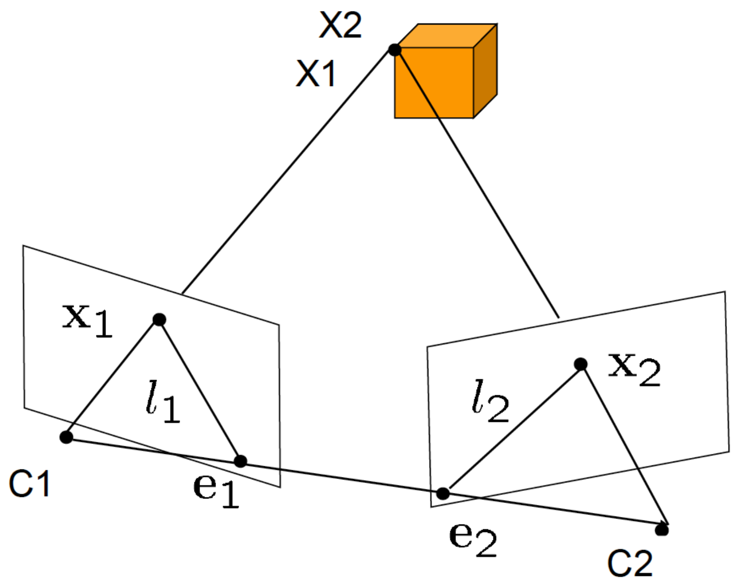
Bundle Adjustment

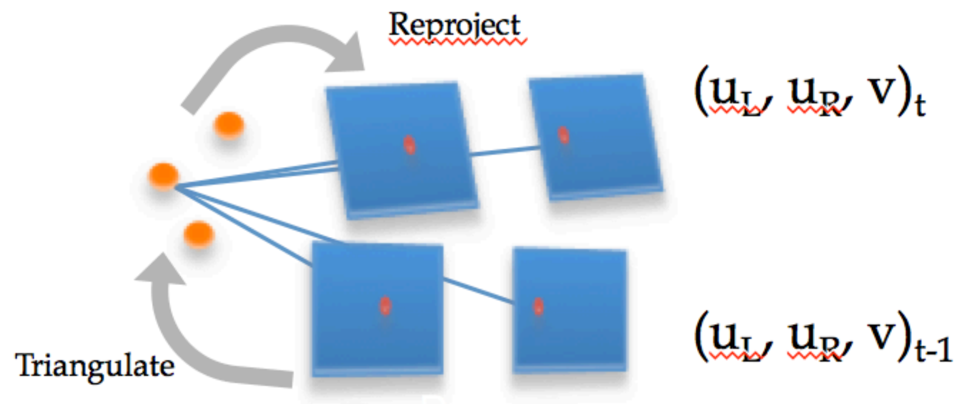
Frank Dellaert

CVPR 2014 Visual SLAM Tutorial

Motivation

- VO: just two frames -> R,t using 5-pt or 3-pt



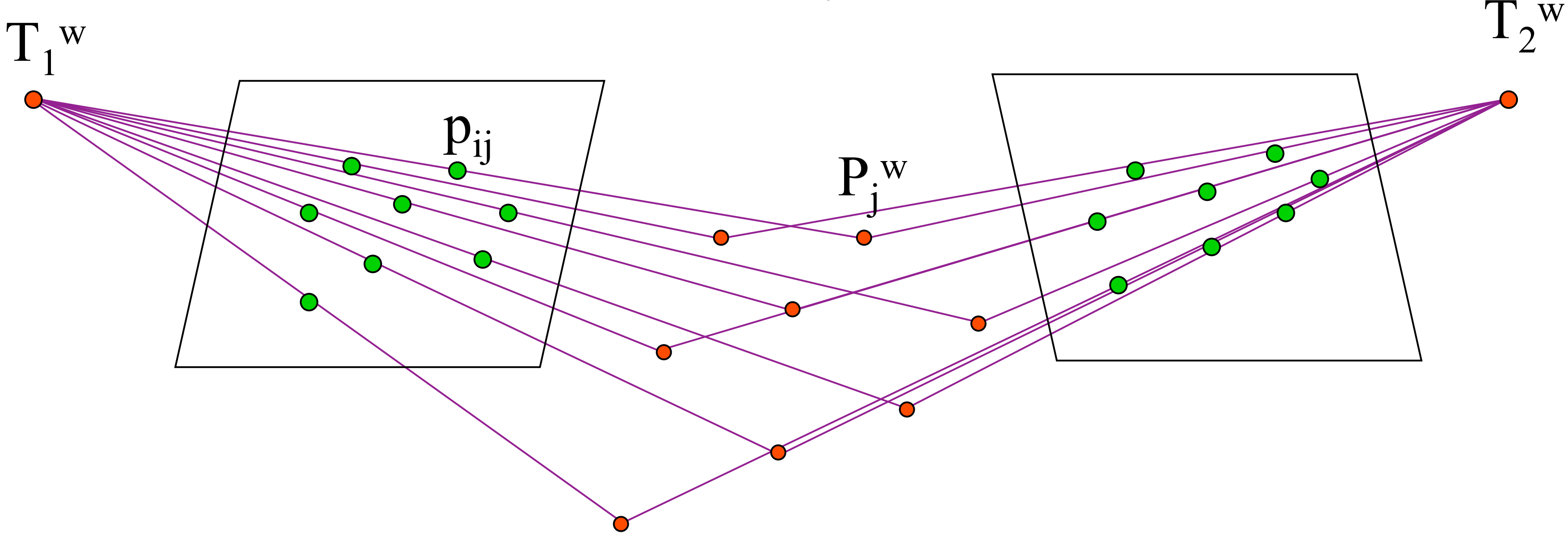


- Can we do better? SFM, SLAM -> VSLAM
- Later: integrate IMU, other sensors

Objective Function

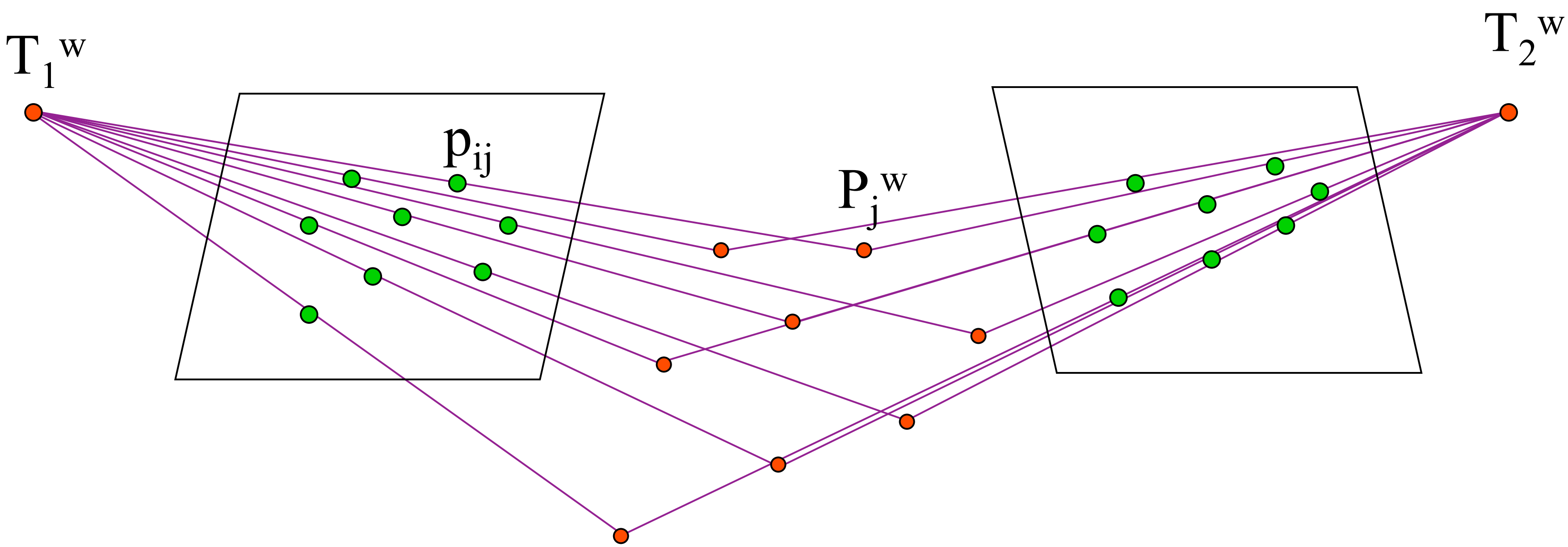
- refine VO by non-linear optimization

$$E\left(T_1^w, T_2^w, \{P_j\}\right) \triangleq \sum_{i=1,2} \sum_j \|h(T_i^w, P_j^w) - p_{ij}\|_\Sigma^2$$

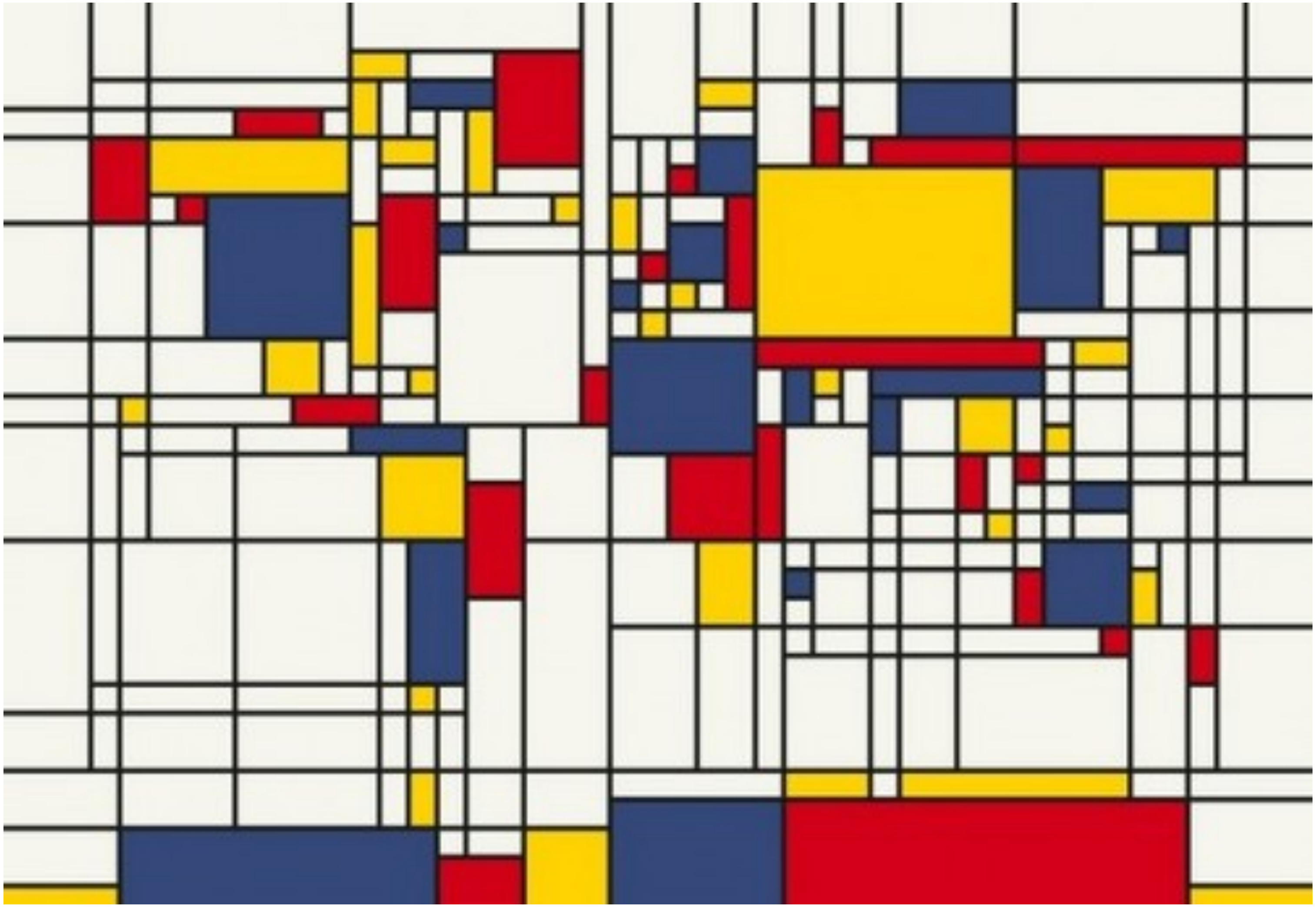


Two Views

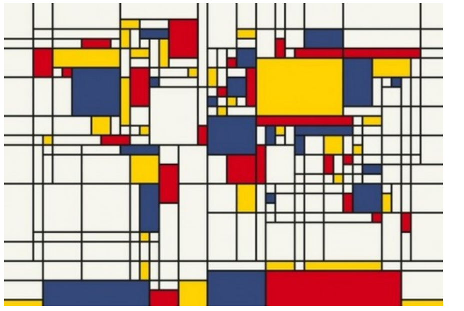
- Unknowns: poses and points $T_1^w, T_2^w, \{P_j\}$
- Measurements p_{ij}: normalized (x,y), known K!



If we lived in a Linear World:



In a Linear World…



- Linear measurement function:

$$\hat{p}_{ij} = h_{ij}\left(\xi_i, \delta_j\right) \stackrel{\Delta}{=} F_{ij}\xi_i + G_{ij}\delta_j$$

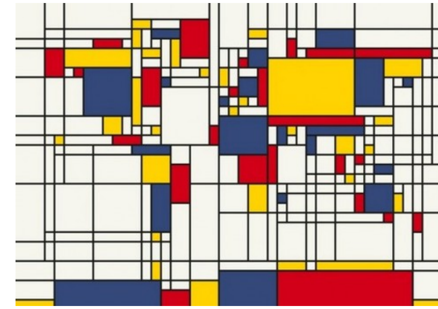
- …and objective function:

$$E\left(\xi_1, \xi_2, \{\delta_j\}\right) \stackrel{\Delta}{=} \sum_{i=1,2} \sum_j \|F_{ij}\xi_i + G_{ij}\delta_j - b_{ij}\|_\Sigma^2$$

- Linear least-squares !

- Note: ξ_i = 6D, δ_j = 3D

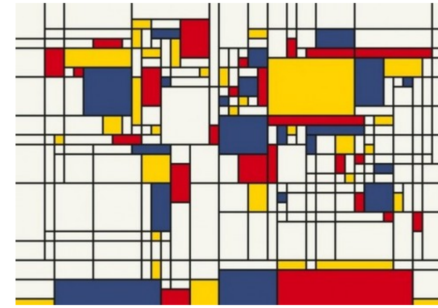
Sparse Matters



- Rewrite as $E\left(\xi_1, \xi_2, \{\delta_j\}\right) \triangleq \|Ax - b\|^2_{\Sigma'}$ where

$$A = \begin{bmatrix} F_{11} & & G_{11} & & \\ F_{12} & & & G_{12} & \\ F_{13} & & & & G_{13} \\ & F_{21} & G_{21} & & \\ & F_{22} & & G_{22} & \\ & F_{23} & & & G_{23} \end{bmatrix}, \quad x = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix}, \quad b = \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{14} \\ b_{15} \\ b_{16} \end{bmatrix}$$

Normal Equations



- Least-squares criterion

$$E\left(\xi_1,\xi_2,\{\delta_j\}\right) \stackrel{\Delta}{=} \|Ax-b\|^2_{\Sigma'}$$

- Take derivative, set to zero:

$$\left(A'A\right)x = A'b$$

- Solve using cholmod, GTSAM…
- In MATLAB: x=A\b

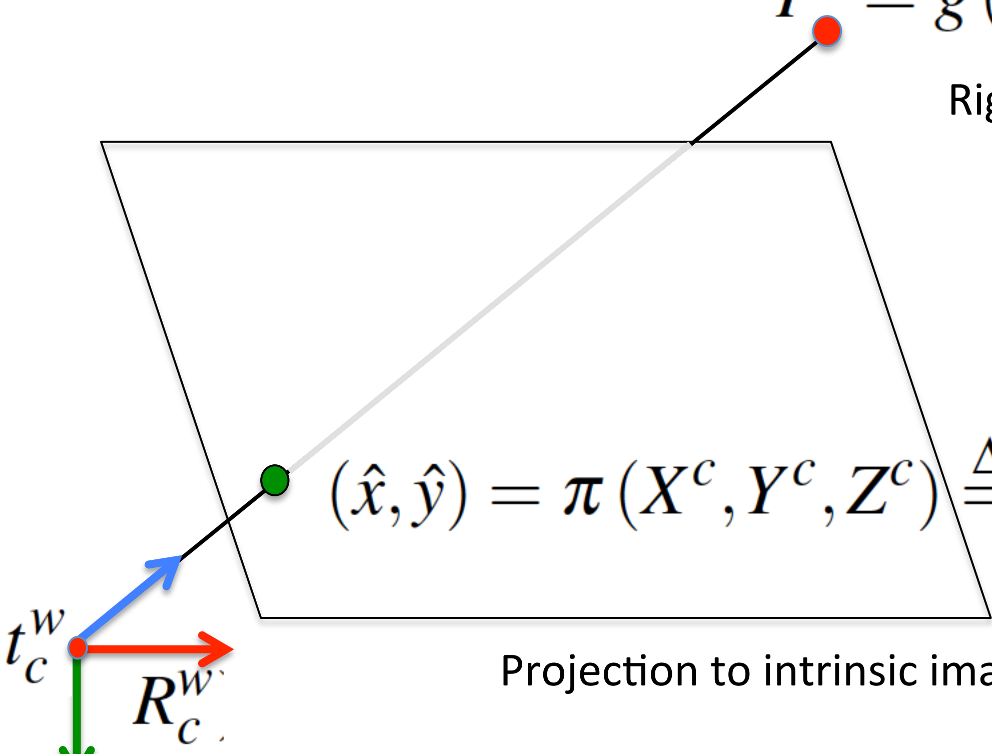
Generative Model

- Measurement Function, calibrated setting!

$$\hat{p} = h\left(T_c^w, P^w\right) \triangleq \pi\left(g\left(T_c^w, P^w\right)\right)$$

$$P^c = g\left(T_c^w, P^w\right) \triangleq \boxed{\left(R_c^w\right)^T \left(P^w - t_c^w\right)}$$

Rigid 3D transform to camera frame



$$(\hat{x}, \hat{y}) = \pi\left(X^c, Y^c, Z^c\right) \triangleq \boxed{\left(\frac{X^c}{Z^c}, \frac{Y^c}{Z^c}\right)}$$

Projection to intrinsic image coordinates

Taylor Expansion Epic Fail

- Taylor expansion?

$$h(T_c^w + \xi, P^w + \delta) \approx h(T_c^w, P^w) + F\xi + G\delta.$$

Taylor Expansion Epic Fail

- Taylor expansion?

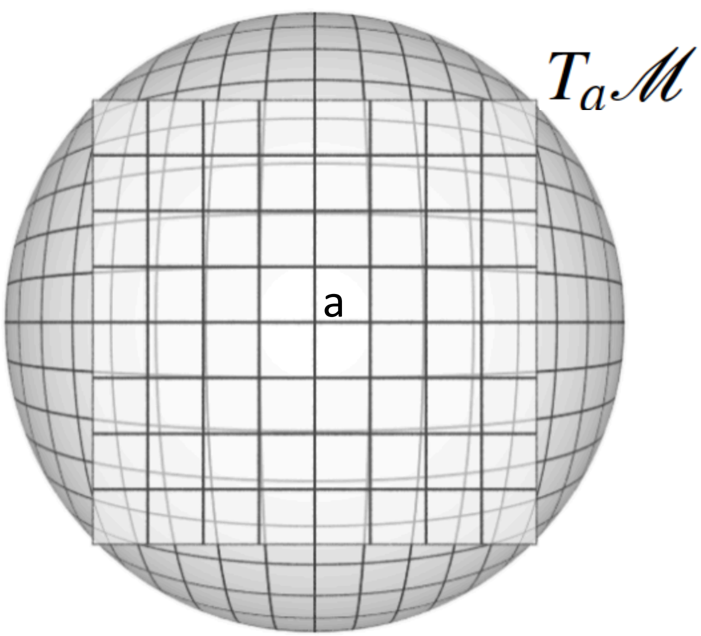
$$h(T_c^w + \xi, P^w + \delta) \approx h(T_c^w, P^w) + F\xi + G\delta.$$

- Oops: $T_c^w + \xi$?

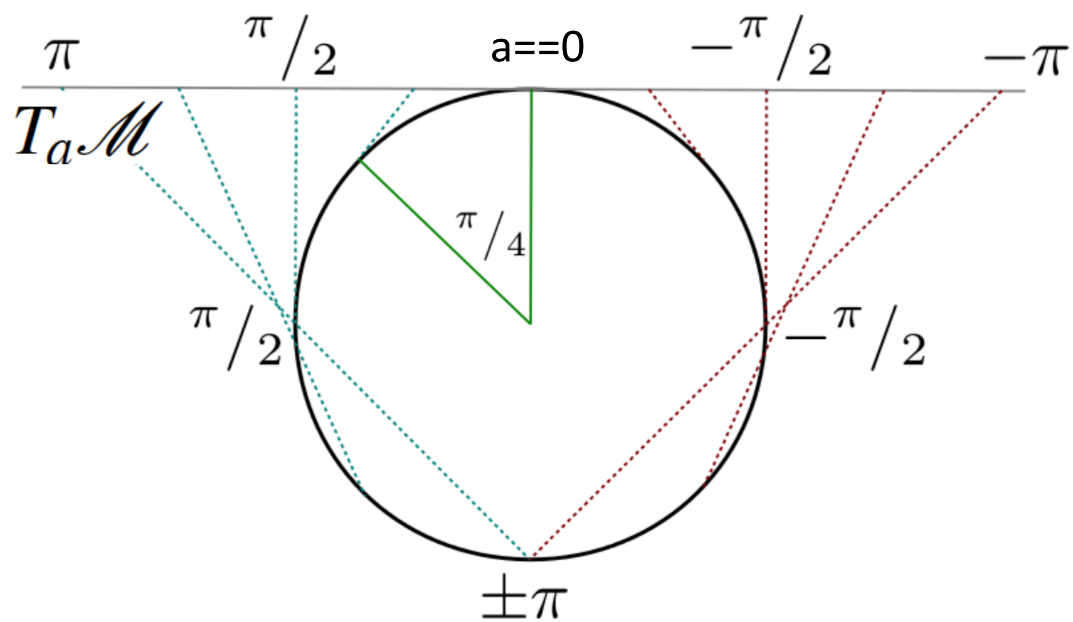
- T is a 4x4 matrix, but is over-parameterized!
- T in SE(3): only 6DOF (3 rotation, 3 translation)

Tangent Spaces

- An incremental change on a manifold $\mathscr{M}$ can be introduced via the notion of an n-dimensional tangent space $T_a\mathscr{M}$ at a
- Sphere SO(2)

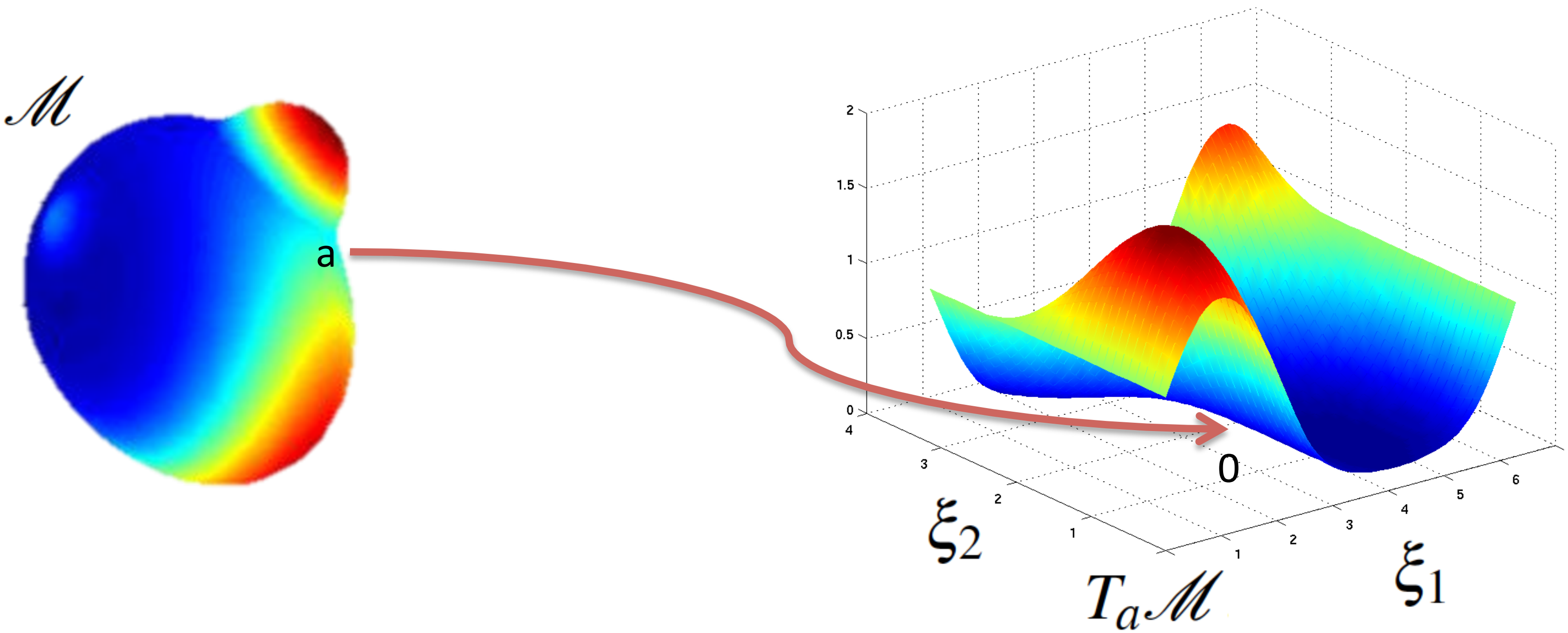


Thanks to Steven Lovegrove



Tangent Spaces

- Provides local coordinate frame for manifold



SE(3): A Twist of Li(m)e

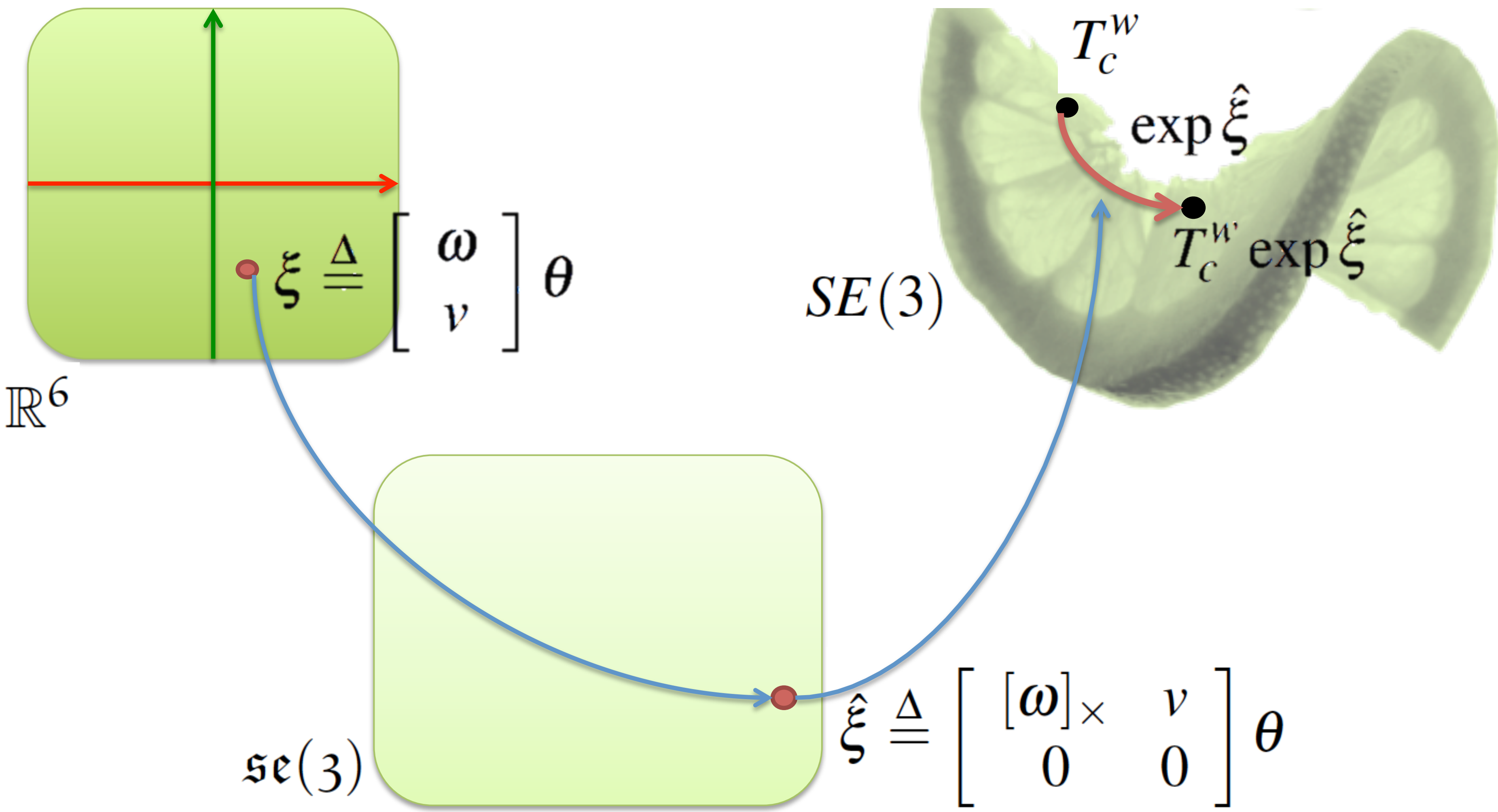
- **Lie group** = group + manifold
  - *SE(3)* is group!
  - *SE(3)* is 6DOF manifold embedded in R^{4*4}
- For Lie groups, we have exponential maps:

$$T_c^w \leftarrow T_c^w \exp \hat{\xi}$$

- se<2> **twist**:

$$\xi \triangleq \begin{bmatrix} \omega \\ v \end{bmatrix} \theta \longrightarrow \hat{\xi} \triangleq \begin{bmatrix} [\omega]_\times & v \\ 0 & 0 \end{bmatrix} \theta \longrightarrow \exp \hat{\xi}$$

Exponential Map for SE(3)



Generators for SE(3)

$$G^1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} G^2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} G^3 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$G^4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} G^5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} G^6 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\xi \triangleq \begin{bmatrix} \omega \\ v \end{bmatrix} \theta \rightarrow \hat{\xi} \triangleq \begin{bmatrix} [\omega]_\times & v \\ 0 & 0 \end{bmatrix} \theta = \boxed{\sum_{i=1}^{6} \xi_i G^i}$$

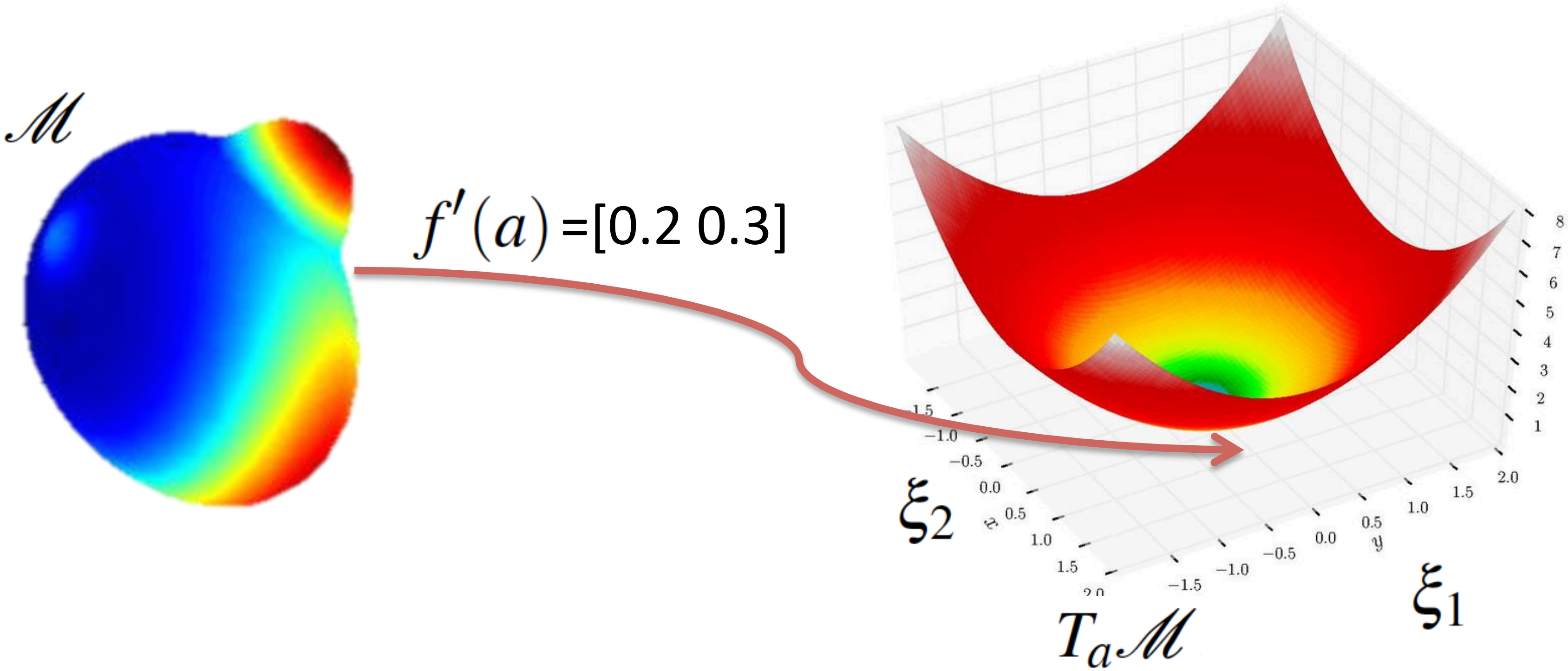
Exponential map closed form:

$$\exp\left(\widehat{\begin{bmatrix} \omega \\ v \end{bmatrix}} t \right) = \begin{bmatrix} e^{[\omega]_\times \theta} & \left(I - e^{[\omega]_\times \theta}\right)(\omega \times v) + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix}$$

Generalized Taylor Expansion

- Define f'(a) to satisfy:

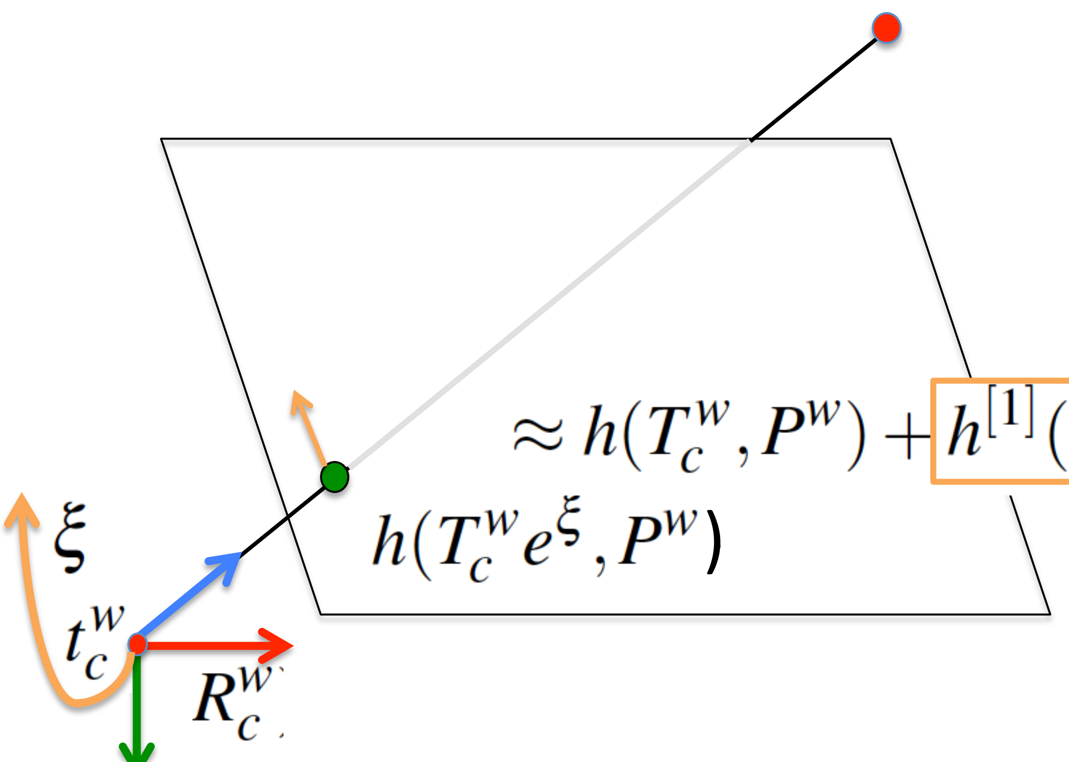
$$f(ae^{\hat{\xi}}) \approx f(a) + f'(a)\xi$$



Taylor Expansion for Projection

- Projection: function of two variables,

$$\hat{p} = h\left(T_c^w, P^w\right) \stackrel{\Delta}{=} \pi\left(g\left(T_c^w, P^w\right)\right)$$



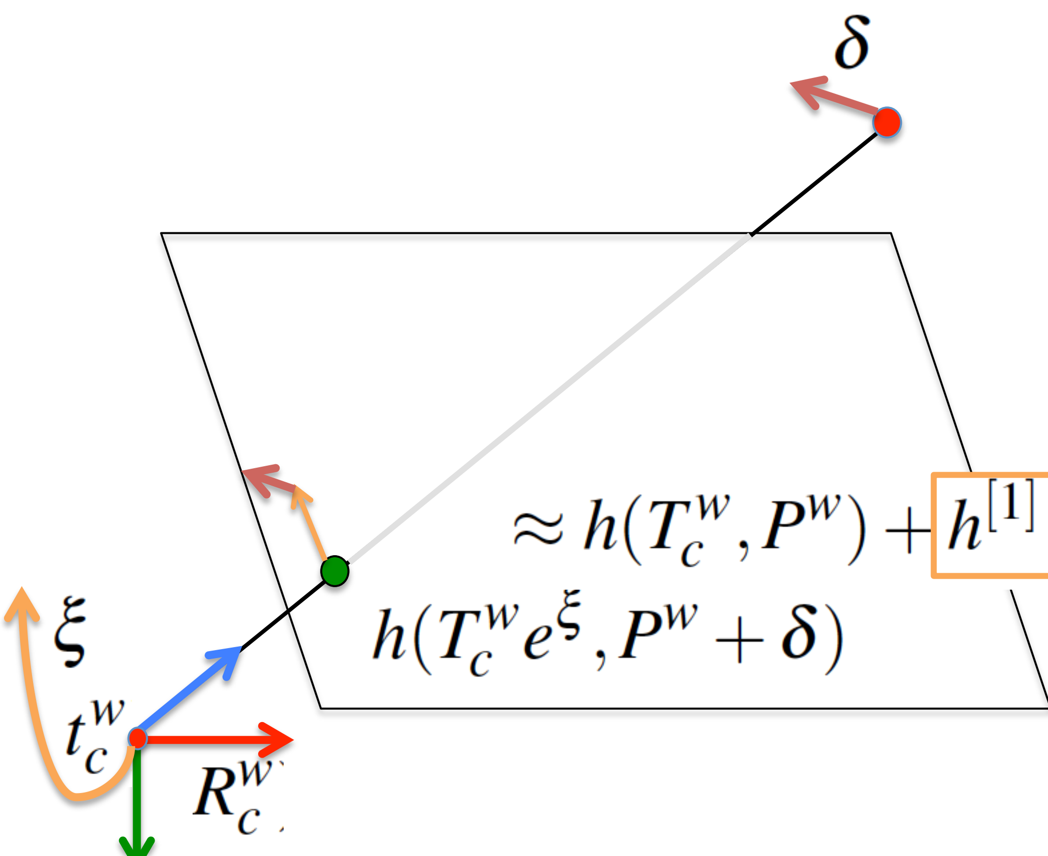
$$\approx h(T_c^w, P^w) + \underbrace{\boxed{h^{[1]}(T_c^w, P^w)}}_{2\times 6}\xi$$

$$h(T_c^w e^{\xi}, P^w)$$

Taylor Expansion for Projection

- Projection: function of two variables,

$$\hat{p} = h\left(T_c^w, P^w\right) \stackrel{\Delta}{=} \pi\left(g\left(T_c^w, P^w\right)\right)$$

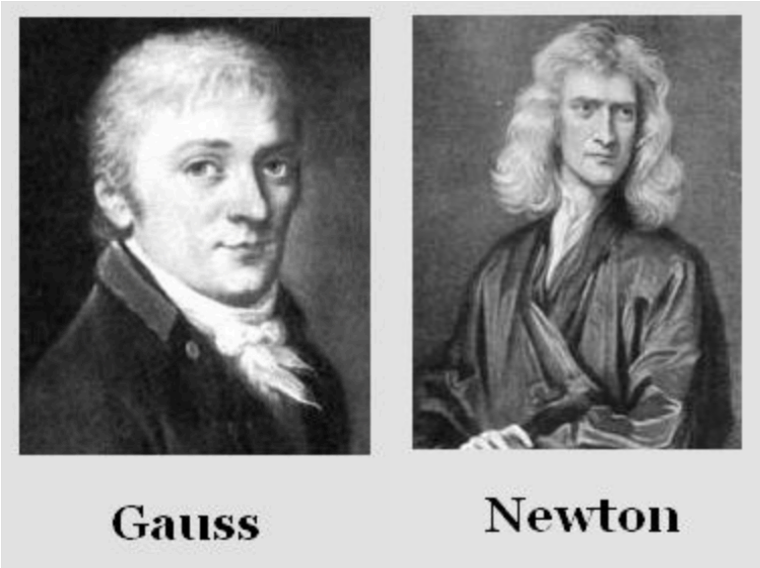


$$h(T_c^w e^{\xi}, P^w + \delta) \approx h(T_c^w, P^w) + \overset{\text{2x6}}{\boxed{h^{[1]}(T_c^w, P^w)}}\,\xi + \overset{\text{2x3}}{\boxed{h^{[2]}(T_c^w, P^w)}}\,\delta$$

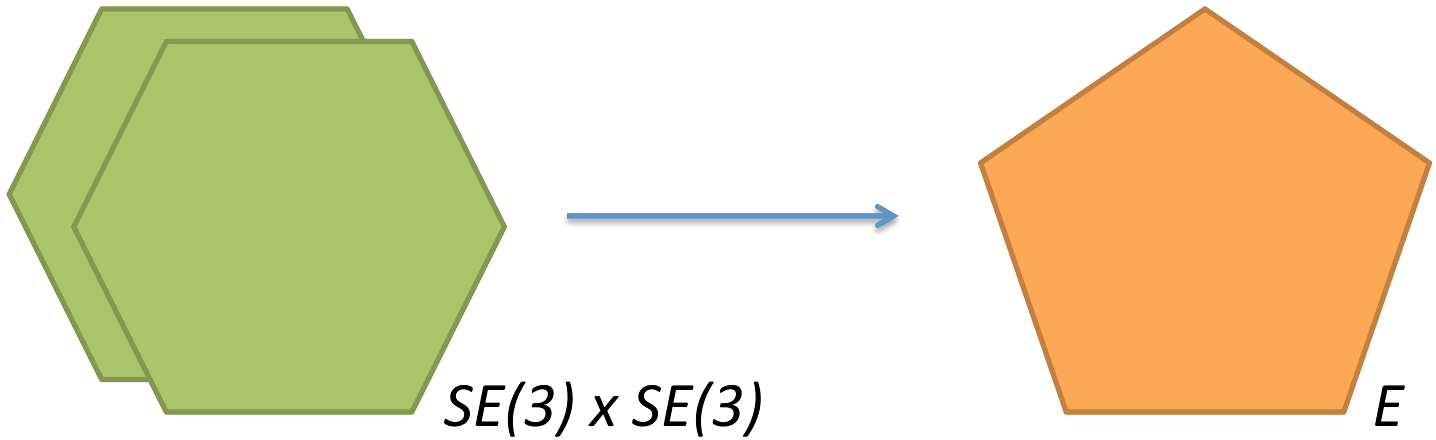
Gauss-Newton

- Iteratively Linearize, solve normal equations on tangent space, update Lie group elements

1. Start with a good initial estimate $\theta^0 = T_1^w, T_2^w, \{P_j\}$
2. Linearize (1) around θ^{it} to obtain A and b
3. Solve for $x = \xi_1, \xi_2, \{\delta_j\}$ using the normal equations
$$(A'A)\,x = A'b$$
where $A'A$ the Gauss-Newton approximation
4. Update the nonlinear estimate θ^{it+1}
   (a) $T_1^w \leftarrow T_1^w \exp \hat{\xi}_1$
   (b) $T_2^w \leftarrow T_2^w \exp \hat{\xi}_2$
   (c) $P_j \leftarrow P_j + \delta_j$
5. If not converged, go to 2.



Too much Freedom!



- A'A will be singular! 7DOF **gauge freedom**
  - Switch to 5DOF Essential Manifold
  - Use photogrammetry "inner constraints"
  - Add prior terms
  - Fuse in other sensors, e.g., IMU/GPS

Levenberg-Marquardt Algorithm

- **Idea:** Add a damping factor

$$\left(A'A + \lambda D\right) x = A'b$$

- What is the effect of this damping factor?
  - Small λ ?
  - Large λ ?

Slide by Dr. Jürgen Sturm,
Computer Vision Group, TUM

Levenberg-Marquardt Algorithm

- **Idea:** Add a damping factor

$$(A'A + \lambda D)\, x = A'b$$

- What is the effect of this damping factor?
  - Small λ → same as least squares
  - Large λ → steepest descent (with small step size)
- **Algorithm**
  - If error decreases, accept $\Delta \mathbf{x}$ and reduce λ
  - If error increases, reject $\Delta \mathbf{x}$ and increase λ

Slide by Dr. Jürgen Sturm,
Computer Vision Group, TUM

Linearizing Re-projection Error

- Chain rule: $h^{[1]}(T_c^w, P^w) = \pi'(p_c) \cdot g^{[1]}(T_c^w, P^w)$

$$h^{[2]}(T_c^w, P^w) = \pi'(p_c) \cdot g^{[2]}(T_c^w, P^w)$$

$$\pi(X^c, Y^c, Z^c) \triangleq \left(\frac{X^c}{Z^c}, \frac{Y^c}{Z^c}\right) \longrightarrow \pi'(p_c) = \frac{1}{Z^c}\begin{bmatrix} 1 & 0 & -X^c/Z^c \\ 0 & 1 & -Y^c/Z^c \end{bmatrix}$$

$$g(T_c^w, P^w) \triangleq (R_c^w)^T (P^w - t_c^w) \longrightarrow g^{[2]}(T_c^w, P^w) = (R_c^w)^T$$

$$g\left(T_c^w e^{\hat{\xi}}, P^w\right) = P^c + P^c \times \omega - v$$

$$\longrightarrow g^{[1]}(T_c^w, P^w) = \begin{bmatrix} [P^c]_\times & -I_3 \end{bmatrix}$$

Linearizing Re-projection Error

- Chain rule: $$h^{[1]}(T_c^w, P^w) = \pi'(p_c) \cdot g^{[1]}(T_c^w, P^w)$$
$$h^{[2]}(T_c^w, P^w) = \pi'(p_c) \cdot g^{[2]}(T_c^w, P^w)$$

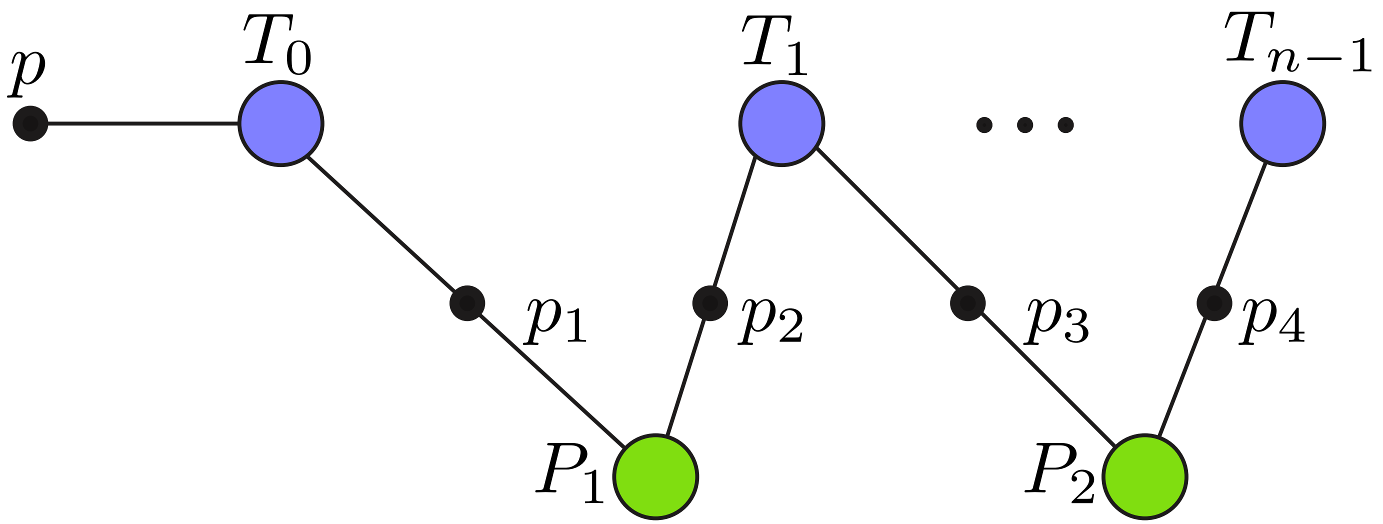
$$\boxed{\begin{aligned} h^{[1]}(T_c^w, P^w) &= \frac{1}{Z^c} \begin{bmatrix} 1 & 0 & -\hat{x} \\ 0 & 1 & -\hat{y} \end{bmatrix} \begin{bmatrix} [P^c]_\times & -I_3 \end{bmatrix} \\ h^{[2]}(T_c^w, P^w) &= \frac{1}{Z^c} \begin{bmatrix} 1 & 0 & -\hat{x} \\ 0 & 1 & -\hat{y} \end{bmatrix} (R_c^w)^T \end{aligned}}$$

Multiple frames = Full BA

- Simple to extend. Typically not fully connected:

$$E\left(\{T_i^w\},\{P_j\}\right) \triangleq \sum_k \|h(T_{i_k}^w, P_{j_k}^w) - p_k\|_\Sigma^2$$

- Factor graph representation:



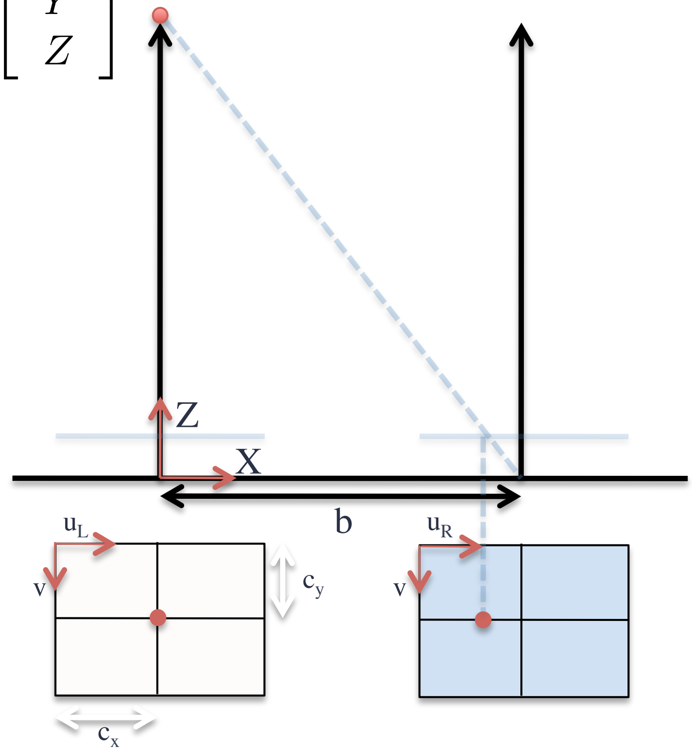
SFM Packages

- SBA: pioneer
- Google Ceres: great at large-scale BA
- GTSAM (Georgia Tech Smoothing and Mapping)
  - Has iSAM, iSAM2, ideal for sensor fusion
  - Factor-graph based throughout:

```
%% Add factors for all measurements
noise = noiseModel.Isotropic.Sigma(2,measurementNoiseSigma);
for i=1:length(Z),
    for k=1:length(Z{i})
        j = J{i}{k};
        G.add(GenericProjectionFactorCal3_S2(
                Z{i}{k}, noise, symbol('x',i), symbol('p',j), K));
    end
end
```

Stereo Camera Projection Model

$$l = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$



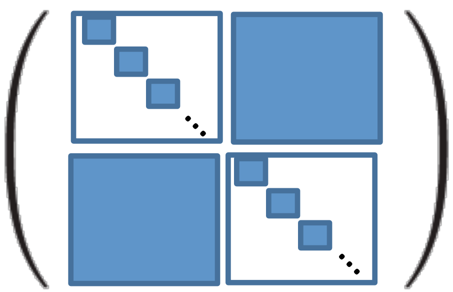
$$K = \begin{bmatrix} f & & c_x \\ & f & c_y \\ & & 1 \end{bmatrix}$$

Projection equations: $u_L = f\frac{X}{Z} + c_x \qquad u_R = f\frac{X-b}{Z} + c_x$

Due to rectification $v_L = v_R = v$ $\qquad v = f\frac{Y}{Z} + c_y$

Primary Structure

- **Insight:** $H_{\mathbf{cc}}$ and $H_{\mathbf{pp}}$ are block-diagonal (because each constraint depends only on one camera and one point)



$$\begin{pmatrix} \Delta\mathbf{c} \\ \Delta\mathbf{p} \end{pmatrix} = \begin{pmatrix} -J_{\mathbf{c}}^{\top}\mathbf{e}_{\mathbf{c}} \\ -J_{\mathbf{p}}^{\top}\mathbf{e}_{\mathbf{p}} \end{pmatrix}$$

- This can be solved using the Schur Complement

Schur Complement

- Given: Linear system

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}$$

- If D is invertible, then (using Gauss elimination)

$$\begin{aligned} (A - BD^{-1}C)\mathbf{x} &= \mathbf{a} - BD^{-1}\mathbf{b} \\ \mathbf{y} &= D^{-1}(\mathbf{b} - C\mathbf{x}) \end{aligned}$$

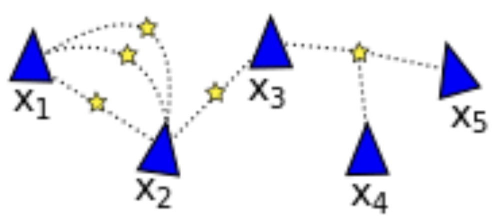
- **Reduced complexity**, i.e., invert one $p \times p$ and $p \times p$ matrix instead of one $(p+q) \times (p+q)$ matrix

Smart Factors

Vision-based navigation:

Smart factors approach:

- A smart factor for each 3D landmark
- At each iteration of nonlinear optimization
  1. SF **triangulates** the point, given camera poses
  2. SF eliminates the point via Schur complement
  3. One only needs to solve a small system including the camera poses



- **Fast(er), Robust**

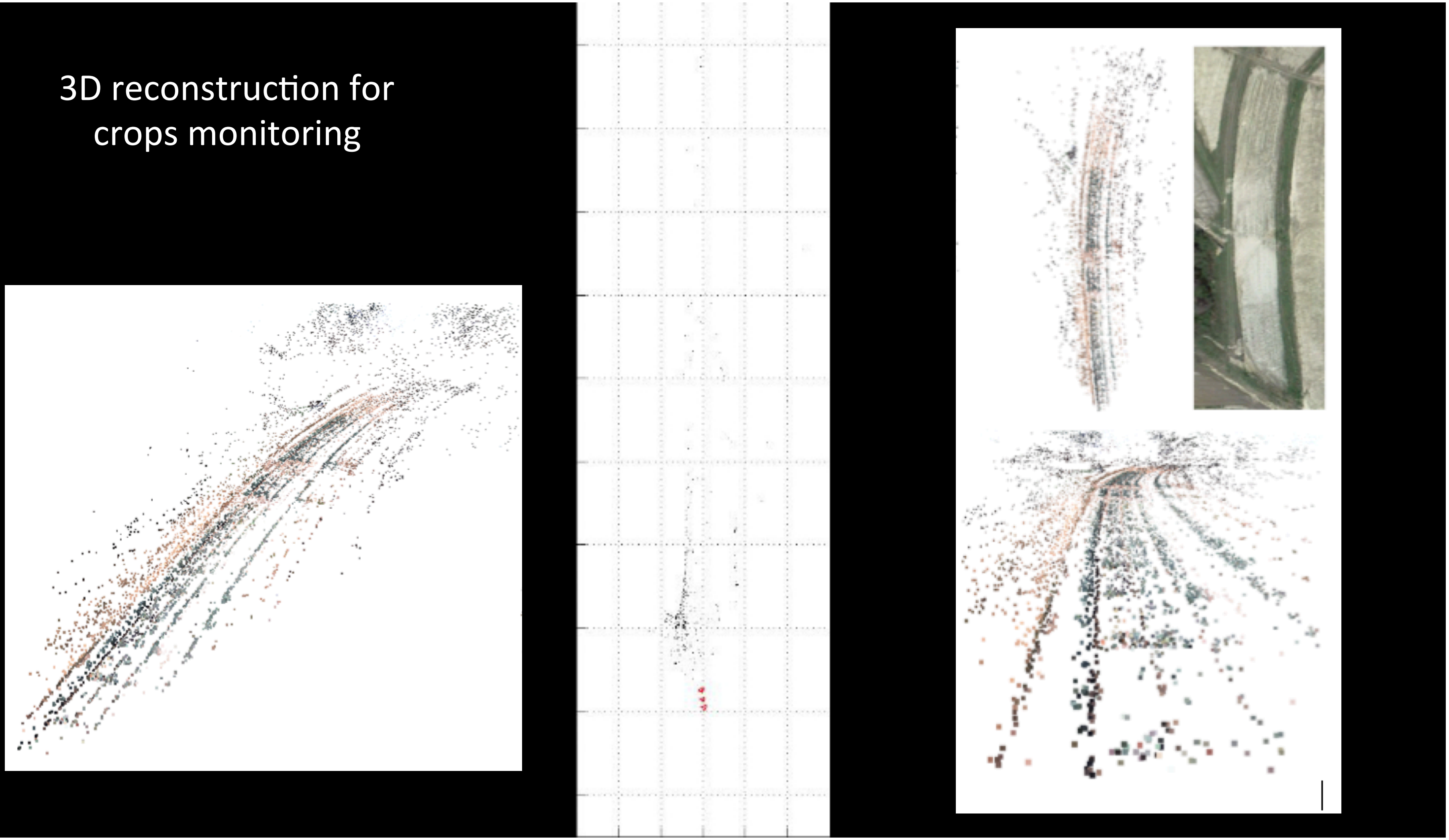
We can easily manage degenerate instances ..

e.g., if the triangulation is degenerate the smart factor can generate a different potential (only on rotations)

.. and we can do outlier rejection inside each factor!

Smart Factors

3D reconstruction for crops monitoring



NEW RELEASE: GTSAM 3.1

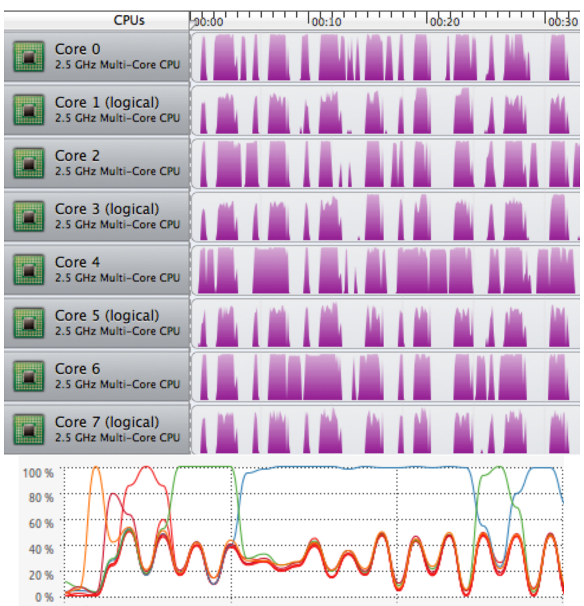
Georgia **T**ech **S**moothing **A**nd **M**apping

Download: collab.cc.gatech.edu/borg/gtsam

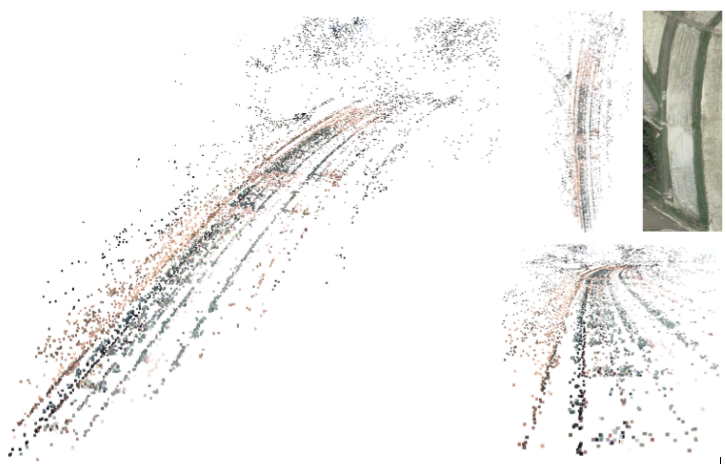
Includes numerous performance improvements and features:

- Multi-threading with TBB (funded by DARPA)
- Smart Projection Factors for SfM
- LAGO Initialization for planar SLAM (Luca Carlone et. al)

Multi-threading

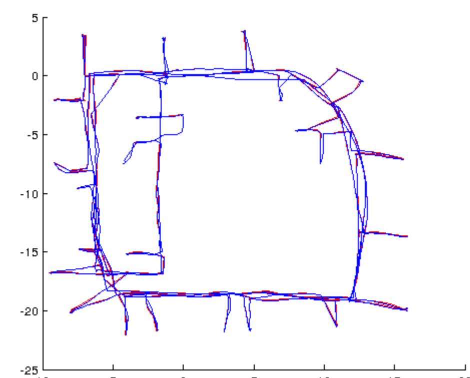


Smart Factors



Lago

Linear Approximation
for Graph Optimization



Bibliography

- Hartley and Zisserman, 2004
- Murray, Li & Sastry, 1994
- Absil, Mahony & Sepulchre, 2007