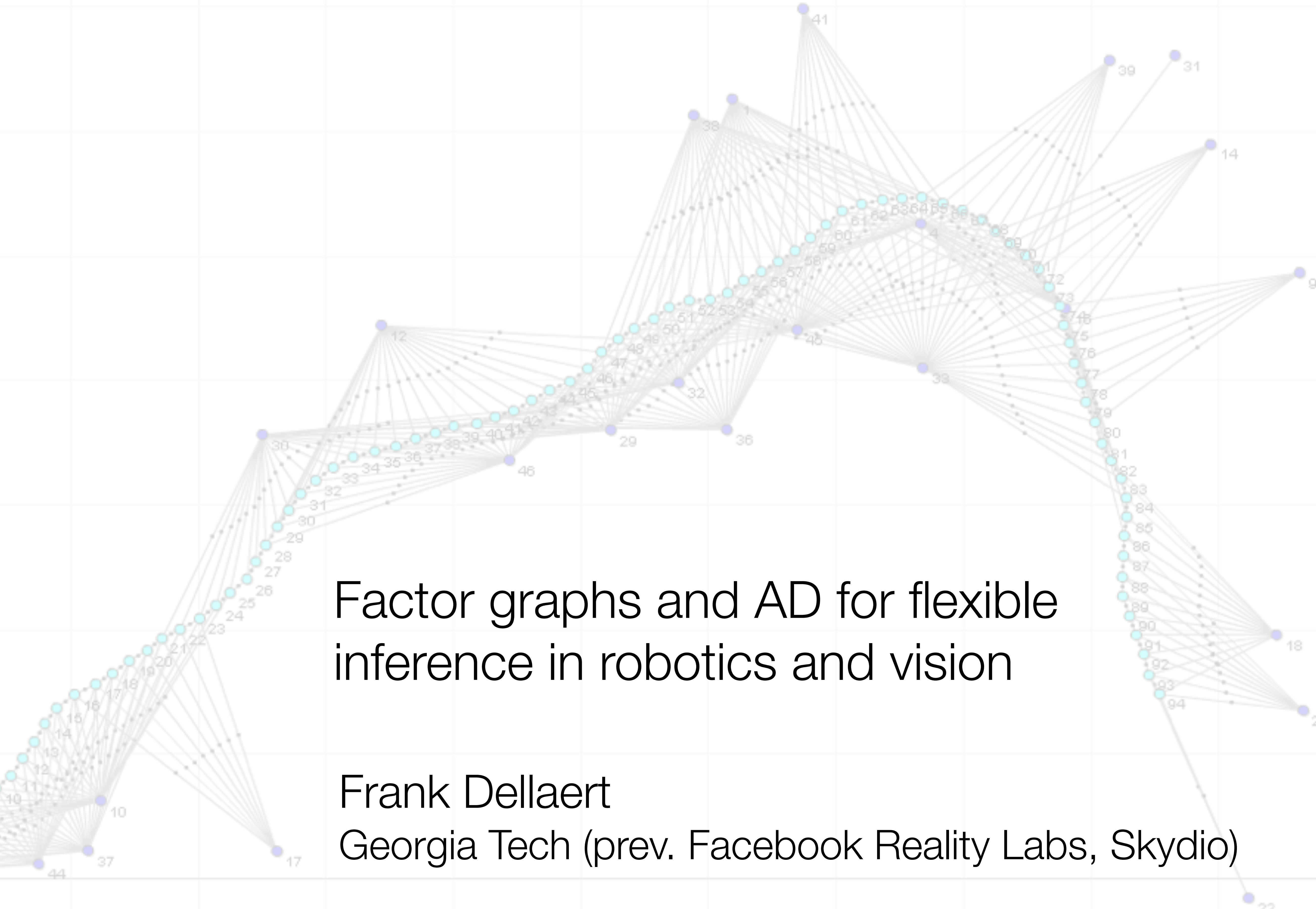


Factor graphs and AD for flexible inference in robotics and vision

Frank Dellaert

Georgia Tech (prev. Facebook Reality Labs, Skydio)

Outline

Introduction

Factor Graphs

Manifold Optimization

Plug and Play AD

Conclusion

Outline

Introduction

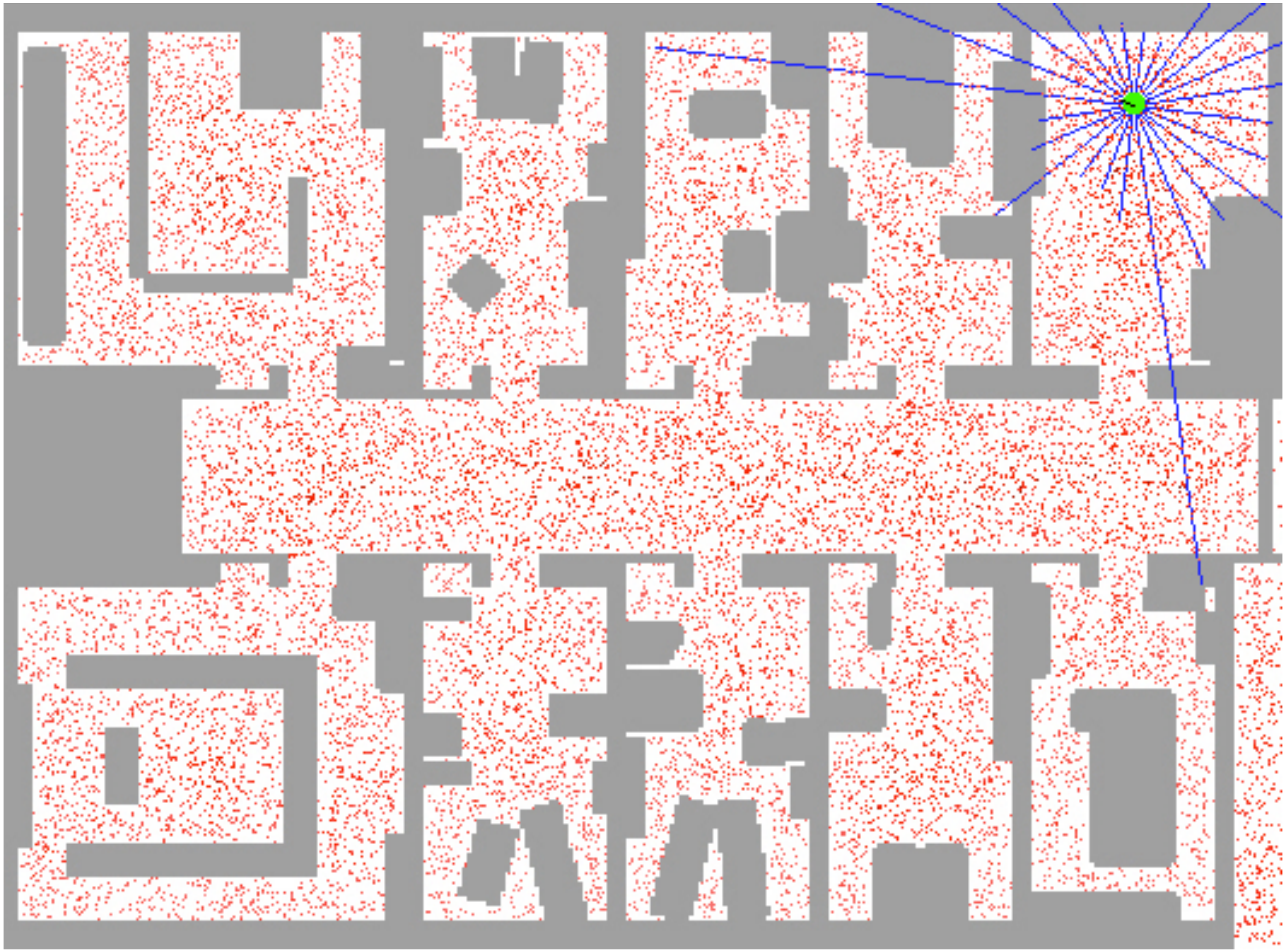
Factor Graphs

Manifold Optimization

Plug and Play AD

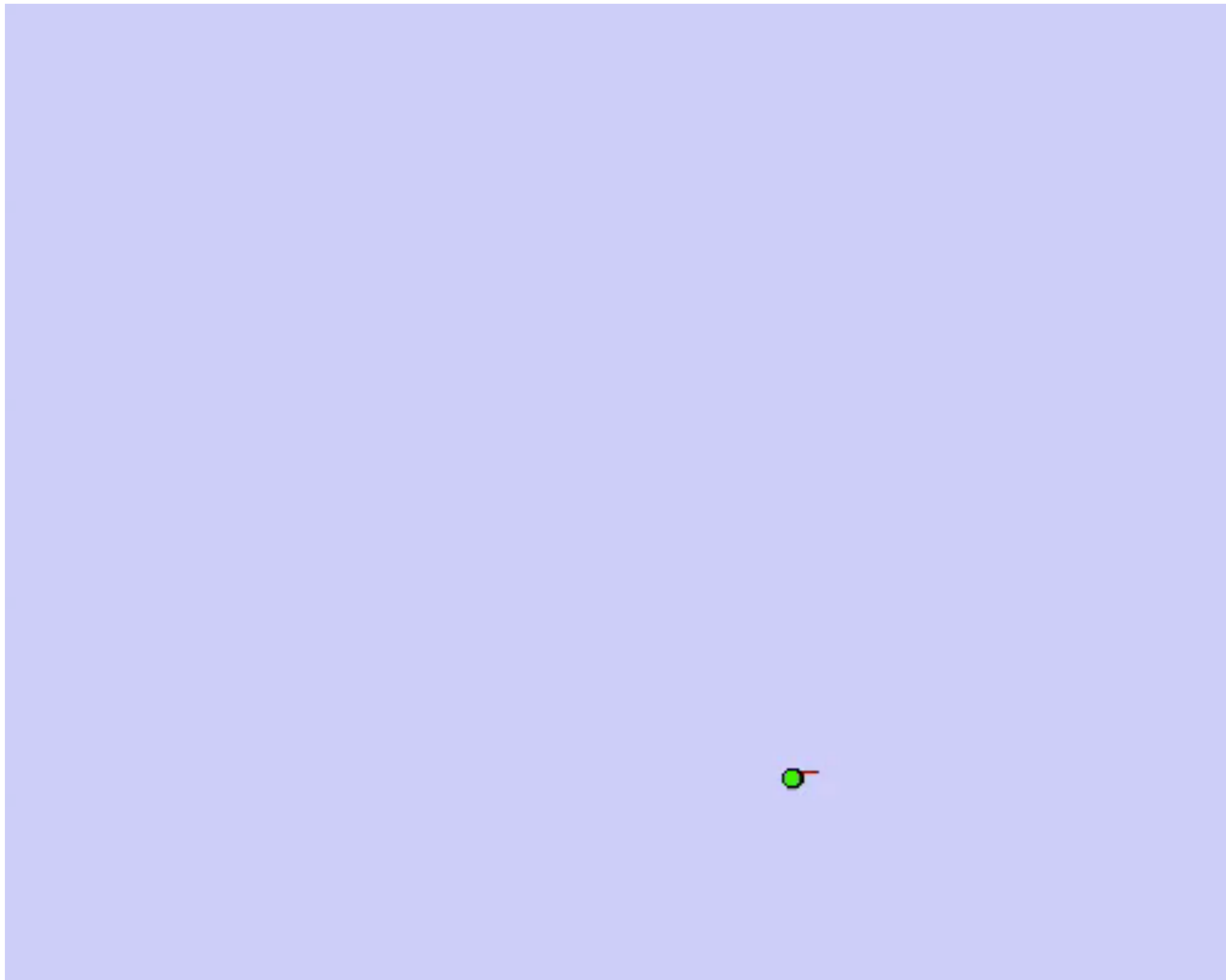
Conclusion

Monte Carlo Localization, at Carnegie Mellon!



Dellaert, Fox, Burgard & Thrun, ICRA 1999
Fox, Dellaert, Burgard & Thrun, AAAI 1999

FastSLAM: Particle Filter on Trajectories

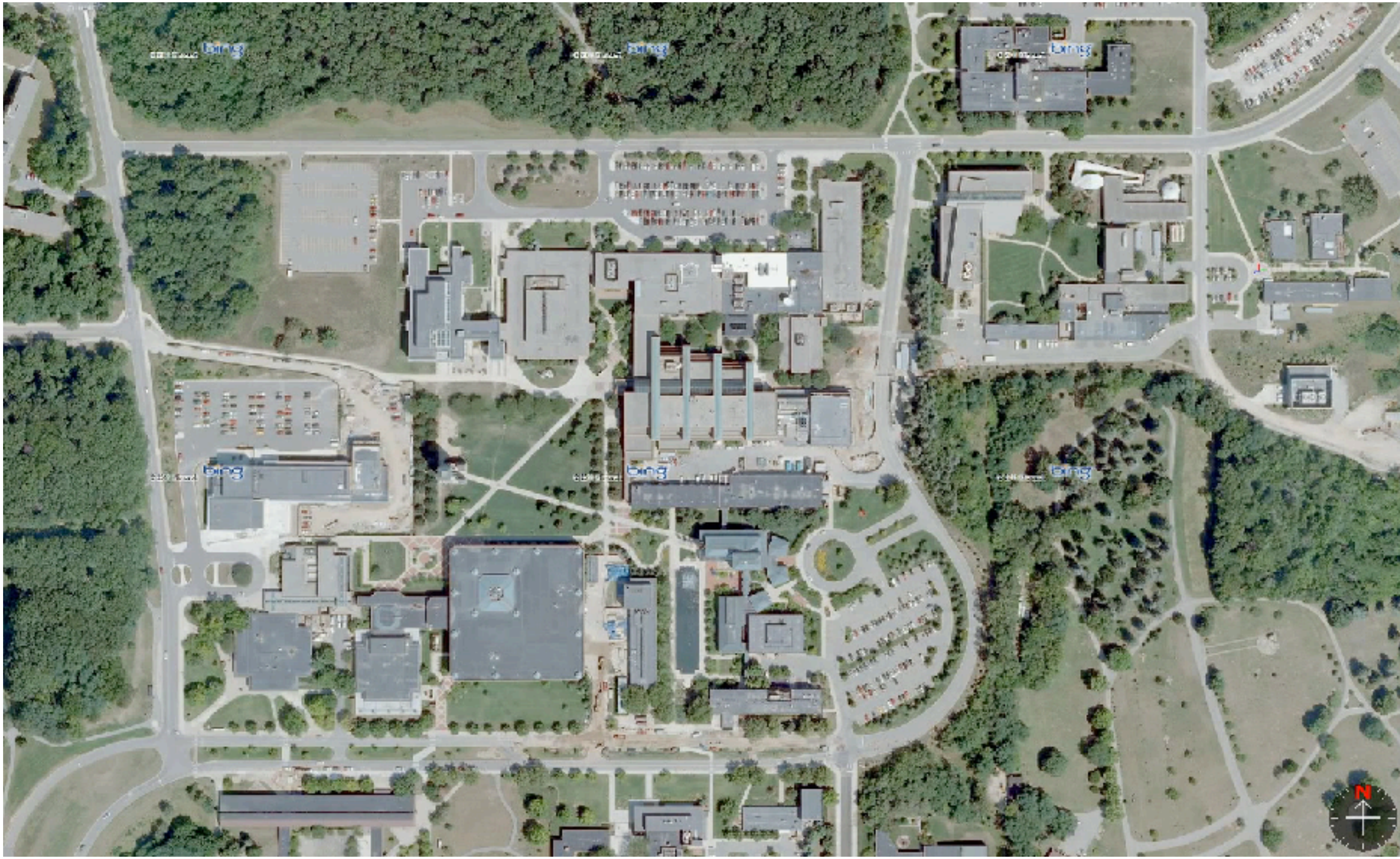


Montemerlo, Thrun, Koller, & Wegbreit, AAAI 2002

Figure by Mike Montemerlo

2013: Full 3D LIDAR Mapping

perl The Perceptual Robotics Laboratory at the University of Michigan



Data/Movie by Nick Carlevaris-Bianco and Ryan Eustice (U. Michigan)

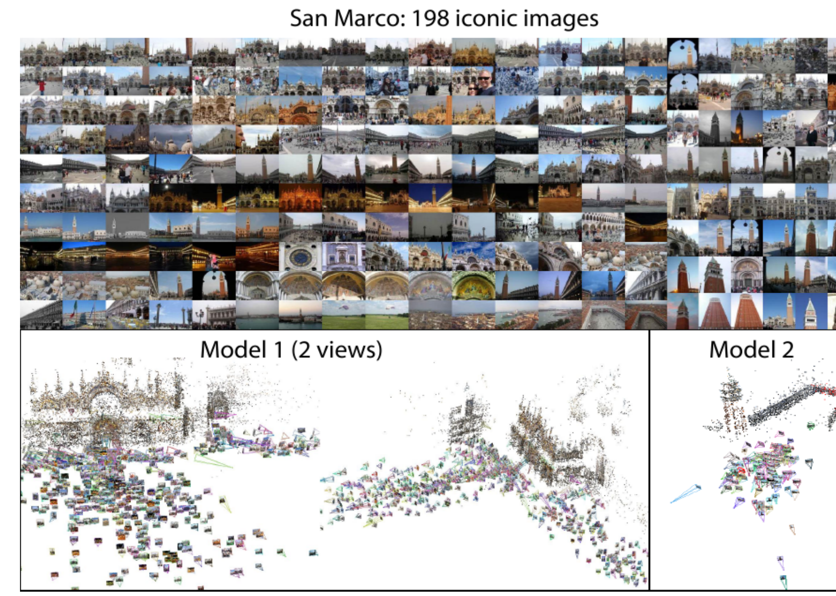
Large-Scale Structure from Motion

Photo Tourism

[Snavely et al. SIGGRAPH'06]

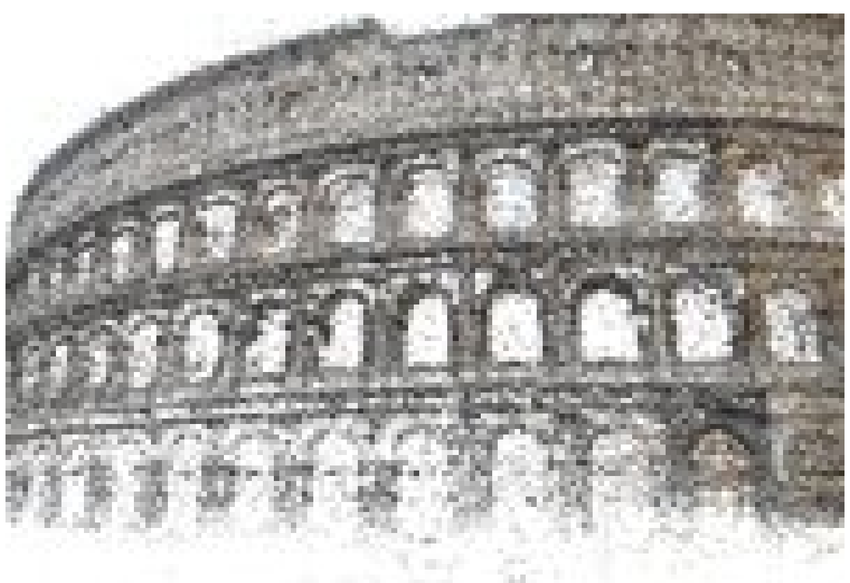
Photosynth

[Microsoft]



Iconic Scene Graphs

[Li et al. ECCV'08]

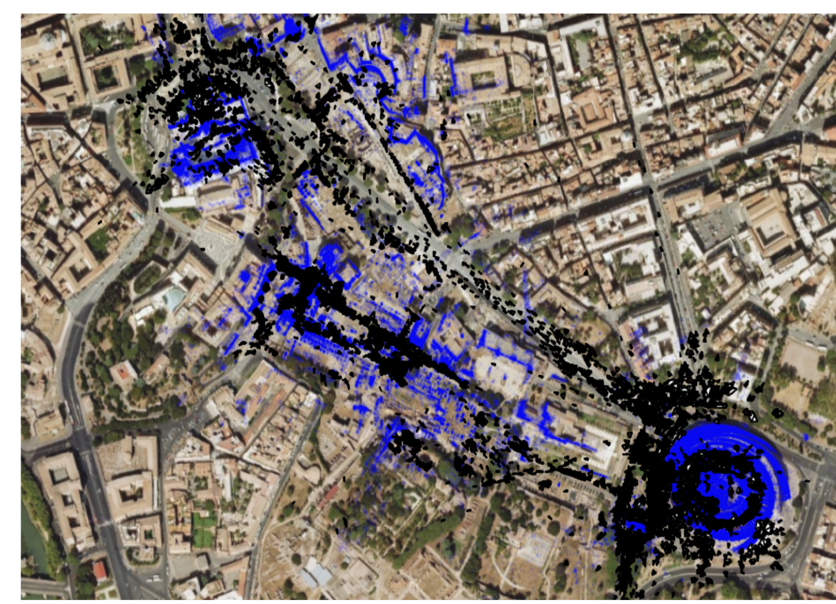


Build Rome in a Day

[Agarwal et al. ICCV'09]

Build Rome on a Cloudless Day

[Frahm et al. ECCV'10]



Discrete-Continuous Optim.

[Crandall et al. CVPR'11]

Figures by cited authors

3D Models from Community Databases

- E.g., Google image search on “Dubrovnik”

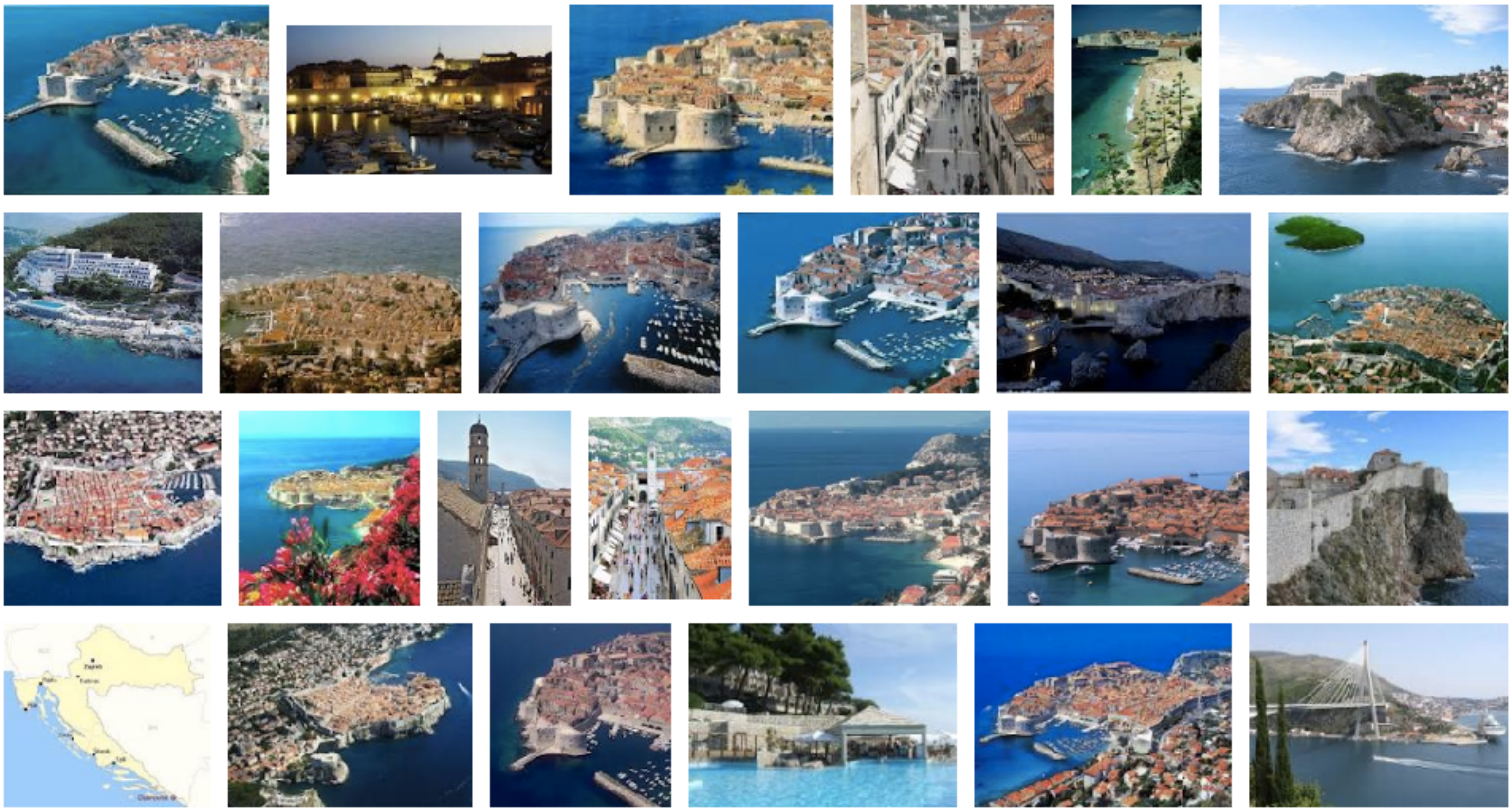


Figure by Aggarwal et al.

3D Models from Community Databases

Agarwal, Snavely, Seitz et al. at UW

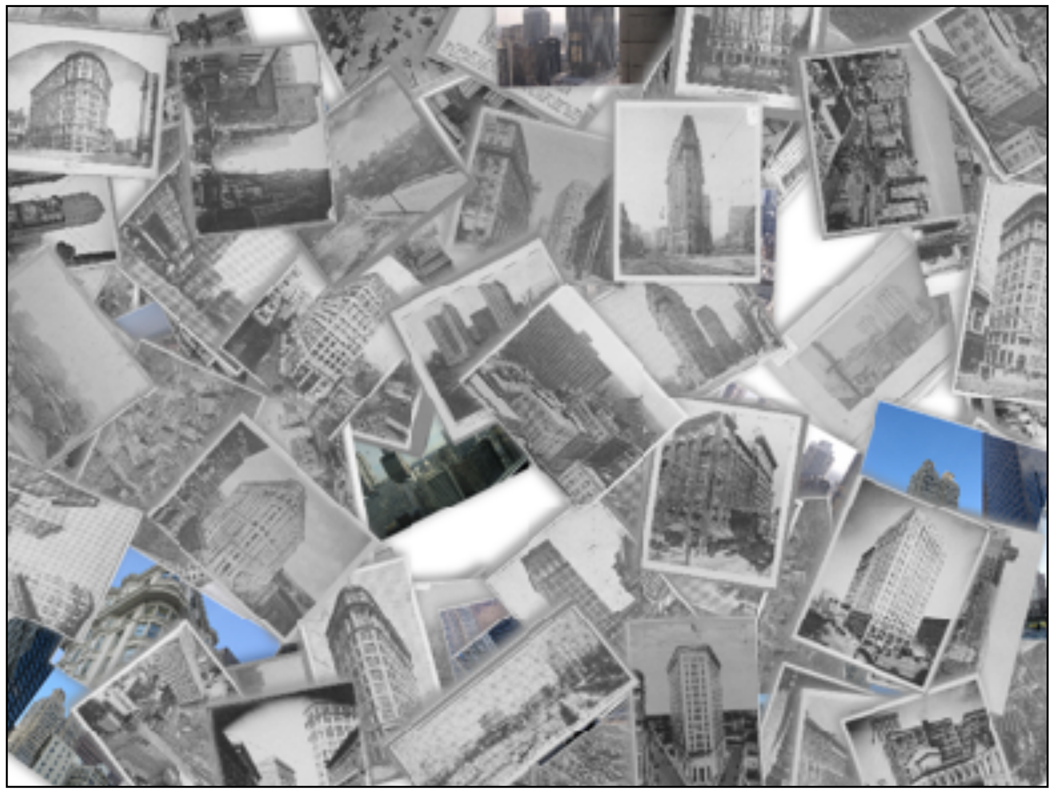
http://grail.cs.washington.edu/rome/

5K images, 3.5M points, >10M factors

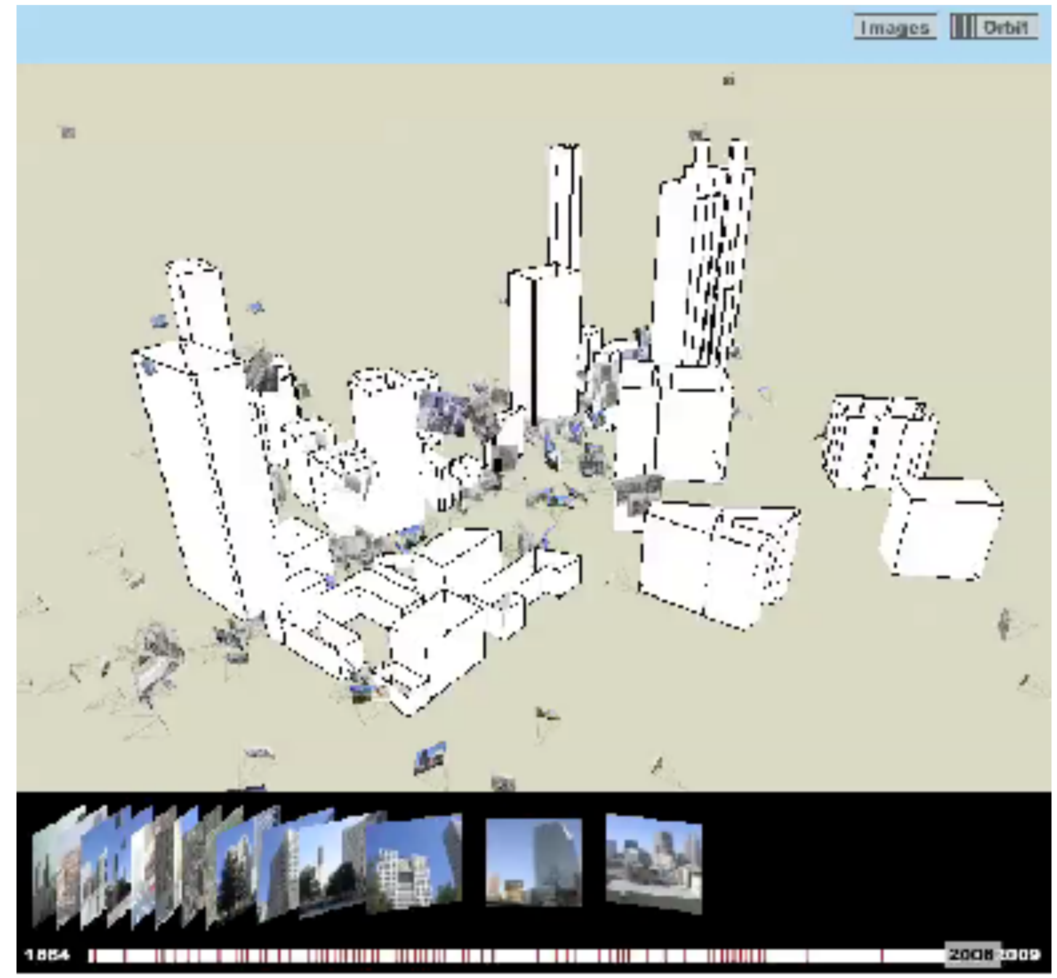
Movie by Aggarwal et al.

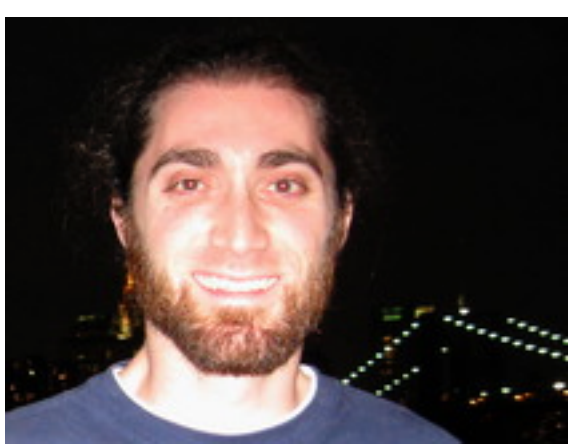
Spatiotemporal Reconstruction

4D Cities: 3D + Time



Historical Image Collection





Grant Schindler

Supported by NSF CAREER, Microsoft

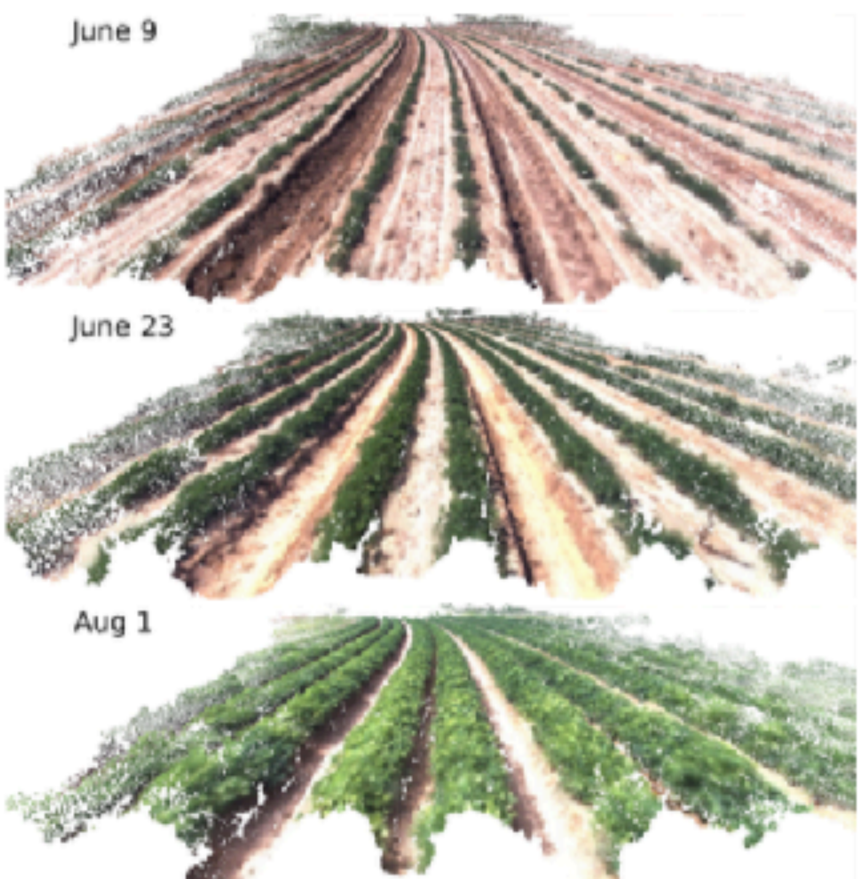
Recent revival: NSF NRI award on 4D crops for precision agriculture…

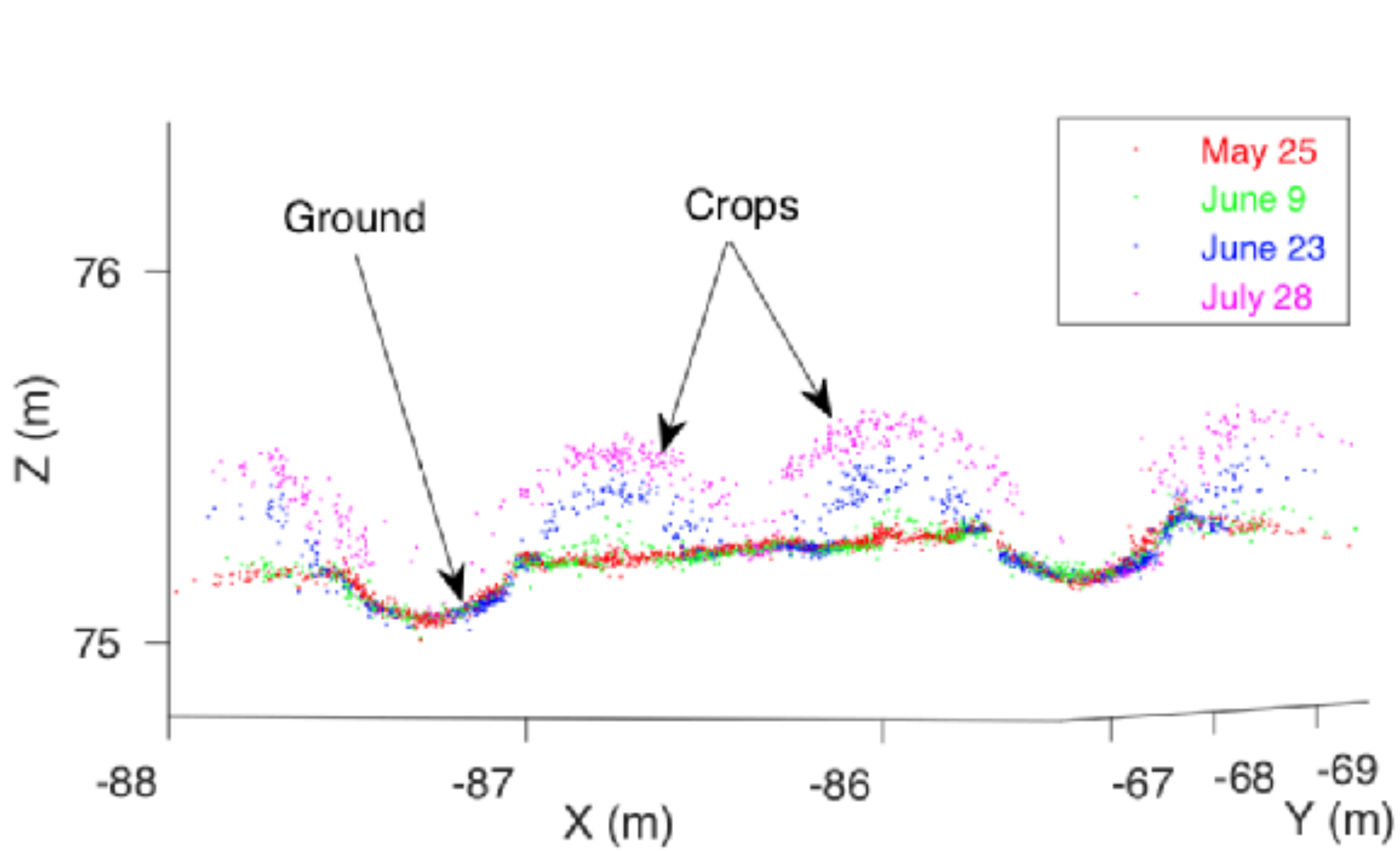
3D Reconstruction

Probabilistic Temporal Inference on Reconstructed 3D Scenes, G. Schindler and F. Dellaert, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

4D Structure over Time

Recently: 4D crop monitoring (NSF-NRI), w. Jong Ding





Outline

Introduction

Factor Graphs

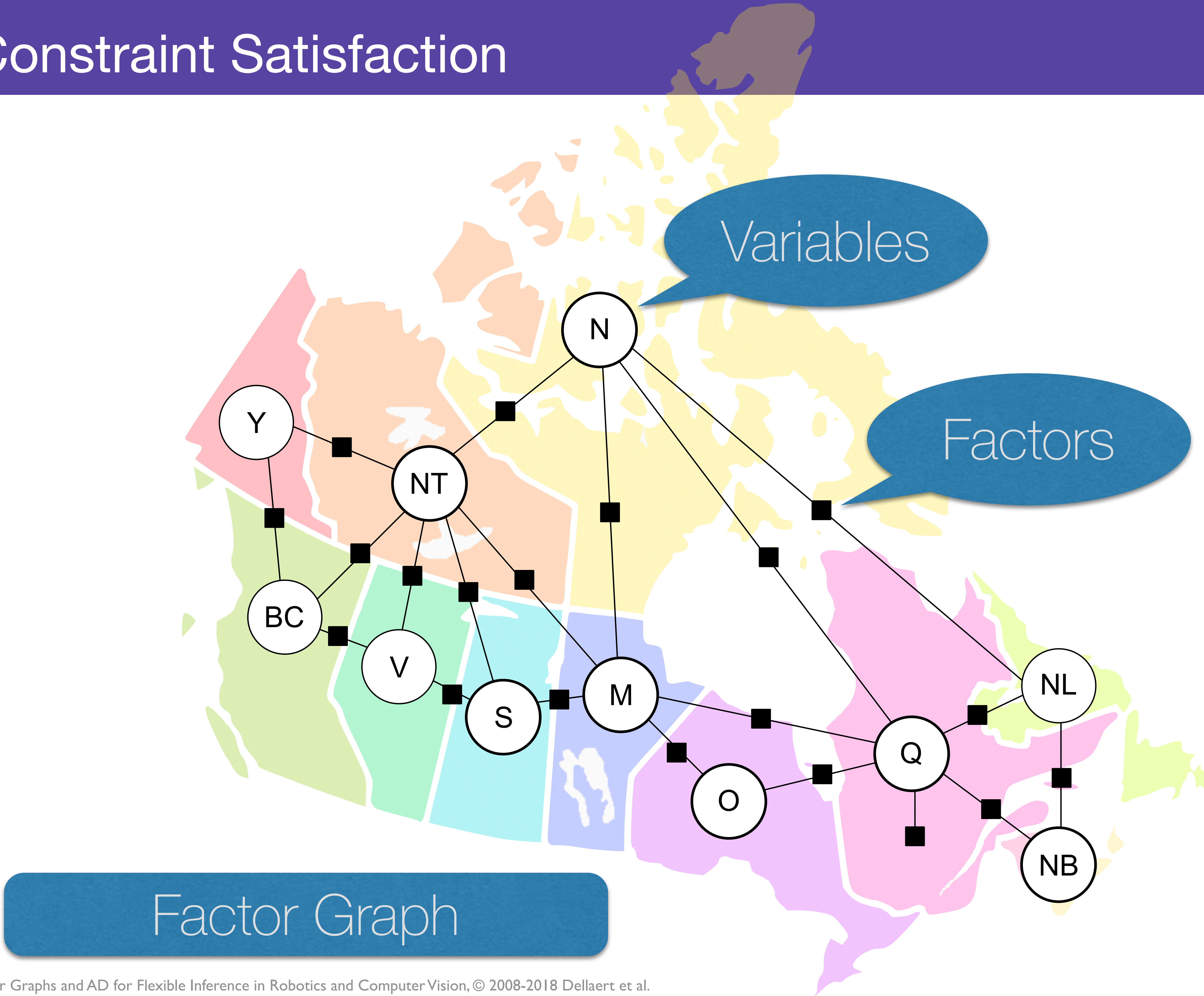
Manifold Optimization

Plug and Play AD

Conclusion

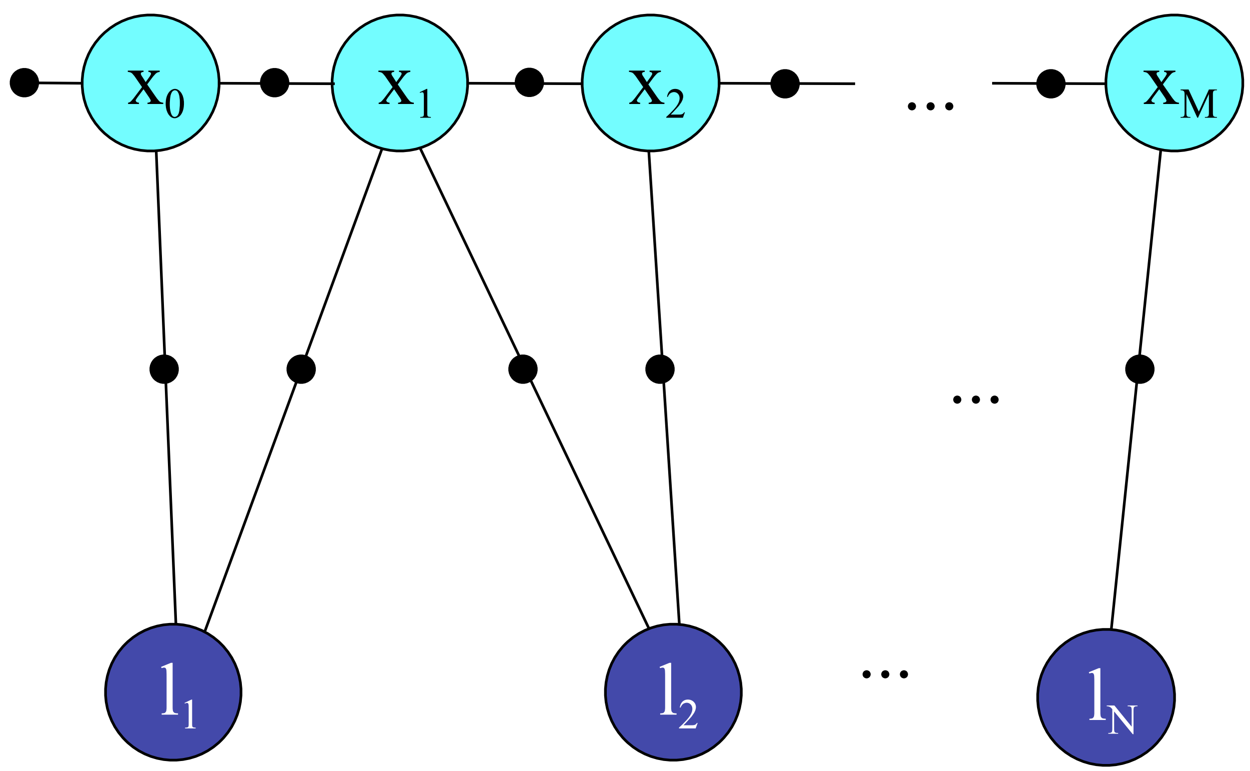
Constraint Satisfaction

Constraint Satisfaction



Simultaneous Localization and Mapping (SLAM)

Continuous Probability Densities



- Trajectory of Robot

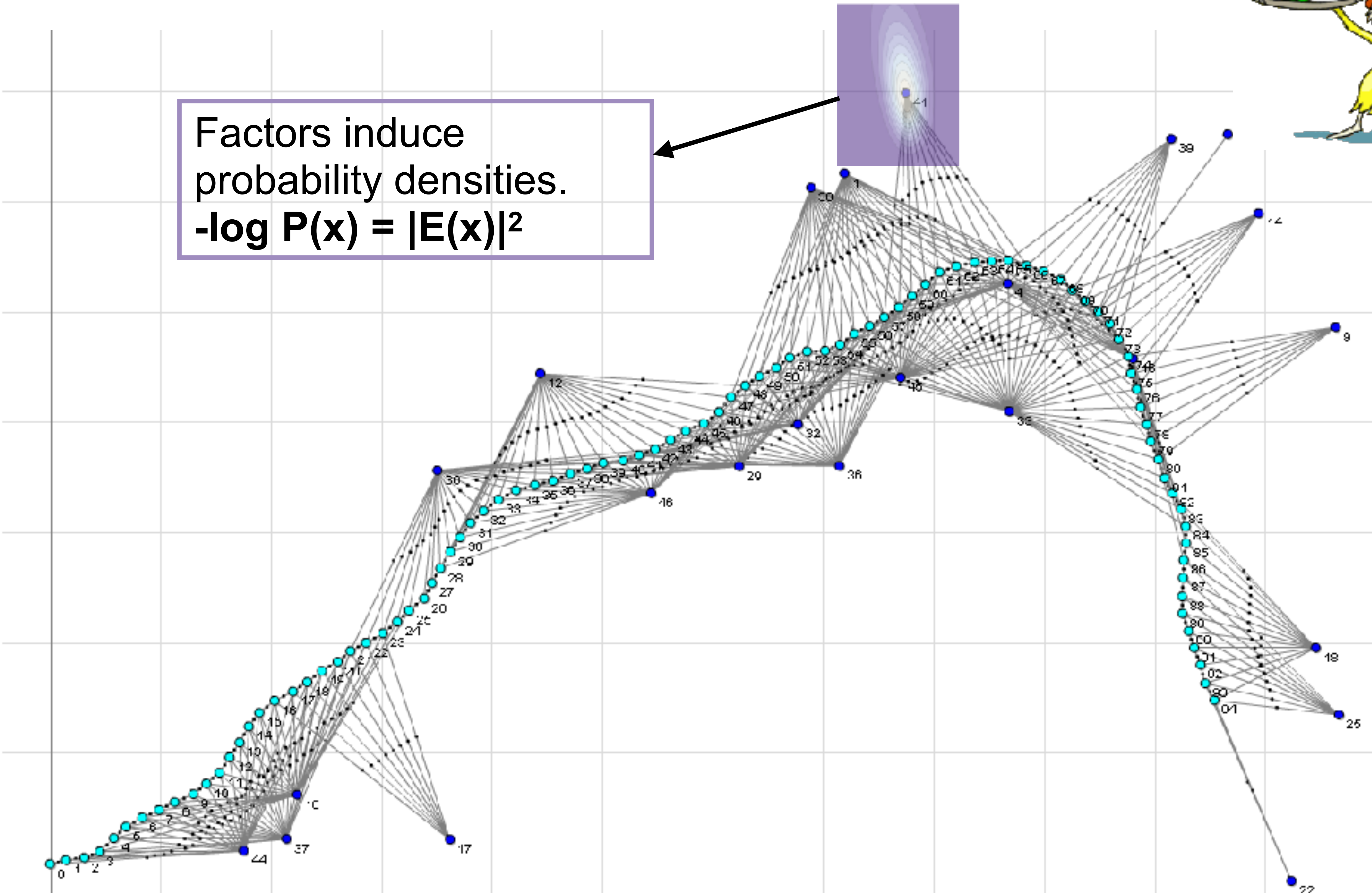
- Measurements

- Landmarks

$$P(X,M) = k^* P(x_0) \prod_{i=1}^{M} P(x_i | x_{i-1}, u_i) \times \prod_{k=1}^{K} P(z_k | x_{i_k}, l_{j_k})$$

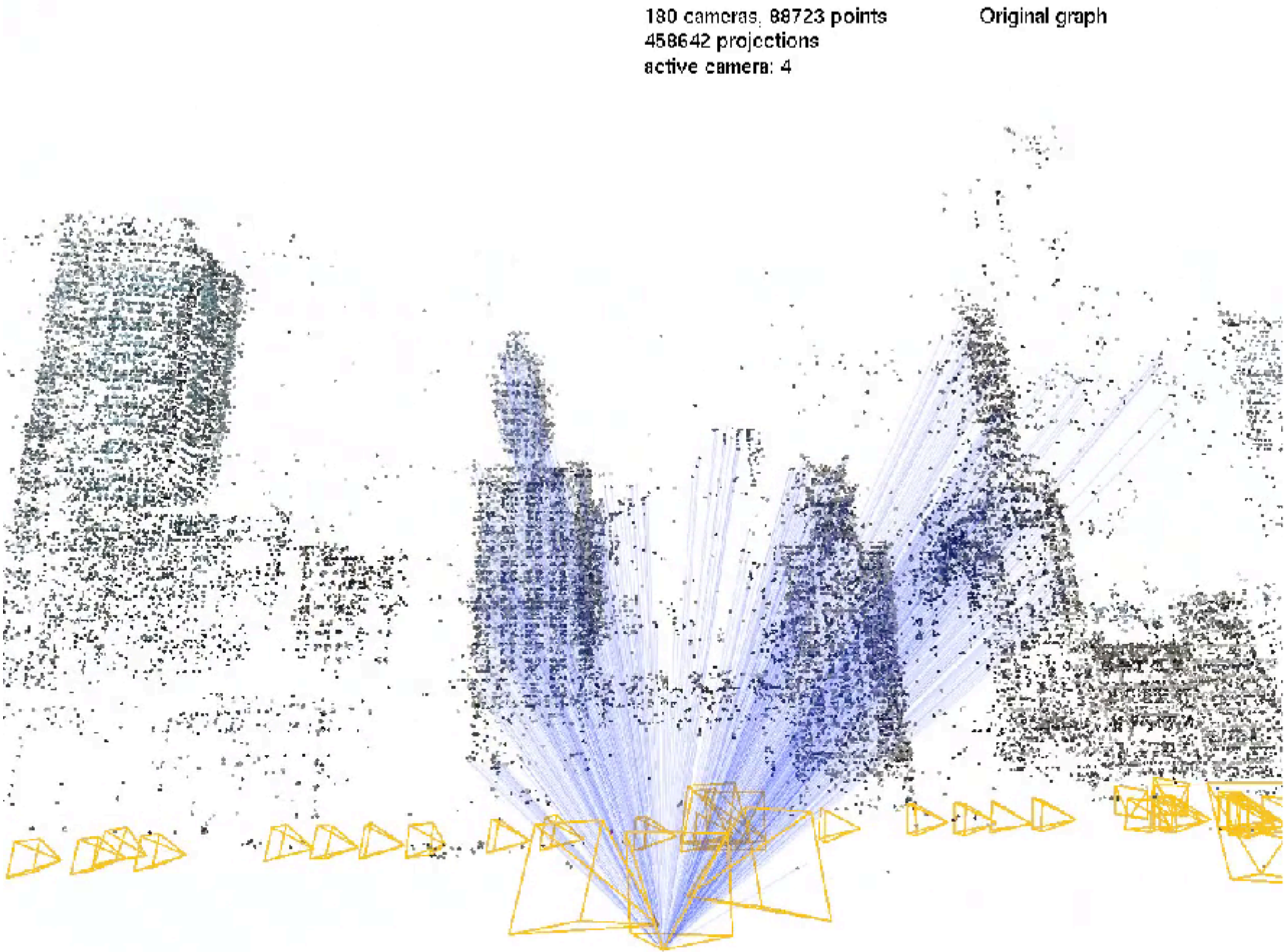
Factor Graph -> Smoothing and Mapping (SAM) !

Variables are multivariate! Factors are non-linear!



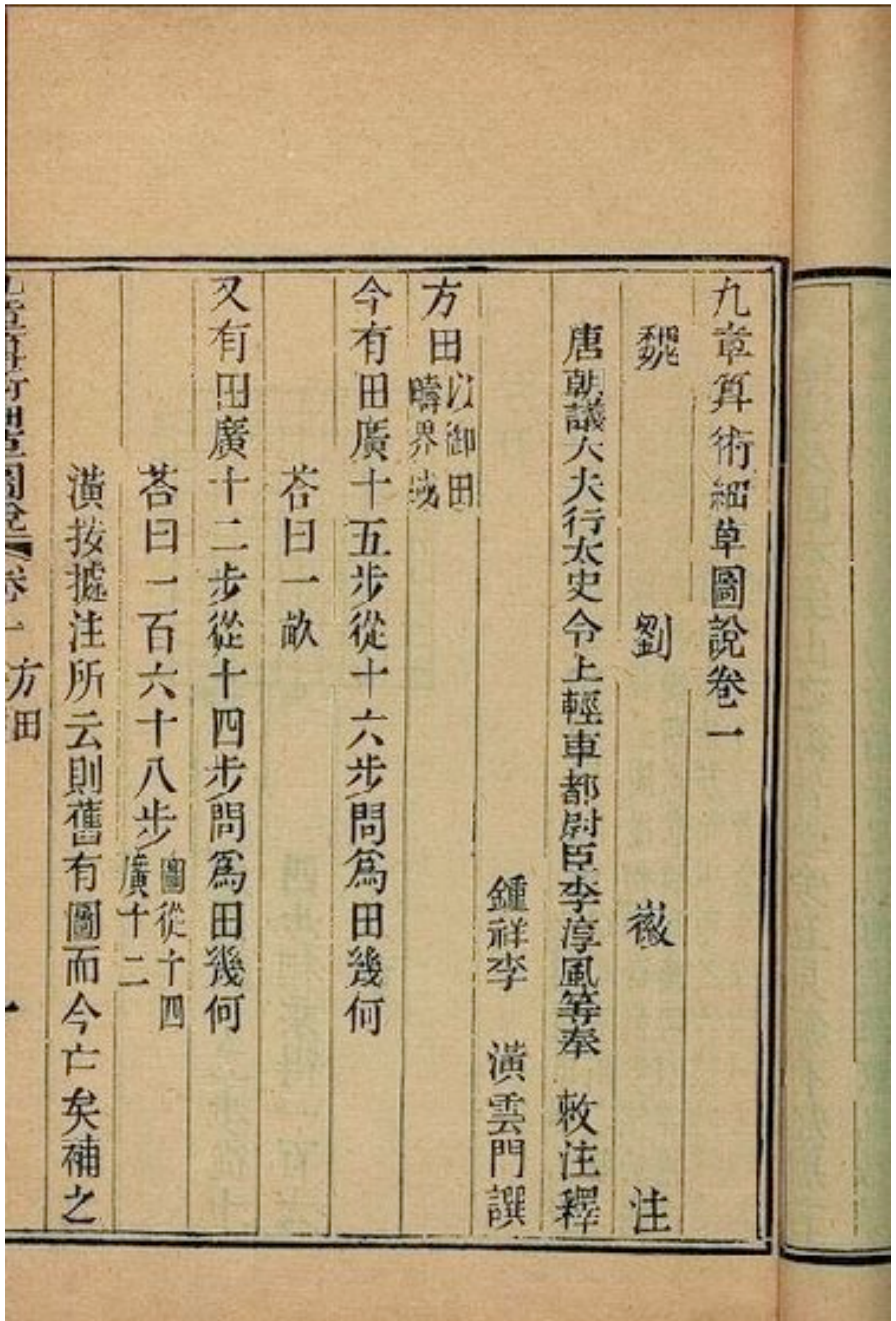
Factors induce
probability densities.
-log P(x) = |E(x)|²

Structure from Motion (Chicago, movie by Yong Dian Jian)

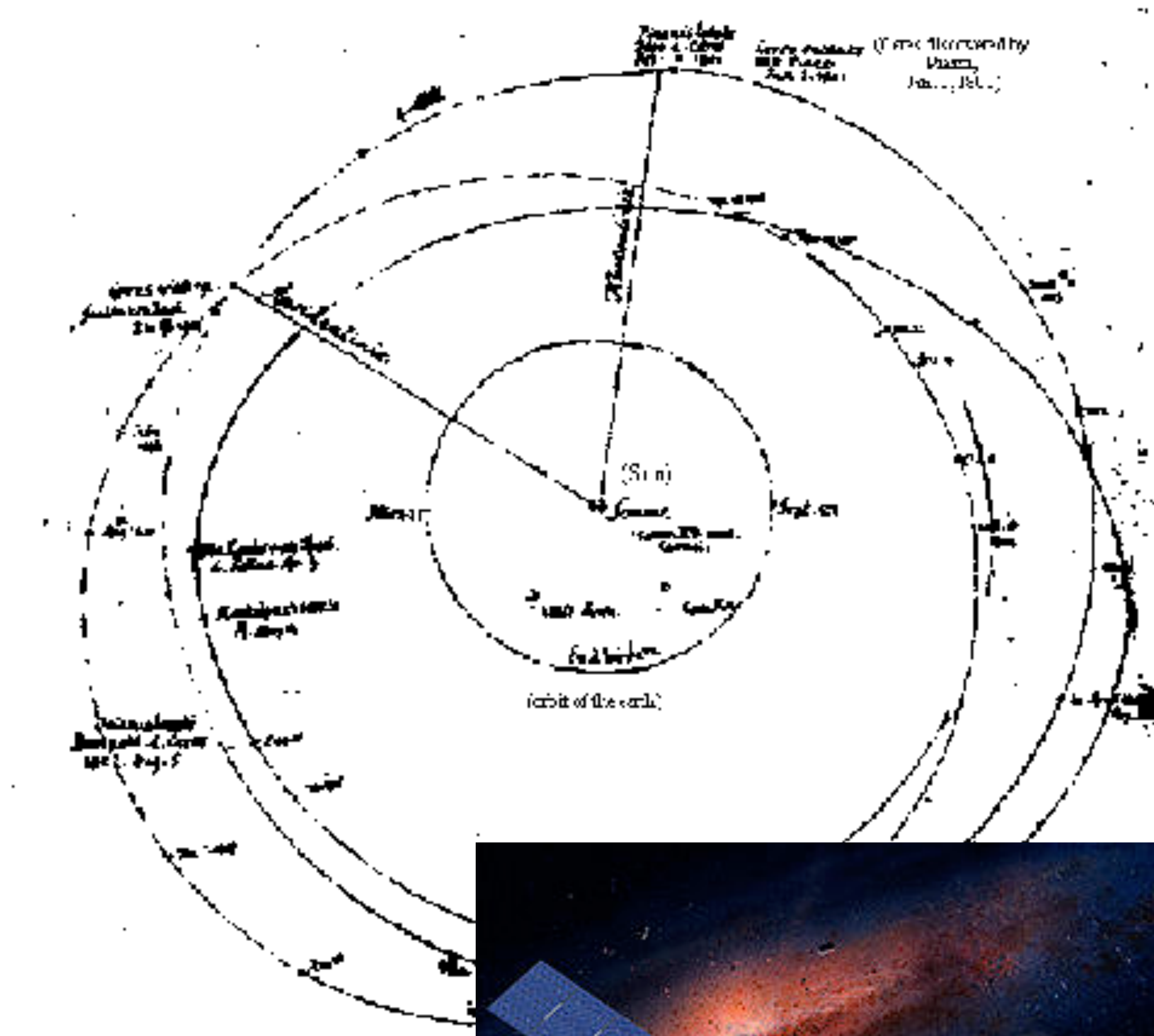


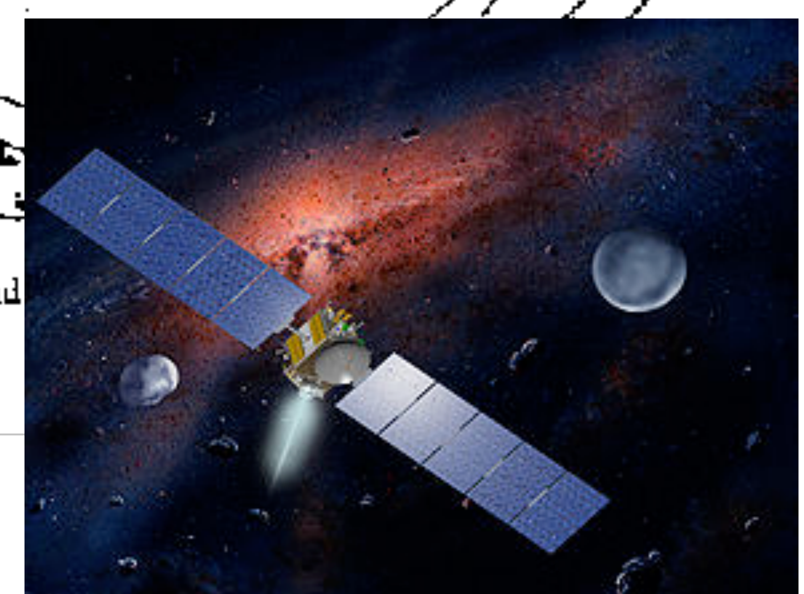
Direct LS optimization == “Gaussian” elimination.

The Nine Chapters on the Mathematical Art, 200 B.C.


九章算術細草圖說卷一
魏 劉 徽 注
唐朝議大夫行太史令上輕車都尉臣李淳風等奉 敕注釋
鍾祥李 潢 雲門 譔
方田 以御田疇界域
今有田廣十五步從十六步問為田幾何
荅曰一畝
又有田廣十二步從十四步問為田幾何
荅曰一百六十八步 圖從十四廣十二
潢按據注所云則舊有圖而今亡矣補之
九章算術細草圖說 卷一 方田

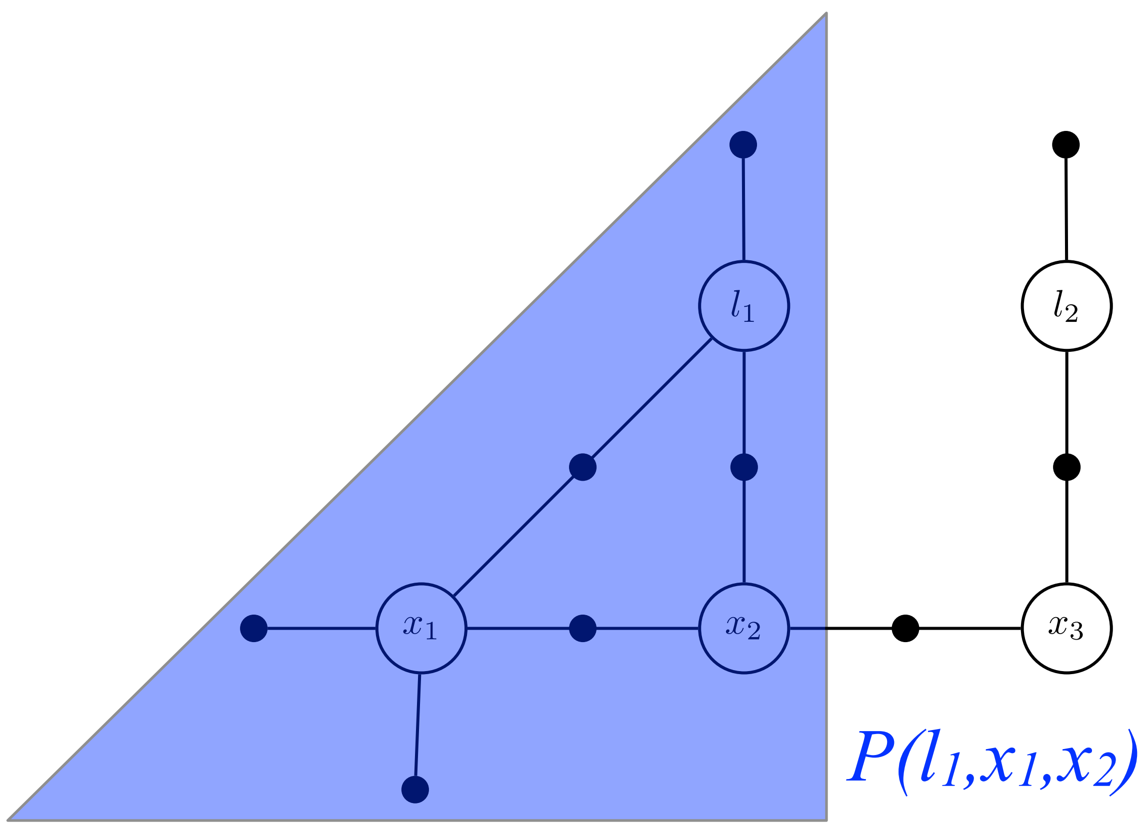
Gauss re-invented elimination for determining the orbit of Ceres





Variable Elimination

- Choose Ordering
- Eliminate one node at a time



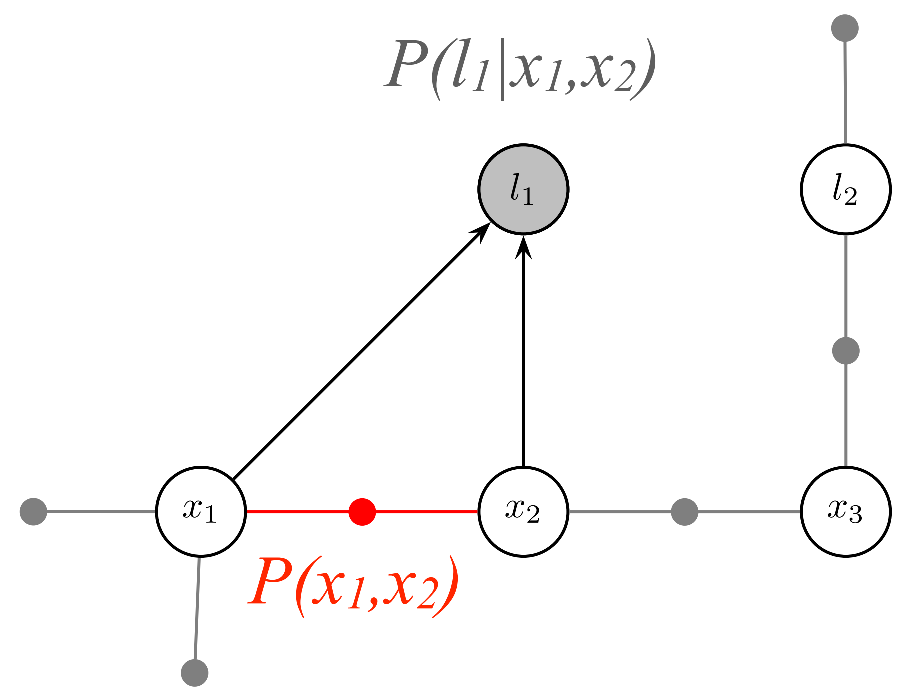
l_1	l_2	x_1	x_2	x_3
x				
x		x		
x			x	
	x			
	x			x
		x		
		x		
		x	x	
			x	x

- Express node as function of adjacent nodes

Basis = Chain Rule ! e.g. $P(l_1,x_1,x_2)=P(l_1|x_1,x_2)P(x_1,x_2)$

Small Example

- Choose Ordering
- Eliminate one node at a time



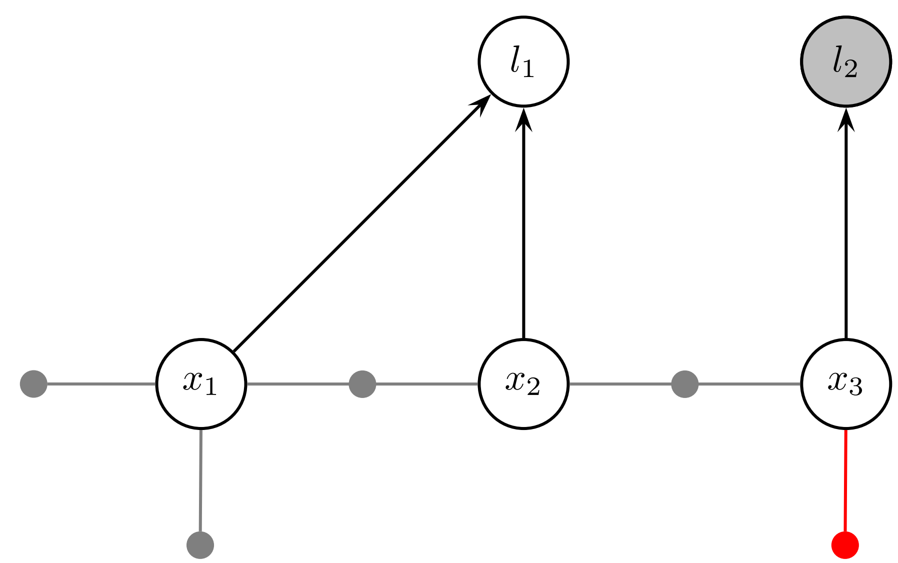
l_1	l_2	x_1	x_2	x_3
x		x	x	
	x			
	x			x
		x	x	
		x		
		x		
		x	x	
			x	x

- Express node as function of adjacent nodes

Basis = Chain Rule ! e.g. $P(l_1,x_1,x_2)=P(l_1|x_1,x_2)P(x_1,x_2)$

Small Example

- Choose Ordering
- Eliminate one node at a time

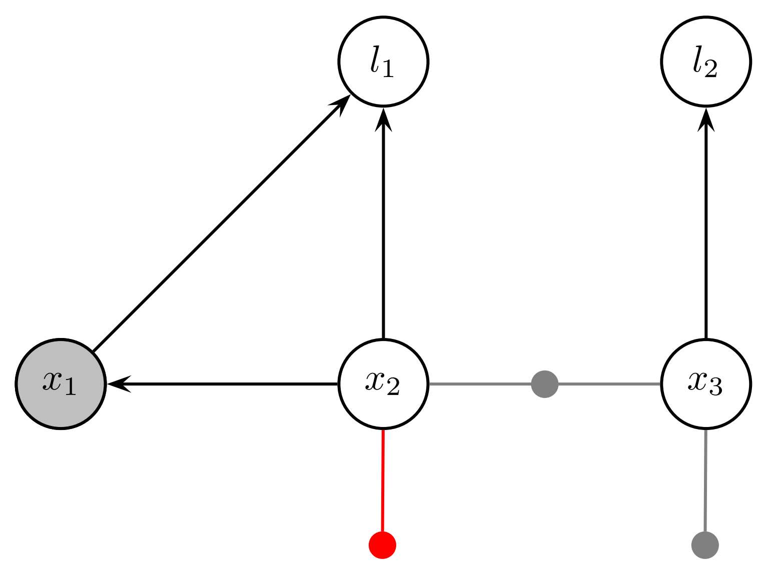


l_1	l_2	x_1	x_2	x_3
x		x	x	
	x			x
		x	x	
		x		
		x		
		x	x	
			x	x
				x

- Express node as function of adjacent nodes

Small Example

- Choose Ordering
- Eliminate one node at a time

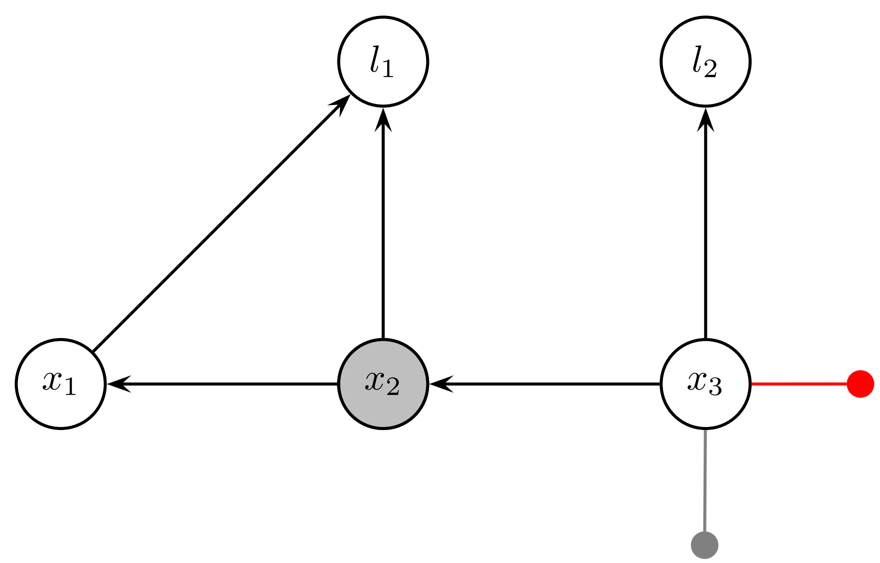


l_1	l_2	x_1	x_2	x_3
x		x	x	
	x			x
		x	x	
			x	
			x	x
				x

- Express node as function of adjacent nodes

Small Example

- Choose Ordering
- Eliminate one node at a time



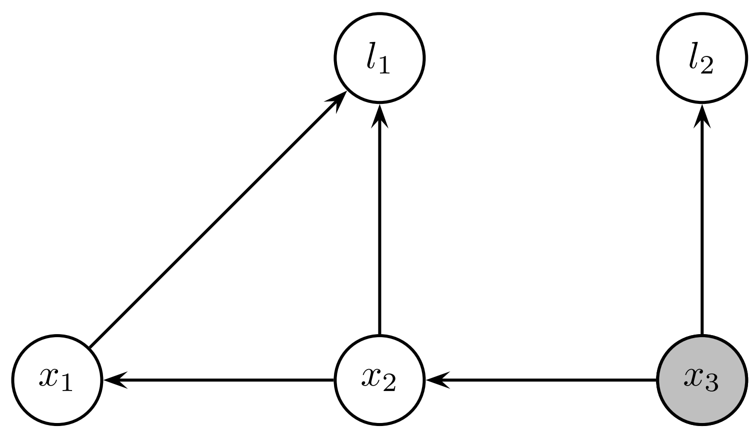
l_1	l_2	x_1	x_2	x_3
x		x	x	
	x			x
		x	x	
			x	x
				x
				x

- Express node as function of adjacent nodes

Small Example

- Choose Ordering
- Eliminate one node at a time

l_1	l_2	x_1	x_2	x_3
x		x	x	
	x			x
		x	x	
			x	x
				x

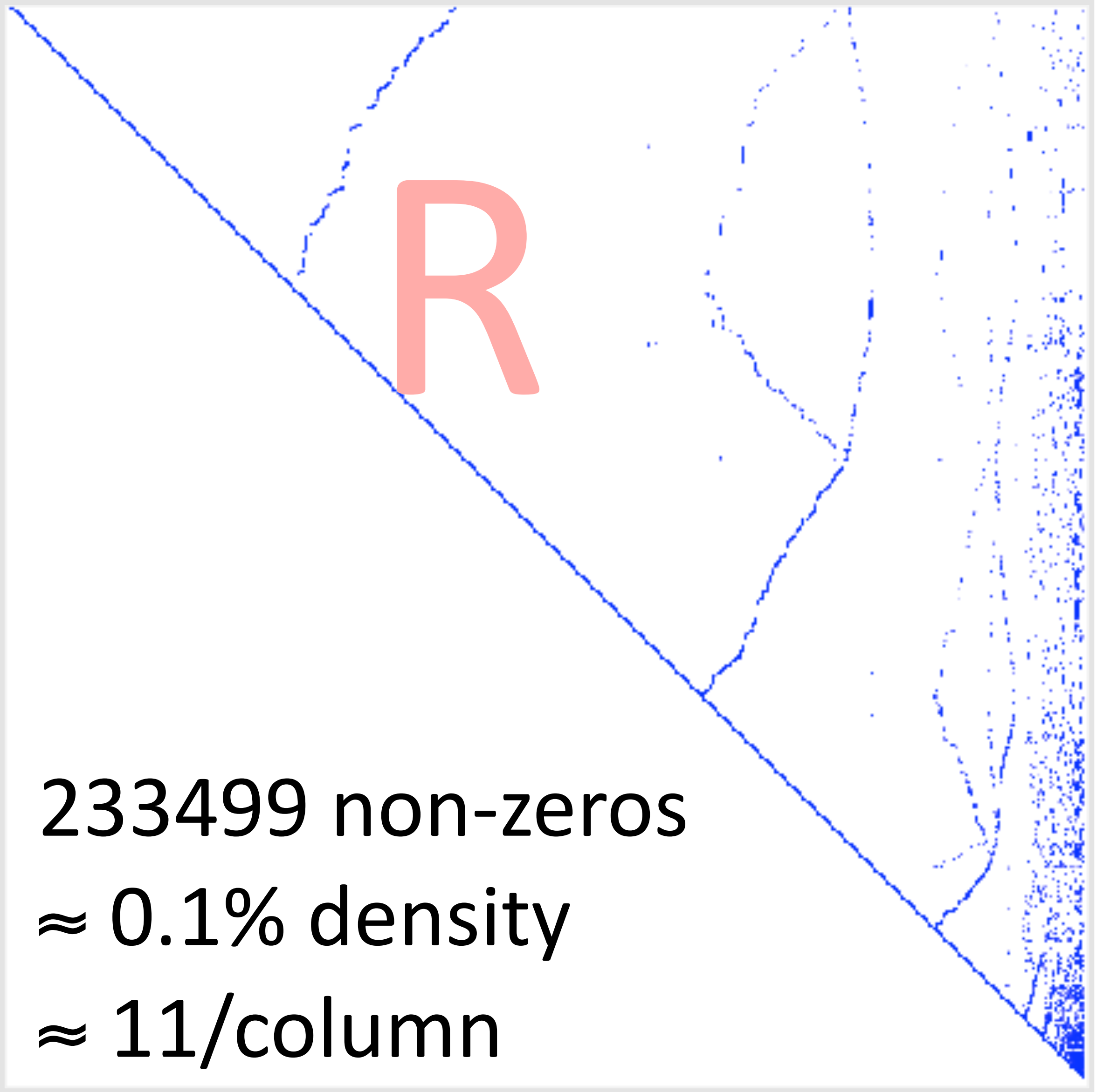


- Express node as function of adjacent nodes

Example

Example from real sequence:
Square root inf. matrix

Side length: 21000 variables
Dense: 1.7GB, sparse: 1MB



Victoria Park in Sydney, Australia

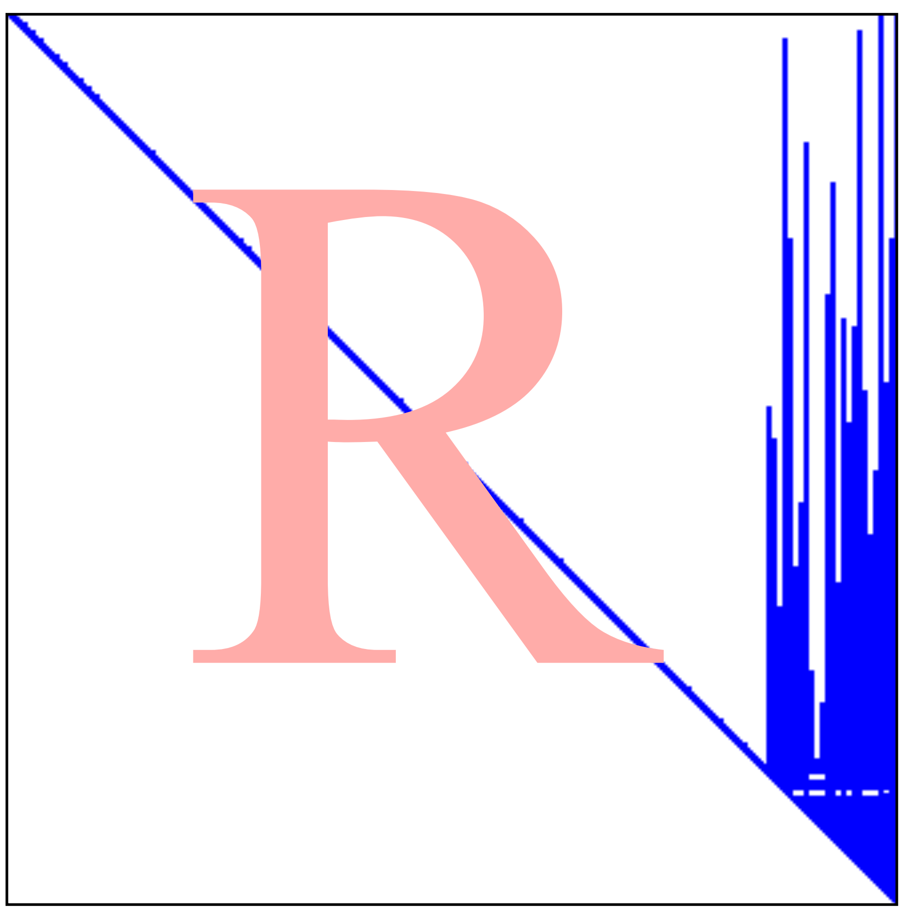
4 km trajectory

6968 frames, 140 landmarks

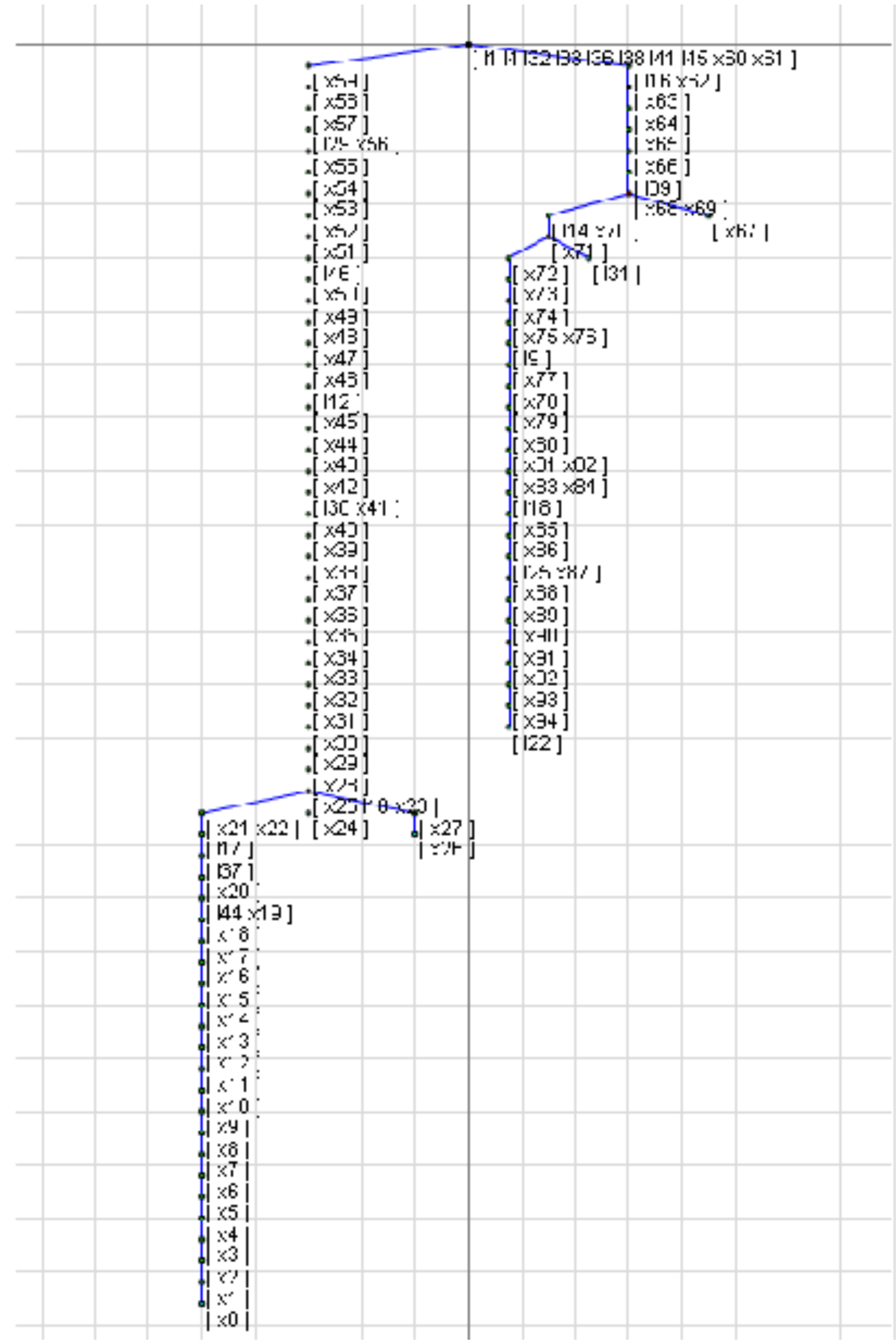
Incremental Smoothing and Mapping (iSAM)

Michael Kaess

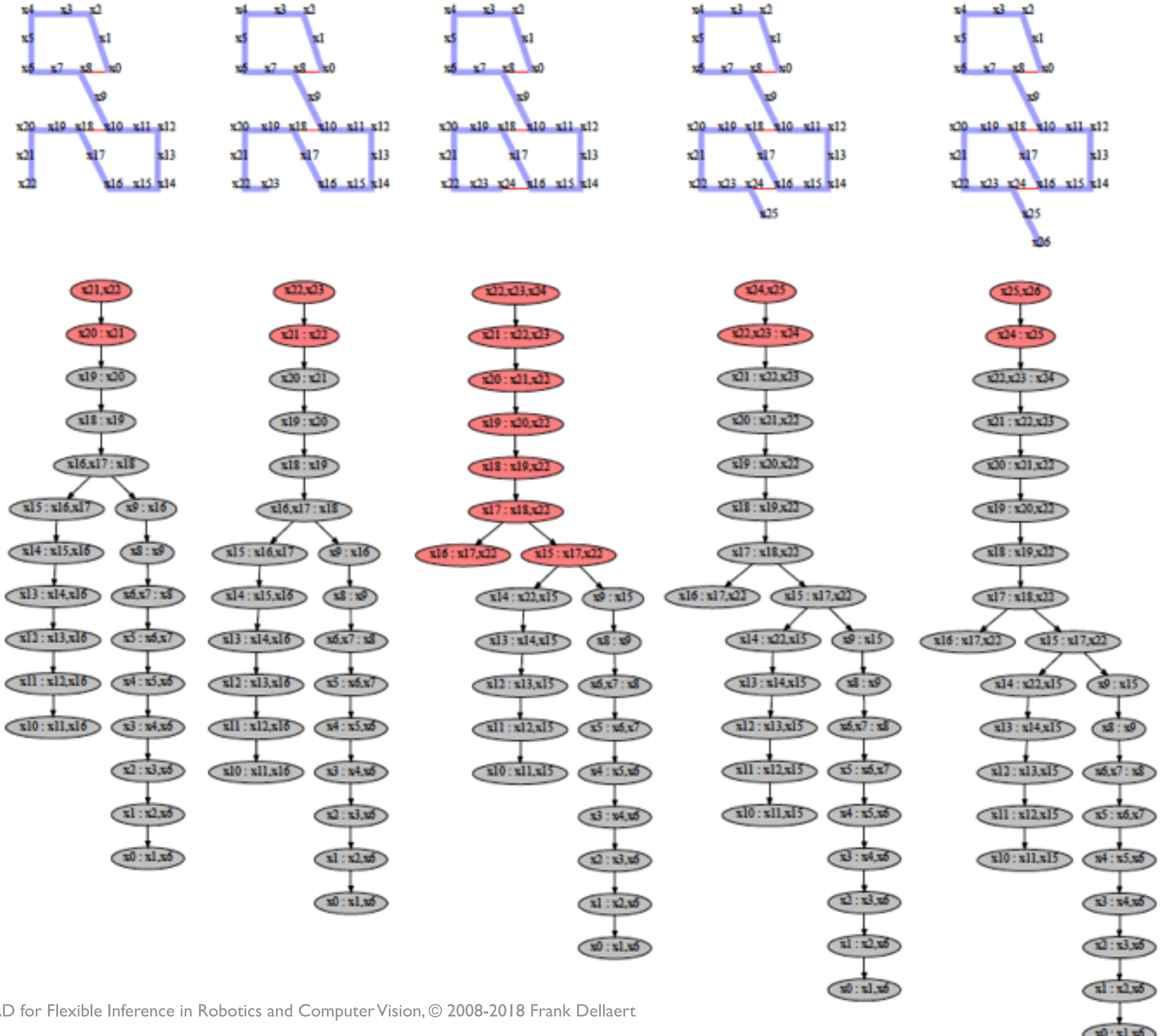
Exploit the fact that sparse Cholesky factor can be understood as a tree-structure: the Bayes tree



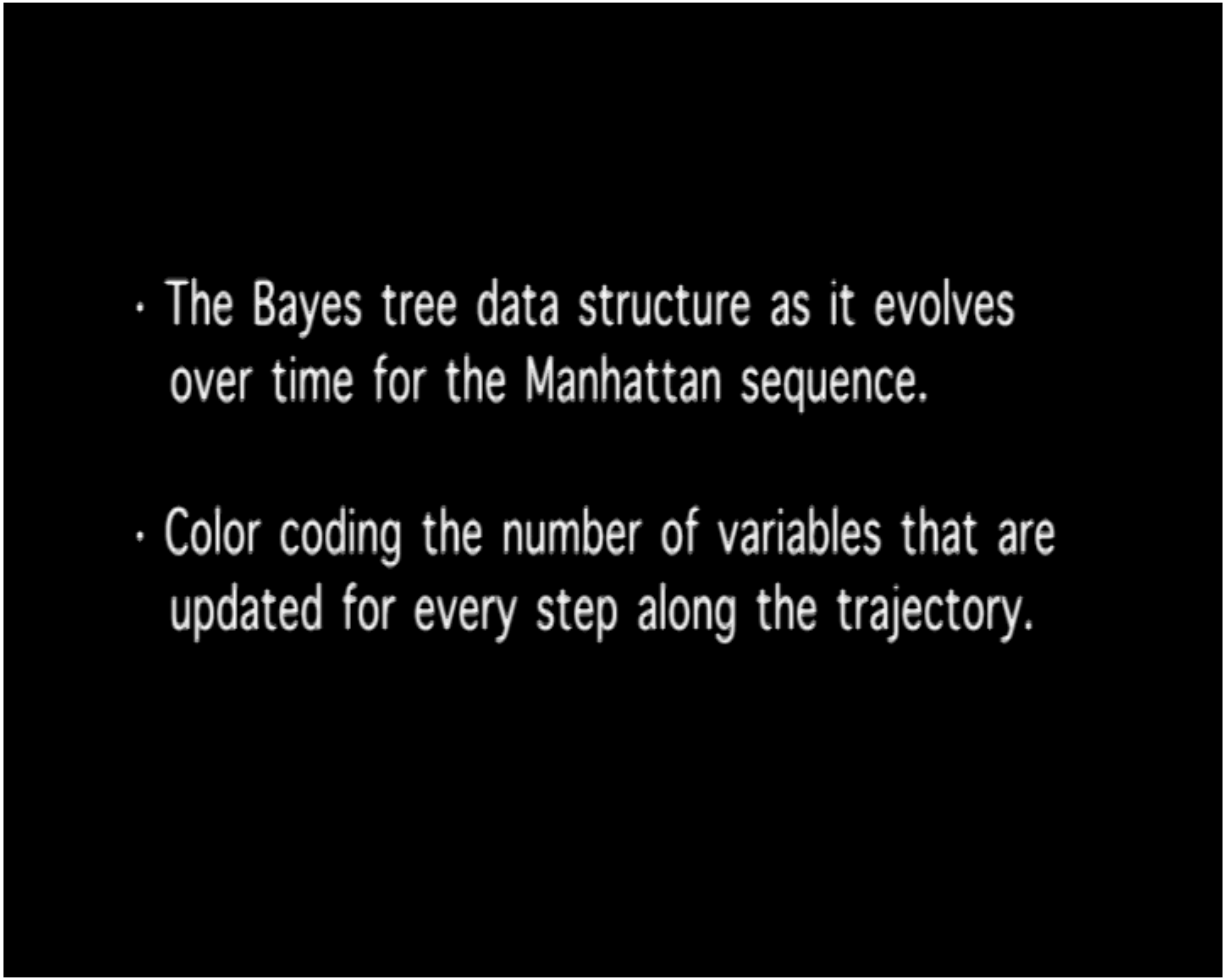
==



iSAM: Incremental inference on Bayes Tree [IJRR 12]

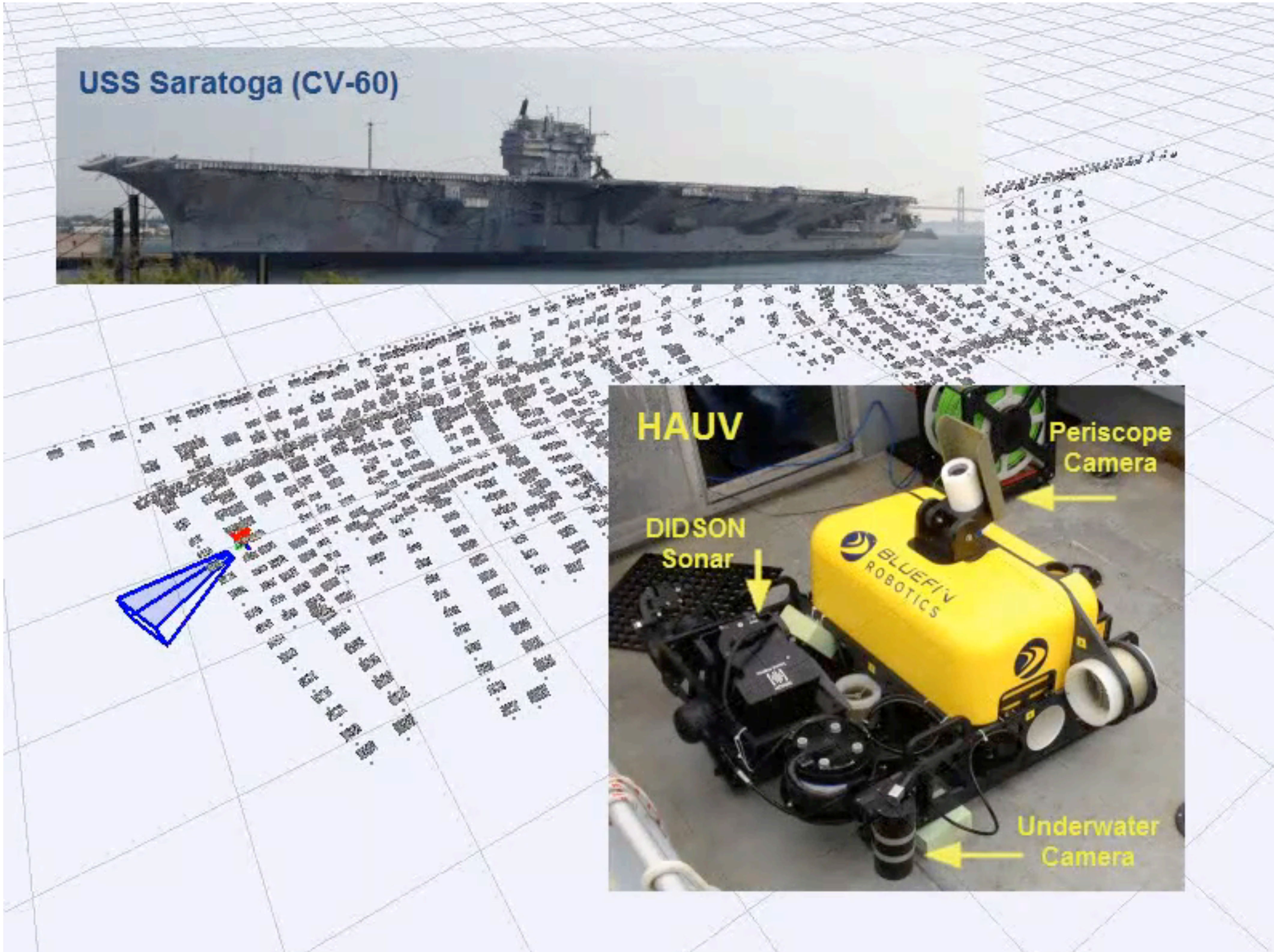


iSAM2 [IJRR 12] Example on synthetic sequence



Georgia Tech & MIT, Kaess et al., IJRR 12

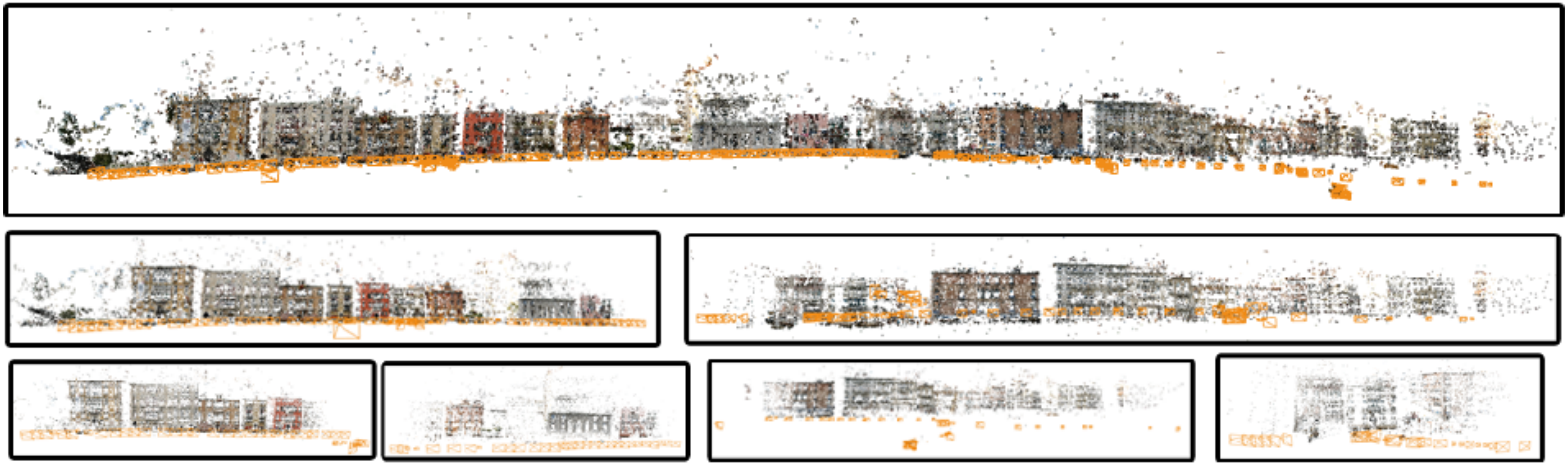
Mapping Aircraft Carriers etc....



August 2013, Work by Ryan Eustice et al., University of Michigan & MIT

Movie by Eustice et al.

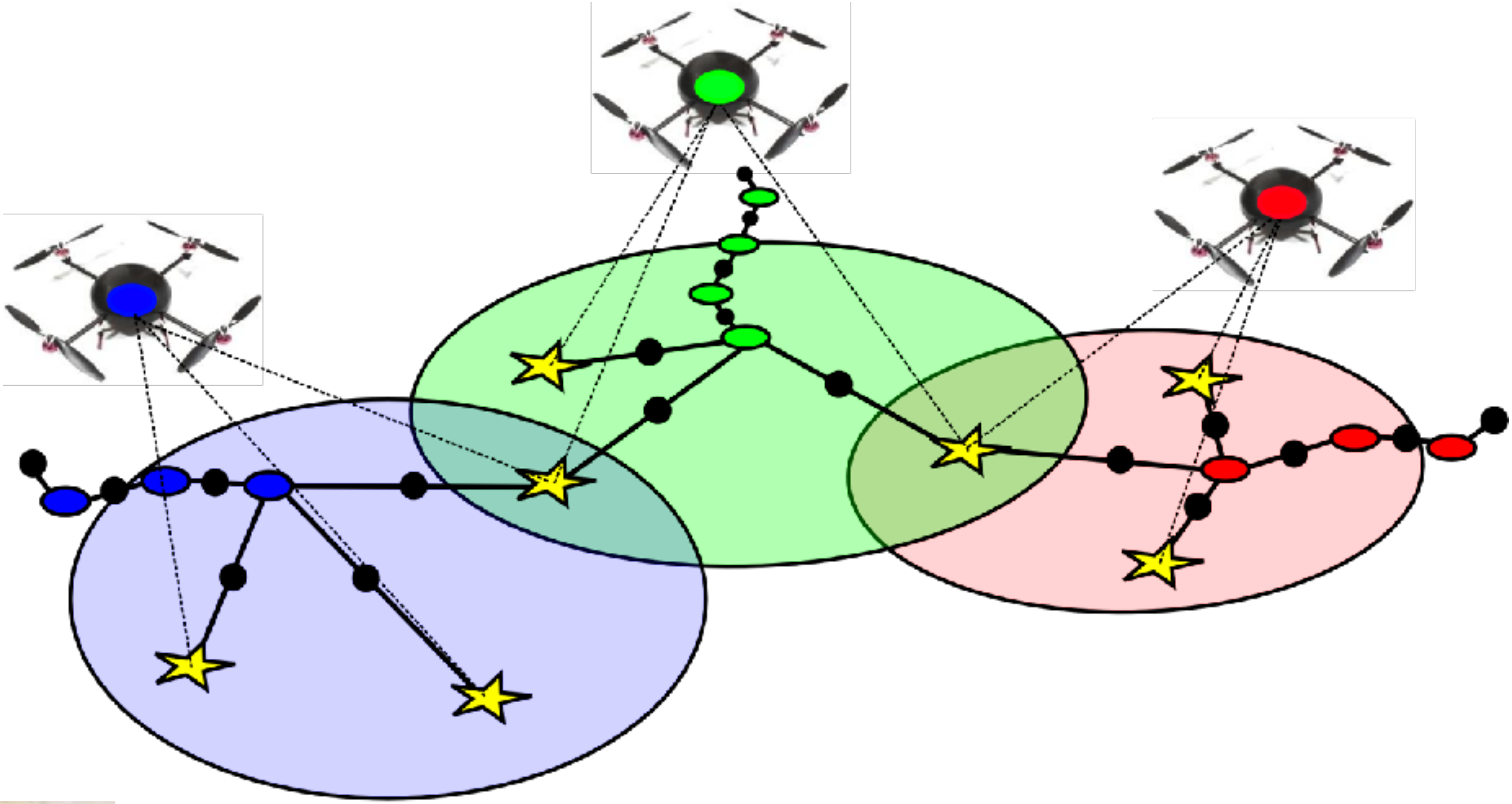
Nested-Dissection for SFM: Hyper-SFM



Kai now leads an autonomous driving startup in China

Kai Ni, and Frank Dellaert, **HyperSfM**, *IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012*.

Distributed SLAM: Alex Cunningham



Alexander Cunningham, Vadim Indelman, and Frank Dellaert, ***DDF-SAM 2.0: Consistent Distributed Smoothing and Mapping***, *IEEE International Conference on Robotics and Automation (ICRA), 2013*

Demo: collaboration with Nathan Michael at CMU

At Georgia Tech: Jing Dong and Vadim Indelman (now faculty at Technion)

Distributed Real-time Cooperative Localization and Mapping using an Uncertainty-Aware Expectation Maximization Approach

Jing Dong, Erik Nelson, Vadim Indelman,
Nathan Michael, Frank Dellaert

Georgia Tech | Institute for Robotics and Intelligent Machines

Carnegie Mellon University

TECHNION Israel Institute of Technology

Outline

Introduction

Factor Graphs

$\boxed{\text{Manifold Optimization}}$

Plug and Play AD

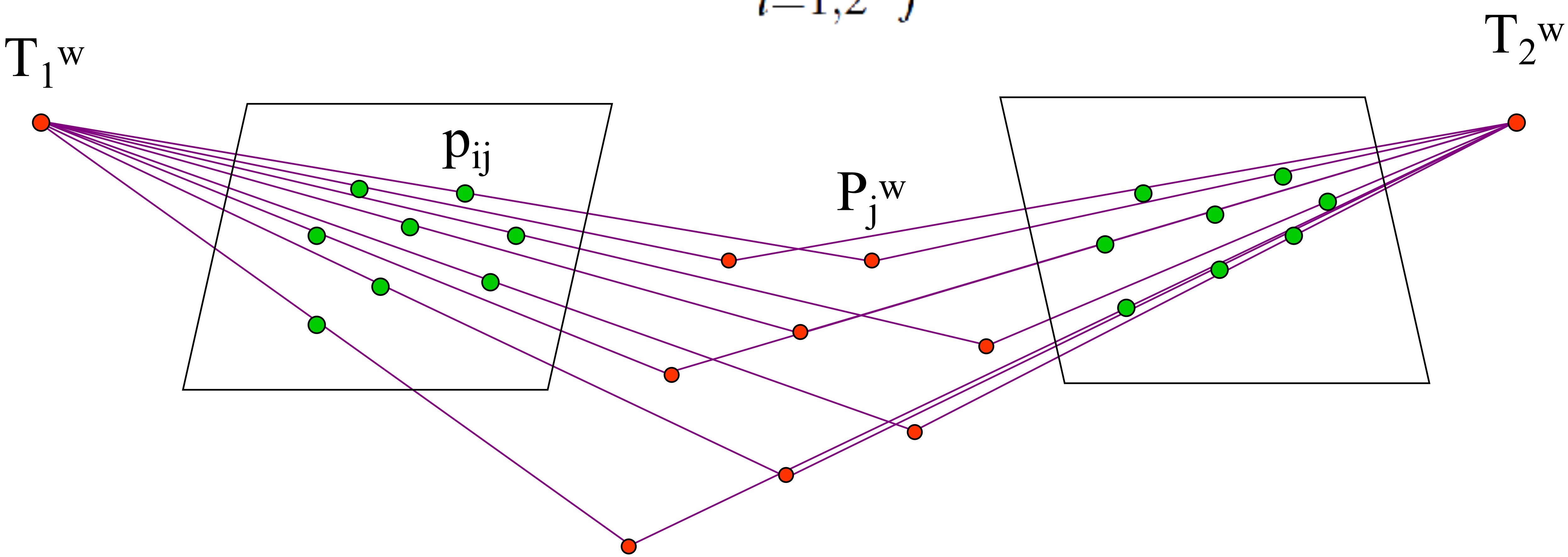
Conclusion

Skydio

Objective Function for SFM

- A sum of least-squares factors

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



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?

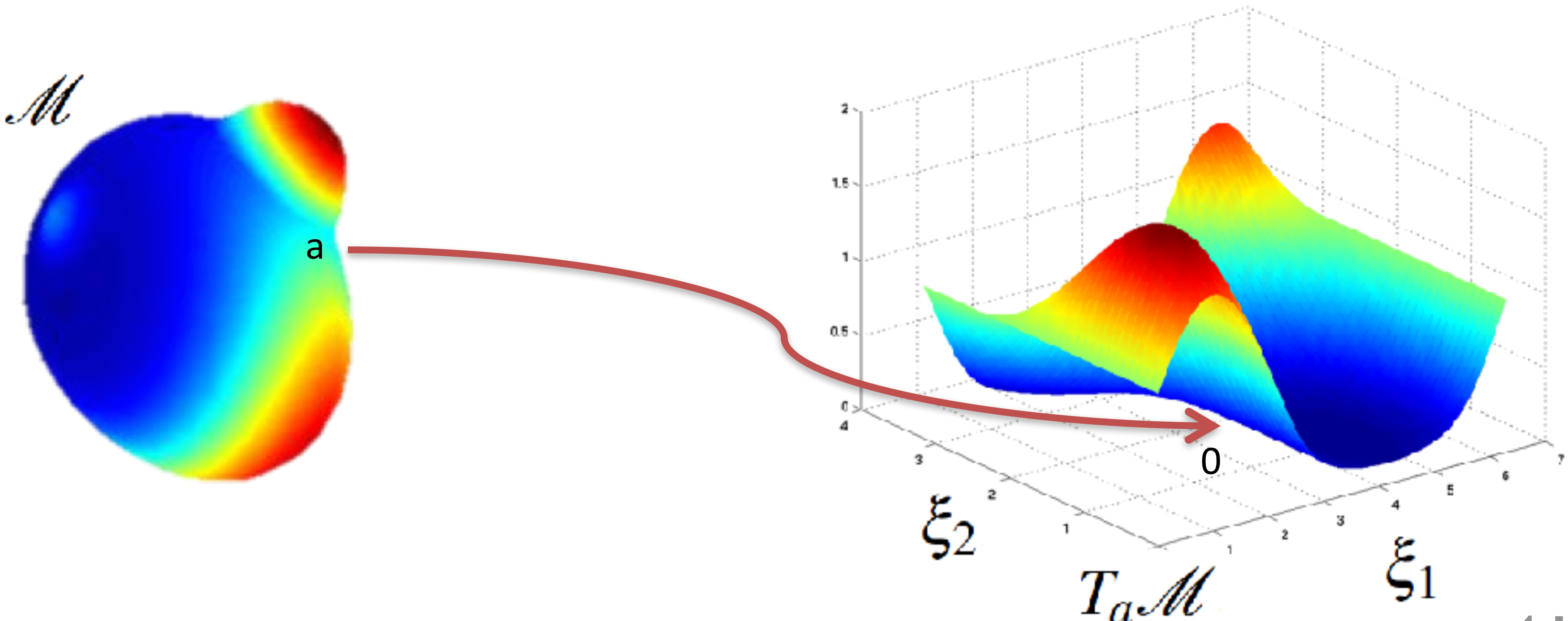
$$h(T_c^w + \xi, P^w + \delta) \approx h(T_c^w, P^w) + F\xi + G\delta.$$

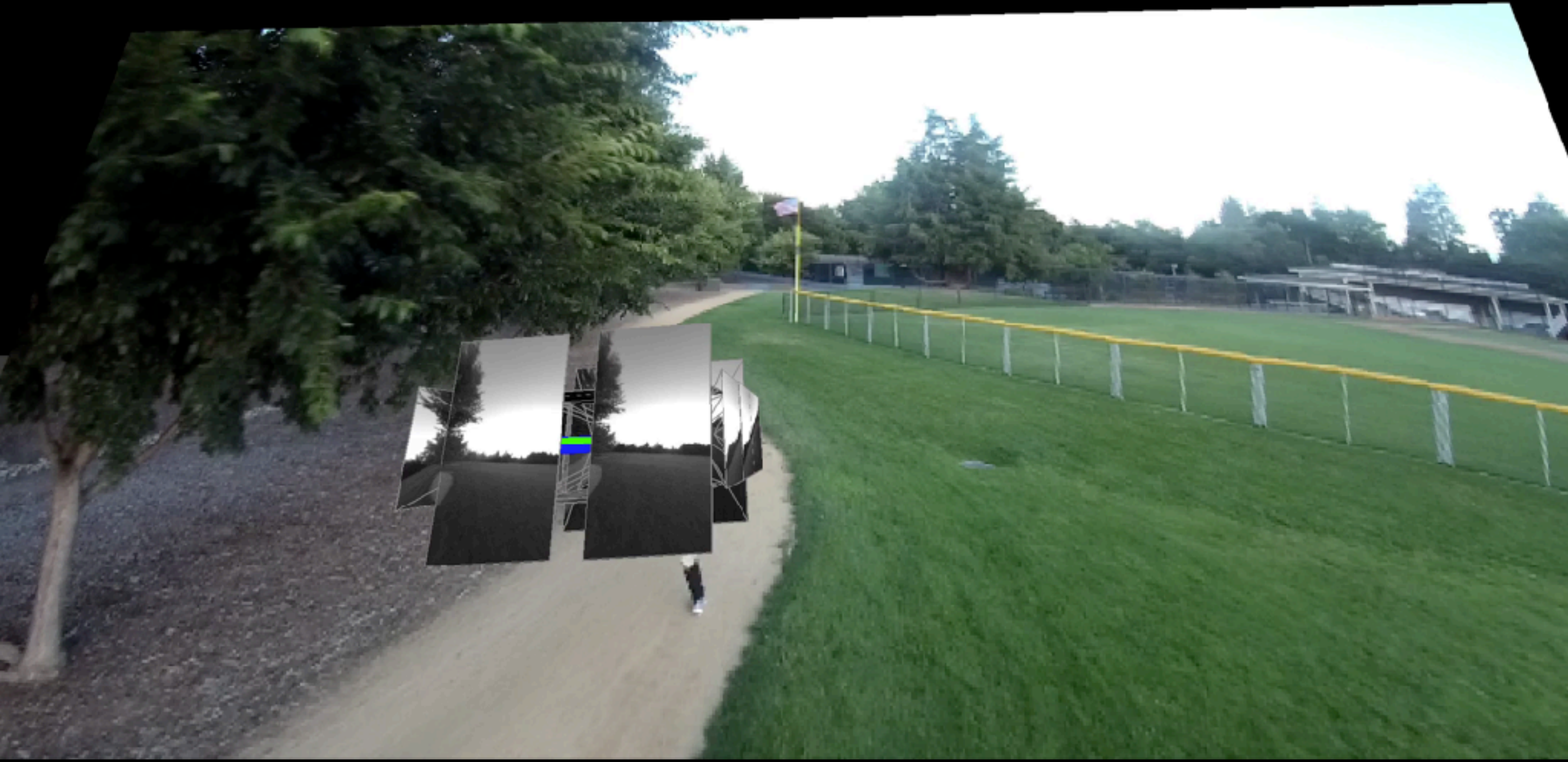
- Oops: $T_c^w + \xi$?

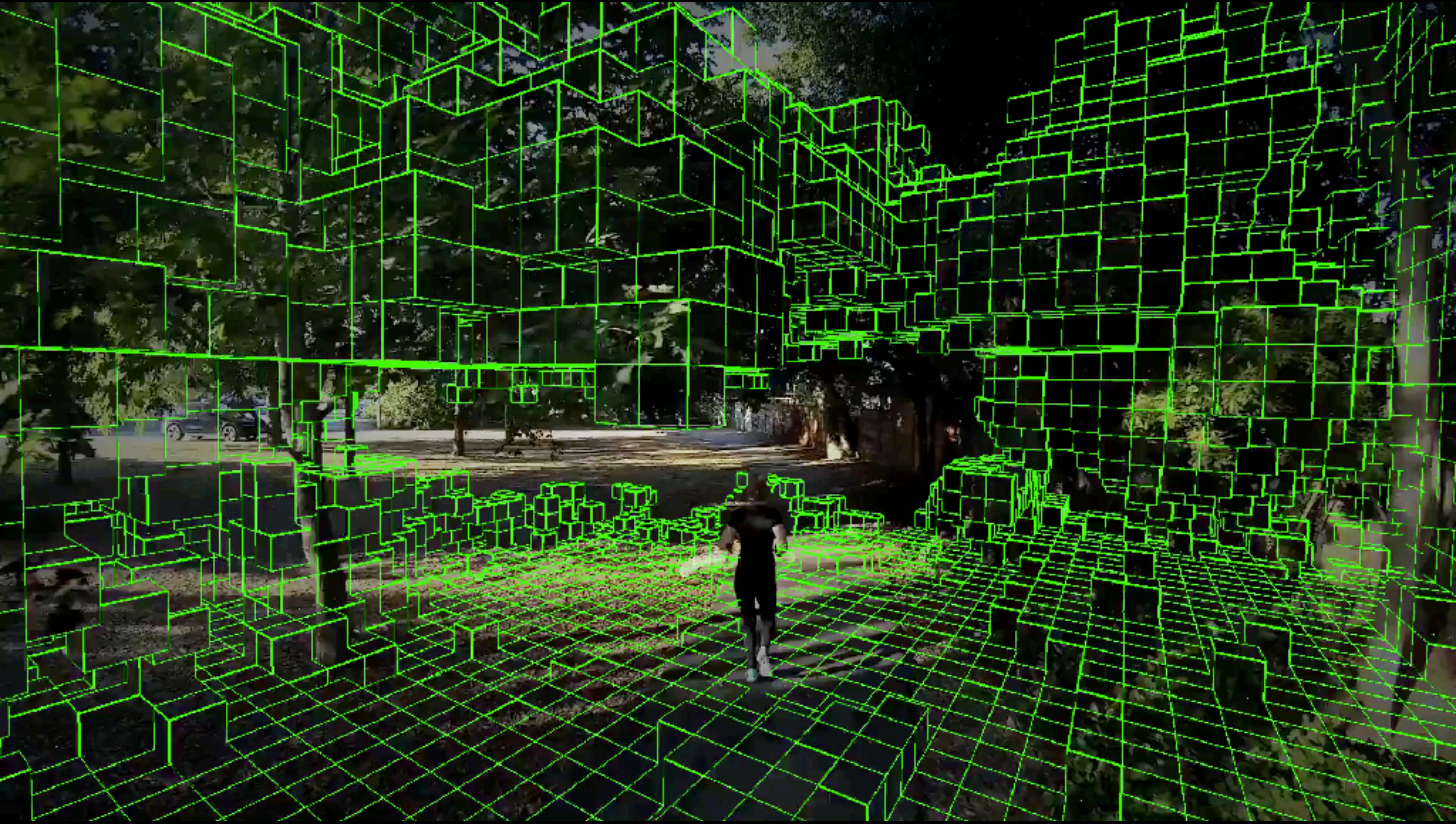
- T is a 4x4 matrix, but is over-parameterized!
- T in SE(3): only 6DOF (3 rotation, 3 translation)

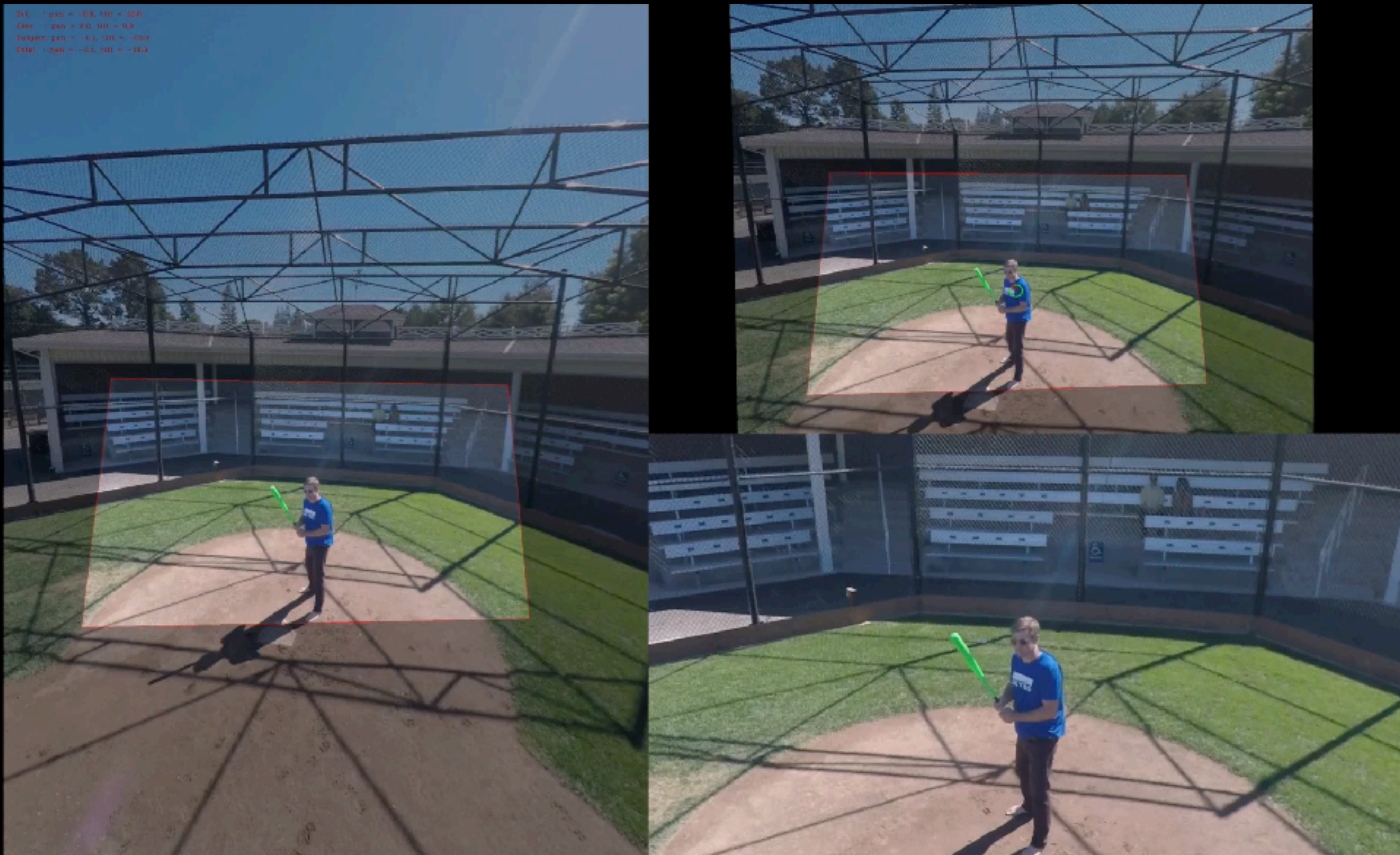
Optimization and Tangent Spaces

- Great 2008 book:
  - "Optimization Algorithms on Matrix Manifolds", by Absil, Mahony, and Sepulchre
- Tangent space provides a local coordinate frame for the manifold, as long as we can "retract" back









Outline

Introduction

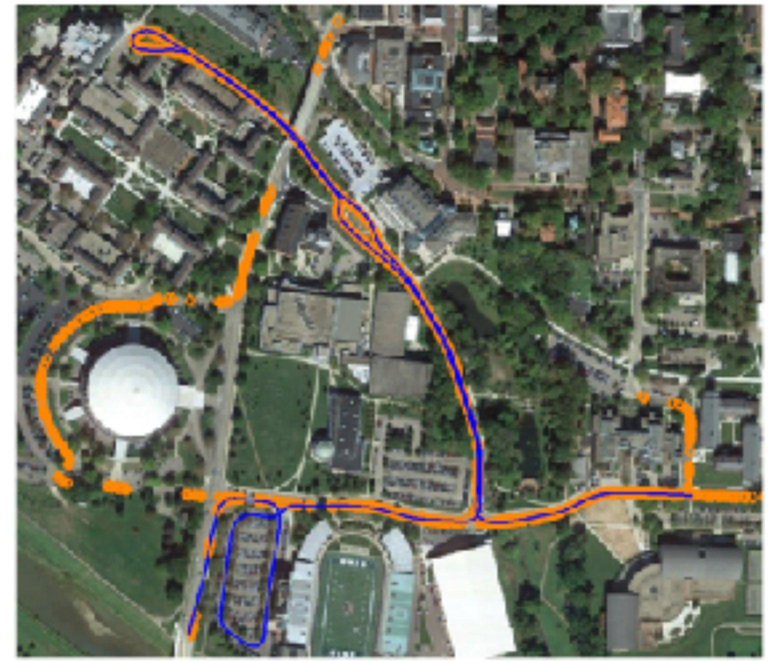
Factor Graphs

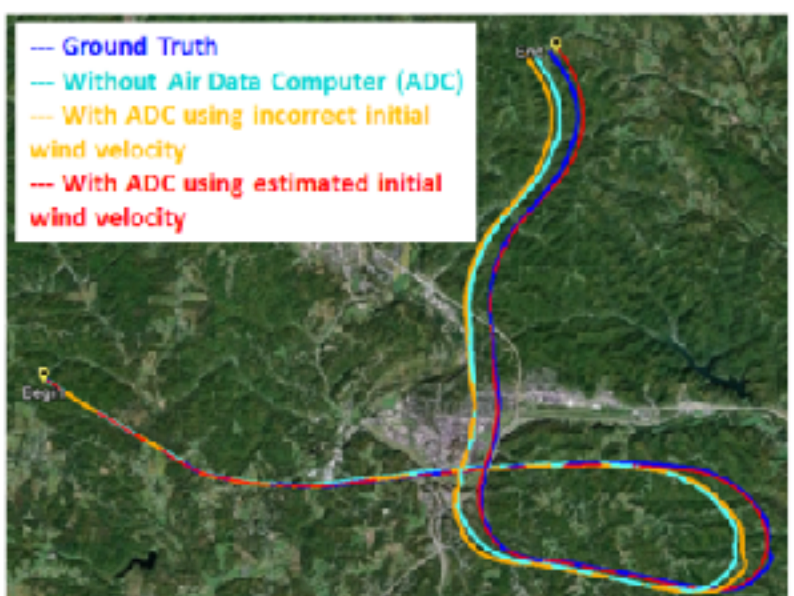
Manifold Optimization

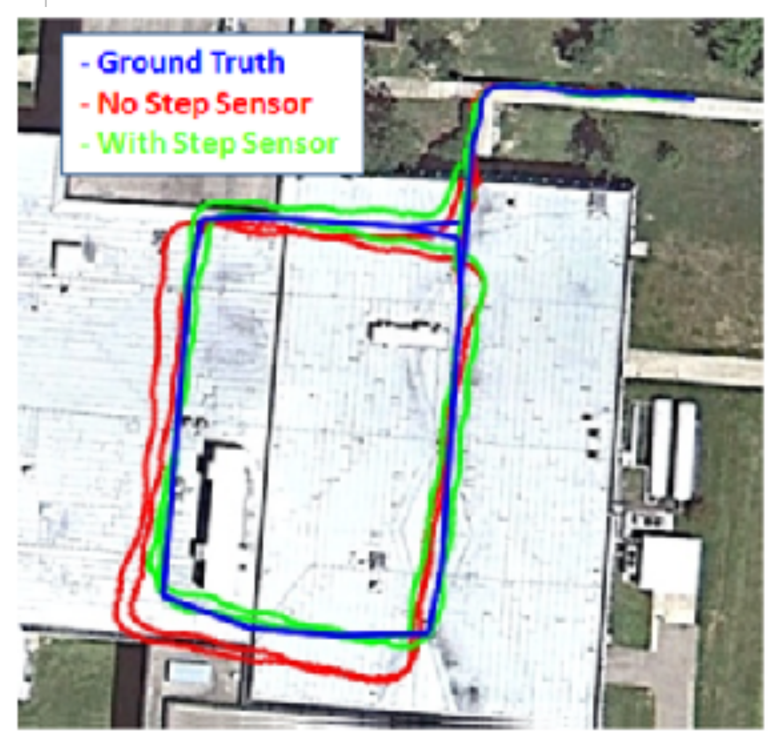
$\boxed{\text{Plug and Play AD}}$

Conclusion

Motivation: Plug-and-Play Navigation (for DARPA)

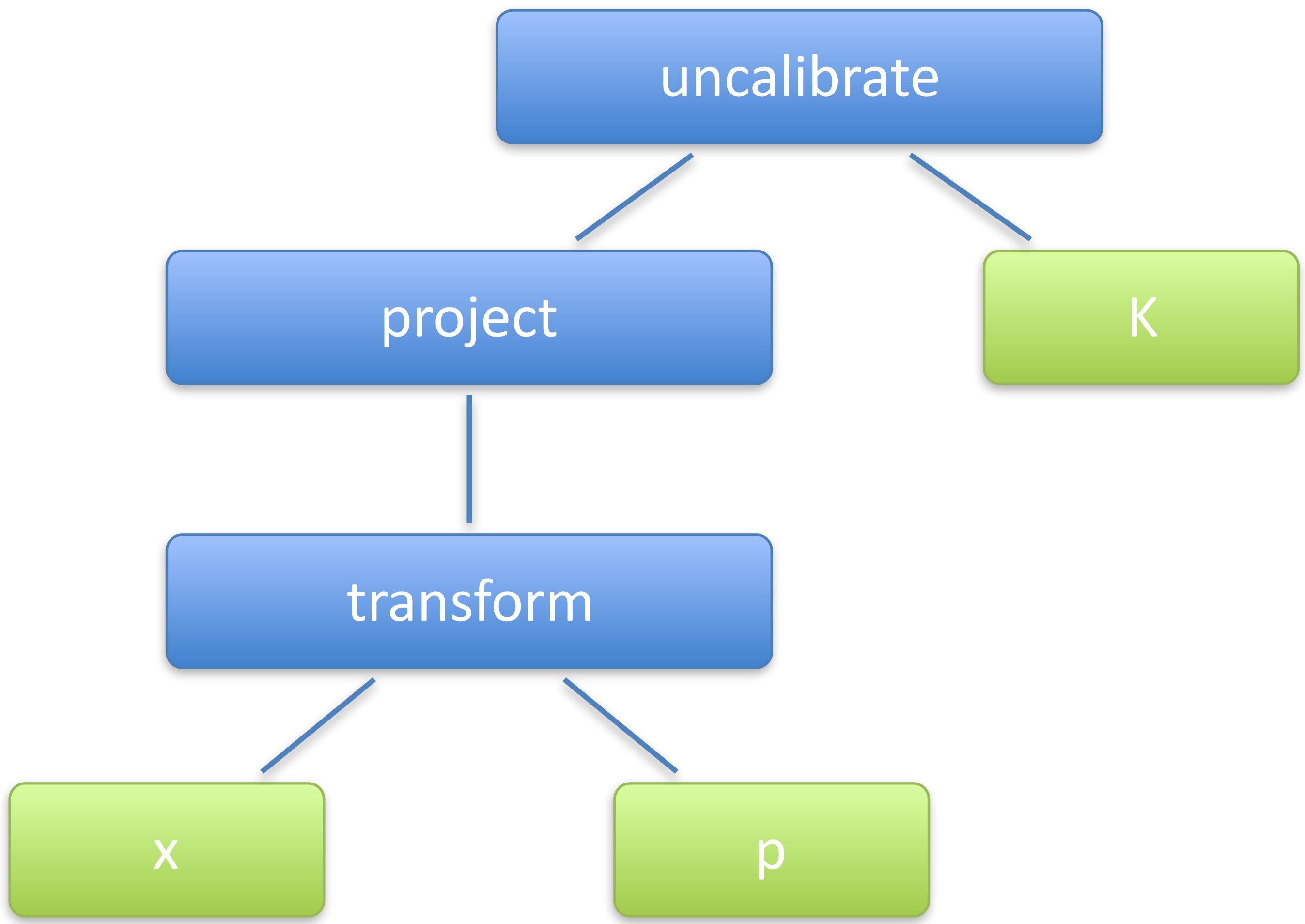






Expressions and Reverse AD

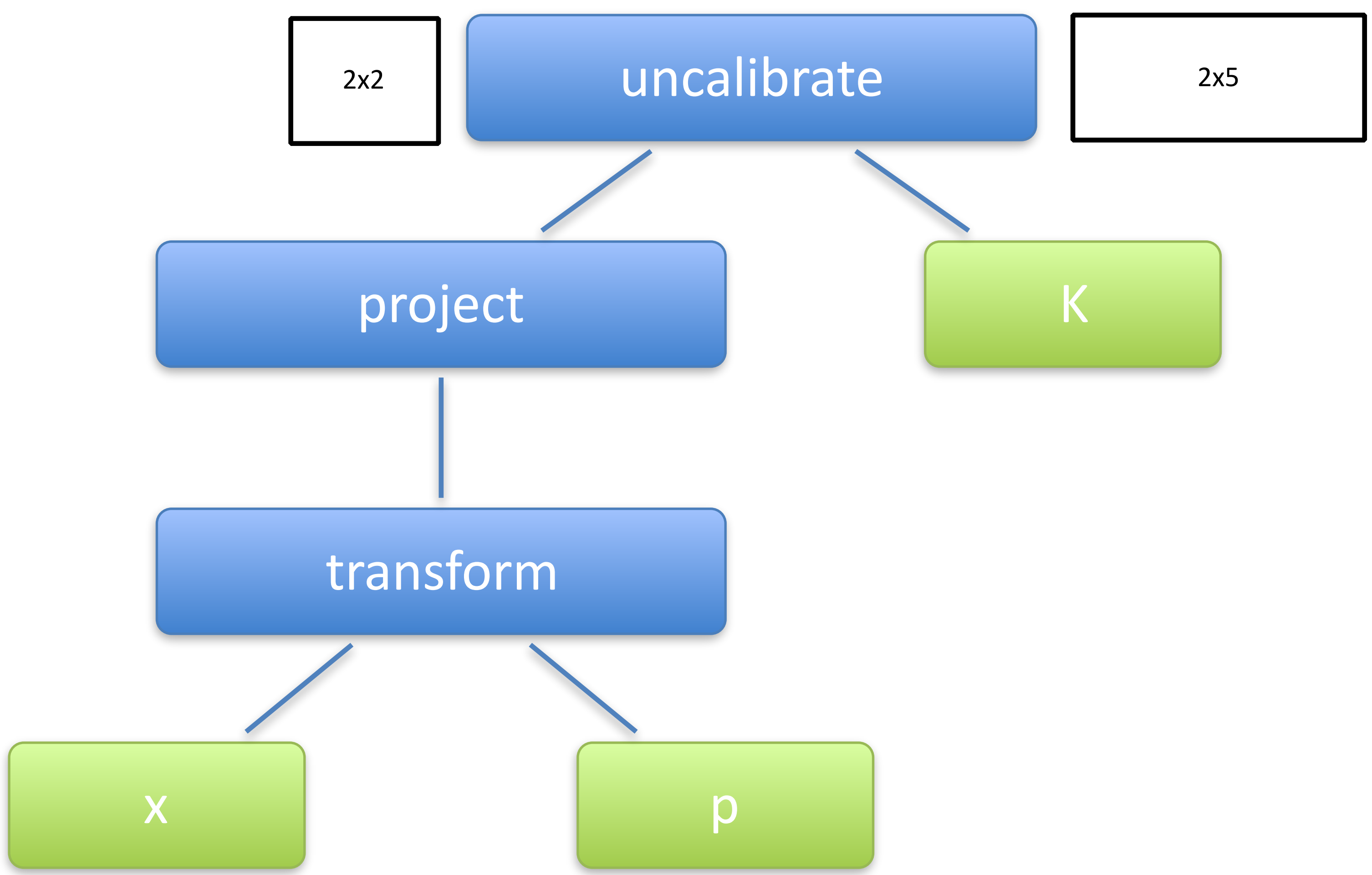
- Writing many different measurement prediction functions with their derivatives (manifold-wise!) is only fun the first 3 times!
- Error-prone and cumbersome any other time.
- Example: `uncalibrate(K, project(transform(x,p)))`



Expressions and Reverse AD

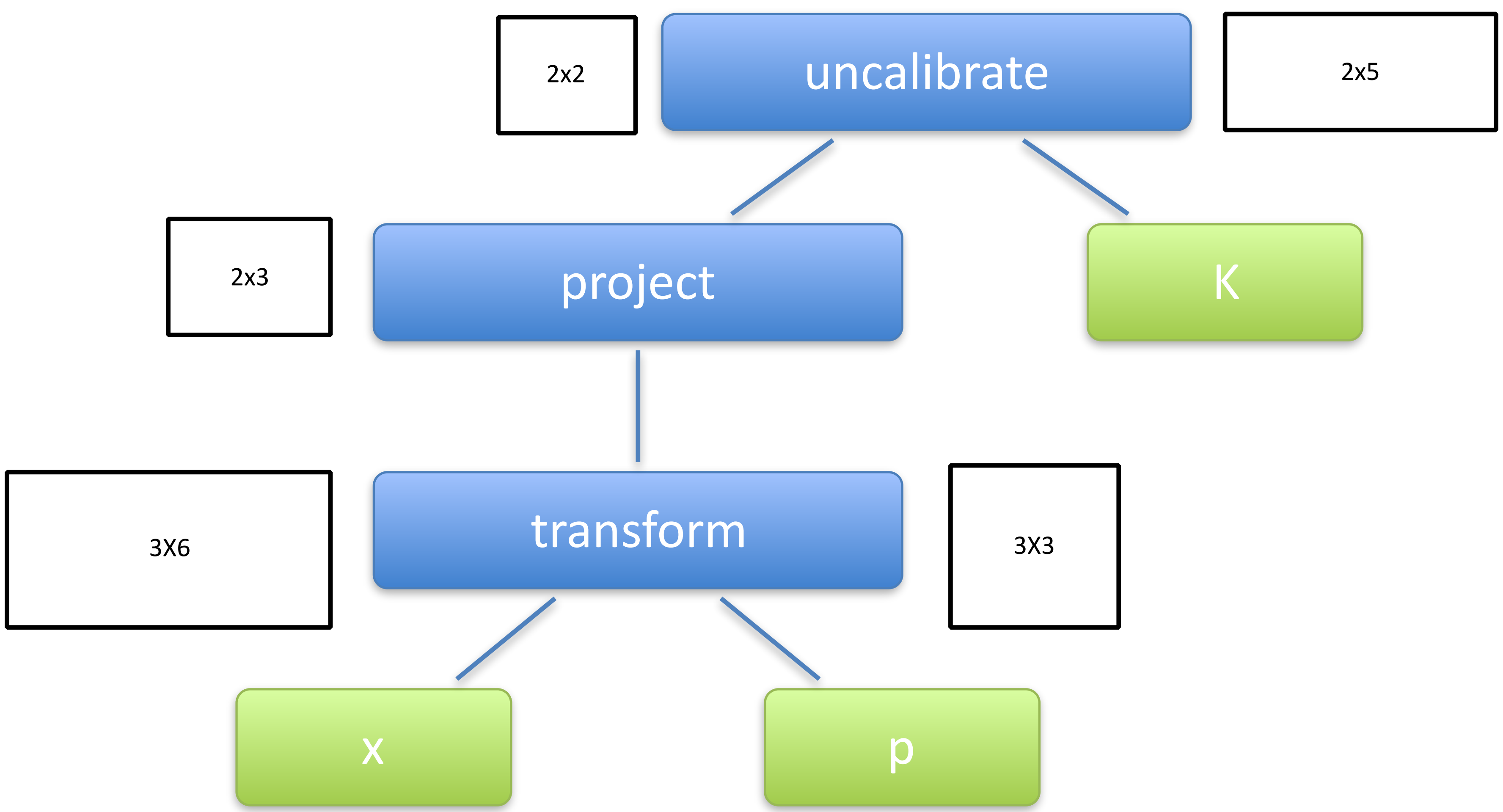
- Base vocabulary of multivariate functions (blue)

```
Point2 Cal3_S2::uncalibrate(const Point2& {x,y}, OptionalJacobian<2, 5> Dcal,
OptionalJacobian<2, 2> Dp) const {
  if (Dcal) *Dcal << x, 0, y, 1, 0, 0, y, 0, 0, 1;
  if (Dp)   *Dp   << fx_, s_, 0, fy_;
  return Point2(fx_ * x + s_ * y + u0_, fy_ * y + v0_);
}
```



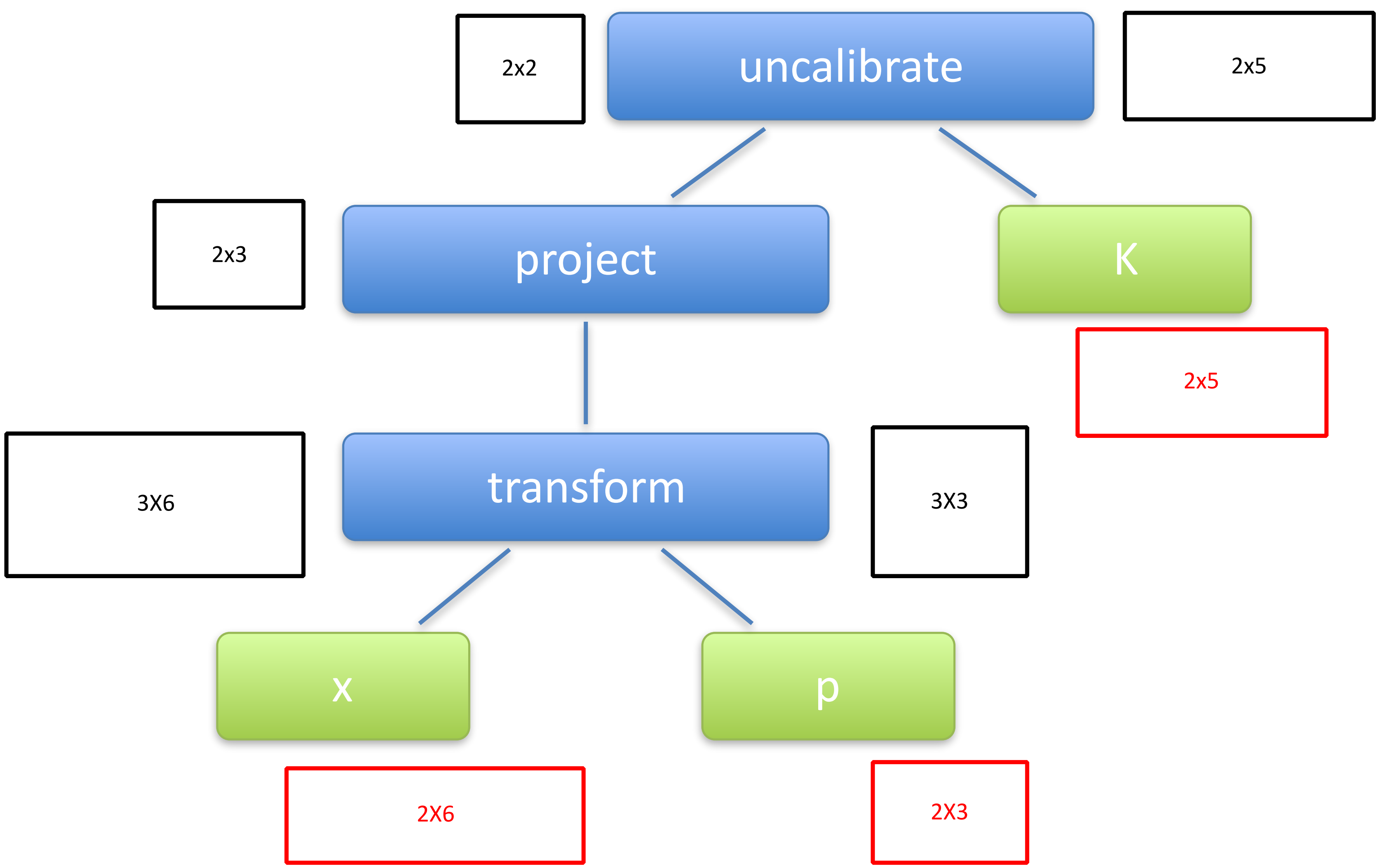
Expressions and Reverse AD

- Reverse AD = same as back-propagation
- Proceed from output to parameter inputs



Expressions and Reverse AD

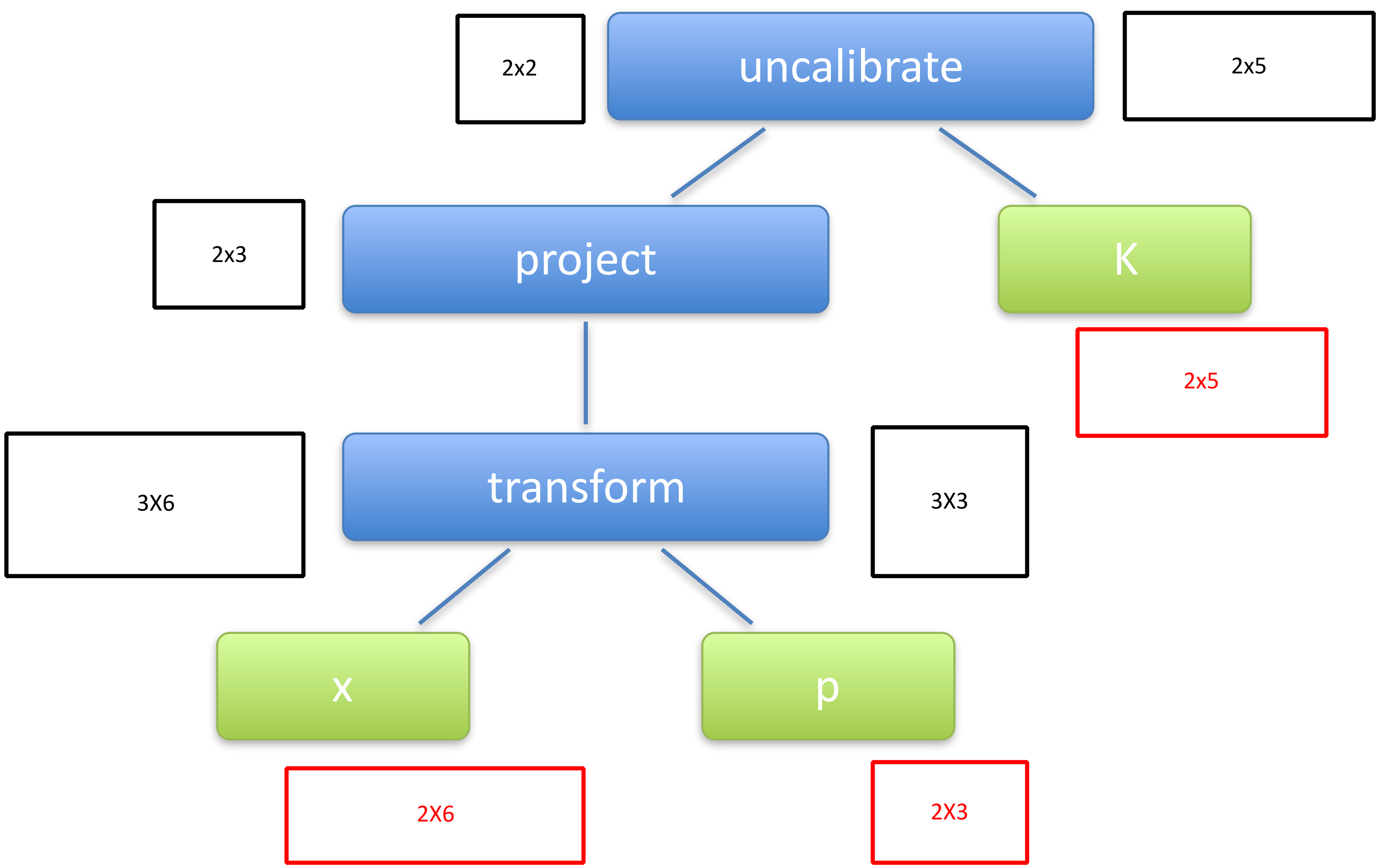
- Reverse AD = same as back-propagation
- Proceed from output to parameter inputs



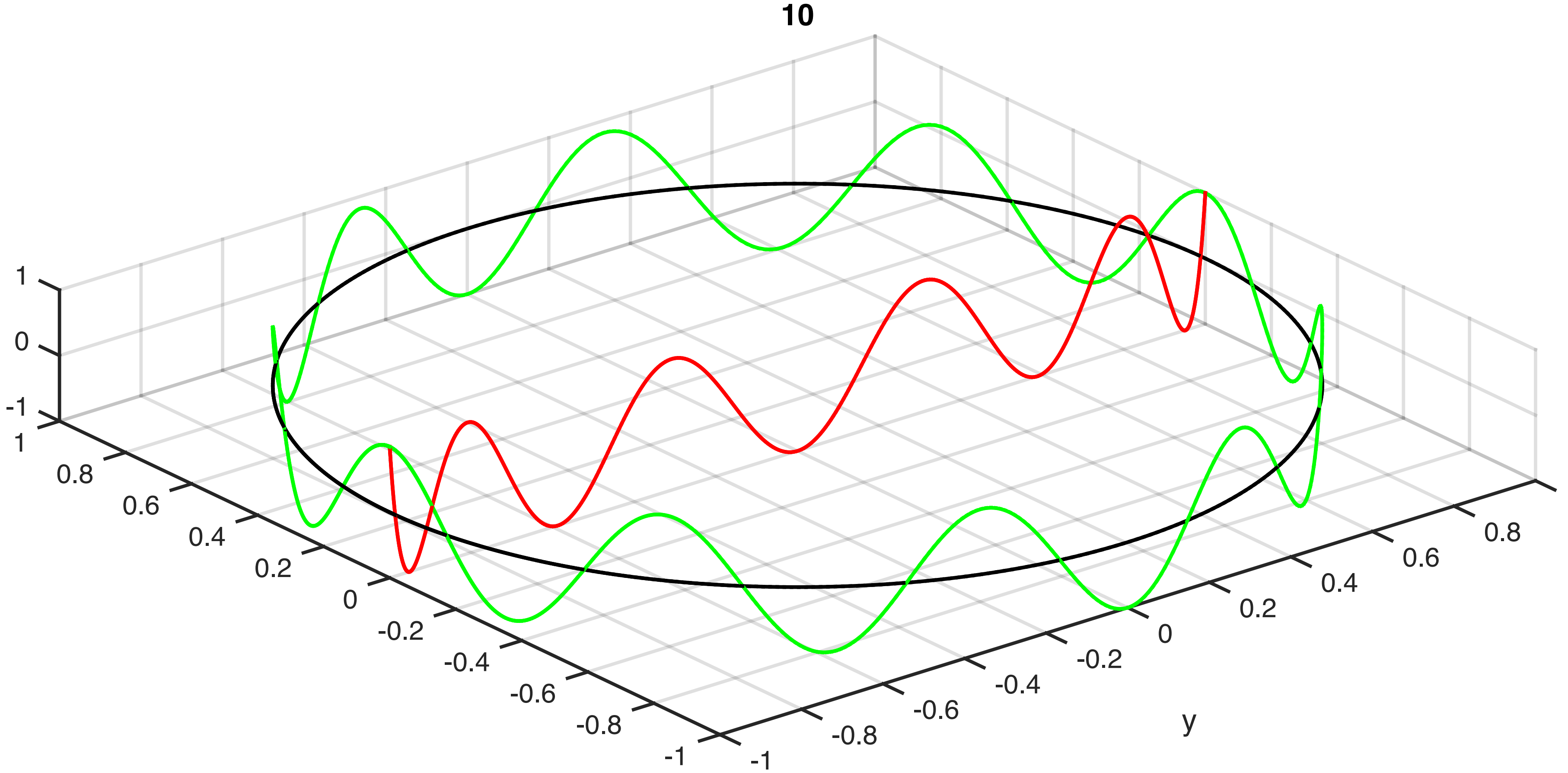
Expressions and Reverse AD

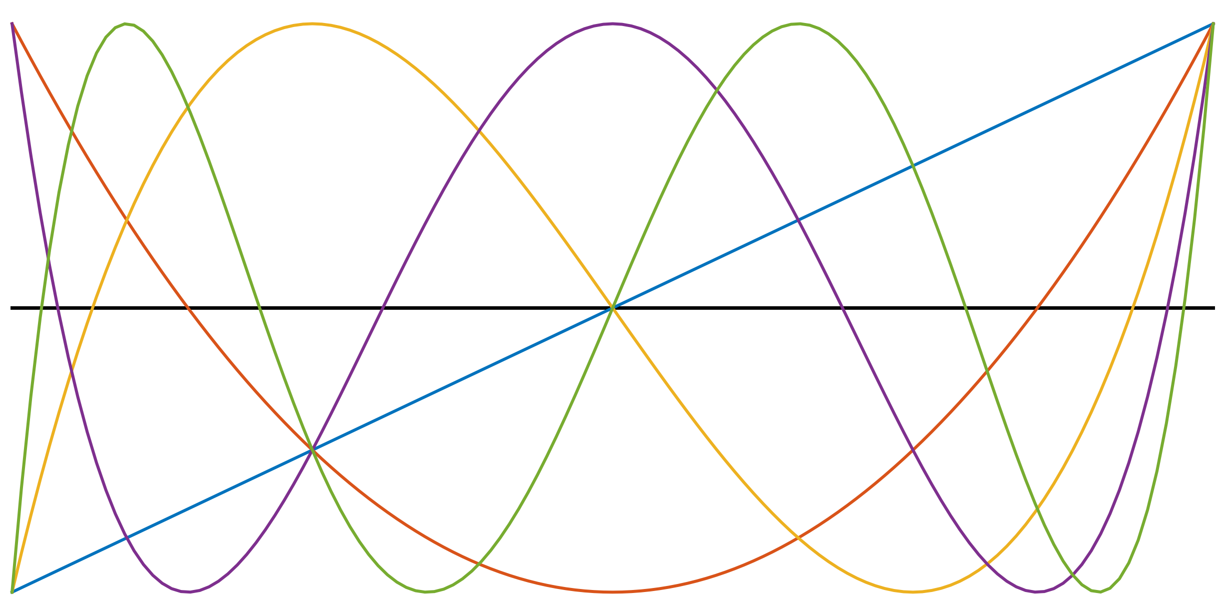
- In GTSAM: lift code to "expressions"
- Efficient reverse AD generated at compile time

```
Expression<Pose3> x; Expression<Point3> p; Expression<Cal3_S2> K;
Expression<Point3> q(x, &Pose3::transform, p);
Expression<Point2> xy(&gtsam::project, q);
Expression<Point2> uv(K, &Cal3_S2::uncalibrate, xy);
```



Chebyshev polynomials **are** Collapsed Cosines



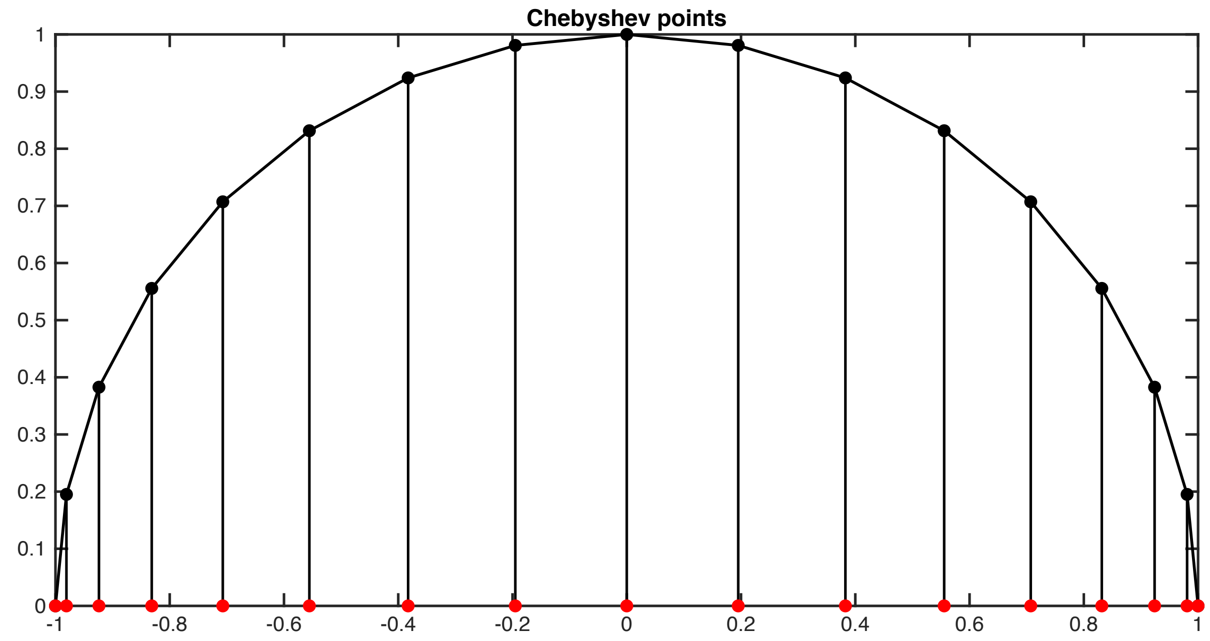


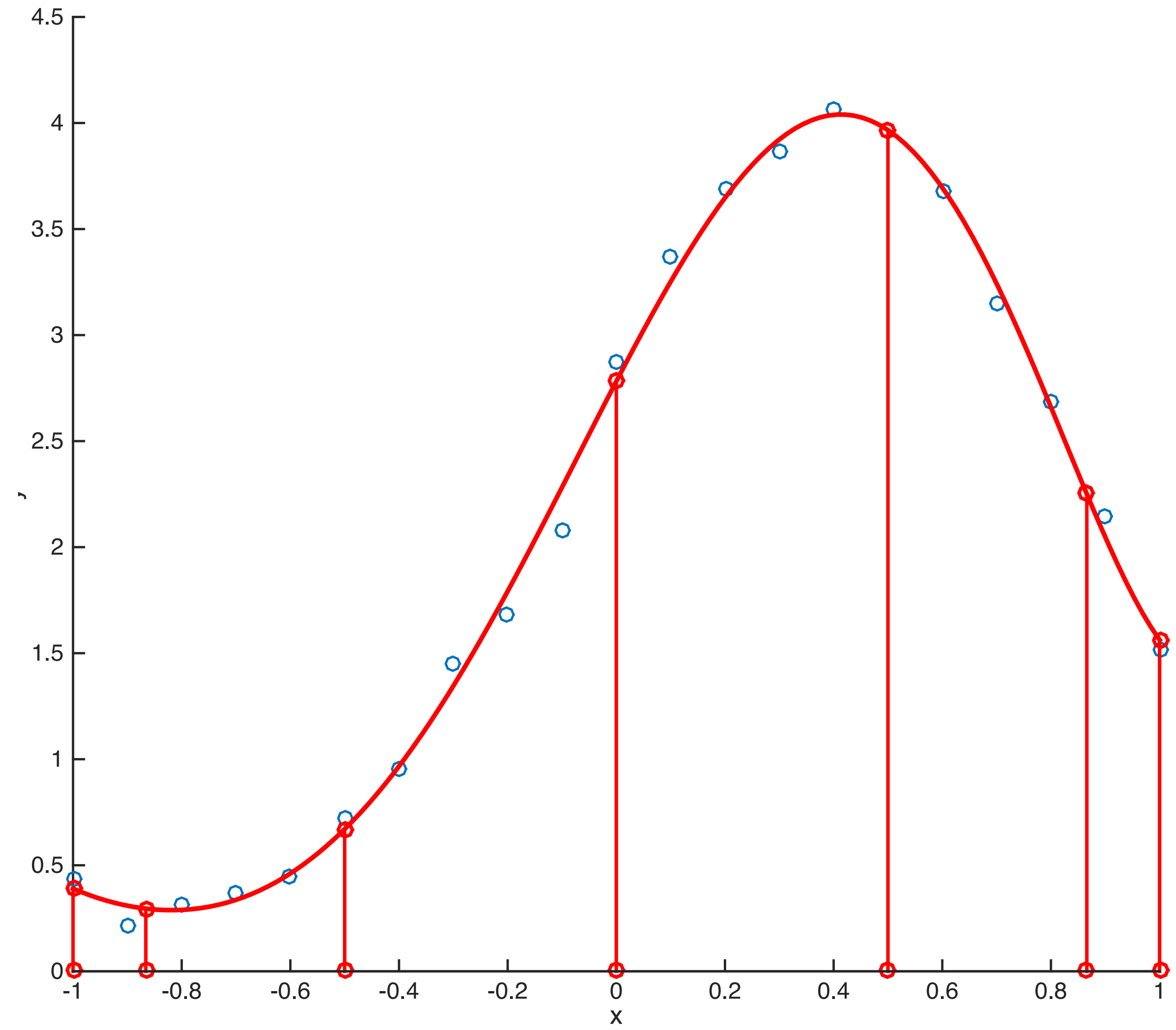
Chebyshev

Nick Trefethen, Oxford

Pseudo-spectral Methods

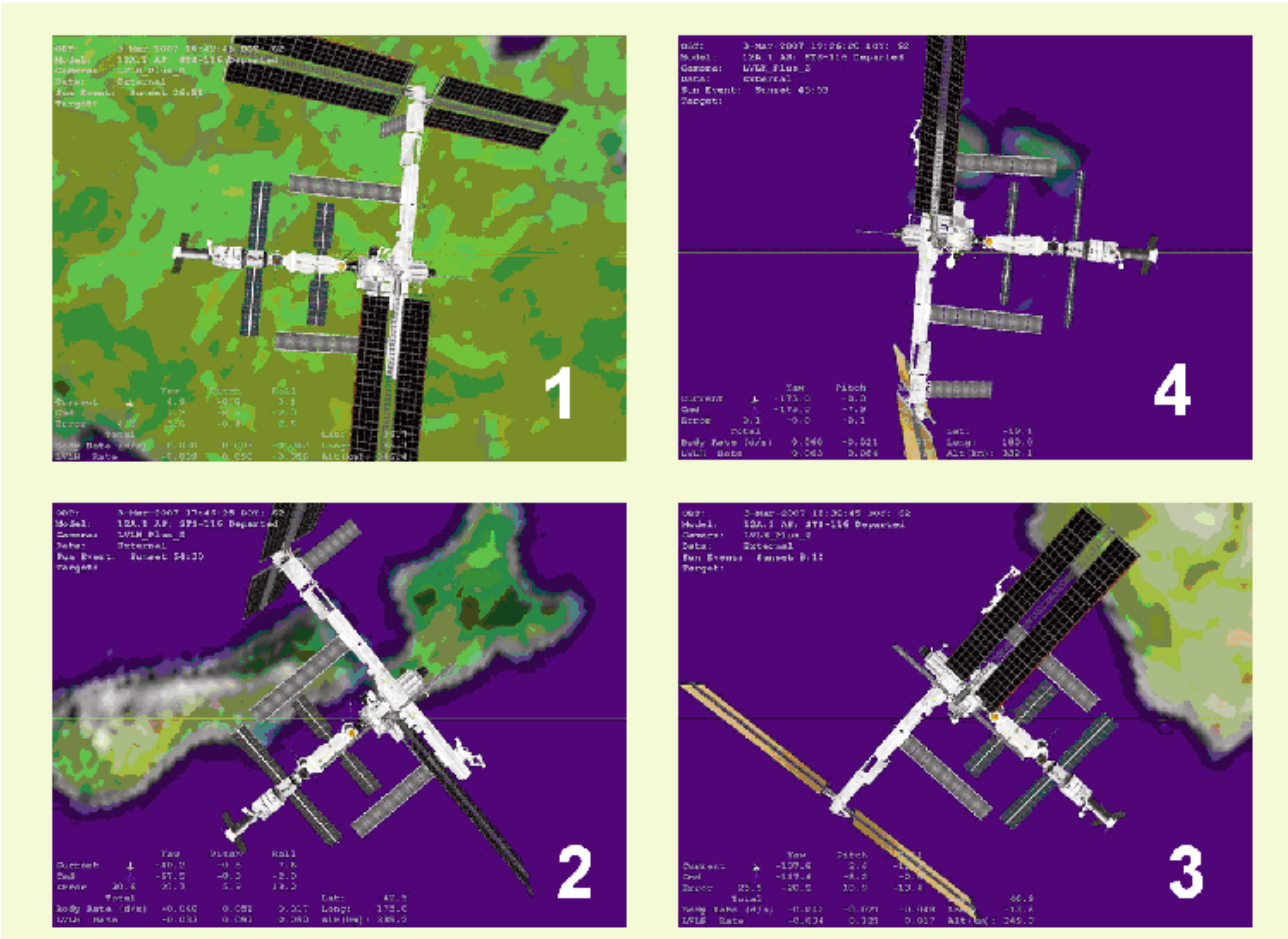
- Parameterize trajectory by values at "Chebyshev" points





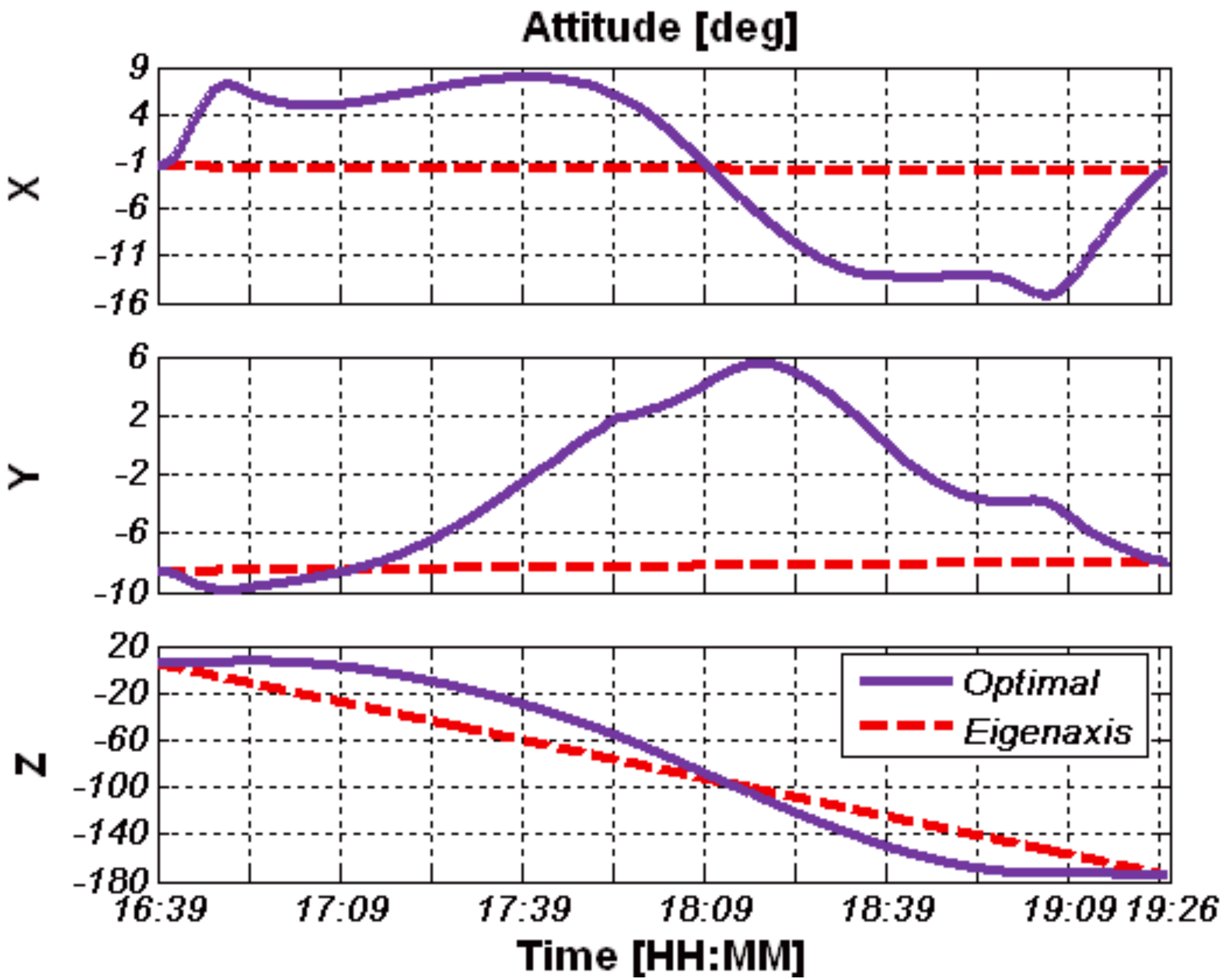
Pseudospectral Optimal Control

- Save NASA $1M !



Pseudospectral Optimal Control

- Save NASA $1M !



Very useful for continuous time trajectory optimization

Use PSOC Ideas for SLAM

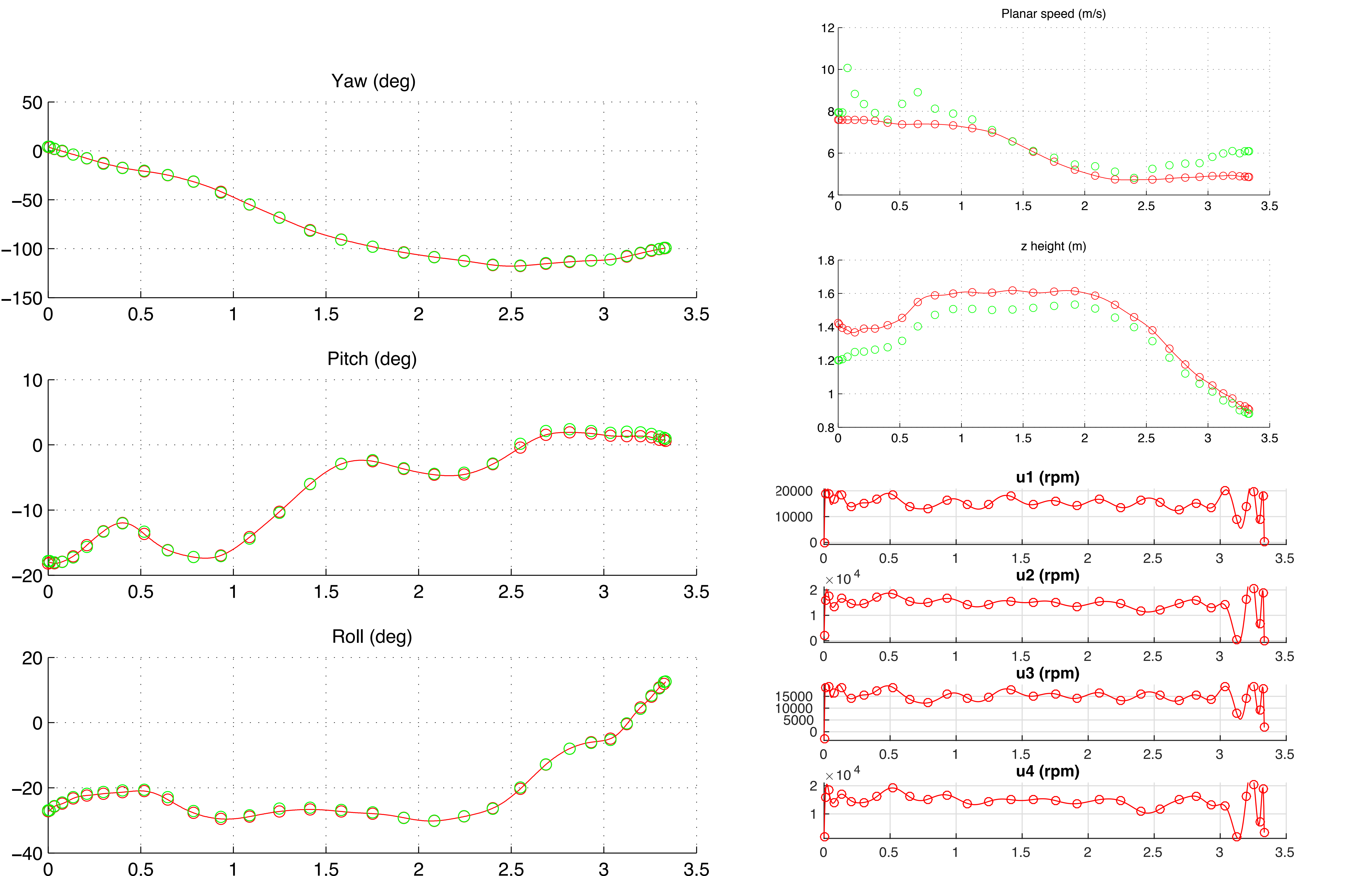
"Collocation": dynamic Defects

$$
\begin{aligned}
\dot{x}(t) &= v(t)\cos\theta(t) \\
\dot{y}(t) &= v(t)\sin\theta(t) \\
\dot{\theta}(t) &= \omega(t)
\end{aligned}
$$

$$E_1(\mathbf{X}, \Theta) = \sum_{i=1}^{m} \left\| \mathbf{z}_i - \mathbf{h}_i\left(\mathbf{X}\mathbf{w_i}, \Theta_i\right) \right\|_{R_i}^2$$

$$E_2\left(\mathbf{X}, \mathbf{U}\right) \triangleq \sum_{j=0}^{N} \left\| \mathbf{X}\mathbf{D}_N\mathbf{w_j} - \mathbf{f}\left(\mathbf{X}\mathbf{w_j}, \mathbf{U}\mathbf{w_j}, t_j\right) \right\|_Q^2$$

Some States and Controls (preliminary)



Outline

Introduction

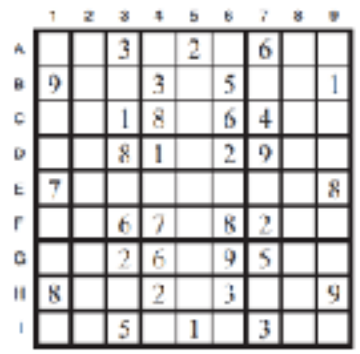
Factor Graphs

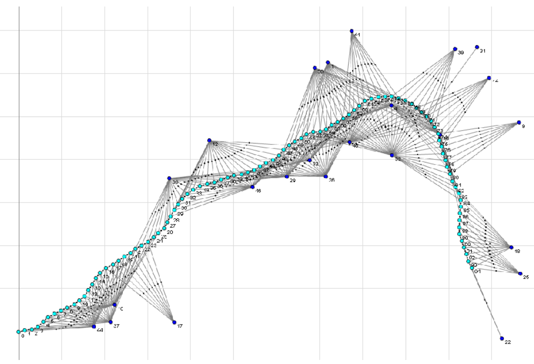
Manifold Optimization

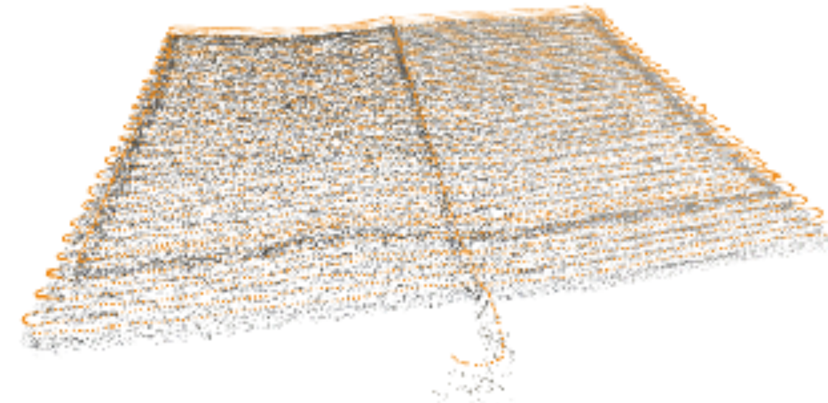
Plug and Play AD

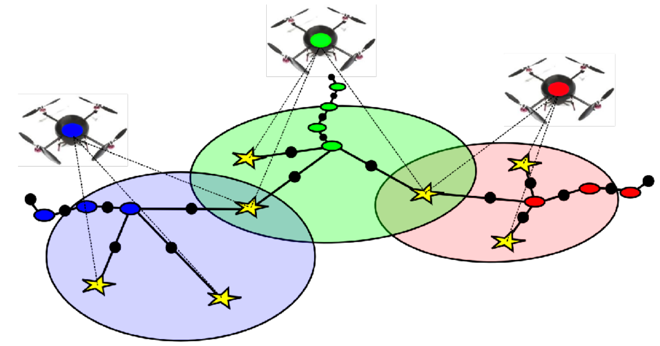
$\boxed{\text{Conclusion}}$

Factor graphs as a Lingua Franca in Robotics…





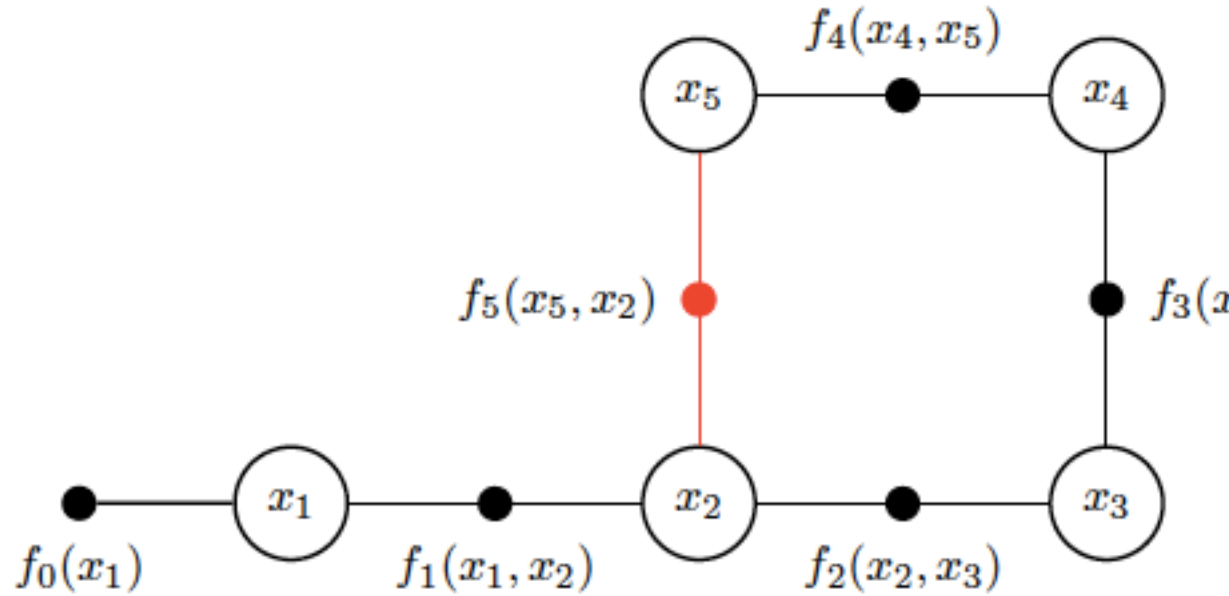




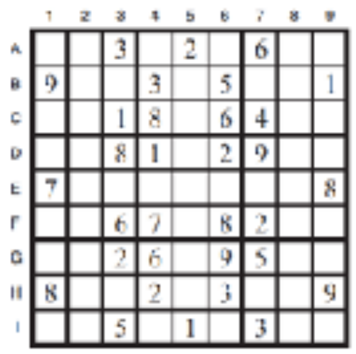
Factor graphs are a graphical language in which to pose a wide variety of inference problems in robotics.

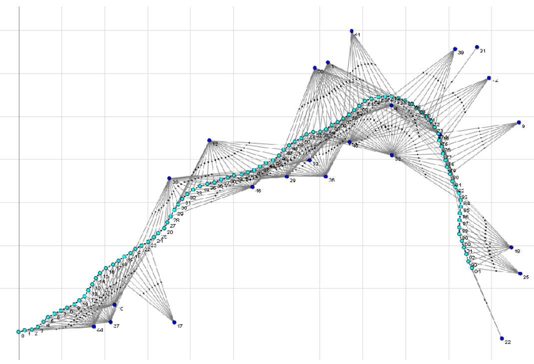
GTSAM embodies many of these ideas

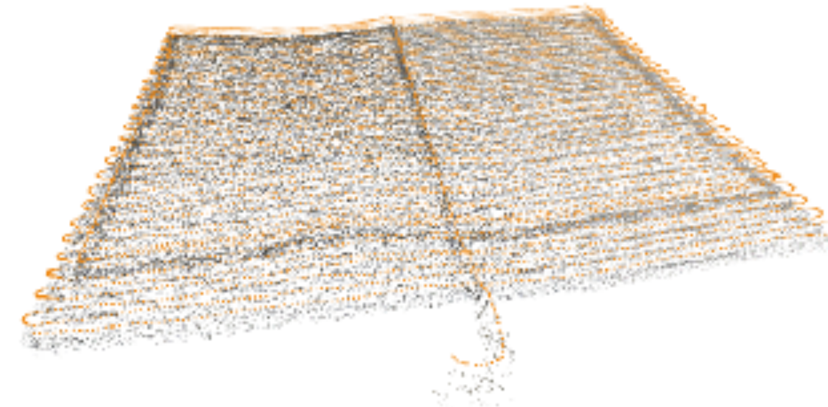
- C++ library
- python & matlab wrappers
- Open, BSD-licensed
- based entirely on Factor Graph:
- Optimization on Manifolds and Lie groups

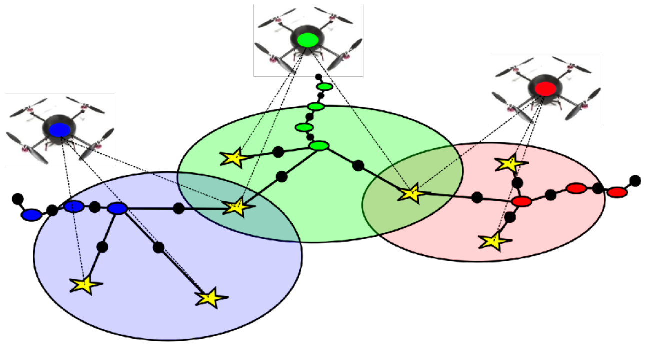



```
NonlinearFactorGraph graph;
noiseModel::Diagonal::shared_ptr priorNoise =
  noiseModel::Diagonal::Sigmas(Vector_(3, 0.3, 0.3, 0.1));
graph.add(PriorFactor<Pose2>(1, Pose2(0,0,0), priorNoise));

// Add odometry factors
noiseModel::Diagonal::shared_ptr model =
  noiseModel::Diagonal::Sigmas(Vector_(3, 0.2, 0.2, 0.1));
graph.add(BetweenFactor<Pose2>(1, 2, Pose2(2, 0, 0     ), model));
graph.add(BetweenFactor<Pose2>(2, 3, Pose2(2, 0, M_PI_2), model));
graph.add(BetweenFactor<Pose2>(3, 4, Pose2(2, 0, M_PI_2), model));
graph.add(BetweenFactor<Pose2>(4, 5, Pose2(2, 0, M_PI_2), model));

// Add pose constraint
graph.add(BetweenFactor<Pose2>(5, 2, Pose2(2, 0, M_PI_2), model));
```



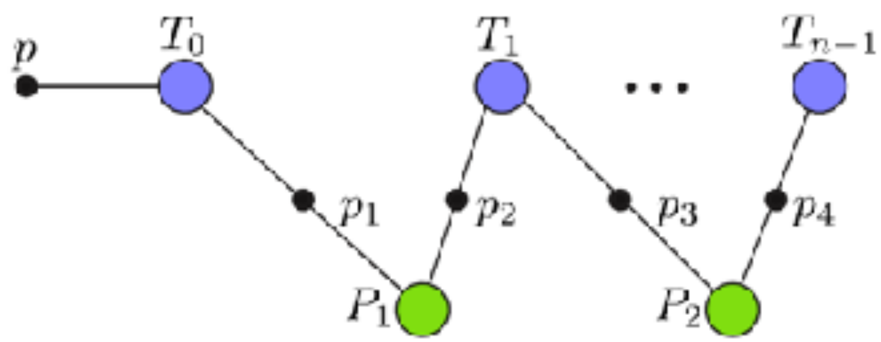






Questions ??

Nonlinear Least-Squares



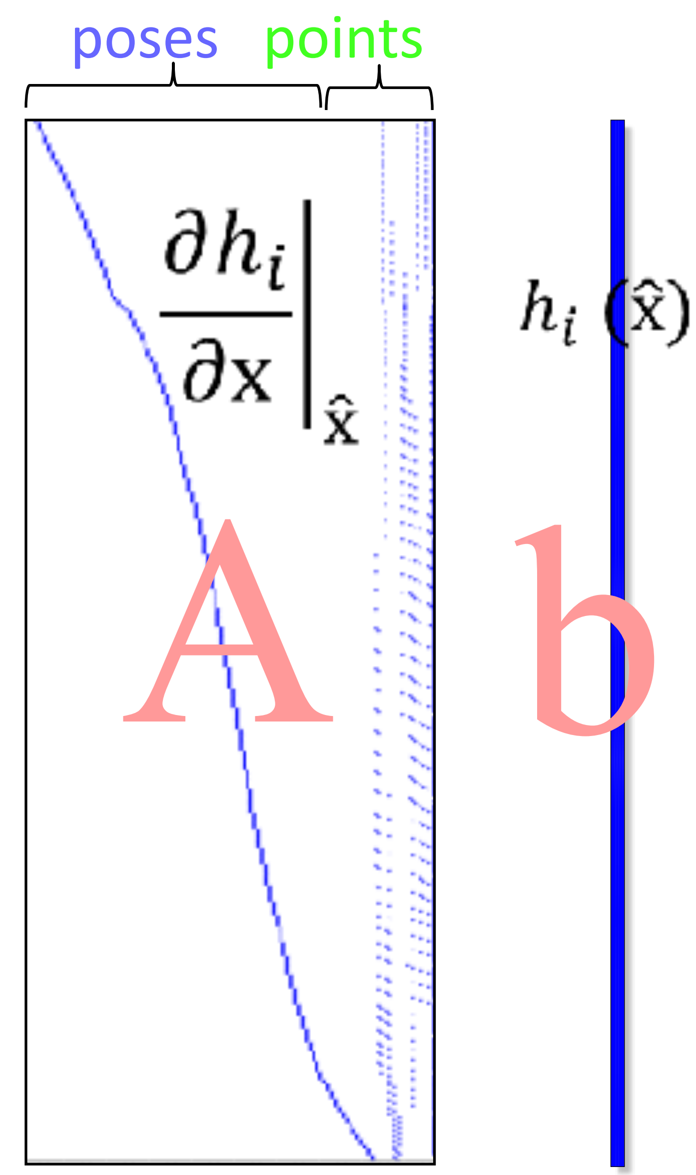
Gaussian noise

$$\operatorname{argmin}_{\mathrm{x}} \sum_i \|h_i(\mathrm{x})\|_{\Xi}{}^2$$

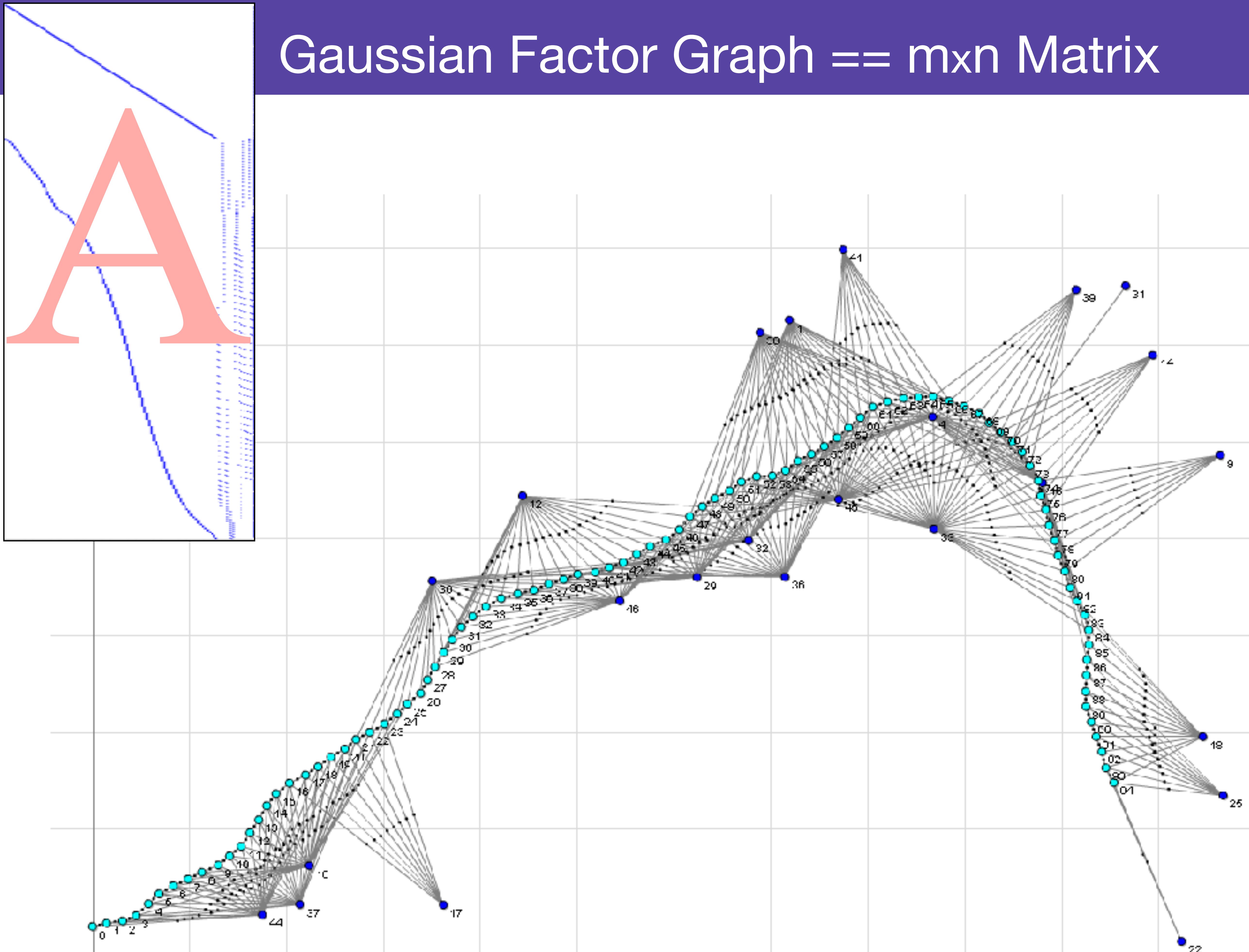
Repeatedly solve linearized system (GN)

$$\operatorname{argmin}_x \|Ax - b\|^2$$

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

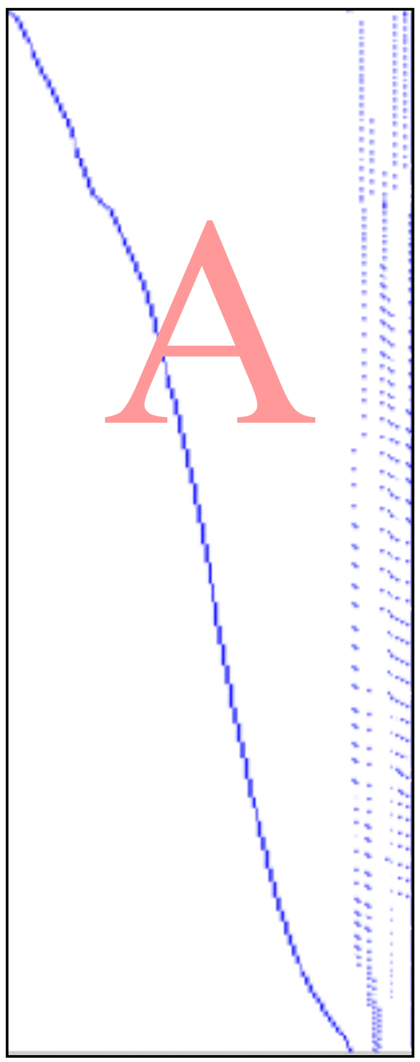


Gaussian Factor Graph == mxn Matrix



Solving the Linear Least-Squares System

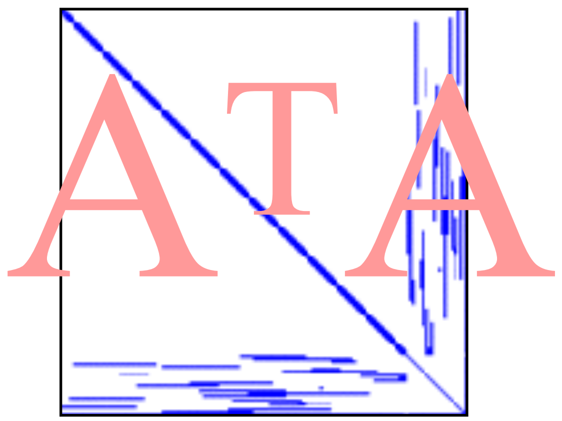
Solve: $\operatorname{argmin}_x \|Ax - b\|^2$



Measurement Jacobian

Normal equations

$$A^T A x = A^T b$$



Information matrix

Solving the Linear Least-Squares System

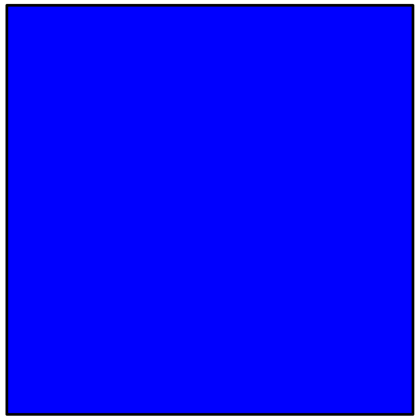
- Can we simply invert A^TA to solve for x?

Normal equations

$$A^T A x = A^T b$$

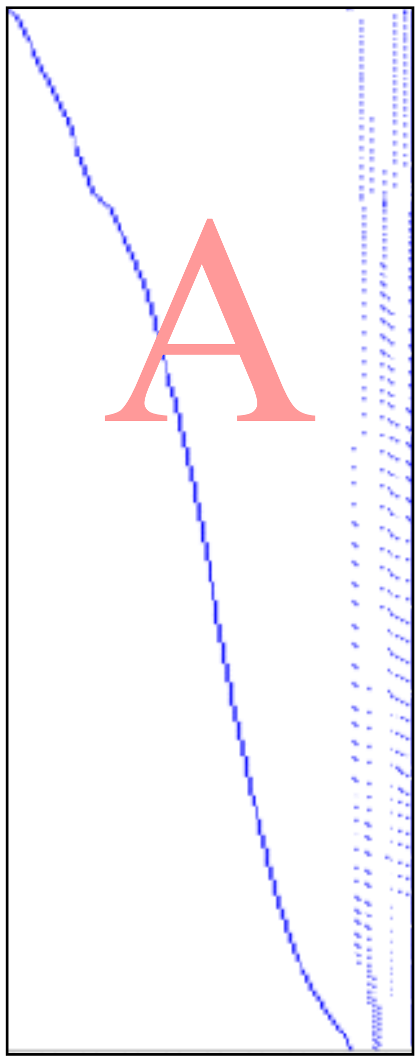
Information matrix

- Yes, but we shouldn't…
  The inverse of A^TA is dense -> O(n^3)
- Can do much better by taking advantage of sparsity!



Solving the Linear Least-Squares System

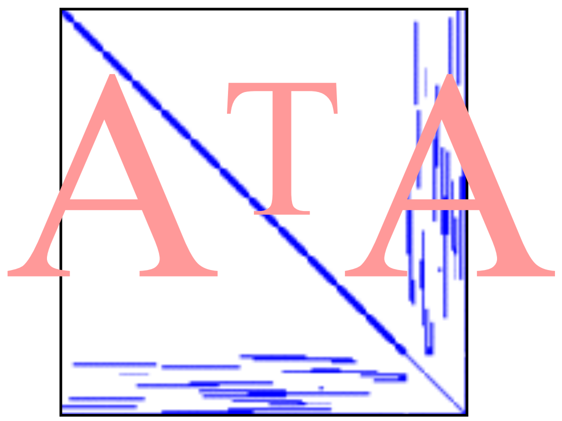
Solve: $\mathrm{argmin}_x \|Ax - b\|^2$



Measurement Jacobian

Normal equations

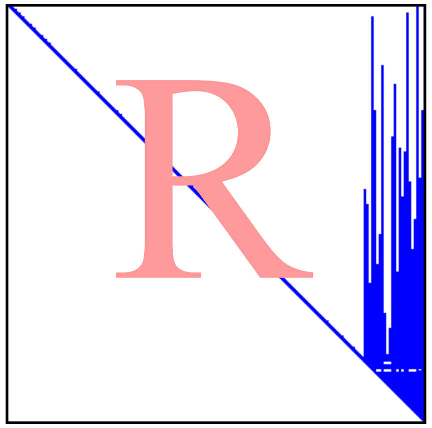
$$A^T A x = A^T b$$



Information matrix

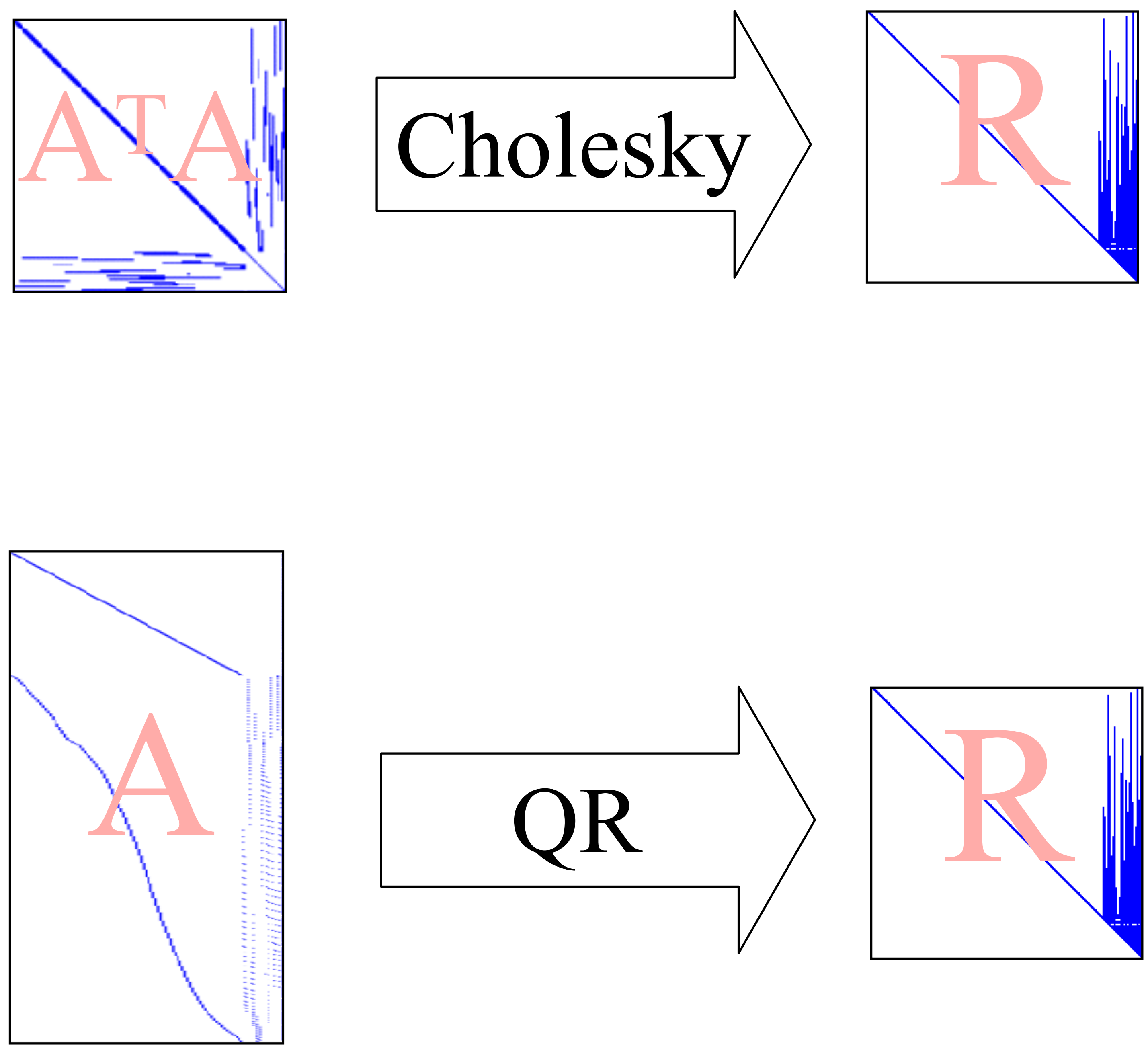
Matrix factorization

$$A^T A = R^T R$$



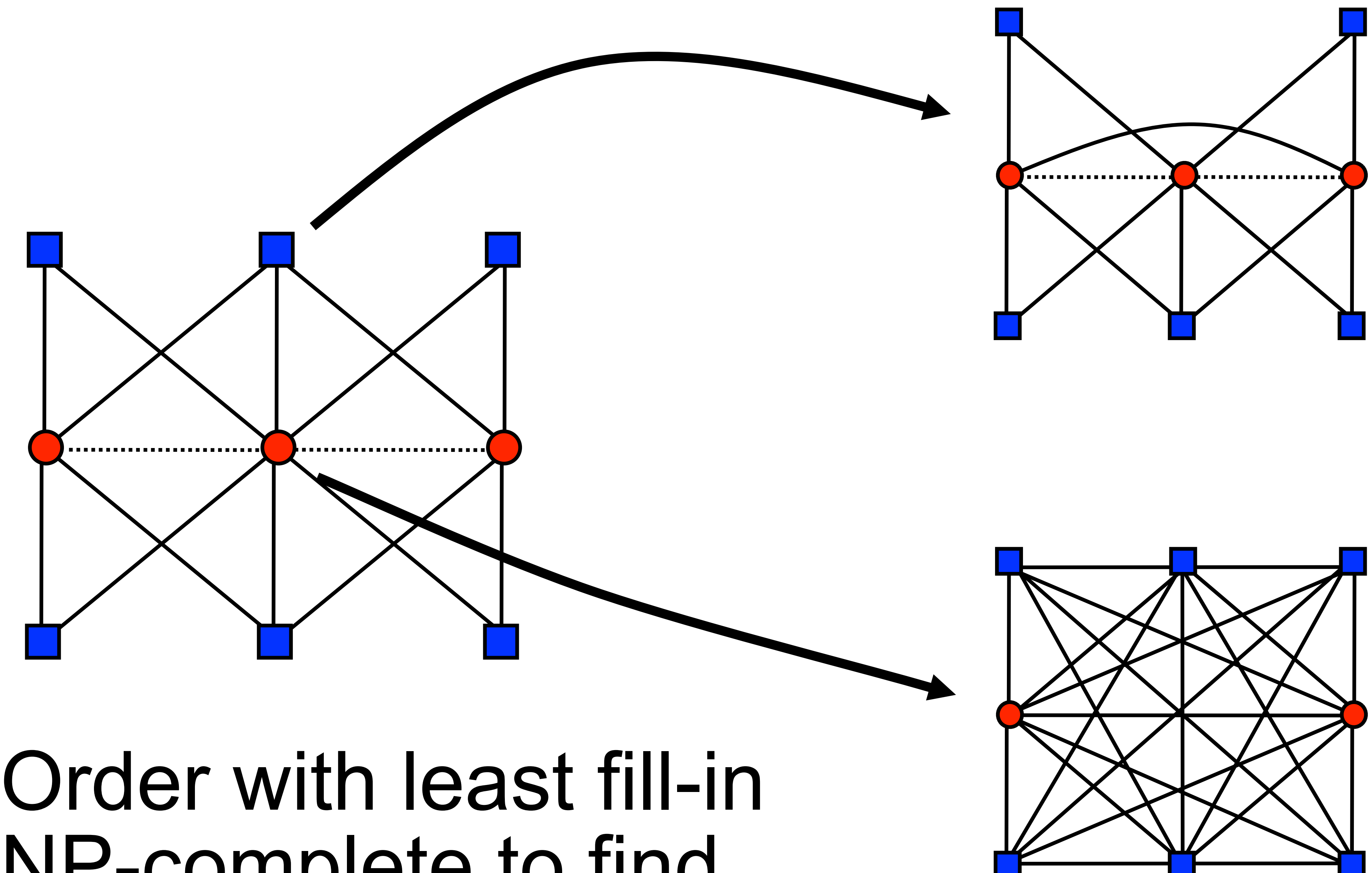
Square root information matrix

Square Root Factorization



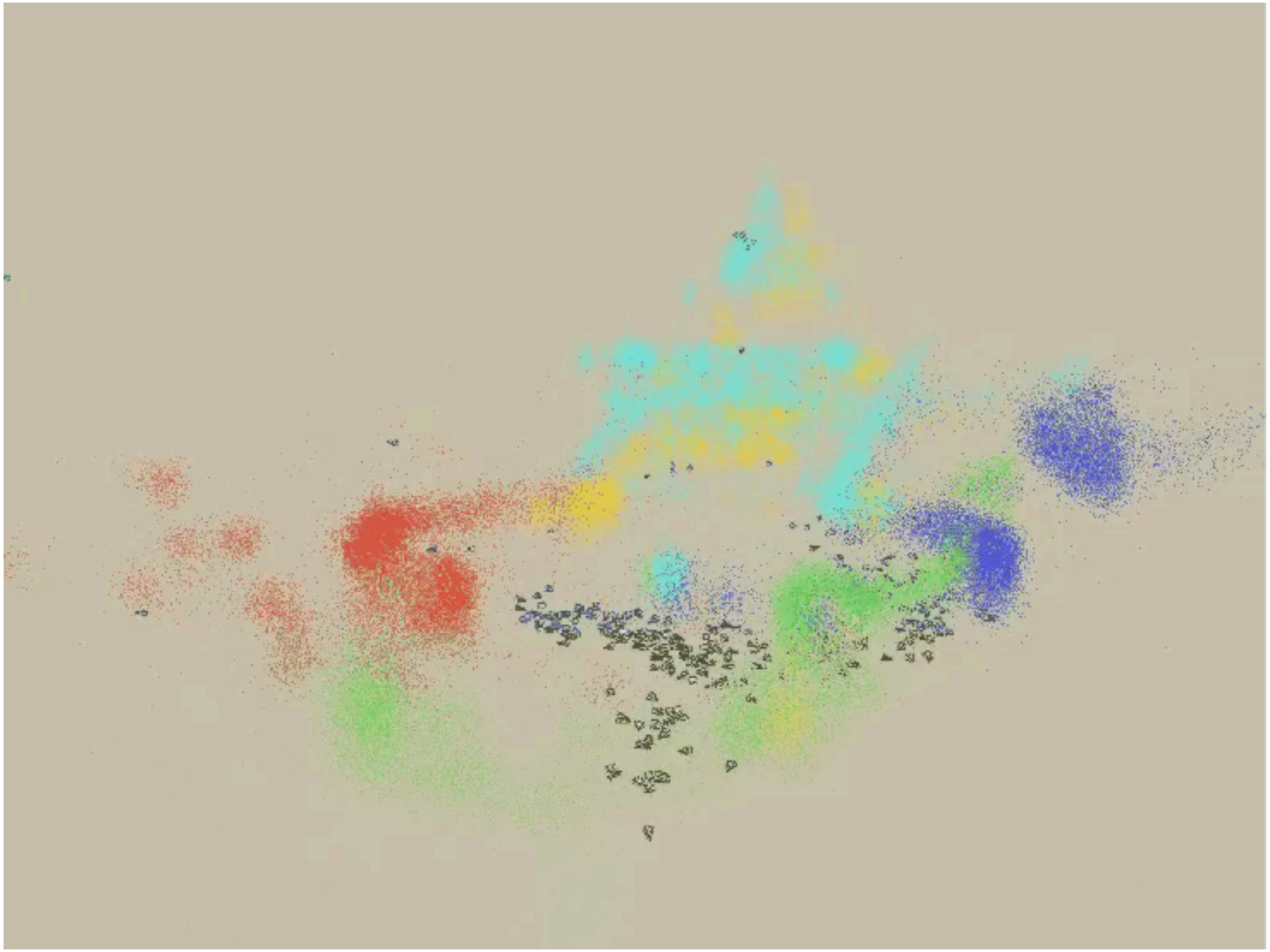
Elimination order is again CRUCIAL

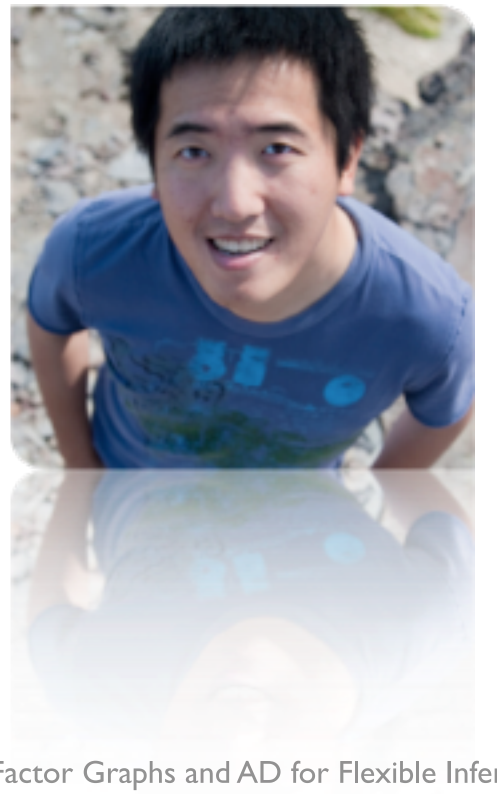
Square Root SAM: Simultaneous Location and Mapping via Square Root Information Smoothing, *Frank Dellaert*, **Robotics: Science and Systems, 2005**
Exploiting Locality by Nested Dissection For Square Root Smoothing and Mapping, *Peter Krauthausen, Frank Dellaert, and Alexander Kipp*, **Robotics: Science and Systems, 2006**



- Order with least fill-in NP-complete to find

St. Peter’s Basilica, Rome



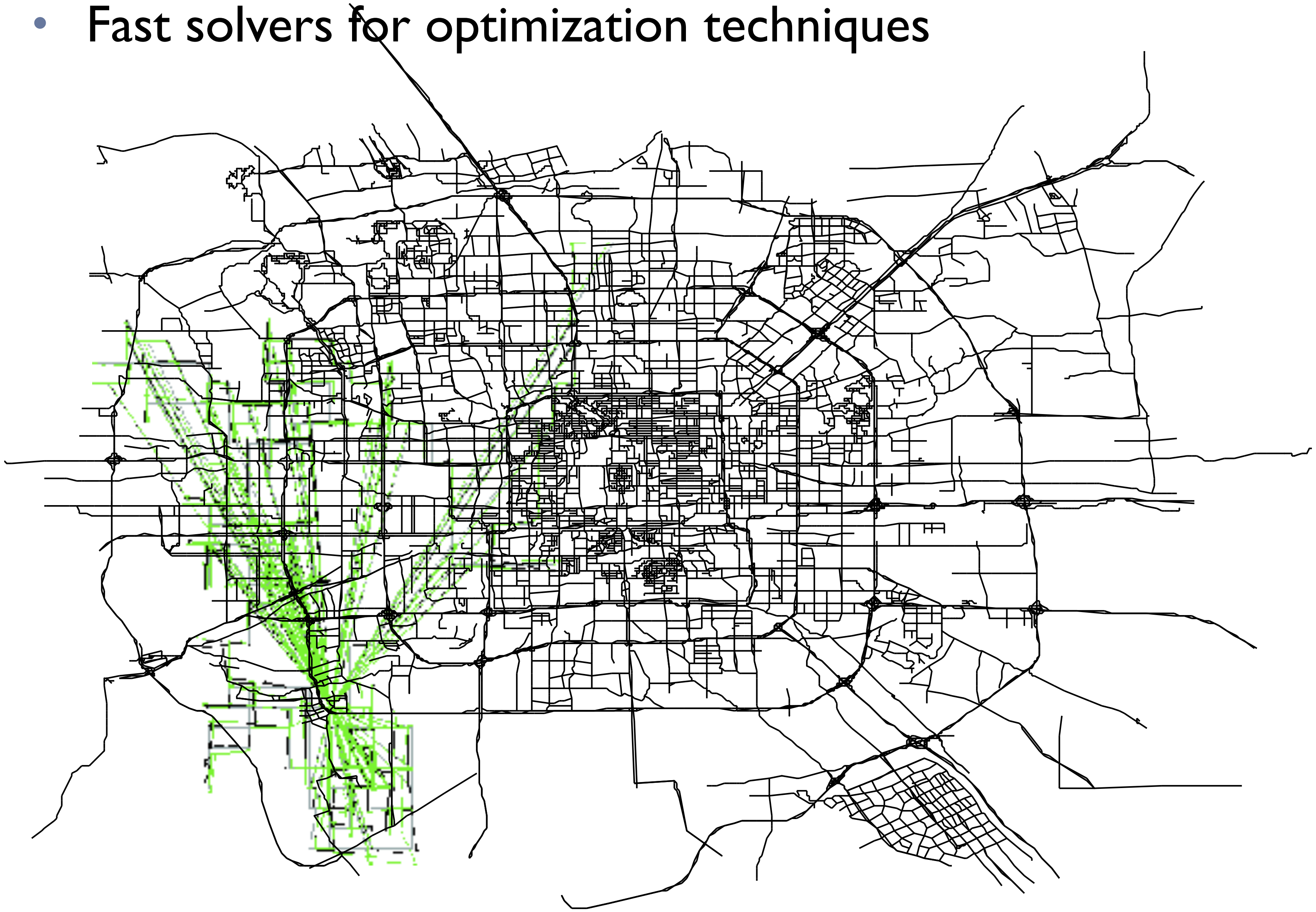


Data Courtesy Microsoft Research

Kai Ni, Drew Steedly, and Frank Dellaert, Out-of-Core Bundle Adjustment for Large-Scale 3D Reconstruction, *IEEE International Conference on Computer Vision (ICCV), 2007*.

Of Eiffel Towers....

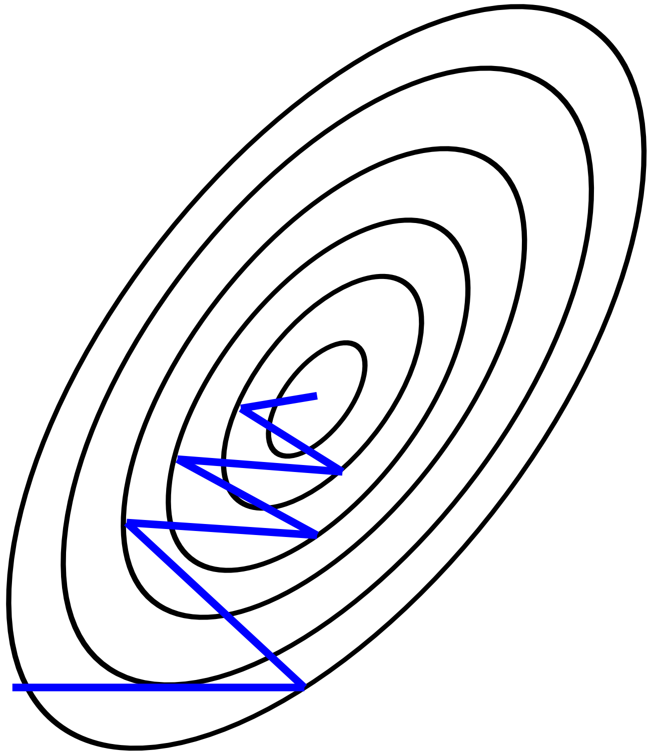
- SLAM is a large optimization problem
- Fast solvers for optimization techniques



Iterative Methods

- Better than direct methods for large problems because they
  - Perform only simple operations and no variable elimination
  - Require constant (linear) memory

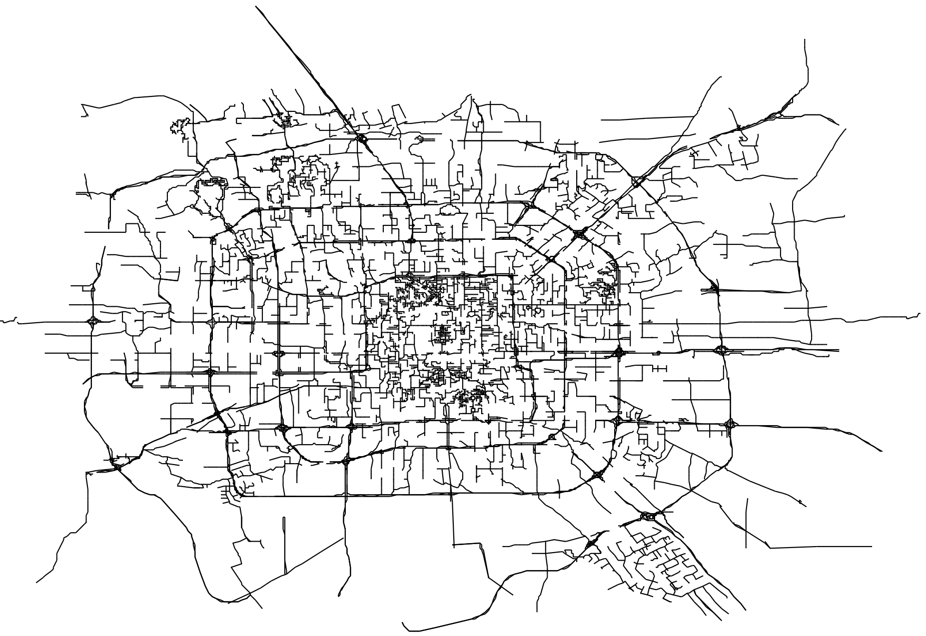
- But they may converge slowly
  when the problem is ill-conditioned



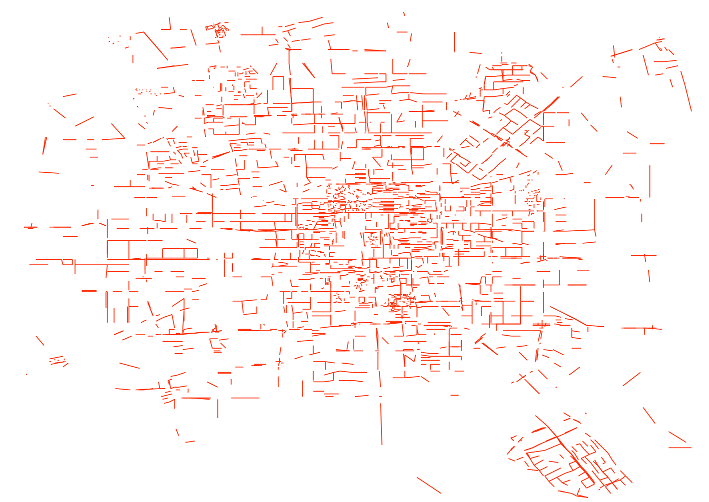
- The conjugate gradient method: The convergence speed is related to the condition number $\kappa(\mathbf{A}^t\mathbf{A}) = \lambda_{max}/\lambda_{min}$

Main Idea (with Viorela Ila and Doru Balcan)

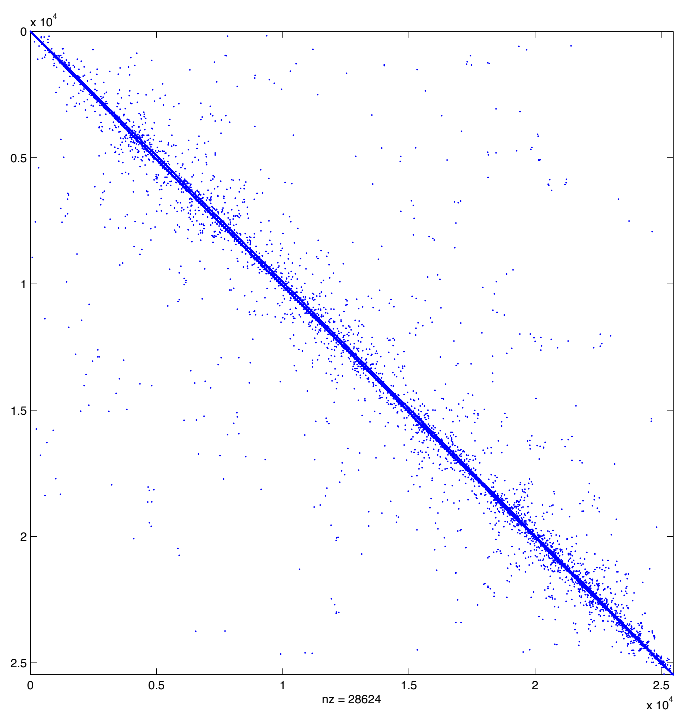
The whole problem =

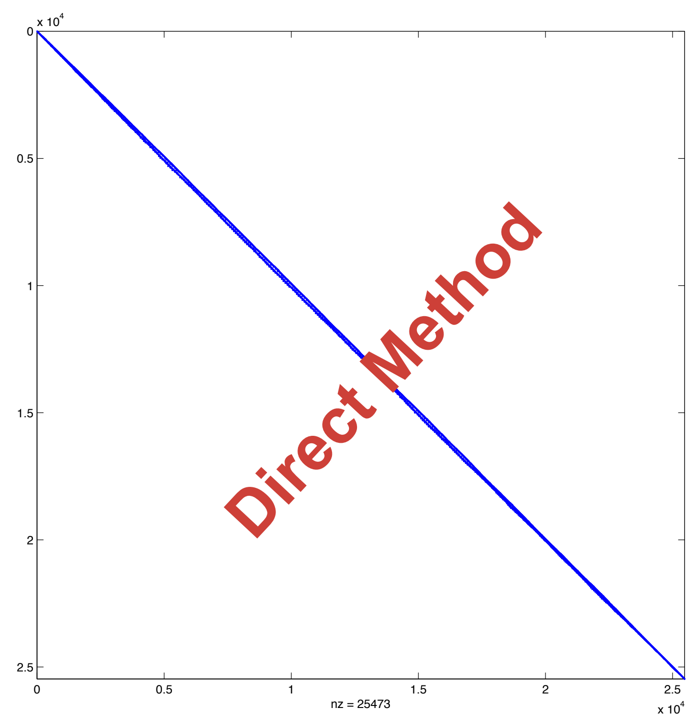


The easy part (subgraph) +



The hard part (loop closures)





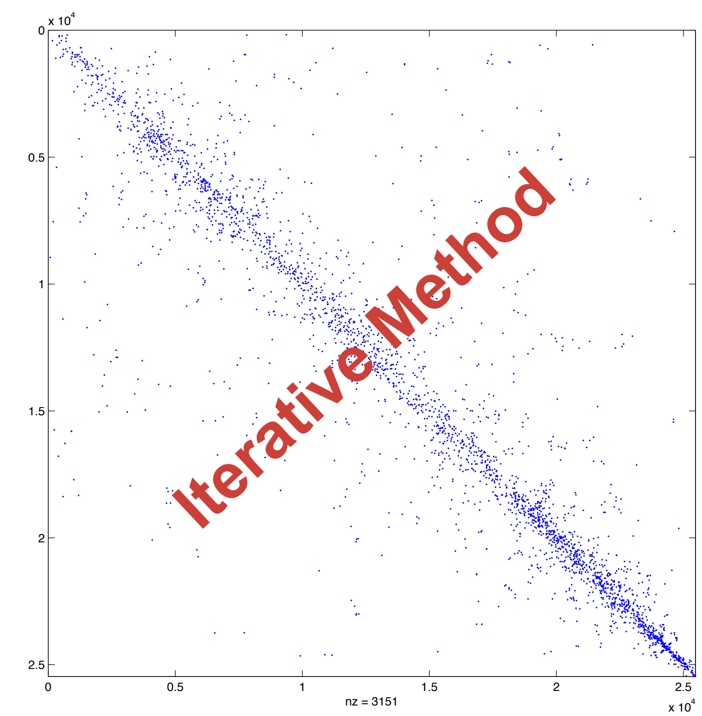


Illustration with Real Data : SFM, by Yong Dian Jian

- Solution to a sparse subgraph
  - An approximated solution
  - Efficient to compute
- Solution to the entire graph
  - The optimal solution
  - Expensive to compute directly

Yong Dian Jian, now works for Kai at Baidu Research :-)

Yong-Dian Jian, Doru C. Balcan and Frank Dellaert
Generalized Subgraph Preconditioners for Large-Scale Bundle Adjustment
Proceedings of 13th International Conference on Computer Vision (ICCV), Barcelona, 2011

Notre Dame

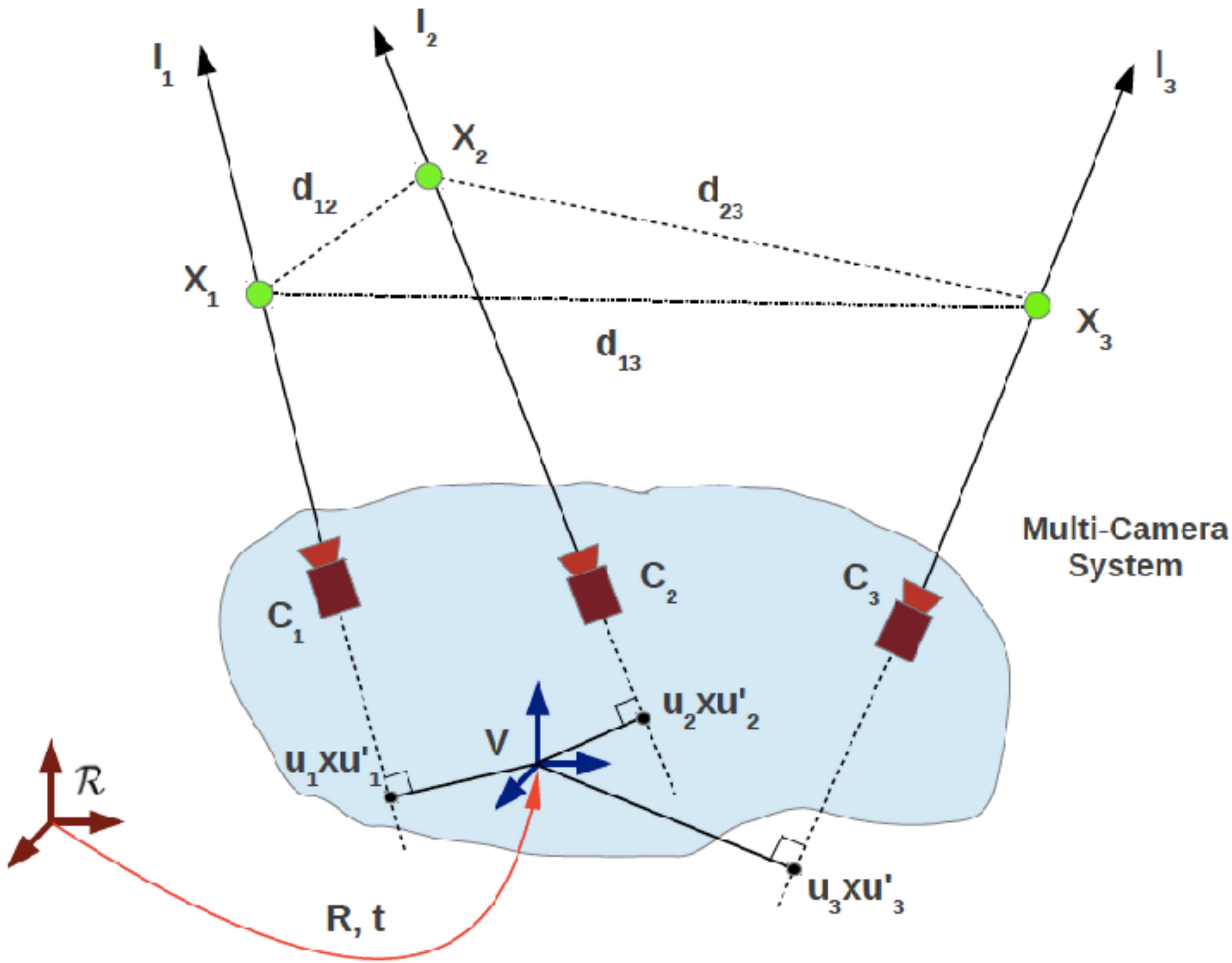
Solution from Subgraph

Solution from the Original Graph

Data from Noah Snavely

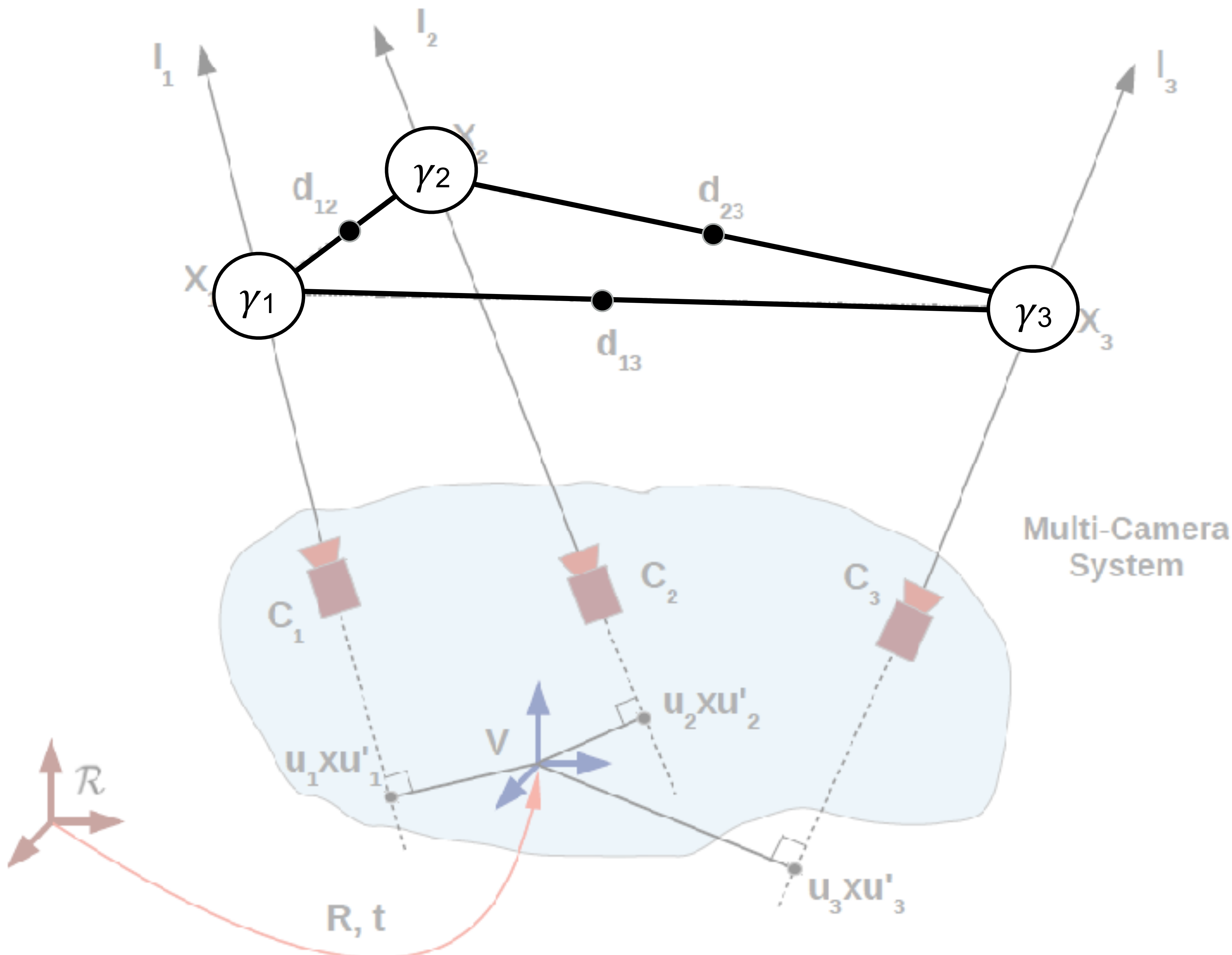
Polynomial Equations

Multi-camera pose estimation (Gim Hee Lee, ETH)

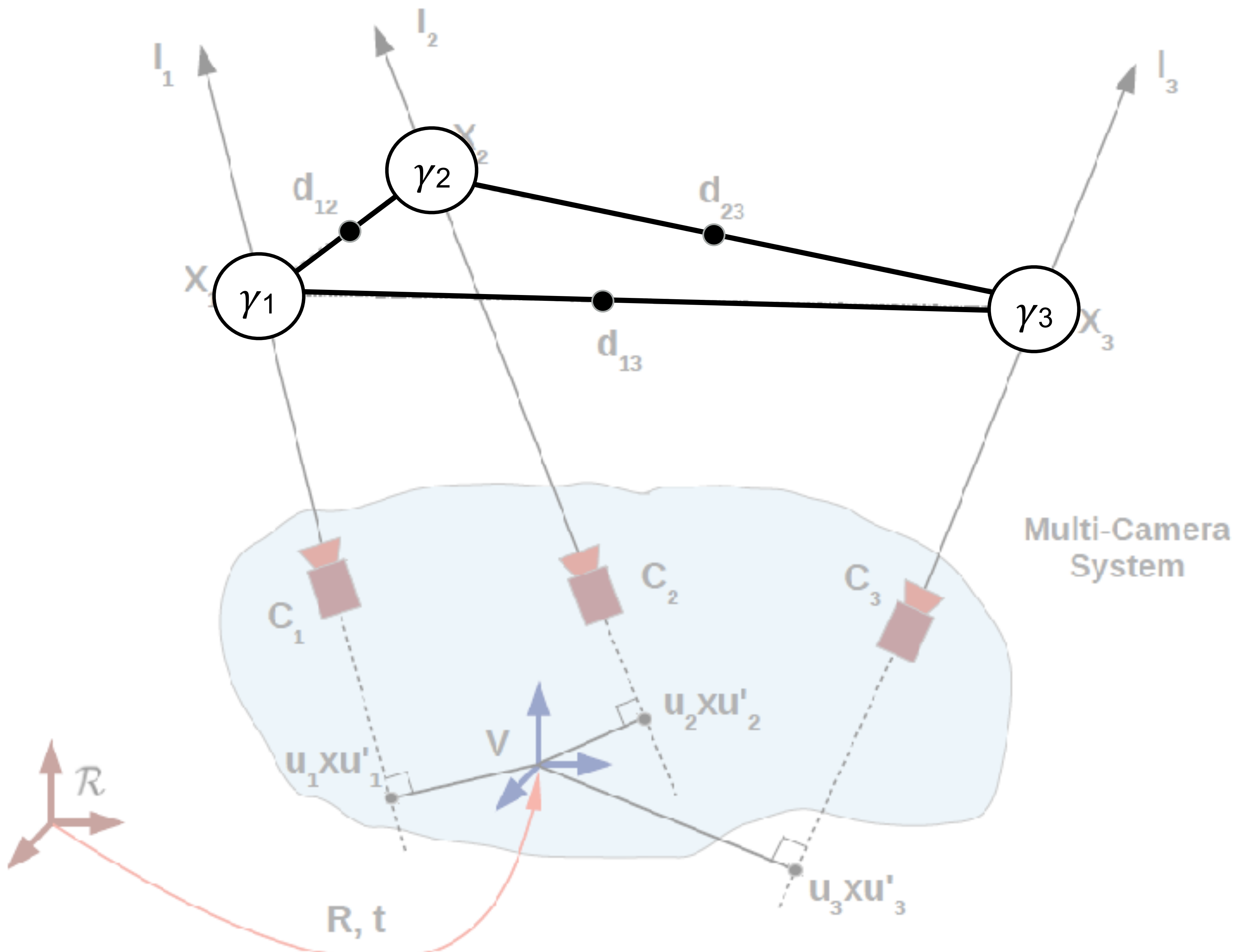


Polynomial Equations

Multi-camera pose estimation (Gim Hee Lee, ETH)

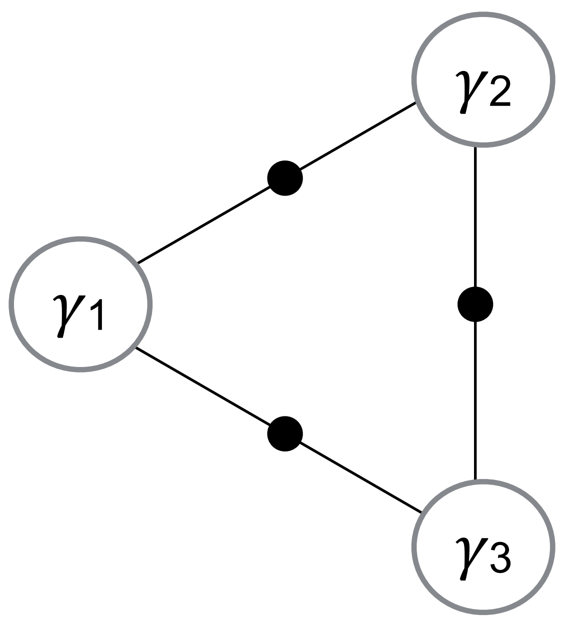


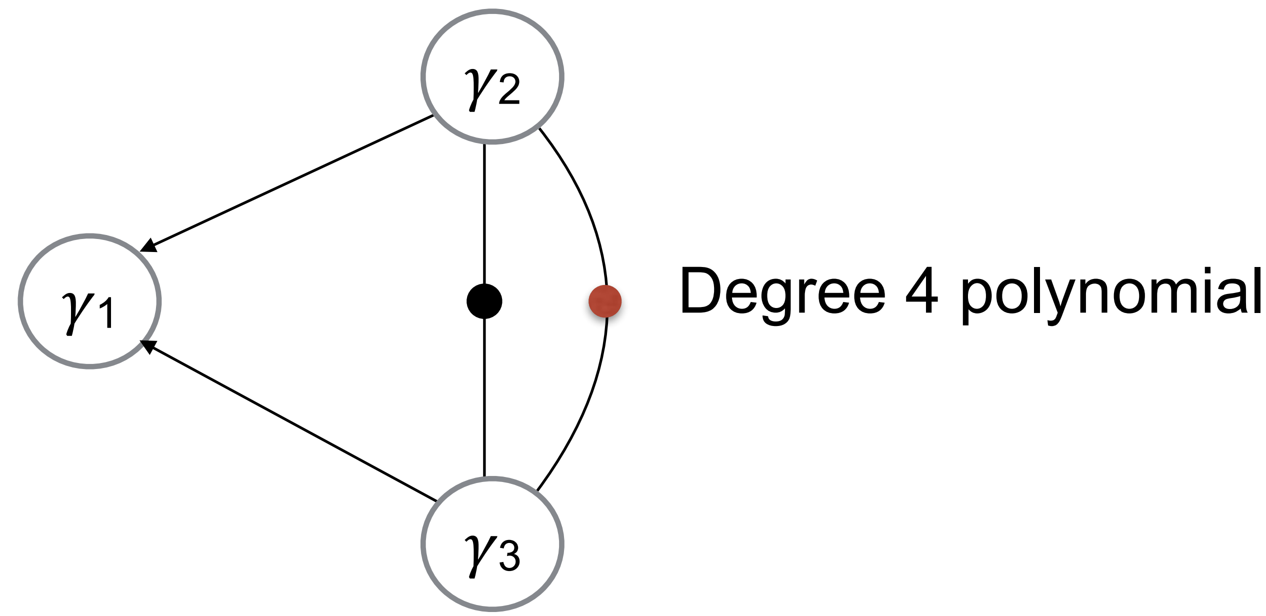
Gim Hee Lee's thesis: multi-camera pose estimation

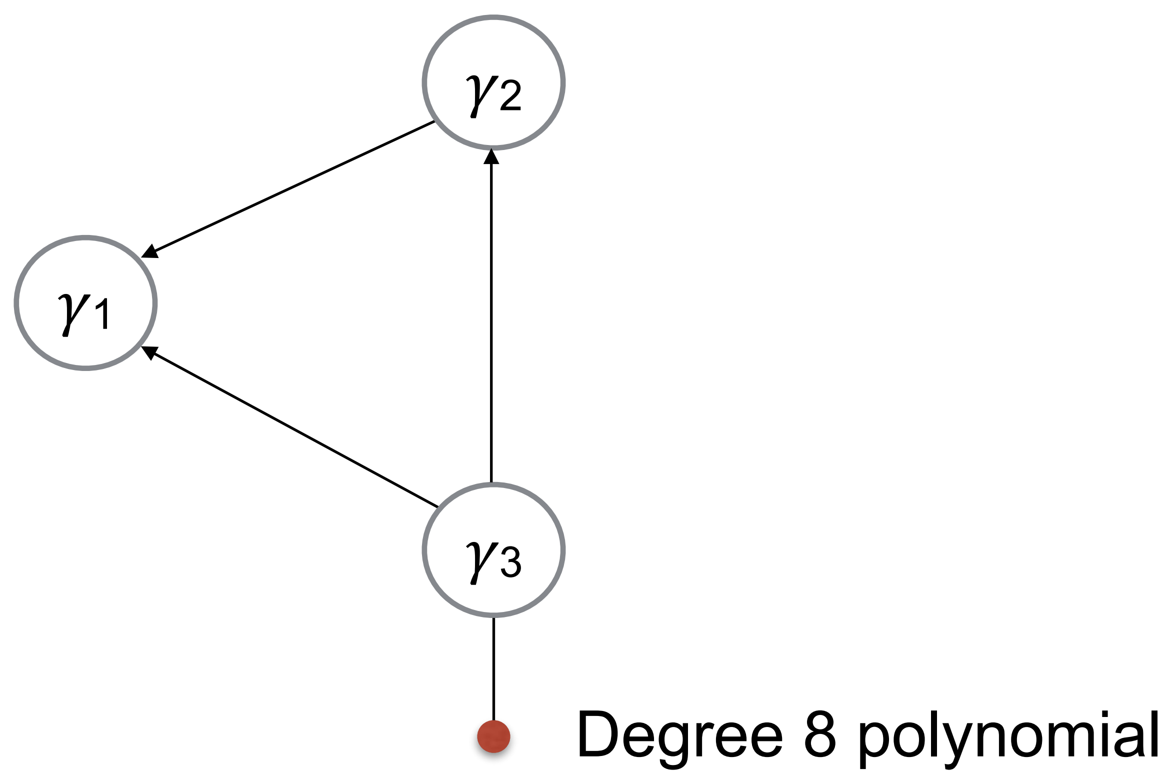


Polynomial Equations: equivalent to Gröbner bases?

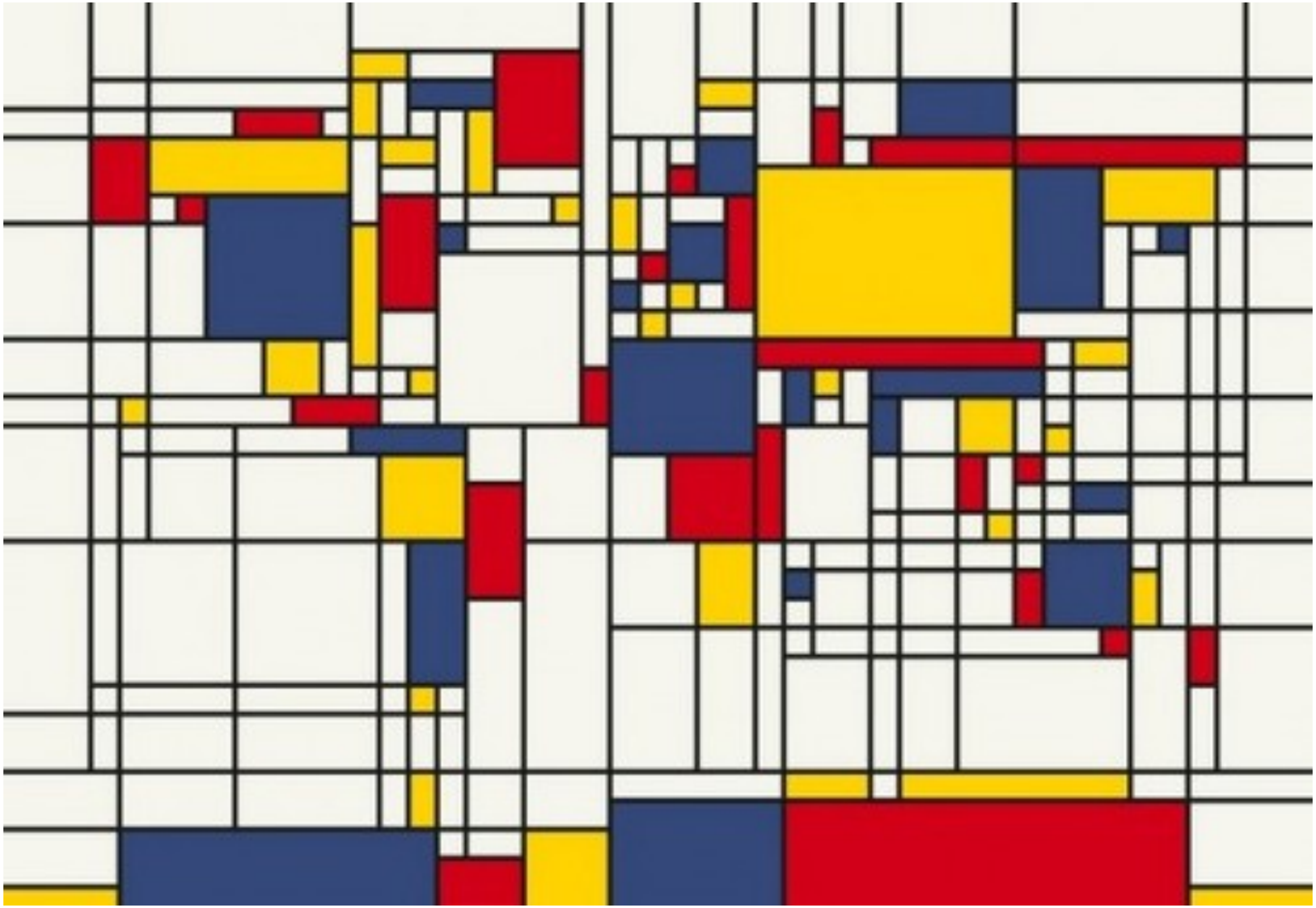
Three degree 2 polynomials, each in only 2 variables



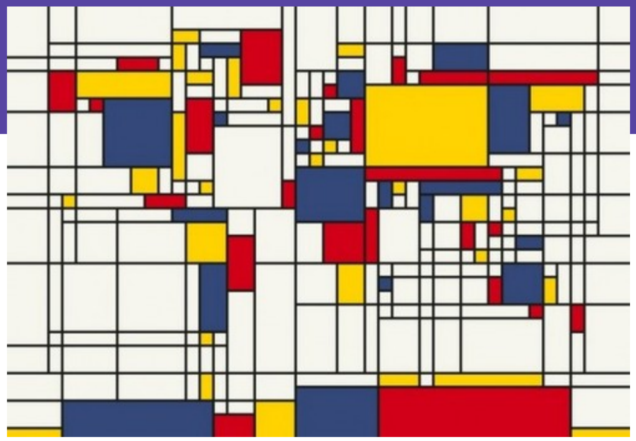




If we lived in a Linear World:



In a Linear World…



- Linear measurement function:

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

- …and objective function:

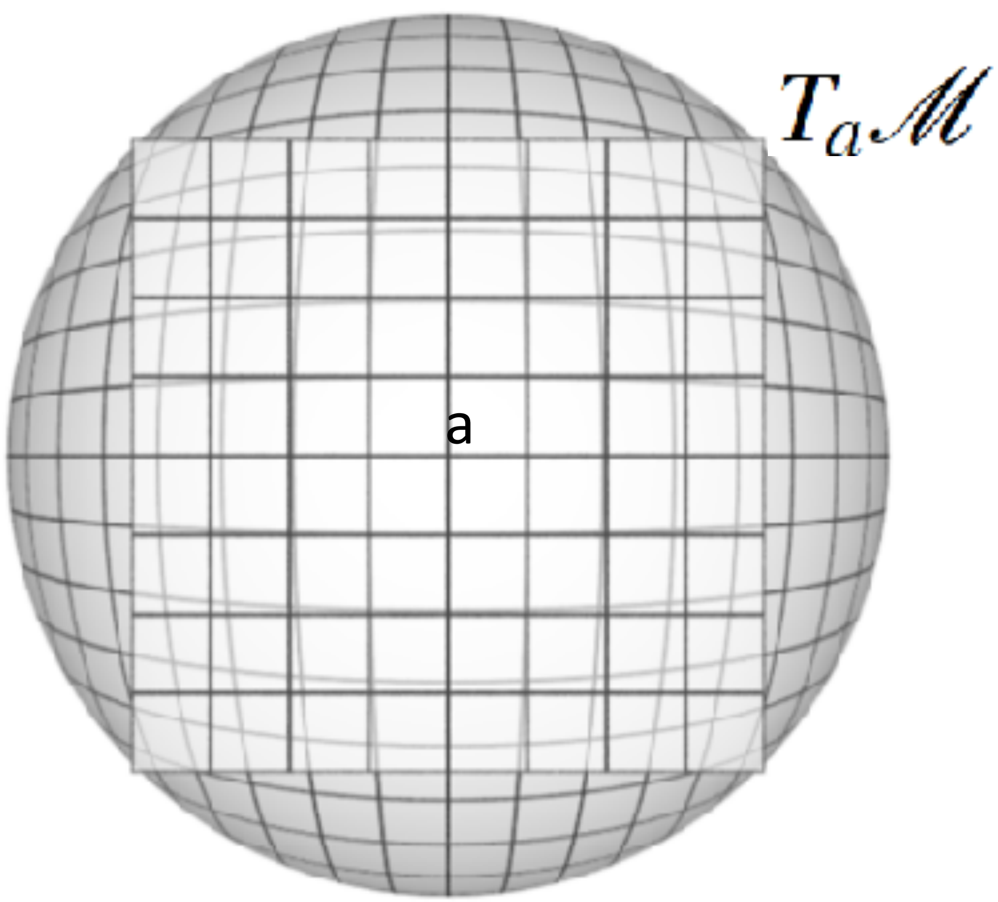
$$E\left(\xi_1, \xi_2, \{\delta_j\}\right) \triangleq \sum_{i=1,2}\sum_j \|F_{ij}\xi_i + G_{ij}\delta_j - b_{ij}\|^2_\Sigma$$

- Linear least-squares !

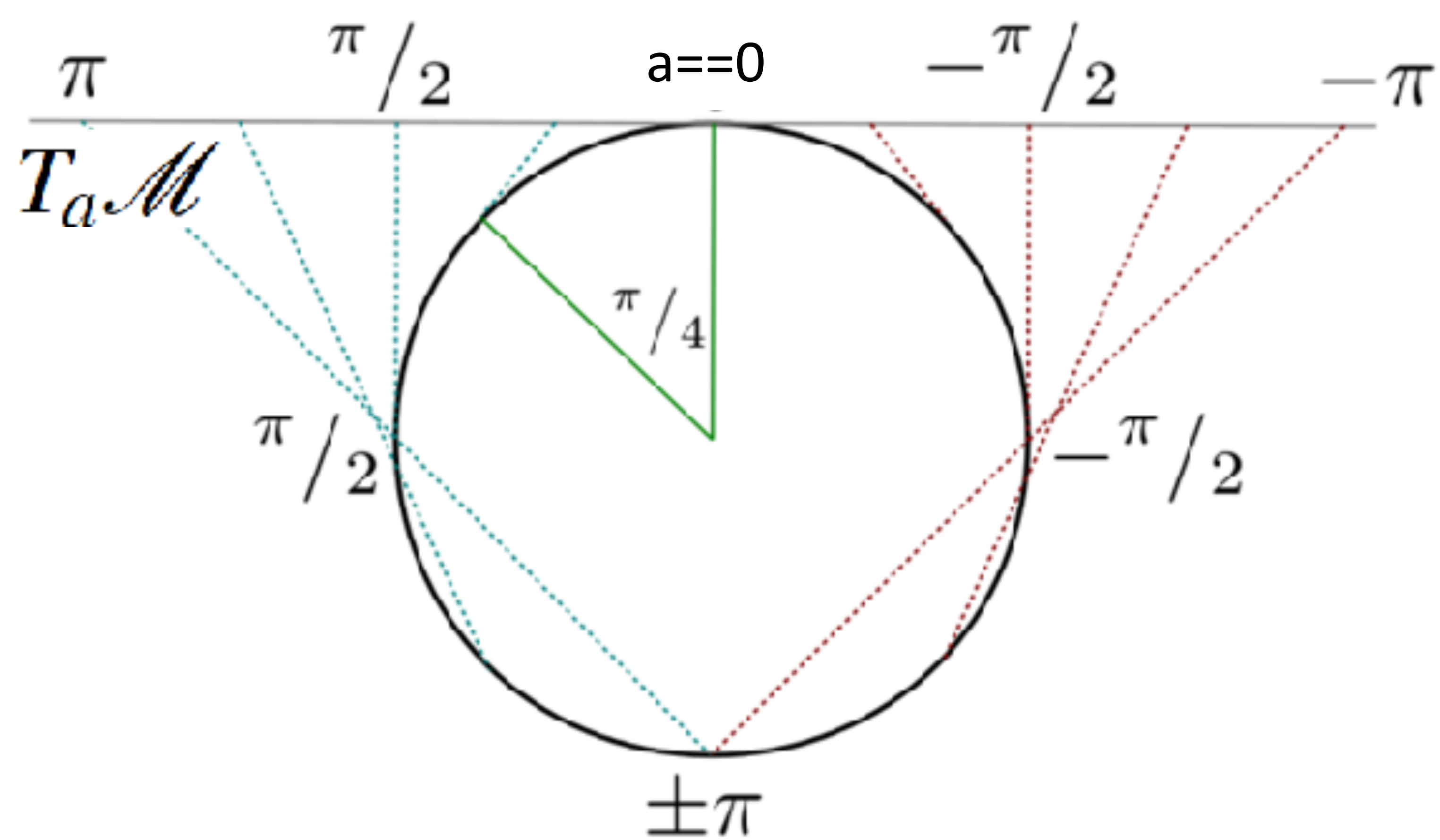
- Note: ξ_i = 6D, δ_j = 3D

Tangent Spaces

- An incremental change on a manifold $\mathcal{M}$ can be introduced via the notion of an n-dimensional tangent space $T_a\mathcal{M}$ at point **a**
- Example: sphere SO(2)



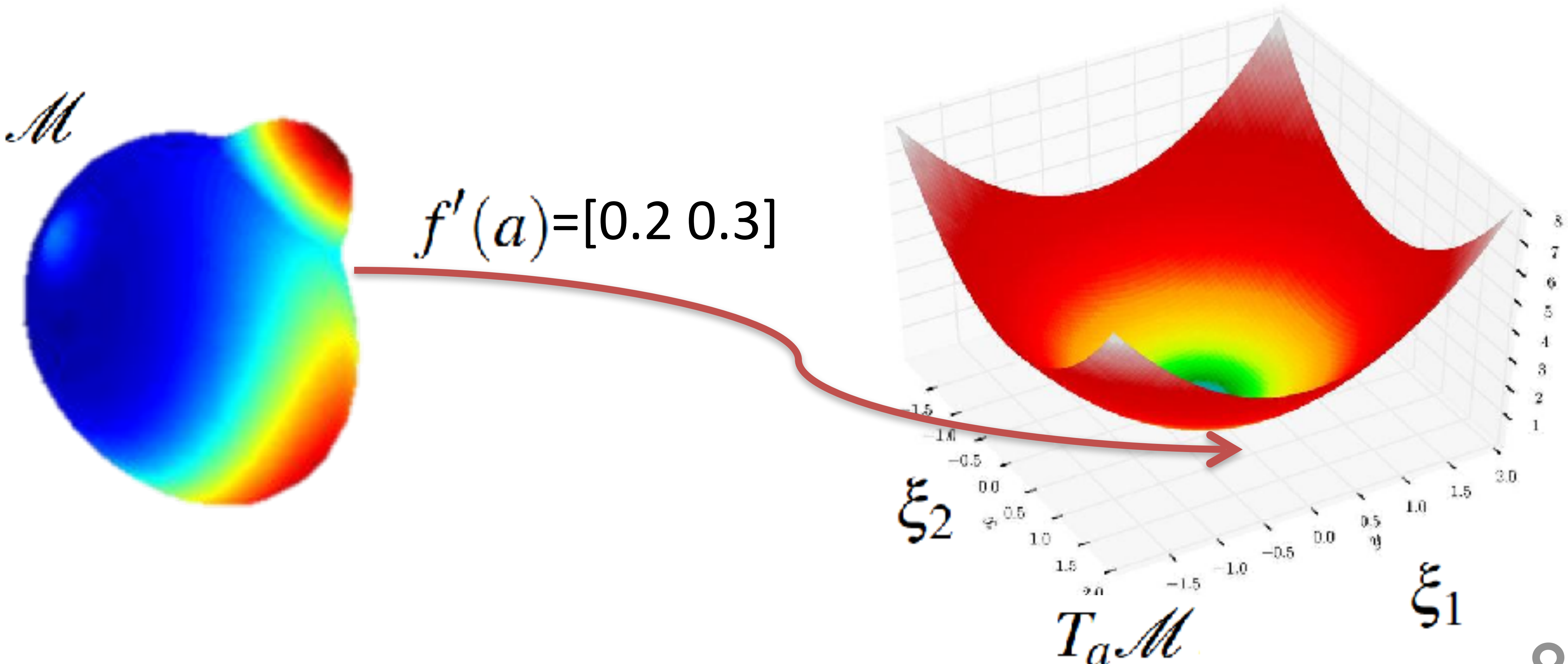
Thanks to Steven Lovegrove



Generalized Taylor Expansion

- Define f’(a) to satisfy:

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



Gauss-Newton

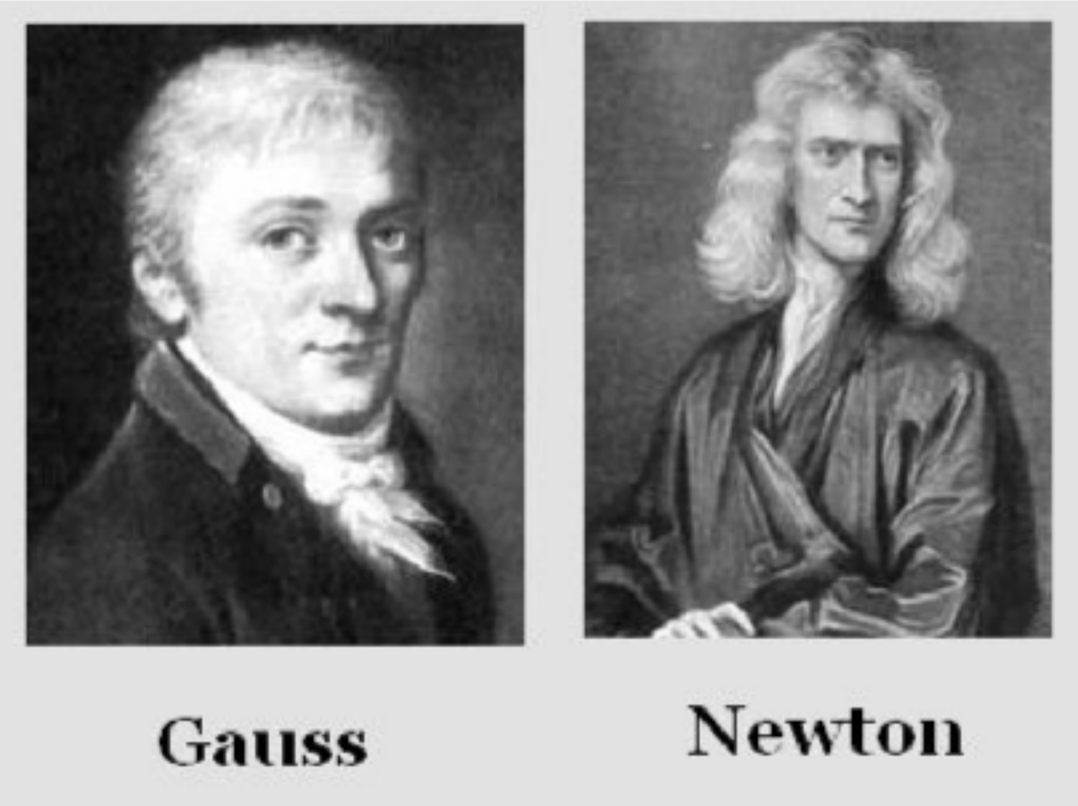
Iteratively Linearize, solve normal equations on tangent space, update Lie group elements

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

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

   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.



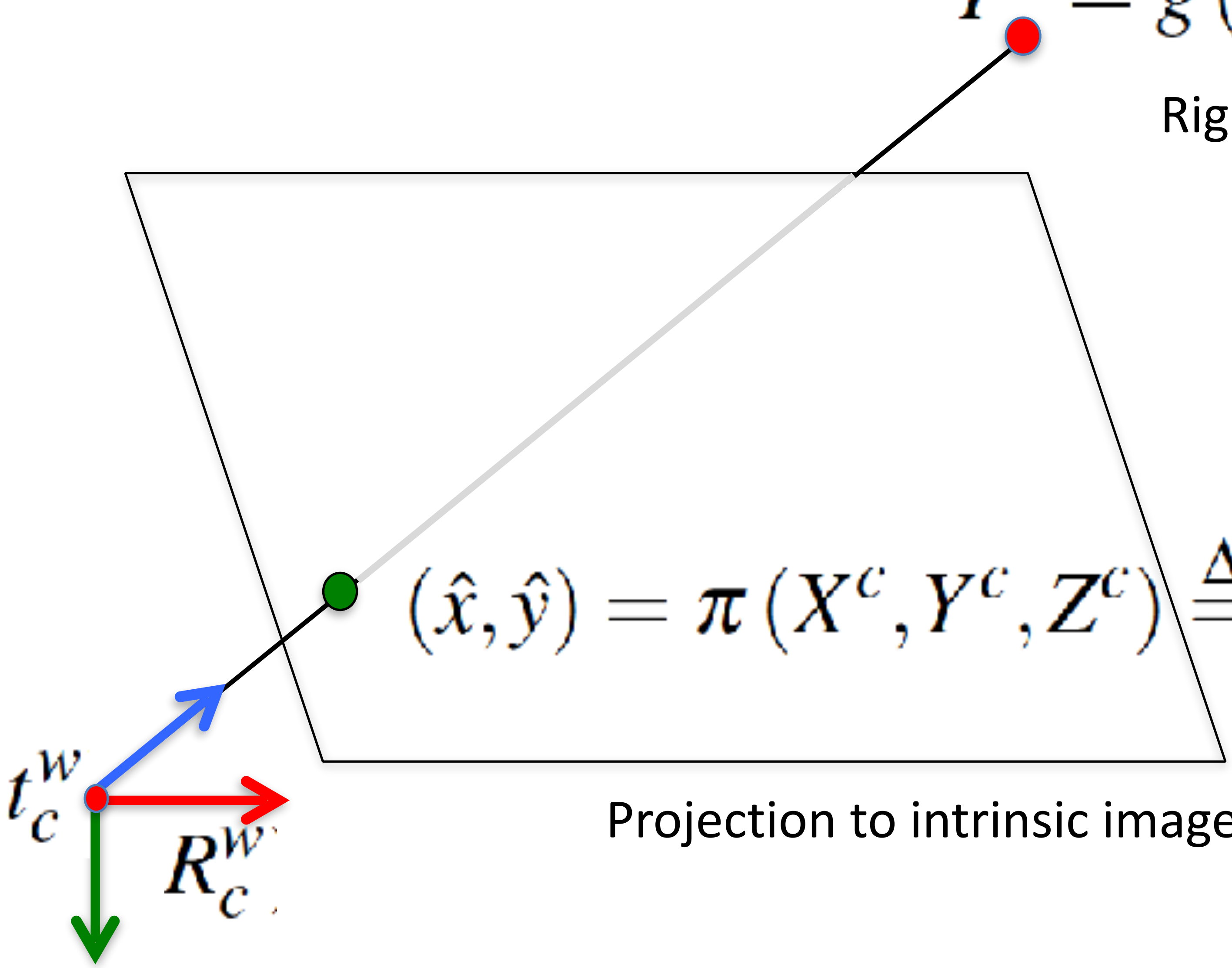
Generative Model

- Measurement Function, calibrated setting!

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

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

Rigid 3D transform to camera frame

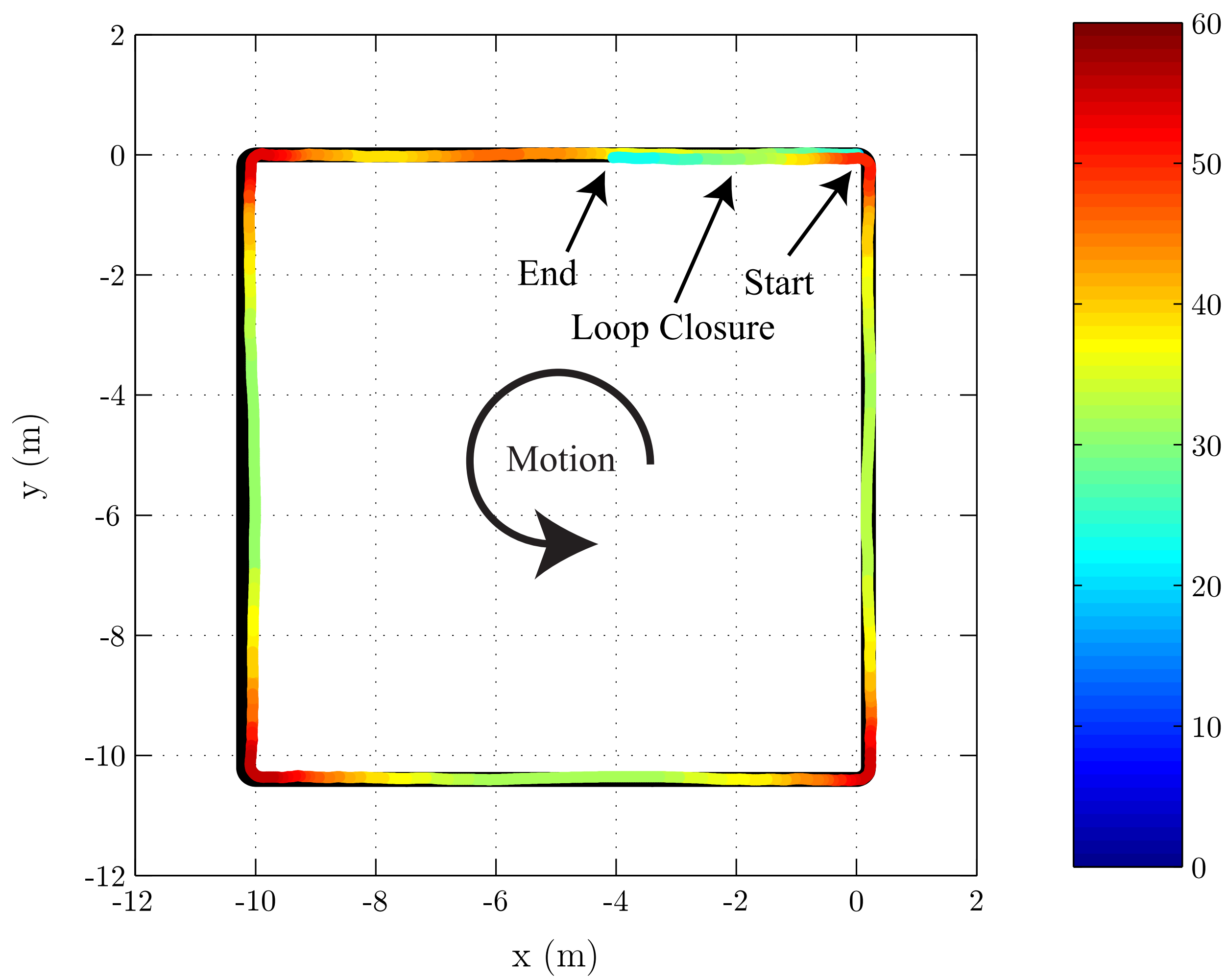


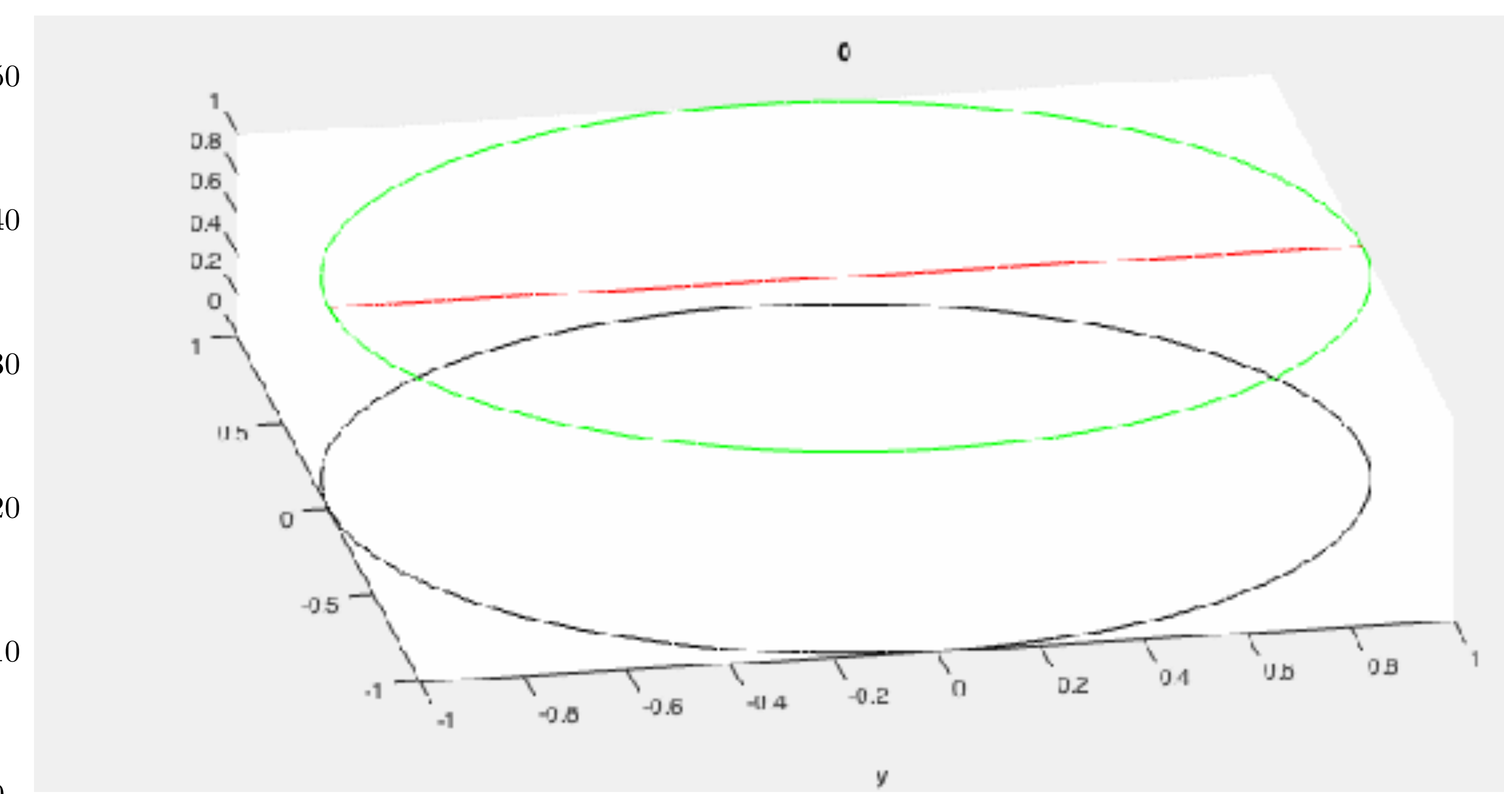
$$(\hat{x}, \hat{y}) = \pi(X^c, Y^c, Z^c) \triangleq \boxed{\left(\frac{X^c}{Z^c}, \frac{Y^c}{Z^c}\right)}$$

Projection to intrinsic image coordinates

GTSAM 4 in Development!

- Join us on BitBucket!
- Collaboration with ETHZ
- Full reverse AD framework via **Expressions**
- Motivation: continuous-time estimation

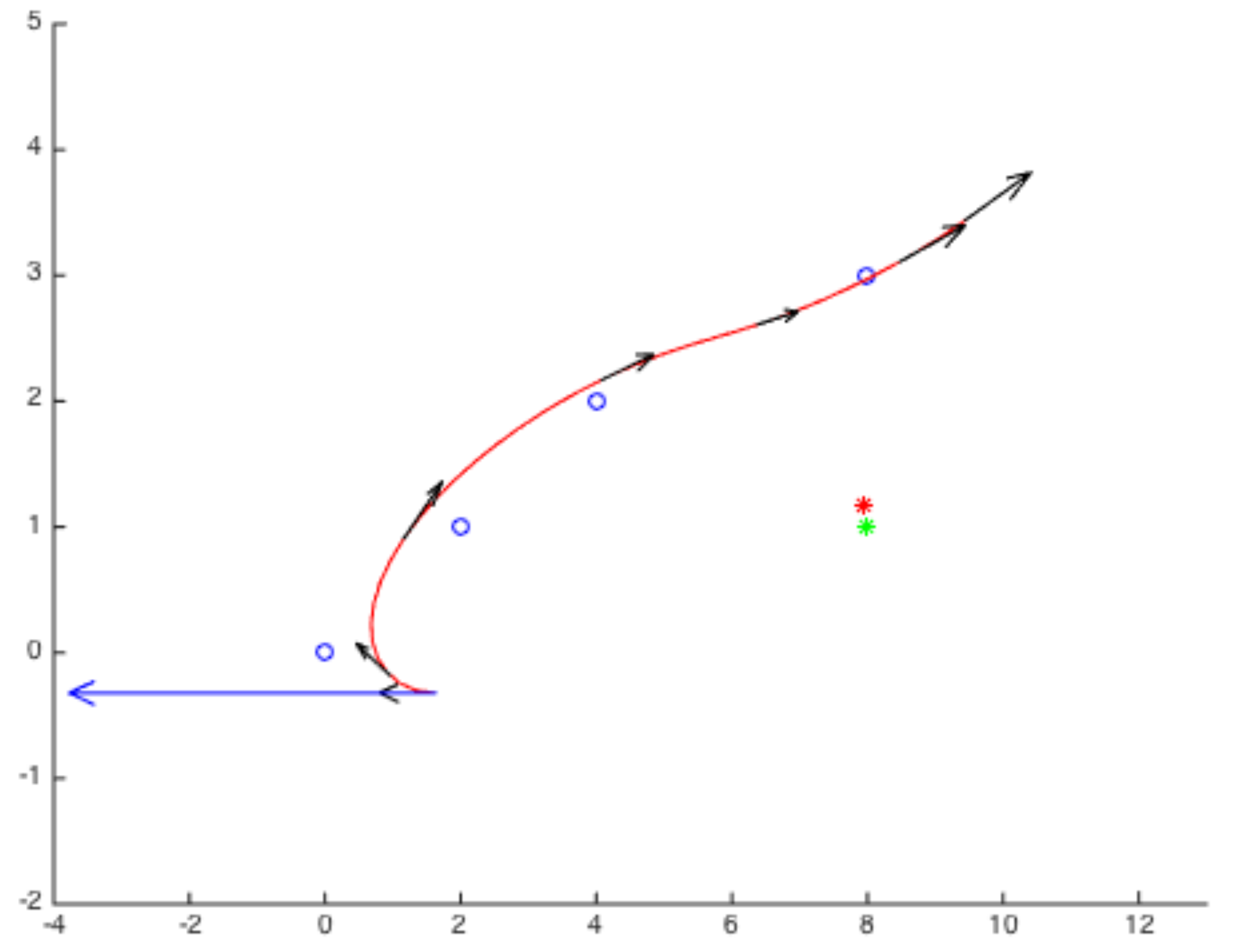




Use PSOC Ideas for SLAM

- Dynamic Defects

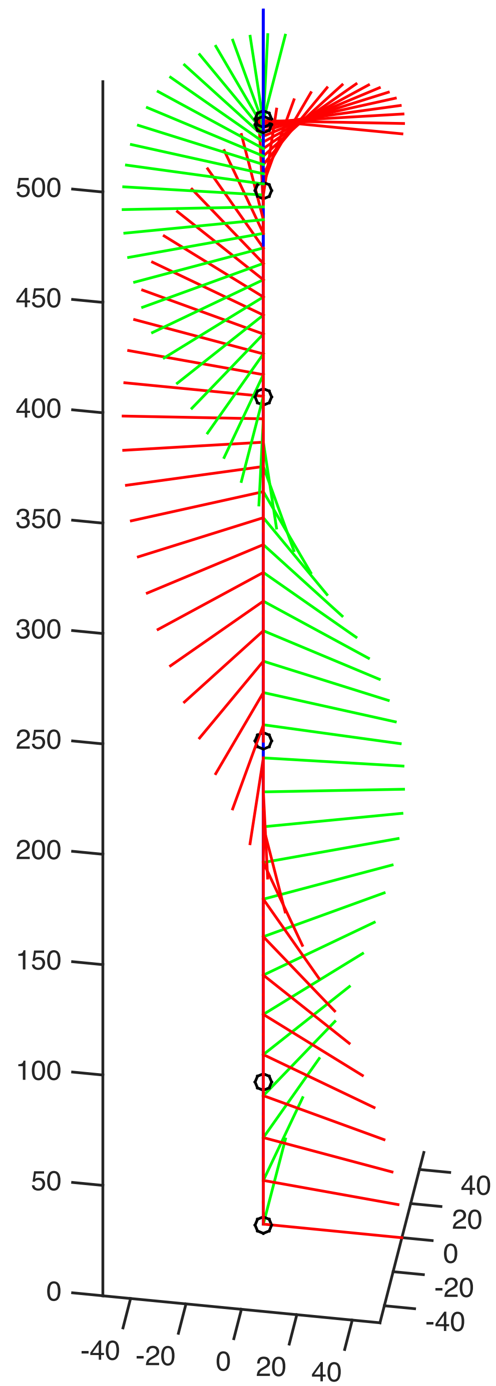
$$\begin{aligned}\dot{x}(t) &= v(t)\cos\theta(t)\\ \dot{y}(t) &= v(t)\sin\theta(t)\\ \dot{\theta}(t) &= \omega(t)\end{aligned}$$

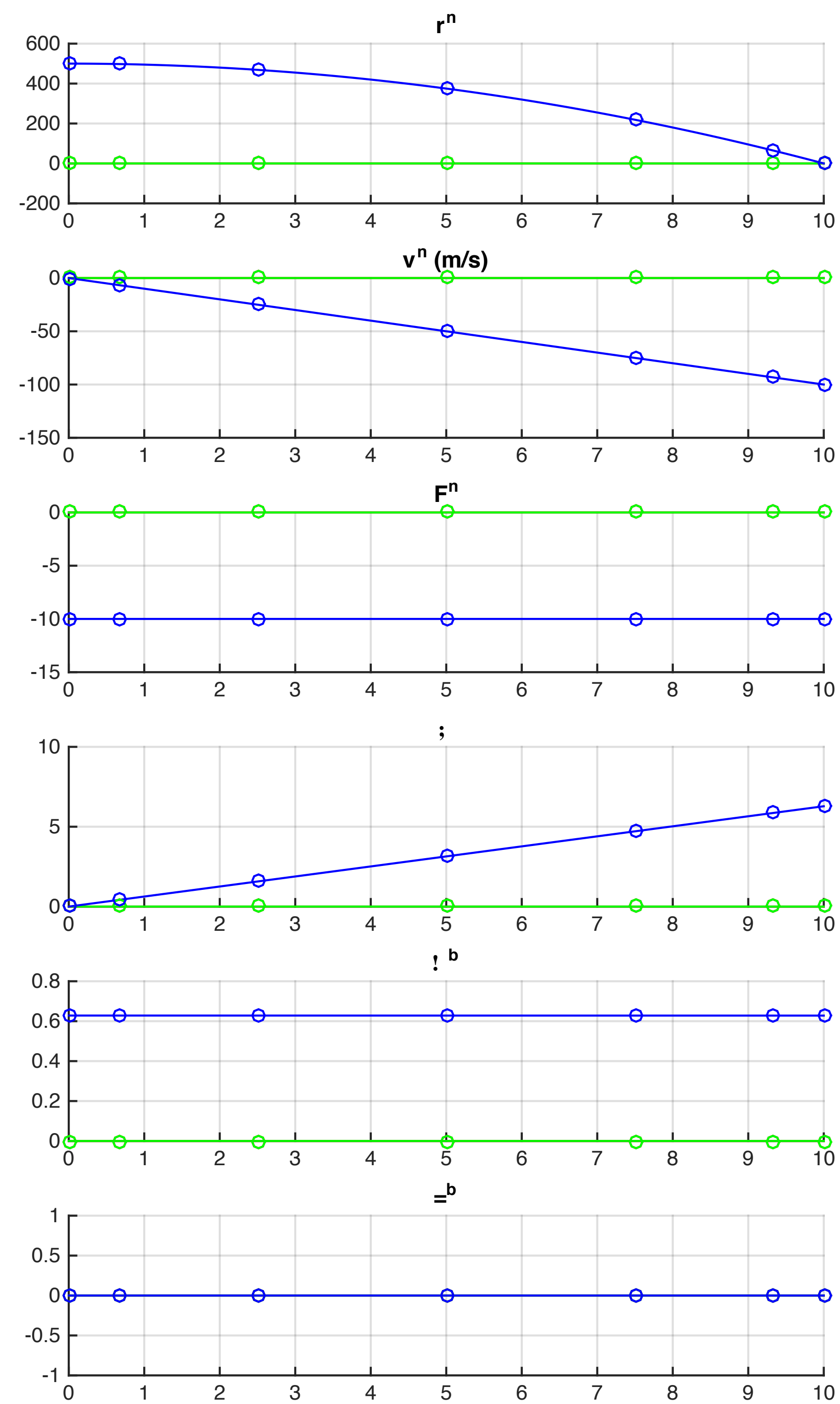


- Velocity constraint at t=0
- 4 “GPS” measurements
- 1 landmark

Rigid 3D Body Dynamics

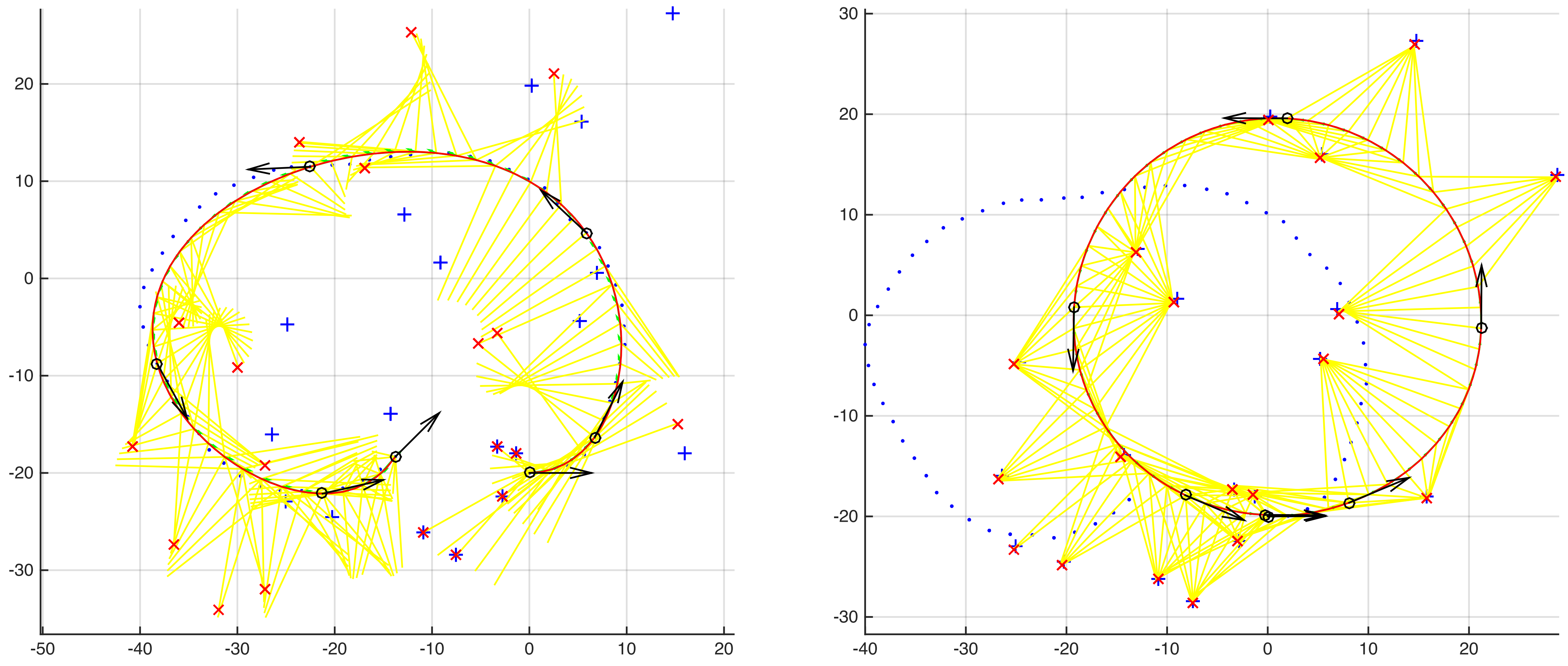
- Use canonical coordinates $\mathbf{R} = \exp \hat{\rho}$



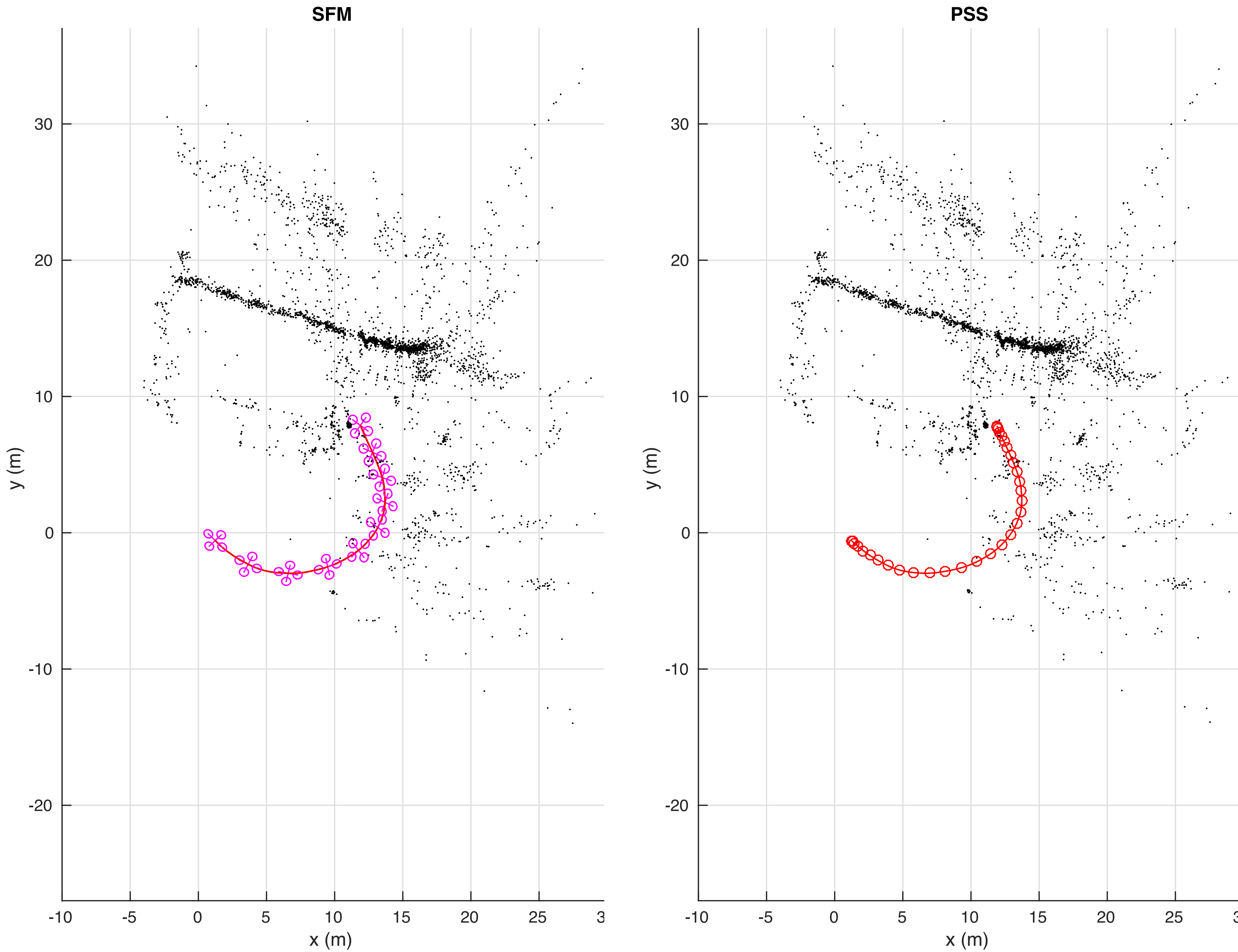


Pseudospectral SLAM

- Perfect circle, 60 poses, 20 landmarks



SFM vs PSS



GTSAM

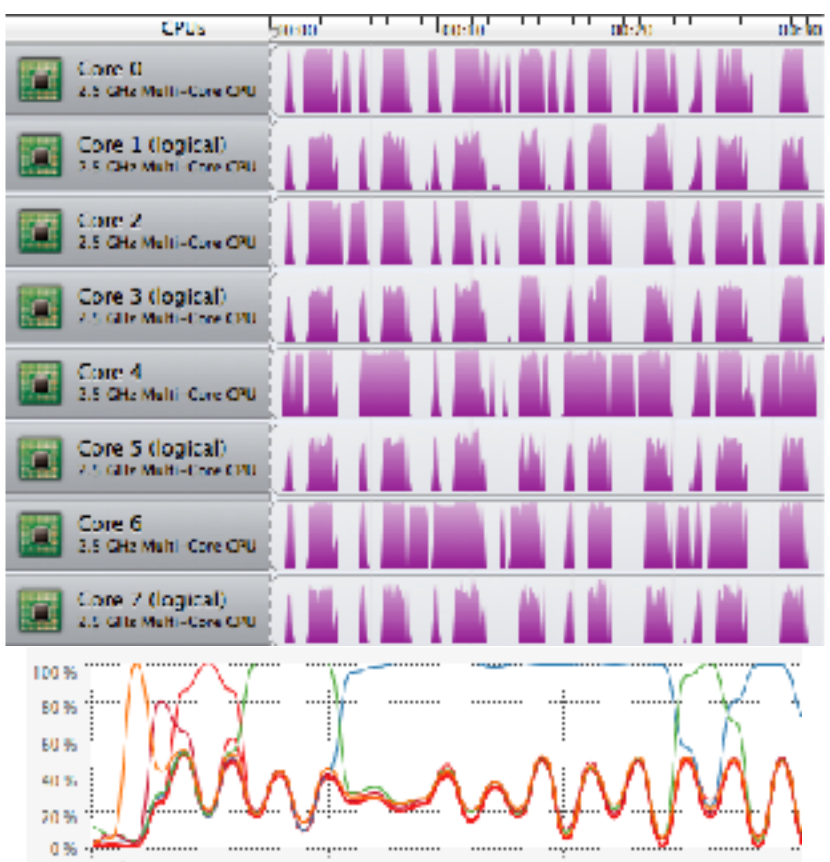
Georgia **T**ech **S**moothing **A**nd **M**apping

Download: tinyurl.com/gtsam

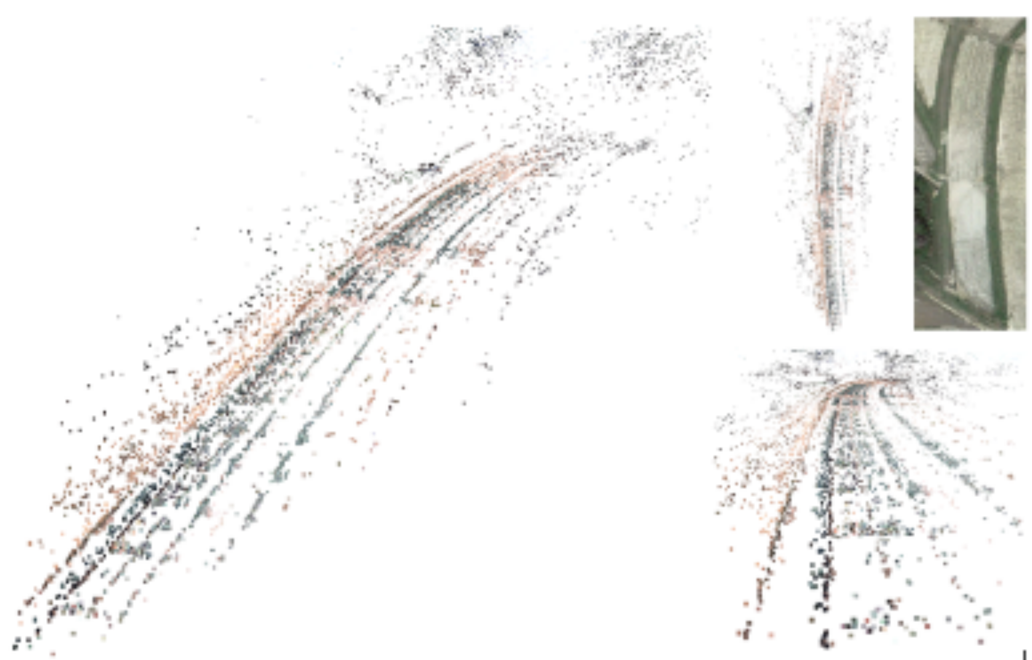
Includes numerous features:

- iSAM and iSAM2
- Multi-threading with TBB (funded by DARPA)
- Smart Projection Factors for SfM
- LAGO Initialization for planar SLAM (Luca Carlone et. al)

Multi-threading

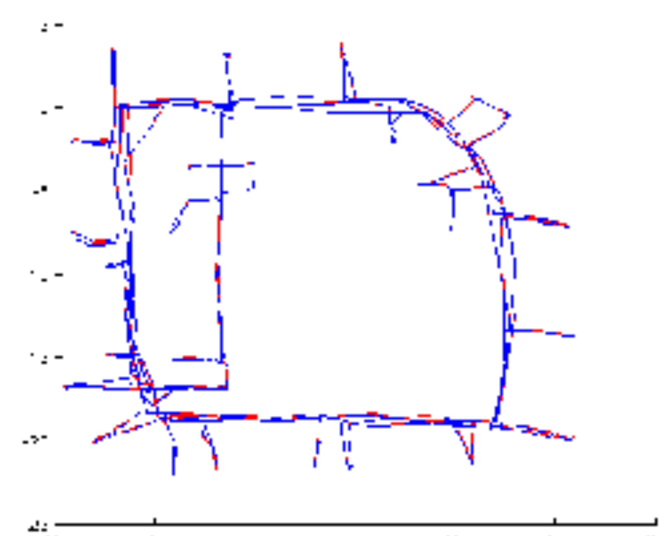


Smart Factors

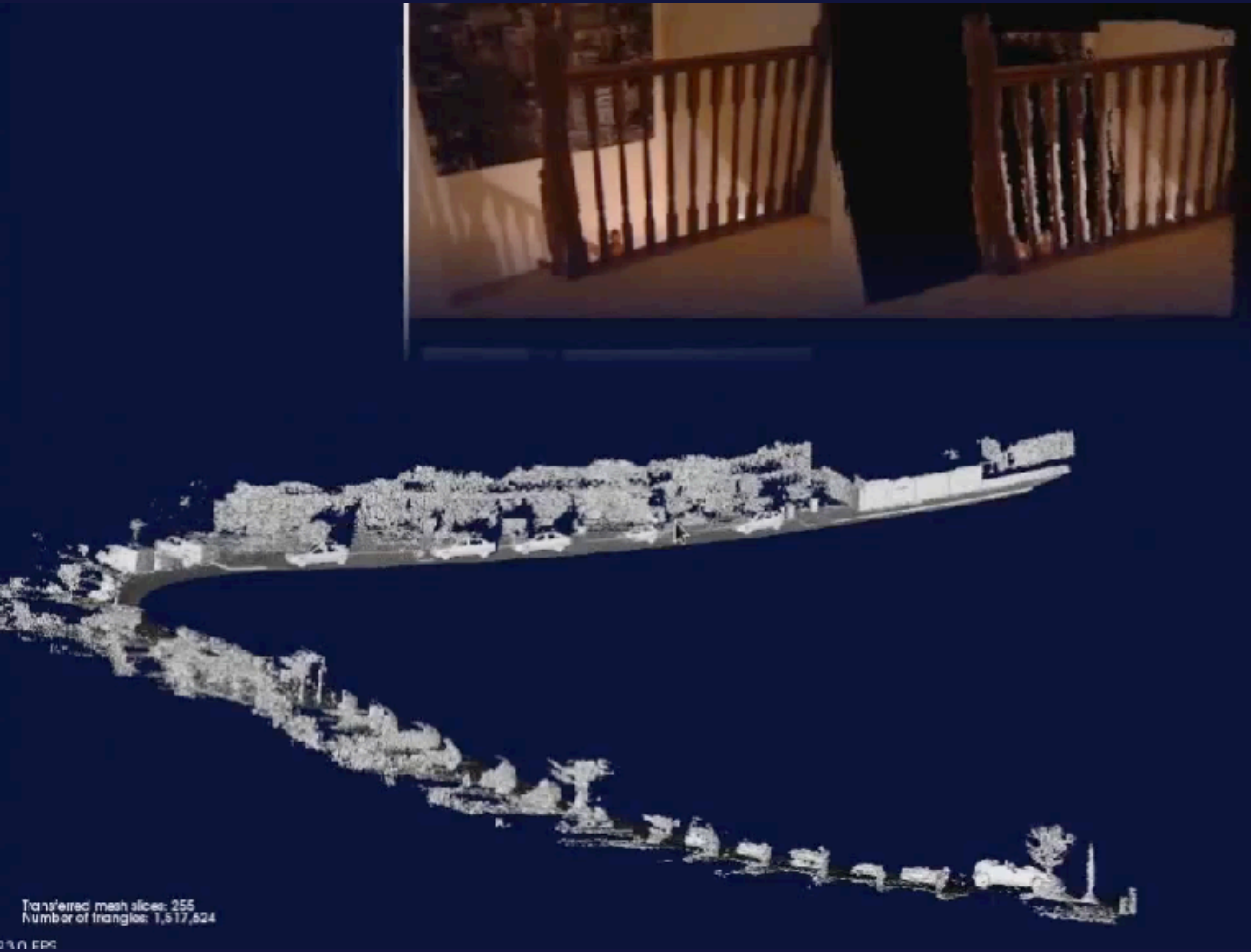


Lago

Linear Approximation for Graph Optimization



Kintinuous (Whelan et al. @ MIT) = KinectFusion + iSAM



Movie by Whelan et al.


SKYDIO