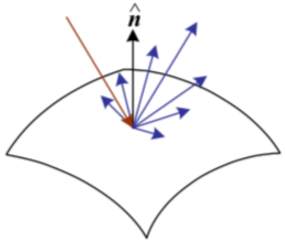
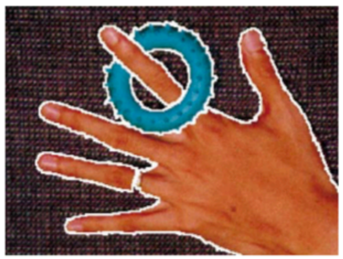


2. Image Formation

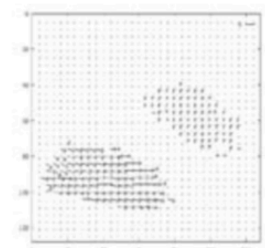
3. Image Processing

4. Features

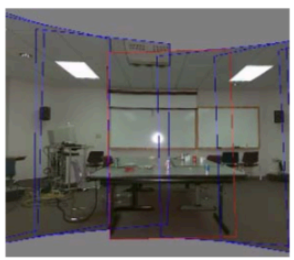


5. Segmentation

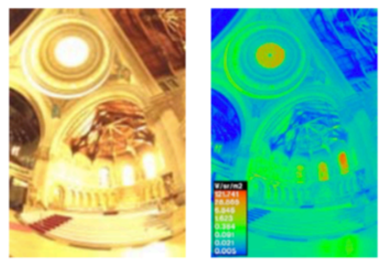
6-7. Structure from Motion



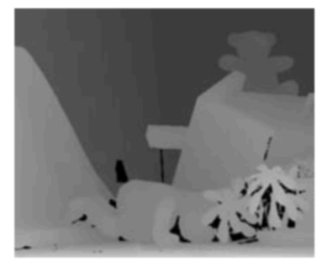
8. Motion



9. Stitching

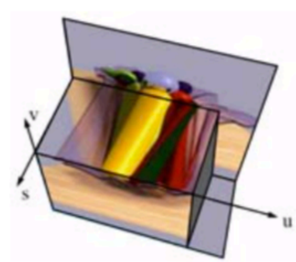


10. Computational Photography



11. Stereo

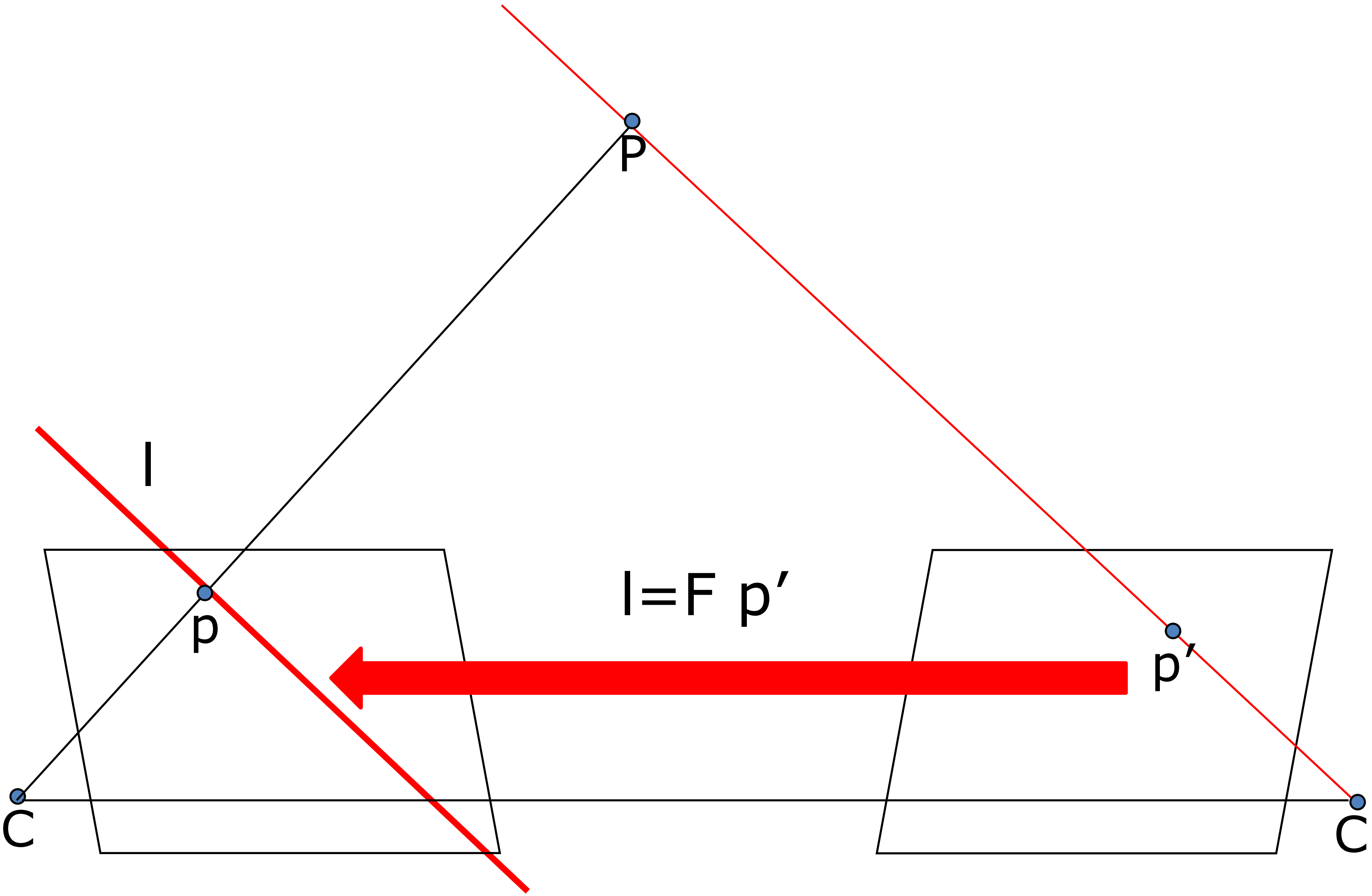
12. 3D Shape



13. Image-based Rendering

14. Recognition

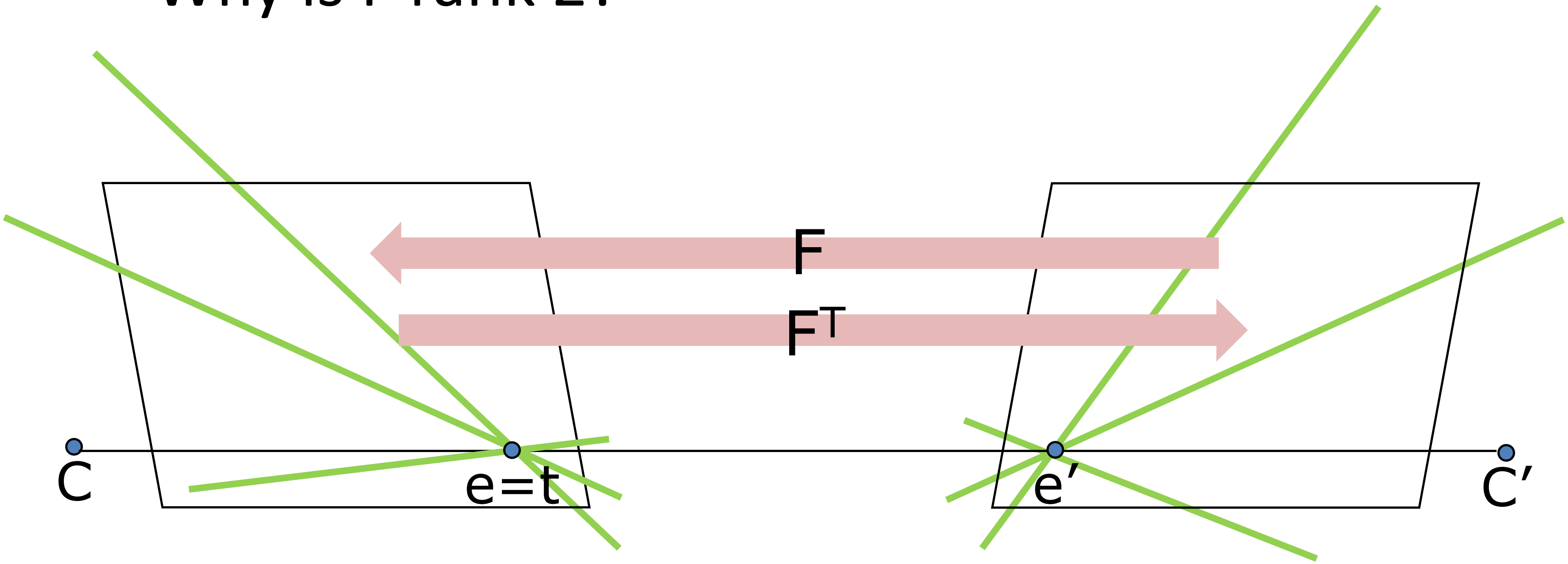
Recap: Two views and Fundamental Matrix F



Frank Dellaert Fall 2019

Rank 2 Constraint

- Why is F rank 2?



- Not invertible! Collection of points is mapped to a pencil of lines. Epipoles map to zero.
- What would it mean to be rank 1?

The Eight-Point Algorithm (Longuet-Higgins, 1981)

$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

$$(uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

$$\begin{pmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' \\ u_2u_2' & u_2v_2' & u_2 & v_2u_2' & v_2v_2' & v_2 & u_2' & v_2' \\ u_3u_3' & u_3v_3' & u_3 & v_3u_3' & v_3v_3' & v_3 & u_3' & v_3' \\ u_4u_4' & u_4v_4' & u_4 & v_4u_4' & v_4v_4' & v_4 & u_4' & v_4' \\ u_5u_5' & u_5v_5' & u_5 & v_5u_5' & v_5v_5' & v_5 & u_5' & v_5' \\ u_6u_6' & u_6v_6' & u_6 & v_6u_6' & v_6v_6' & v_6 & u_6' & v_6' \\ u_7u_7' & u_7v_7' & u_7 & v_7u_7' & v_7v_7' & v_7 & u_7' & v_7' \\ u_8u_8' & u_8v_8' & u_8 & v_8u_8' & v_8v_8' & v_8 & u_8' & v_8' \end{pmatrix}\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = -\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Minimize:

$$\sum_{i=1}^{n} (\boldsymbol{p}_i^T \mathcal{F} \boldsymbol{p}_i')^2$$

under the constraint

$|\mathcal{F}|^2 = 1.$

The Normalized Eight-Point Algorithm (Hartley, 1995)

- Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels:

$$q_i = T\,p_i \qquad q_i' = T'\,p_i'.$$

- Use the eight-point algorithm to compute $\mathcal{F}$ from the points q_i and q'_i.

- Enforce the rank-2 constraint.

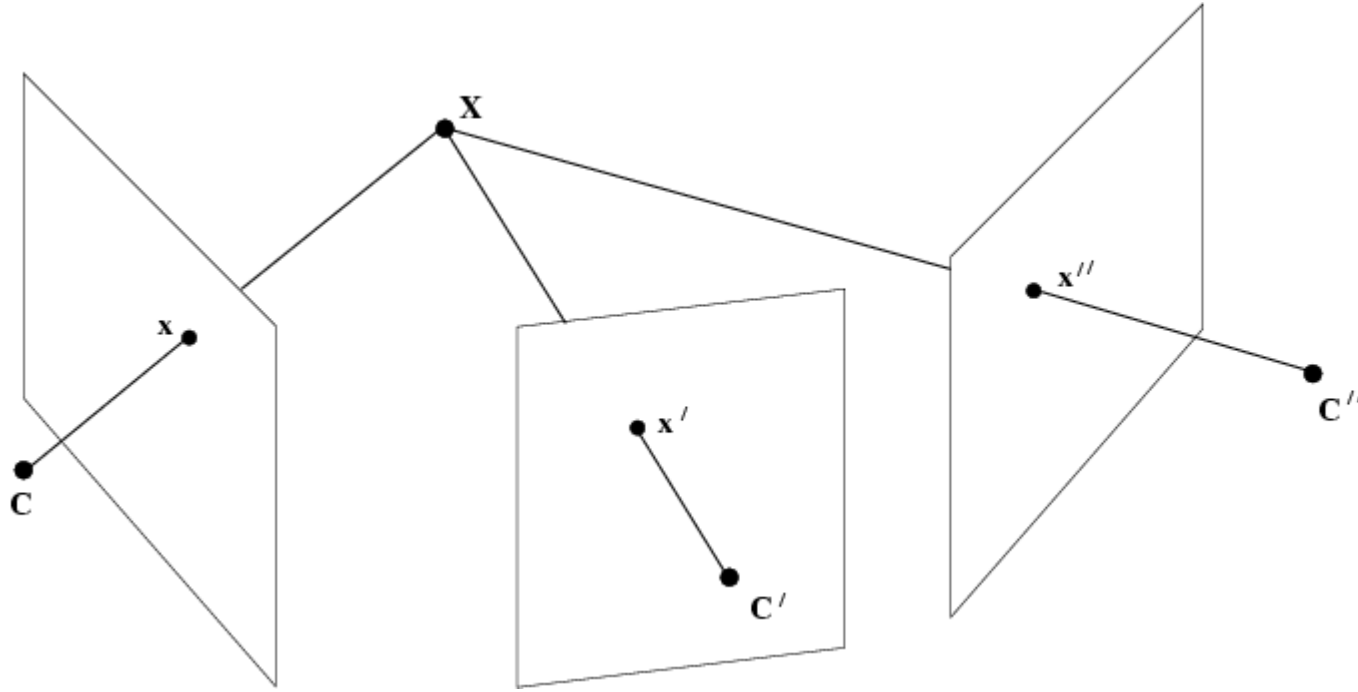
- Output $T^{-1}\mathcal{F}\,T'$.

Trinocular Camera rigs

Navigation Camera System

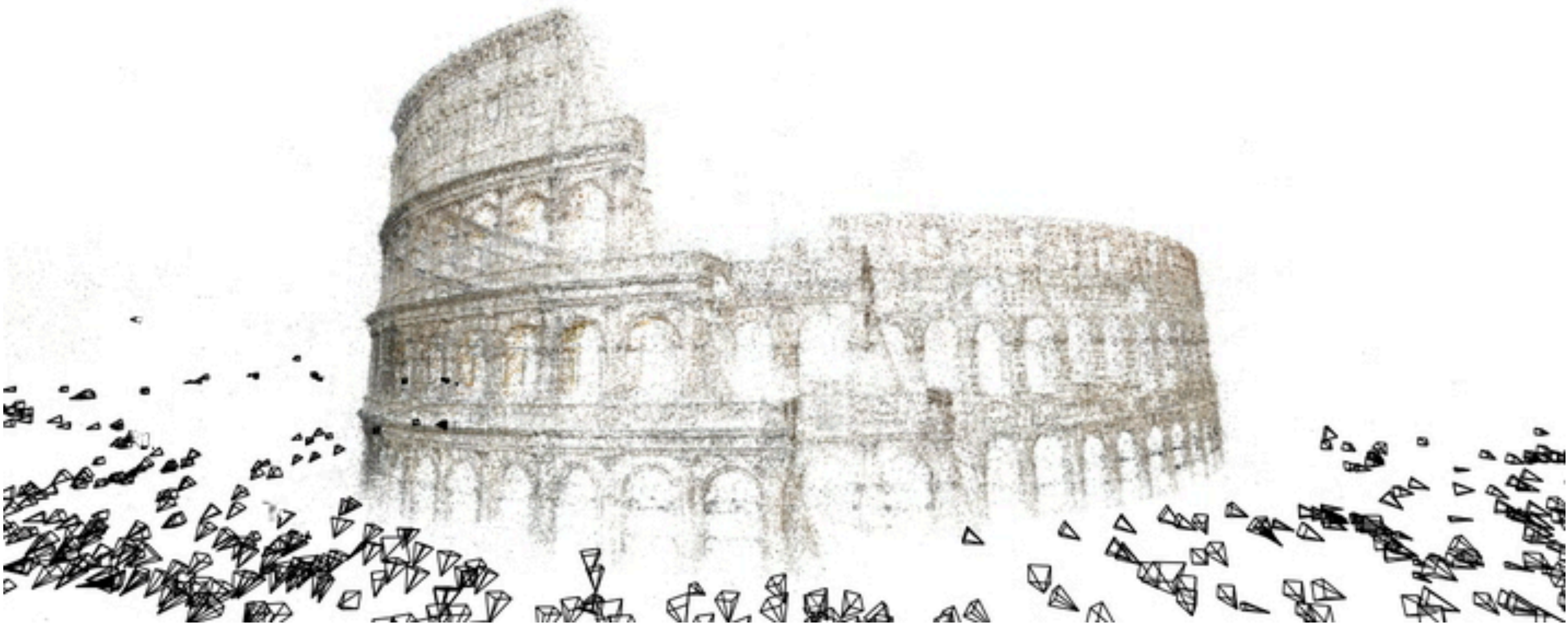
CONFIGURATION	6x cameras in trinocular configuration top and bottom
SENSOR TYPE	Sony 1/3" 4K color CMOS
LENS APERTURE	f/1.8
FIELD-OF-VIEW	200°
ENVIRONMENT COVERAGE	True 360°

Trifocal Geometry



$$[\mathbf{x}']_\times \left(\sum_i x^i \mathrm{T}_i\right)[\mathbf{x}'']_\times = 0_{3\times 3}$$

Structure from Motion



Building Rome in a Day
Agarwal et al

Motivation

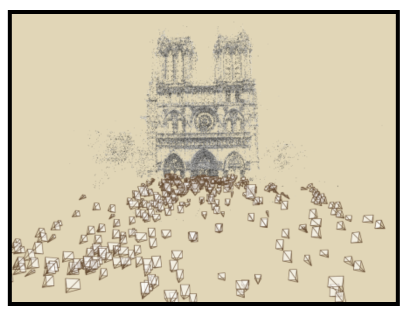
- Photo Tourism
- Photosynth
- Multi-view stereo
- Building Rome in a Day
- Rome on a Cloudless Day

Photo Tourism

Noah Snavely, Steven M. Seitz, Richard Szeliski, Photo tourism: Exploring photo collections in 3D," ACM Transactions on Graphics (SIGGRAPH Proceedings), 25(3), 2006, 835-846.

Input photographs

Scene reconstruction



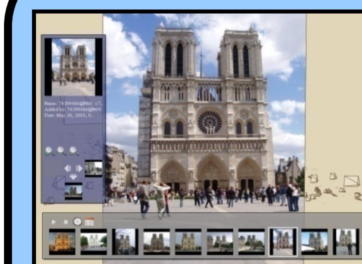
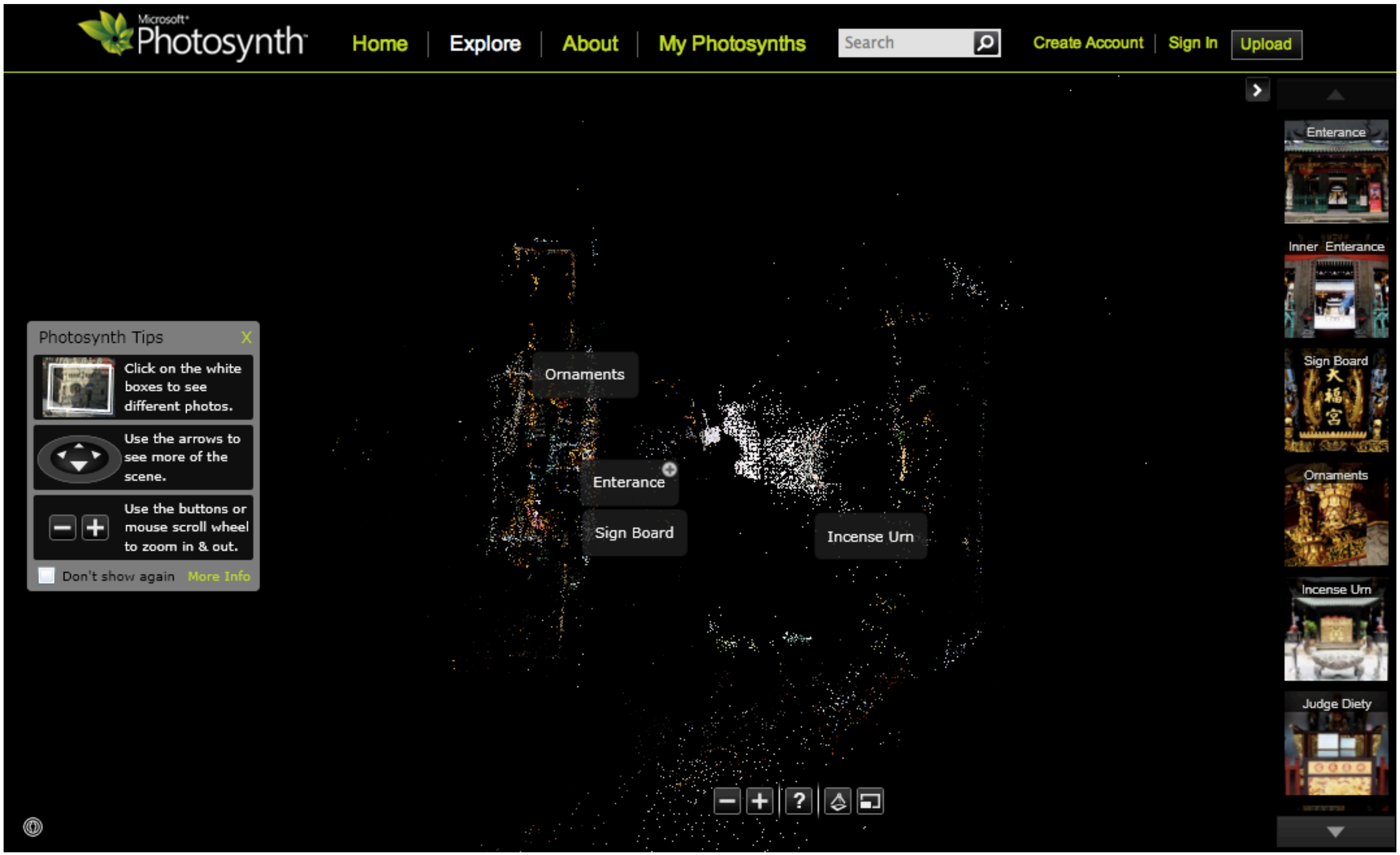


Photo Explorer

http://phototour.cs.washington.edu/

Photosynth

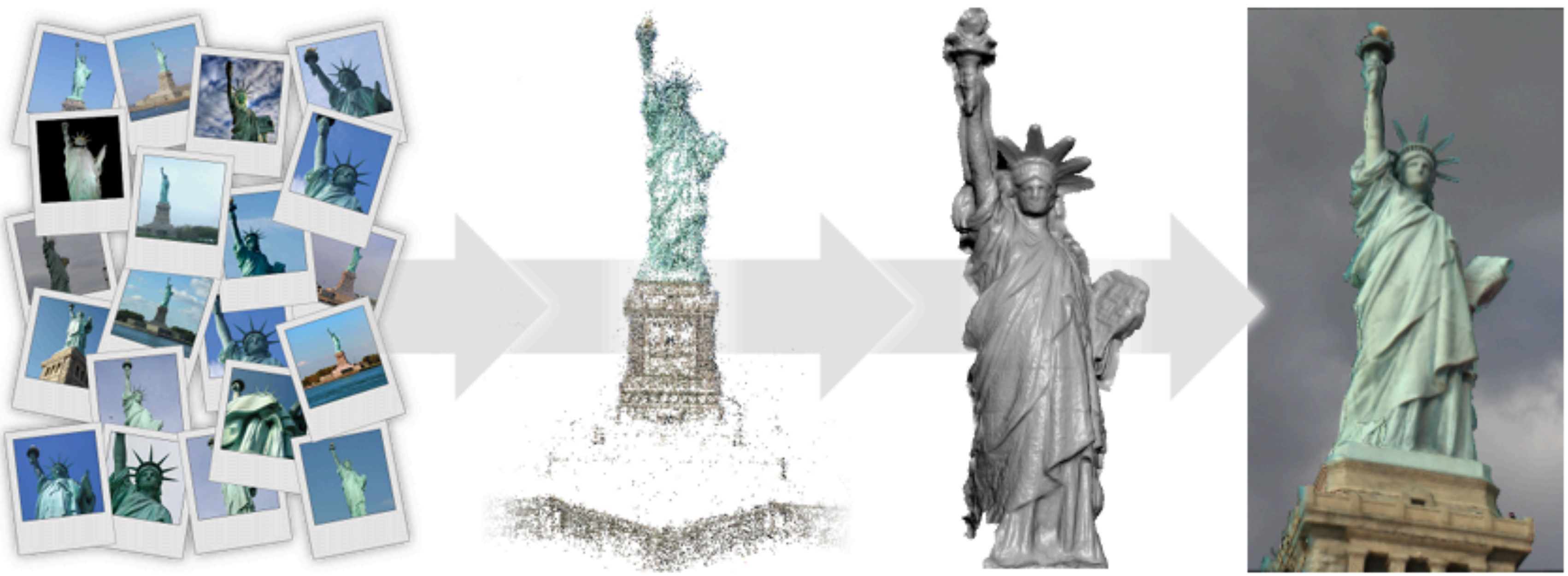
photosynth.net



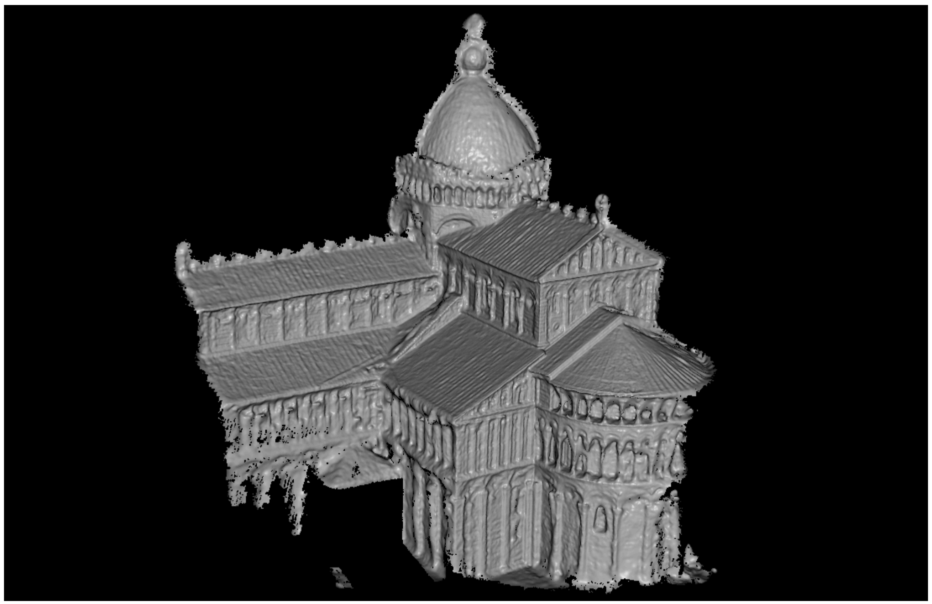
- http://photosynth.net/view.aspx?cid=29aa8616-a43a-43e4-9d6e-b8ad9b50483e

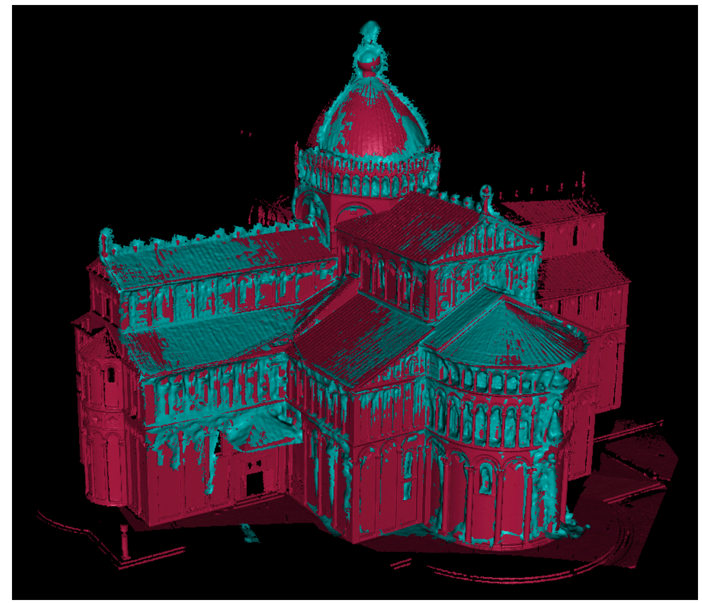
Multi-view Stereo

Multi-View Stereo for Community Photo Collections
Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz
ICCV 2007



Multi-view Stereo

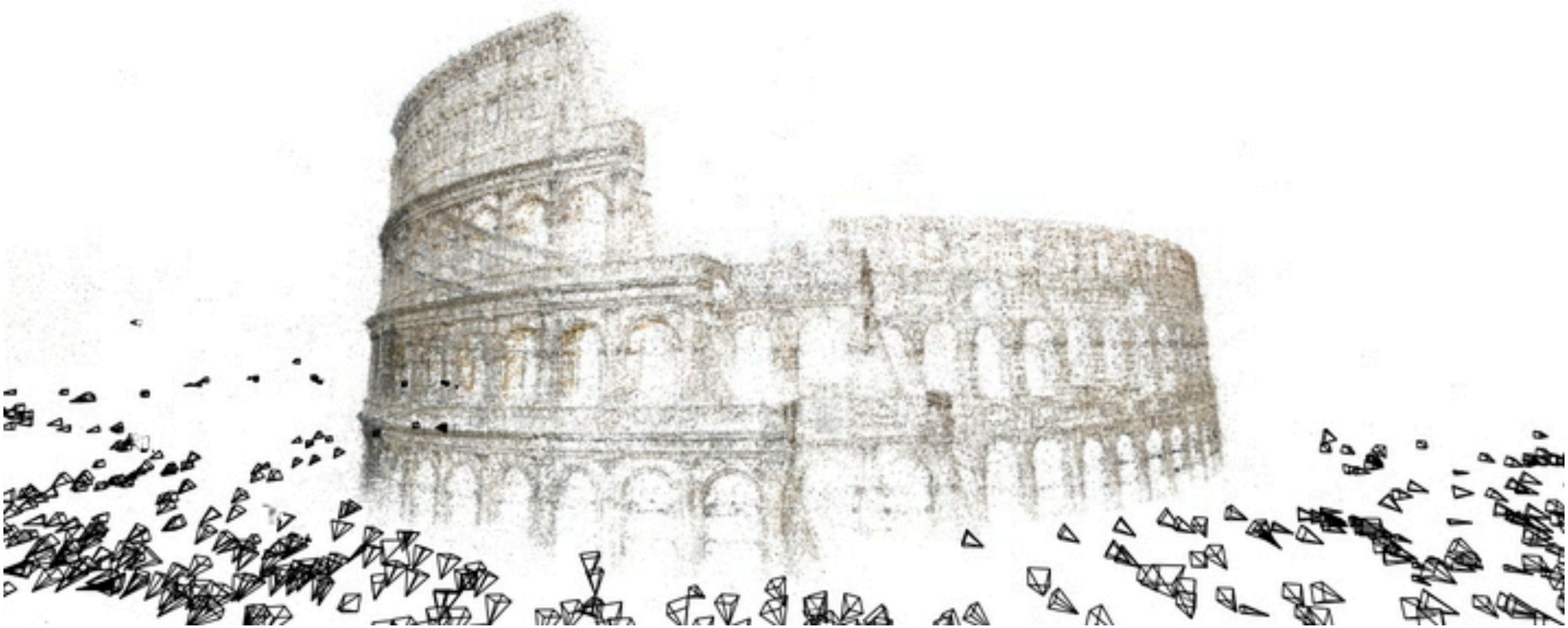




Compared with Laser-Scanner

Building Rome in a Day

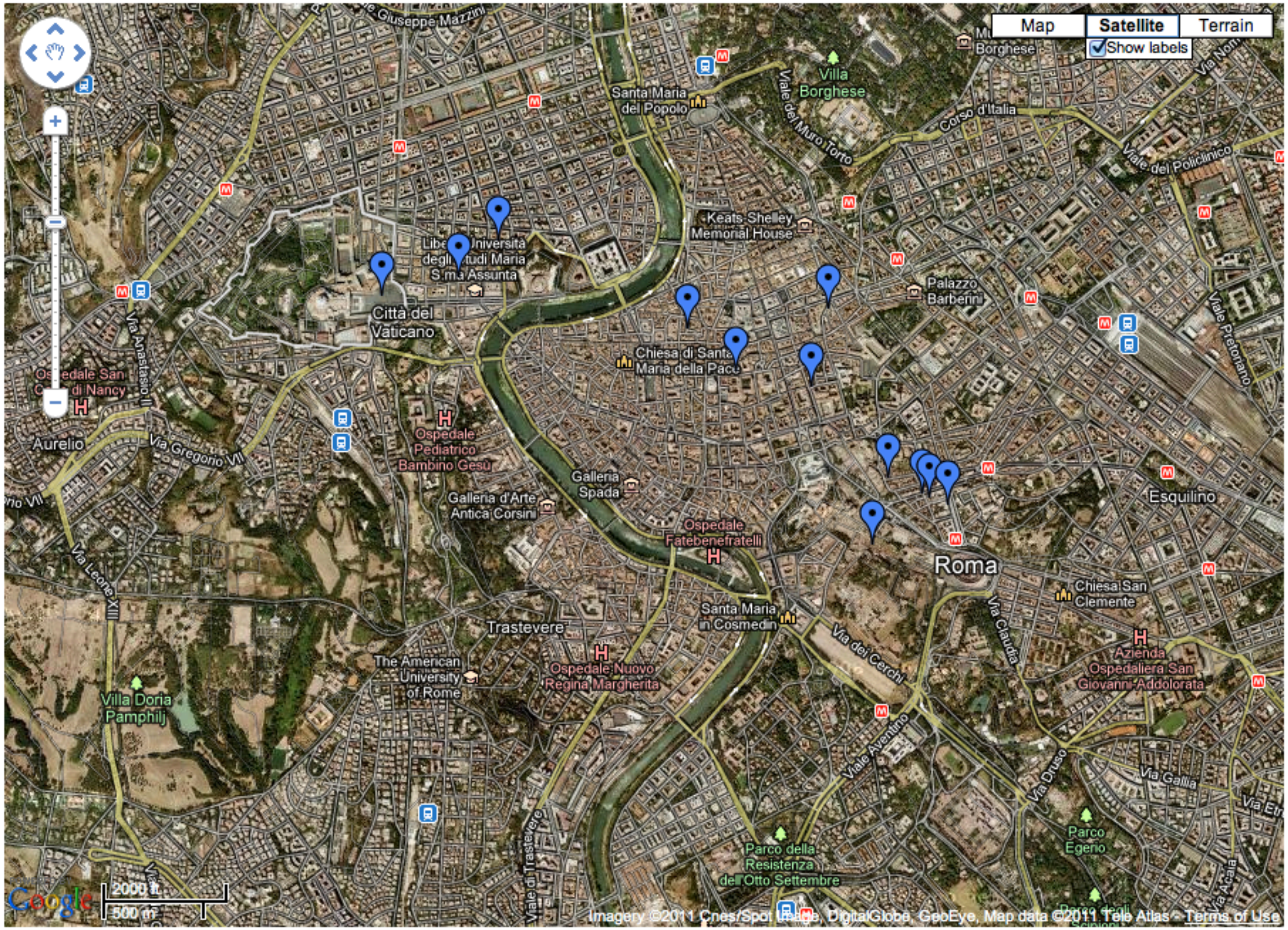
[Building Rome in a Day Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz and Richard Szeliski International Conference on Computer Vision, 2009, Kyoto, Japan.]



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

Rome on a Cloudless Day

Jan-Michael Frahm, Pierre Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, Marc Pollefeys, *ECCV 2010*



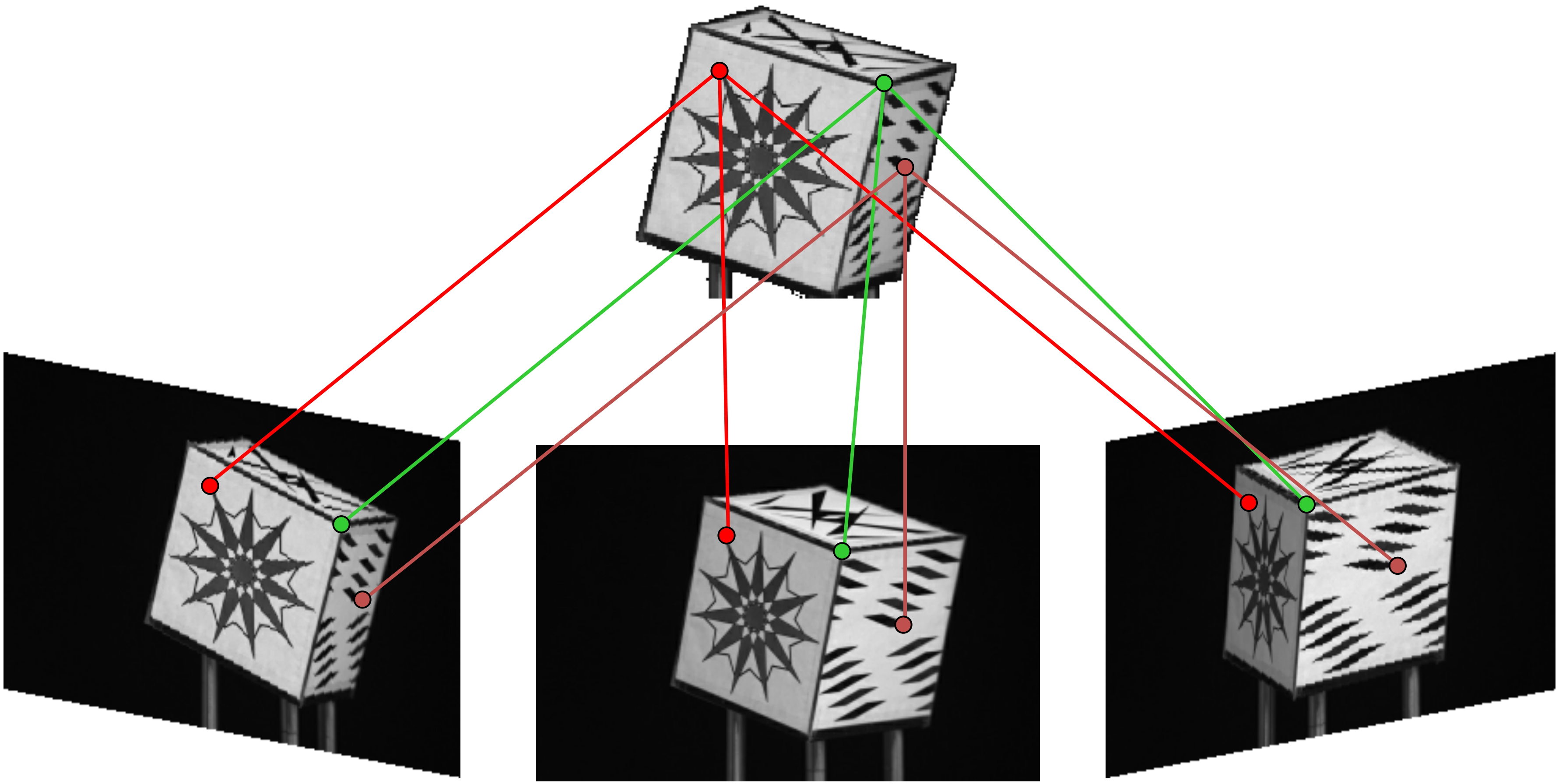
http://www.cs.unc.edu/~jmf/rome_on_a_cloudless_day/

2 Problems !

Correspondence

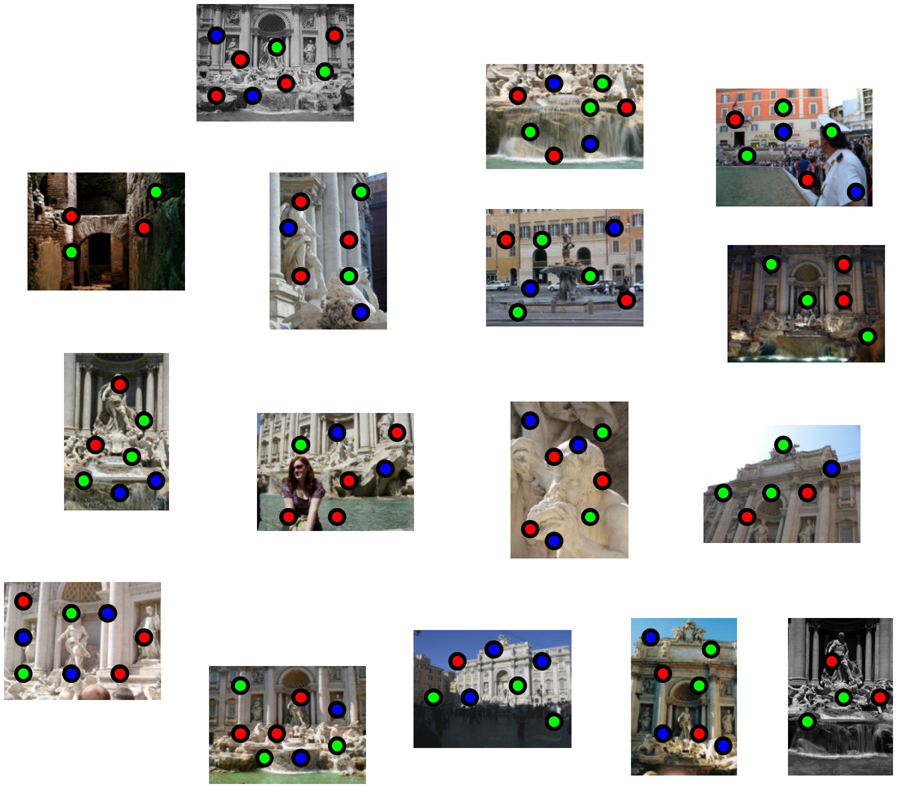
Optimization

A Correspondence Problem



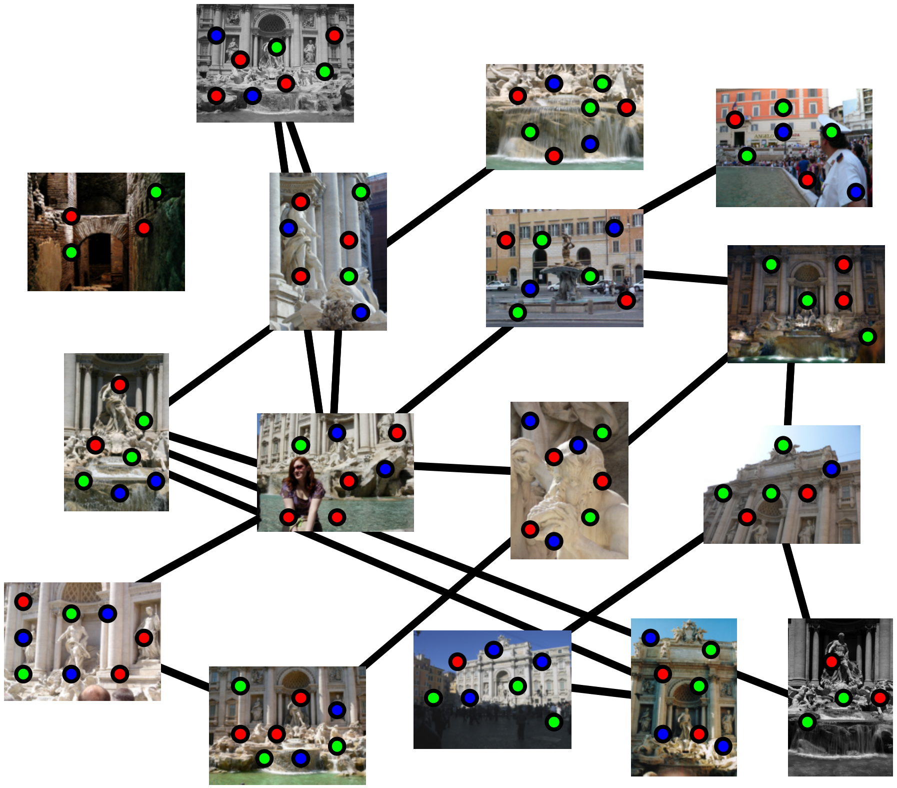
Feature detection

- Detect features using SIFT [Lowe, IJCV 2004]



Feature matching

Refine matching using RANSAC [Fischler & Bolles 1987] to estimate fundamental matrices between pairs



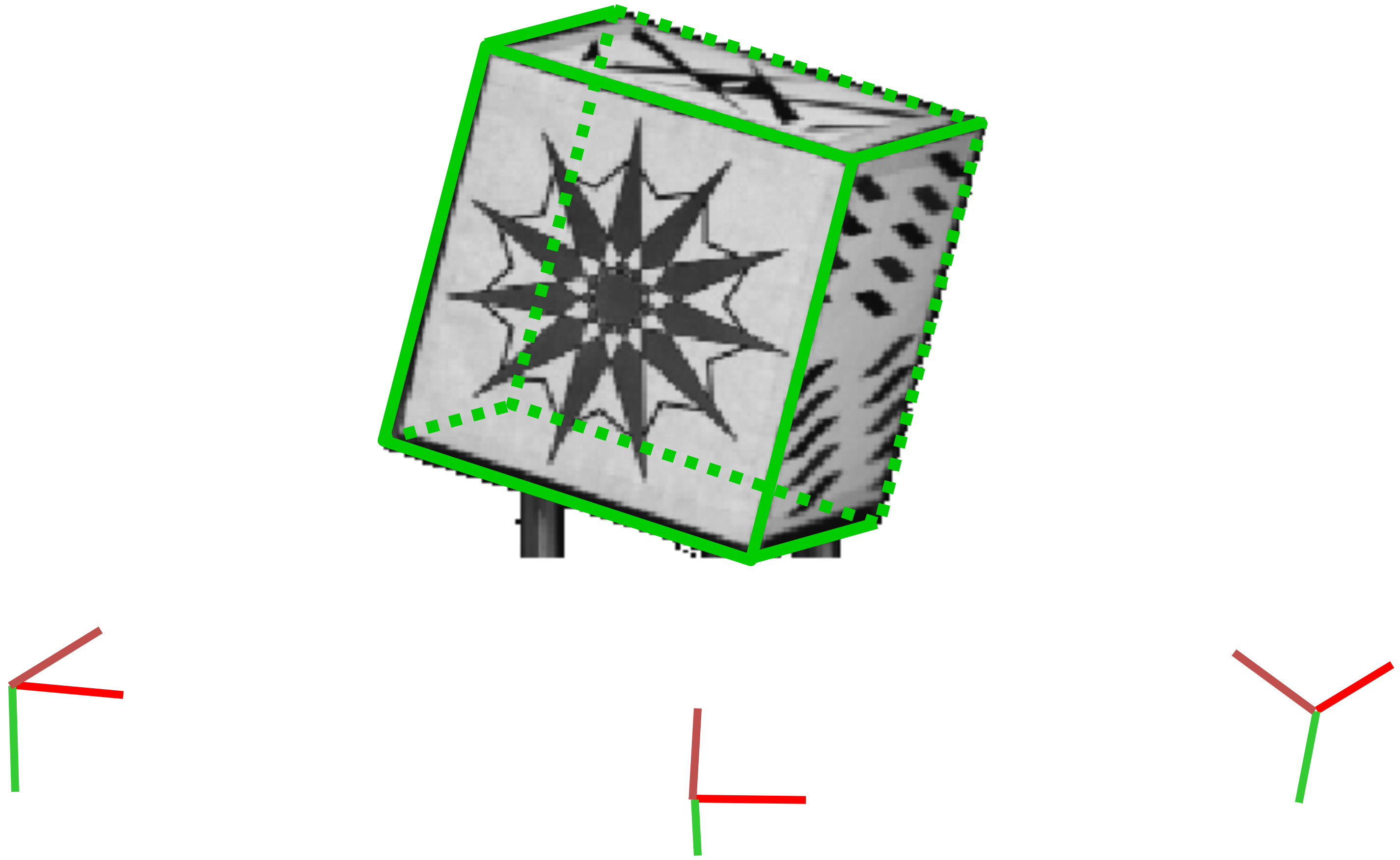
2 Problems !

Correspondence

Optimization

An Optimization Problem

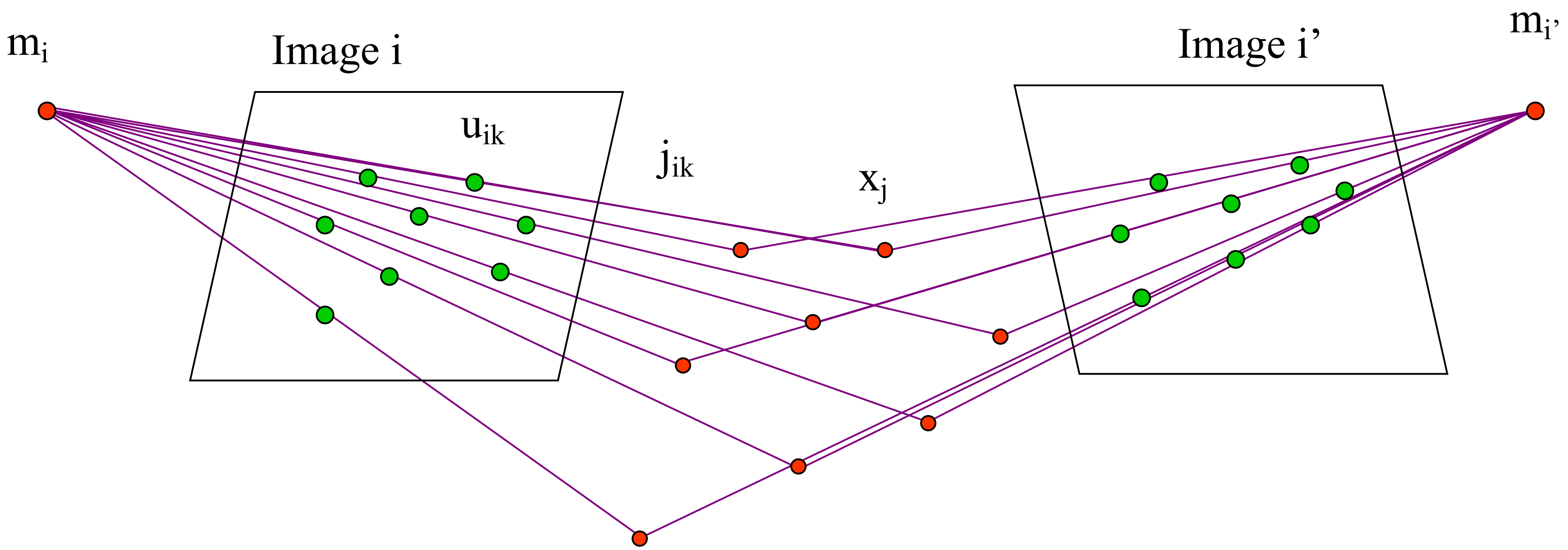
- Find the most likely structure and motion Θ



Optimization

=Non-linear Least-Squares !

$$\sum_{i=1}^{m} \sum_{k=1}^{K_i} \|\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}})\|^2$$



Recall: Nonlinear Least Squares

$$E_{NLS} = \sum_i \|f(x_i; p) - x'_i\|^2$$

Jacobian

Linearize around a current guess p:

$$f(x; p + \Delta p) = f(x; p) + J(x; p)\Delta p$$

$$r = x' - f(x; p) = J(x; p)\Delta p$$

$$E_{NLS} = \sum_i \|f(x; p) + J(x; p)\Delta p - x'_i\|^2 = \sum_i \|J(x; p)\Delta p - r_i\|^2$$

Differentiate and set to 0:

$$2 \sum_i J^T(x_i; p) \left(J(x_i; p)\Delta p - r_i\right) = 0$$

Normal equations —

$$\left[\sum_i J^T(x_i; p) J(x_i; p)\right] \Delta p = \sum_i J^T(x_i; p) r_i$$

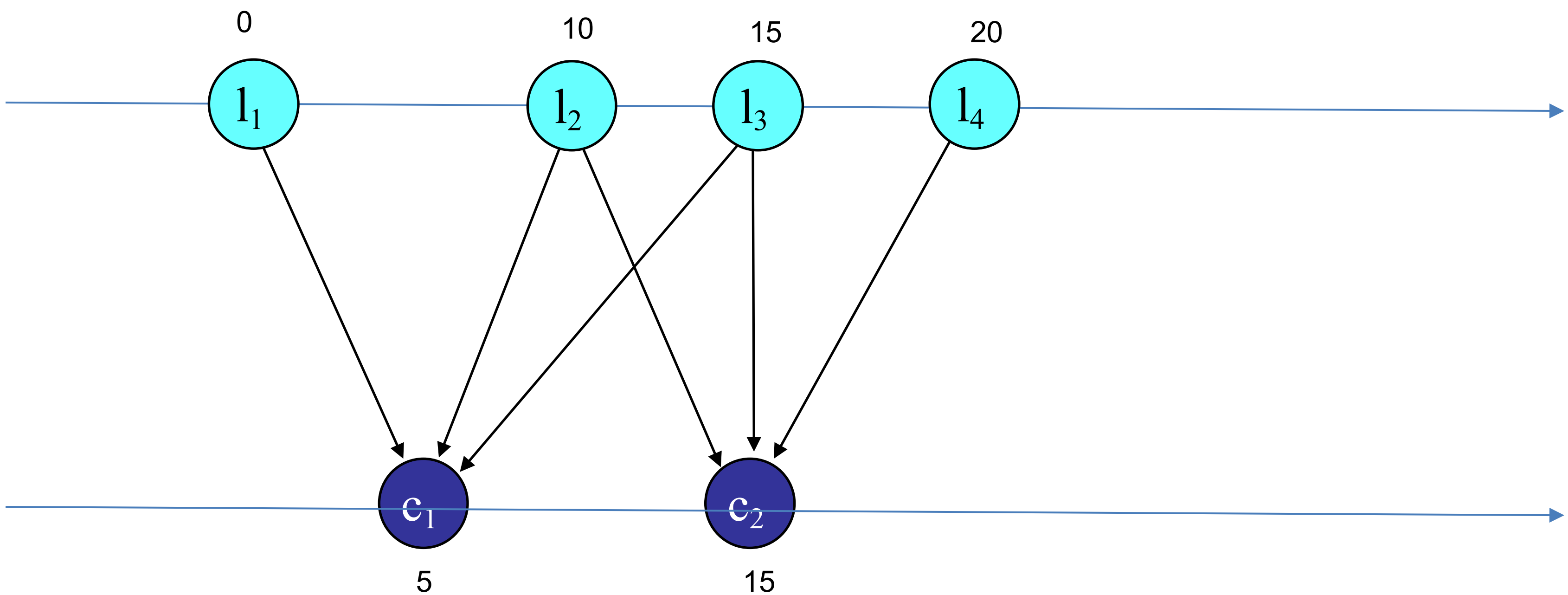
$$A\Delta p = b$$

$$\Delta p* = A^{-1} b$$

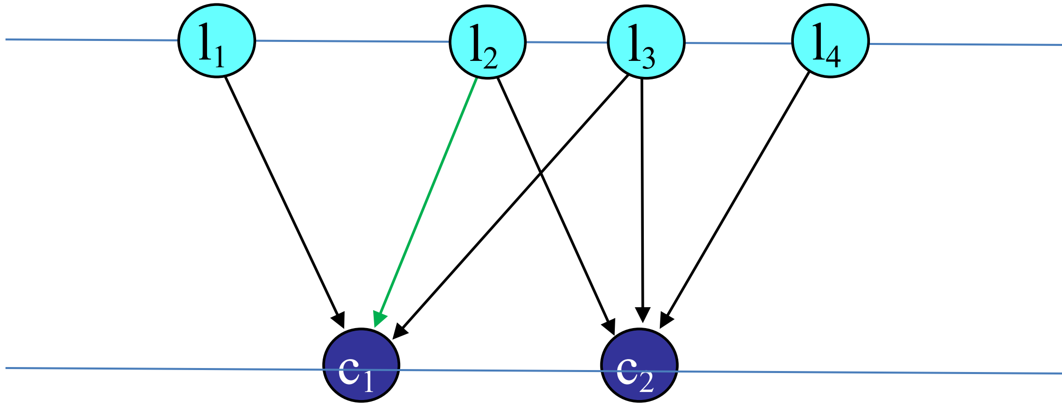
Hessian

Sparse nonlinear least squares

- Simple 1-Dimensional Example
- p = 2 cameras and 4 points {c1 c2 l1 l2 l3 l4}
- $f(u_{ik};p)$ = difference in x position = $l_{j(ik)} - c_i$



Sparse Jacobian and Hessian



A =

c1	c2	l1	l2	l3	l4
1	0	0	0	0	0
-1	0	1	0	0	0
-1	0	0	1	0	0
-1	0	0	0	1	0
0	-1	1	0	0	0
0	-1	0	1	0	0
0	-1	0	0	1	0
0	-1	0	0	0	1

b =

5
-5
5
10
-15
-5
0
5

A'*A

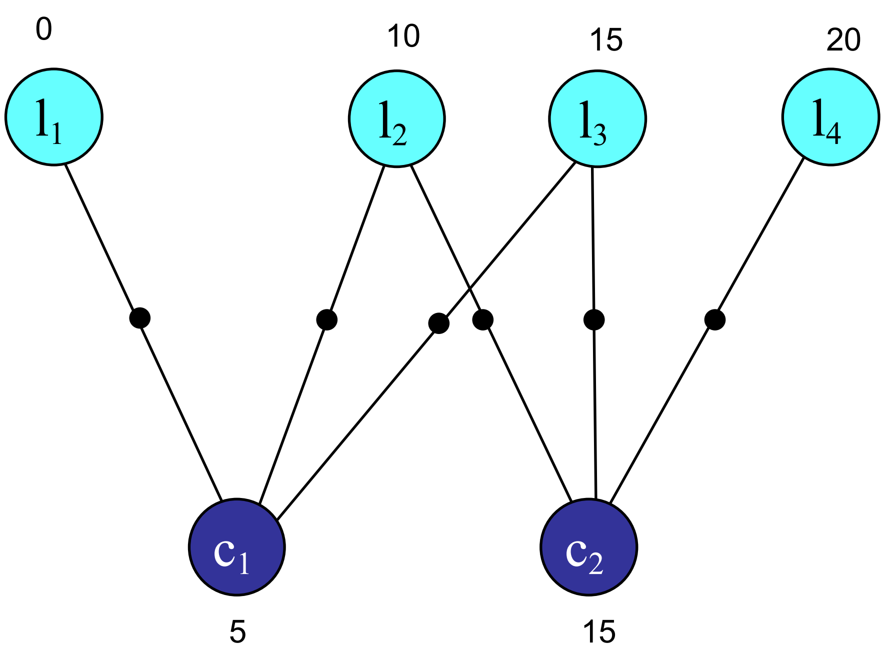
c1	c2	l1	l2	l3	l4
4	0	-1	-1	-1	0
0	4	-1	-1	-1	-1
-1	-1	2	0	0	0
-1	-1	0	2	0	0
-1	-1	0	0	2	0
0	-1	0	0	0	1

(A'*A)\A'*b =

5.0000
15.0000
0.0000
10.0000
15.0000
20.0000

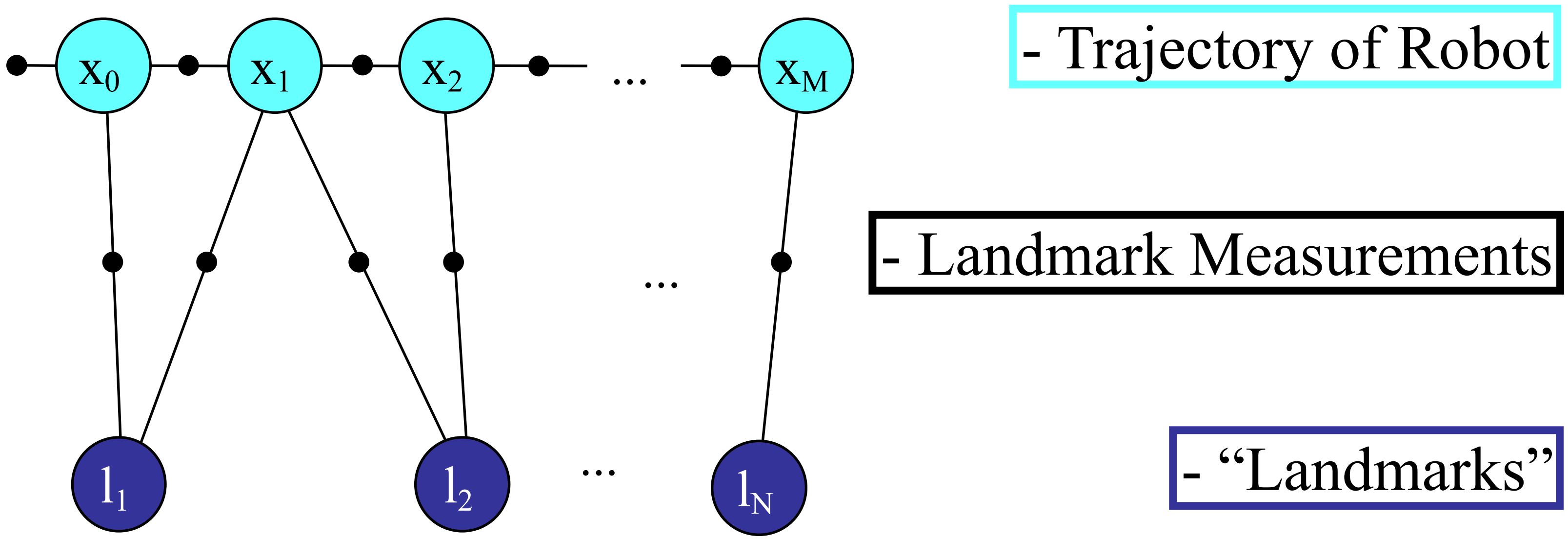
A general formalism: Factor Graphs

- Bipartite graph
- Two types of nodes:
  - Unknowns
  - Factors: correspond to squared errors
- Connectivity = sparsity! Factor is function of small set.

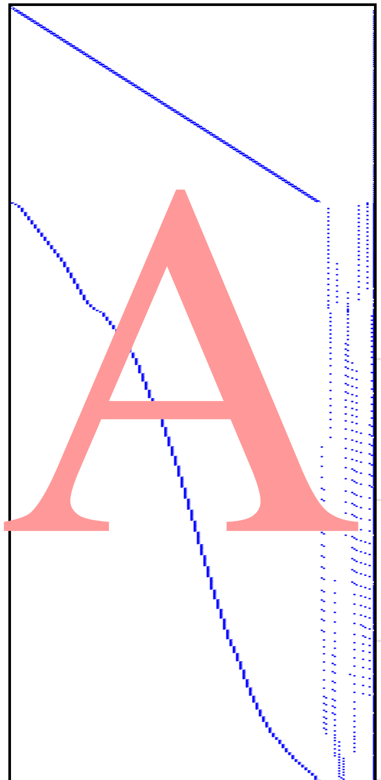


SLAM: Simultaneous Localization and Mapping

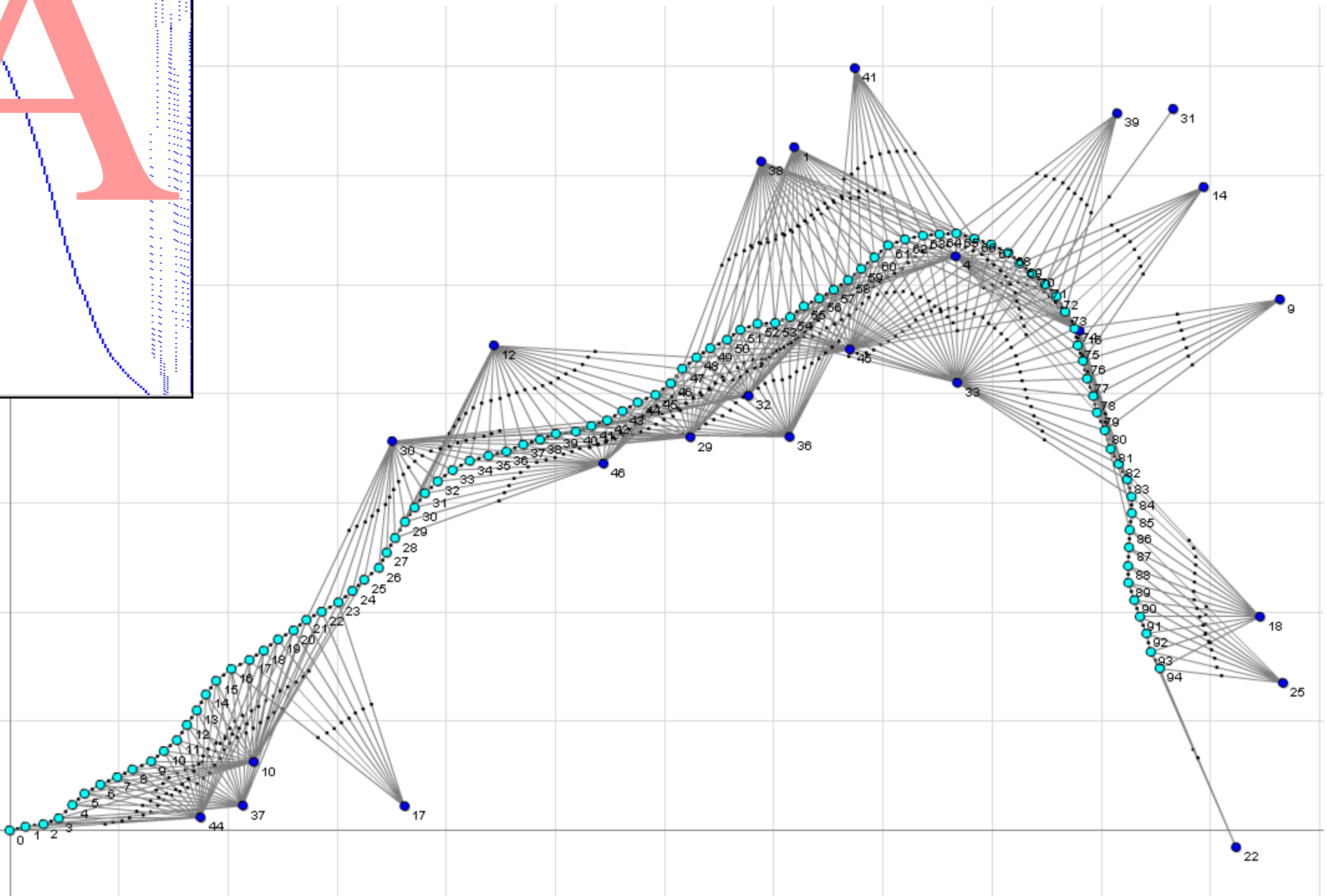
SLAM Factor Graph



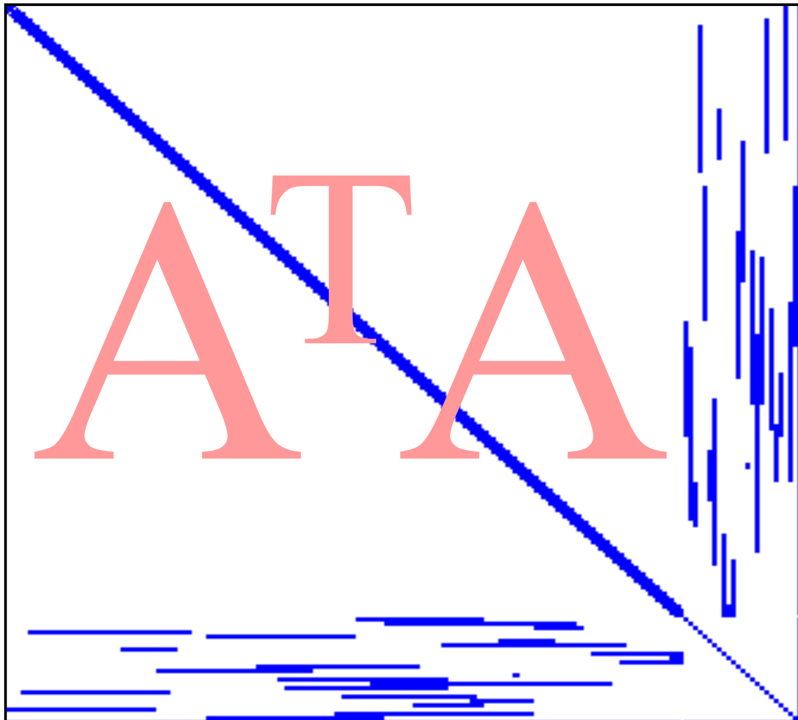
$$P(X,M) = k^* P(x_0) \prod_{i=1}^{M} P(x_i | x_{i-1}, u_i) \times \prod_{k=1}^{K} P(z_k | x_{i_k}, l_{j_k})$$

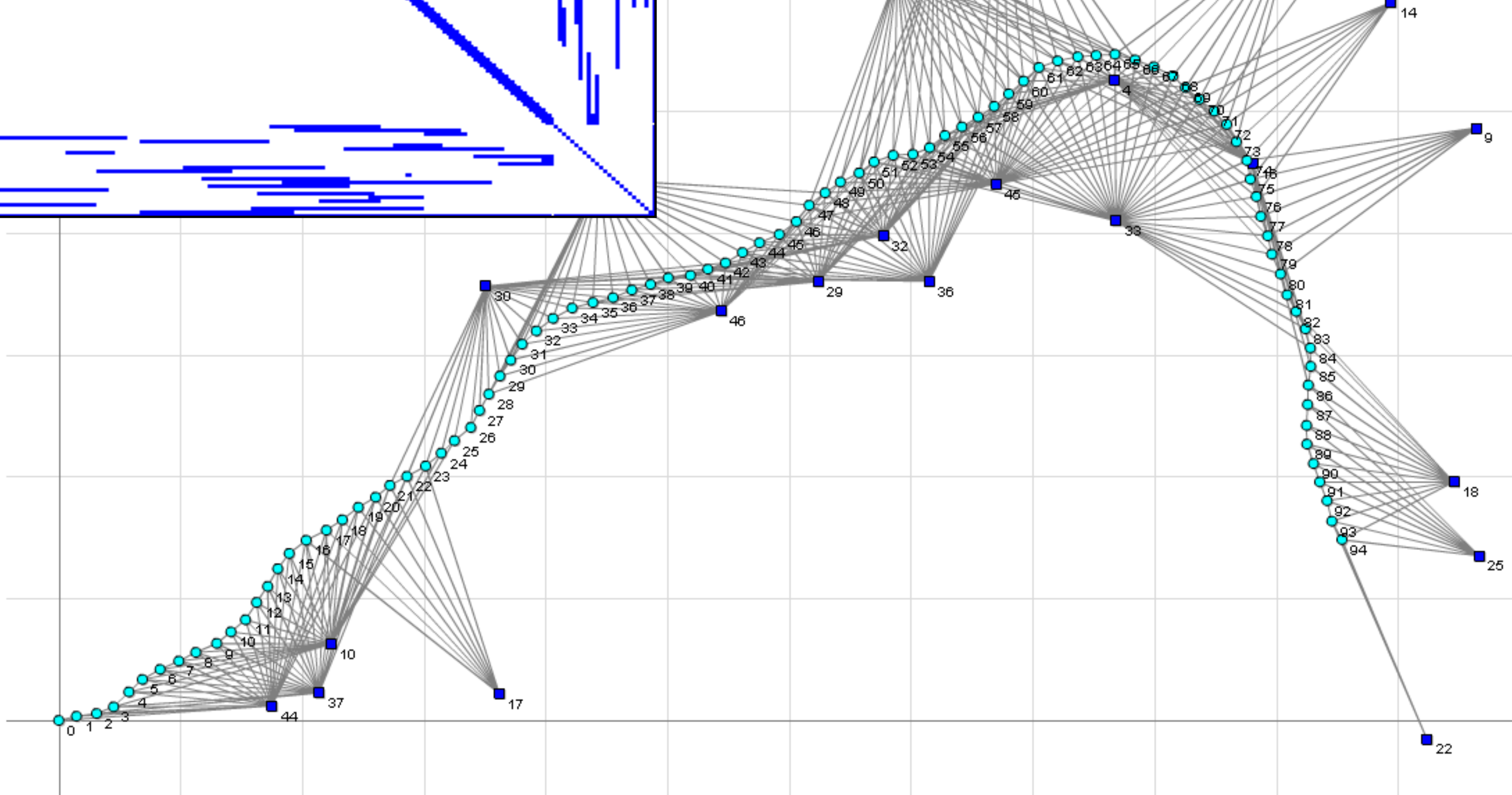


SLAM Factor Graph

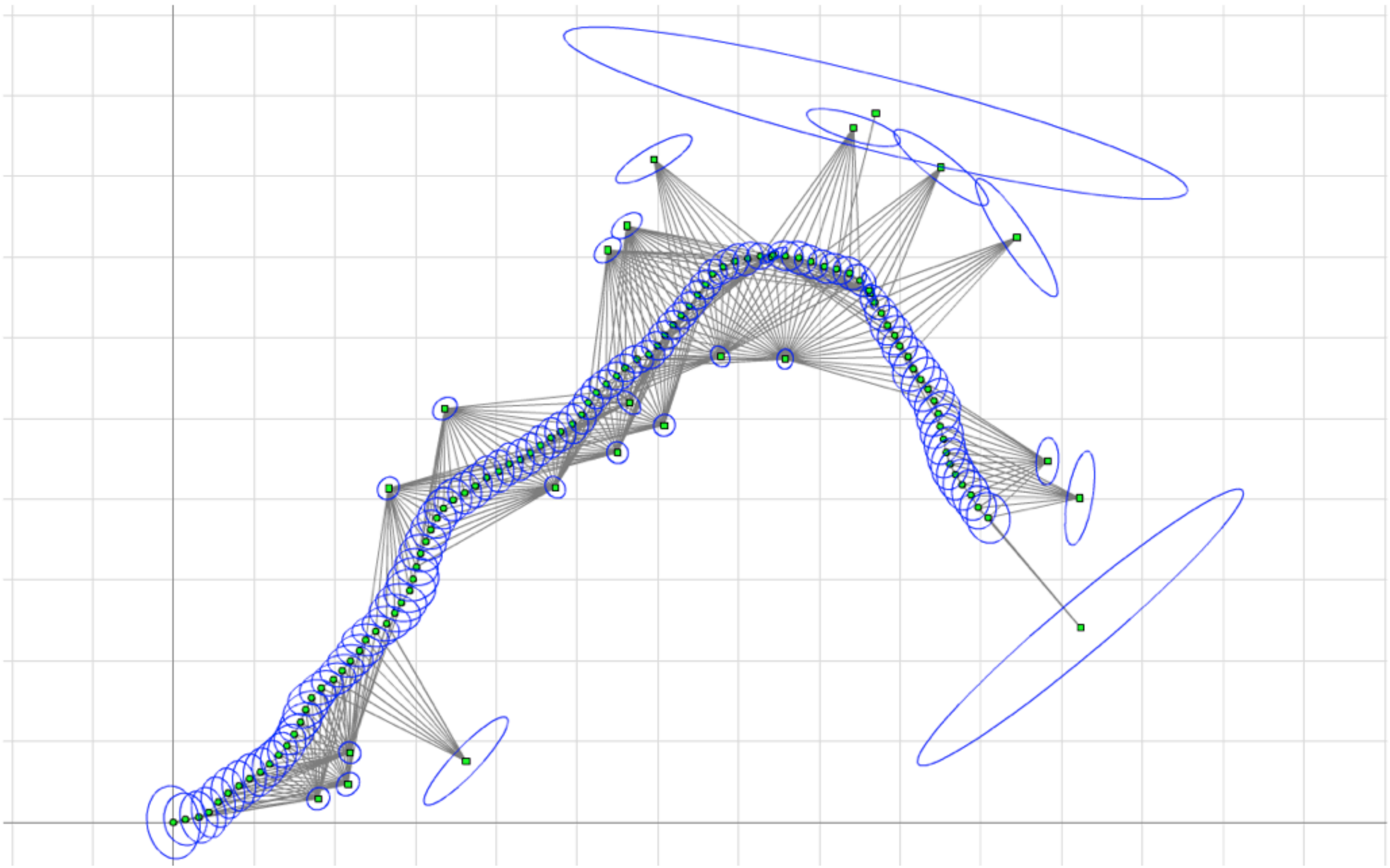


Hessian





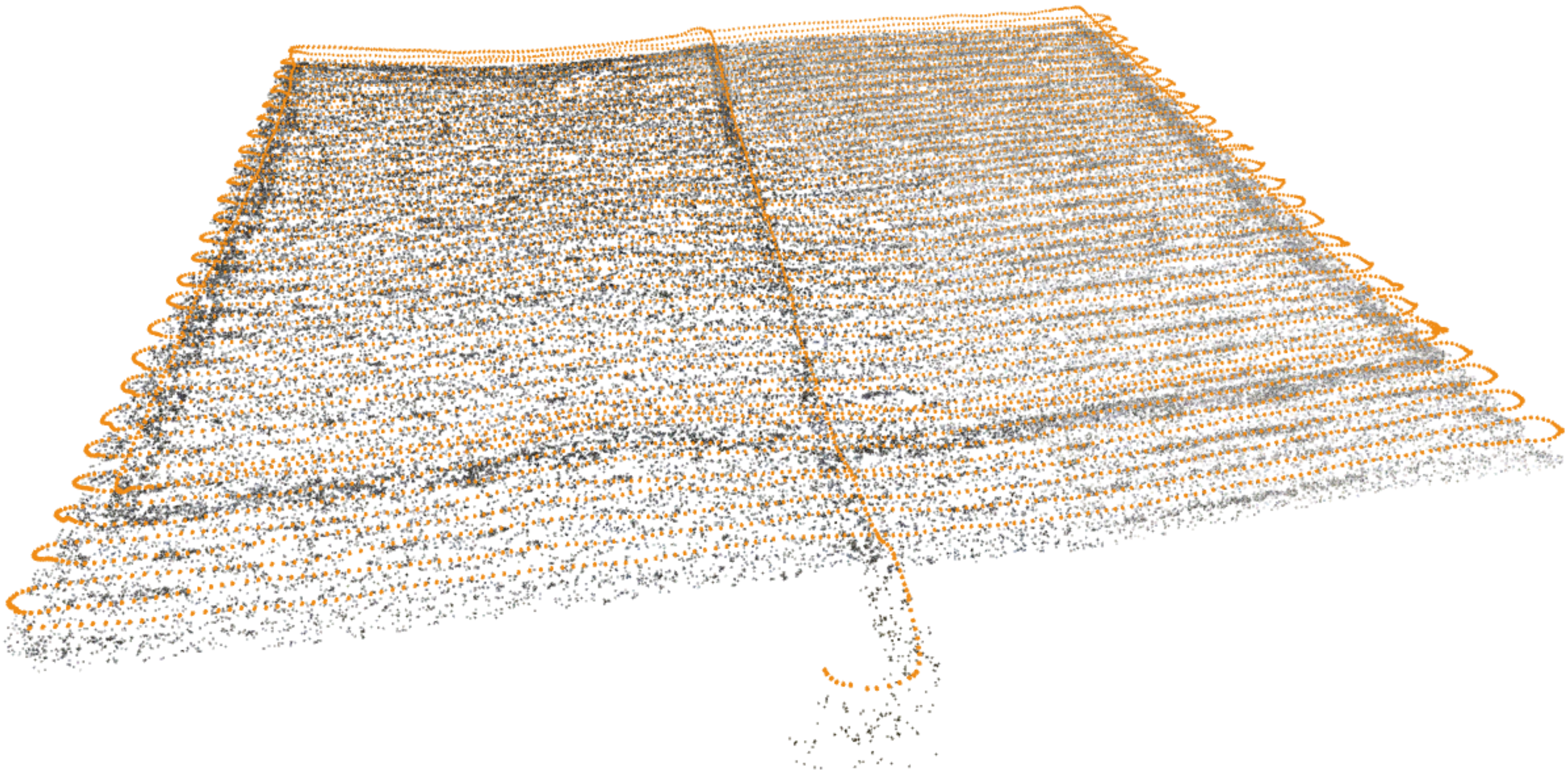
End result: Solution + Sigma



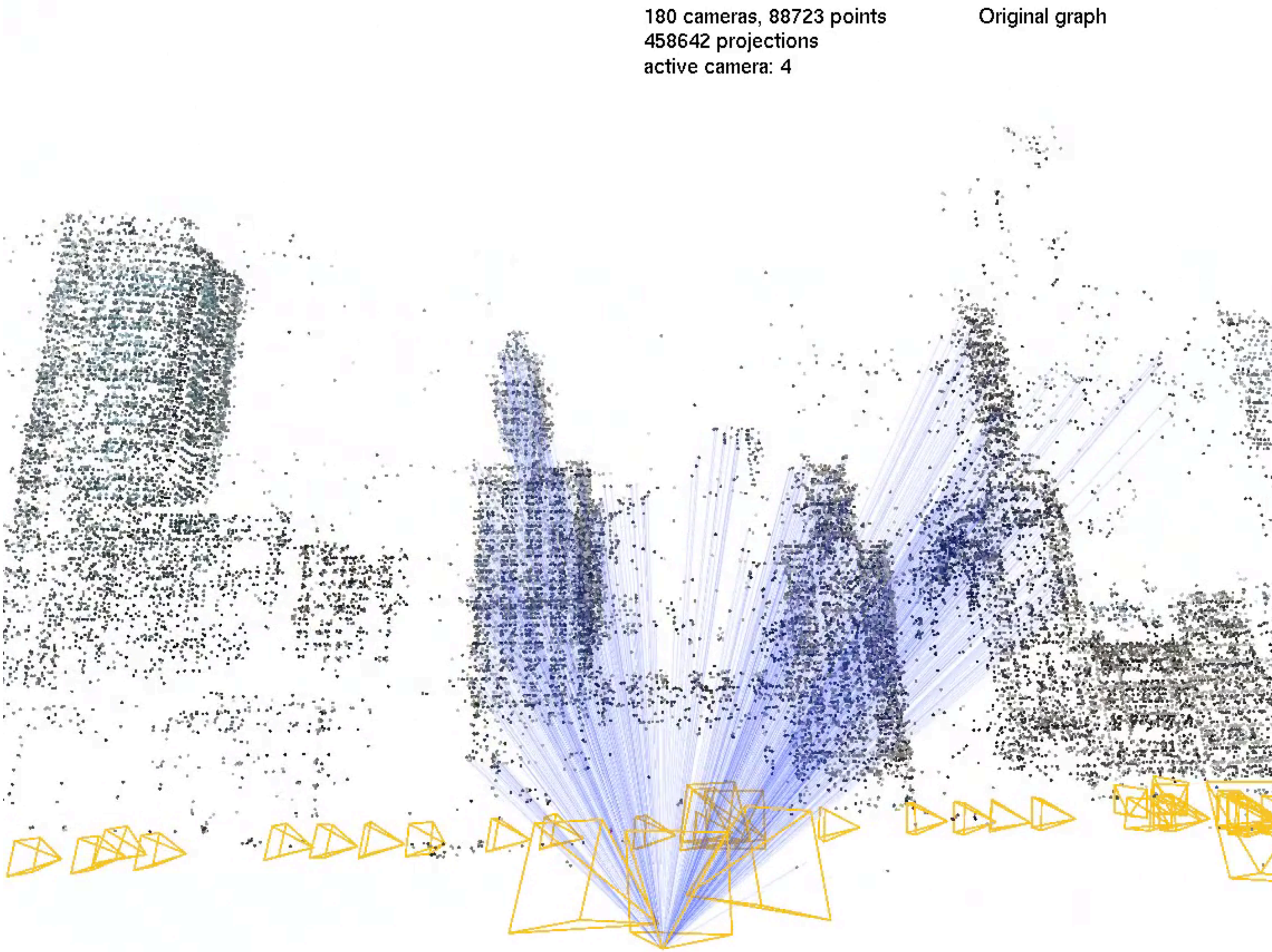
Example: Victoria Park, Sidney

Example: Underwater SLAM

9831 camera poses, 185261 landmarks, and 350988 factors



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



3D Models from Community Databases

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

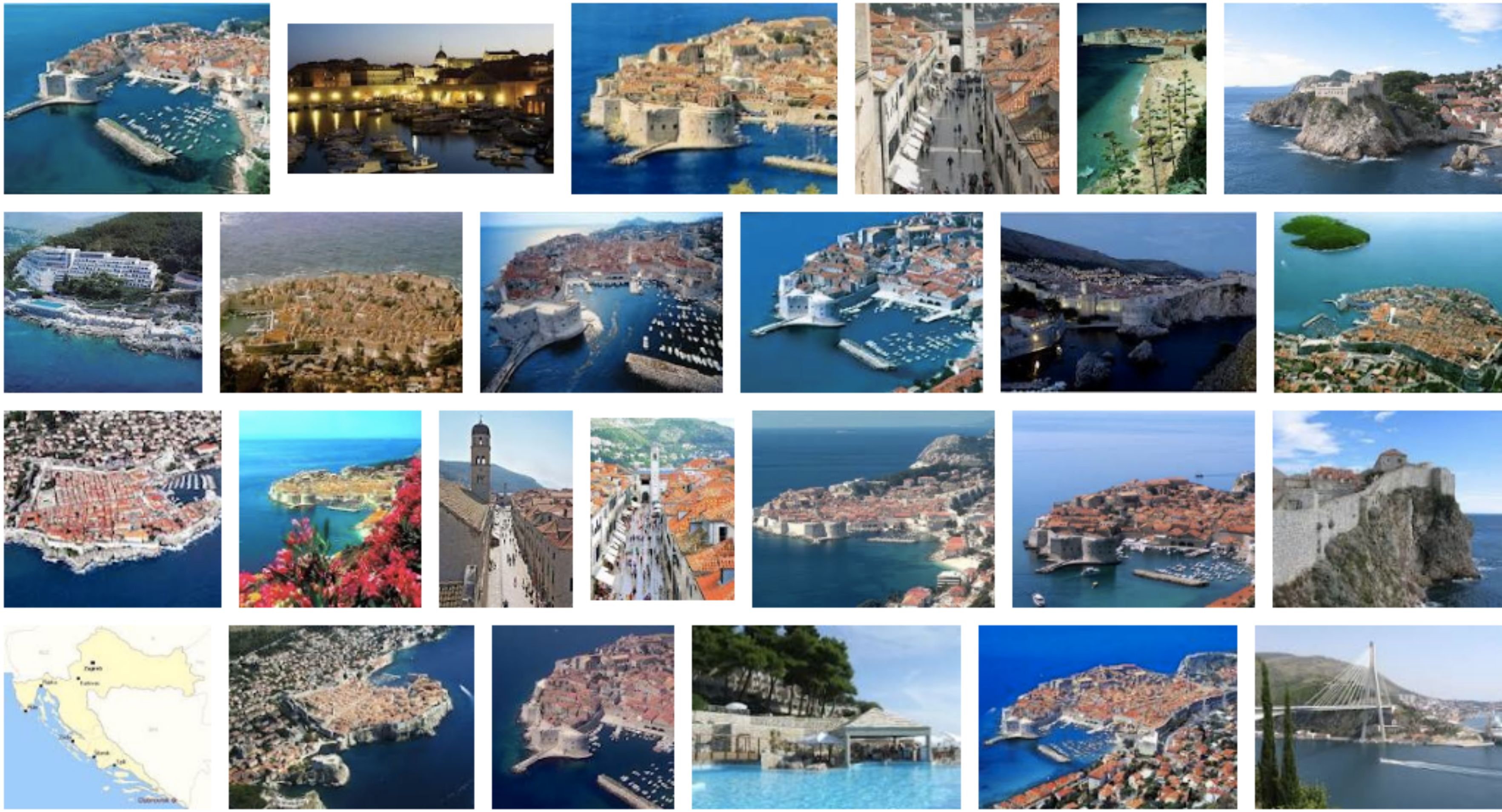


Figure by Aggarwal et al.

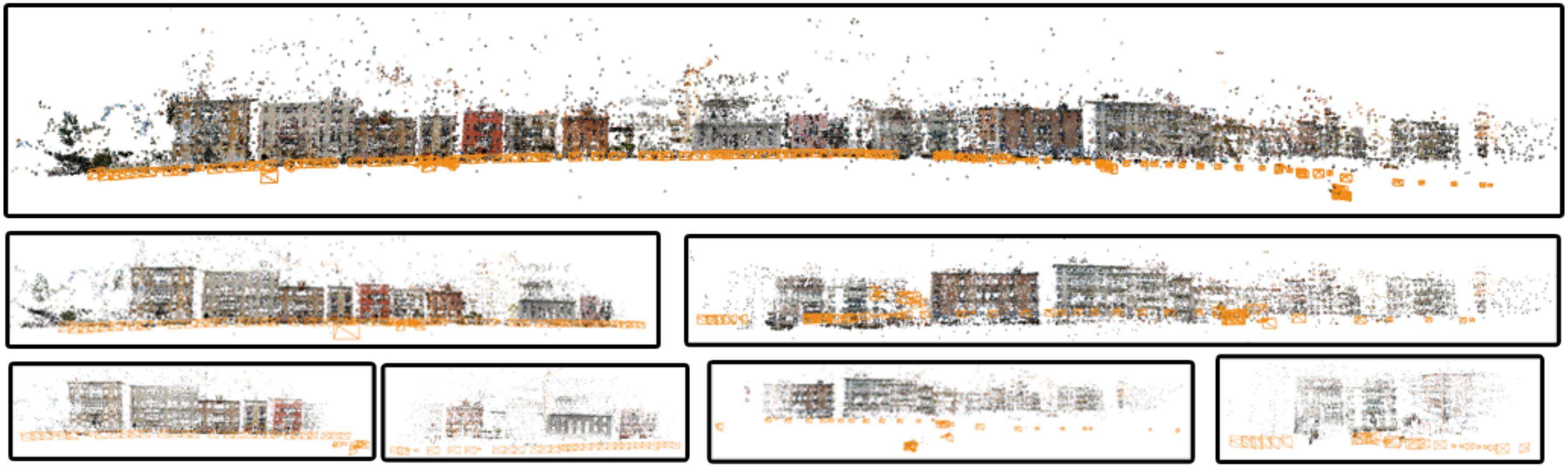
3D Models from Community Databases

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.

Hyper-SFM: Efficient Multi-core



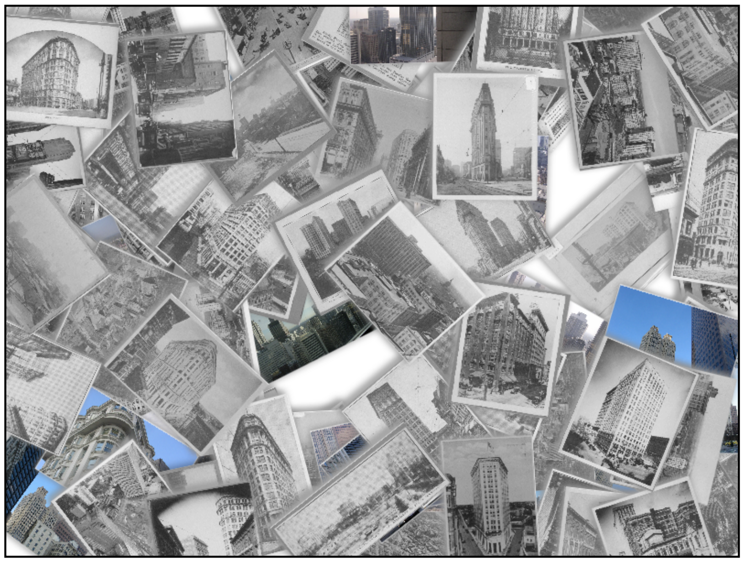
Kai now leads an autonomous driving startup in China

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

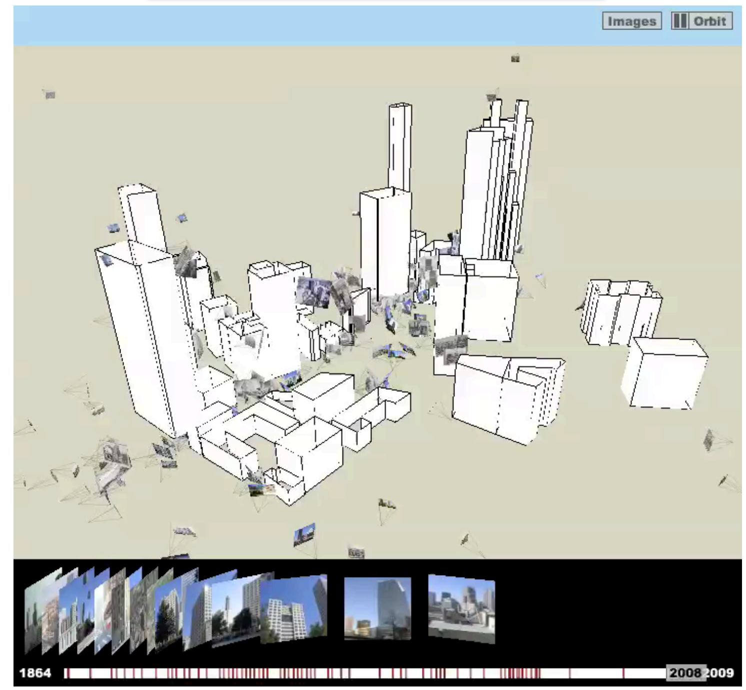
4D Reconstruction

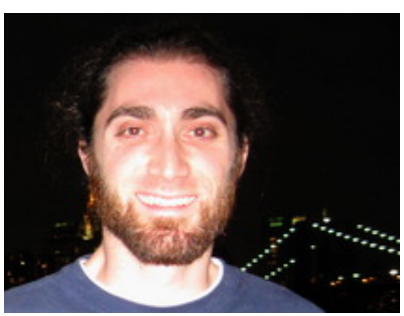
Spatiotemporal Reconstruction

4D Cities: 3D + Time



Historical Image Collection





Grant Schindler

Supported by NSF CAREER, Microsoft

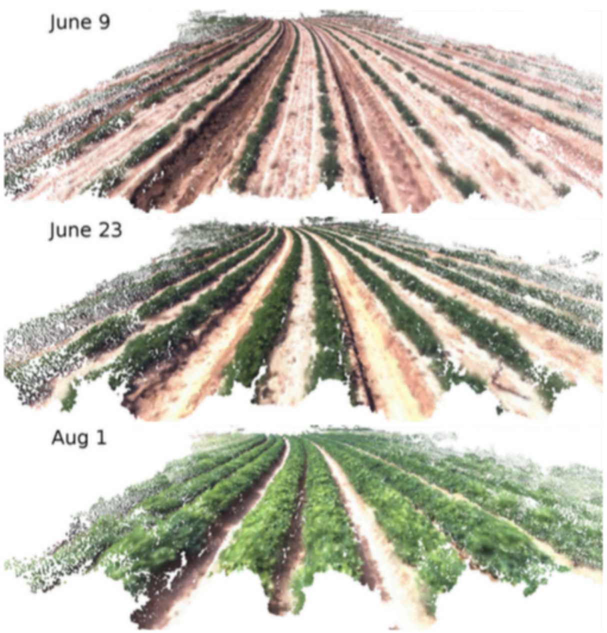
Recent revival: NSF NRI award on 4D crops for precision agriculture…

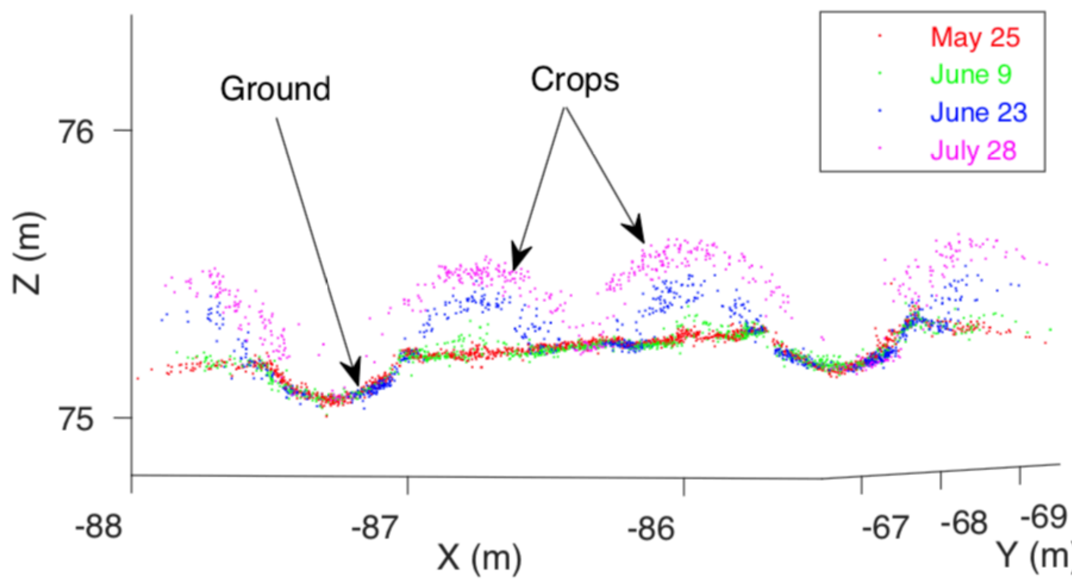
4D Reconstruction of Lower Manhattan

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

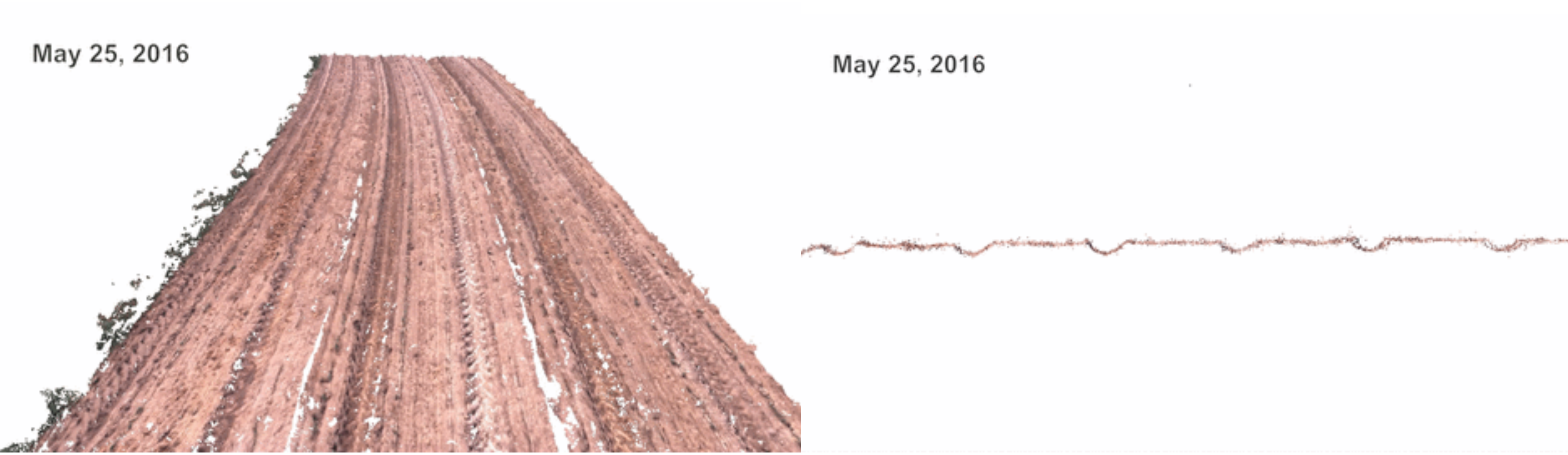
4D Structure over Time

4D crop monitoring (Jing Dong)





Results: video (by Jing Dong)



4D reconstruction results (by PMVS)
and its cross section