COLLEGE OF ENGINEERING
ELECTRICAL ENGINEERING & COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN

Projective Structure from Motion

(Uncalibrated Perspective Cameras)

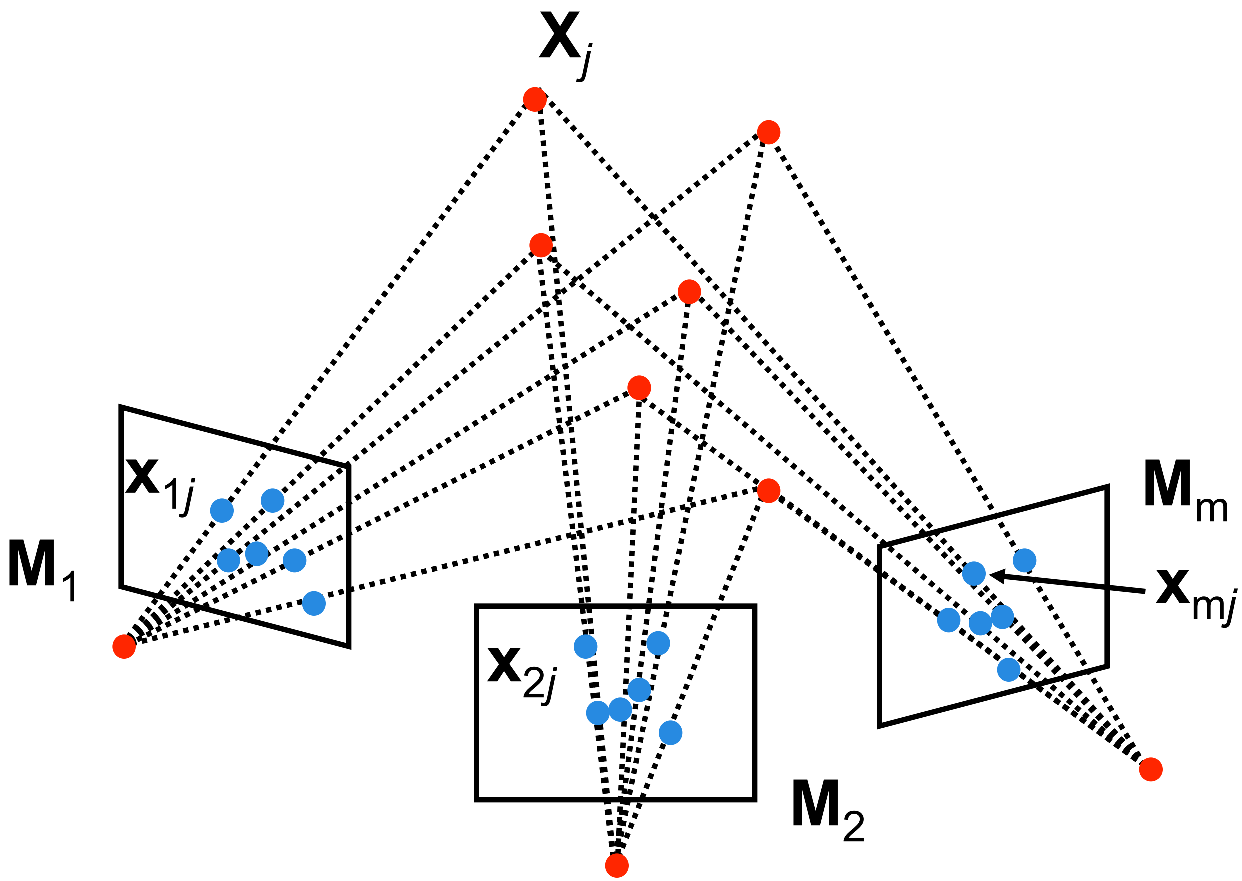
EECS 598-08 Fall 2014
Foundations of Computer Vision

Instructor: Jason Corso (jjcorso)
web.eecs.umich.edu/~jjcorso/t/598F14

Readings: FP 8.3
Date: 11/5/14

Materials on these slides have come from many sources in addition to myself (primarily Silvio Savarese to whom I am ultimately grateful); individual slides reference specific sources.

Structure from motion problem

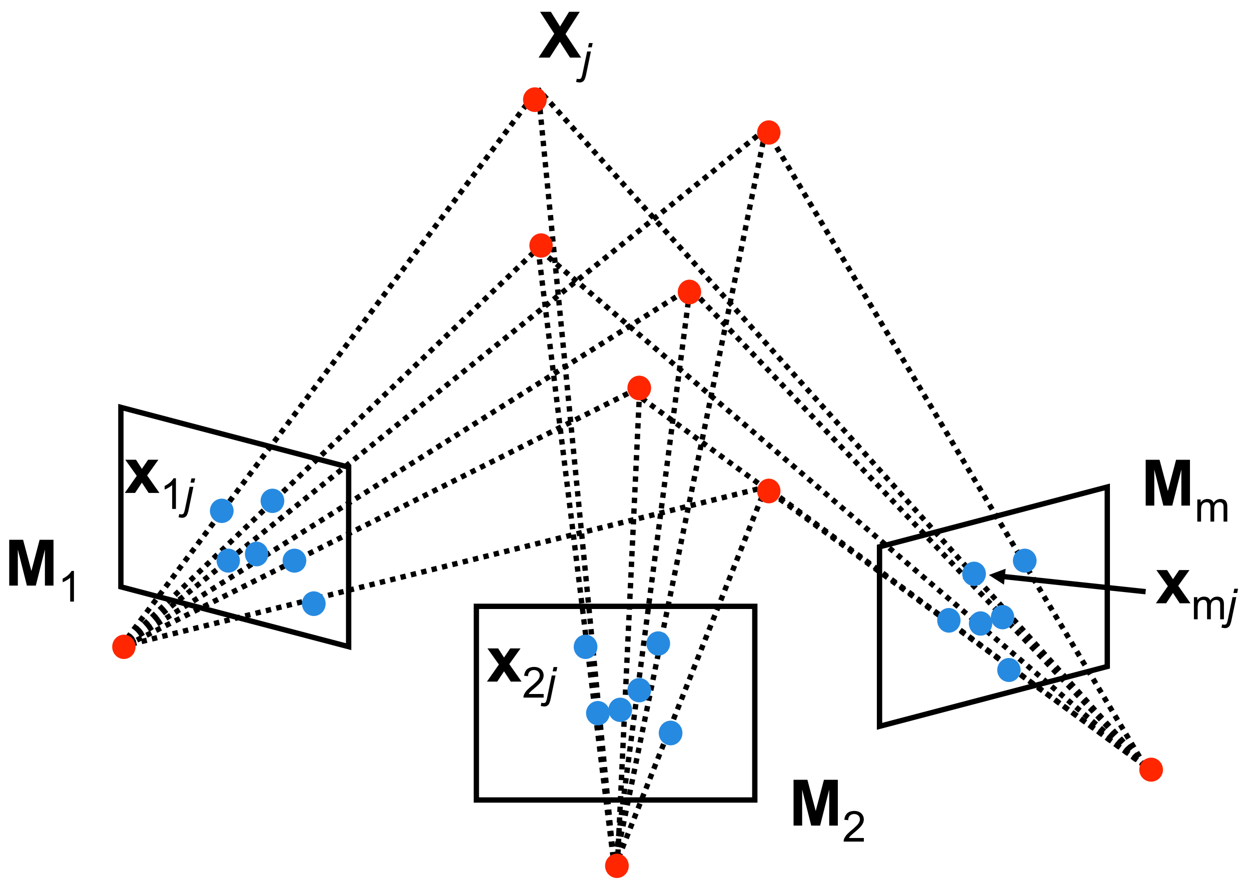


Given *m* images of *n* fixed 3D points

- $\mathbf{x}_{ij} = \mathbf{M}_i \mathbf{X}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$

Courtesy of Silvio Savarese.

Structure from motion problem



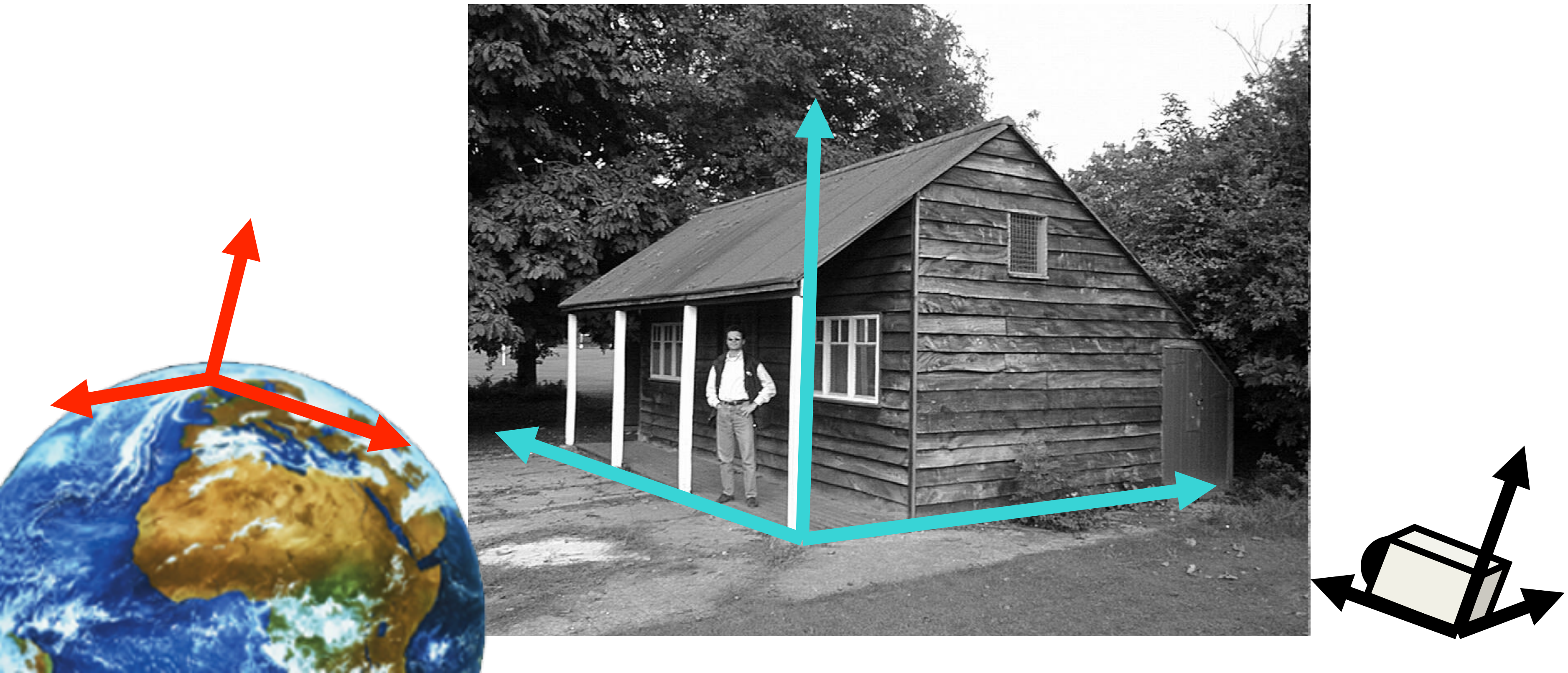
From the mxn correspondences $\mathbf{x}_{ij}$, estimate:

•*m* projection matrices $\mathbf{M}_i$	motion
•*n* 3D points $\mathbf{X}_j$	structure

Courtesy of Silvio Savarese.

Structure from motion ambiguity

- Position ambiguity: it is impossible based on the images alone to estimate the absolute location and pose of the scene w.r.t. a 3D world coordinate frame



$$H_s = \begin{bmatrix} s & & \\ & s & \\ & & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} R & t \\ 0 & 1/s \end{bmatrix}$$

$$x_j = M_i \, X_j \qquad M_i = K_i \begin{bmatrix} R_i & T_i \end{bmatrix}$$

$$X_j \rightarrow H_s \, X_j \qquad M_i \rightarrow M_i \, H_s^{-1}$$

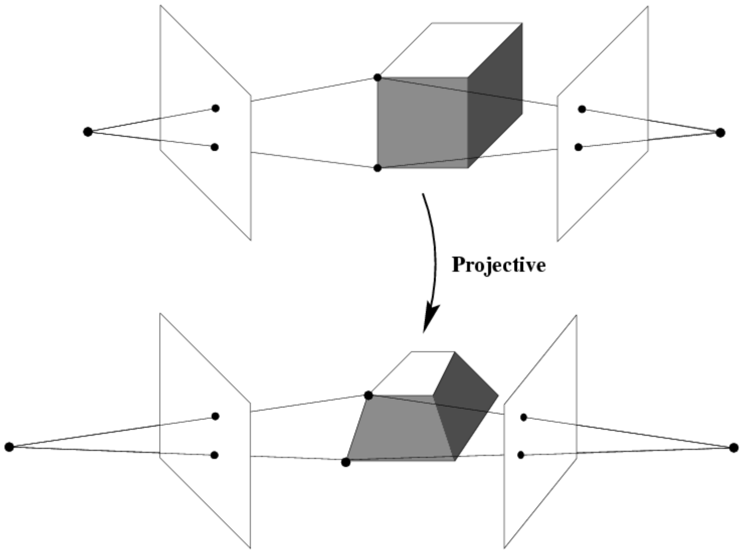
$$\tilde{x}_j = M_i \, H_s^{-1} H_s X_j = M_i \, X_j = x_j$$

$$M_i \, H_s^{-1} = K_i \begin{bmatrix} R_i & T_i \end{bmatrix} H_s^{-1} = K_i \begin{bmatrix} R_i \, R^{-1} & T_i' \end{bmatrix}$$

The calibration matrix has not changed!

Courtesy of Silvio Savarese.

Structure from motion ambiguity



- In the general case (nothing is known) the ambiguity is expressed by an arbitrary affine or projective transformation

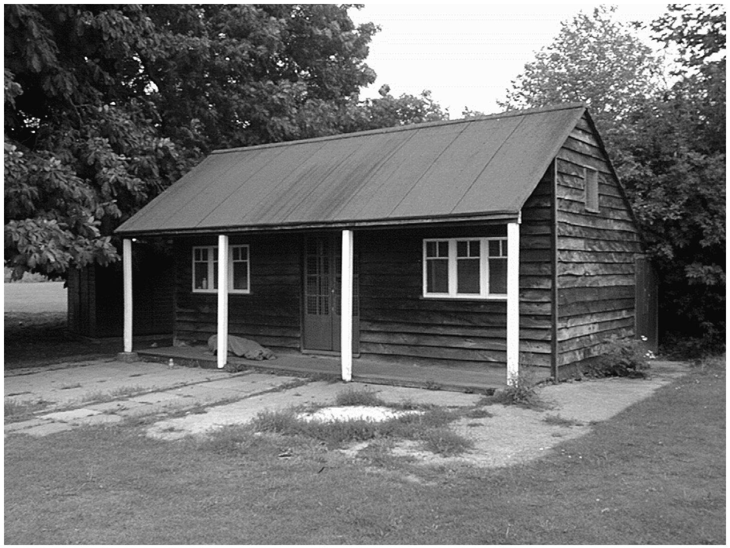
$$x_j = M_i X_j \qquad M_i = K_i \begin{bmatrix} R_i & T_i \end{bmatrix}$$

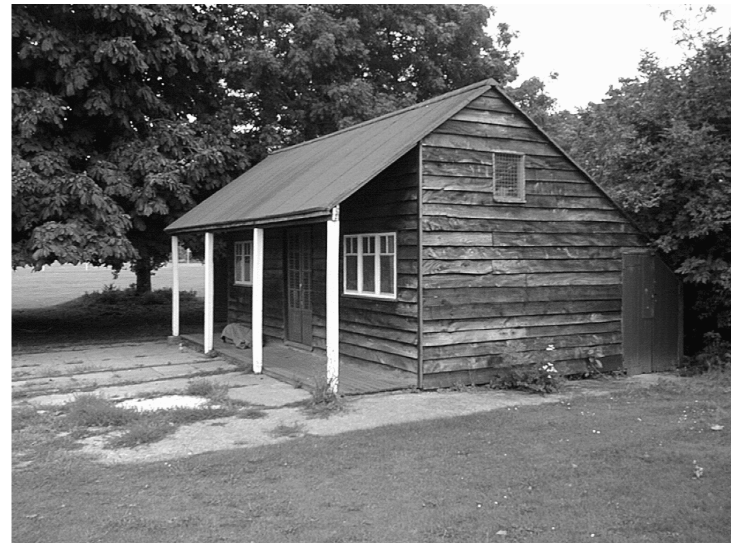
$$X_j \rightarrow H X_j \qquad M_i \rightarrow M_j H^{-1}$$

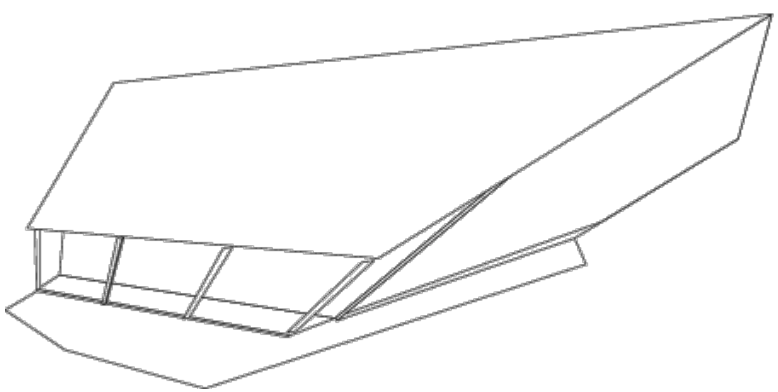
$$x_j = M_i X_j = \left(M_i H^{-1}\right)\left(H X_j\right)$$

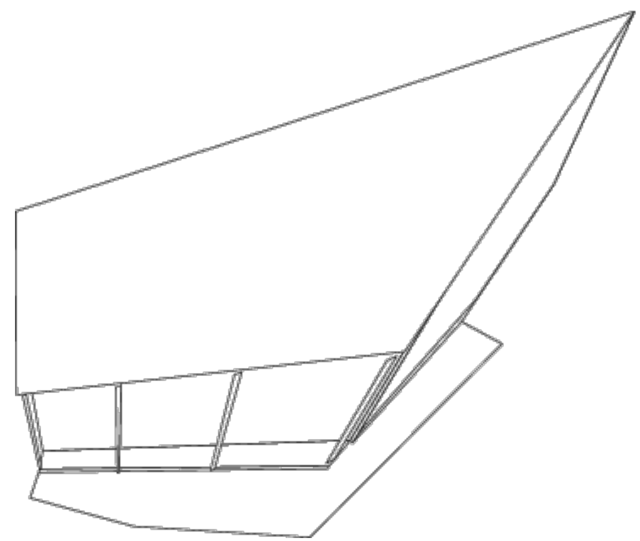
Courtesy of Silvio Savarese.

Projective ambiguity



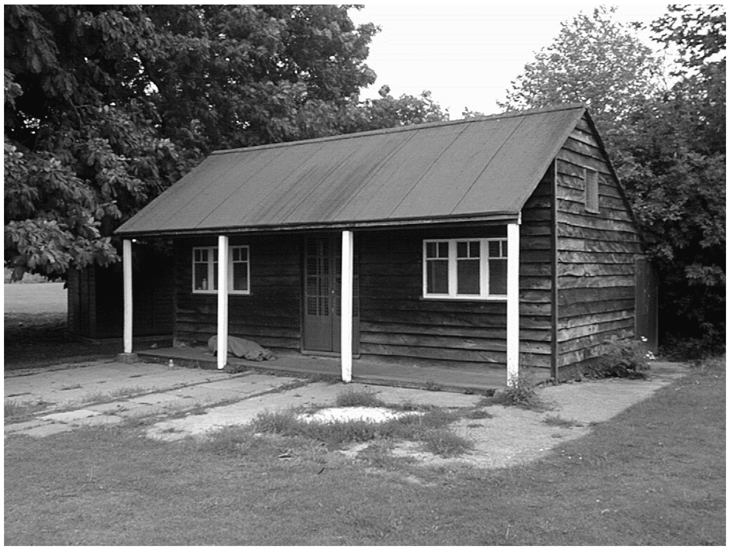


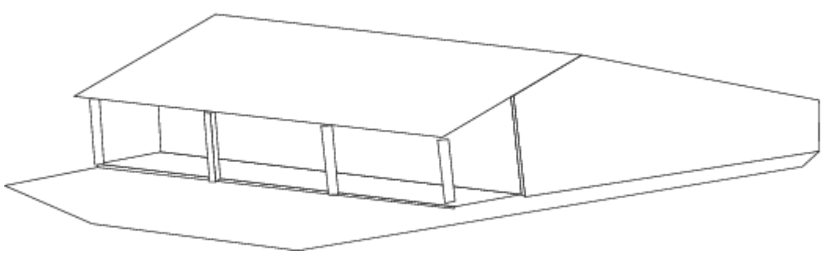


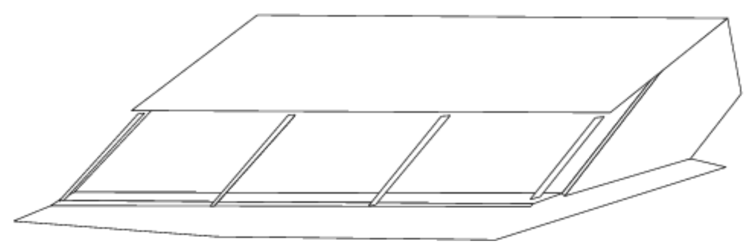


Courtesy of Silvio Savarese.

Affine ambiguity





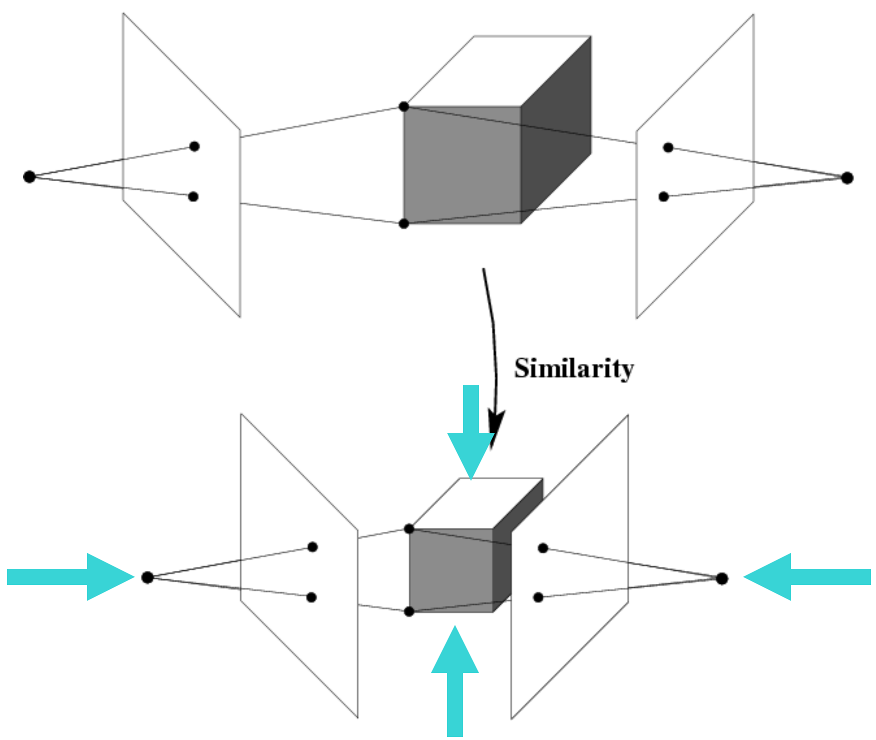


Courtesy of Silvio Savarese.

Structure from motion ambiguity

- The ambiguity exists even for calibrated cameras
- For calibrated cameras, the similarity ambiguity is the only ambiguity

[Longuet-Higgins '81]

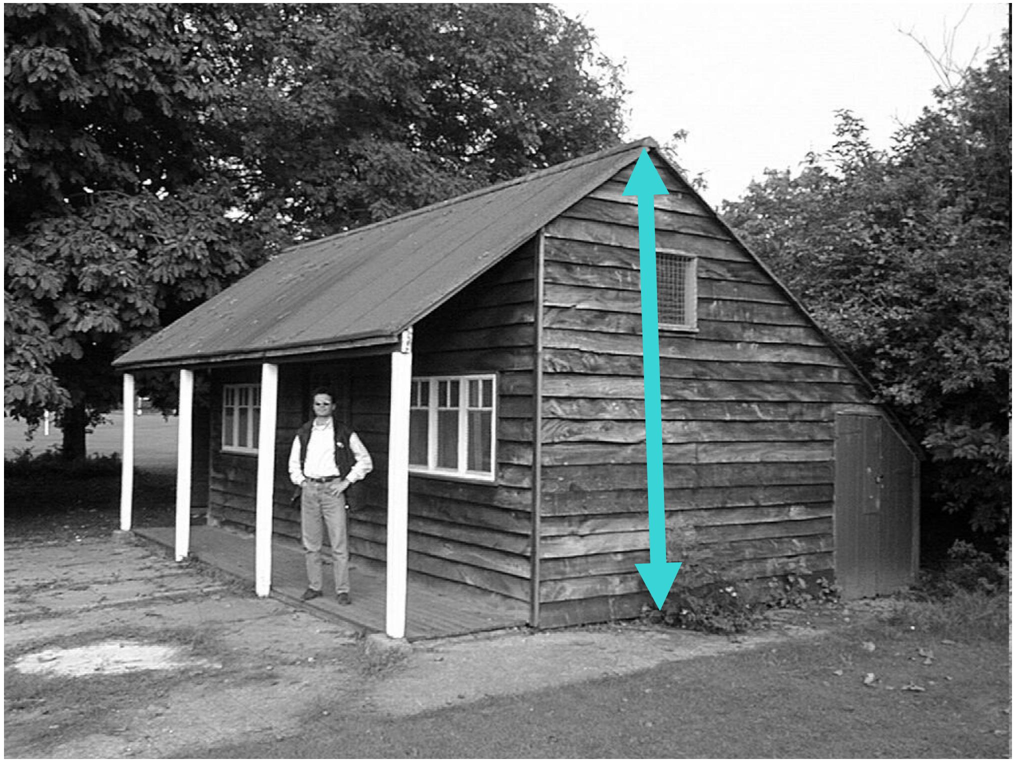


-The scene is determined by the images only up a similarity transformation (rotation, translation and scaling)

Courtesy of Silvio Savarese.

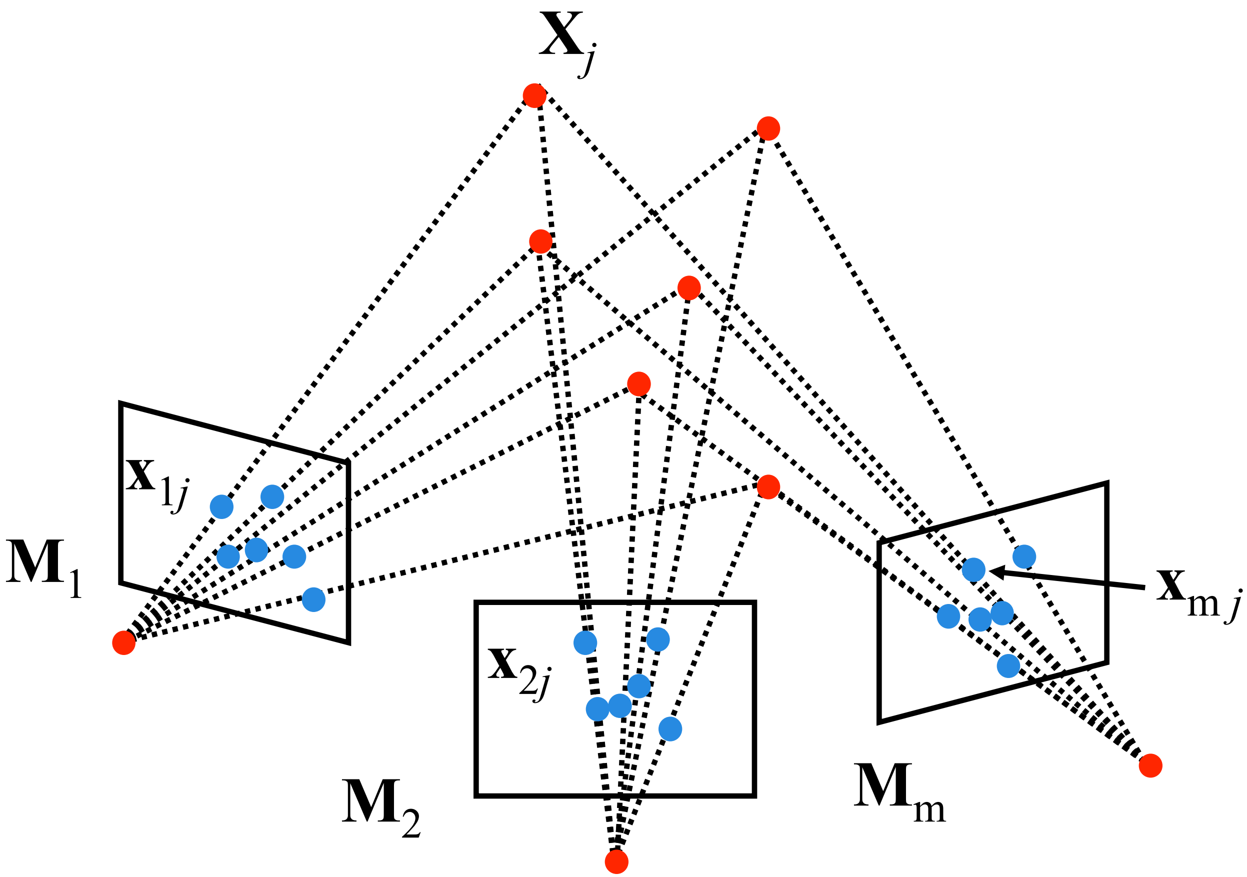
Structure from motion ambiguity

-Scale ambiguity: it is impossible based on the images alone to estimate the absolute scale of the scene (i.e. house height)



Courtesy of Silvio Savarese.

Structure from motion problem

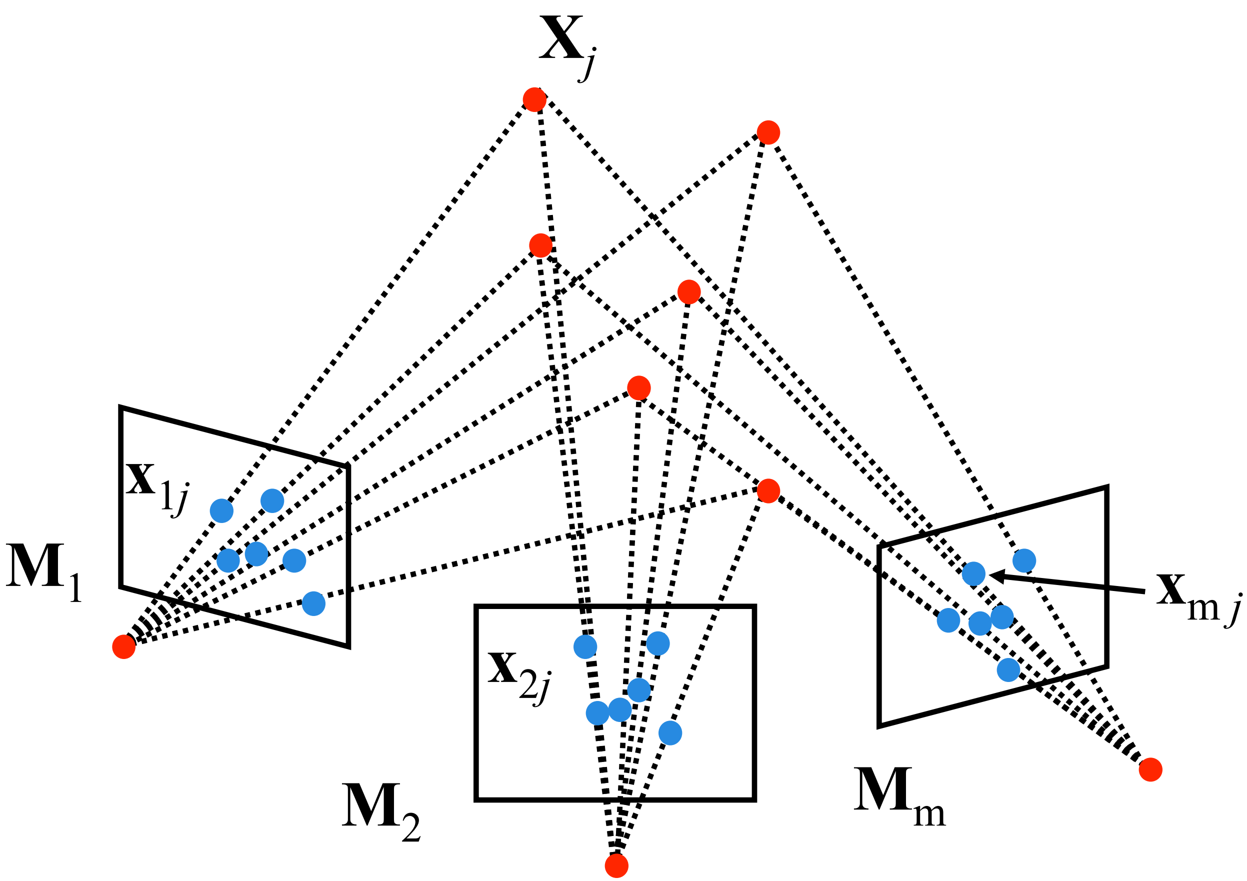


Given *m* images of *n* fixed 3D points

- $\mathbf{x}_{ij} = \mathbf{M}_i \mathbf{X}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$

Courtesy of Silvio Savarese.

Structure from motion problem



m cameras $M_1 \ldots M_m$

$$M_i = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & 1 \end{bmatrix}$$

Courtesy of Silvio Savarese.

The Projective Structure-from-Motion Problem

Given *m* images of *n* fixed points X_j we can write

$$x_{ij} = M_i X_j \quad \text{for} \quad i = 1, \ldots, m \quad \text{and} \quad j = 1, \ldots, n.$$

Problem: estimate the m 3×4 matrices M_i and the n positions X_j from the m×n correspondences x_{ij} .

- With no calibration info, cameras and points can only be recovered up to a 4x4 projective (15 parameters)

- Given two cameras, how many points are needed?
- How many equations and how many unknown?

2m × n equations in 11m+3n – 15 unknowns

So 7 points! [2x2x7 = 28; 11x2 + 3x7 – 15 = 28]

Courtesy of Silvio Savarese.

Structure-from-Motion Algorithms

- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- Bundle adjustment

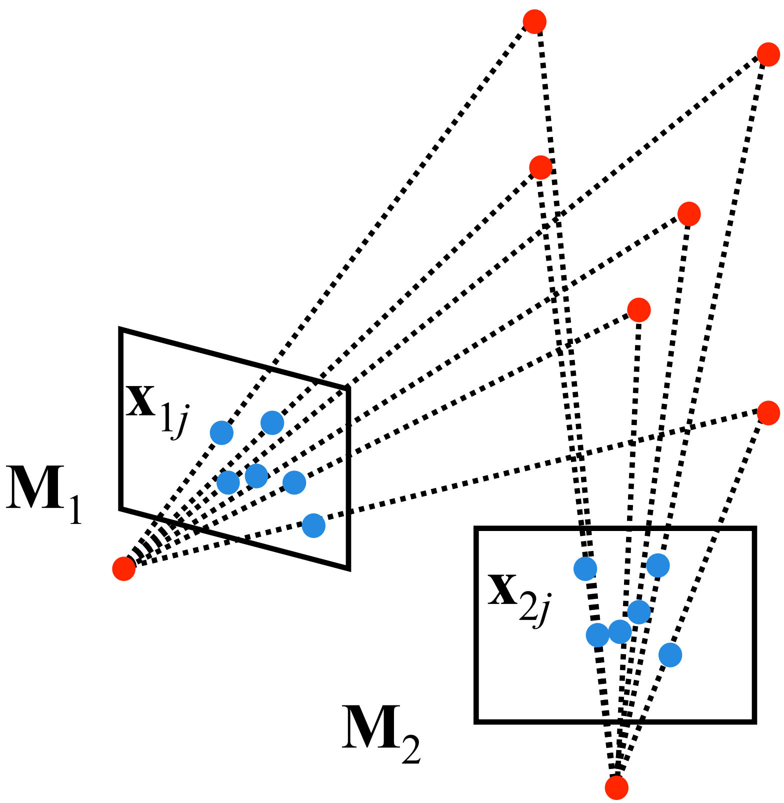
Courtesy of Silvio Savarese.

Algebraic approach (2-view case)

- Compute the fundamental matrix F from two views (eg. 8 point algorithm)
- Use F to estimate projective cameras
- Use these cameras to triangulate and estimate points in 3D

Courtesy of Silvio Savarese.

Algebraic approach (2-view case)



$$x_{ij} = M_i X_j$$

Apply a projective transformation H such that:

$$M_1 H^{-1} = \begin{bmatrix} I & 0 \end{bmatrix} \qquad M_2 H^{-1} = \begin{bmatrix} A & b \end{bmatrix}$$

Canonical perspective cameras

Courtesy of Silvio Savarese.

Algebraic approach (Fundamental matrix)

$$\tilde{\mathbf{X}} = \mathrm{H}\ \mathbf{X}$$

$$\mathbf{x} = \mathrm{M}_1\ \mathrm{H}^{-1}\ \tilde{\mathbf{X}} = [\mathbf{I} \mid \mathbf{0}]\tilde{\mathbf{X}} \qquad \mathbf{x}' = \mathrm{M}_2\ \mathrm{H}^{-1}\ \tilde{\mathbf{X}} = [\mathbf{A} \mid \mathbf{b}]\tilde{\mathbf{X}}$$

$$\mathbf{x}' = [\mathbf{A} \mid \mathbf{b}]\tilde{\mathbf{X}} = [\mathbf{A} \mid \mathbf{b}]\begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \\ 1 \end{bmatrix} = \mathbf{A}[\mathrm{I} \mid 0]\begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \\ 1 \end{bmatrix} + \mathbf{b} = \mathbf{A}\boxed{[\mathrm{I} \mid 0]\ \tilde{\mathbf{X}}}^{?} + \mathrm{b} \ = \mathrm{A}\mathbf{x} + \mathrm{b}$$

$$\mathbf{x}' \times \mathbf{b} = (\mathbf{A}\mathbf{x} + \mathbf{b}) \times \mathbf{b} = \mathbf{A}\mathbf{x} \times \mathbf{b}$$

$$(\mathbf{A}\mathbf{x} \times \mathbf{b}) \cdot \mathbf{x}' = (\mathbf{x}' \times \mathbf{b}) \cdot \mathbf{x}' = 0$$

$$\mathbf{x}'^{T} (\mathbf{A}\mathbf{x} \times \mathbf{b})^{T} = 0$$

$$\mathbf{x}'^{\mathrm{T}} \boxed{[\mathbf{b}_{\times}]\mathbf{A}}\ \mathbf{x} = 0$$

is this familiar?

$$\boxed{\mathbf{F} = [\mathbf{b}_{\times}]\mathbf{A}}$$

$$\boxed{\mathbf{x}'^{\mathrm{T}} \mathrm{F}\ \mathbf{x} = 0}$$

Courtesy of Silvio Savarese.

Cross product as matrix multiplication

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$$

Courtesy of Silvio Savarese.

Algebraic approach (Fundamental matrix)

$$\mathbf{x}'^{\mathrm{T}} \mathrm{F}\, \mathbf{x} = 0 \qquad\qquad \mathbf{F} = [\mathbf{b}_\times]\mathbf{A}$$

- Compute the fundamental matrix F from two views (eg. 8 point algorithm)

Can verify that :

$$\mathbf{F} \cdot \mathrm{b} = [\mathbf{b}_\times]\mathbf{A} \cdot \mathrm{b} = 0$$

→ Compute $\mathbf{b}$ as least sq. solution of $\mathbf{F}\,\mathbf{b} = 0$

det(F)=0; |b|=1

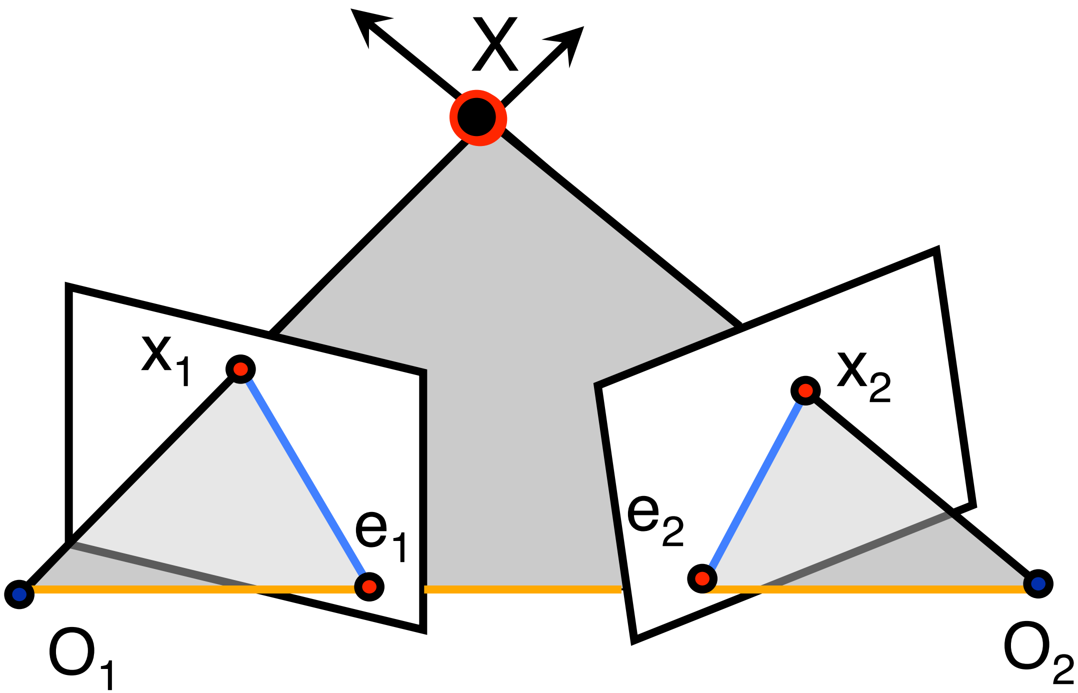
→

$$\mathbf{A} = [\mathbf{b}_\times]^{-1}\,\mathbf{F} = -[\mathbf{b}_\times]\,\mathbf{F}$$

Notice that b is an epipole

Courtesy of Silvio Savarese.

Epipolar Constraint [from earlier lecture]



- $F\,x_2$ is the epipolar line associated with x_2 ($l_1 = F\,x_2$)
- $F^T x_1$ is the epipolar line associated with x_1 ($l_2 = F^T x_1$)
- F is singular (rank two)
- $F\,e_2 = 0$ and $F^T e_1 = 0$
- F is 3x3 matrix; 7 DOF

Courtesy of Silvio Savarese.

Algebraic approach (Fundamental matrix)

$$\mathbf{x}'^{T} F\, \mathbf{x} = 0 \qquad\qquad \mathbf{F} = [\mathbf{b}_\times]\mathbf{A}$$

- Compute the fundamental matrix F from two views (eg. 8 point algorithm)

Can verify that:

$$\mathbf{F} \cdot b = [\mathbf{b}_\times]\mathbf{A} \cdot b = 0$$

→ Compute $\mathbf{b}$ as least sq. solution of $\mathbf{F}\,\mathbf{b} = 0$

det(F)=0; |b|=1

→

$$\mathbf{A} = [\mathbf{b}_\times]^{-1}\,\mathbf{F} = -[\mathbf{b}_\times]\,\mathbf{F}$$

Notice that b is an epipole

$$M^{p}{}_{1} = \begin{bmatrix} I & 0 \end{bmatrix} \qquad\qquad M^{p}_{2} = \begin{bmatrix} -[\mathbf{e}_x]\mathbf{F} & \mathbf{e} \end{bmatrix}$$

Perspective cameras are known

HZ, page 254
PF, page 288

Structure-from-Motion Algorithms

- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- Bundle adjustment

Courtesy of Silvio Savarese.

Projective factorization

$$\mathbf{D} = \begin{bmatrix} z_{11}\mathbf{x}_{11} & z_{12}\mathbf{x}_{12} & \cdots & z_{1n}\mathbf{x}_{1n} \\ z_{21}\mathbf{x}_{21} & z_{22}\mathbf{x}_{22} & \cdots & z_{2n}\mathbf{x}_{2n} \\ & & \ddots & \\ z_{m1}\mathbf{x}_{m1} & z_{m2}\mathbf{x}_{m2} & \cdots & z_{mn}\mathbf{x}_{mn} \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}$$

cameras ($3m \times 4$)

points ($4 \times n$)

$\mathbf{D} = \mathbf{MS}$ has rank 4

Courtesy of Silvio Savarese.

A factorization method - (affine case; last lecture)

- Let's create a 2m × n data (measurement) matrix:

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}$$

S

points ($3 \times n$)

M

($2m \times$ n)

cameras ($2m \times 3$)

The measurement matrix **D = M S** has rank 3

(it's a product of a 2mx3 matrix and 3xn matrix)

Courtesy of Silvio Savarese.

Projective factorization

$$\mathbf{D} = \begin{bmatrix} z_{11}\mathbf{x}_{11} & z_{12}\mathbf{x}_{12} & \cdots & z_{1n}\mathbf{x}_{1n} \\ z_{21}\mathbf{x}_{21} & z_{22}\mathbf{x}_{22} & \cdots & z_{2n}\mathbf{x}_{2n} \\ & & \ddots & \\ z_{m1}\mathbf{x}_{m1} & z_{m2}\mathbf{x}_{m2} & \cdots & z_{mn}\mathbf{x}_{mn} \end{bmatrix} = \begin{bmatrix} \mathrm{M}_1 \\ \mathrm{M}_2 \\ \vdots \\ \mathrm{M}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}$$

cameras ($3m \times 4$)

points ($4 \times n$)

$\mathbf{D} = \mathbf{MS}$ has rank 4

- If we knew the depths *z*, we could factorize **D** to estimate **M** and **S**
- If we knew **M** and **S**, we could solve for *z*
- Solution: iterative approach (alternate between above two steps)

Structure-from-Motion Algorithms

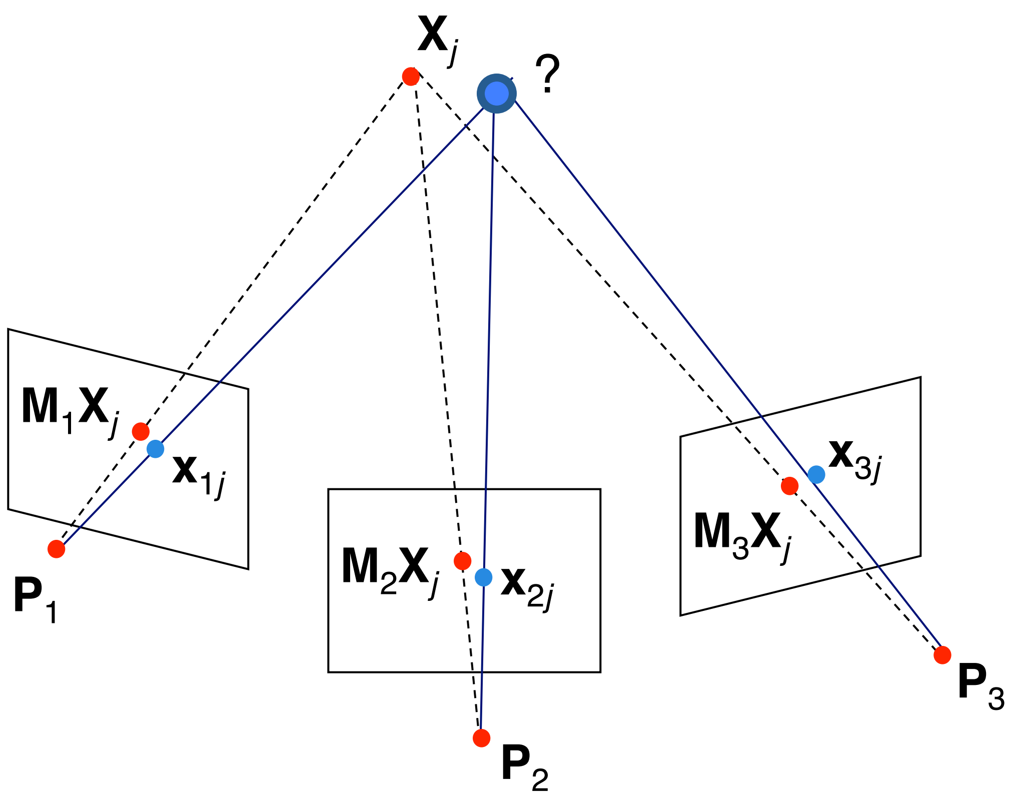
- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- Bundle adjustment

Courtesy of Silvio Savarese.

Bundle adjustment

- Non-linear method for refining structure and motion
- Minimizing re-projection error

$$E(M, \mathbf{X}) = \sum_{i=1}^{m} \sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, M_i \mathbf{X}_j\right)^2$$



Courtesy of Silvio Savarese.

Bundle adjustment

- Non-linear method for refining structure and motion
- Minimizing re-projection error

$$E(M, \mathbf{X}) = \sum_{i=1}^{m} \sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, M_i \mathbf{X}_j\right)^2$$

- Advantages
  - Handle large number of views
  - Handle missing data

- Limitations
  - Large minimization problem (parameters grow with number of views)
  - requires good initial condition

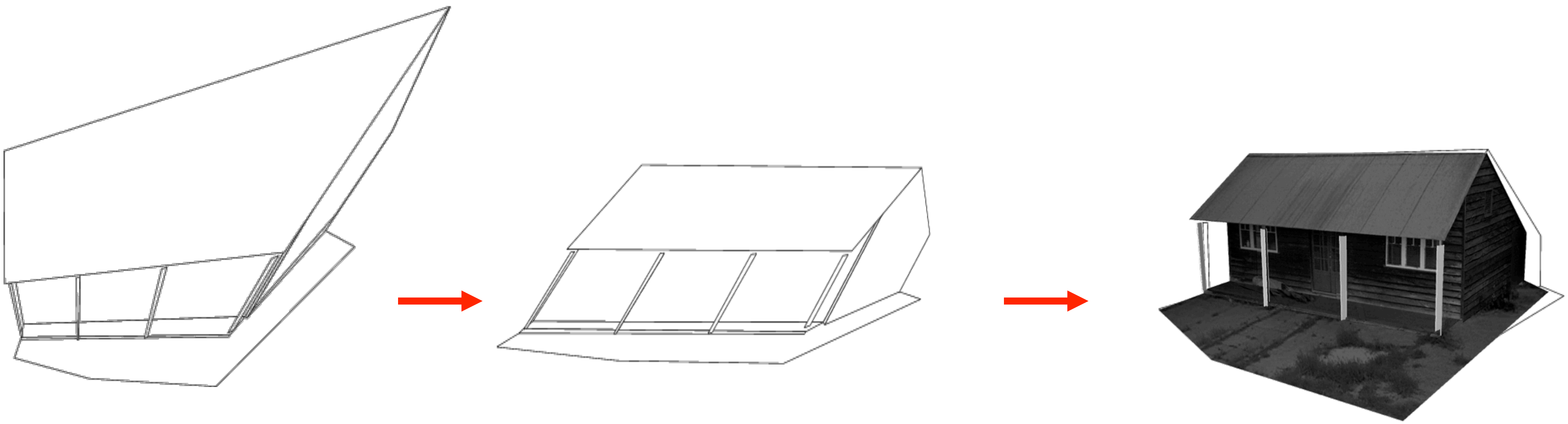
Used as the final step of SFM

Courtesy of Silvio Savarese.

Removing the ambiguities: the Stratified reconstruction

- up grade reconstruction from perspective to affine
  [by measuring the plane at infinity]

- up grade reconstruction from affine to metric
  [by measuring the absolute conic]



Recovering the metric reconstruction
from the perspective one is called self-calibration

Courtesy of Silvio Savarese.

Self-calibration

Process of determining intrinsic camera parameters directly from un-calibrated images

Suppose we have a projective reconstruction $\{M_i, X_j\}$

GOAL: find a rectifying (non-singular) homography H such that

$$\{M_i\,H, H^{-1}X_j\}$$ is a metric reconstruction

$$\overline{M}_i \qquad \overline{X}_j$$

$$\overline{M}_i = M_i\,H \qquad i = 1 \cdots m \qquad \overline{M}_i = K_i[R_i \quad T_i]$$

If world ref. system = camera 1 ref. system: $\overline{M}_1 = K_1[I \quad 0]$

If the perspective camera is canonical: $M_1 = [I \quad 0]$

Courtesy of Silvio Savarese.

Self-calibration

$$\overline{M}_i = M_i \, H$$

$$\downarrow$$

$$[K_1 \quad 0] = [I \quad 0] \, H$$

$$\downarrow$$

$$A = K_1$$

$$t = 0$$

$$\begin{cases} \overline{M}_1 = K_1 [I \quad 0] \\ M_1 = [I \quad 0] \\ H = \begin{bmatrix} A & t \\ v & k \end{bmatrix} \end{cases}$$

We can set k=1
(this fixes the scale of the reconstruction)

$$H = \begin{bmatrix} K_1 & 0 \\ v & 1 \end{bmatrix} \xrightarrow{\text{Planes at infinity}} H = \begin{bmatrix} K_1 & 0 \\ -p^T K_1 & 1 \end{bmatrix}$$

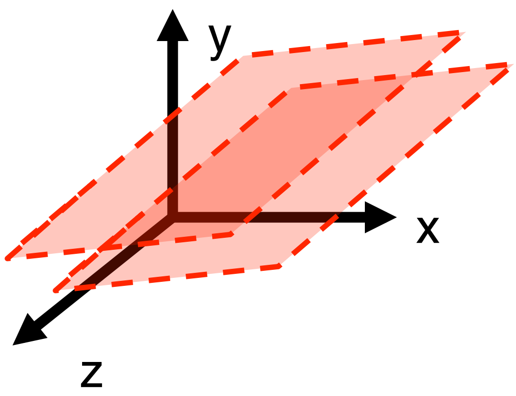
$$\pi_\infty = H^{-1} \, \Pi_\infty = \begin{bmatrix} p \\ 1 \end{bmatrix}$$

See appendix

Courtesy of Silvio Savarese.

Planes at infinity (lecture 5)

$$\Pi_\infty = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



In the metric (Euclidean) world coordinates

2 planes are parallel iff their intersections is a line that belongs to Π_∞

The projective transformation of a plane at infinity can be expressed as

$$\boldsymbol{\pi}_\infty = H^{-1}\,\Pi_\infty = \begin{bmatrix} p \\ 1 \end{bmatrix}$$

Self-calibration

$$\boldsymbol{\pi}_\infty = \begin{bmatrix} p \\ 1 \end{bmatrix} = H^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H = \begin{bmatrix} K_1 & 0 \\ v & 1 \end{bmatrix}$$

$$= \begin{bmatrix} K_1^{-T} & -K_1^{-T} v \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -K_1^{-T} v \\ 1 \end{bmatrix} \rightarrow v = -p^T K_1$$

Courtesy of Silvio Savarese.

Self-calibration

GOAL: find a rectifying homography H such that

$$\{M_i, X_j\} \rightarrow \{M_i\, H, H^{-1} X_j\}$$ is a metric reconstruction

$$H = \begin{bmatrix} K_1 & 0 \\ -p^T K_1 & 1 \end{bmatrix}$$

$$K = \begin{bmatrix} \alpha & -\alpha \cot\theta & u_o \\ 0 & \frac{\beta}{\sin\theta} & v_o \\ 0 & 0 & 1 \end{bmatrix}$$

K_1 = calibration matrix of first camera

5 unknowns

$\pi_\infty = [p \;\; 1]^T$ = plane at infinity in the projective reconstruction

3 unknowns

Courtesy of Silvio Savarese.

Self-calibration basic equation

$$\begin{cases} M_i = [A_i \quad a_i] & = \text{perspective reconstruction of the camera (known)} \\ \overline{M}_i = K_i[R_i \quad T_i] & = \text{metric reconstruction of the camera (unknown)} \\ H = \begin{bmatrix} K_1 & 0 \\ -p^T K_1 & 1 \end{bmatrix} & = \text{rectifying homography (unknown)} \end{cases}$$

$$\overline{M}_i = M_i H \qquad i = 2 \cdots m$$

$$[K_i R_i \quad T_i] = [A_i \quad a_i] \begin{bmatrix} K_1 & 0 \\ -p^T K_1 & 1 \end{bmatrix} = [A_i K_1 - a_i p^T K_1 \quad a_i]$$

$$K_i R_i = (A_i - a_i p^T) K_1 \rightarrow R_i = K_i^{-1} (A_i - a_i p^T) K_1$$

Courtesy of Silvio Savarese.

Self-calibration basic equation

$$\begin{cases} R_i = K_i^{-1}\left(A_i - a_i p^T\right)K_1 \\ R_i^T = K_1^T\left(A_i - a_i p^T\right)^T K_i^{-T} \end{cases}$$

$$R_i \; R_i^T = I$$

$$K_i^{-1}\left(A_i - a_i p^T\right)K_1 \; K_1^T\left(A_i - a_i p^T\right)^T K_i^{-T} = I$$

$$\left(A_i - a_i p^T\right)K_1 K_1^T\left(A_i - a_i p^T\right)^T = \boxed{K_i \; K_i^T} \leftarrow ?$$

Courtesy of Silvio Savarese.

Absolute conic Ω_∞ is a $C \in \Pi_\infty$

Any $x \in \Omega_\infty$ satisfies:

$$x^T \Omega_\infty x = 0 \qquad \Omega_\infty = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 0 \end{bmatrix} \qquad \begin{cases} x_1^2 + x_2^2 + x_3^2 = 0 \\ x_4 = 0 \end{cases}$$

Projective transformation of Ω_∞

$$\boldsymbol{\omega} = (K^T K)^{-1}$$

$$\boldsymbol{\omega}^* = K K^T$$

Dual image of the absolute conic

Properties of ω

$$\omega = (K^T K)^{-1}$$

- It is not function of R, T

- symmetric (5 unknowns) $\quad \omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_4 \\ \omega_2 & \omega_3 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{bmatrix}$

Self-calibration basic equation

$$\left(A_i - a_i p^T\right) K_1 K_1^T \left(A_i - a_i p^T\right)^T = K_i \; K_i^T$$

$$\boxed{\left(A_i - a_i p^T\right) \boldsymbol{\omega}_1^* \left(A_i - a_i p^T\right)^T = \boldsymbol{\omega}_i^*} \quad i=2\ldots m$$

[A_i and a_i are known]

How many unknowns?
- 3 from $\mathbf{p}$
- 5 from ω_i [per view]

How many equations? 5 independent equations [per view]

Art of self-calibration:

use constraints on ω (K) to generate enough equations on the unknowns

Self-calibration – identical Ks

$$\left(A_i - a_i p^T\right)\omega_1^*\left(A_i - a_i p^T\right)^T = \omega_i^*$$

$$\left(A_i - a_i p^T\right)\omega^*\left(A_i - a_i p^T\right)^T = \omega^*$$

- For m views, 5(m-1) constraints
- Number of unknowns: 8

→ m>=3 provides enough constraints

To solve the self-calibration problem with identical cameras we need at least 3 views

Properties of ω

$$\omega = (K^T K)^{-1}$$

1. $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_4 \\ \omega_2 & \omega_3 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{bmatrix}$

2. $\omega_2 = 0$ zero-skew

3. $\omega_2 = 0$
   $\omega_1 = \omega_3$
   square pixel

4. $\omega_4 = \omega_5 = 0$
   zero-offset

Courtesy of Silvio Savarese.

Self-calibration – other constraints

$$\left(A_i - a_i p^T\right)\omega_1^*\left(A_i - a_i p^T\right)^T = \omega_i^*$$

- zero-offset $\omega_4 = \omega_5 = 0$ → 2 m linear constraints
- zero-skew $\omega_2 = 0$ → m linear constraints

etc…

Courtesy of Silvio Savarese.

Self-calibration - summary

Condition	N. Views
•Constant internal parameters	3
•Aspect ratio and skew known •Focal length and offset vary	4
•Aspect ratio and skew constant •Focal length and offset vary	5
•skew =0, all other parameters vary	8

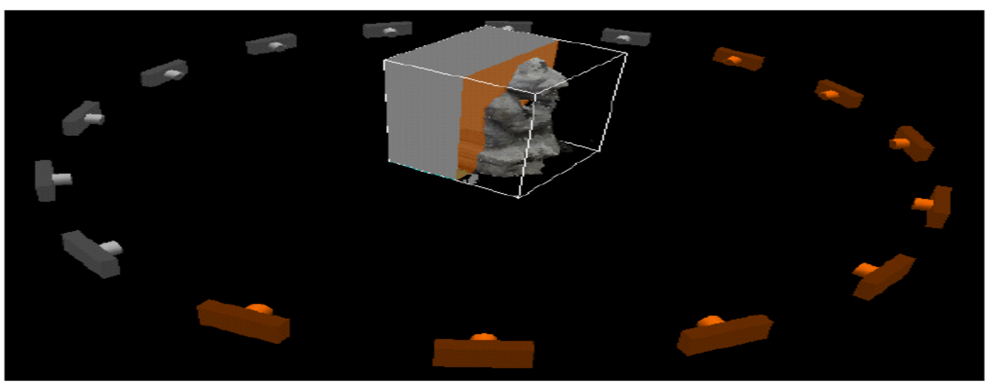
Issue: the larger is the number of view,
the harder is the correspondence problem

Bundle adjustment helps!

Self-calibration - summary

Constraints on camera motion can be incorporated

- Linearly translating camera



- Single axis of rotation: turntable motion

Courtesy of Silvio Savarese.

SFM problem - summary

1. Estimate structure and motion up perspective transformation
    1. Algebraic
    2. factorization method
    3. bundle adjustment

2. Convert from perspective to metric (self-calibration)

3. Bundle adjustment

** or **

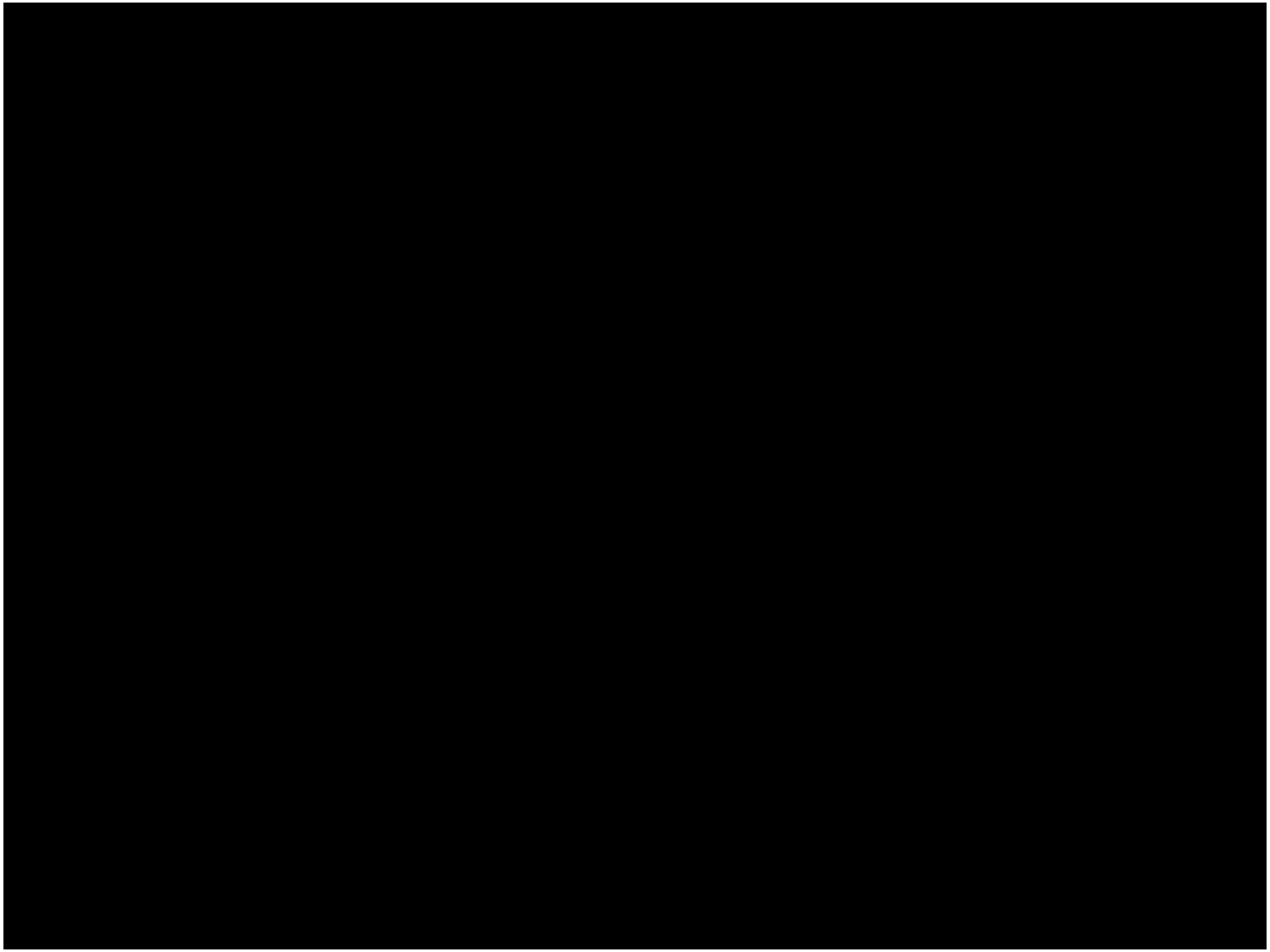
1. Bundle adjustment with self-calibration constraints

Courtesy of Silvio Savarese.

Correspondences

- Can refine feature matching after a structure and motion estimate has been produced
  - decide which ones obey the *epipolar geometry*
  - decide which ones are *geometrically consistent*
  - (optional) iterate between correspondences and SfM estimates using MCMC [Dellaert et al., Machine Learning 2003]

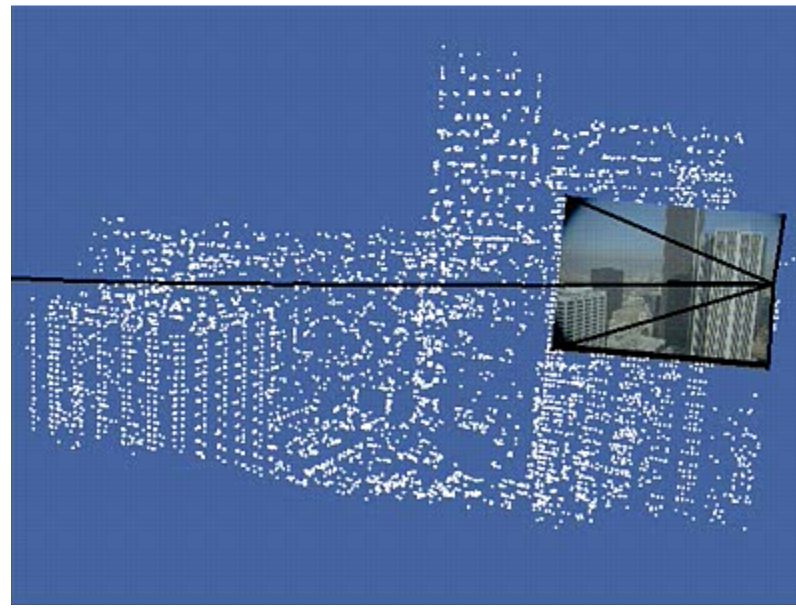
SFM Summed Up...



From: http://www.youtube.com/watch?v=i7ierVkXYa8

Applications

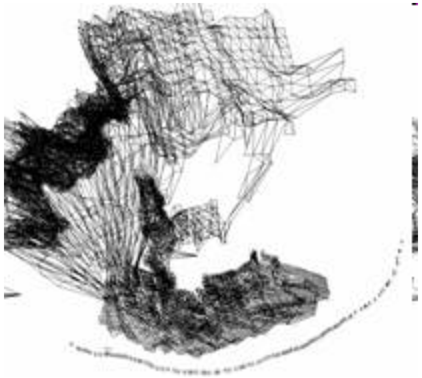
Courtesy of Oxford **Visual Geometry Group**



Courtesy of Silvio Savarese.

Applications

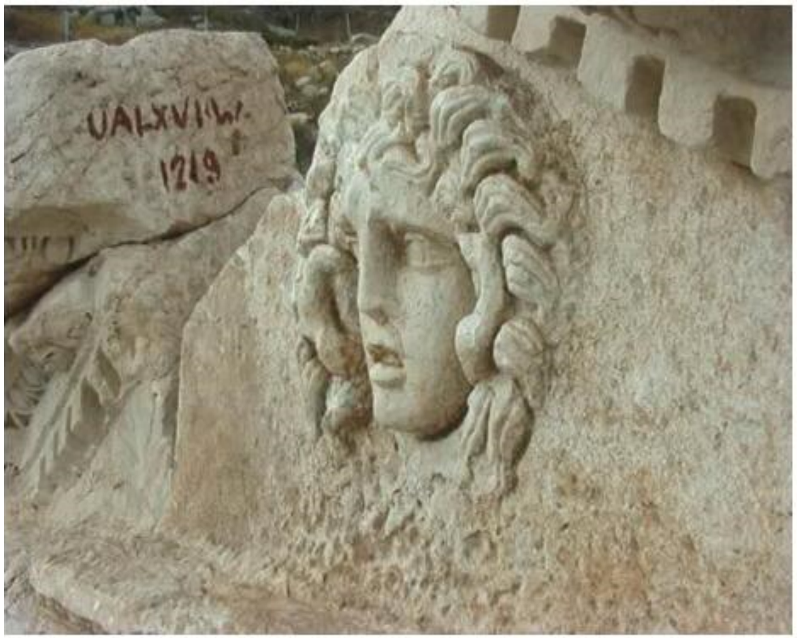
D. Nistér, PhD thesis ‘01

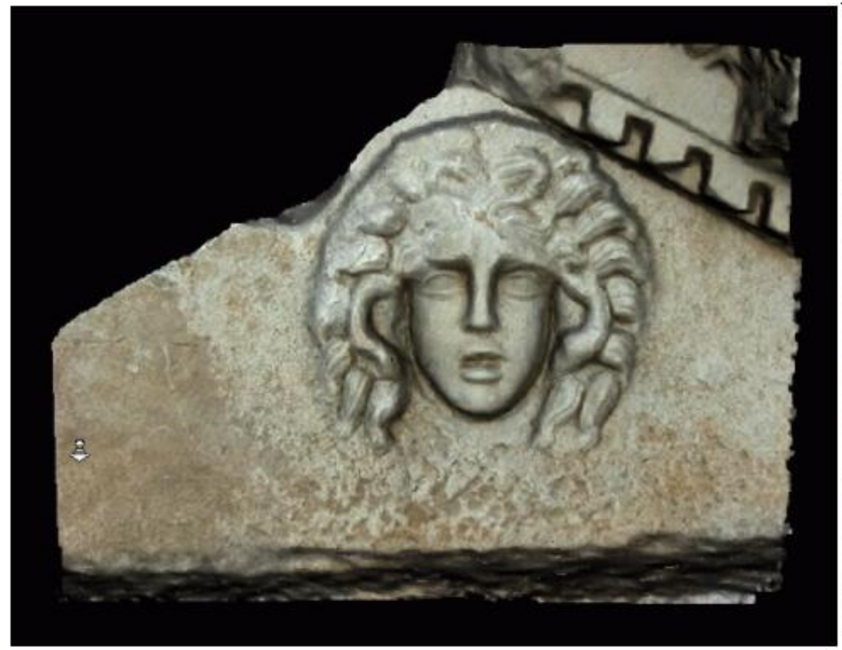


Courtesy of Silvio Savarese.

Applications

M. Pollefeys et al 98---

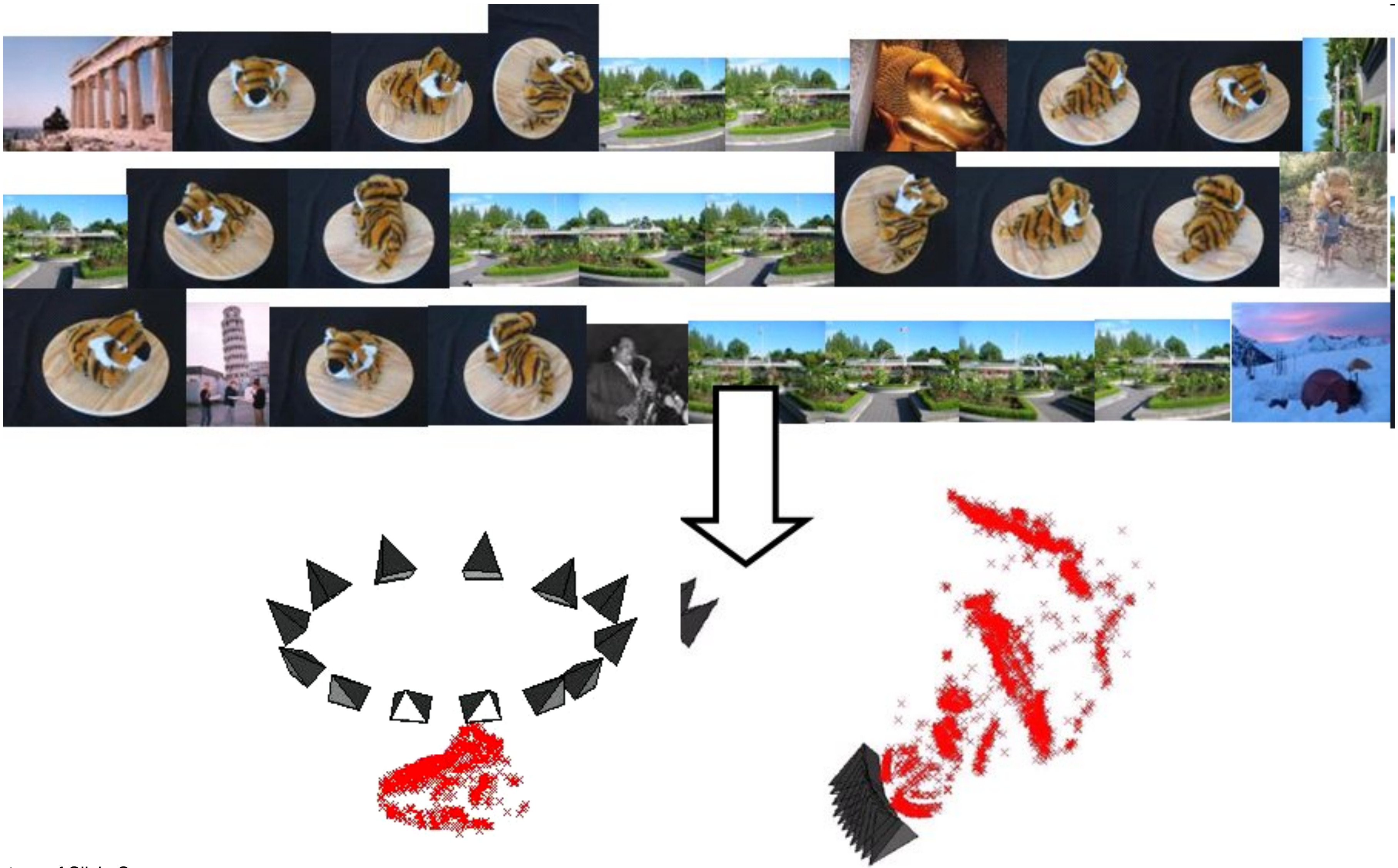




Courtesy of Silvio Savarese.

Applications

M. Brown and D. G. Lowe. Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets. *(3DIM2005)*



Courtesy of Silvio Savarese.

Photo synth

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

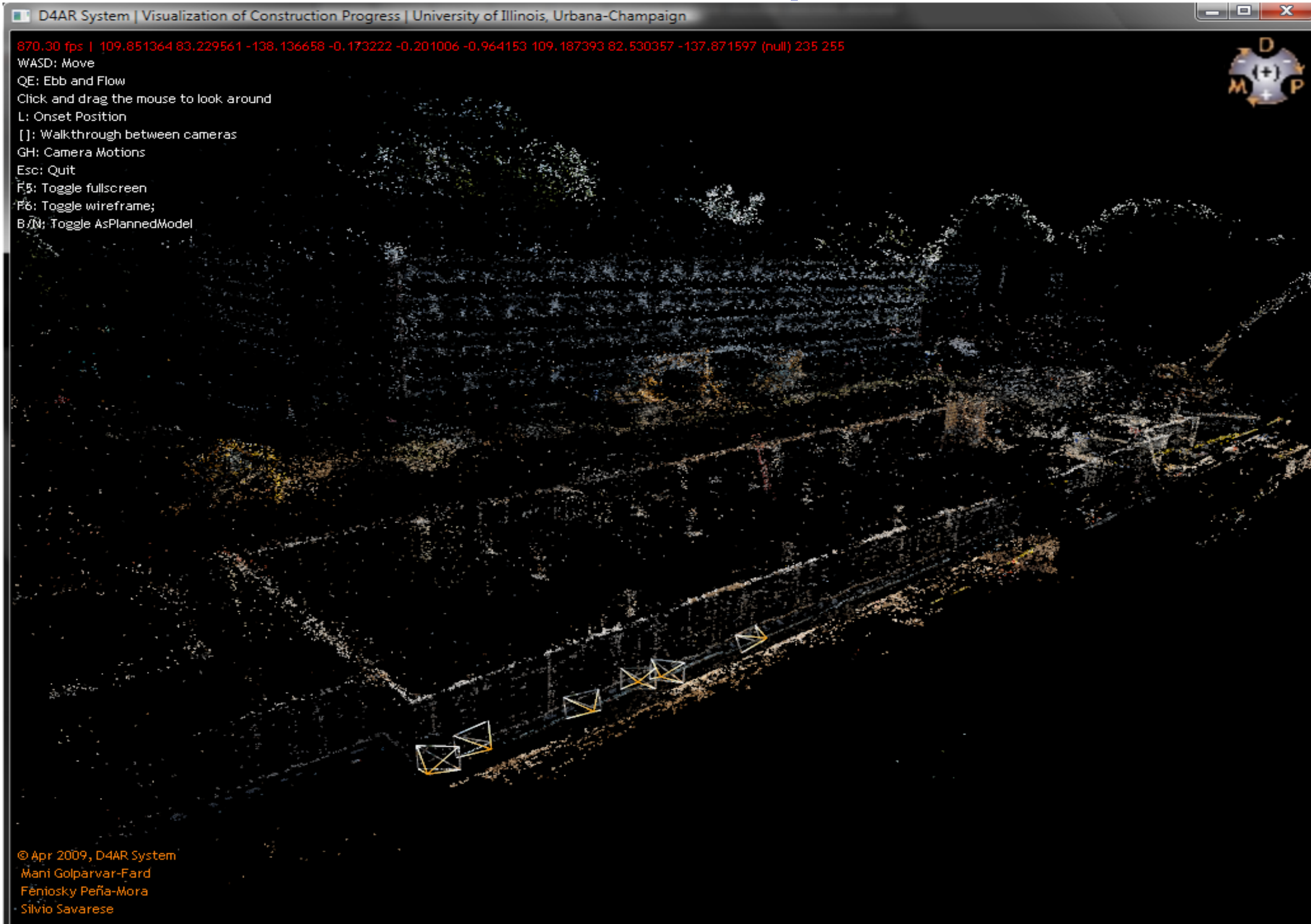
Courtesy of Silvio Savarese.

Incremental reconstruction of construction sites

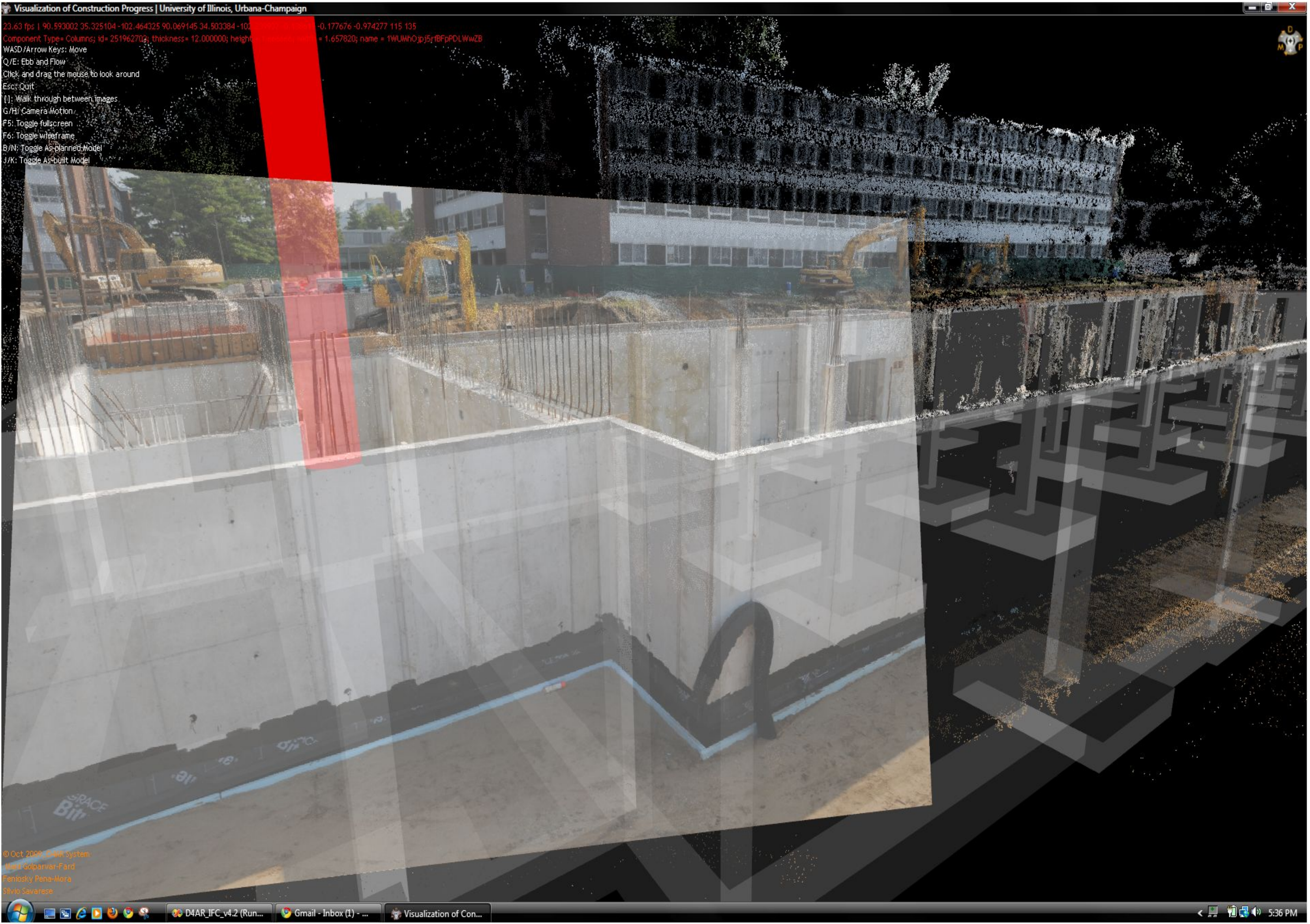
Initial pair – 2168 & Complete Set 62,323 points, 160 images

Courtesy of Silvio Savarese.

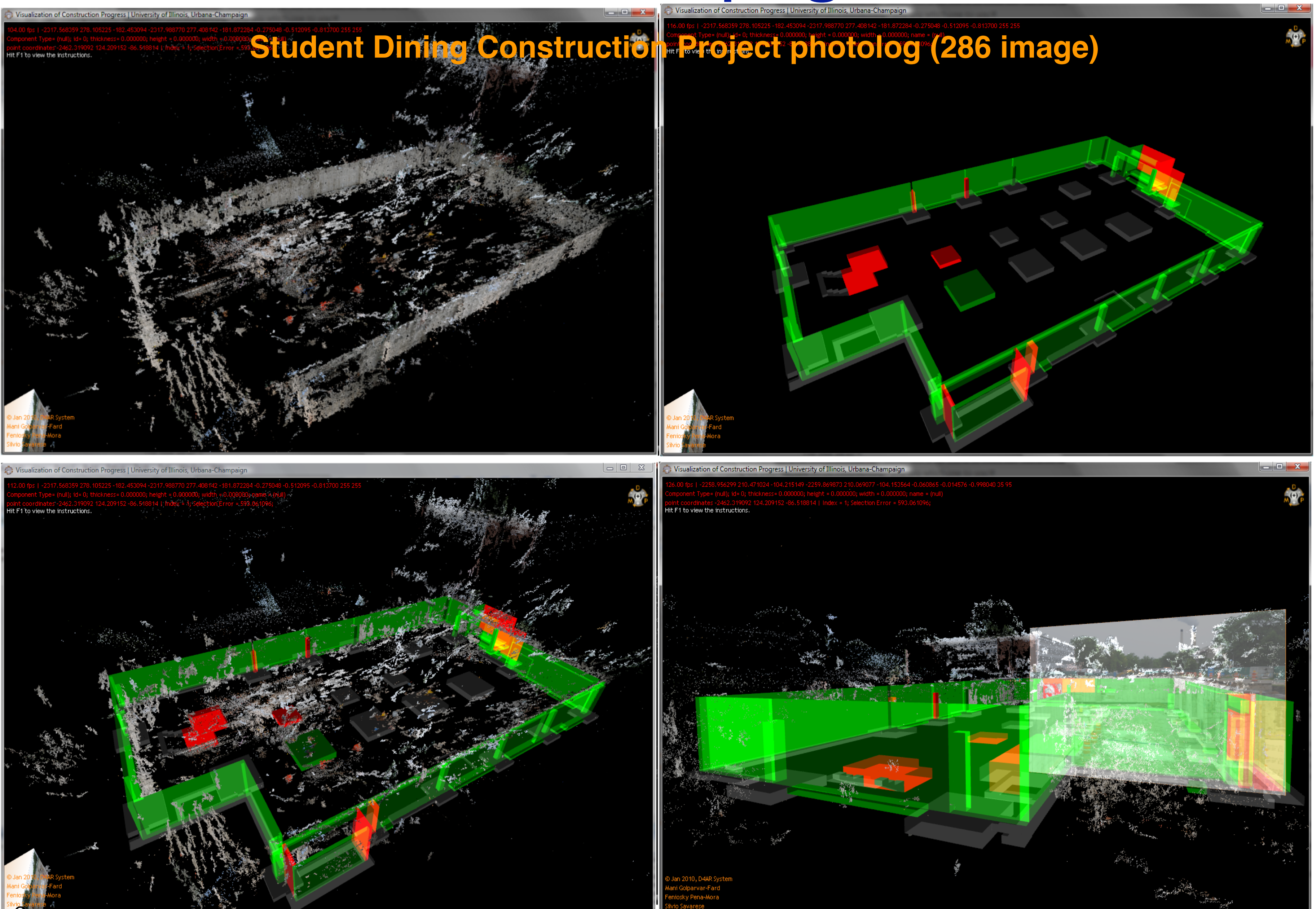
Reconstructed scene + Site photos



Dense reconstruction results for RH 160 dataset

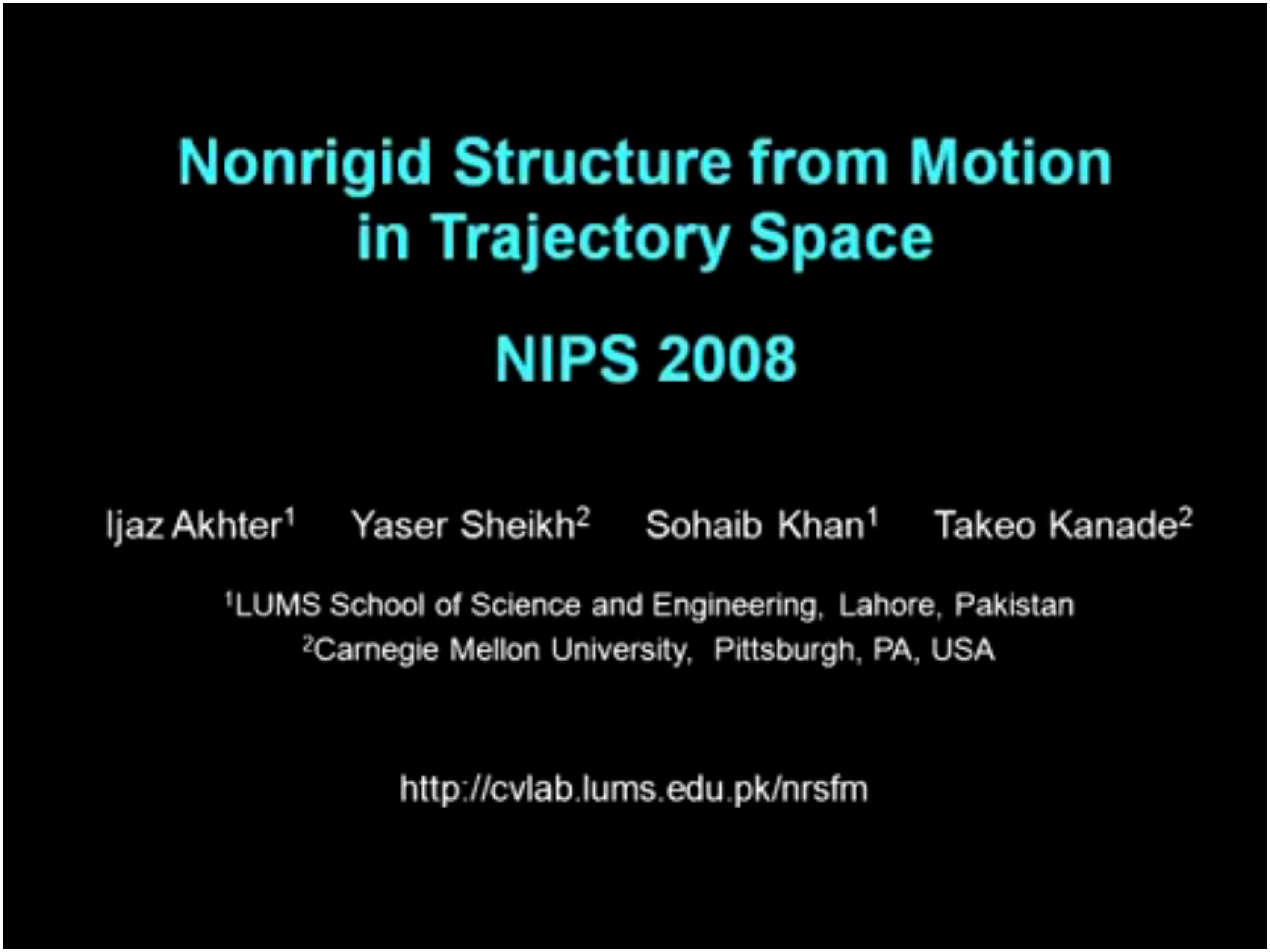


The results of automated progress detection



Courtesy of Silvio Savarese.

Non-rigid SFM…an example



Next Lecture: Introduction to Visual Recognition

- Readings: FP 15.1, 18.1