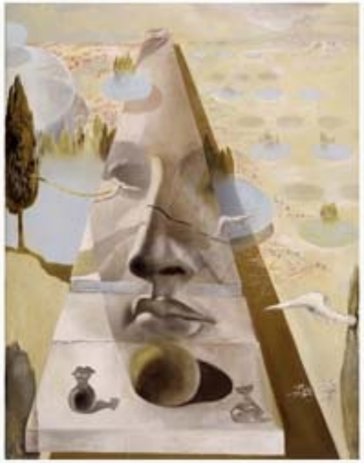


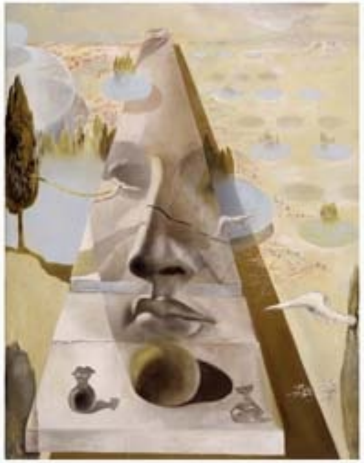
EECS 442 – Computer vision

Announcements

-Project proposal are due two weeks from now (10/20)
-HW2 Due (10/13)
-Project progress reports are due (11/1)

Project Proposal

Project Proposal Format
- max 4 pages;
- 3 sections:
  * title and authors
  * sec 1. intro: problem you want to solve and why
  * sec 2. technical part: how do you propose to solve it?
  * sec 3. milestones (dates and sub-goals)
  * references
- final format: pdf, please!



EECS 442 – Computer vision

Multiple view geometry

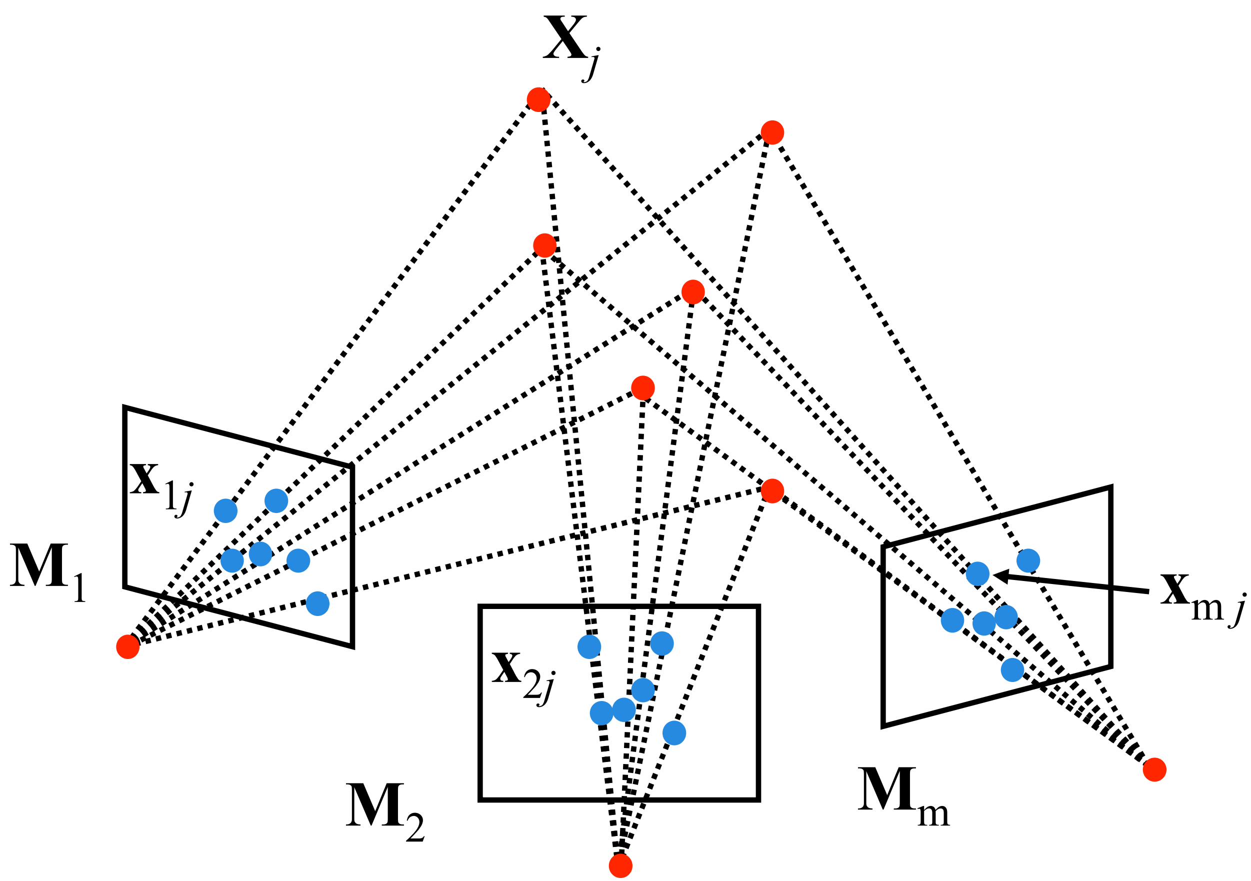
Perspective Structure from Motion

- Perspective structure from motion problem
- Ambiguities
- Algebraic methods
- Factorization methods
- Bundle adjustment
- Self-calibration

Reading: [HZ] Chapters: 10,18,19
[FP] Chapter: 13

Some slides of this lectures are courtesy of prof. S. Lazebnik

Structure from motion problem

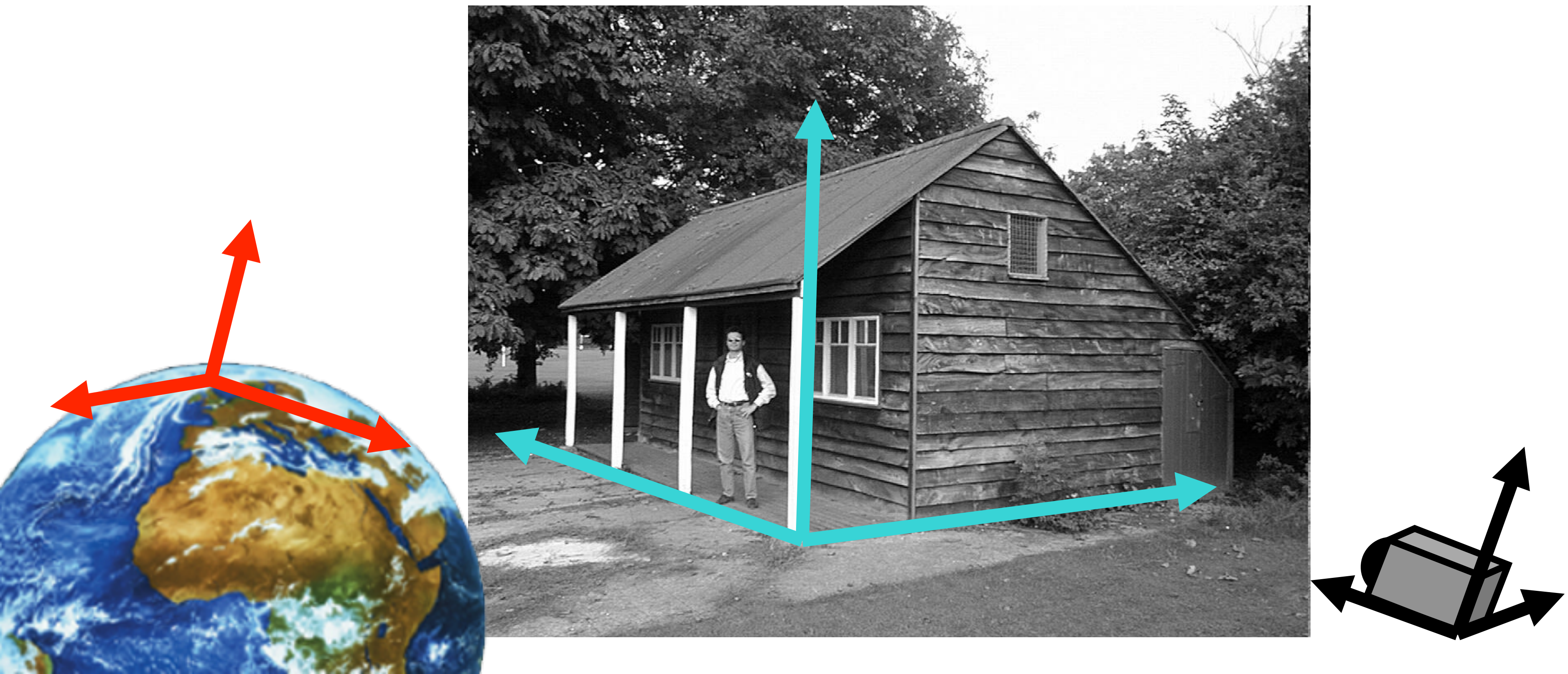


From the m×n correspondences $\mathbf{x}_{ij}$, estimate:

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

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



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

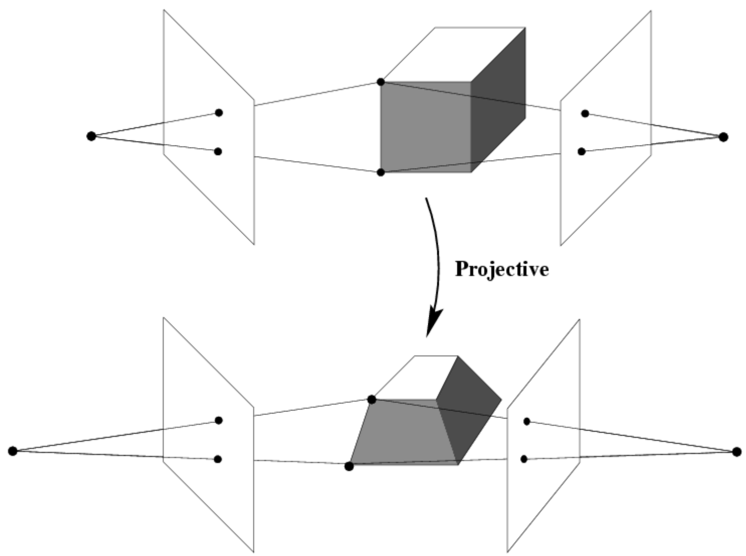
$$x_j = M_i \underbrace{\boxed{X_j}}_{\boxed{H_s X_j}} \qquad \underbrace{\boxed{M_i}}_{\boxed{M_i H_s^{-1}}} = K_i \begin{bmatrix} R_i & T_i \end{bmatrix}$$

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

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

The $\boxed{\text{calibration matrix}}$ has not changed!

Structure from motion ambiguity



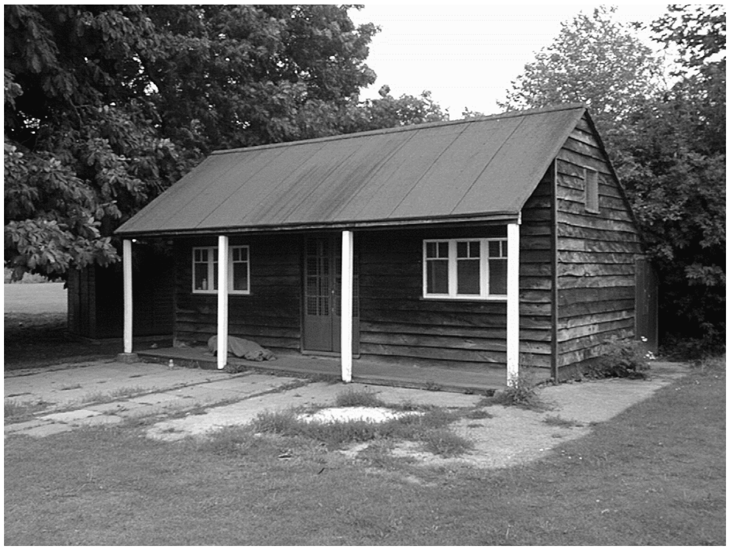
- In the general case (nothing is known) the ambiguity is expressed by an arbitrary affine or projective transformation

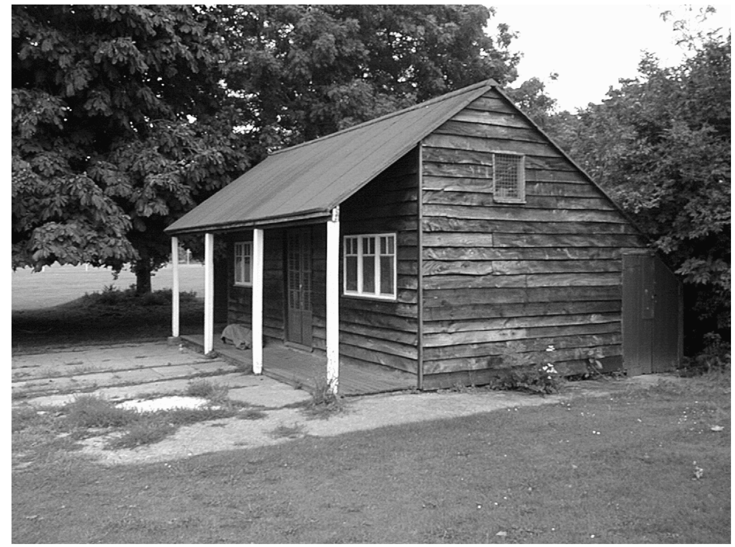
$$x_j = M_i \boxed{X_j} \qquad \boxed{M_i} = K_i \begin{bmatrix} R_i & T_i \end{bmatrix}$$

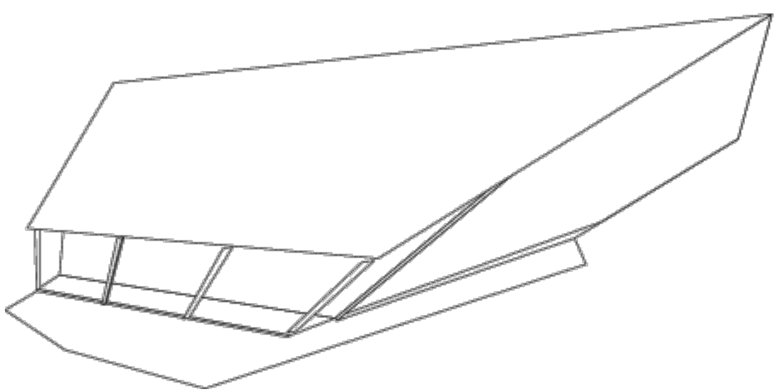
$$\boxed{H\, X_j} \qquad \boxed{M_j\, H^{-1}}$$

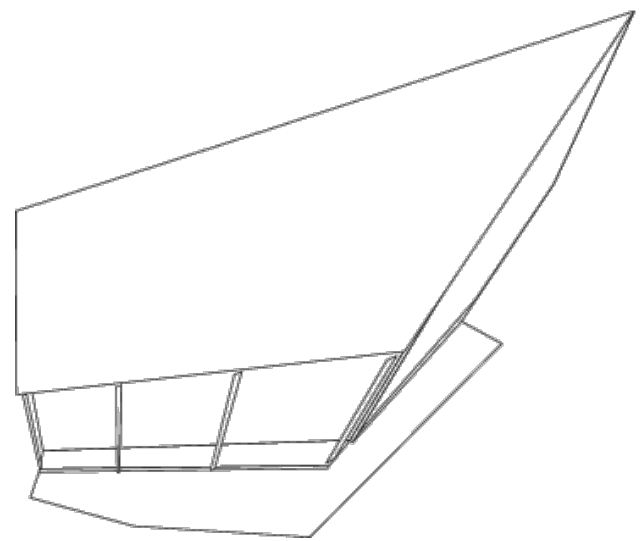
$$x_j = M_i\, X_j = \left(M_i\, H^{-1}\right)\left(H\, X_j\right)$$

Projective ambiguity

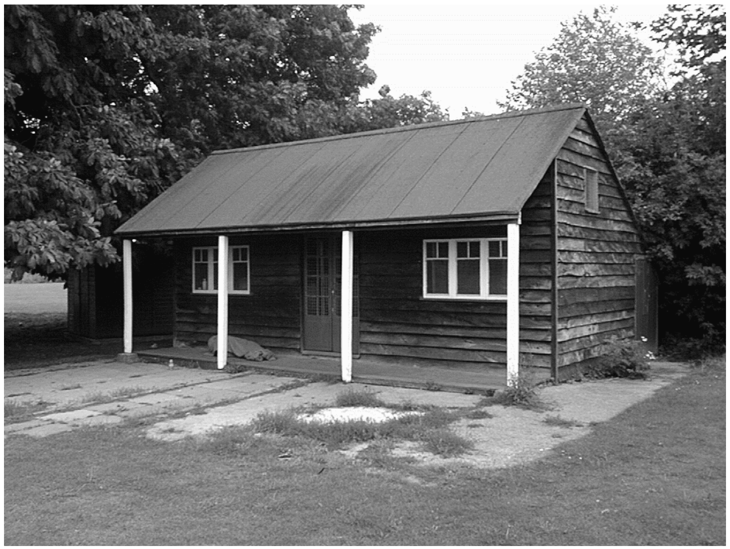
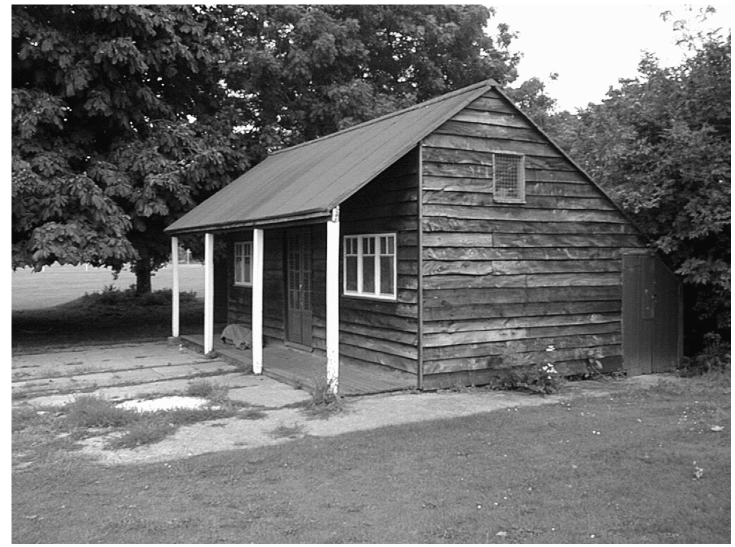
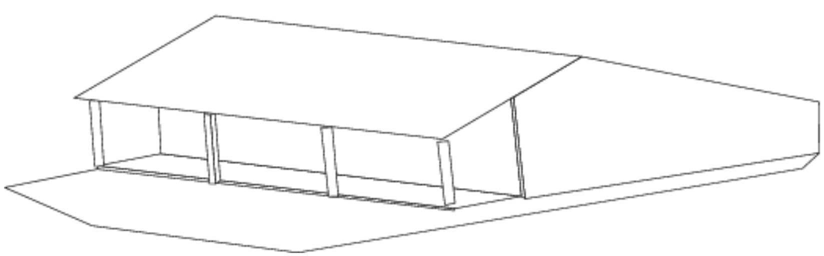
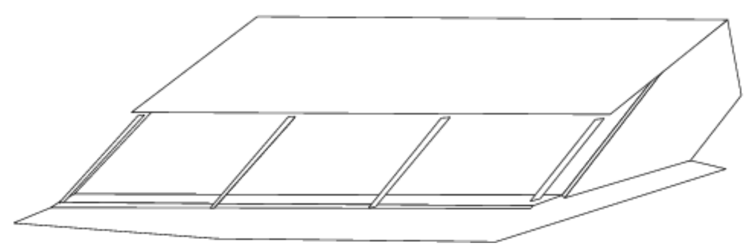








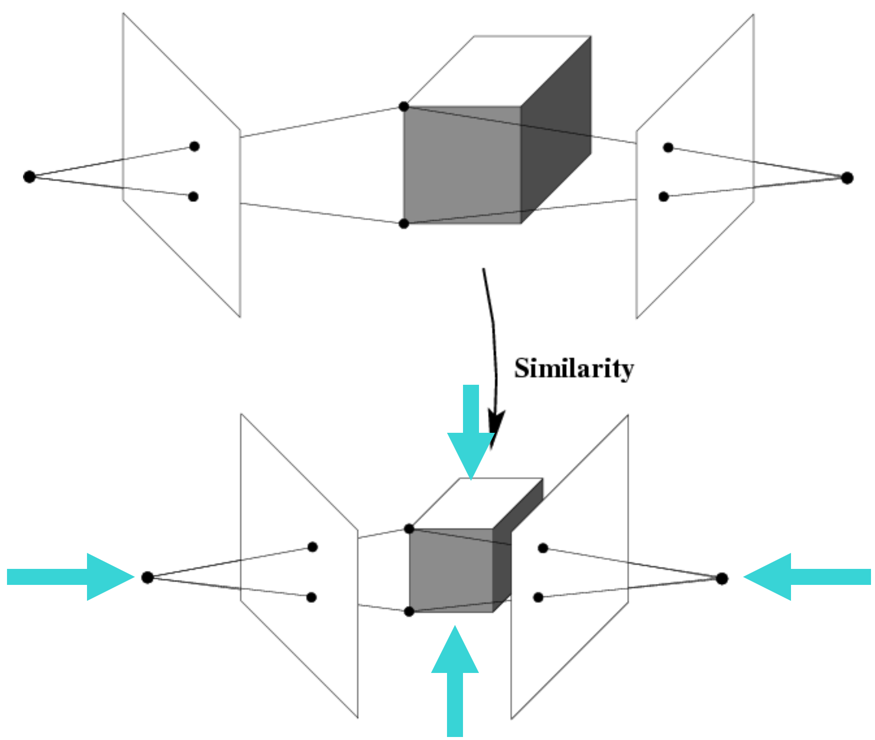
Affine ambiguity

Structure from motion ambiguity

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

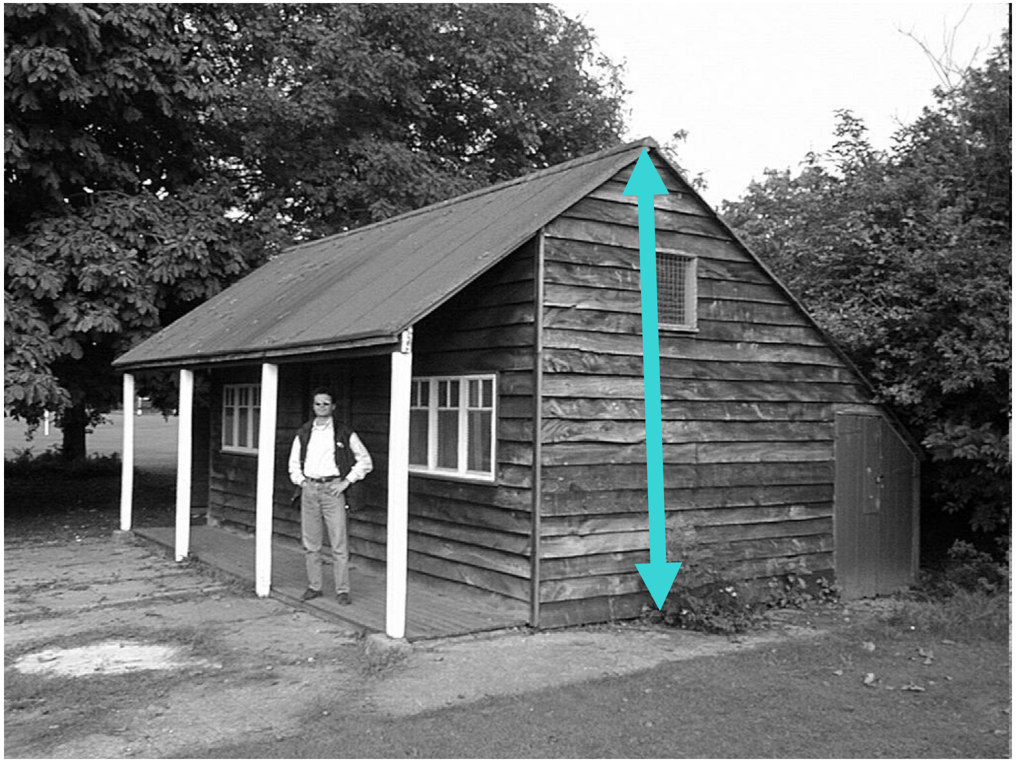
[Longuet-Higgins '81]



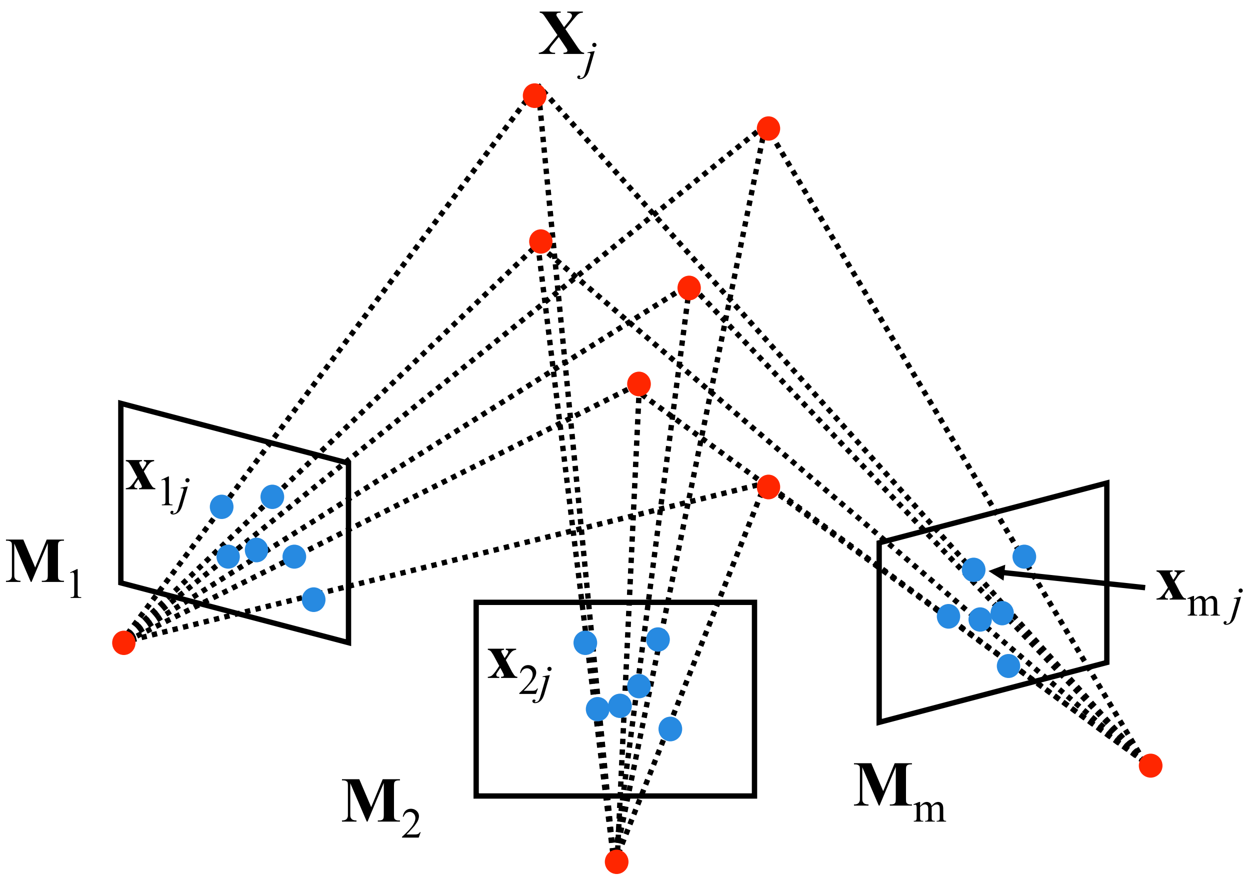
-The scene is determined by the images only up a similarity transformation (rotation, translation and scaling)

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)



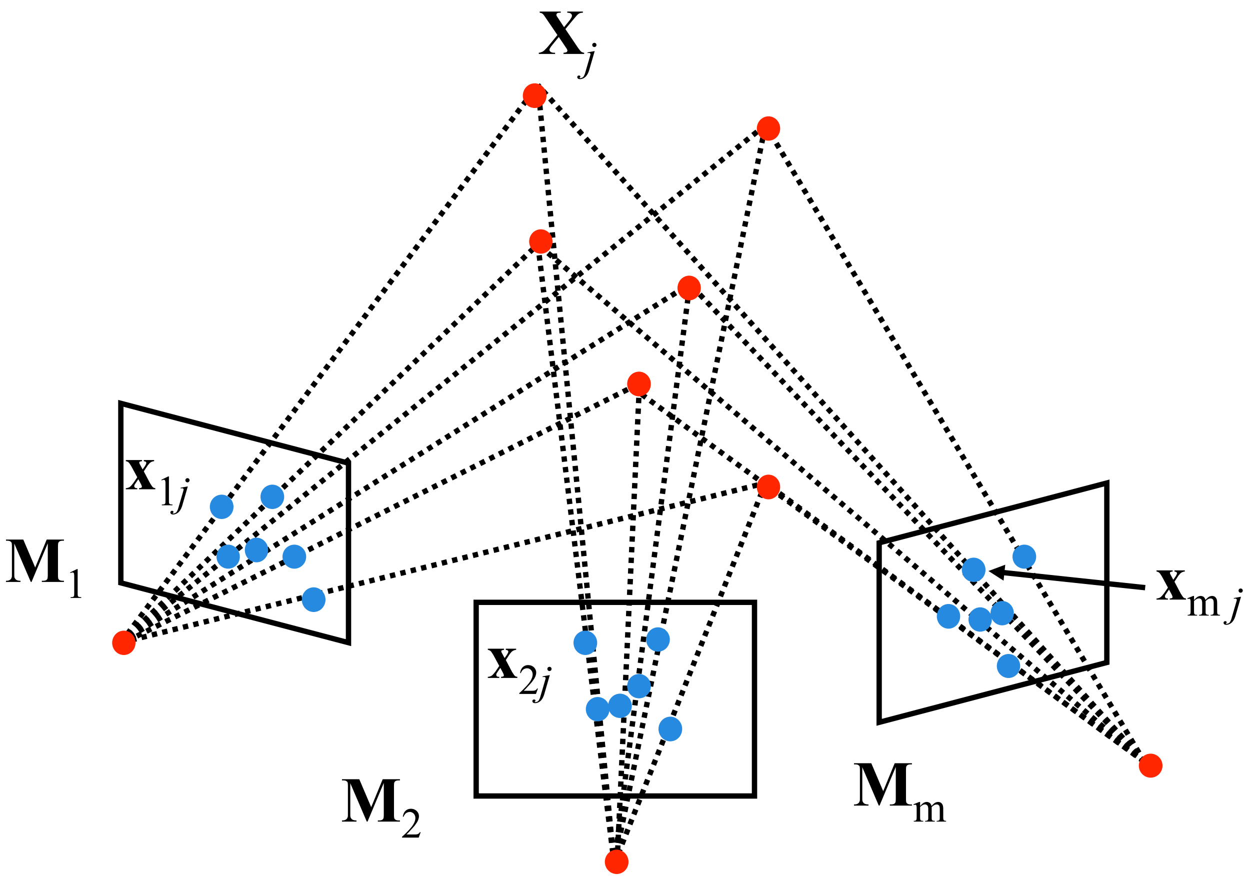
Structure from motion problem



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

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

Structure from motion problem



m cameras $M_1 \ldots M_m$

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

The 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]

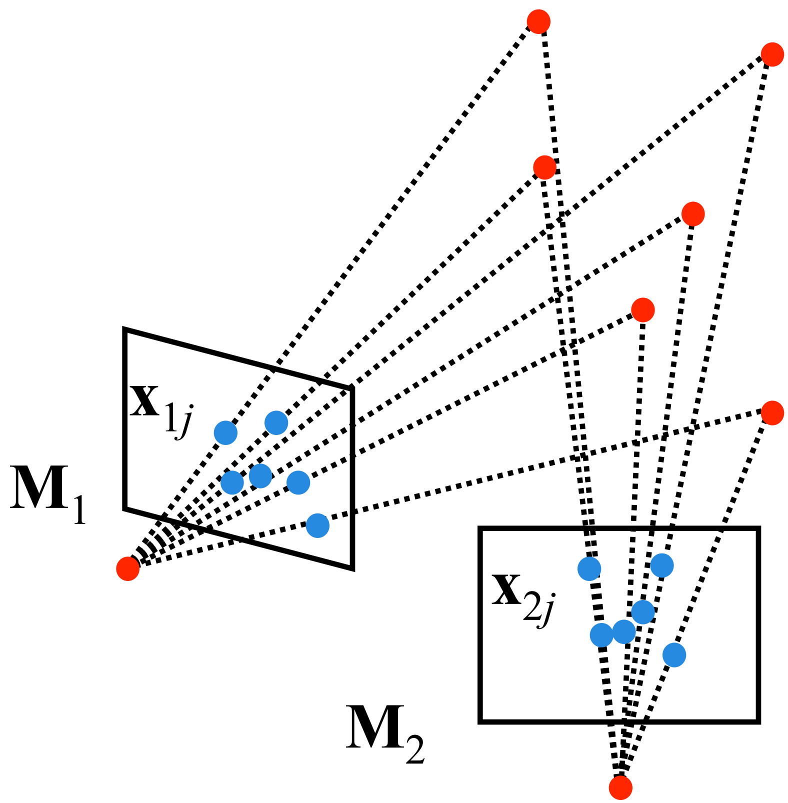
Structure-from-Motion Algorithms

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

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 camera to triangulate and estimate points in 3D

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

Algebraic approach (Fundamental matrix)

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

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

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

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

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

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

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

is this familiar?

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

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

Cross product as matrix multiplication

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

Algebraic approach (Fundamental matrix)

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

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

It's easy to verify that :

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

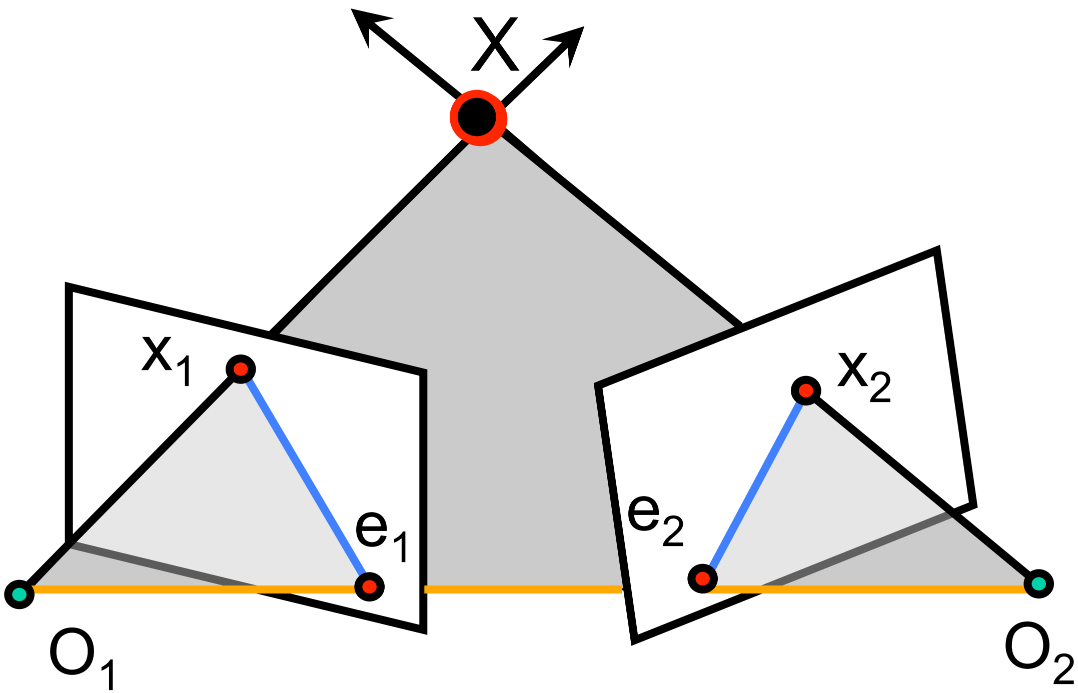
Compute $\mathbf{b}$ as least sq. solution of $\mathbf{F}\,\mathbf{b} = 0$

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

$$\boxed{\begin{aligned} \mathbf{A} &= [\mathbf{b}_\times]^{-1}\,\mathbf{F} \\ &= -[\mathbf{b}_\times]\,\mathbf{F} \end{aligned}}$$

Notice that b is an epipole

Epipolar Constraint [lecture 6]



$F x_2$ 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

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)

It's easy to verify that :

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

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

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

$$\boxed{\begin{aligned}\mathbf{A} &= [\mathbf{b}_\times]^{-1}\,\mathbf{F} \\ &= -[\mathbf{b}_\times]\,\mathbf{F}\end{aligned}}$$

Notice that b is an epipole

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

Perspective cameras are known

HZ, page 254
PF, page 288

Structure-from-Motion Algorithms

- Algebraic approach (by fundamental matrix)
- $\boxed{\text{Factorization method (by SVD)}}$
- Bundle adjustment

Projective factorization

$$\mathbf{D} = \begin{bmatrix} z_{11}\mathbf{x}_{11} & z_{12}\mathbf{x}_{12} & \cdots & z_{1n}\mathbf{x}_{1n} \\ z_{21}\mathbf{x}_{21} & z_{22}\mathbf{x}_{22} & \cdots & z_{2n}\mathbf{x}_{2n} \\ & & \ddots & \\ z_{m1}\mathbf{x}_{m1} & z_{m2}\mathbf{x}_{m2} & \cdots & z_{mn}\mathbf{x}_{mn} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathrm{M}_1 \\ \mathrm{M}_2 \\ \vdots \\ \mathrm{M}_m \end{bmatrix}}_{\substack{\text{cameras} \\ (3m \times 4)}} \underbrace{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}}_{\text{points } (4 \times n)}$$

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

A factorization method - (affine case; lecture 8)

Let's create a $2m \times n$ data (measurement) matrix:

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

($2m$ × n)

cameras ($2m$ × 3)

points (3 × n)

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

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

Projective factorization

$$\mathbf{D} = \begin{bmatrix} z_{11}\mathbf{x}_{11} & z_{12}\mathbf{x}_{12} & \cdots & z_{1n}\mathbf{x}_{1n} \\ z_{21}\mathbf{x}_{21} & z_{22}\mathbf{x}_{22} & \cdots & z_{2n}\mathbf{x}_{2n} \\ & & \ddots & \\ z_{m1}\mathbf{x}_{m1} & z_{m2}\mathbf{x}_{m2} & \cdots & z_{mn}\mathbf{x}_{mn} \end{bmatrix} = \underbrace{\begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_m \end{bmatrix}}_{\text{cameras } (3m \times 4)} \underbrace{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}}_{\text{points } (4 \times n)}$$

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

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

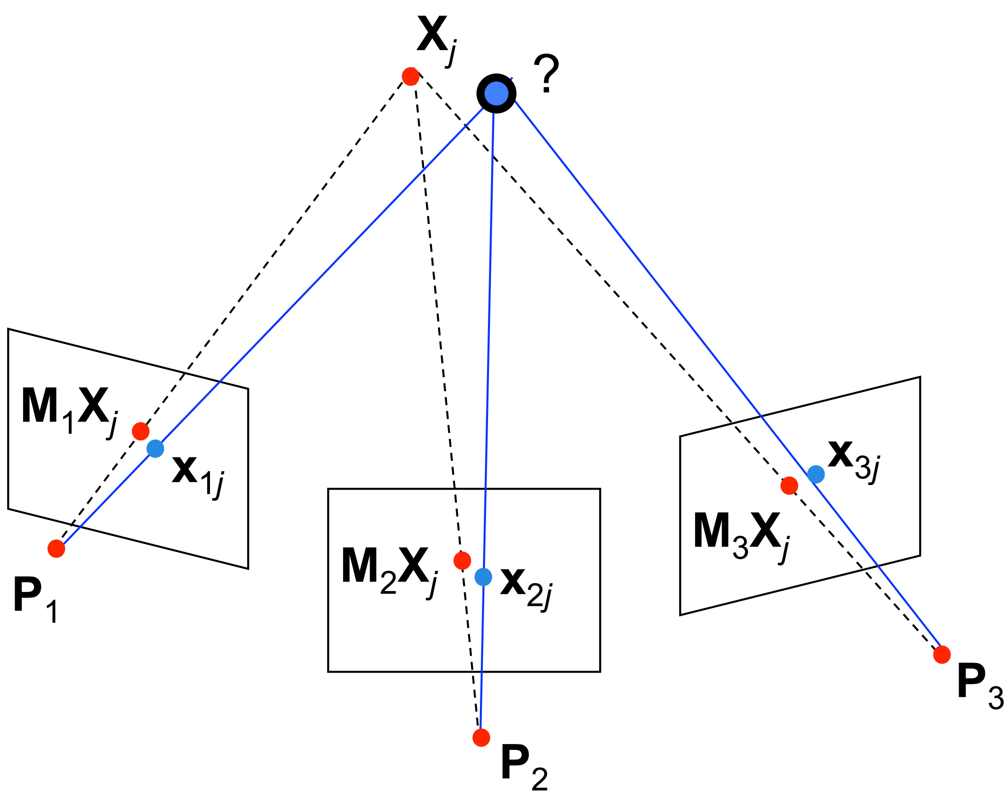
- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- $\boxed{\text{Bundle adjustment}}$

Bundle adjustment

Non-linear method for refining structure and motion

Minimizing re-projection error

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



Bundle adjustment

Non-linear method for refining structure and motion

Minimizing re-projection error

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

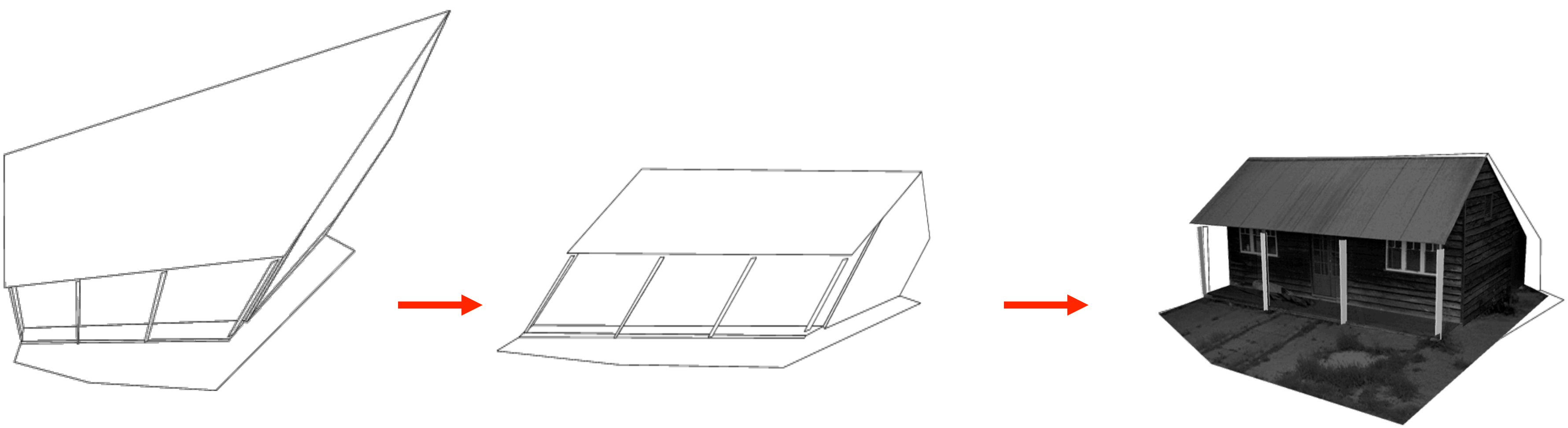
- Advantages
  - Handle large number of views
  - Handle missing data
- Limitations
  - Large minimization problem (parameters grow with number of views)
  - requires good initial condition

Used as the final step of SFM

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

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

$\{\boxed{M_i H}, \boxed{H^{-1} X_j}\}$ 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: $M_1 = [I \quad 0]$

Self-calibration

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

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

$$A = K_1$$

$$t = 0$$

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

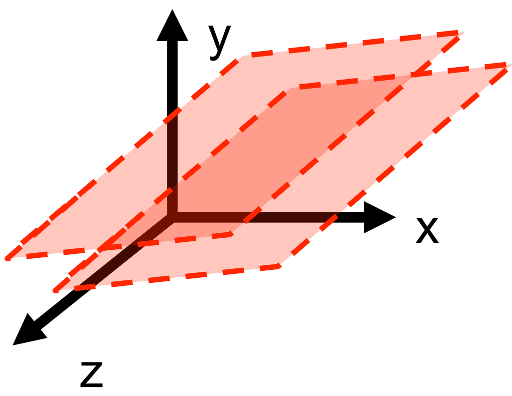
We can set k=1
(this fixes the scale of the reconstruction)

$$H = \begin{bmatrix} K_1 & 0 \\ v & 1 \end{bmatrix} \xrightarrow[\pi_\infty = H^{-1}\,\Pi_\infty = \begin{bmatrix} p \\ 1 \end{bmatrix}]{\text{Planes at infinity}} H = \begin{bmatrix} K_1 & 0 \\ -p^T K_1 & 1 \end{bmatrix}$$

See appendix

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

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

Self-calibration

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

$$\mathrm{H} = \begin{bmatrix} \mathrm{K}_1 & 0 \\ \mathrm{v} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathrm{K}_1^{-\mathrm{T}} & -\mathrm{K}_1^{-\mathrm{T}}\mathrm{v} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\mathrm{K}_1^{-\mathrm{T}}\mathrm{v} \\ 1 \end{bmatrix} \longrightarrow \mathrm{v} = -\mathrm{p}^{\mathrm{T}}\,\mathrm{K}_1$$

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

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

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

K_1 = calibration matrix of first camera

5 unknowns

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

3 unknowns

Self-calibration basic equation

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

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

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

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

Self-calibration basic equation

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

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

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

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

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_∞

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

Dual image of the absolute conic

Properties of $\boldsymbol{\omega}$

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

- It is not function of R, T

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

Self-calibration basic equation

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

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

$$\left(A_i - a_i p^T\right)\boldsymbol{\omega}_1^*\left(A_i - a_i p^T\right)^T = \boldsymbol{\omega}_i^*$$

$$\downarrow$$

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

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

$\longrightarrow$ 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

Self-calibration – other constraints

$$\left(\mathrm{A}_i - \mathrm{a}_i \mathrm{p}^T\right) \boldsymbol{\omega}_1^* \left(\mathrm{A}_i - \mathrm{a}_i \mathrm{p}^T\right)^T = \boldsymbol{\omega}_i^*$$

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

etc…

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

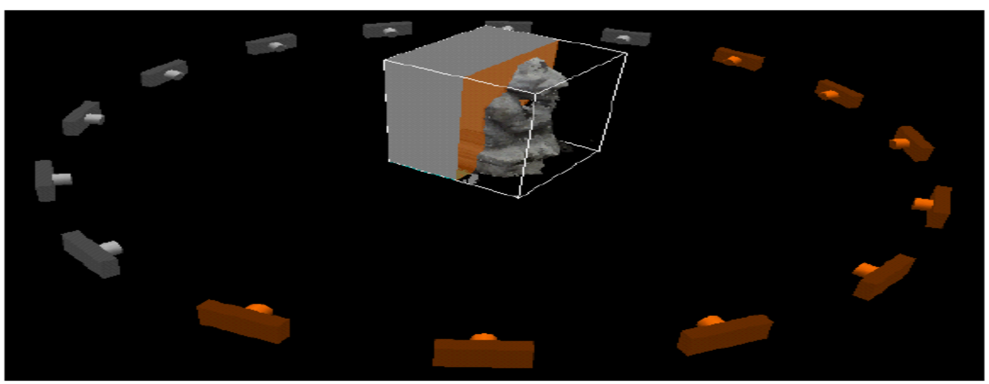
Issue: the larger is the number of view,
the harder is the correspondence problem

Bundle adjustment helps!

Self-calibration - summary

Constraints on camera motion can be incorporated

- Linearly translating camera



- Single axis of rotation: turntable motion

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 **

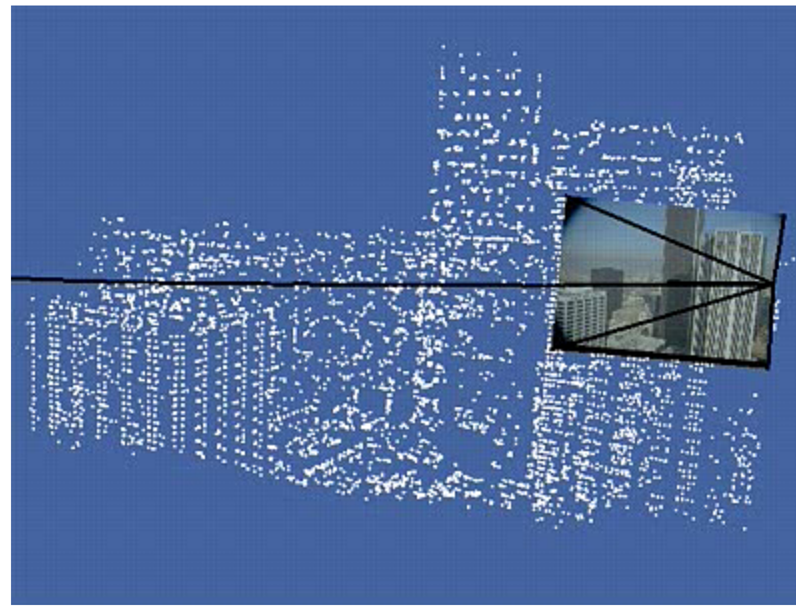
1. Bundle adjustment with self-calibration constraints

Correspondences

- Can refine feature matching <u>after</u> 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]

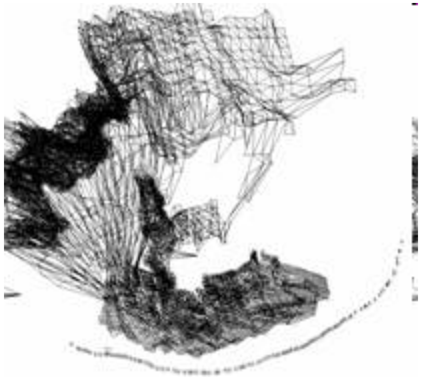
Applications

Courtesy of Oxford **Visual Geometry Group**



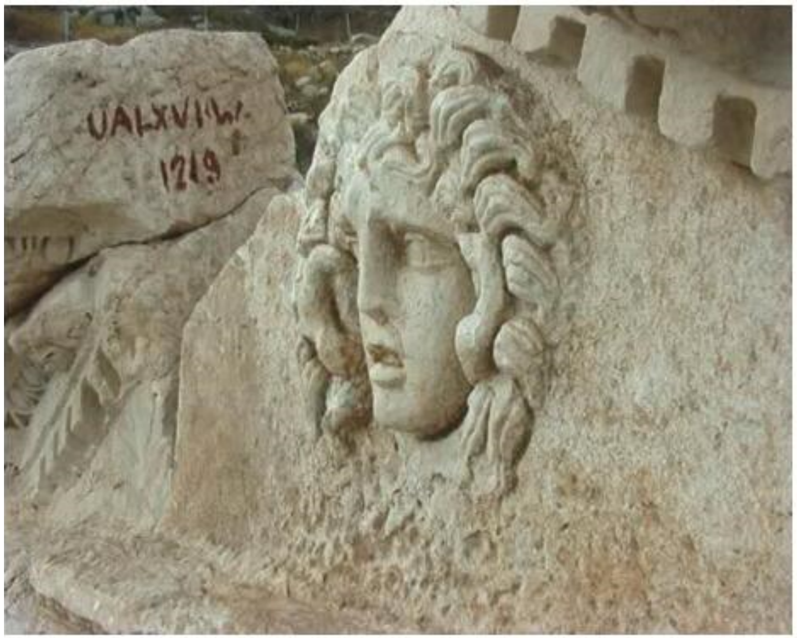
Applications

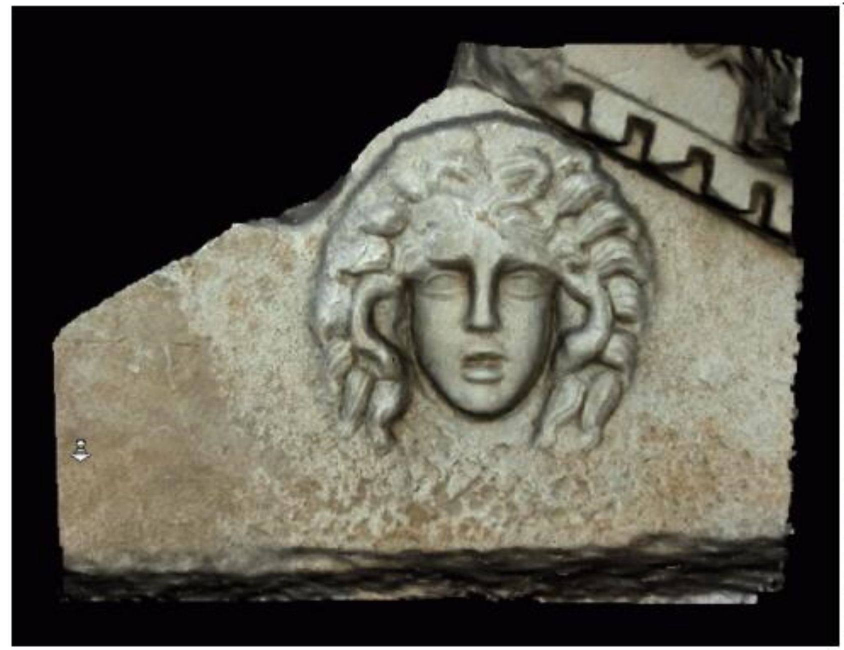
D. Nistér, PhD thesis ‘01



Applications

M. Pollefeys et al 98---





Applications

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

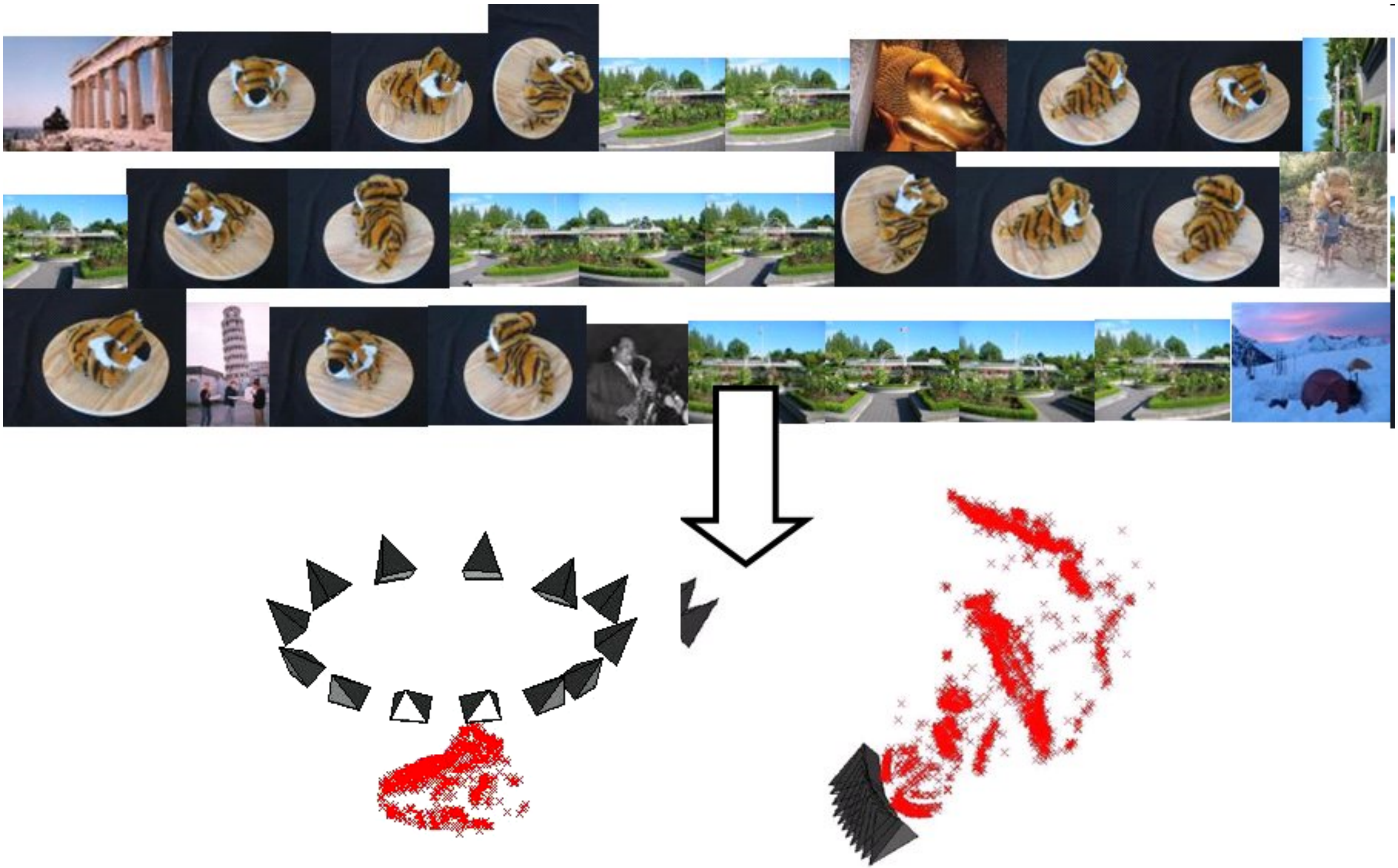


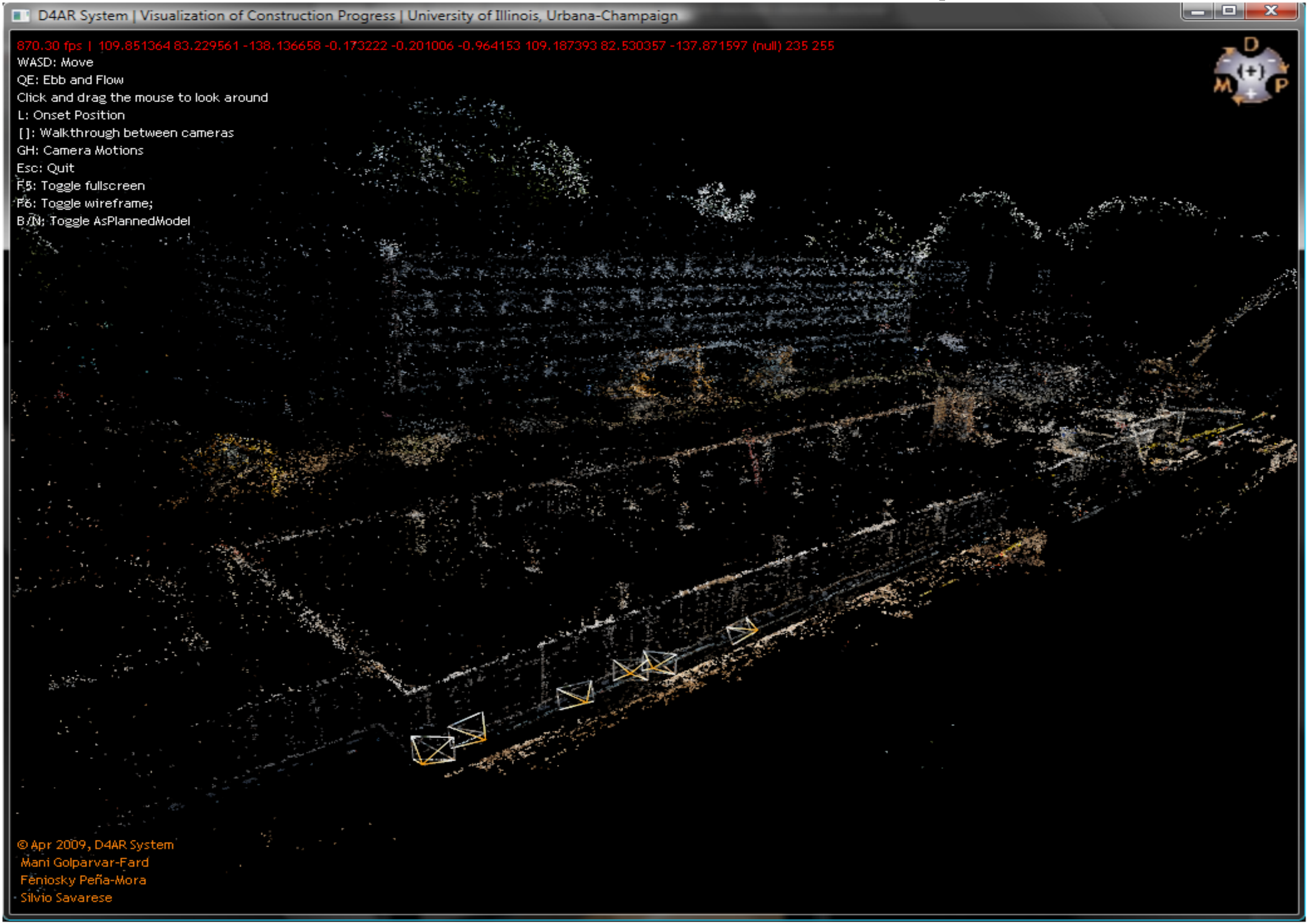
Photo synth

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

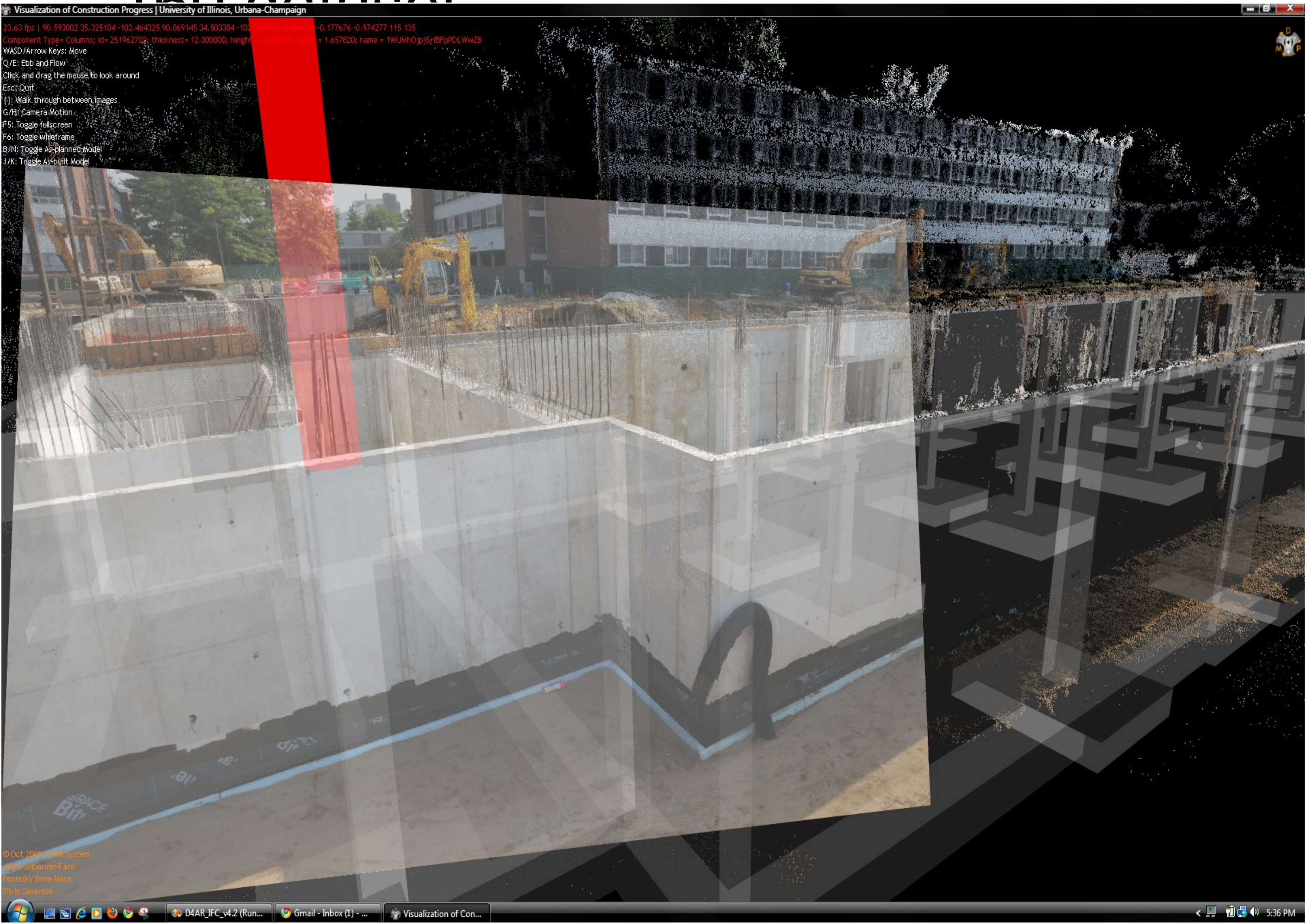
Incremental reconstruction of construction sites

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

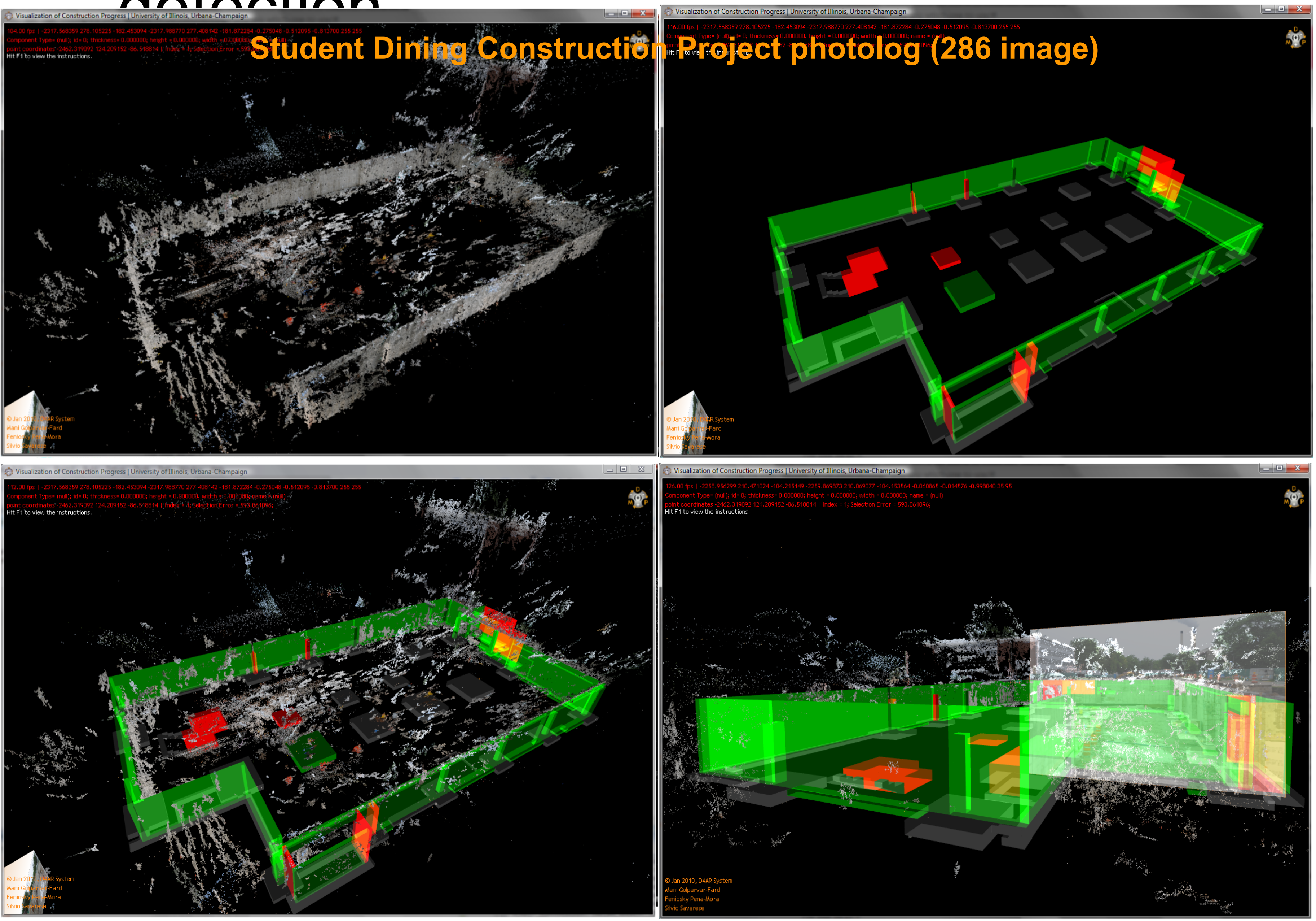
Reconstructed scene + Site photos



Dense reconstruction results for RH 160 dataset



The results of automated progress detection



Next lecture

Volumetric stereo