C280, Computer Vision

Prof. Trevor Darrell
trevor@eecs.berkeley.edu

Lecture 11: Structure from Motion

Roadmap

- Previous: Image formation, filtering, local features, (Texture)…
- Tues: Feature-based Alignment
  - Stitching images together
  - Homographies, RANSAC, Warping, Blending
  - Global alignment of planar models
- Today: Dense Motion Models
  - Local motion / feature displacement
  - Parametric optic flow
- No classes next week: ICCV conference
- Oct 6th: Stereo / ‘Multi-view’: Estimating depth with known inter-camera pose
- **Oct 8th: ‘Structure-from-motion’: Estimation of pose and 3D structure**
  - **Factorization approaches**
  - **Global alignment with 3D point models**

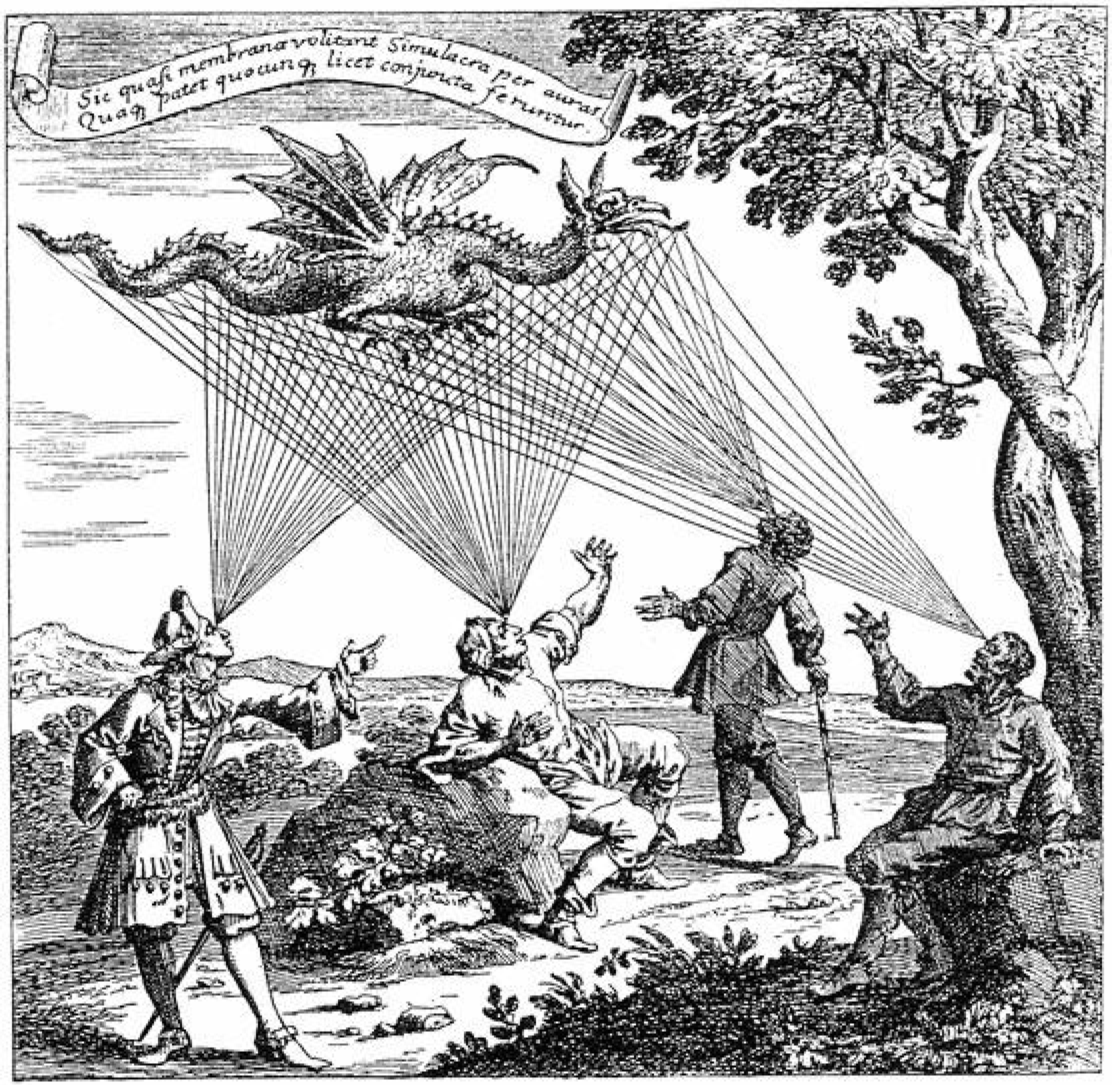
Last time: Stereo

- Human stereopsis & stereograms
- Epipolar geometry and the epipolar constraint
  - Case example with parallel optical axes
  - General case with calibrated cameras
- Correspondence search
- The Essential and the Fundamental Matrix
- Multi-view stereo

Today: SFM

- SFM problem statement
- Factorization
- Projective SFM
- Bundle Adjustment
- Photo Tourism
- “Rome in a day:

Structure from motion


Sic quaſi membrana volitant Simulacra per auras
Quaq; patet quocunq; licet conjuncta feruntur.

Драконъ, видимый подъ различными углами зрѣнія

По гравюрѣ на мѣди изъ „Oculus artificialis teledioptricus“ Цана. 1702 года.

Lazebnik

Multiple-view geometry questions

- **Scene geometry (structure):** Given 2D point matches in two or more images, where are the corresponding points in 3D?

- **Correspondence (stereo matching):** Given a point in just one image, how does it constrain the position of the corresponding point in another image?

- **Camera geometry (motion):** Given a set of corresponding points in two or more images, what are the camera matrices for these views?

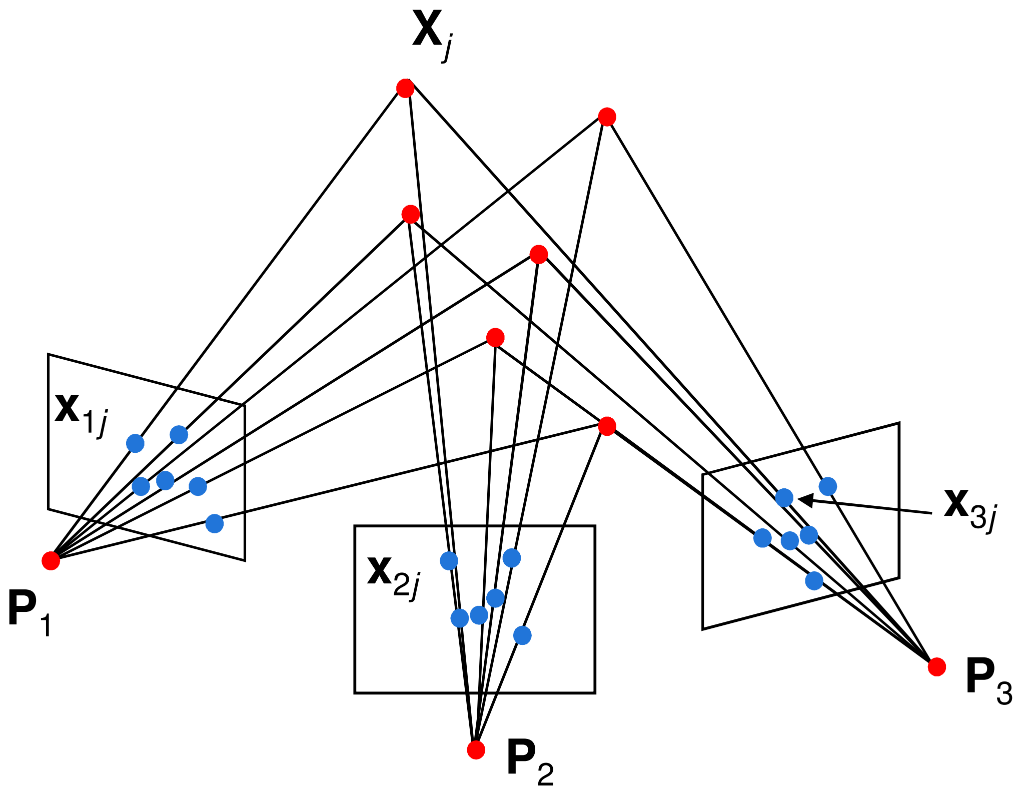
Lazebnik

Structure from motion

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

$$\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \qquad i = 1, \dots, m, \quad j = 1, \dots, n$$

- Problem: estimate *m* projection matrices $\mathbf{P}_i$ and *n* 3D points $\mathbf{X}_j$ from the *mn* correspondences $\mathbf{x}_{ij}$



Lazebnik

Structure from motion ambiguity

- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the projections of the scene points in the image remain exactly the same:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\frac{1}{k}\mathbf{P}\right)(k\,\mathbf{X})$$

It is impossible to recover the absolute scale of the scene!

Lazebnik

Structure from motion ambiguity

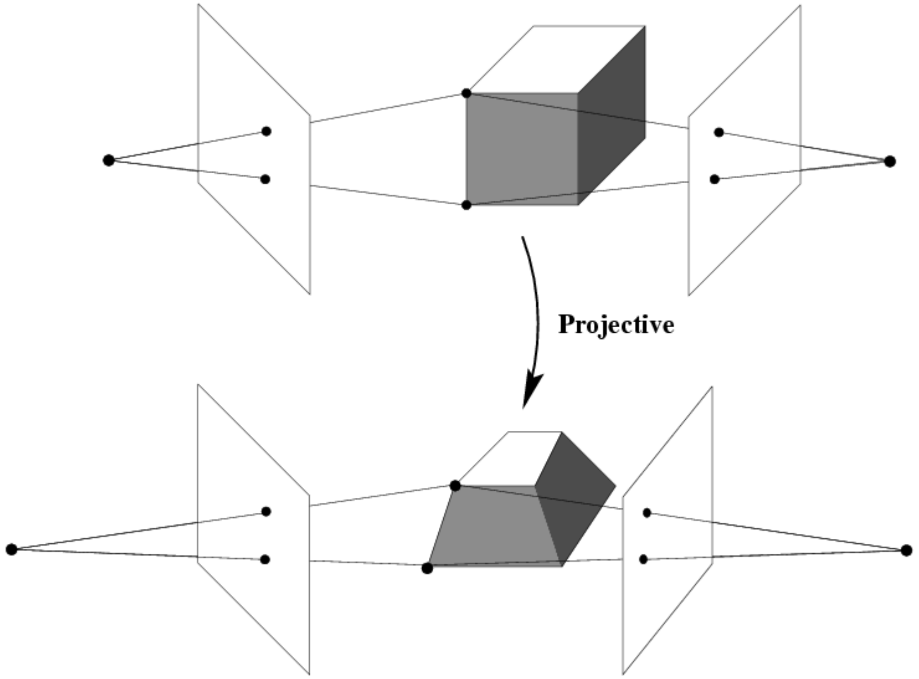
- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the projections of the scene points in the image remain exactly the same

- More generally: if we transform the scene using a transformation **Q** and apply the inverse transformation to the camera matrices, then the images do not change

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}^{-1}\right)\left(\mathbf{Q}\mathbf{X}\right)$$

Lazebnik

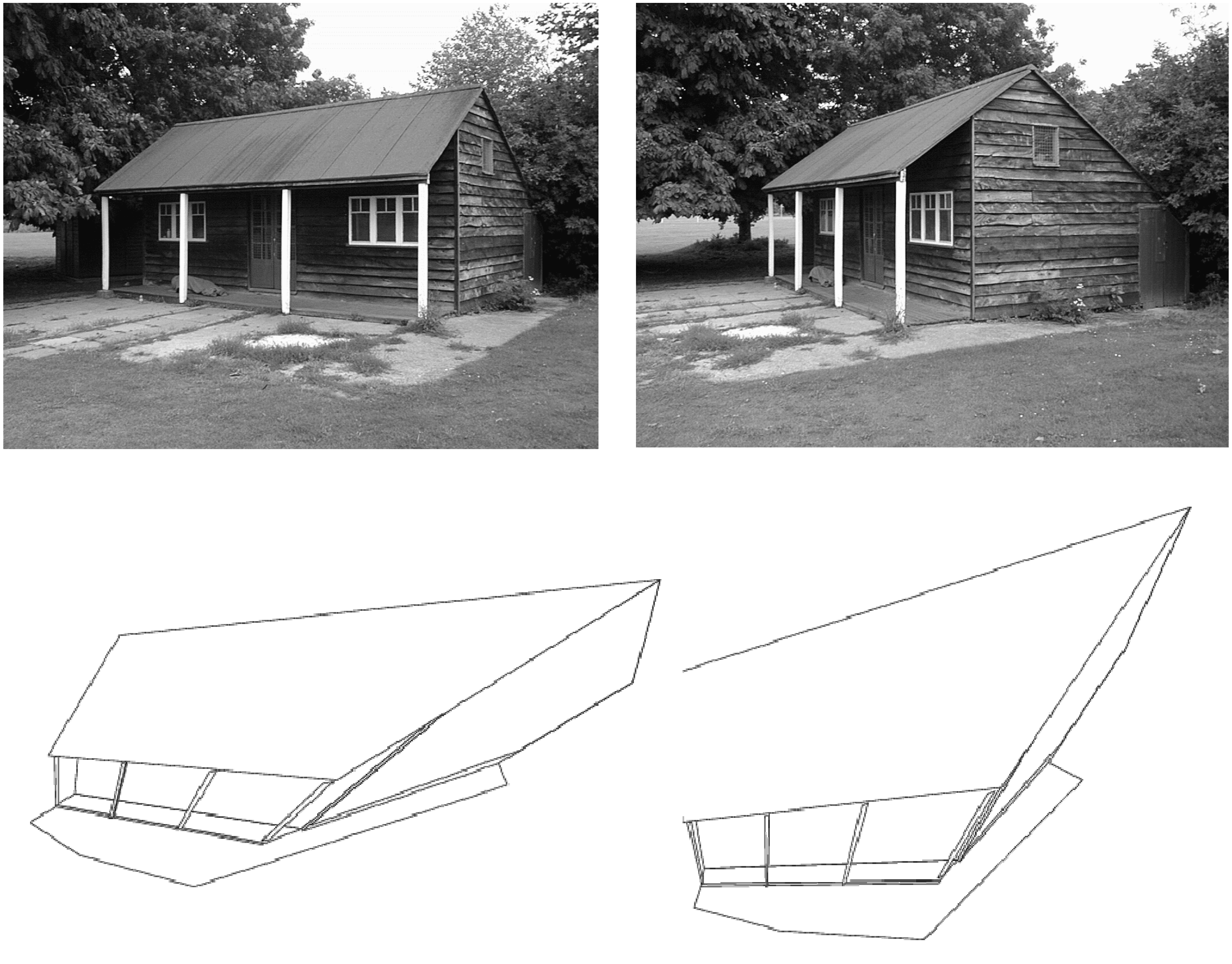
Projective ambiguity



$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_{\mathbf{P}}^{\mathbf{-1}}\right)\left(\mathbf{Q}_{\mathbf{P}}\,\mathbf{X}\right)$$

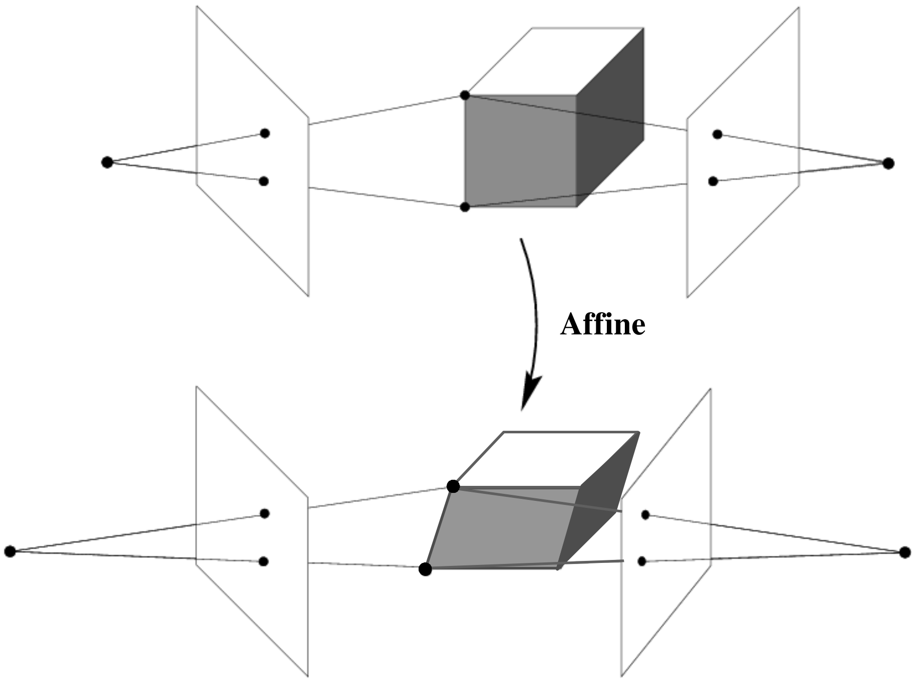
Lazebnik

Projective ambiguity



Lazebnik

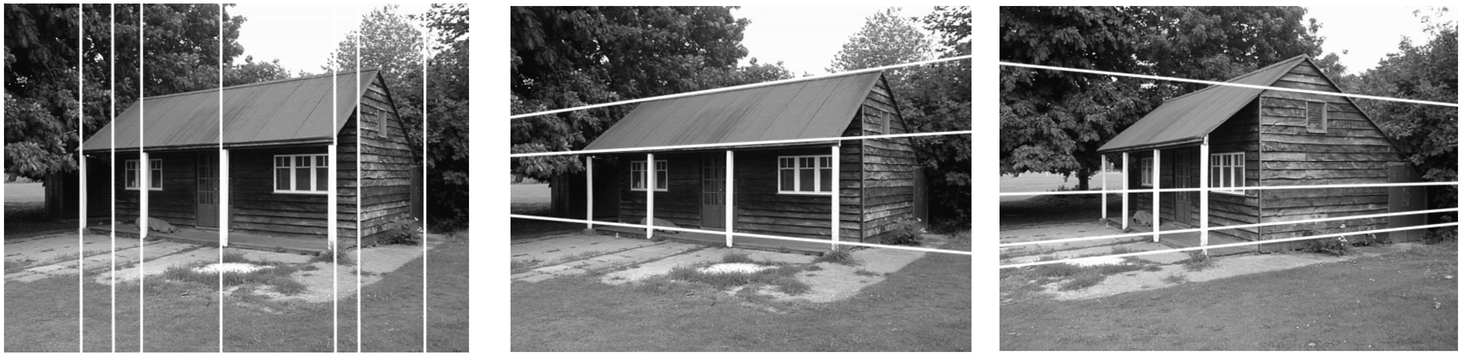
Affine ambiguity

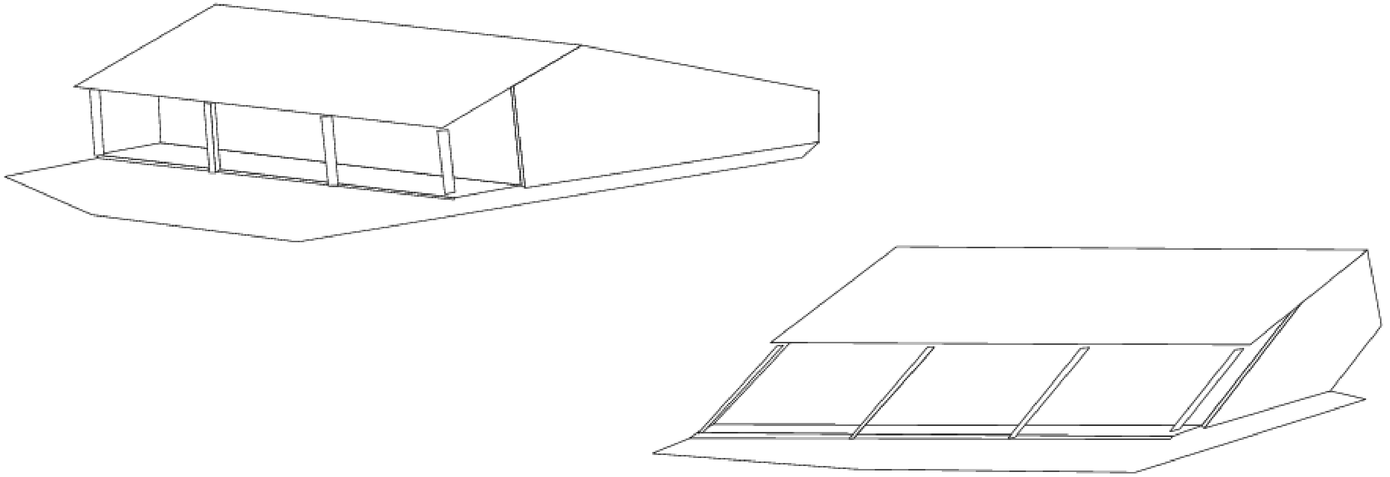


$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_{\mathbf{A}}^{\mathbf{-1}}\right)\left(\mathbf{Q}_{\mathbf{A}}\,\mathbf{X}\right)$$

Lazebnik

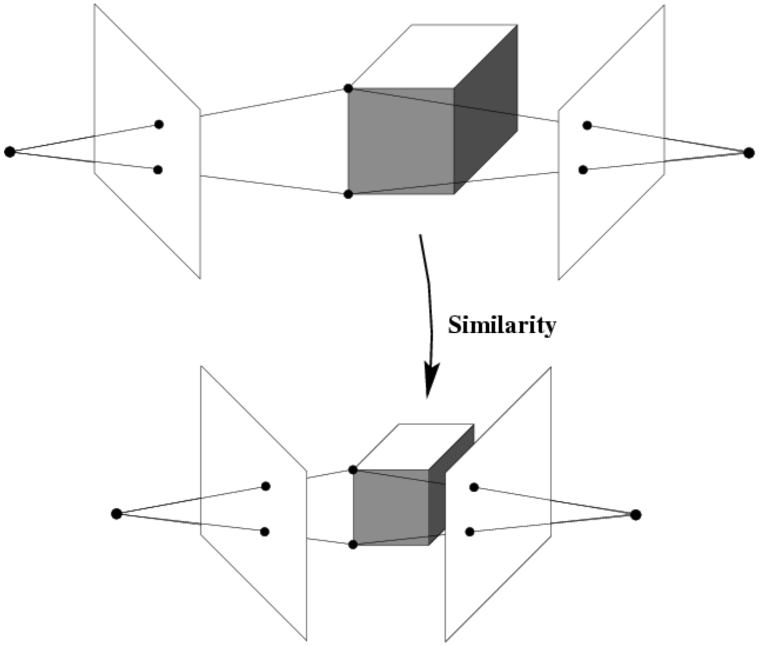
Affine ambiguity





Lazebnik

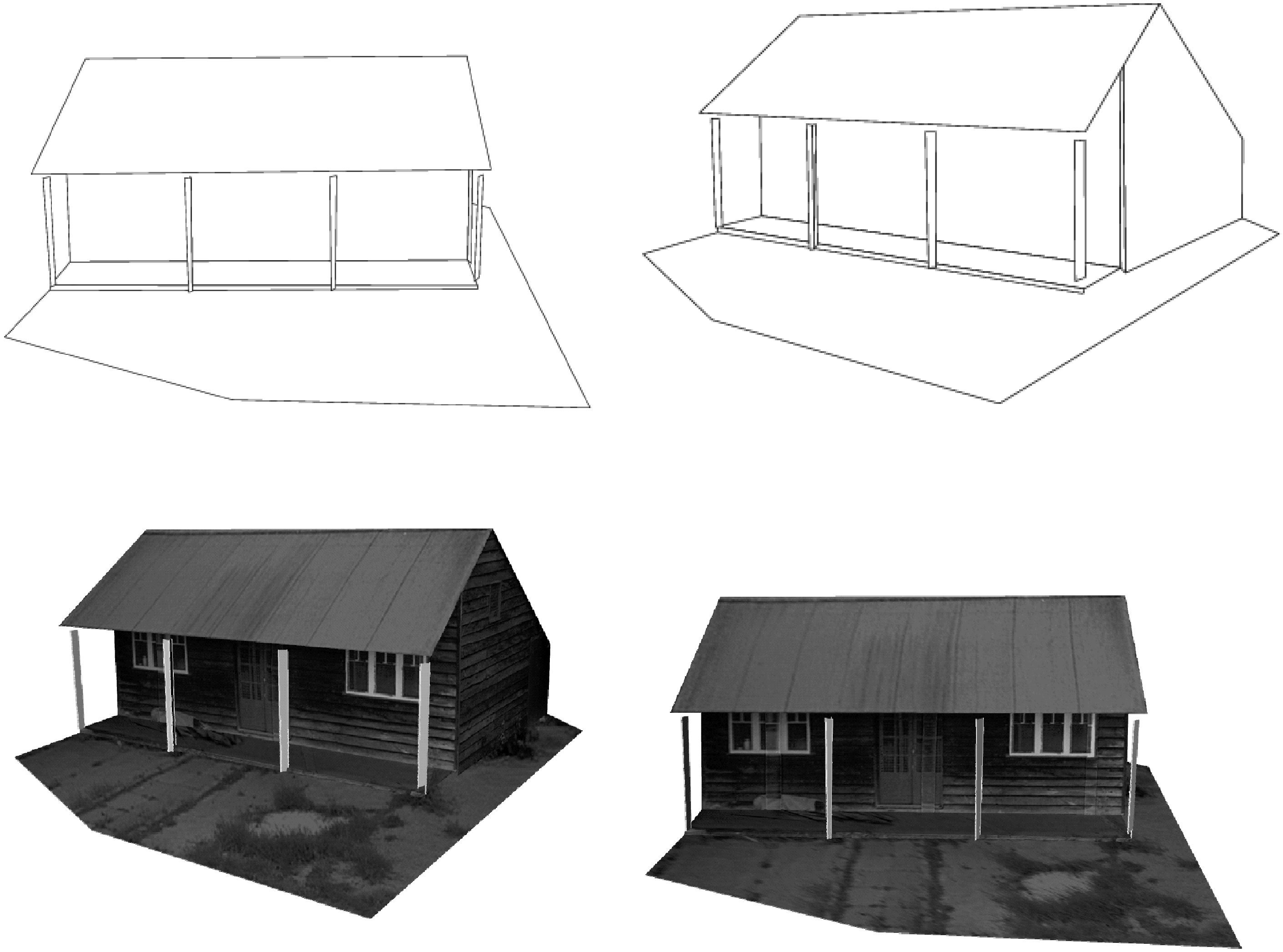
Similarity ambiguity



$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_S^{-1}\right)\left(\mathbf{Q}_S\mathbf{X}\right)$$

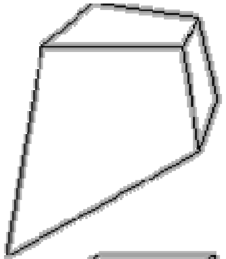
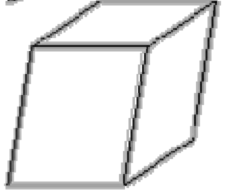
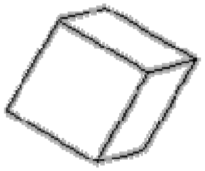
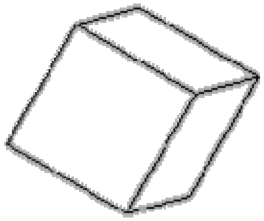
Lazebnik

Similarity ambiguity



Lazebnik

Hierarchy of 3D transformations

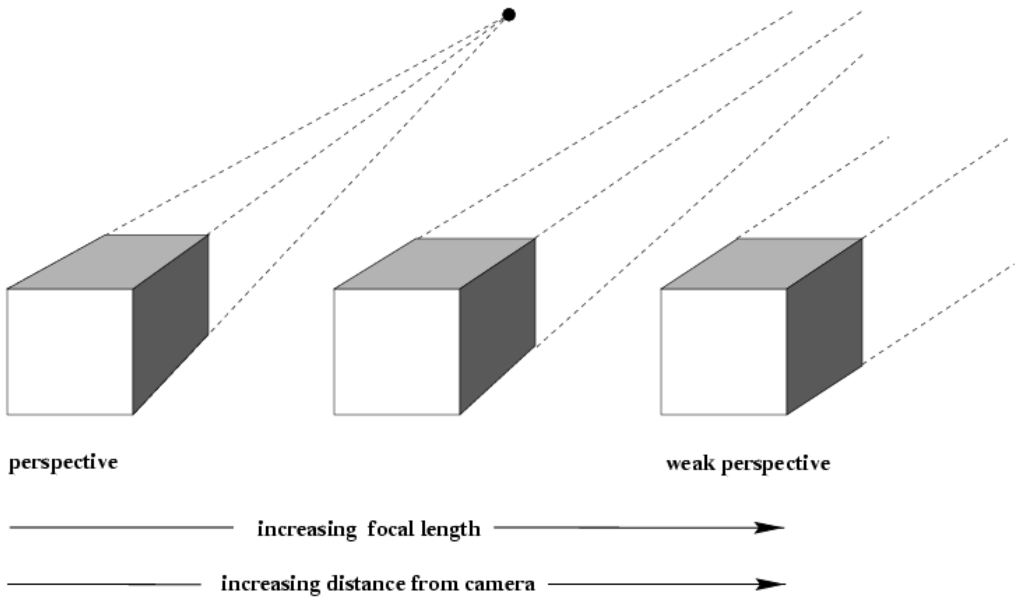
Projective 15dof	$\begin{bmatrix} \mathrm{A} & \mathrm{t} \\ \mathrm{v}^{\mathrm{T}} & v \end{bmatrix}$		Preserves intersection and tangency
Affine 12dof	$\begin{bmatrix} \mathrm{A} & \mathrm{t} \\ 0^{\mathrm{T}} & 1 \end{bmatrix}$		Preserves parallellism, volume ratios
Similarity 7dof	$\begin{bmatrix} s\mathrm{R} & \mathrm{t} \\ 0^{\mathrm{T}} & 1 \end{bmatrix}$		Preserves angles, ratios of length
Euclidean 6dof	$\begin{bmatrix} \mathrm{R} & \mathrm{t} \\ 0^{\mathrm{T}} & 1 \end{bmatrix}$		Preserves angles, lengths

- With no constraints on the camera calibration matrix or on the scene, we get a *projective* reconstruction
- Need additional information to *upgrade* the reconstruction to affine, similarity, or Euclidean

Lazebnik

Structure from motion

- Let’s start with *affine cameras* (the math is easier)



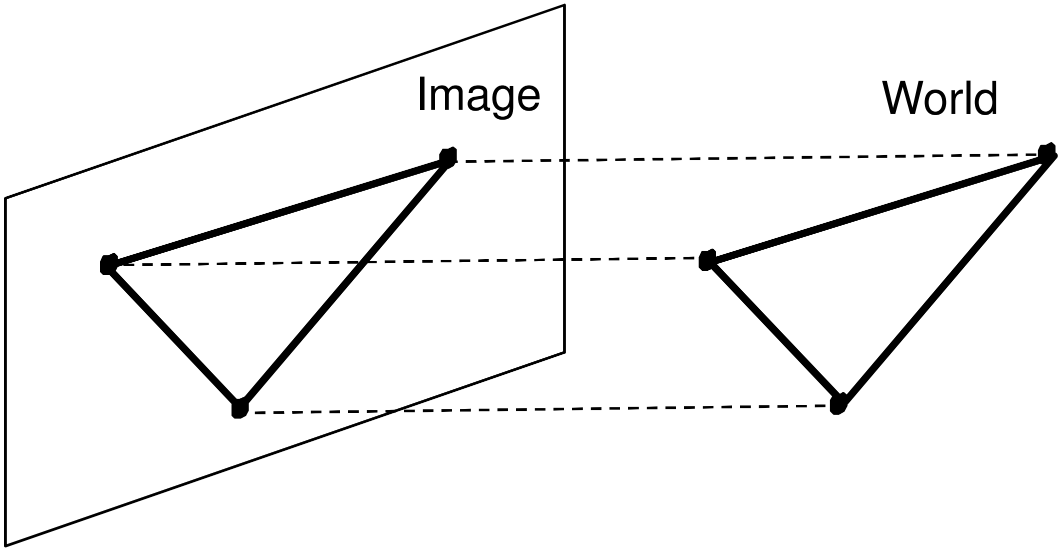
center at infinity

Lazebnik

Recall: Orthographic Projection

Special case of perspective projection

- Distance from center of projection to image plane is infinite



- Projection matrix:

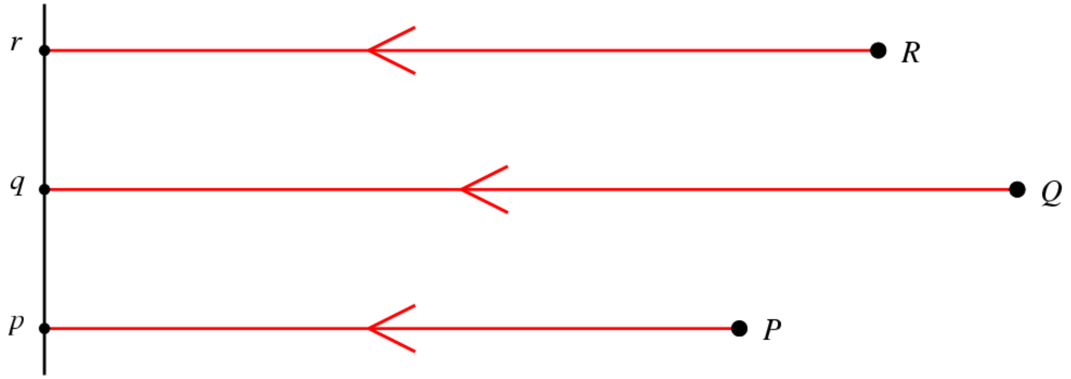
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

Lazebnik

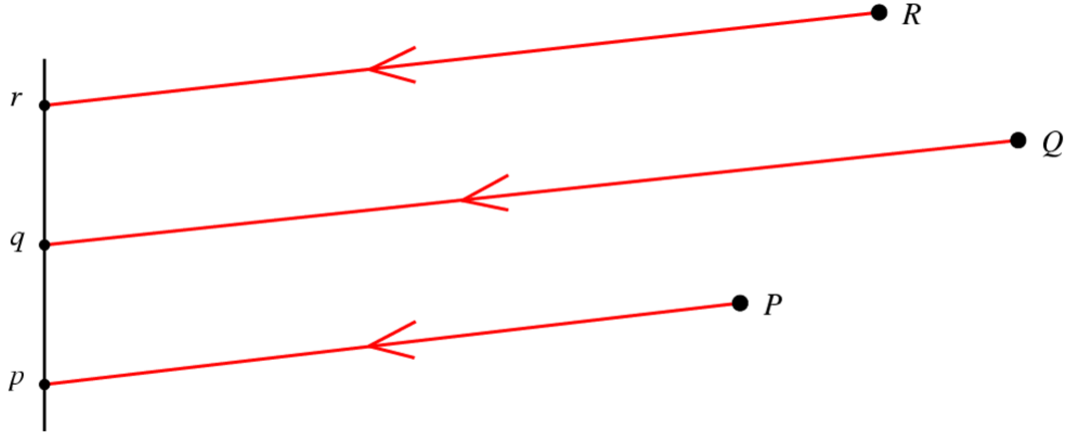
Slide by Steve Seitz

Affine cameras

Orthographic Projection



Parallel Projection



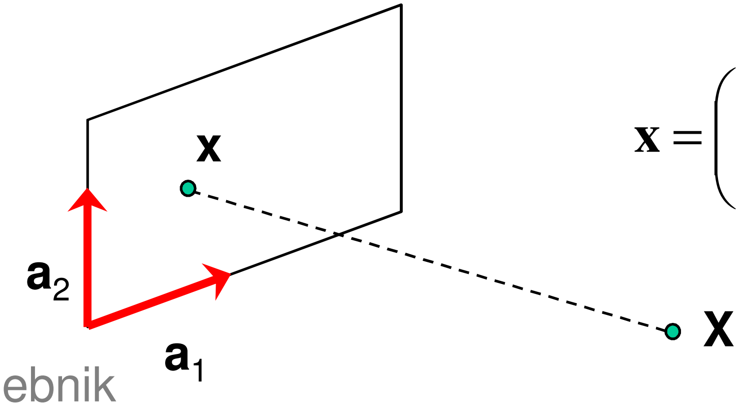
Lazebnik

Affine cameras

- A general affine camera combines the effects of an affine transformation of the 3D space, orthographic projection, and an affine transformation of the image:

$$\mathbf{P} = [3\times 3\,\text{affine}]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}[4\times 4\,\text{affine}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

- Affine projection is a linear mapping + translation in inhomogeneous coordinates



$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \mathbf{AX} + \mathbf{b}$$

Projection of world origin

Lazebnik

Affine structure from motion

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

$$\mathbf{x}_{ij} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i, \quad i = 1,\dots, m, \; j = 1, \dots, n$$

- Problem: use the *mn* correspondences $\mathbf{x}_{ij}$ to estimate *m* projection matrices $\mathbf{A}_i$ and translation vectors $\mathbf{b}_i$, and *n* points $\mathbf{X}_j$
- The reconstruction is defined up to an arbitrary *affine* transformation **Q** (12 degrees of freedom):

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \mathbf{Q}^{-1}, \qquad \begin{pmatrix} \mathbf{X} \\ \mathbf{1} \end{pmatrix} \rightarrow \mathbf{Q} \begin{pmatrix} \mathbf{X} \\ \mathbf{1} \end{pmatrix}$$

- We have 2*mn* knowns and 8*m* + 3*n* unknowns (minus 12 dof for affine ambiguity)
- Thus, we must have 2*mn* >= 8*m* + 3*n* – 12
- For two views, we need four point correspondences

Lazebnik

Affine structure from motion

- Centering: subtract the centroid of the image points

$$\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik} = \mathbf{A}_i\mathbf{X}_j + \mathbf{b}_i - \frac{1}{n}\sum_{k=1}^{n}\left(\mathbf{A}_i\mathbf{X}_k + \mathbf{b}_i\right)$$

$$= \mathbf{A}_i\left(\mathbf{X}_j - \frac{1}{n}\sum_{k=1}^{n}\mathbf{X}_k\right) = \mathbf{A}_i\hat{\mathbf{X}}_j$$

- For simplicity, assume that the origin of the world coordinate system is at the centroid of the 3D points
- After centering, each normalized point $\mathbf{x}_{ij}$ is related to the 3D point $\mathbf{X}_i$ by

$$\hat{\mathbf{x}}_{ij} = \mathbf{A}_i\mathbf{X}_j$$

Lazebnik

Affine structure from motion

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

$$\begin{array}{cc}
\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix} & \Bigg\downarrow \begin{matrix} \text{cameras} \\ (2\,m) \end{matrix} \\
\xrightarrow{\hspace{6em}} & \\
\text{points } (n) &
\end{array}$$

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Affine structure from motion

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

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

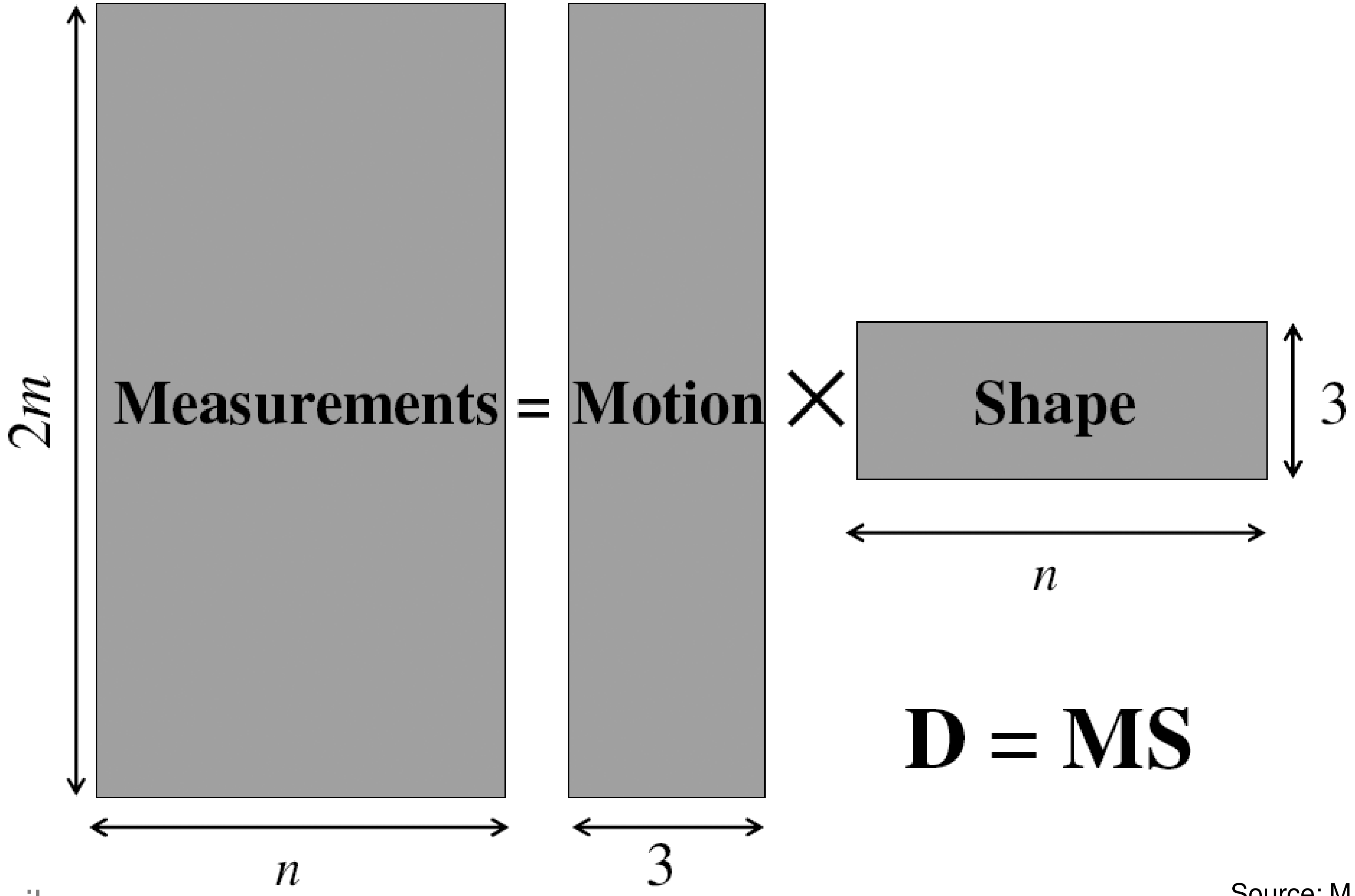
points (3 × *n*)

cameras
(2 *m* × 3)

The measurement matrix **D** = **MS** must have rank 3!

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Factorizing the measurement matrix



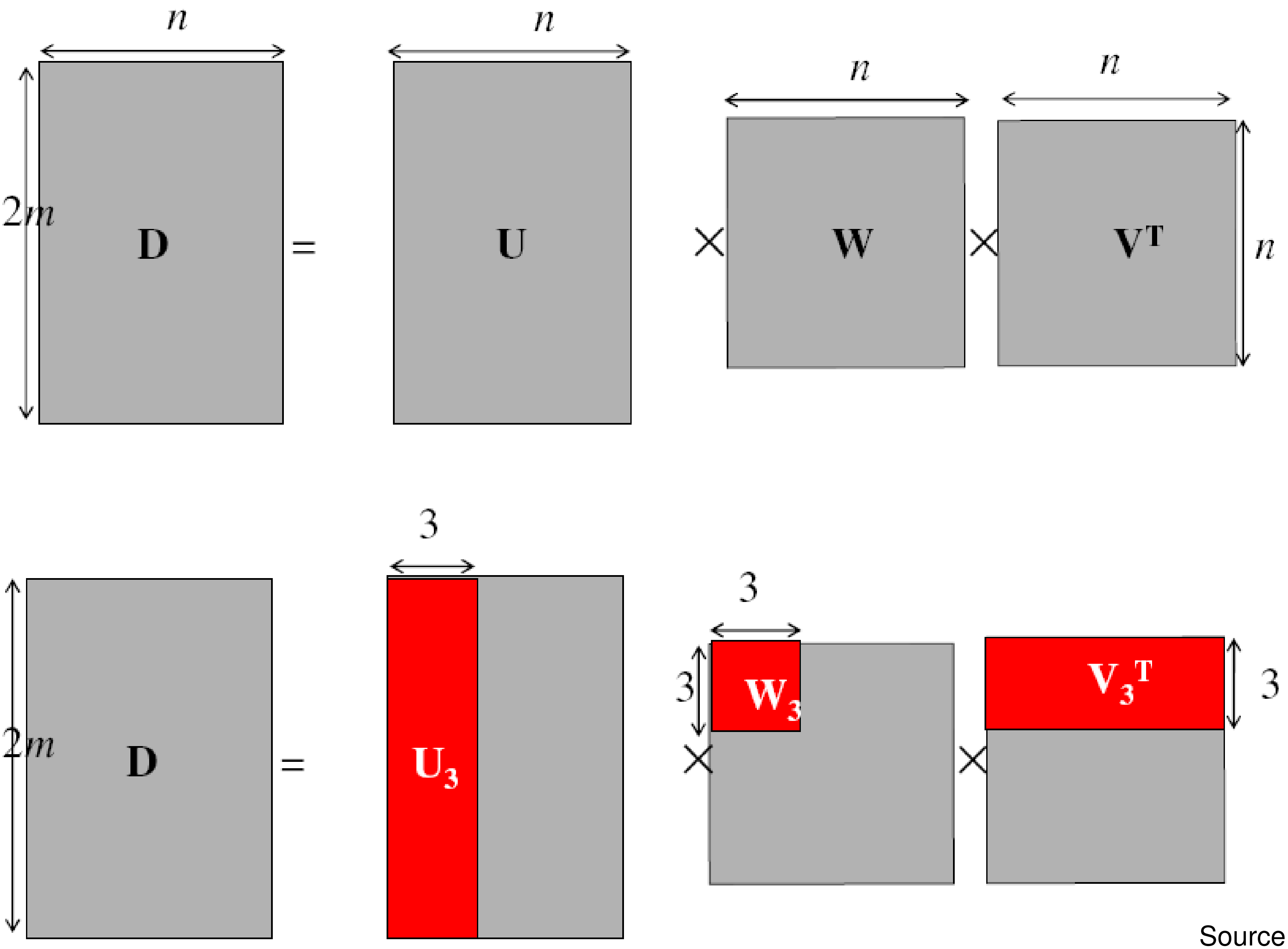
$$\mathbf{D} = \mathbf{MS}$$

Source: M. Hebert

Lazebnik

Factorizing the measurement matrix

- Singular value decomposition of D:

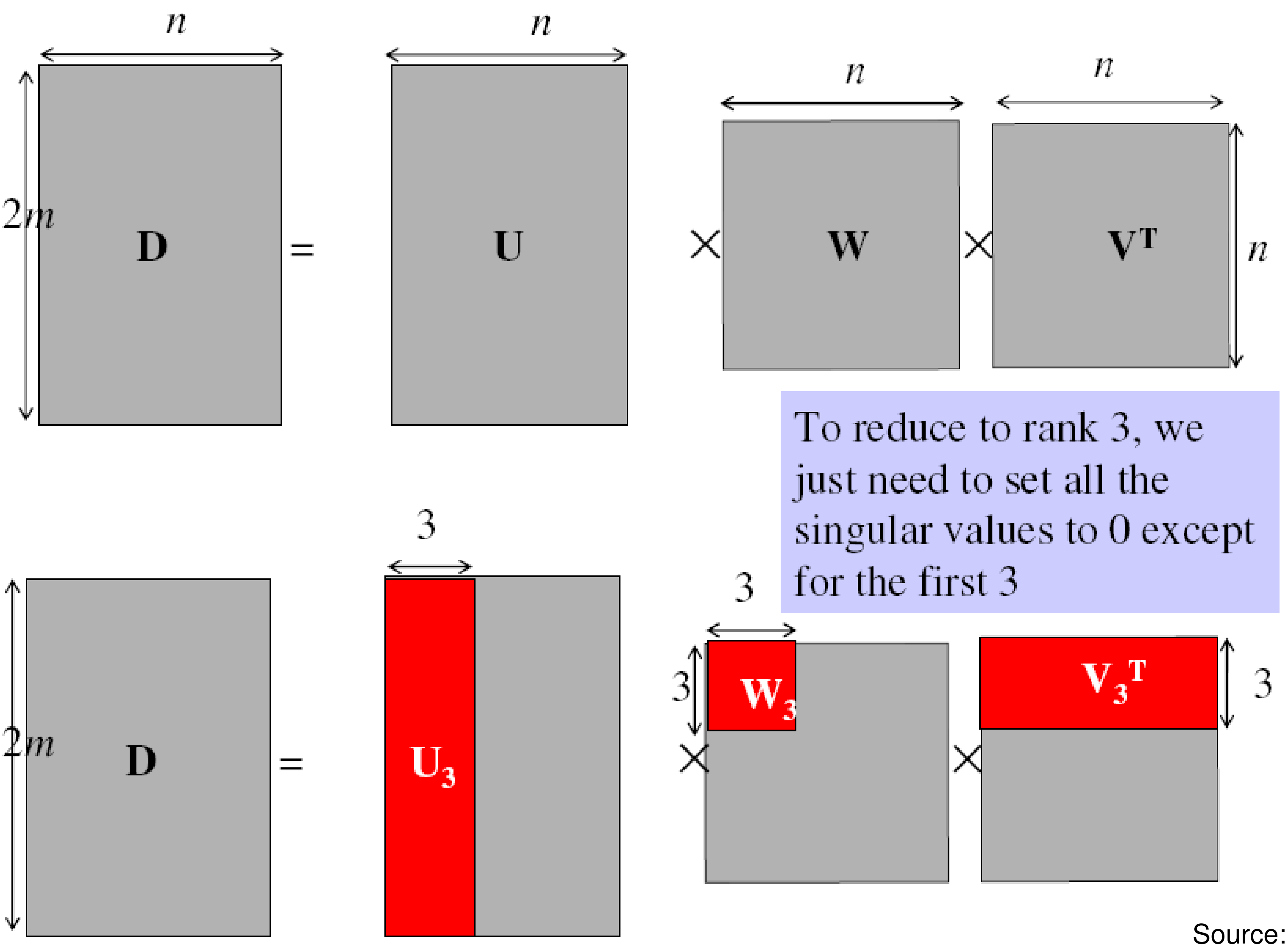


Lazebnik

Source: M. Hebert

Factorizing the measurement matrix

- Singular value decomposition of D:



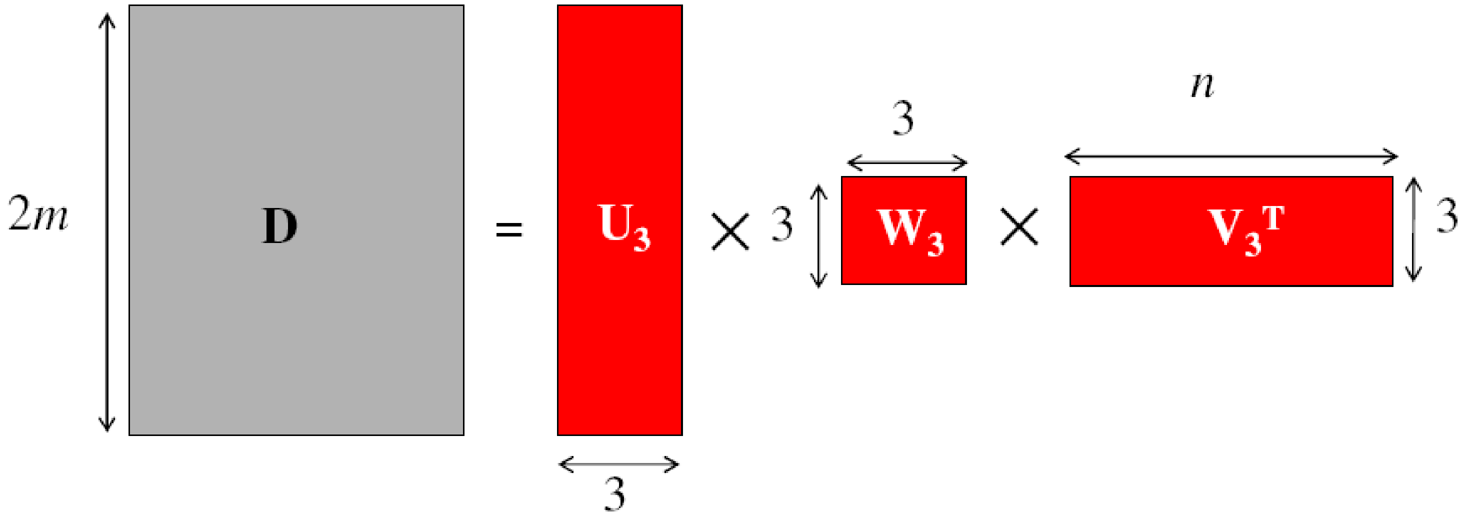
To reduce to rank 3, we just need to set all the singular values to 0 except for the first 3

Source: M. Hebert

Lazebnik

Factorizing the measurement matrix

- Obtaining a factorization from SVD:

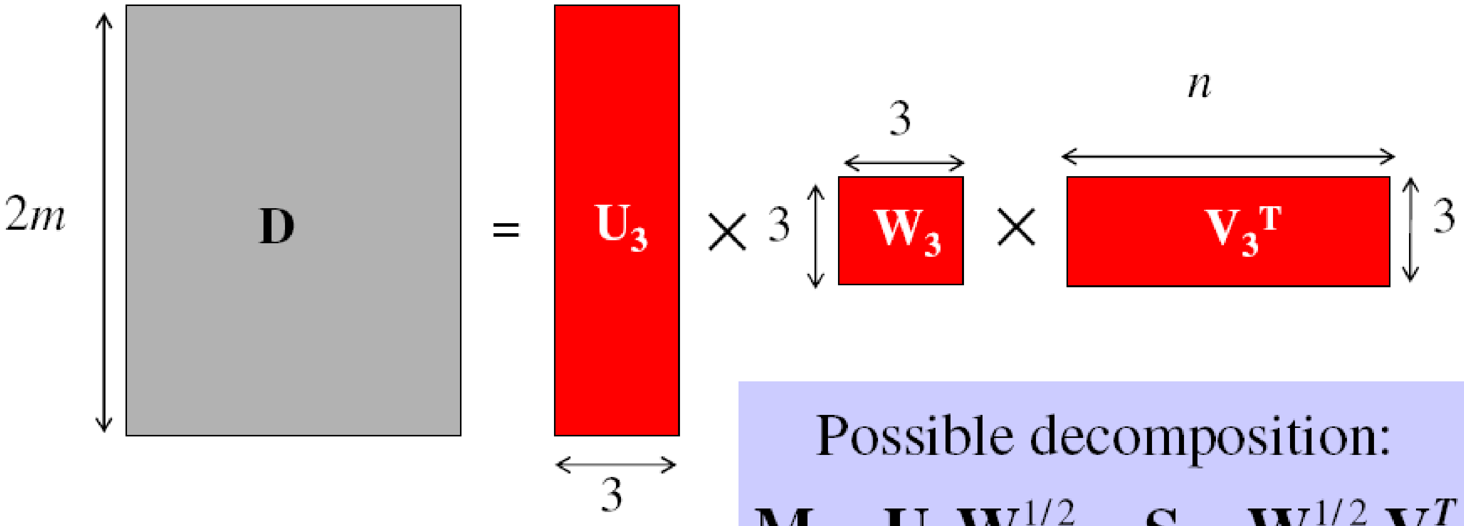


Lazebnik

Source: M. Hebert

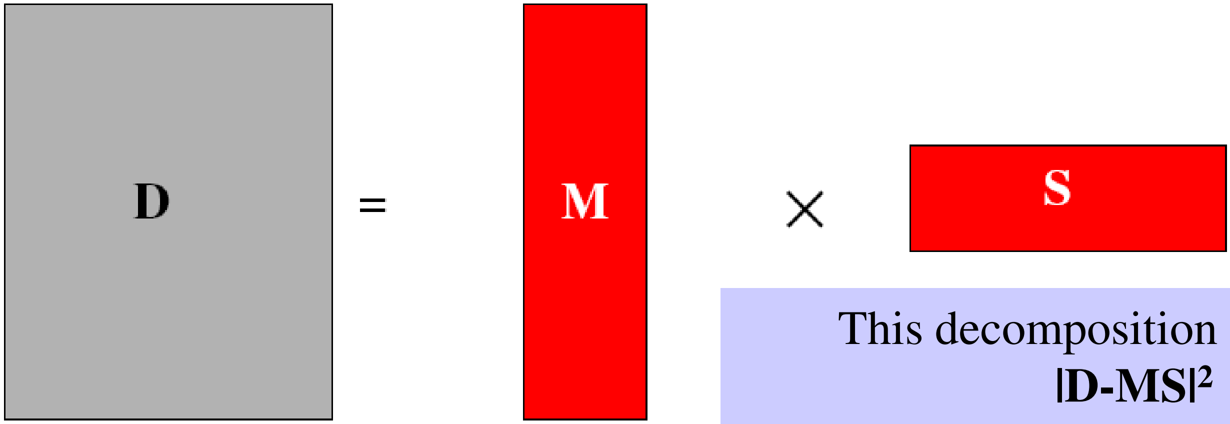
Factorizing the measurement matrix

- Obtaining a factorization from SVD:



Possible decomposition:

$$\mathbf{M} = \mathbf{U}_3 \mathbf{W}_3^{1/2} \qquad \mathbf{S} = \mathbf{W}_3^{1/2} \mathbf{V}_3^T$$

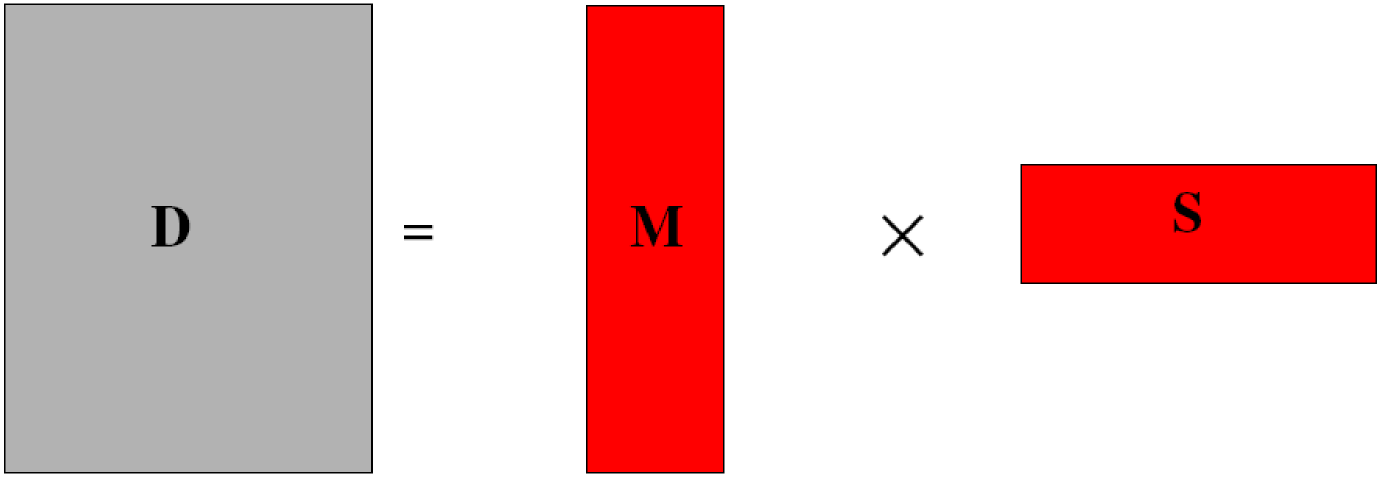


This decomposition minimizes
$|\mathbf{D}-\mathbf{MS}|^2$

Lazebnik

Source: M. Hebert

Affine ambiguity



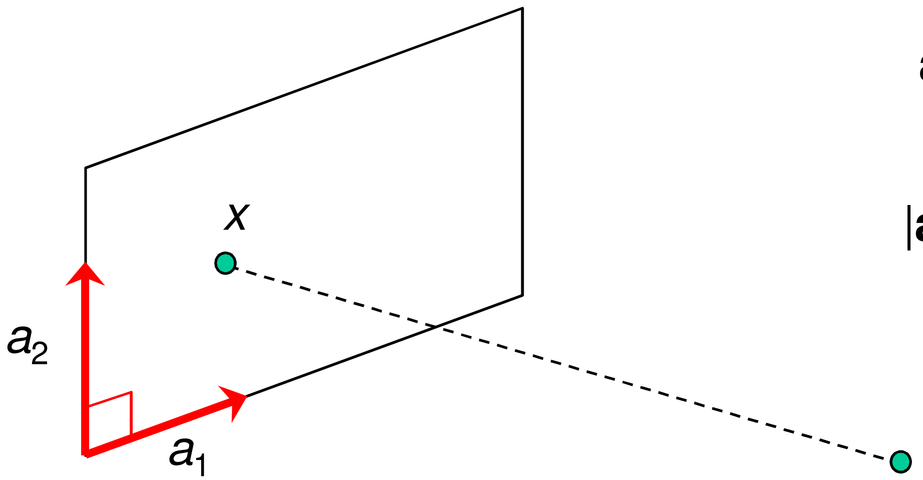
- The decomposition is not unique. We get the same **D** by using any 3×3 matrix **C** and applying the transformations **M → MC, S →C^{-1}S**
- That is because we have only an affine transformation and we have not enforced any Euclidean constraints (like forcing the image axes to be perpendicular, for example)

Lazebnik

Source: M. Hebert

Eliminating the affine ambiguity

- Orthographic: image axes are perpendicular and scale is 1



$$\mathbf{a}_1 \cdot \mathbf{a}_2 = 0$$

$$|\mathbf{a}_1|^2 = |\mathbf{a}_2|^2 = 1$$

- This translates into 3*m* equations in $\mathbf{L = CC^T}$:

$$\mathbf{A_i \, L \, A_i^T = Id}, \qquad i = 1, \ldots, m$$

  - Solve for **L**
  - Recover **C** from **L** by Cholesky decomposition: $\mathbf{L = CC^T}$
  - Update **M** and **S**: $\mathbf{M = MC, S = C^{-1}S}$

Lazebnik

Source: M. Hebert

Algorithm summary

- Given: *m* images and *n* features $\mathbf{x}_{ij}$
- For each image *i, c*enter the feature coordinates
- Construct a 2*m* × *n* measurement matrix **D**:
  - Column *j* contains the projection of point *j* in all views
  - Row *i* contains one coordinate of the projections of all the *n* points in image *i*
- Factorize **D**:
  - Compute SVD: $\mathbf{D} = \mathbf{U}\ \mathbf{W}\ \mathbf{V}^{\mathbf{T}}$
  - Create $\mathbf{U}_3$ by taking the first 3 columns of **U**
  - Create $\mathbf{V}_3$ by taking the first 3 columns of **V**
  - Create $\mathbf{W}_3$ by taking the upper left 3 × 3 block of **W**
- Create the motion and shape matrices:
  - $\mathbf{M} = \mathbf{U}_3\mathbf{W}_3^{½}$ and $\mathbf{S} = \mathbf{W}_3^{½}\ \mathbf{V}_3^{T}$ (**or** $\mathbf{M} = \mathbf{U}_3$ and $\mathbf{S} = \mathbf{W}_3\mathbf{V}_3^{T}$)
- Eliminate affine ambiguity

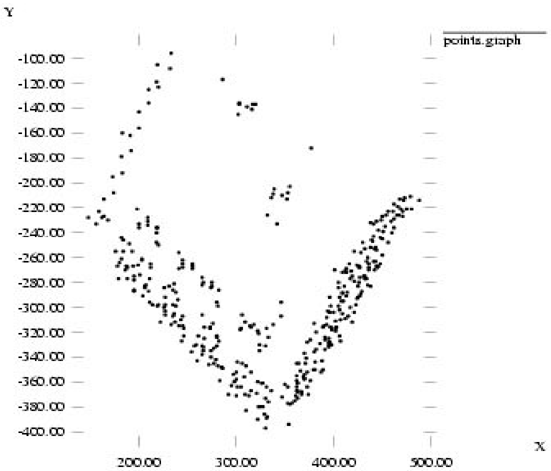
Lazebnik

Source: M. Hebert

Reconstruction results

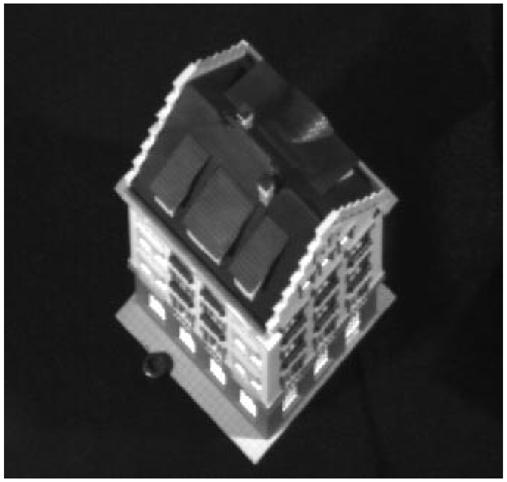
1

60



120

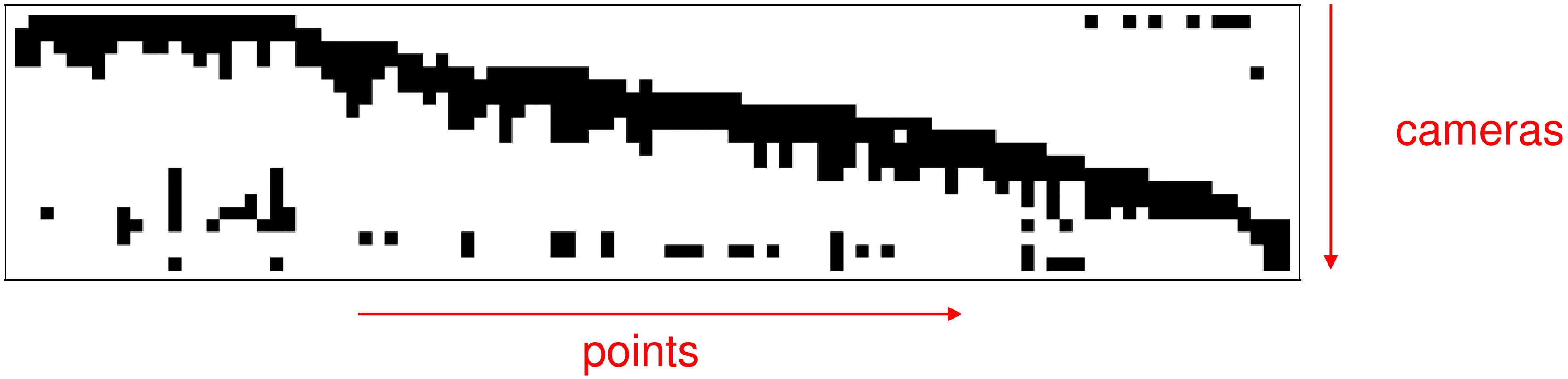
150



C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Dealing with missing data

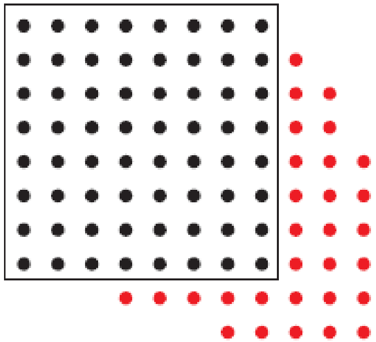
- So far, we have assumed that all points are visible in all views
- In reality, the measurement matrix typically looks something like this:



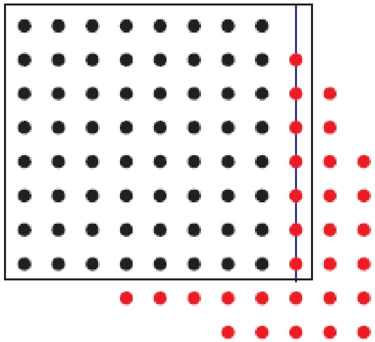
Lazebnik

Dealing with missing data

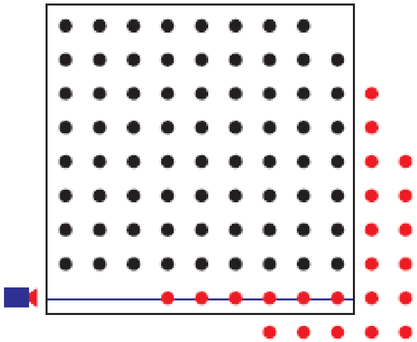
- Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results
  - Finding dense maximal sub-blocks of the matrix is NP-complete (equivalent to finding maximal cliques in a graph)
- Incremental bilinear refinement



(1) Perform factorization on a dense sub-block



(2) Solve for a new 3D point visible by at least two known cameras (linear least squares)



(3) Solve for a new camera that sees at least three known 3D points (linear least squares)

F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. Segmenting, Modeling, and Matching Video Clips Containing Multiple Moving Objects. PAMI 2007.

Further Factorization work

Factorization with uncertainty

(Irani & Anandan, IJCV'02)

Factorization for indep. moving objects (now)

(Costeira and Kanade '94)

Factorization for articulated objects (now)

(Yan and Pollefeys '05)

Factorization for dynamic objects (now)

(Bregler et al. 2000, Brand 2001)

Perspective factorization (next week)

(Sturm & Triggs 1996, …)

Factorization with outliers and missing pts.

(Jacobs '97 (affine), Martinek & Pajdla'01 Aanaes'02 (perspective))

Structure from motion of multiple moving objects

$$\mathcal{D}^{(i)} \overset{\text{def}}{=} \begin{pmatrix} \boldsymbol{p}_{11}^{(i)} & \dots & \boldsymbol{p}_{1n_i}^{(i)} \\ \dots & \dots & \dots \\ \boldsymbol{p}_{m1}^{(i)} & \dots & \boldsymbol{p}_{mn_i}^{(i)} \end{pmatrix} \qquad i = 1, \dots, k,$$

$$\mathcal{D}^{(i)} = \mathcal{M}^{(i)}\mathcal{P}^{(i)} \qquad \mathcal{M}^{(i)} \overset{\text{def}}{=} \begin{pmatrix} \mathcal{M}_1^{(i)} & \boldsymbol{o}_1^{(i)} \\ \dots & \dots \\ \mathcal{M}_m^{(i)} & \boldsymbol{o}_m^{(i)} \end{pmatrix} \qquad \mathcal{P}^{(i)} \overset{\text{def}}{=} \begin{pmatrix} \boldsymbol{P}_1^{(i)} & \dots & \boldsymbol{P}_{n_i}^{(i)} \\ 1 & \dots & 1 \end{pmatrix}$$

$$\mathcal{D} \overset{\text{def}}{=} (\mathcal{D}^{(1)}\mathcal{D}^{(2)} \dots \mathcal{D}^{(k)})$$

$$\mathcal{D} = \mathcal{M}\mathcal{P} \qquad \mathcal{M} \overset{\text{def}}{=} (\mathcal{M}^{(1)}\mathcal{M}^{(2)} \dots \mathcal{M}^{(k)}) \qquad \mathcal{P} \overset{\text{def}}{=} \begin{pmatrix} \mathcal{P}^{(1)} & 0 & \dots & 0 & 0 \\ 0 & \mathcal{P}^{(2)} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & \mathcal{P}^{(k)} \end{pmatrix}$$

$\mathcal{D}$ has rank $4k$ (or less)

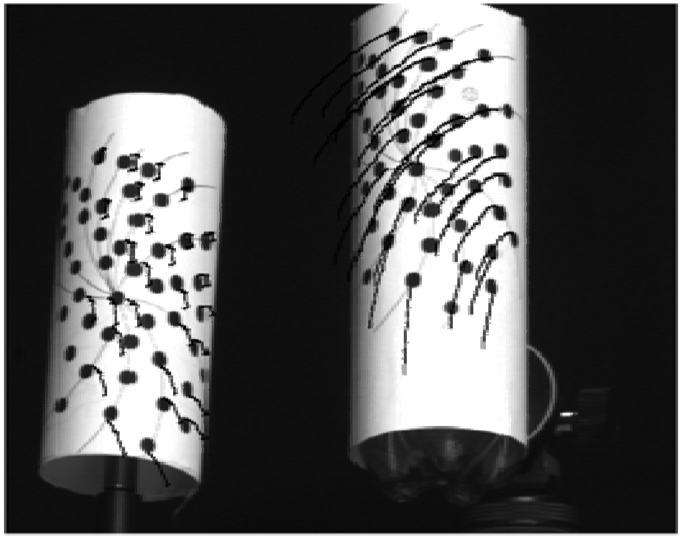
The Shape Interaction Matrix

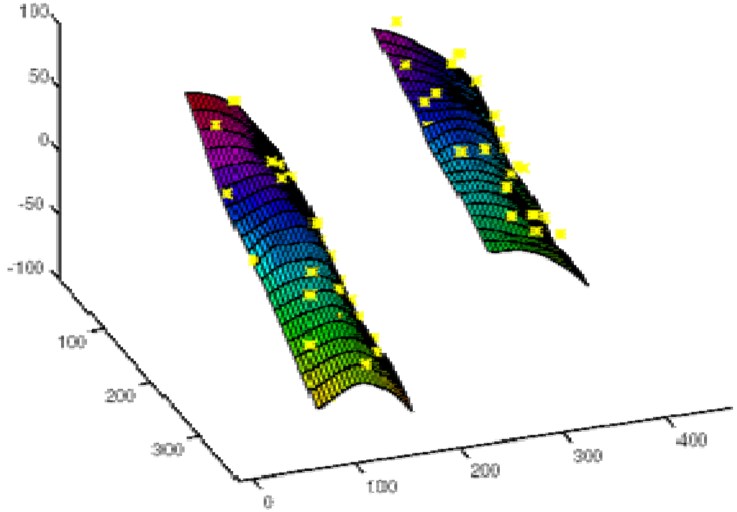
$$\mathcal{Q} \overset{\text{def}}{=} \mathcal{V}_r\mathcal{V}_r^T = \mathcal{P}^T(B^TB)\mathcal{P} \qquad C \overset{\text{def}}{=} (\mathcal{P}\mathcal{P}^T)^{-1}$$

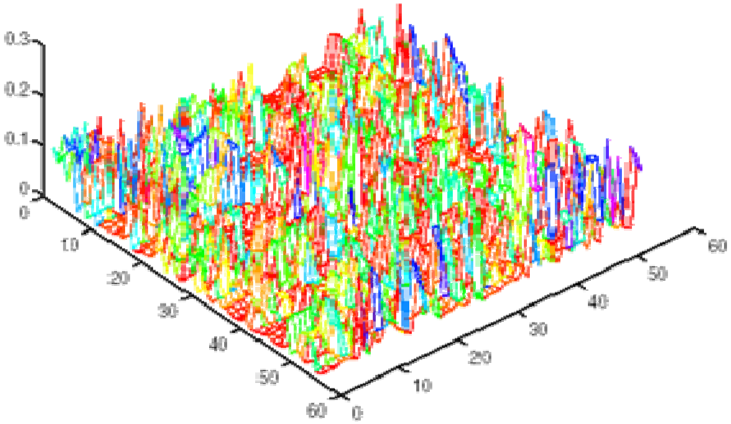
$$\mathcal{Q} = \begin{pmatrix} \mathcal{P}^{(1)T}C\mathcal{P}^{(1)} & 0 & \dots & 0 \\ 0 & \mathcal{P}^{(2)T}C\mathcal{P}^{(2)} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mathcal{P}^{(k)T}C\mathcal{P}^{(k)} \end{pmatrix}$$

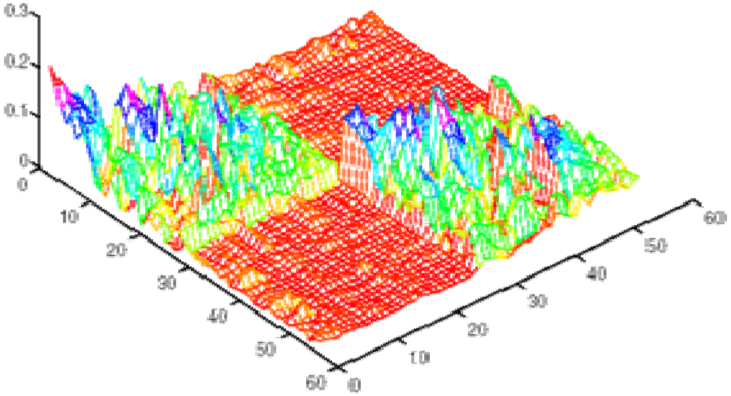
Pollefeys

Structure from motion of multiple moving objects







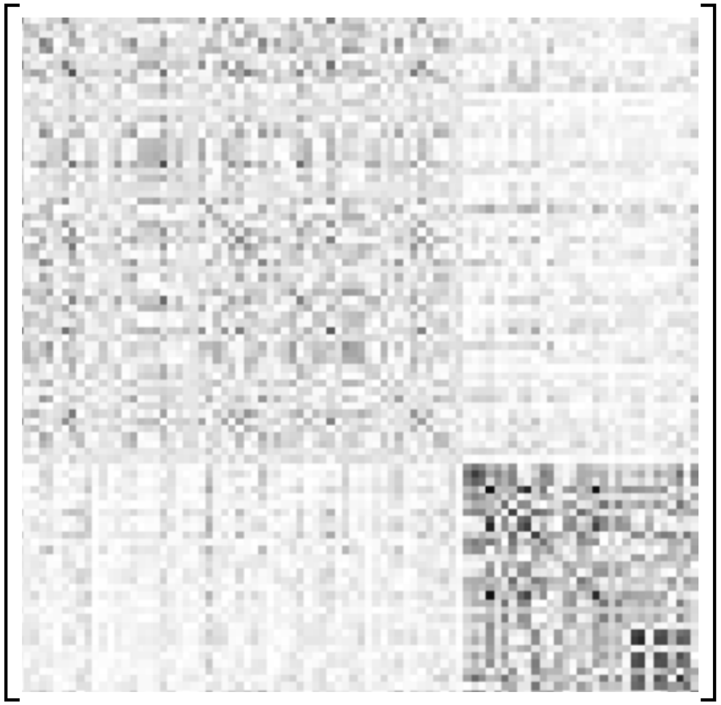


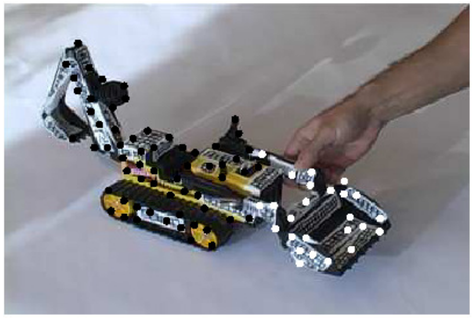
Pollefeys

Shape interaction matrix

Shape interaction matrix for articulated objects looses block diagonal structure

$$\mathbf{Q} = \mathbf{V}\mathbf{V}^\top =$$





Costeira and Kanade's approach is not usable for articulated bodies (assumes independent motions)

Pollefeys

Articulated motion subspaces

Motion subspaces for articulated bodies intersect

(Yan and Pollefeys, CVPR'05)

(Tresadern and Reid, CVPR'05)

Joint (1D intersection)

$$\underset{\text{(rank=8-1)}}{\mathbf{W}} = \begin{bmatrix} \mathbf{R}_1|\mathrm{t}_1 & \mathbf{R}_1\mathbf{R}'_1|\mathrm{t}_1 \\ \vdots & \vdots \\ \mathbf{R}_m|\underset{\uparrow}{\mathrm{t}_m} & \mathbf{R}_m\mathbf{R}'_m|\underset{\uparrow}{\mathrm{t}_m} \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 & 0 \\ 0 & \mathbf{S}_2 \end{bmatrix}$$

(joint=origin)

Hinge (2D intersection)

$$\underset{\text{(rank=8-2)}}{\mathbf{W}} = \begin{bmatrix} \mathbf{R}_1|\mathrm{t}_1 & \mathbf{R}_1\mathbf{R}'_1|\mathrm{t}_1 \\ \vdots & \vdots \\ \mathbf{R}_m|\mathrm{t}_m & \mathbf{R}_m\mathbf{R}'_m|\mathrm{t}_m \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 & 0 \\ 0 & \mathbf{S}_2 \end{bmatrix}$$

(hinge=z-axis)

$$\mathbf{R}'_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & \underset{\uparrow}{1} \end{bmatrix}$$

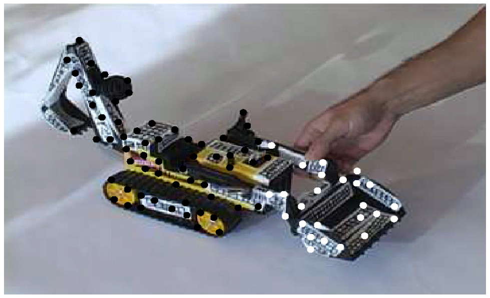
Exploit rank constraint to obtain better estimate

Also for non-rigid parts if $\sum_{i=1}^{K} c_i^f = 1$ (Yan & Pollefeys, 06?)

Results

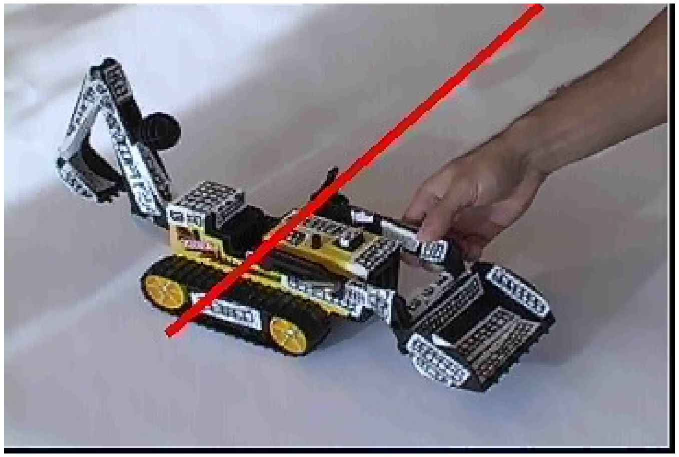
Toy truck

Segmentation



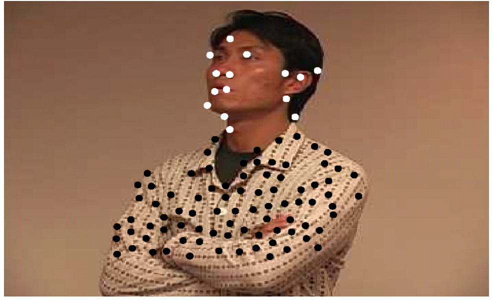
$(\mathbf{W}_1 \cap \mathbf{W}_2)$

Intersection



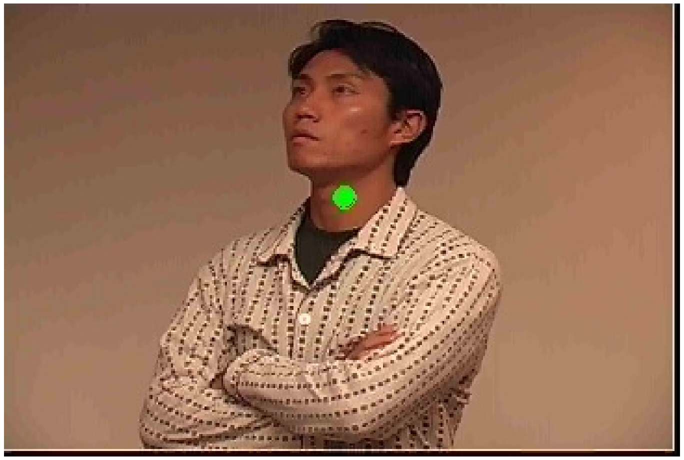
Student

Segmentation



$(\mathbf{W}_1 \cap \mathbf{W}_2)$

Intersection



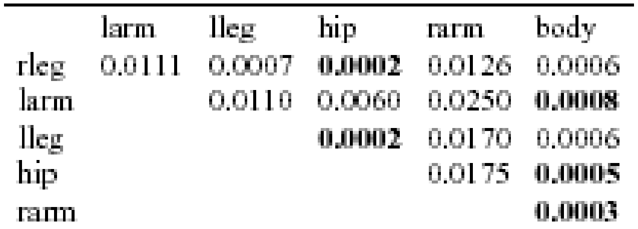
Pollefeys

Articulated shape and motion factorization

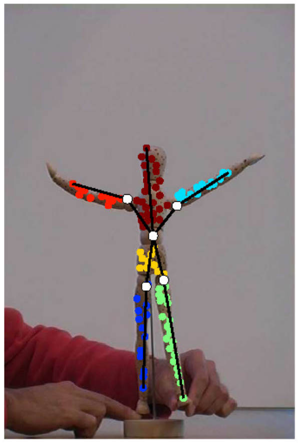
(Yan and Pollefeys, 2006?)

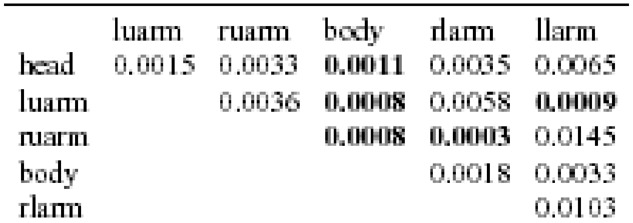
Automated kinematic chain building for articulated & non-rigid obj.

- Estimate principal angles between subspaces
- Compute affinities based on principal angles
- Compute minimum spanning tree

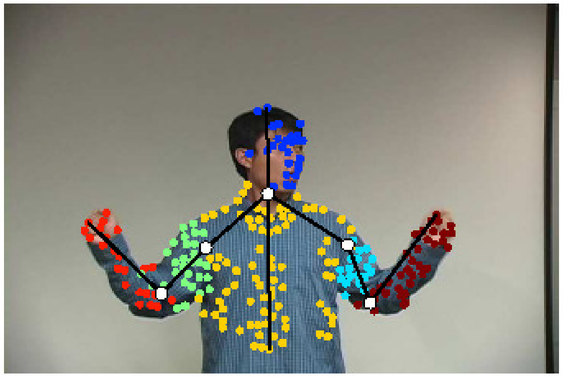


	larm	lleg	hip	rarm	body
rleg	0.0111	0.0007	**0.0002**	0.0126	0.0006
larm		0.0110	0.0060	0.0250	**0.0008**
lleg			**0.0002**	0.0170	0.0006
hip				0.0175	**0.0005**
rarm					**0.0003**





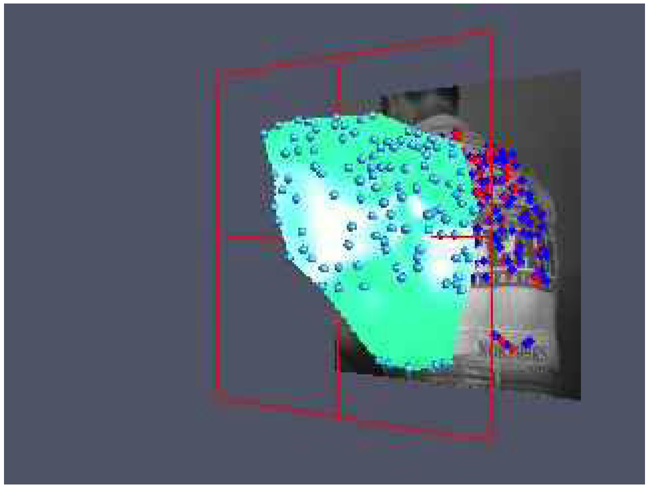
	luarm	ruarm	body	rlarm	llarm
head	0.0015	0.0033	**0.0011**	0.0035	0.0065
luarm		0.0036	**0.0008**	0.0058	**0.0009**
ruarm			**0.0008**	**0.0003**	0.0145
body				0.0018	0.0033
rlarm					0.0103



Structure from motion of deforming objects

(Bregler et al ’00;
Brand ‘01)

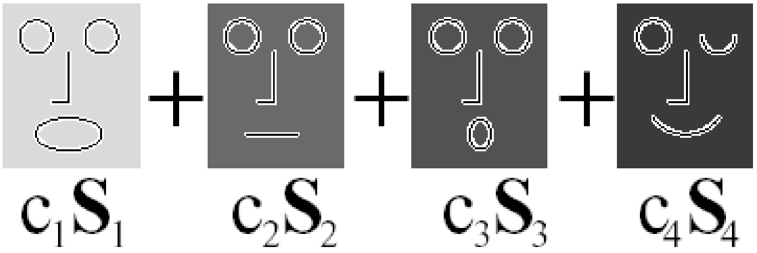
Extend factorization approaches to deal with dynamic shapes



Pollefeys

Representing dynamic shapes

$$S(t) = \sum_k c_k(t) S_k$$

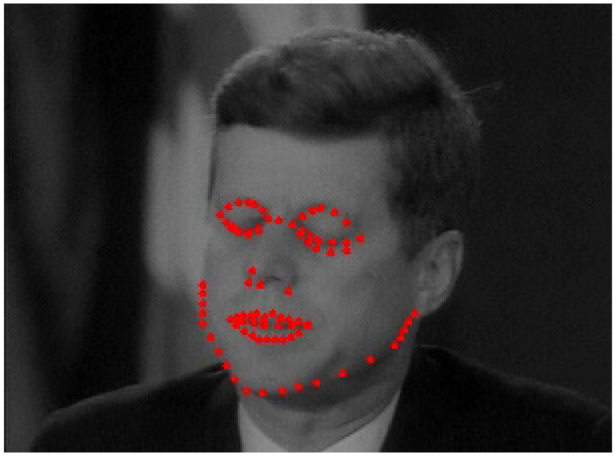


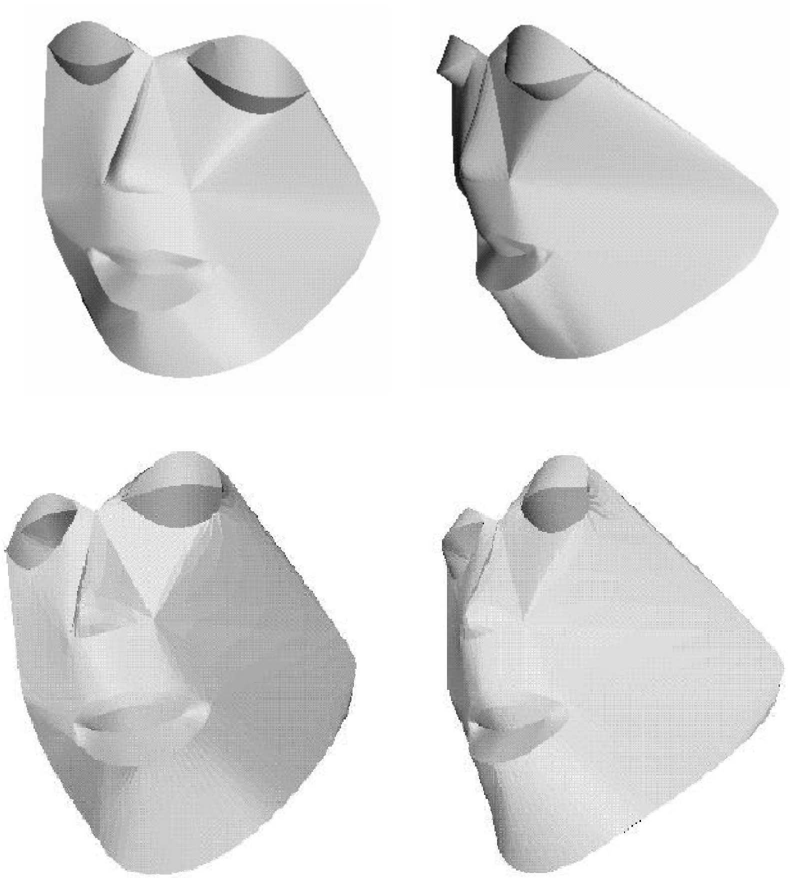
(fig. M.Brand)

represent dynamic shape as
varying linear combination of basis shapes

Results

(Bregler et al ’00)

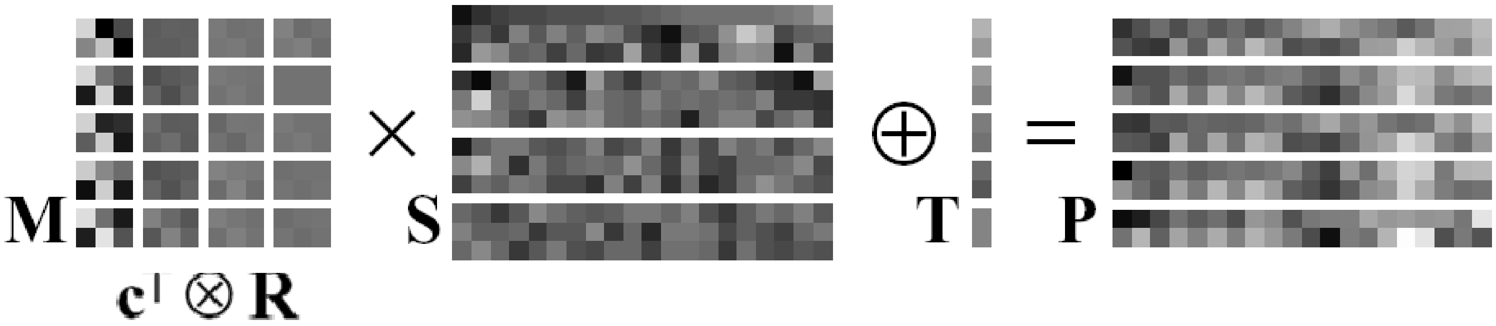




Pollefeys

Dynamic SfM factorization

(Brand '01)



constraints to be satisfied for M

$$\forall_{\hat{\mathbf{M}}_f \in \hat{\mathbf{M}}} \quad (\operatorname*{vec}_3 \hat{\mathbf{M}}_f^\top)^\top (\operatorname*{vec}_3 \hat{\mathbf{M}}_f^\top) = \frac{1}{D}\mathbf{I}_D \otimes ((\operatorname*{vec}_{3D} \hat{\mathbf{M}}_f)^\top (\operatorname*{vec}_{3D} \hat{\mathbf{M}}_f))$$

constraints to be satisfied for M, use to compute J

$\hat{\mathbf{M}} = \tilde{\mathbf{M}}\mathbf{J}^{-1}$ hard!

(different methods are possible,
not so simple and also not optimal)

Pollefeys

Non-rigid 3D subspace flow

Same is also possible using optical flow in stead of features, also takes uncertainty into account

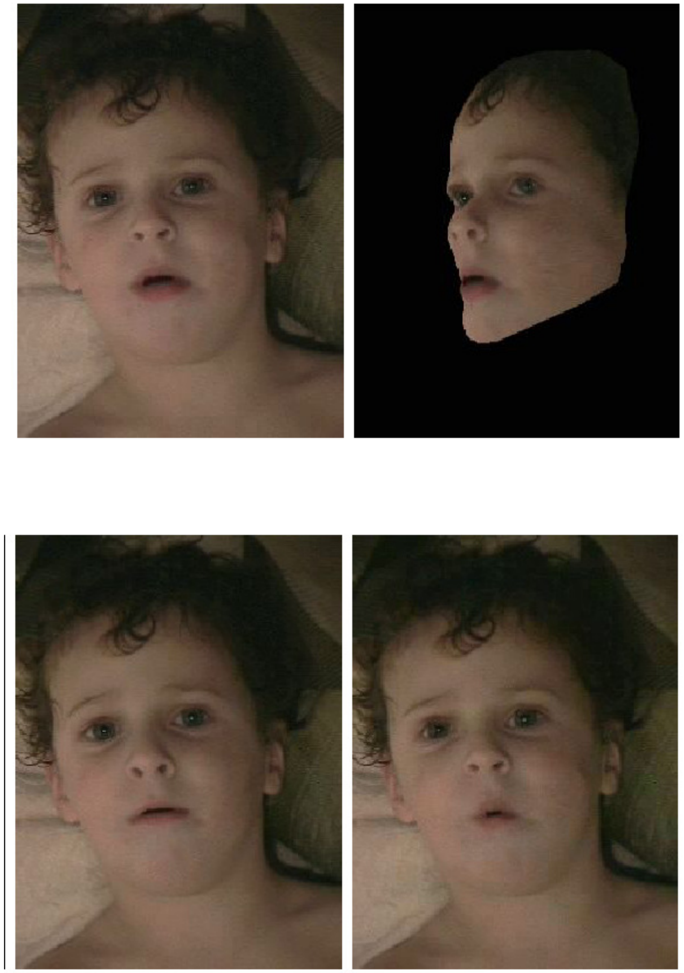
(Brand '01)

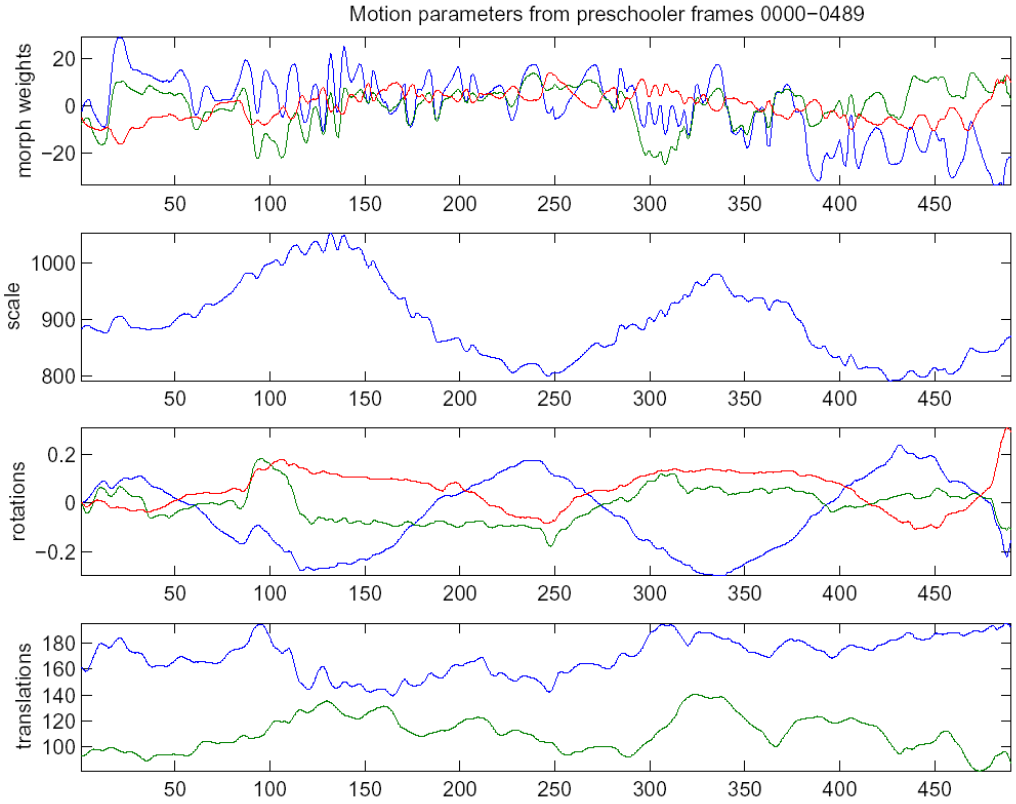
$$\begin{bmatrix} \sum_w I_x * I_x & \sum_w I_x * I_y \\ \sum_w I_x * I_y & \sum_w I_x * I_x \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum_w I_x * I_t \\ \sum_w I_y * I_t \end{bmatrix}$$

Pollefeys

Results

(Brand '01)

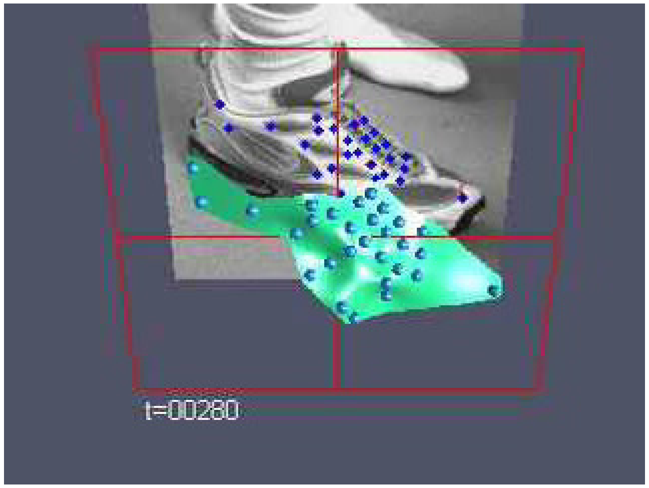


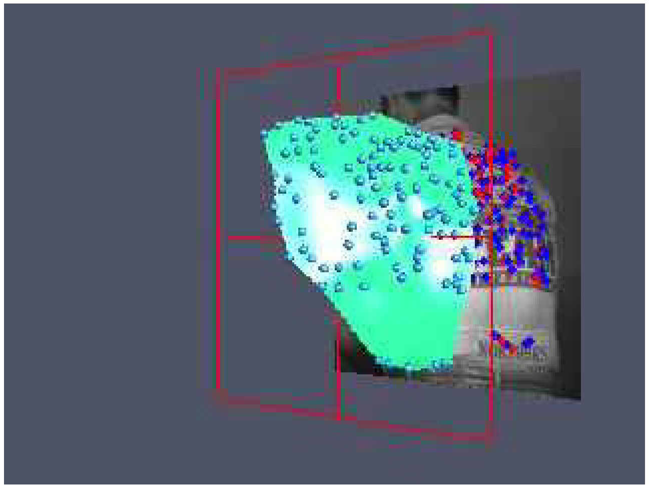


Pollefeys

Results

(Bregler et al ’01)



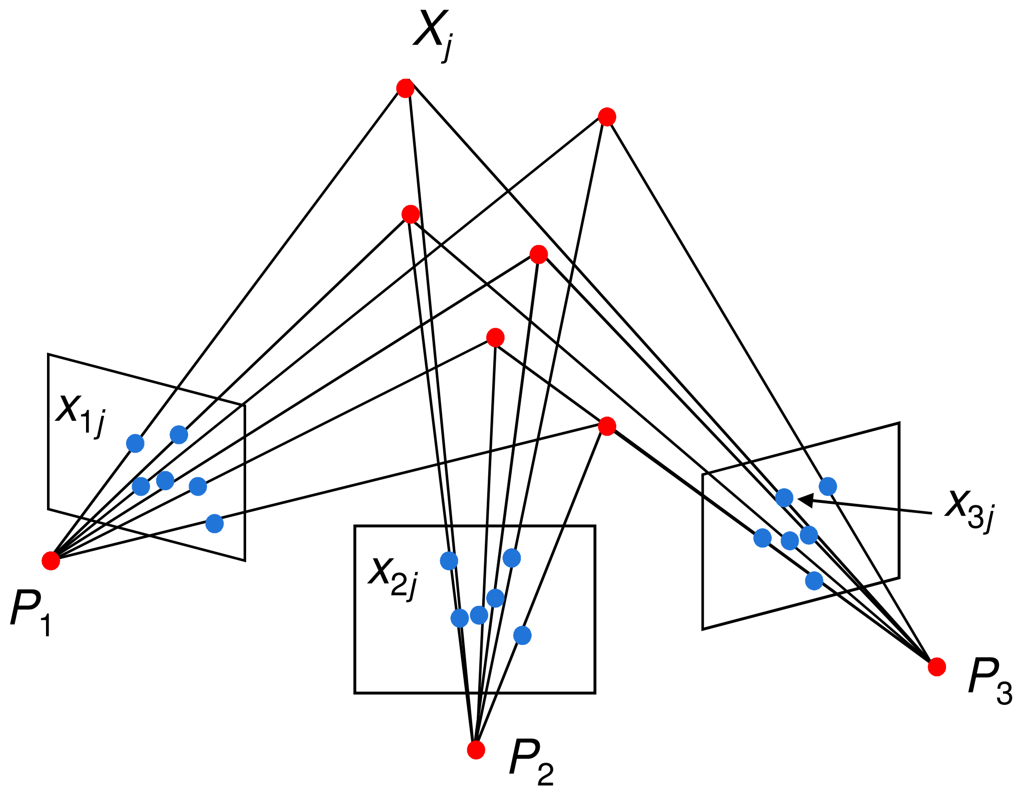


Projective structure from motion

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

$$z_{ij}\,\mathbf{x}_{ij} = \mathbf{P}_i\,\mathbf{X}_j\,,\quad i = 1,\dots,m,\quad j = 1,\dots,n$$

- Problem: estimate *m* projection matrices $\mathbf{P}_i$ and *n* 3D points $\mathbf{X}_j$ from the *mn* correspondences $\mathbf{x}_{ij}$



Lazebnik

Projective structure from motion

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

$$z_{ij}\,\mathbf{x}_{ij} = \mathbf{P}_i\,\mathbf{X}_j\,, \quad i = 1,\dots, m, \quad j = 1, \dots, n$$

- Problem: estimate *m* projection matrices $\mathbf{P}_i$ and *n* 3D points $\mathbf{X}_j$ from the *mn* correspondences $\mathbf{x}_{ij}$
- With no calibration info, cameras and points can only be recovered up to a 4x4 projective transformation **Q**:

$$\mathbf{X} \rightarrow \mathbf{QX},\ \mathbf{P} \rightarrow \mathbf{PQ}^{-1}$$

- We can solve for structure and motion when

$$2mn >= 11m + 3n – 15$$

- For two cameras, at least 7 points are needed

Lazebnik

Projective SFM: Two-camera case

- Compute fundamental matrix $\mathbf{F}$ between the two views
- First camera matrix: $[\mathbf{I}|\mathbf{0}]$
- Second camera matrix: $[\mathbf{A}|\mathbf{b}]$
- Then $z\mathbf{x} = [\mathbf{I} \mid \mathbf{0}]\mathbf{X}, \quad z'\mathbf{x}' = [\mathbf{A} \mid \mathbf{b}]\mathbf{X}$

$$z'\mathbf{x}' = \mathbf{A}[\mathbf{I} \mid \mathbf{0}]\mathbf{X} + \mathbf{b} = z\mathbf{A}\mathbf{x} + \mathbf{b}$$

$$z'\mathbf{x}' \times \mathbf{b} = z\mathbf{A}\mathbf{x} \times \mathbf{b}$$

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

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

$\mathbf{F} = [\mathbf{b}_\times]\mathbf{A}$ $\mathbf{b}$: epipole ($\mathbf{F}^{\mathrm{T}}\mathbf{b} = 0$), $\mathbf{A} = -[\mathbf{b}_\times]\mathbf{F}$

Lazebnik

F&P sec. 13.3.1

Projective factorization

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

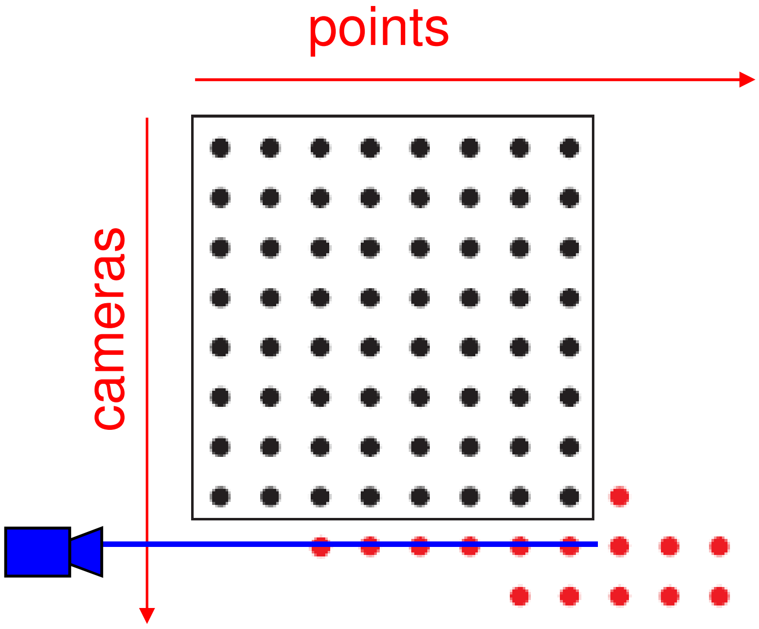
D = MS has rank 4

- 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)

Lazebnik

Sequential structure from motion

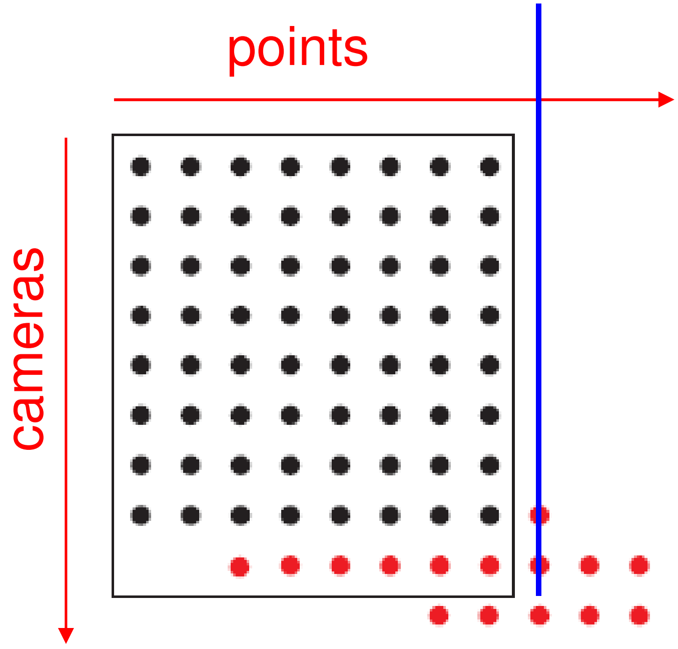
- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
    - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*



Lazebnik

Sequential structure from motion

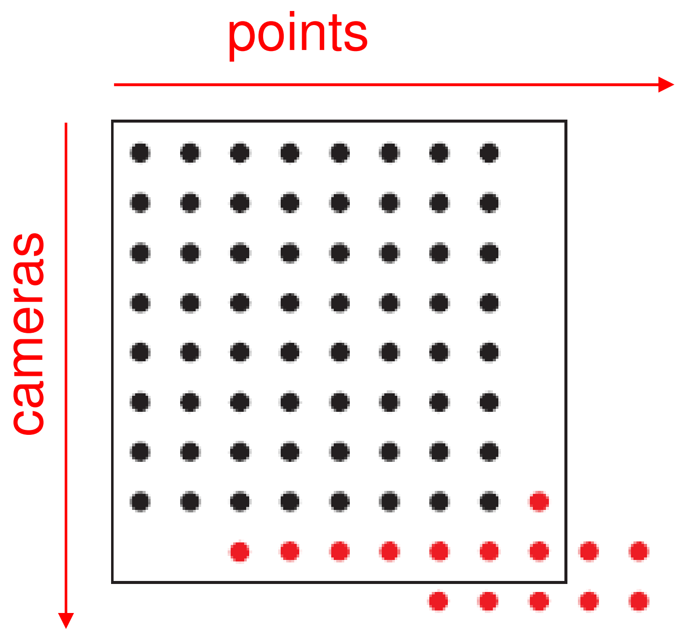
- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
    - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*
    - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – *triangulation*



Lazebnik

Sequential structure from motion

- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – *triangulation*
- Refine structure and motion: bundle adjustment

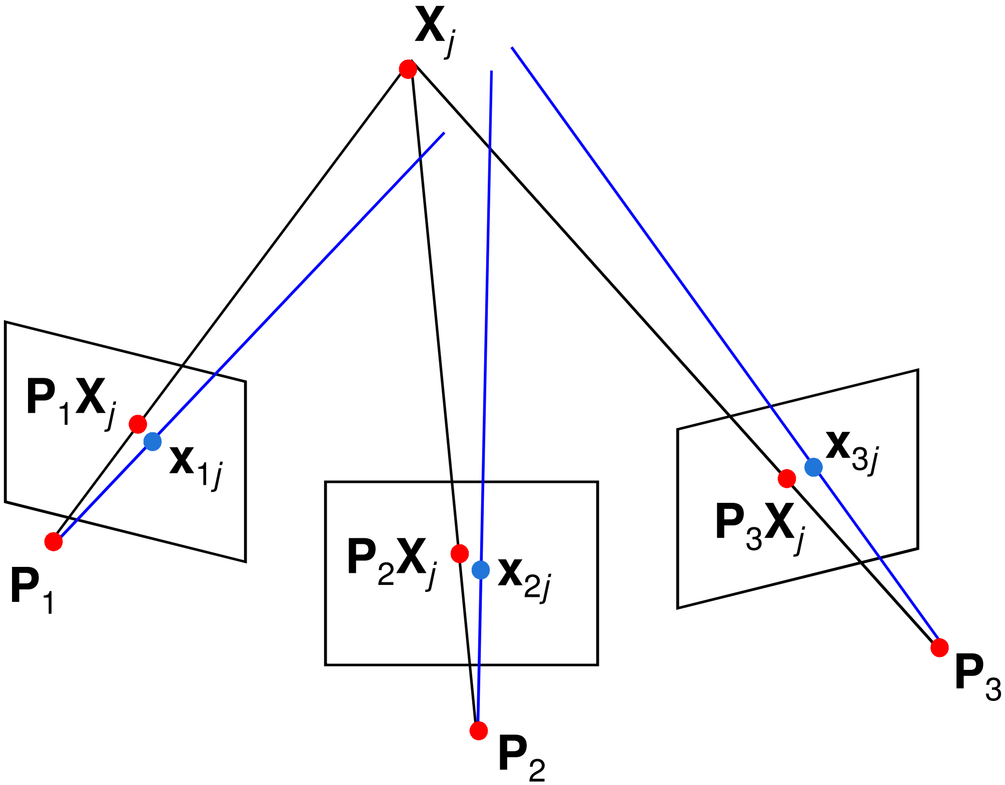


Lazebnik

Bundle adjustment

- Non-linear method for refining structure and motion
- Minimizing reprojection error

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



Lazebnik

Self-calibration

- Self-calibration (auto-calibration) is the process of determining intrinsic camera parameters directly from uncalibrated images
- For example, when the images are acquired by a single moving camera, we can use the constraint that the intrinsic parameter matrix remains fixed for all the images
    - Compute initial projective reconstruction and find 3D projective transformation matrix **Q** such that all camera matrices are in the form $\mathbf{P}_i = \mathbf{K}\,[\mathbf{R}_i \mid \mathbf{t}_i]$
- Can use constraints on the form of the calibration matrix: zero skew

Lazebnik

Summary: Structure from motion

- Ambiguity
- Affine structure from motion: factorization
- Dealing with missing data
- Projective structure from motion: two views
- Projective structure from motion: iterative factorization
- Bundle adjustment
- Self-calibration

Lazebnik

(Included presentation…available from http://phototour.cs.washington.edu/)

Photo Tourism:

Exploring Photo Collections in 3D

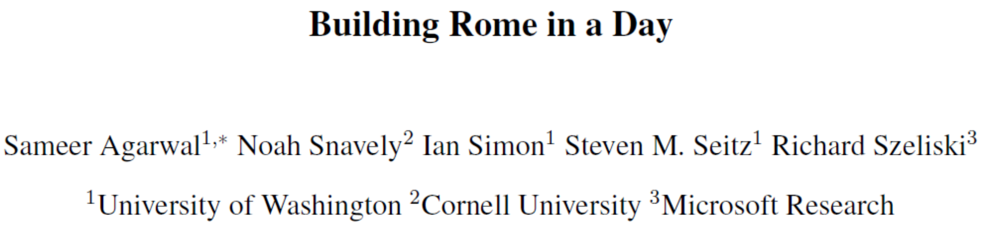
Noah Snavely

Steven M. Seitz

University of Washington

Richard Szeliski

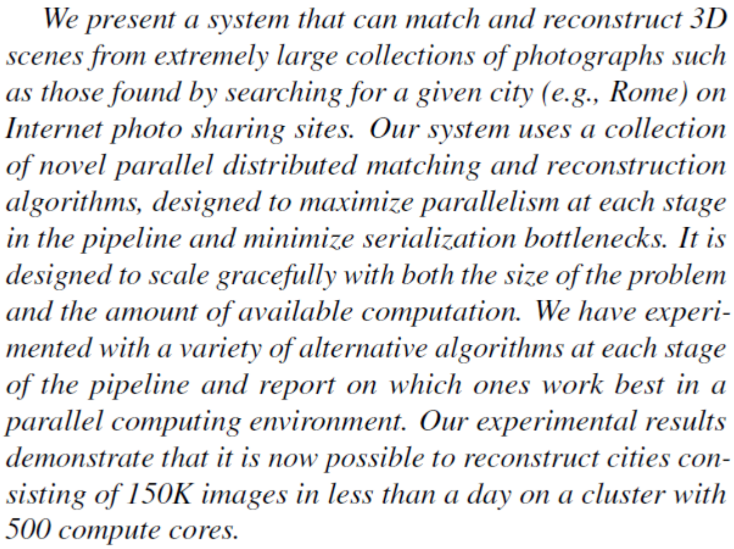
Microsoft Research



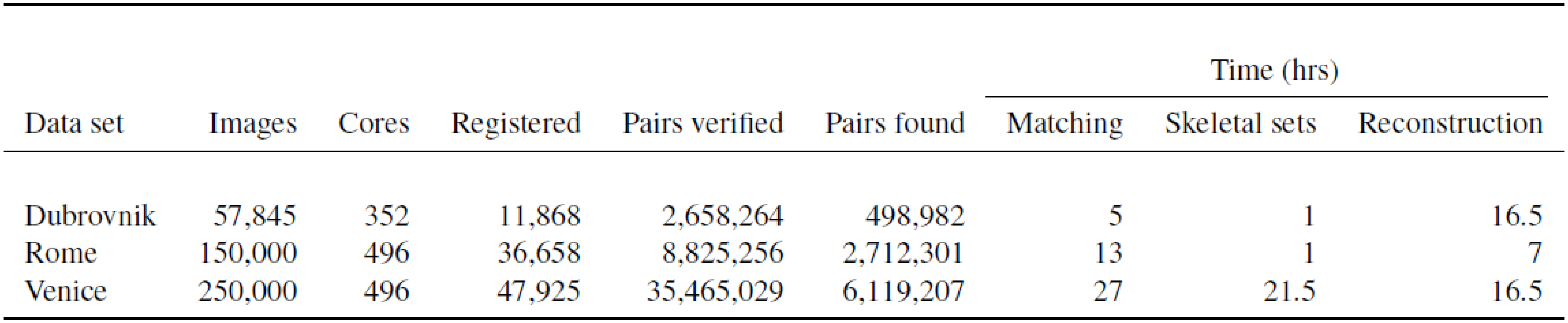
Building Rome in a Day

Sameer Agarwal[1,*] Noah Snavely[2] Ian Simon[1] Steven M. Seitz[1] Richard Szeliski[3]

[1]University of Washington [2]Cornell University [3]Microsoft Research



We present a system that can match and reconstruct 3D scenes from extremely large collections of photographs such as those found by searching for a given city (e.g., Rome) on Internet photo sharing sites. Our system uses a collection of novel parallel distributed matching and reconstruction algorithms, designed to maximize parallelism at each stage in the pipeline and minimize serialization bottlenecks. It is designed to scale gracefully with both the size of the problem and the amount of available computation. We have experimented with a variety of alternative algorithms at each stage of the pipeline and report on which ones work best in a parallel computing environment. Our experimental results demonstrate that it is now possible to reconstruct cities consisting of 150K images in less than a day on a cluster with 500 compute cores.



						Time (hrs)		
Data set	Images	Cores	Registered	Pairs verified	Pairs found	Matching	Skeletal sets	Reconstruction
Dubrovnik	57,845	352	11,868	2,658,264	498,982	5	1	16.5
Rome	150,000	496	36,658	8,825,256	2,712,301	13	1	7
Venice	250,000	496	47,925	35,465,029	6,119,207	27	21.5	16.5

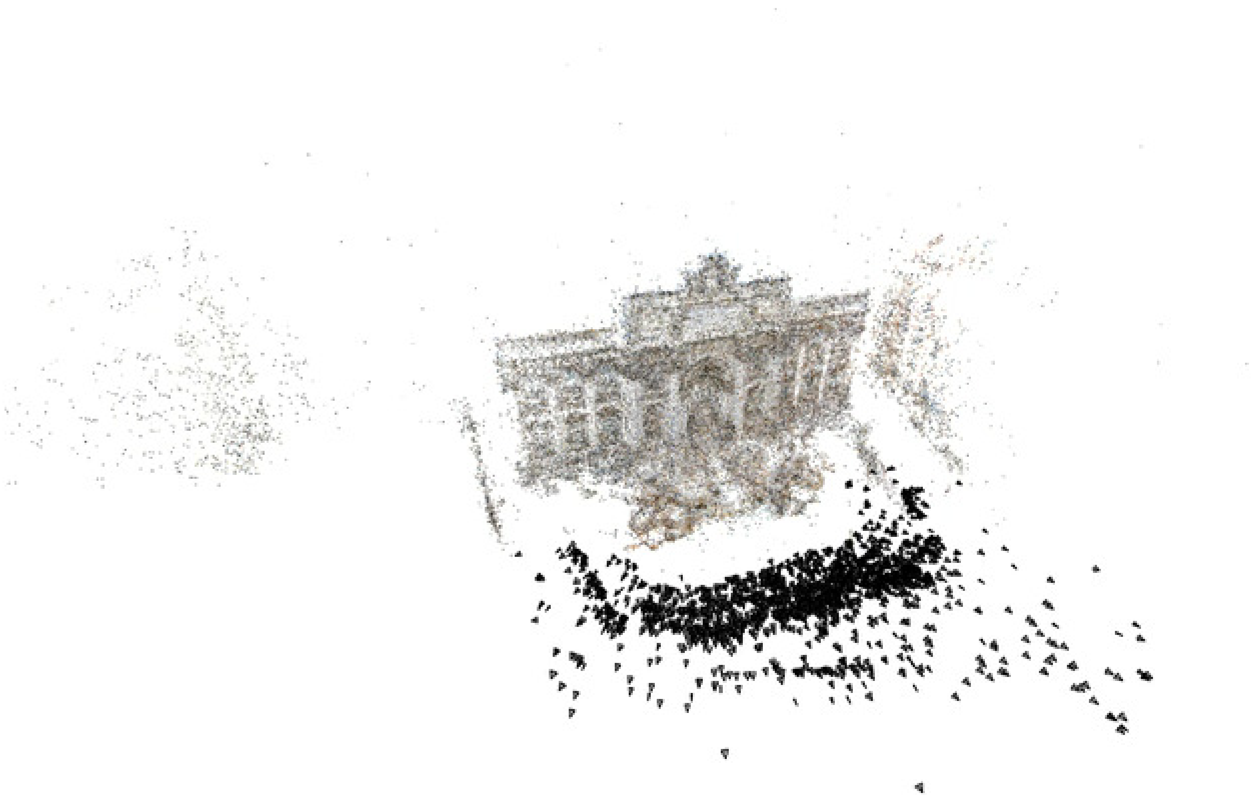
Table 1. Matching and reconstruction statisics for the three data sets.

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

“Rome in a day”: Coliseum video

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

“Rome in a day”: Trevi video



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

“Rome in a day”: St. Peters video

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

Slide Credits

- Svetlana Lazebnik
- Marc Pollefeys
- Noah Snaveley & co-authors

Today: SFM

- SFM problem statement
- Factorization
- Projective SFM
- Bundle Adjustment
- Photo Tourism
- “Rome in a day:

Roadmap

- Previous: Image formation, filtering, local features, (Texture)…
- Tues: Feature-based Alignment
  - Stitching images together
  - Homographies, RANSAC, Warping, Blending
  - Global alignment of planar models
- Today: Dense Motion Models
  - Local motion / feature displacement
  - Parametric optic flow
- No classes next week: ICCV conference
- Oct 6th: Stereo / 'Multi-view': Estimating depth with known inter-camera pose
- **Oct 8th: 'Structure-from-motion': Estimation of pose and 3D structure**
  - **Factorization approaches**
  - **Global alignment with 3D point models**