计算机视觉与模式识别

Computer Vision and Pattern Recognition

-- Geometric Vision

Structure from Motion:

Affine structure & Projective structure from Motion / Bundle adjustment

人工智能与机器人研究所

Institute of Artificial Intelligence and Robotics

袁泽剑

Email:yuan.ze.jian@xjtu.edu.cn

科学馆102室

Geometric Vision

- **3D ⟶ 2D** (Projective transform, Camera Model & Calibration)

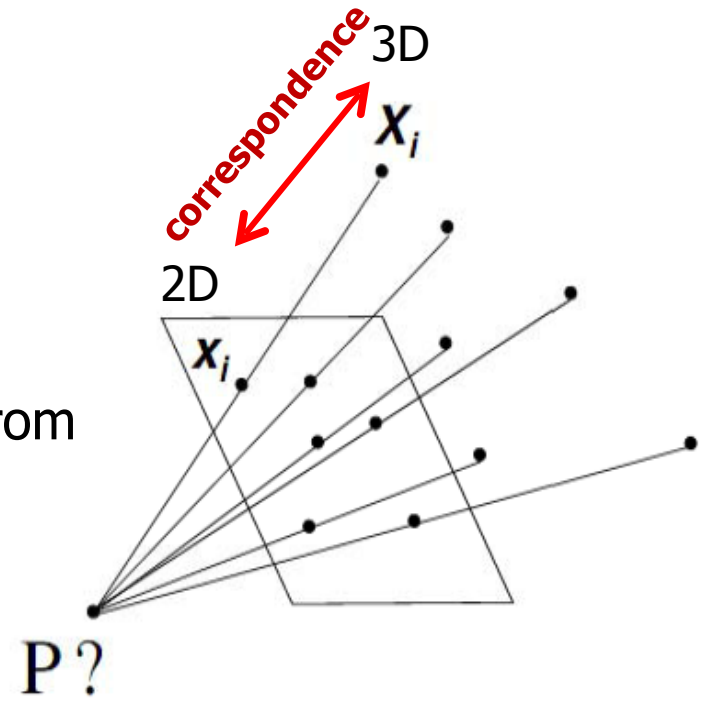
$$\lambda \mathbf{x}_i = \mathbf{P}\mathbf{X}_i$$

- **Problem:**

$$\{(\mathbf{x}_i, \mathbf{X}_i)\}_{i=1\cdots N} \Rightarrow \mathbf{P}_{3\times 4}$$

- Only two linearly independent equations from one pair of correspondent point ($\mathbf{x_i}$, $\mathbf{X_i}$)

$$\begin{bmatrix} 0 & -\mathbf{X}_i^T & y_i\mathbf{X}_i^T \\ \mathbf{X}_i^T & 0 & -x_i\mathbf{X}_i^T \\ -y_i\mathbf{X}_i^T & x_i\mathbf{X}_i^T & 0 \end{bmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = 0$$



Camera Calibration:

✓Camera pose **R, t**

✓Camera parameter **K**

Geometric Vision

- **2D → 3D** (Projective transform, Reconstruction)

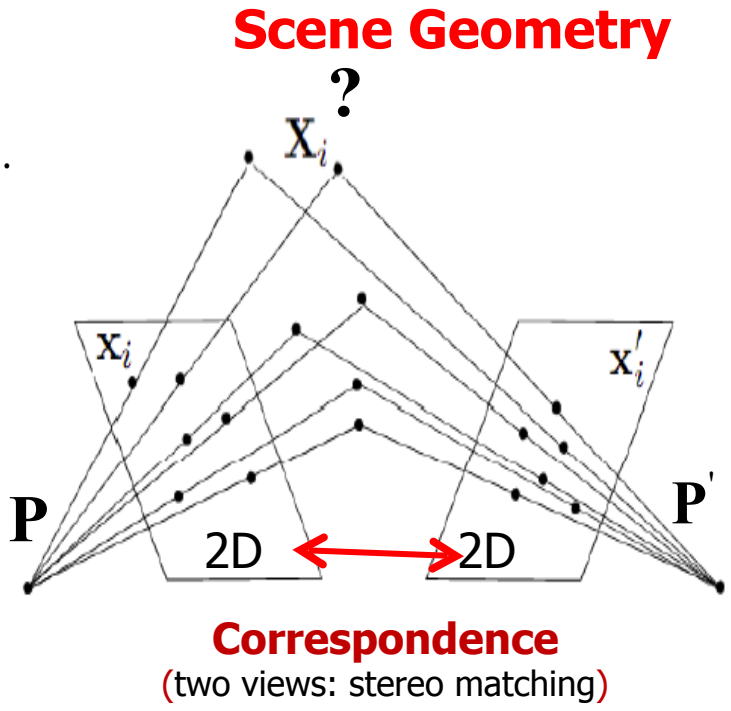
$$\lambda \mathbf{x}_i = \mathbf{P}\mathbf{X}_i \qquad \lambda' \mathbf{x}'_i = \mathbf{P}'\mathbf{X}_i$$

- **Problem:**

$$\mathbf{P}, \mathbf{P}', \{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1\cdots N} \Rightarrow \{\mathbf{X}_i\}_{i=1..}$$

- 4 linearly independent equations from one pair of correspondent point ($\mathbf{x_i}$, $\mathbf{x_i}'$)

$$\begin{bmatrix} \mathbf{x}_{i\times}\mathbf{P} \\ \mathbf{x}'_{i\times}\mathbf{P}' \end{bmatrix} \mathbf{X}_i = 0$$



Geometric Vision

- 2D ↔ 2D (Epipolar constraints, **Stereo Correspondence** / Matching)

$$\mathbf{x}_i^T \mathbf{E} \mathbf{x}_i^{'} = 0$$

- **Problem:**

$$\mathbf{E} \Rightarrow \left\{ \left(\mathbf{p}, \mathbf{p}^{'} \right) \right\}_{\mathbf{p} \in \Omega_l, \mathbf{p}^{'} \in \Omega_r}$$

Stereo Correspondence

- Epipolar constraints for correspondent point (**p, p'**)
  - **Stereo calibration:**
    - ✓ calibrate camera (P, P')
    - ✓ rectify image (**H, H'**)

$$\left.\begin{array}{l} \mathbf{E} = [\mathbf{t}_\times]\mathbf{R} \\ \mathbf{R} = \mathbf{I} \\ \mathbf{t} = [-d \quad 0 \quad 0] \end{array}\right\} \Rightarrow y = y^{'}$$

  - **Additional constraints**
    - ✓ Similarity
    - ✓ Uniqueness
    - ✓ Ordering
    - ✓ Disparity gradient

=> Disparity & Depth

Geometric Vision

- 2D ↔ 2D (Epipolar constraints, Camera **Relative Pose / Motion**)

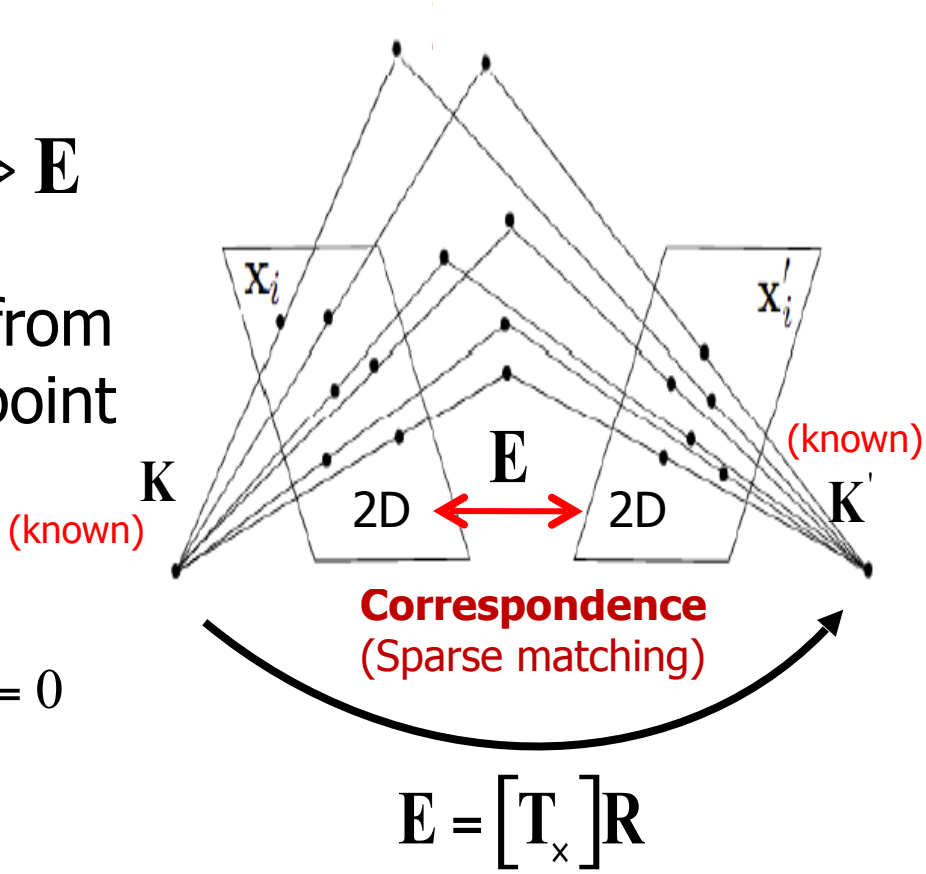
$$\mathbf{x}_i^T \mathbf{E} \mathbf{x}_i^{'} = 0$$

- **Problem:**

$$\left\{\left(\mathbf{x}_i, \mathbf{x}_i^{'}\right)\right\}_{i=1\cdots N} \Rightarrow \mathbf{E}$$

- Only one linearly equation from one pair of correspondent point ($\mathbf{x_i}$, $\mathbf{x_i'}$)

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = 0$$



Geometric Vision

- 2D ↔ 2D (Epipolar constraints, Camera **Weak Calibration**)

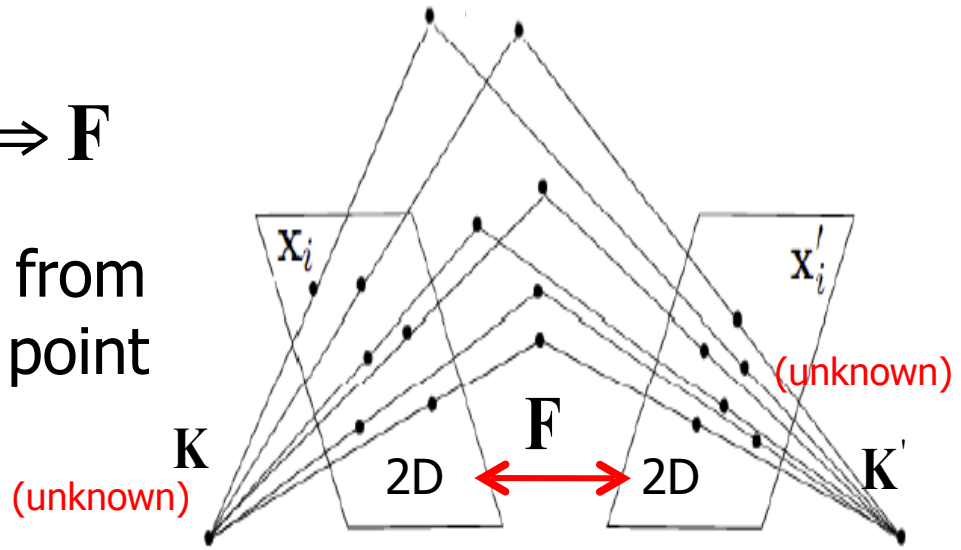
$$\mathbf{x}_i^T \mathbf{F} \mathbf{x}_i^{'} = 0$$

- **Problem:**

$$\left\{\left(\mathbf{x}_i, \mathbf{x}_i^{'}\right)\right\}_{i=1\cdots N} \Rightarrow \mathbf{F}$$

- Only one linearly equation from one pair of correspondent point ($\mathbf{x_i}$, $\mathbf{x_i}'$)

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



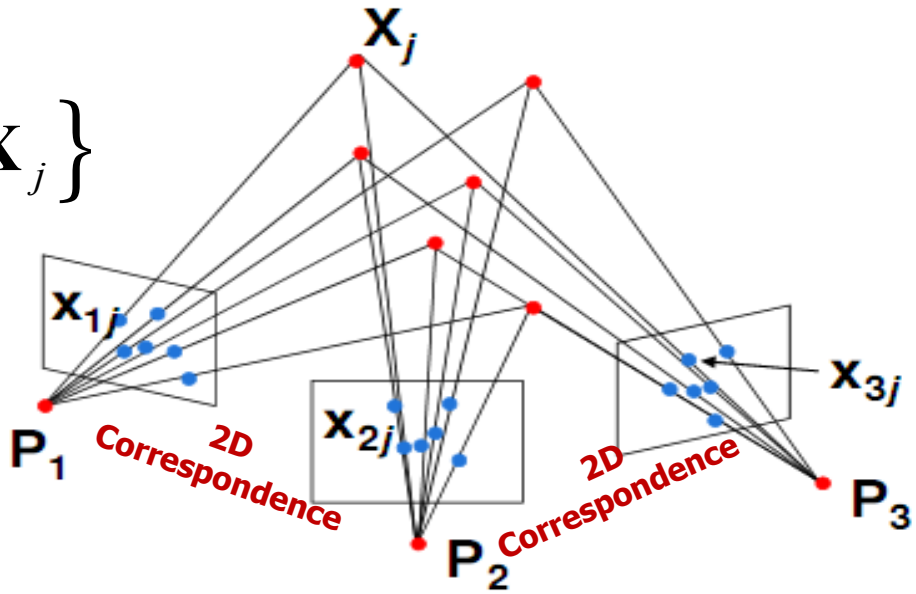
Correspondence
(Sparse matching)

$$\mathbf{F} = \mathbf{K}^{-T}\mathbf{E}\mathbf{K}'^{-1}$$

Geometric Vision

- 2D ⟶ 3D (Reconstruction with unknown projective matrix)
  - **Problem:**

$$\{\mathbf{x}_{ij}\} \Rightarrow \{\mathbf{P}_i\}, \{\mathbf{X}_j\}$$



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

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

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

----→ **Structure from Motion**

Structure from Motion

- Structure from Motion (SfM)
  - Motivation
  - Ambiguity
- **Affine SfM**
  - Affine cameras
  - **Affine factorization**
  - Euclidean upgrade
  - Dealing with missing data
- **Projective SfM**
  - Two-camera case
  - **Projective factorization**
  - Bundle adjustment
  - Practical considerations
- **Applications**

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:

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

$$\mathbf{x}_{ij} = \left(\frac{1}{k}\mathbf{P}_i\right)(k\mathbf{X}_j), \quad \forall i, j$$

$\Rightarrow$ It is impossible to recover the absolute scale of the scene!

Slide credit: Svetlana 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.

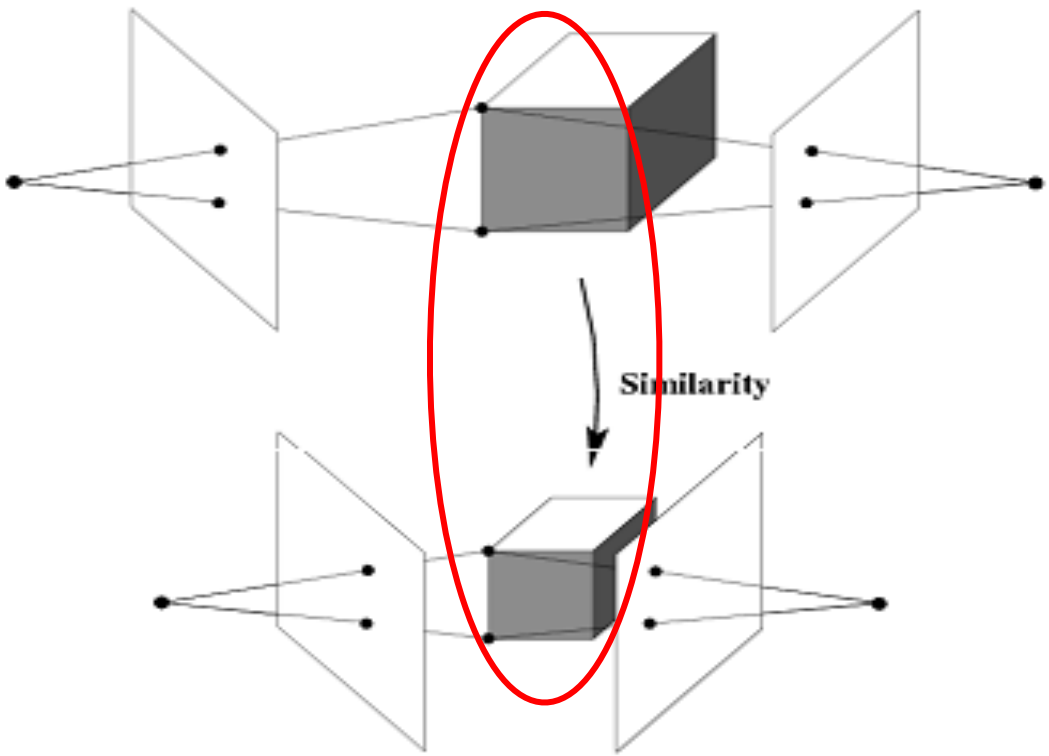
- More generally: if we transform the scene using a transformation $\mathbf{Q}$ and apply the inverse transformation to the camera matrices, then the images do not change

$$\mathbf{x} = \mathbf{PX} = (\mathbf{PQ}^{-1})\mathbf{QX}$$

$$\mathbf{x}_{ij} = \left(\mathbf{P}_i\mathbf{Q}^{-1}\right)\mathbf{QX}_j, \quad \forall i, j$$

Slide credit: Svetlana Lazebnik

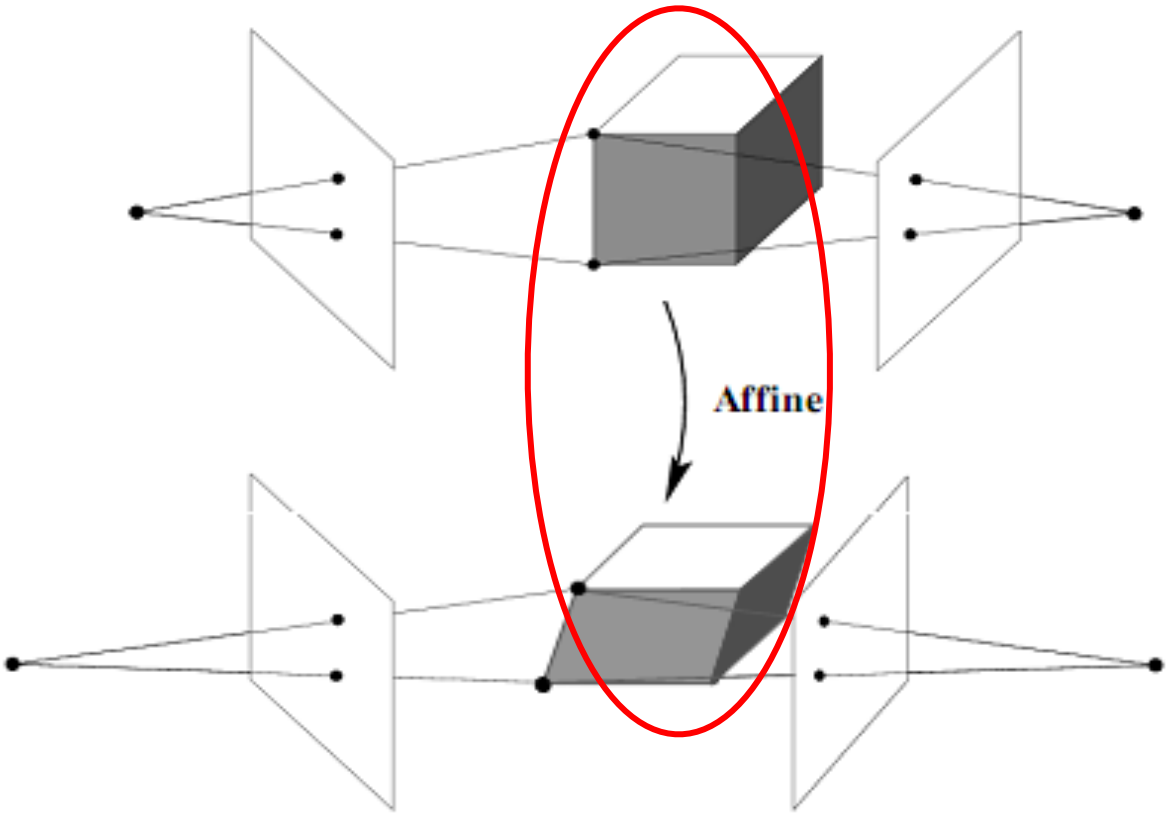
Reconstruction Ambiguity: Similarity



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

Slide credit: Svetlana Lazebnik

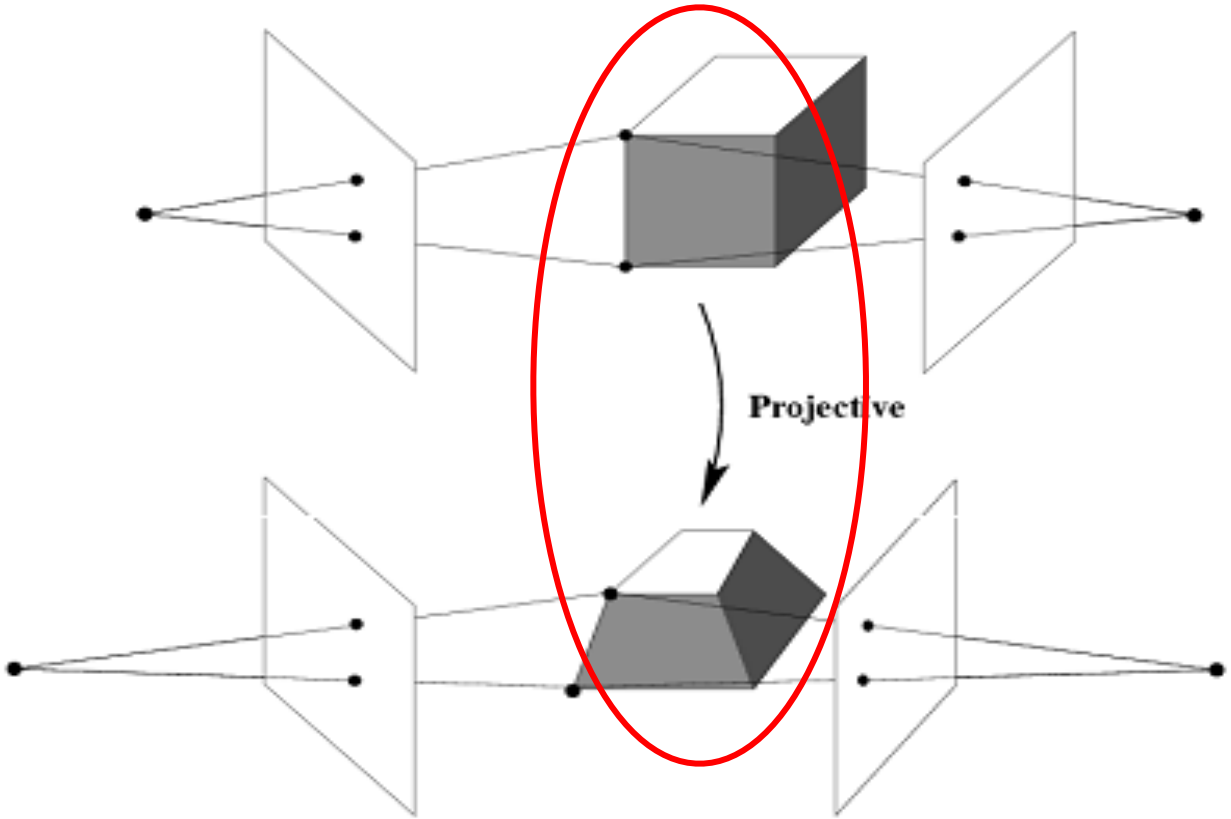
Reconstruction Ambiguity: Affine



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

Slide credit: Svetlana Lazebnik

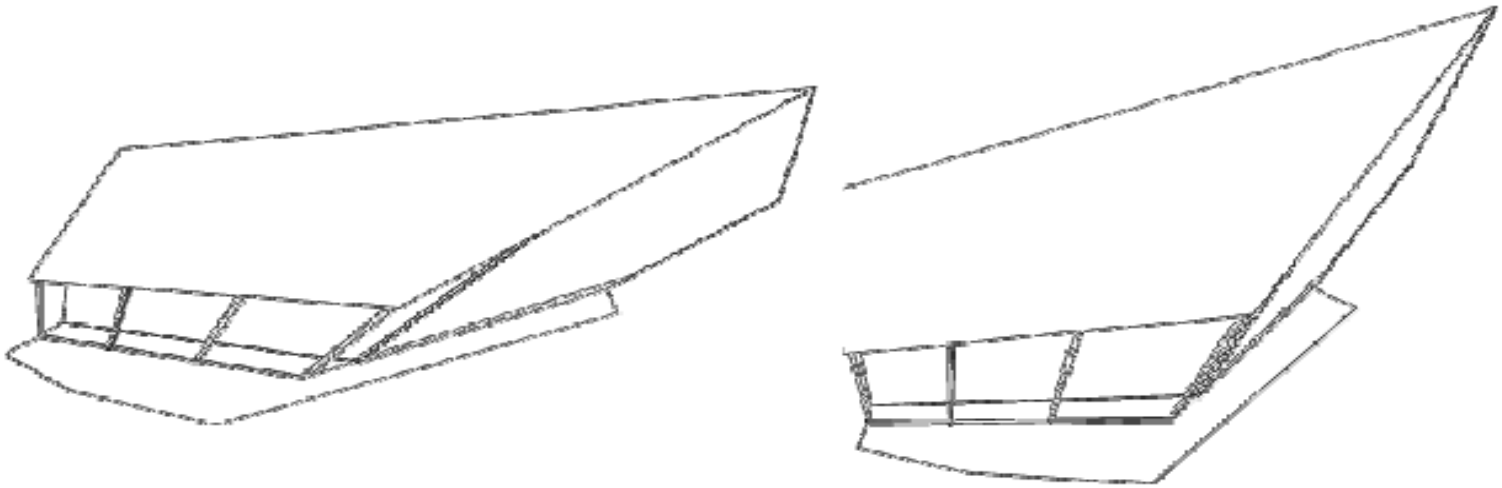
Reconstruction Ambiguity: Projective



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

Slide credit: Svetlana Lazebnik

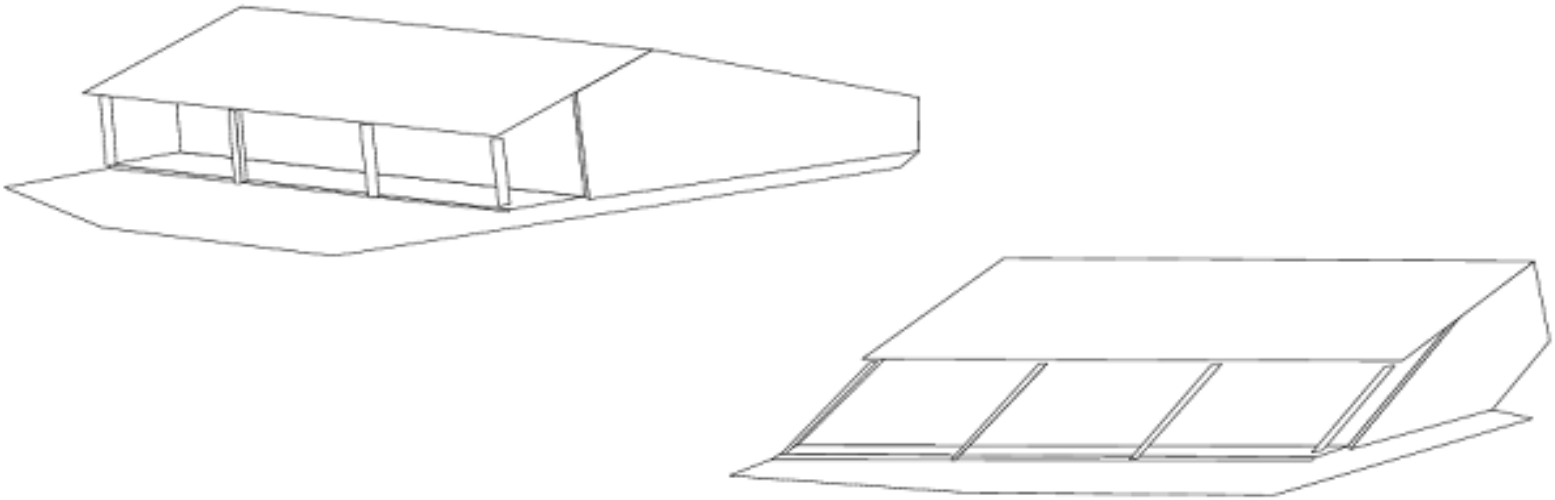
Projective Ambiguity



2 Views of Projective reconstruction

Slide credit: Svetlana Lazebnik

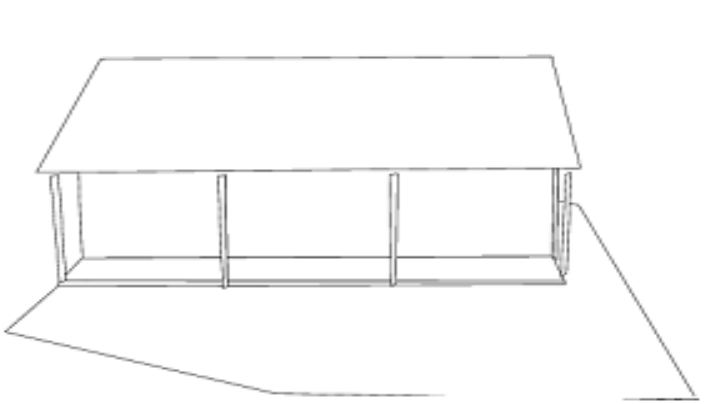
From Projective to Affine

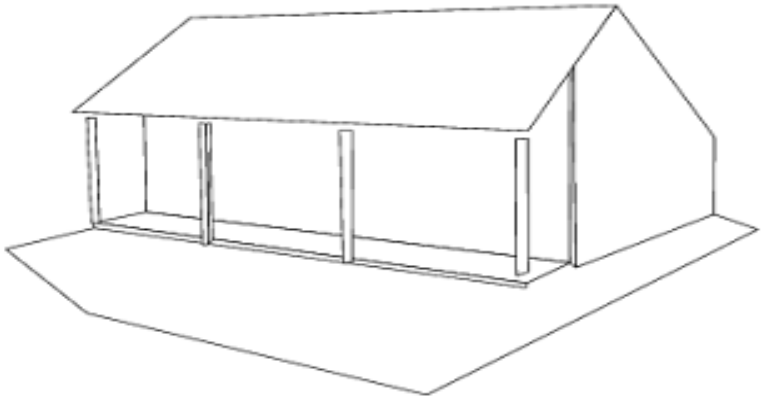


2 Views of Affine reconstruction

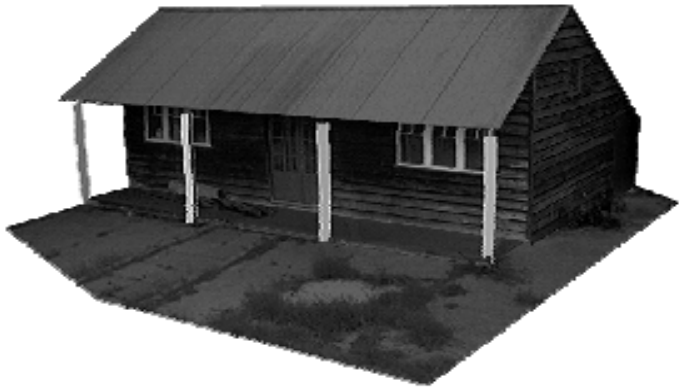
Slide credit: Svetlana Lazebnik

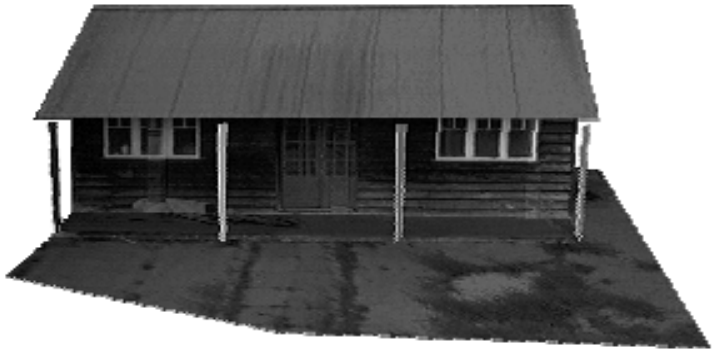
From Affine to Similarity





2 Views of Similarity reconstruction

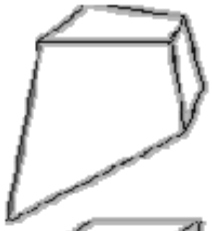




2 Views of a texture mapped

Slide credit: Svetlana Lazebnik

Hierarchy of 3D Transformations

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

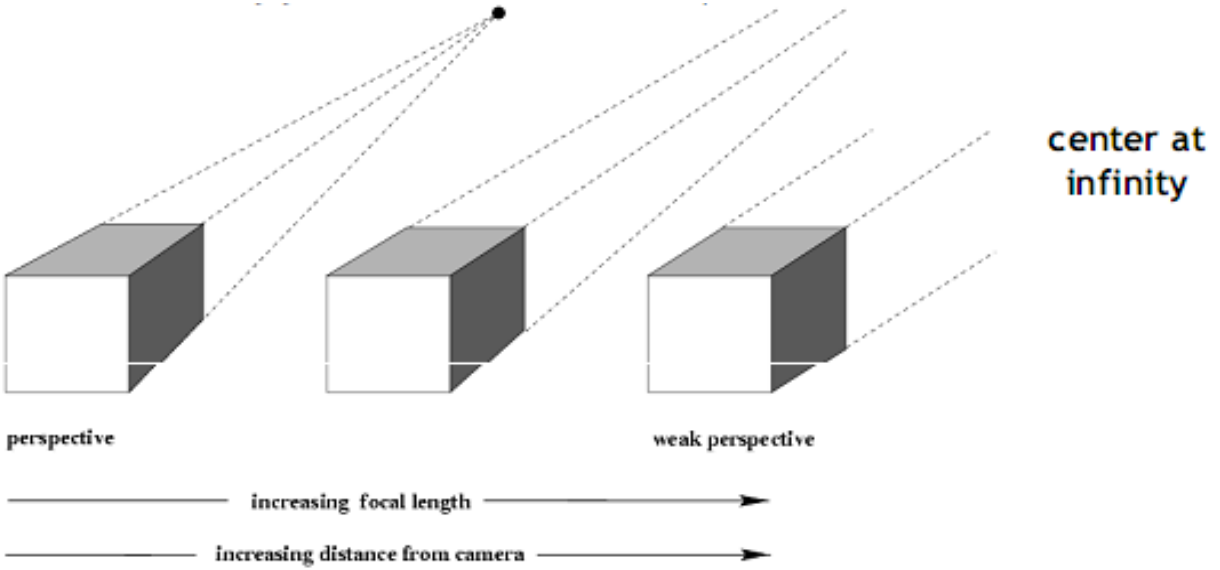
Slide credit: Svetlana Lazebnik

Affine SfM

- Affine cameras
- Affine factorization
- Euclidean reconstruction
- Dealing with **missing data**

Structure from Motion

- Affine cameras

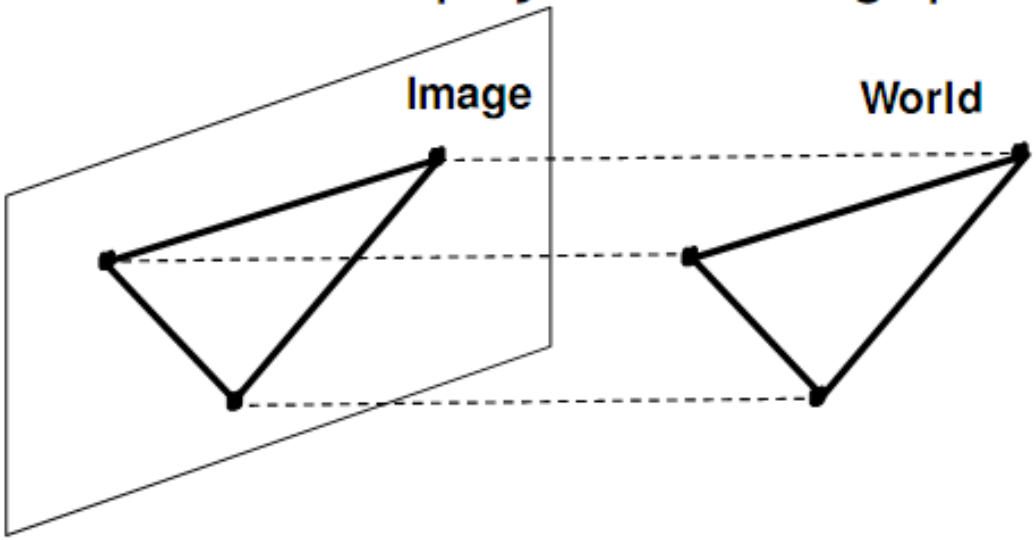


Increasing focal length

Slide credit: Svetlana Lazebnik

Orthographic Projection

- Special case of perspective projection
  - Distance from center of projection to image plane is infinite



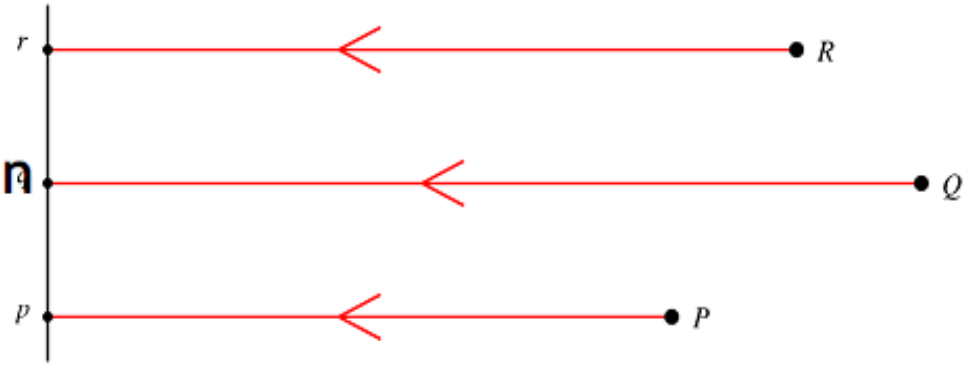
  - Projection matrix:

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

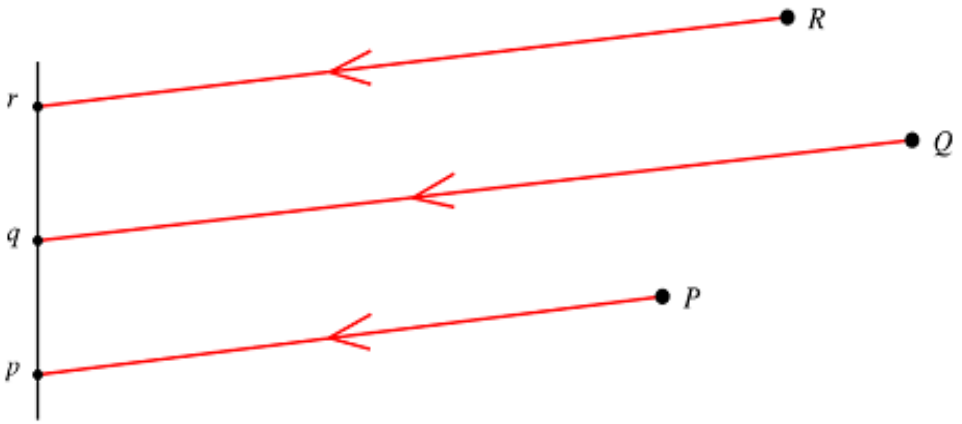
Slide credit: Steve Seitz

Affine Cameras

Orthographic Projection



Parallel Projection



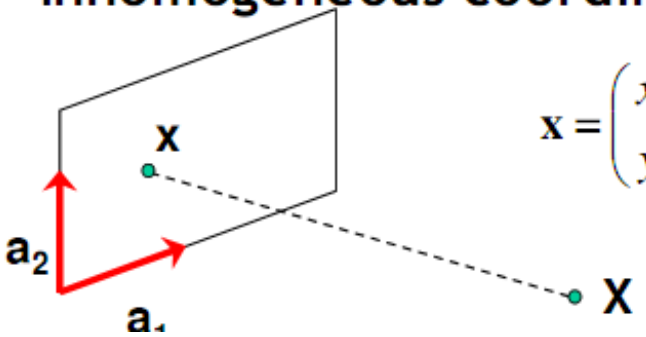
Slide credit: Svetlana Lazebnik

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:

$$\mathbf{P} = [3\times3\text{affine}]\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\end{bmatrix}[4\times4\text{affine}] = \begin{bmatrix}a_{11}&a_{12}&a_{13}&b_1\\a_{21}&a_{22}&a_{23}&b_2\\0&0&0&1\end{bmatrix} = \begin{bmatrix}\mathbf{A}&\mathbf{b}\\\mathbf{0}&1\end{bmatrix}$$

- 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

Slide credit: Svetlana Lazebnik

Affine Structure from Motion

- Given: *m* images of *n* fixed 3D points:
  - $\mathbf{x}_{ij} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i, \quad i = 1, \ldots, m, \; j = 1, \ldots, n$
- 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):

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

- 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.

m=2, n=4

Slide credit: Svetlana Lazebnik

Affine Structure from Motion

- Centering: subtract the centroid of the image points

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

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

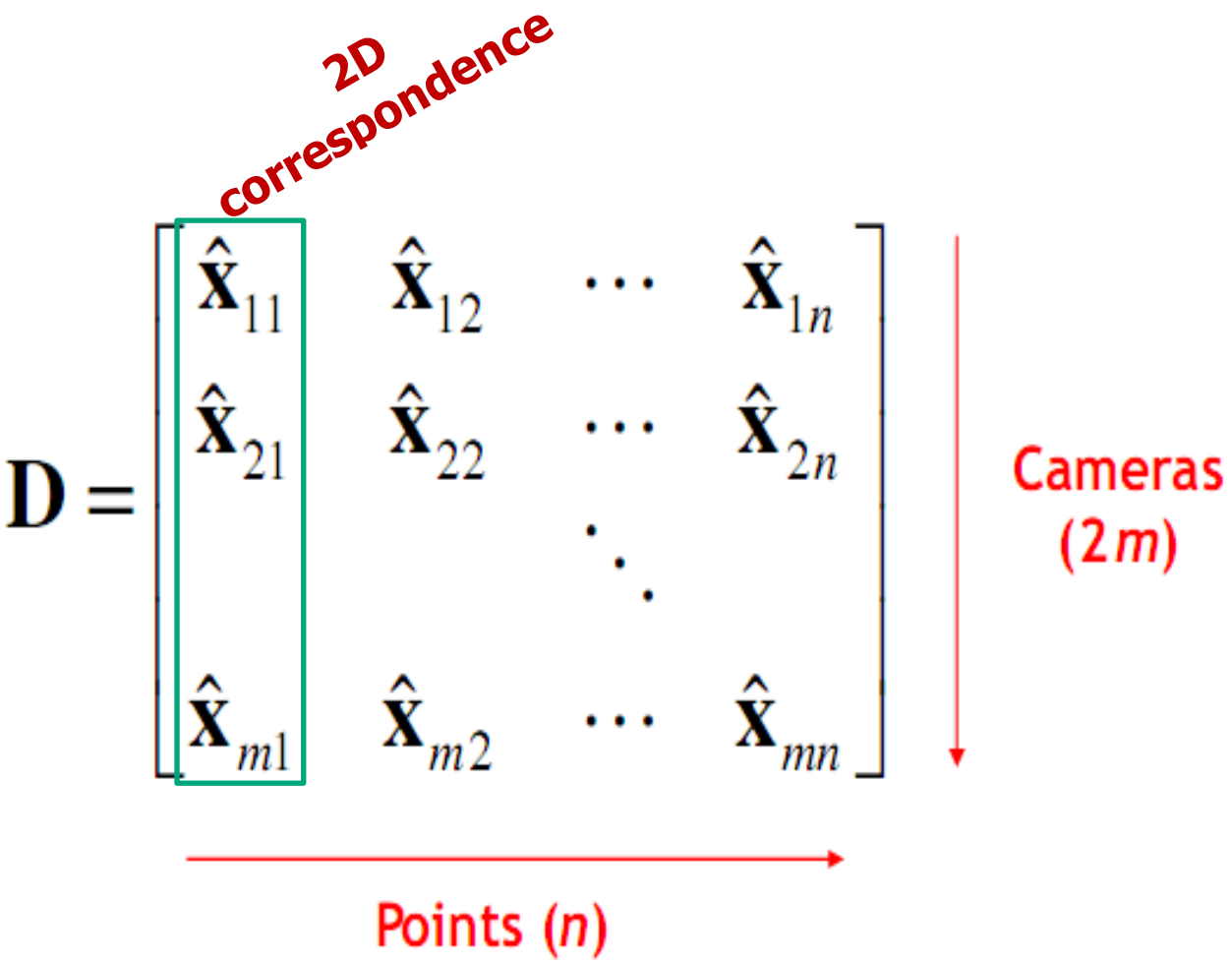
- For simplicity, assume that the origin of the world coordinate system is at the centroid of the 3D points.
- After centering, each normalized point x_{ij} is related to the 3D point X_i by

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

Slide credit: Svetlana Lazebnik

Affine Structure from Motion

- **Create a 2m × n data (measurement) matrix:**



Slide credit: Svetlana Lazebnik

Affine Structure from Motion

- **Create a 2m × n data (measurement) matrix:**

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

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

Points ($3 \times n$)

Cameras ($2m \times 3$)

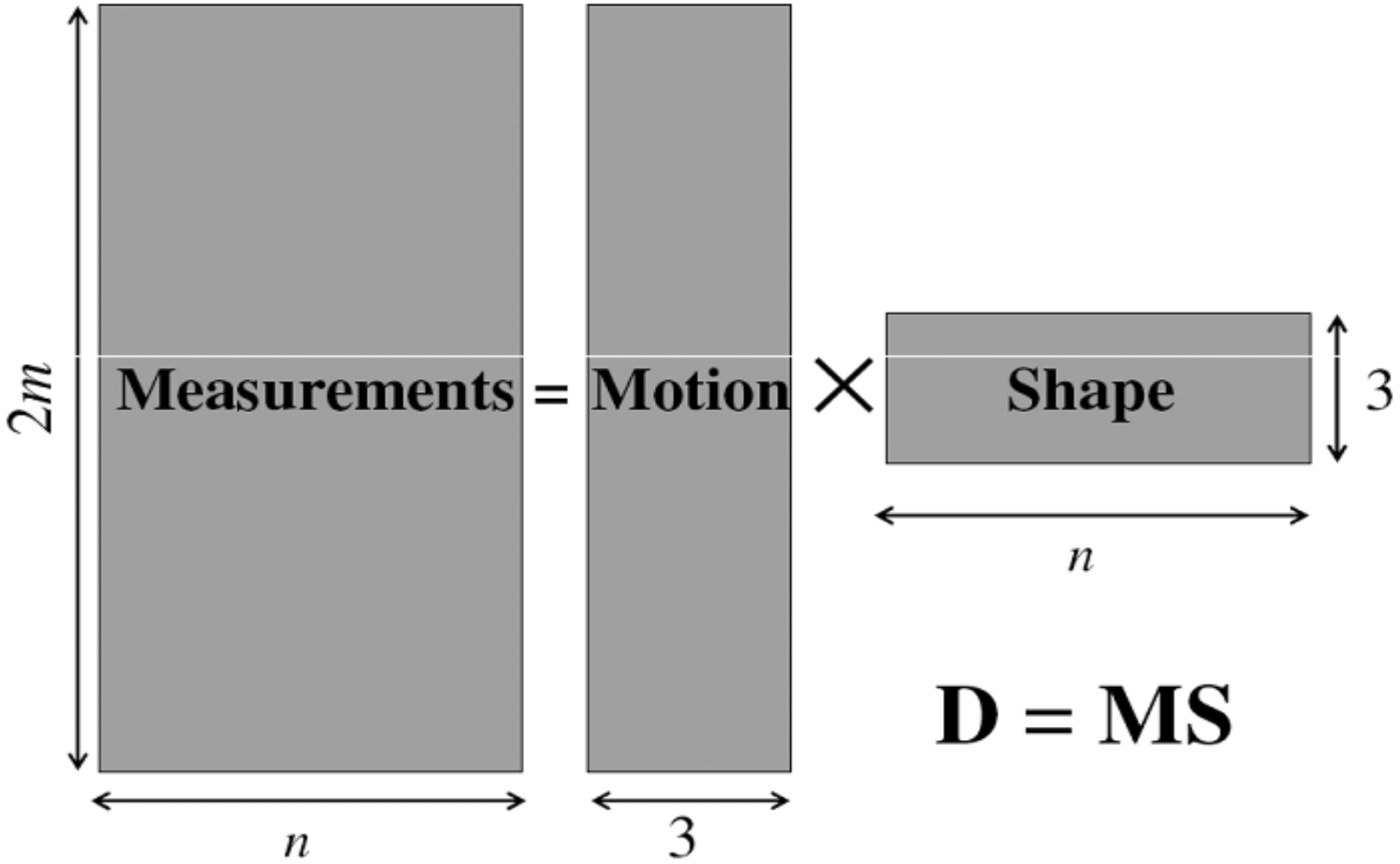
$$\mathbf{A}_i = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

（linear mapping）

The measurement matrix D = MS must have rank 3.

Slide credit: Svetlana Lazebnik

Factorizing the Measurement Matrix

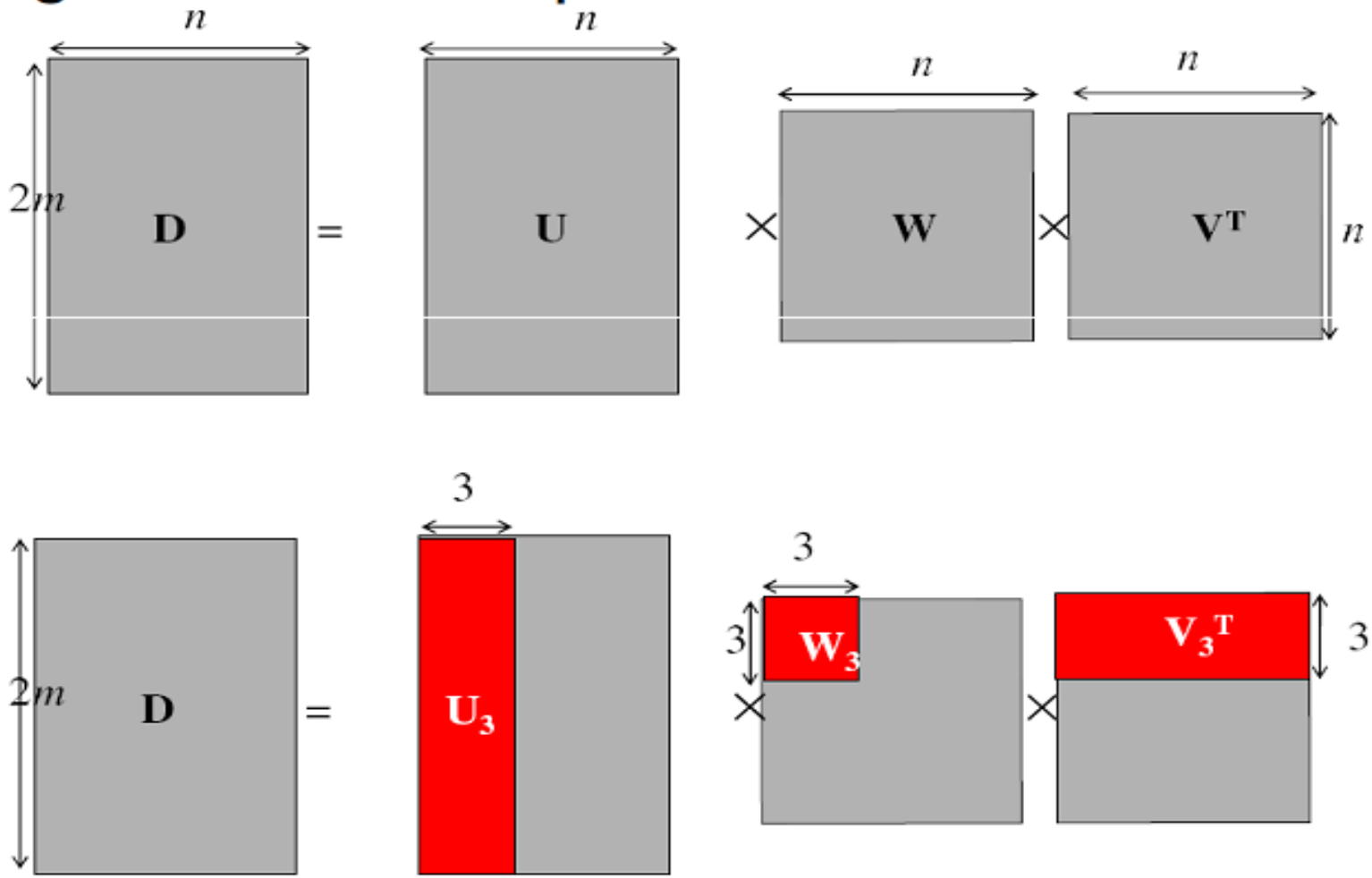


C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Slide credit: Martial Hebert

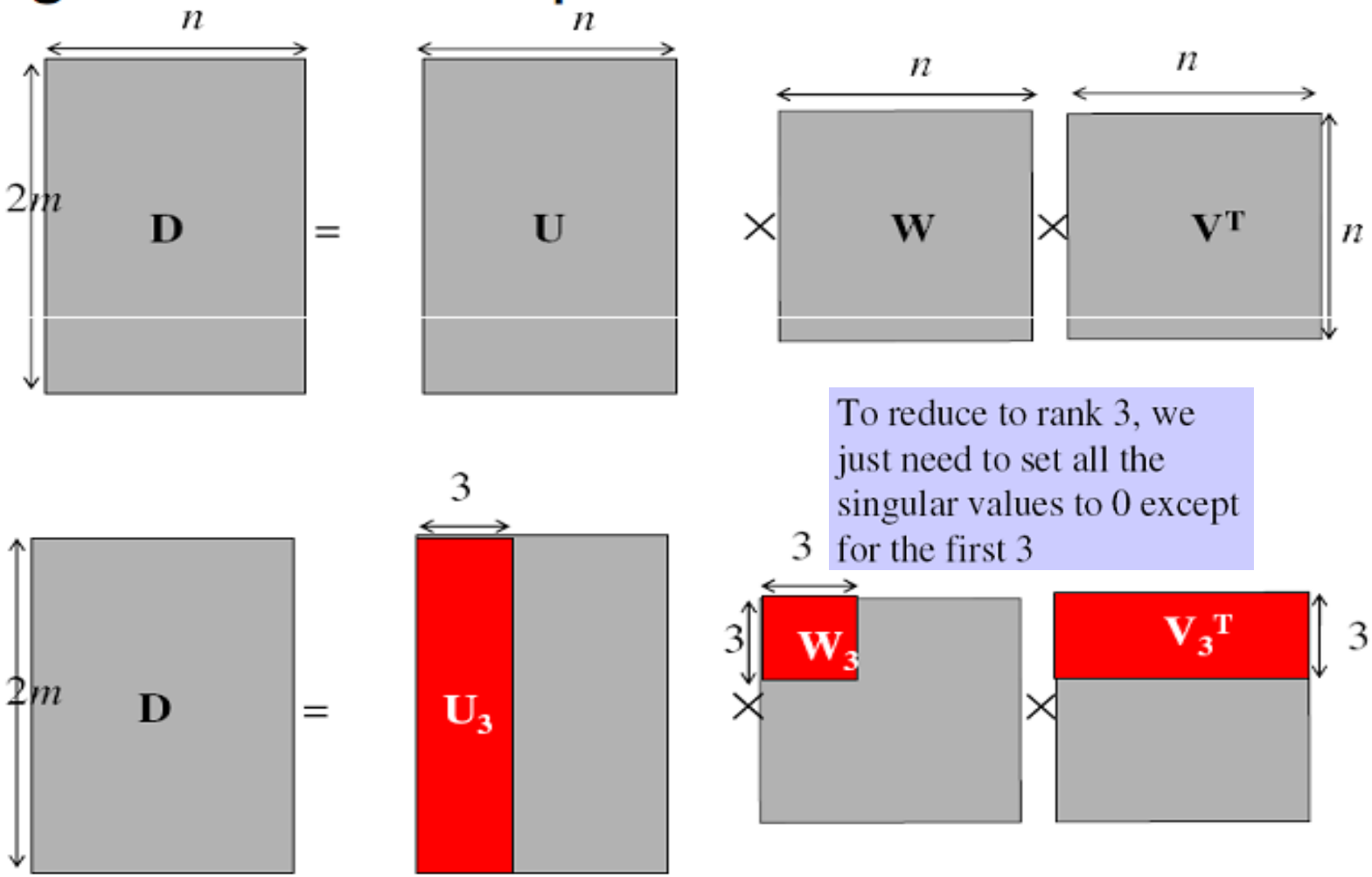
Factorizing the Measurement Matrix

- Singular value decomposition of D:



Factorizing the Measurement Matrix

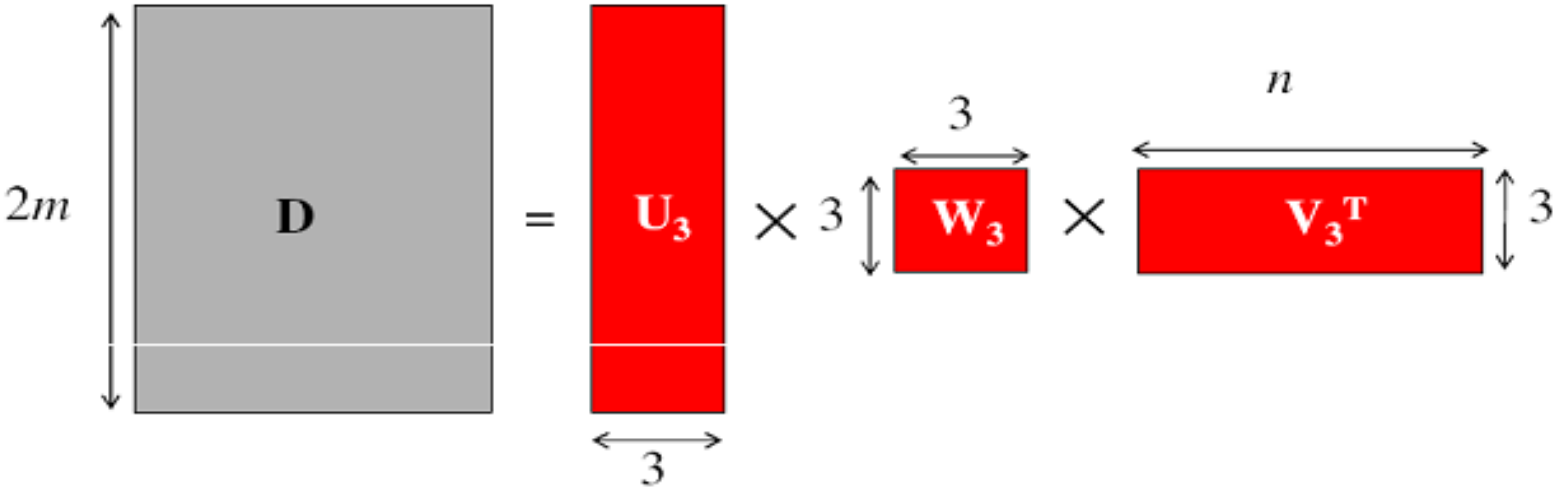
- **Singular value decomposition of D:**



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

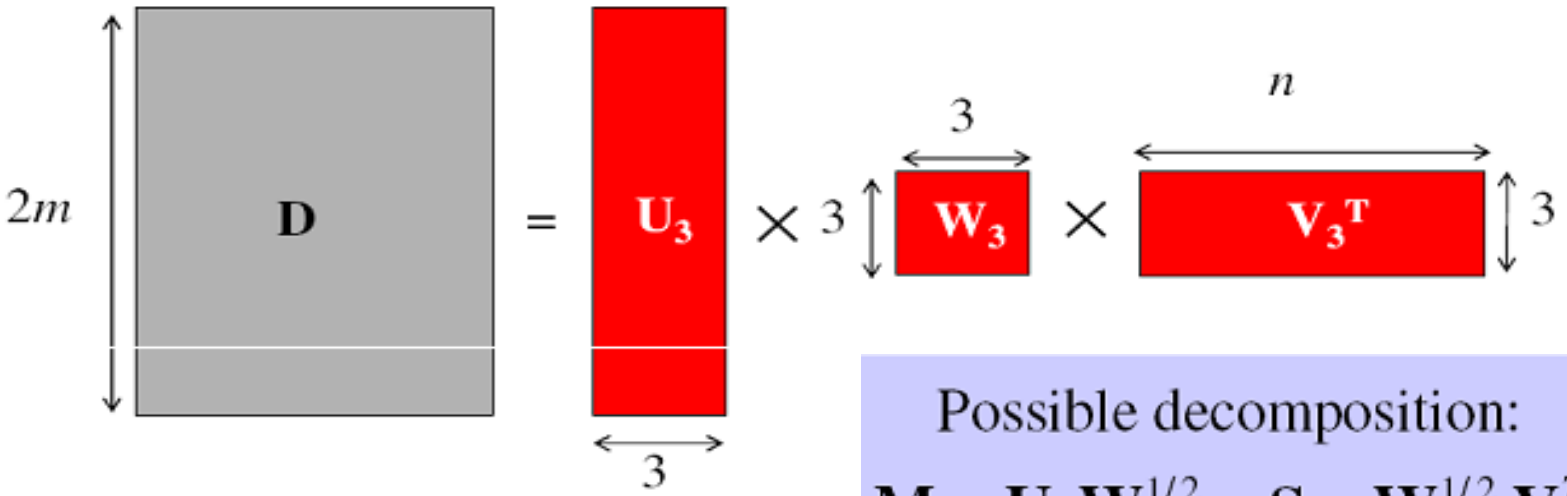
Factorizing the Measurement Matrix

- **Obtaining a factorization from SVD:**



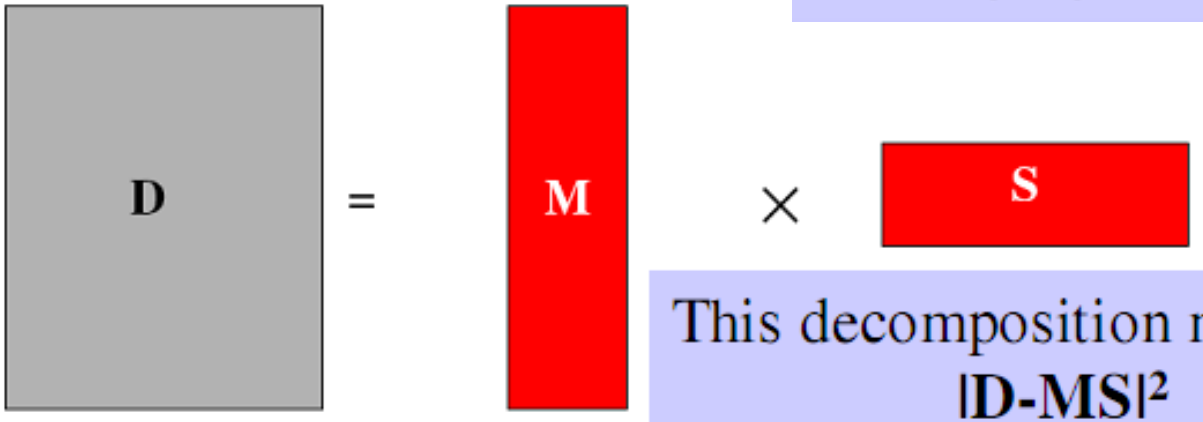
Factorizing the Measurement Matrix

- **Obtaining a factorization from SVD:**



Possible decomposition:

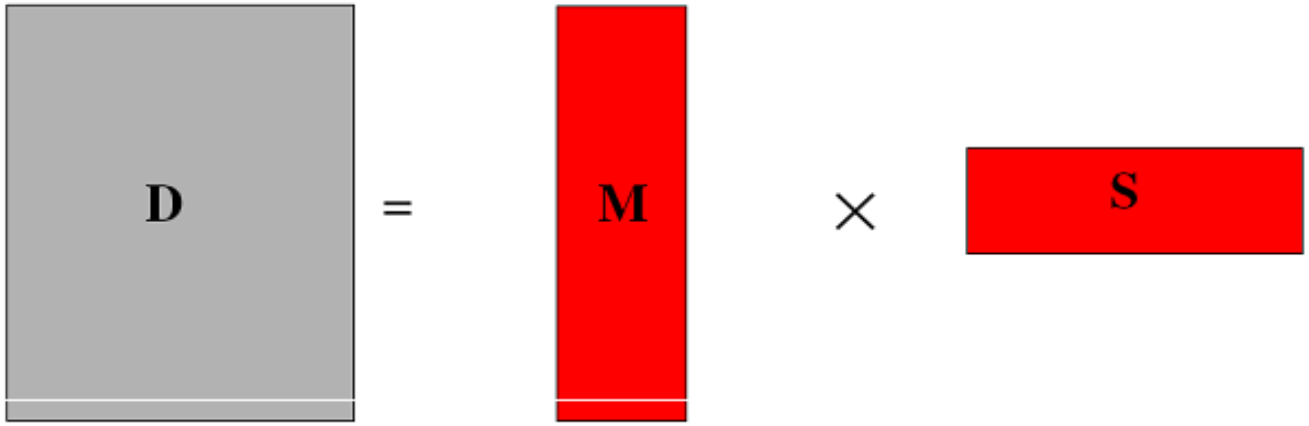
$$\mathbf{M} = \mathbf{U}_3 \mathbf{W}_3^{1/2} \qquad \mathbf{S} = \mathbf{W}_3^{1/2} \mathbf{V}_3^T$$



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

Slide credit: Martial Hebert

Affine Ambiguity



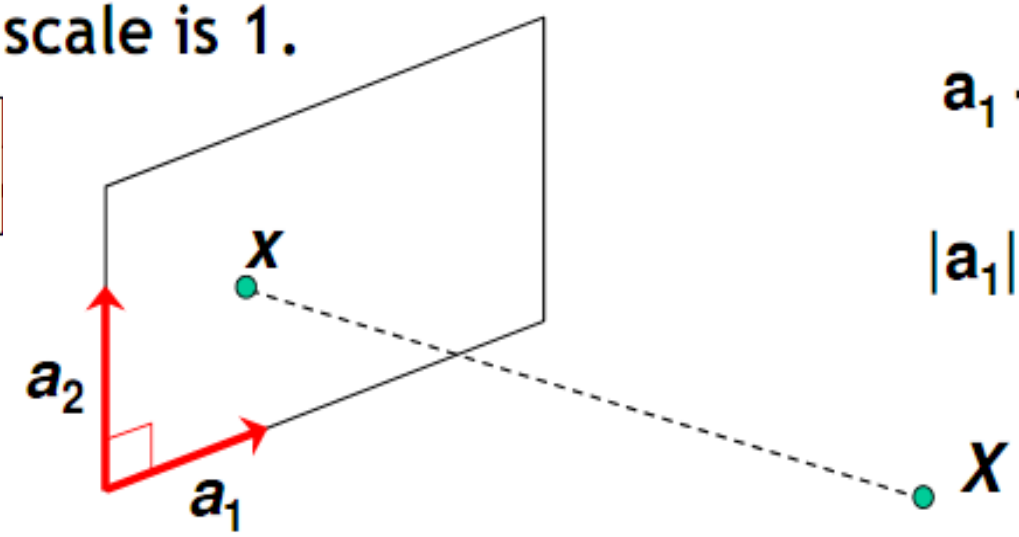
- The decomposition is not unique. We get the same D by using any 3×3 matrix C and applying the transformations $M \rightarrow MC$, $S \rightarrow C^{-1}S$.
- That is because we have only an affine transformation and we have not enforced any Euclidean constraints (like forcing the image axes to be perpendicular,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

Slide credit: Martial Hebert

Estimating the Euclidean Structure

- Orthographic assumption: image axes are perpendicular and scale is 1.

$$\mathbf{A}_i = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$



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

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

- This can be converted into a system of $3m$ equations:

$$\begin{cases} \hat{a}_{i1} \cdot \hat{a}_{i2} = 0 \\ |\hat{a}_{i1}| = 1 \\ |\hat{a}_{i2}| = 1 \end{cases} \Leftrightarrow \begin{cases} a_{i1}^T C C^T a_{i2} = 0 \\ a_{i1}^T C C^T a_{i1} = 1, \\ a_{i2}^T C C^T a_{i2} = 1 \end{cases} \quad i = 1, \ldots, m$$

Slide credit: Martial Hebert

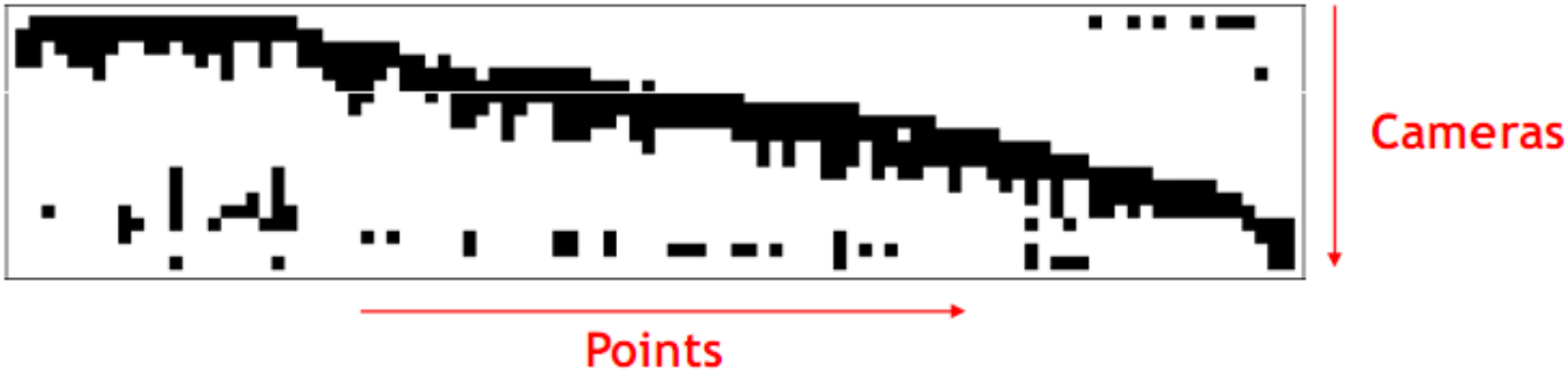
Algorithm Summary

- Given: *m* images and *n* features x_{ij}
- For each image *i*, center the feature coordinates.
- Construct a $2m \times n$ 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: $D = U W V^T$
  - Create U_3 by taking the first 3 columns of U
  - Create V_3 by taking the first 3 columns of V
  - Create W_3 by taking the upper left 3×3 block of W
- Create the motion and shape matrices:
  - $M = U_3 W_3^{1/2}$ and $S = W_3^{1/2} V_3^T$ (or $M = U_3$ and $S = W_3 V_3^T$)
- Eliminate affine ambiguity (**Estimating the Euclidean Structure**)

Slide credit: Martial Hebert

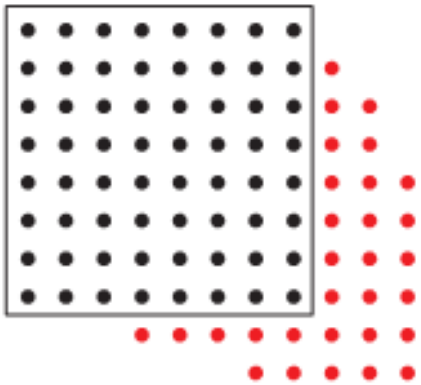
Dealing with Missing Data

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



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



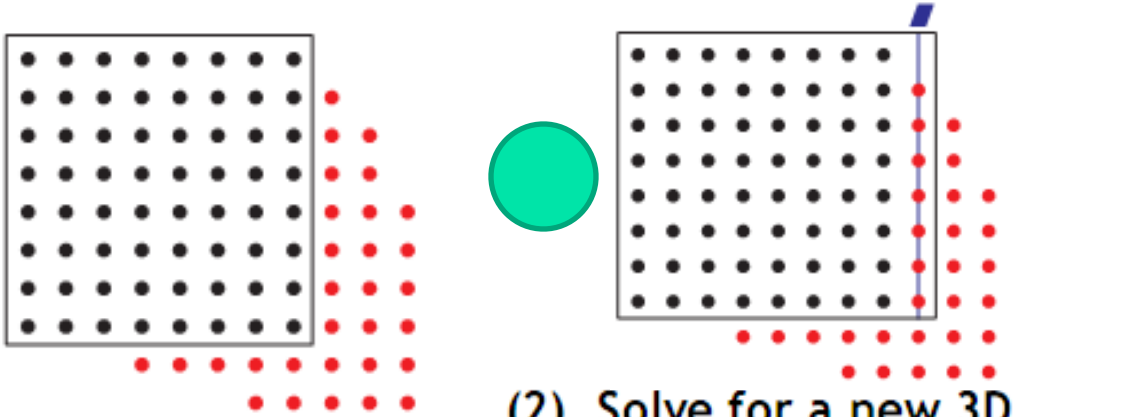
(1) Perform factorization on a dense sub-block

F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. Segmenting, Modeling, and Matching Video Clips Containing Multiple Moving Objects. PAMI 2007. **CVPR04**

Slide credit: Svetlana 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

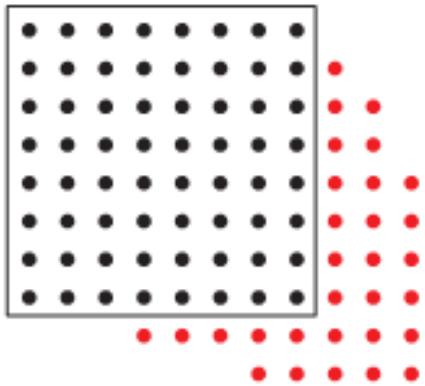
(2) Solve for a new 3D point visible by at least two known cameras (linear least squares)

F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. Segmenting, Modeling, and Matching Video Clips Containing Multiple Moving Objects. PAMI 2007. **CVPR04**

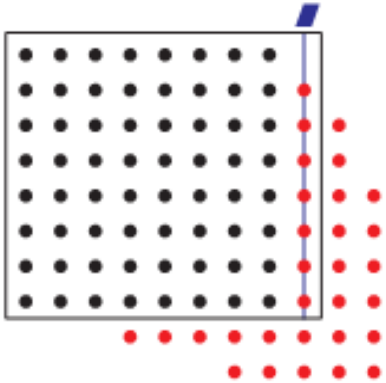
Slide credit: Svetlana Lazebnik

Dealing with Missing Data

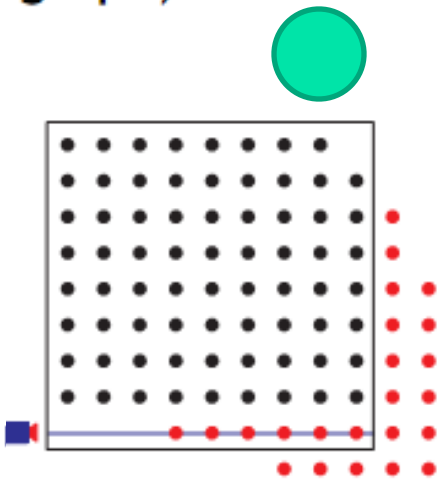
- Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results
  - Finding dense maximal sub-blocks of the matrix is NP-complete (equivalent to finding maximal cliques in a graph)
- Incremental bilinear refinement



(1) Perform factorization on a dense sub-block



(2) Solve for a new 3D point visible by at lea 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. **CVPR04**

Slide credit: Svetlana Lazebnik

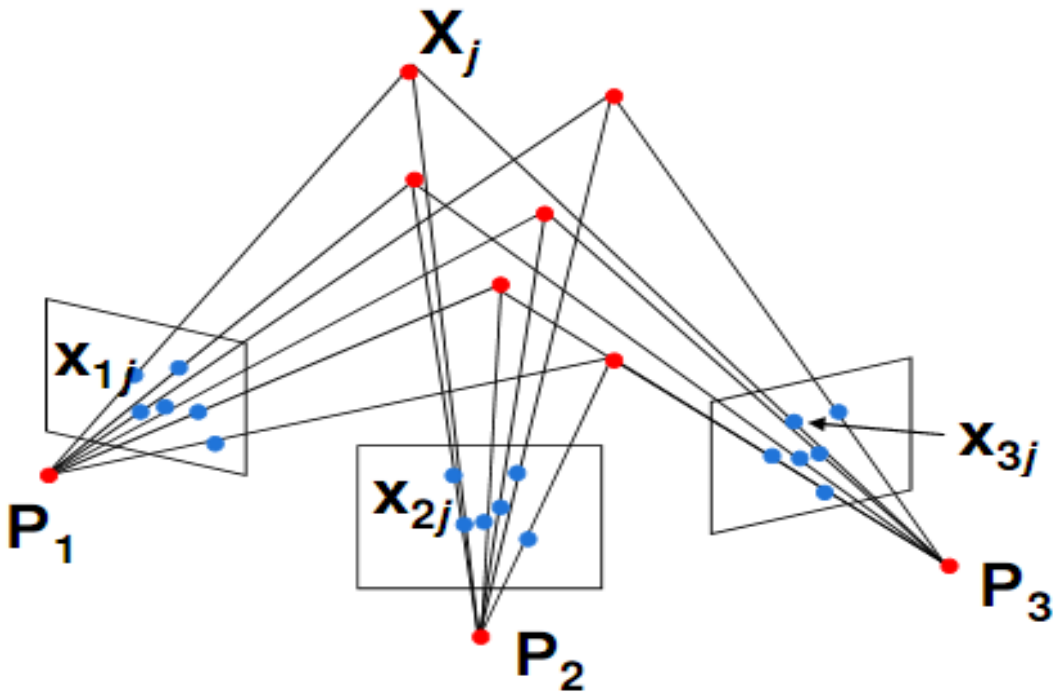
Comments: Affine SfM

- Affine SfM was historically developed first.
- It is valid under the assumption of *affine cameras*.
  - Which does not hold for real physical cameras…
  - …but which is still tolerable if the scene points are far away from the camera.
- For good results with real cameras, we typically need projective SfM.
  - Harder problem, more ambiguity
  - Math is a bit more involved…
    (Here, only basic ideas. If you want to implement it, please look at the H&Z book for details).

Source: B. Leibe

Projective Structure from Motion

Problem:



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

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

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

Slide credit: Svetlana Lazebnik

Projective Structure from Motion

- Given: *m* images of *n* fixed 3D points
  - $z_{ij}\,\mathbf{x}_{ij} = \mathbf{P}_i\,\mathbf{X}_j, \qquad i = 1,\ldots,m, \quad j = 1,\ldots,n$
- Problem: estimate *m* projection matrices P_i and *n* 3D points X_j from the *mn* correspondences x_{ij}
- With no calibration info, cameras and points can only be recovered up to a 4x4 projective transformation Q:

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

- We can solve for structure and motion when

$$2mn >= 11m + 3n - 15$$

- For two cameras, at least 7 points are needed.

Slide credit: Svetlana Lazebnik

Projective Factorization

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

Points ($4 \times n$)

Cameras ($3m \times 4$)

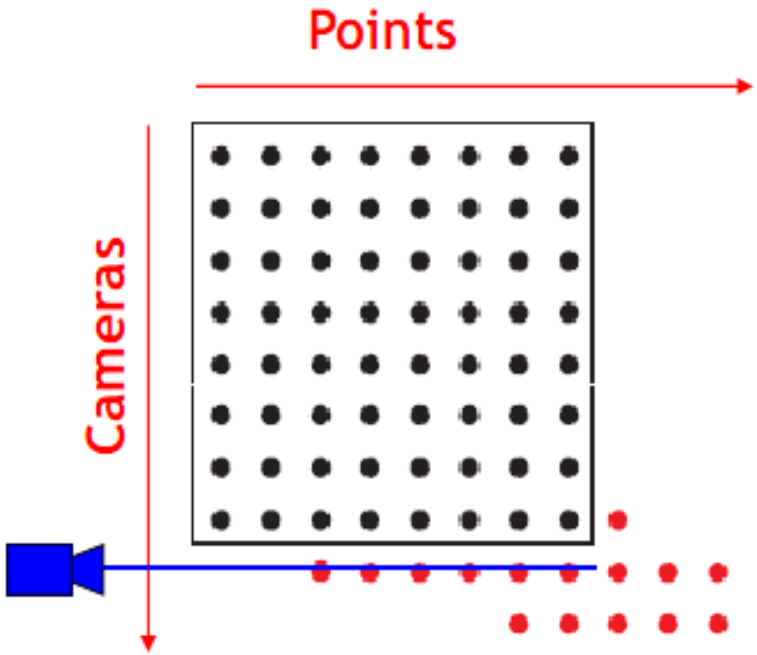
$\mathbf{D} = \mathbf{MS}$ has rank 4

- 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).

Slide credit: Svetlana Lazebnik

Sequential Structure from Motion

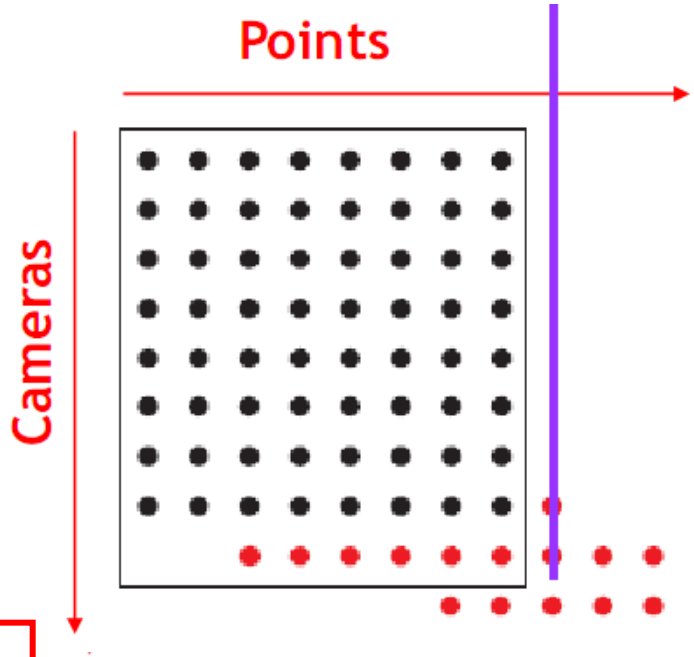
- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image - *calibration* 3D → 2D , **P**



Slide credit: Svetlana Lazebnik

Sequential Structure from Motion

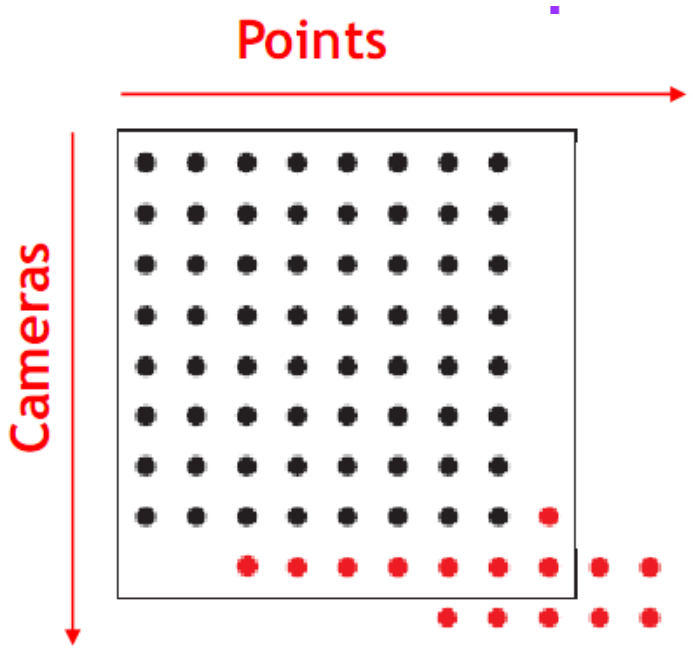
- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image - *calibration* 3D → 2D , **P**
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera - *triangulation* 2D → 3D



Slide credit: Svetlana 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*

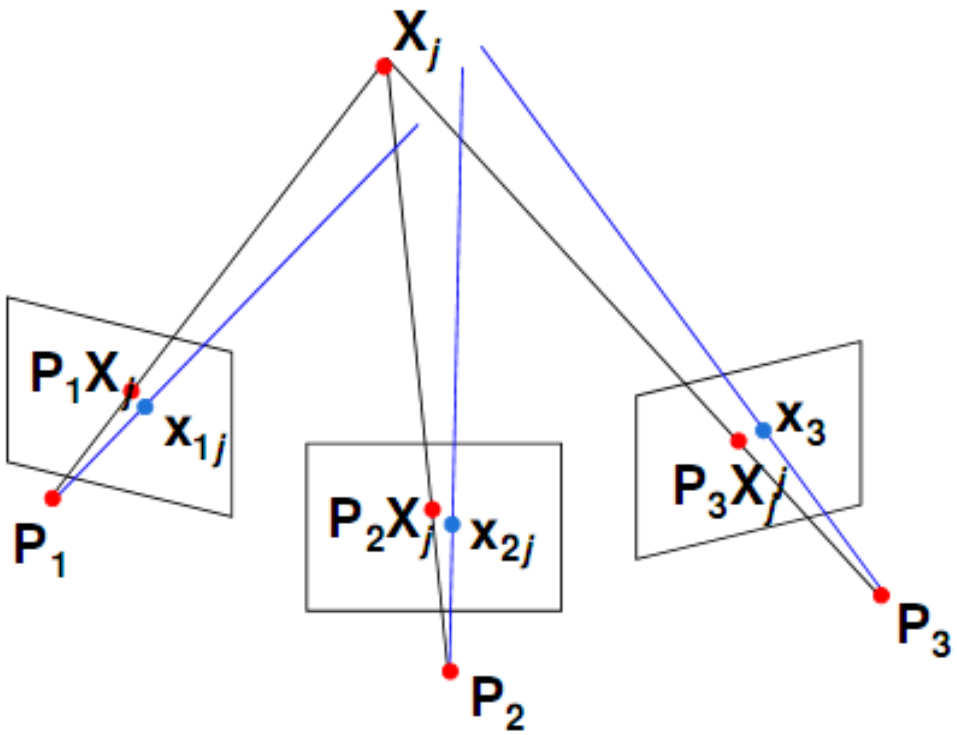


Slide credit: Svetlana Lazebnik

Bundle Adjustment

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

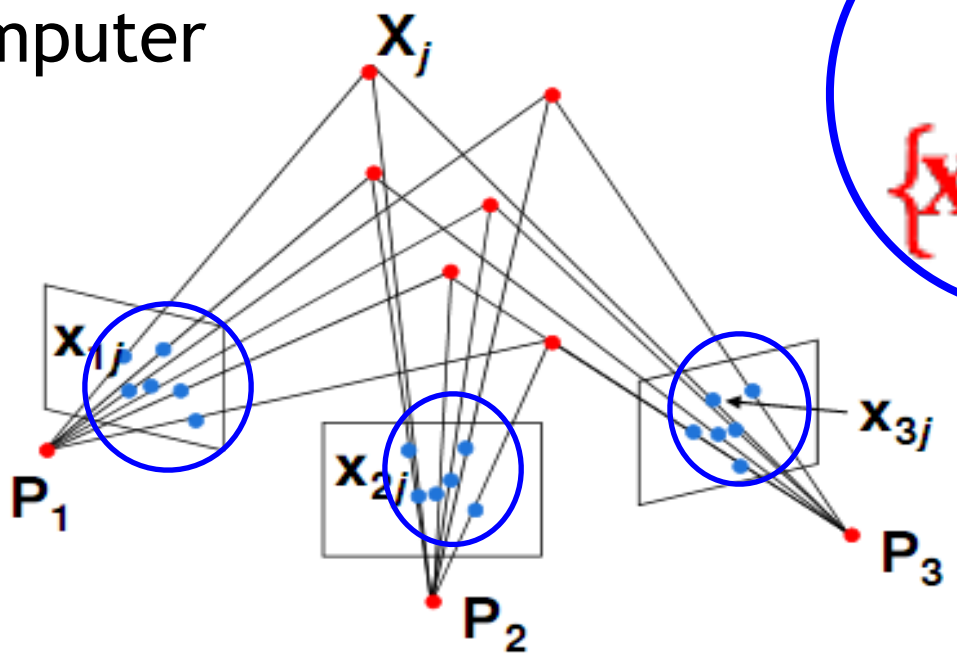
$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$

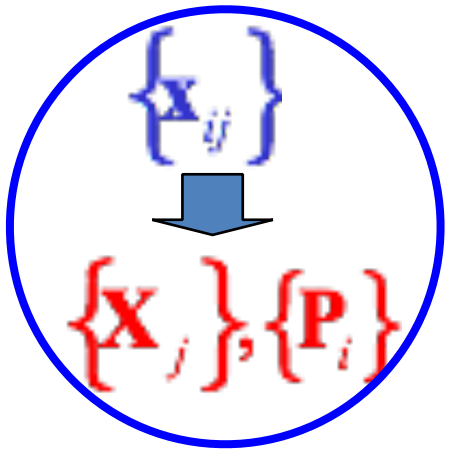


Slide credit: Svetlana Lazebnik

Bundle Adjustment (BA) &NLS

◆ Recall: the **SFM** in geometric computer vision





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

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

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

Bundle Adjustment (BA) &NLS

- SFM: minimizing mean-square **re-projection** error

✓Actual observed: $\mathbf{x}_{ij}$

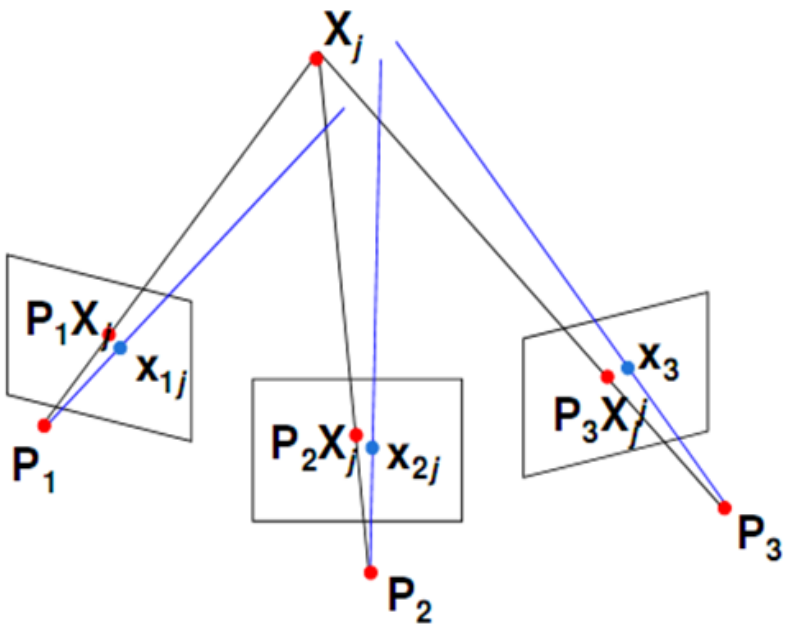
$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$

✓Measurement Model (predicte

$$\bar{\mathbf{x}}_{ij} = h\left(\mathbf{P}_i, \mathbf{X}_j\right)$$

$$= \mathbf{P}_i\mathbf{X}_j$$

$$= K[R_i \mid t_i]\begin{bmatrix}\mathbf{X}_j \\ 1\end{bmatrix}$$



- **Goal:**

  Minimize the error between observed and predicted.

Bundle Adjustment (BA) &NLS

- **NLS**: Measurements are considered to be normally distributed (independently), BA is the optimal **N**onlinear **L**east-**S**quares (NLS) algorithm.
- General **NLS** formulation
  - Observation model (the *h* is a nonlinear function w.r.t. **x**)

$$Z = h(\mathbf{x}) + \mathbf{v}; \quad \mathbf{v} \sim N(0, R)$$

  - Actual measurement Z
  - Error or Cost function

$$E(\mathbf{x}) = \left\| Z - h(\mathbf{x}) \right\|_{R^{-1}} = \left(Z - h(\mathbf{x}) \right)^T R^{-1} \left(Z - h(\mathbf{x}) \right)$$

$$= \sum_{k,i} \left(Z_{k,i} - h(\mathbf{x}_{k,i}) \right)^T R_{k,i}^{-1} \left(Z_{k,i} - h(\mathbf{x}_{k,i}) \right)$$

Bundle Adjustment (BA) &NLS

- Solutions to minimizing the cost function
  - ✓ **S**teepest **D**escent **M**ethod (**SDM**), 1st-order approximation about $\mathbf{x} = \bar{\mathbf{x}}$, linear convergence rate (slower)

$$\Delta\mathbf{x} = -\alpha \frac{\partial E}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \alpha J(\bar{\mathbf{x}})^T R^{-1} e(\bar{\mathbf{x}}); \quad \Delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$$

$$JacobianMatrix: J(\bar{\mathbf{x}}) = \frac{\partial h}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\bar{\mathbf{x}}}$$

$$Errorvector: e(\bar{\mathbf{x}}) = Z - h(\bar{\mathbf{x}})$$

  - ✓ **N**ewton **M**ethod (2nd-order approximation), quadric convergence rate (faster)

$$\Delta\mathbf{x} = \left(\frac{\partial^2 E}{\partial \mathbf{x}^2}\Big|_{\mathbf{x}=\bar{\mathbf{x}}}\right)^{-1} \frac{\partial E}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\bar{\mathbf{x}}}$$

✓ Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. Robotics and Automation, ICRA '04.

Bundle Adjustment (BA) &NLS

- Approximation Newton method
  - **Gauss Newton Method (GNM)**, Hessian matrix is approximated using Jacobian, (Jacobian pseudo-inverse)

$$\Delta \mathbf{x} = \left(J_{\bar{\mathbf{x}}}^T R^{-1} J_{\bar{\mathbf{x}}} \right)^{-1} J_{\bar{\mathbf{x}}}^T R^{-1} e\left(\bar{\mathbf{x}}\right)$$

  - **LM** (Levenberg - Marquardt) **Method** (Damped least Squares), allows to smoothly pass from **SDM** to **GNM**.

$$\Delta \mathbf{x} = \left(J_{\bar{\mathbf{x}}}^T R^{-1} J_{\bar{\mathbf{x}}} + \lambda D \right)^{-1} J_{\bar{\mathbf{x}}}^T R^{-1} e\left(\bar{\mathbf{x}}\right)$$

$$D = \begin{cases} I & ,\mathbf{L}evenberg \\ diag(J^T R^{-1} J), & \mathbf{M}arquardt \end{cases}$$

SDM: Far from minimum

GNM: Closer to minimum

Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. Robotics and Automation, ICRA '04.

Bundle Adjustment (BA) &NLS

- **Given matrix equaiton**

$$A\mathbf{x} = b$$

- **Normal equation:** which minimizes the sum of the square differences between the left and right sides,

$$A^T A\mathbf{x} = A^T b$$

$A^T A$: Normal matrix or symmetry positive define matrix

- Normal equations from **GNM** or **LMM**

$$\left(J_{\bar{\mathbf{x}}}^T R^{-1} J_{\bar{\mathbf{x}}}\right)\Delta\mathbf{x} = J_{\bar{\mathbf{x}}}^T R^{-1} e\left(\bar{\mathbf{x}}\right)$$

$$\left(J_{\bar{\mathbf{x}}}^T R^{-1} J_{\bar{\mathbf{x}}} + \lambda D\right)\Delta\mathbf{x} = J_{\bar{\mathbf{x}}}^T R^{-1} e\left(\bar{\mathbf{x}}\right)$$

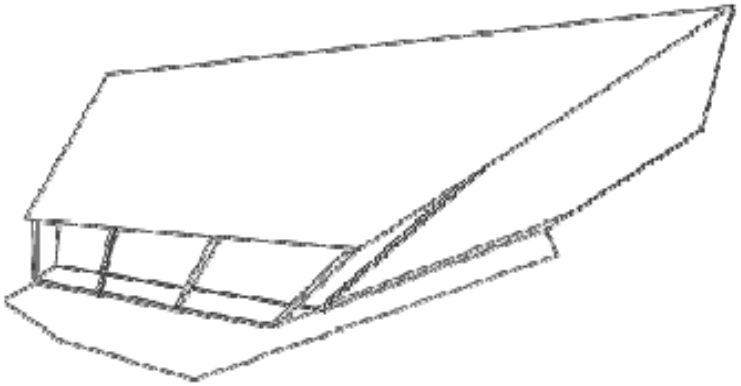
Bundle Adjustment

- Seeks the Maximum Likelihood (ML) solution assuming the measurement noise is Gaussian.
- It involves adjusting the bundle of rays between each camera center and the set of 3D points.
- Bundle adjustment should generally be used as the final step of any multi-view reconstruction algorithm.
  - Considerably improves the results.
  - Allows assignment of individual covariances to each measurement.
- However…
  - It needs a good initialization.
  - It can become an extremely large minimization problem.
- Very efficient algorithms available.

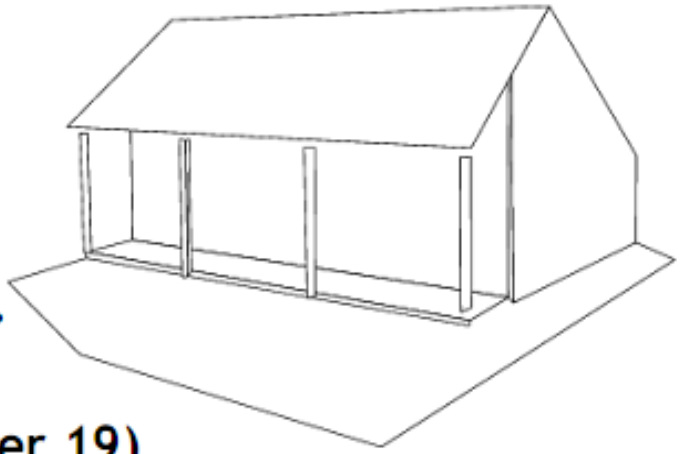
Source: B. Leibe

Projective Ambiguity

- If we don't know anything about the camera or the scene, the best we can get with this is a reconstruction up to a projective ambiguity Q.
  - This can already be useful.
  - E.g. we can answer questions like "at what point does a line intersect a plane"?



- If we want to convert this to a "true" reconstruction, we need a *Euclidean upgrade*.
  - Need to put in additional knowledge about the camera (calibration) or about the scene (e.g. from markers).
  - Several methods available (see F&P Chapter 13.5 or H&Z Chapter 19)

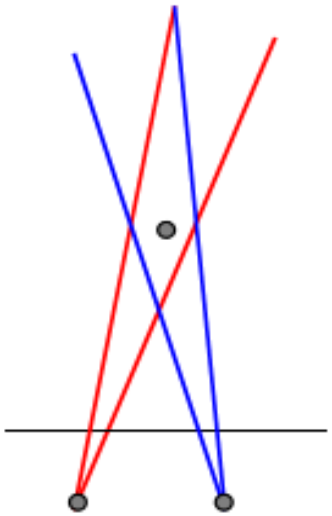


Self-Calibration

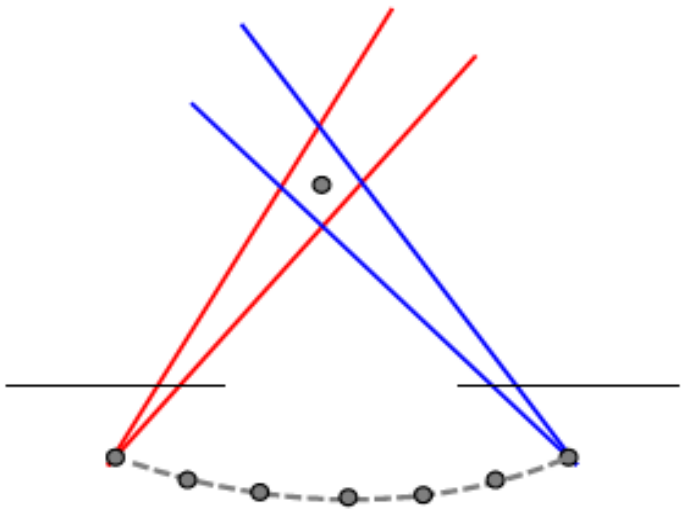
- Self-calibration (auto-calibration) is the process of determining intrinsic camera parameters directly from uncalibrated images.
- For example, when the images are acquired by a single moving camera, we can use the constraint that the intrinsic parameter matrix remains fixed for all the images.
  - Compute initial projective reconstruction and find 3D projective transformation matrix Q such that all camera matrices are in the form $P_i = K [R_i \mid t_i]$.
- Can use constraints on the form of the calibration matrix: square pixels, zero skew, fixed focal length, etc.

Slide credit: Svetlana Lazebnik

Practical Considerations (1)



Small Baseline



Large Baseline

1. Role of the baseline
   - Small baseline: large depth error
   - Large baseline: difficult search problem

- Solution
  - Track features between frames until baseline is sufficient.

Source: Steve Seitz

Practical Considerations (2)

2. There will still be many outliers

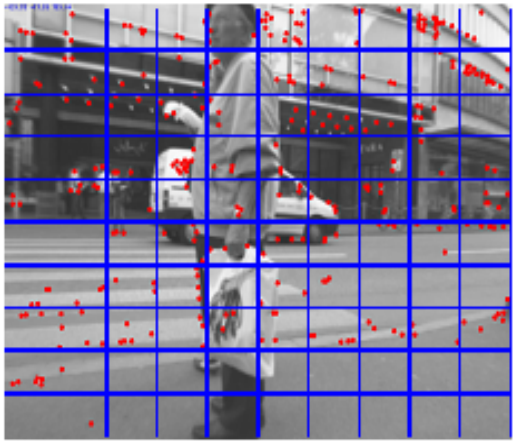
- Incorrect feature matches
- Moving objects

⇒ Apply RANSAC to get robust estimates based on the inlier points.

3. Estimation quality depends on the point configuration

- Points that are close together in the image produce less stable solutions.

⇒ Subdivide image into a grid and try to extract about the same number of features per grid cell.



Bucketing technique

Source: B. Leibe

General Guidelines

- Use calibrated cameras wherever possible.
  - It makes life so much easier, especially for SfM.
- SfM with 2 cameras is *far* more robust than with a single camera.
  - Triangulate feature points in 3D using stereo.
  - Perform 2D-3D matching to recover the motion.
  - More robust to loss of scale (main problem of 1-camera SfM).
- Any constraint on the setup can be useful
  - E.g. square pixels, zero skew, fixed focal length in each camera
  - E.g. fixed baseline in stereo SfM setup
  - E.g. constrained camera motion on a ground plane
  - Making best use of those constraints may require adapting the algorithms (some known results are described in H&Z).

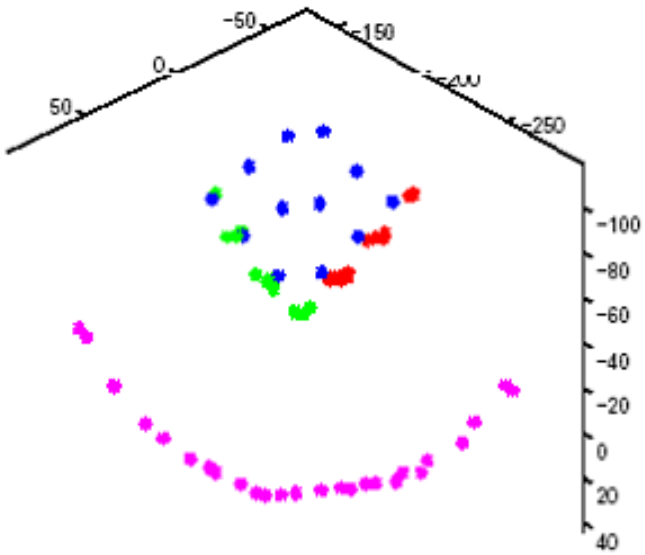
Source: B. Leibe

Structure-from-Motion: Limitations

- Very difficult to reliably estimate metric SfM unless
  - Large (x or y) motion *or*
  - Large field-of-view and depth variation
- Camera calibration important for Euclidean reconstruction
- Need good feature tracker

✓**Correspondence**

✓**Matching**



Source: Steve Seitz

Applications: Large-Scale SfM from Flick

S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, R. Szeliski, Building Rome in a Day, ICCV'09, 2009. (Video from http://grail.cs.washington.edu/rome/)

- Commercial Software Packages

- Voodoo Camera Tracker (http://www.digilab.uni-hannover.de/)

References and Further Reading

- The basic ideas and algorithms for affine and projective SFM:

  D. Forsyth, J. Ponce, Computer Vision – A Modern Approach. Prentice Hall, 2003, **Chapters 12-13**.

- More detailed information

  R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd Ed., Cambridge Univ. Press, 2004. **Chapters 10,18,19**.