CS 532: 3D Computer Vision
7th Set of Notes

Instructor: Philippos Mordohai
Webpage: www.cs.stevens.edu/~mordohai
E-mail: Philippos.Mordohai@stevens.edu
Office: Lieb 215

Logistics

- No class on October 21
- A lecture will be recorded on Friday October 30 at 3:30 in Webcampus studio
 - Please attend if possible

Lecture Outline

- Structure from Motion

- Sources:
  - Slides by R. Szeliski, S. Seitz, N. Snavely. S. Lazebnik, M. Hebert, S. Choudhary
  - Visual Odometry by D. Nister, O. Naroditsky, J. Bergen (2006)
  - Parallel Tracking and Mapping by G. Klein and D. Murray (2007)
  - Visual SLAM: Why filter? by H. Strasdat, J.M.M. Montiel, A.J. Davison (2012)

Structure from Motion

Unknown camera viewpoints
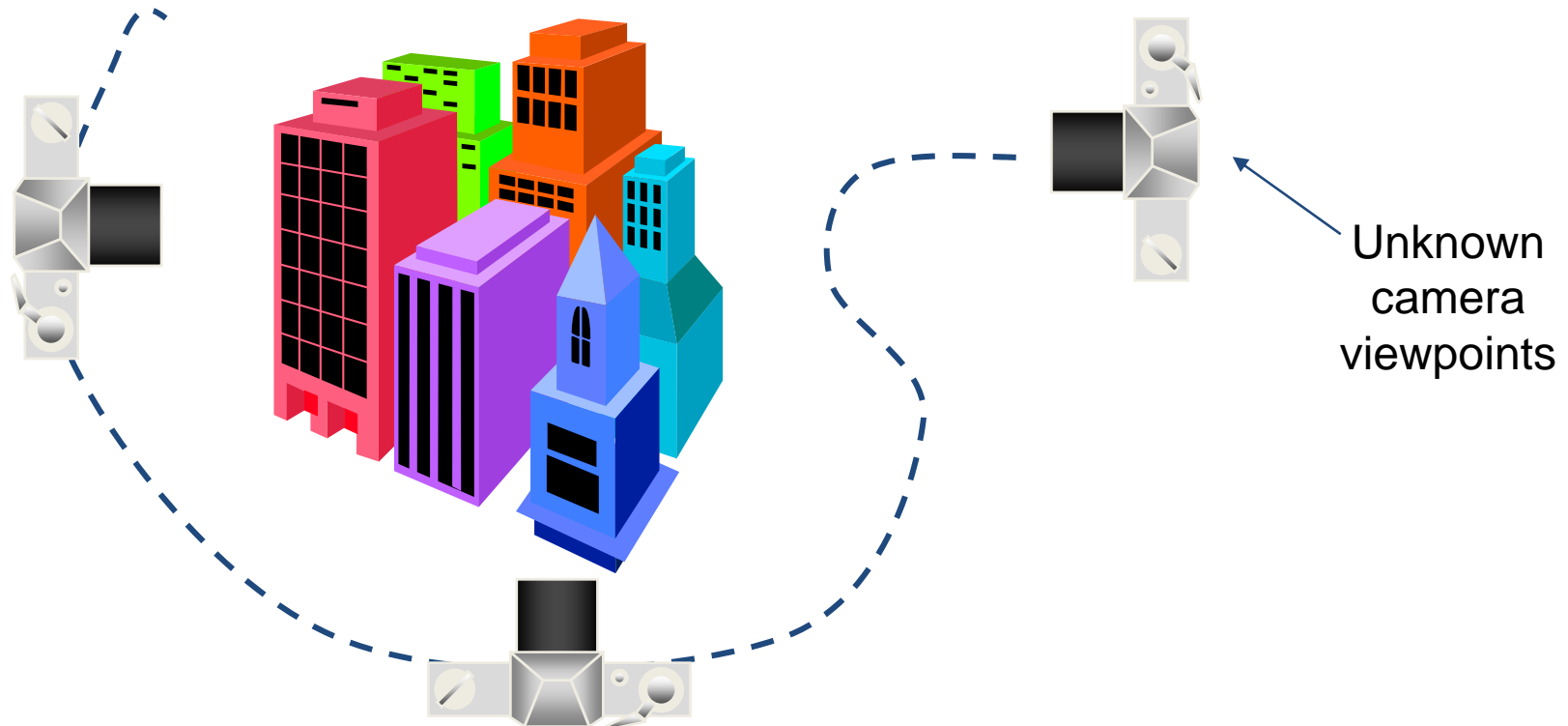
- Reconstruct
  - Scene geometry
  - Camera motion

Input: Feature Tracks
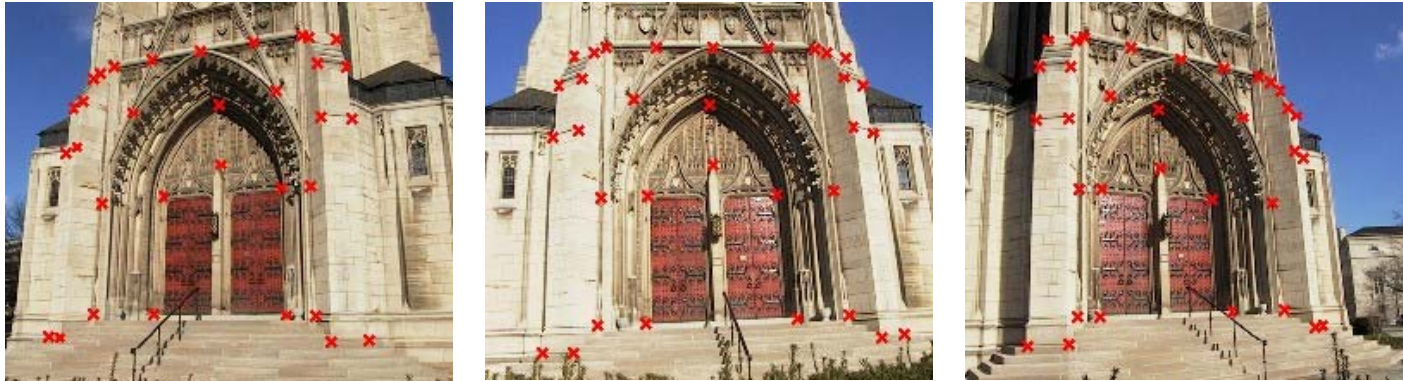
- **Detect good features**
  - corners, line segments

- **Find correspondences between frames**
  - Lucas & Kanade-style motion estimation
  - window-based correlation

Structure from Motion

- Given many points in *correspondence* across several images, $\{(u_{ij}, v_{ij})\}$, simultaneously compute the 3D location $\mathbf{x}_i$ and camera (or *motion*) parameters $(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j)$

$$\hat{u}_{ij} = f(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{x}_i)$$
$$\hat{v}_{ij} = g(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{x}_i)$$

- Two main variants: calibrated, and uncalibrated (sometimes associated with Euclidean and projective reconstructions)

Number of Constraints

$$\widehat{u}_{ij} = f(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{x}_i)$$
$$\widehat{v}_{ij} = g(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{x}_i)$$

- How many points do we need to match?
- 2 frames:

  (R,t): 5 dof + 3n point locations $\leq$

  4n point measurements $\Rightarrow$ n $\geq$ 5

- k frames:

 6(k-1)-1 + 3n $\leq$ 2kn
- always want to use many more

=> why 5 dof for 2 cameras and 6(k-1)-1 for k cameras?

Bundle Adjustment

- What makes this non-linear minimization hard?
  - many parameters: potentially slow
  - poorer conditioning (high correlation)
  - potentially lots of outliers
 - gauge (coordinate) freedom

Structure from Motion

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates
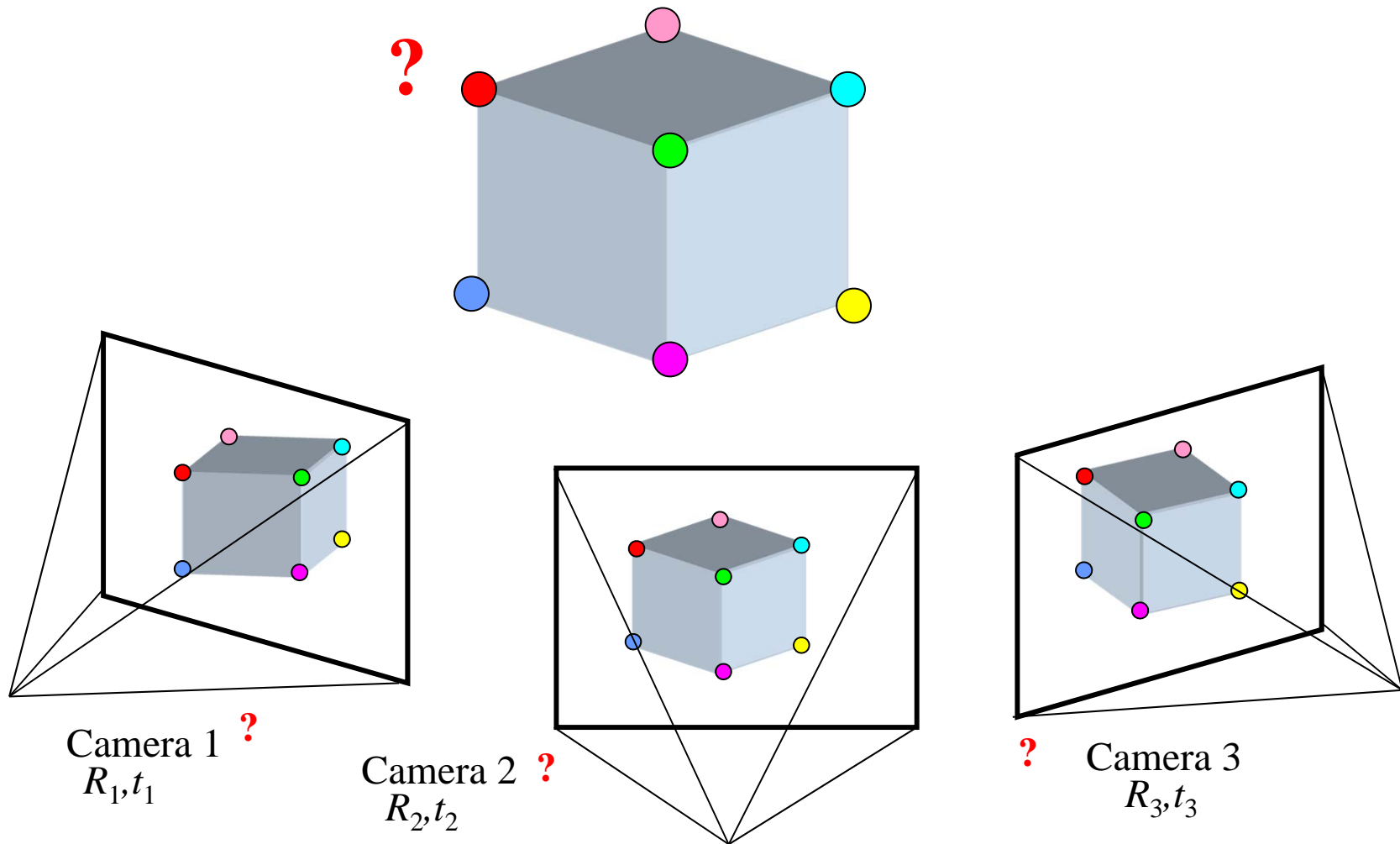
?

Camera 1 **?**
R_1, t_1

Camera 2 **?**
R_2, t_2

? Camera 3
R_3, t_3

9

Structure from Motion

- Given: m images of n fixed 3D points

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

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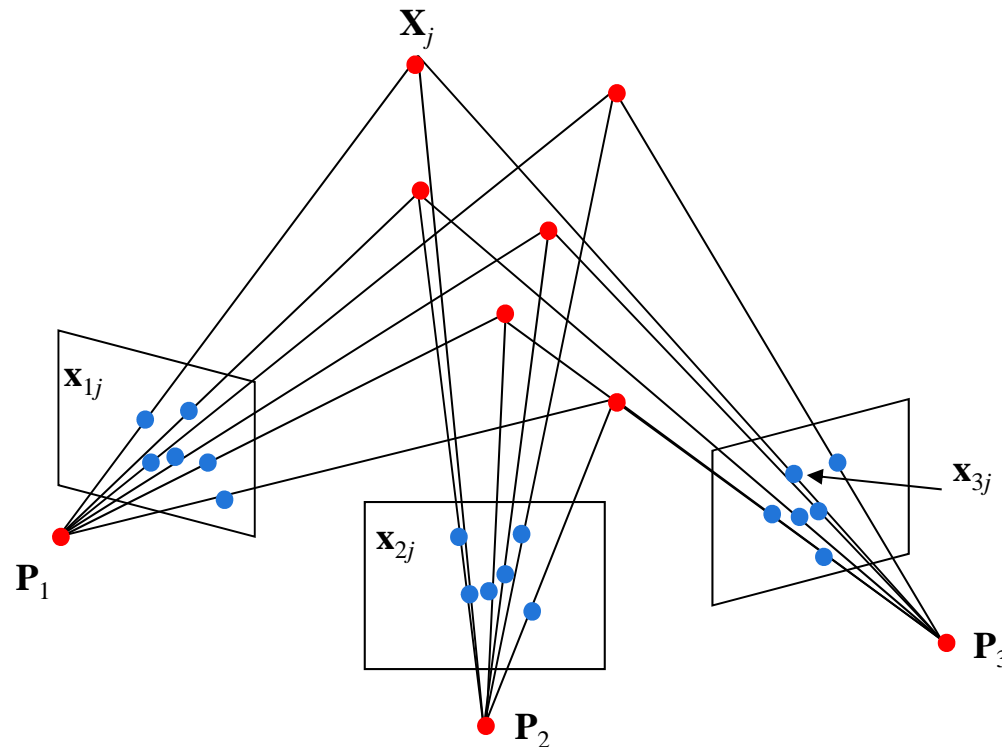
Structure from Motion Ambiguity

- 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:

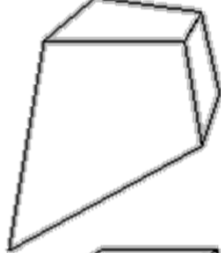$$\mathbf{x} = \mathbf{PX} = \left(\frac{1}{k} \mathbf{P} \right) (k\mathbf{X})$$
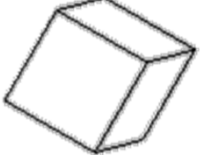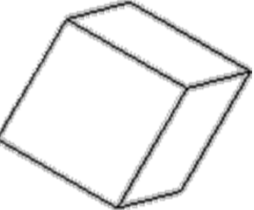
It is impossible to recover the absolute scale of the scene!

Structure from Motion Ambiguity

- 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

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

Types of Ambiguity

Projective
15dof
$$\begin{bmatrix} A & t \\ v^{\mathsf{T}} & v \end{bmatrix}$$
Preserves intersection and tangency

Affine
12dof
$$\begin{bmatrix} A & t \\ 0^{\mathsf{T}} & 1 \end{bmatrix}$$
Preserves parallellism, volume ratios

Similarity
7dof
$$\begin{bmatrix} s\,R & t \\ 0^{\mathsf{T}} & 1 \end{bmatrix}$$
Preserves angles, ratios of length

Euclidean
6dof
$$\begin{bmatrix} R & t \\ 0^{\mathsf{T}} & 1 \end{bmatrix}$$
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
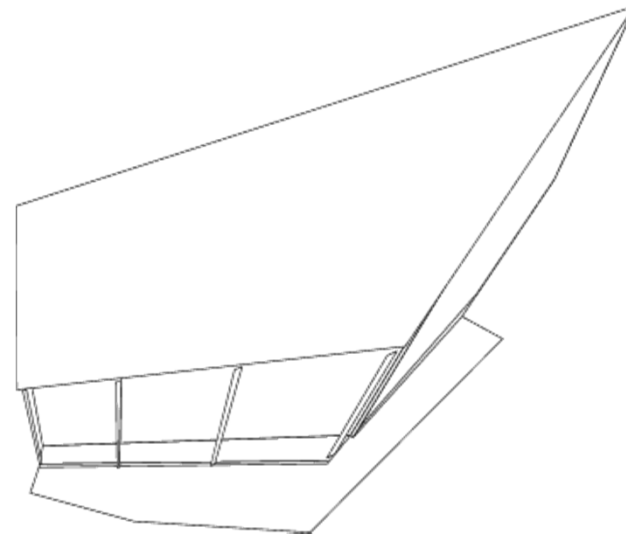
13

Projective Ambiguity
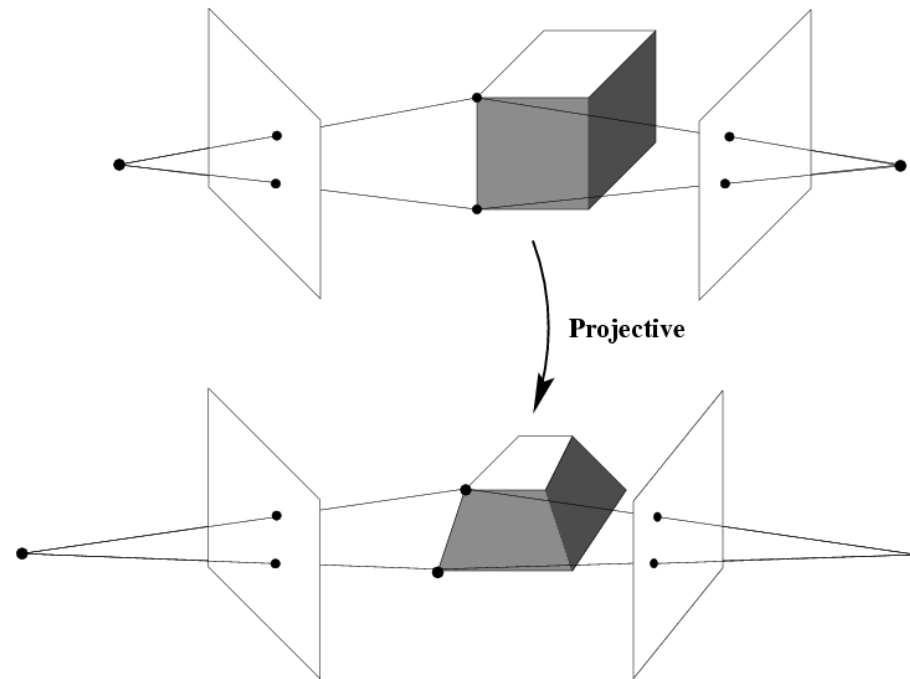
Projective

$$Q_p = \begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$$

$$x = PX = \left(PQ_P^{-1}\right)\left(Q_P \, X\right)$$

Projective Ambiguity

Affine Ambiguity

Affine

$$\mathbf{Q_A} = \begin{bmatrix} A & t \\ 0^\mathsf{T} & 1 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{PX} = \left(\mathbf{PQ_A^{-1}}\right)\left(\mathbf{Q_A\,X}\right)$$
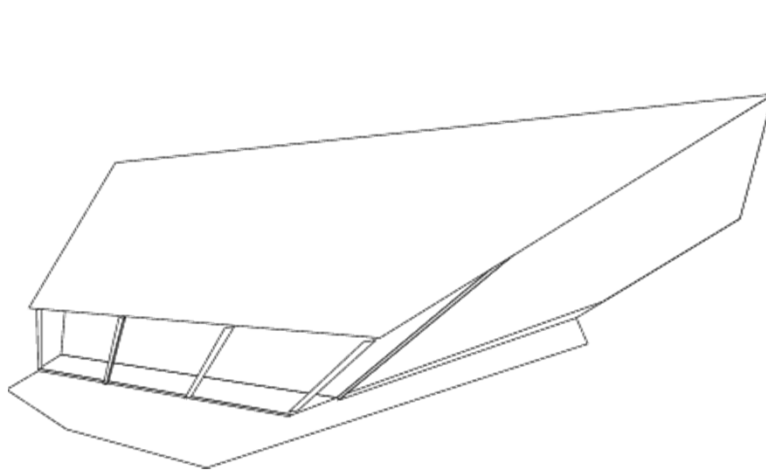
Affine Ambiguity
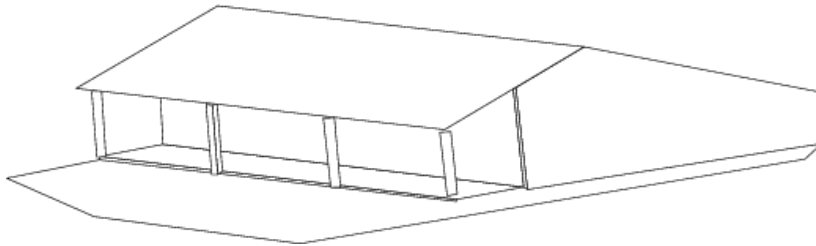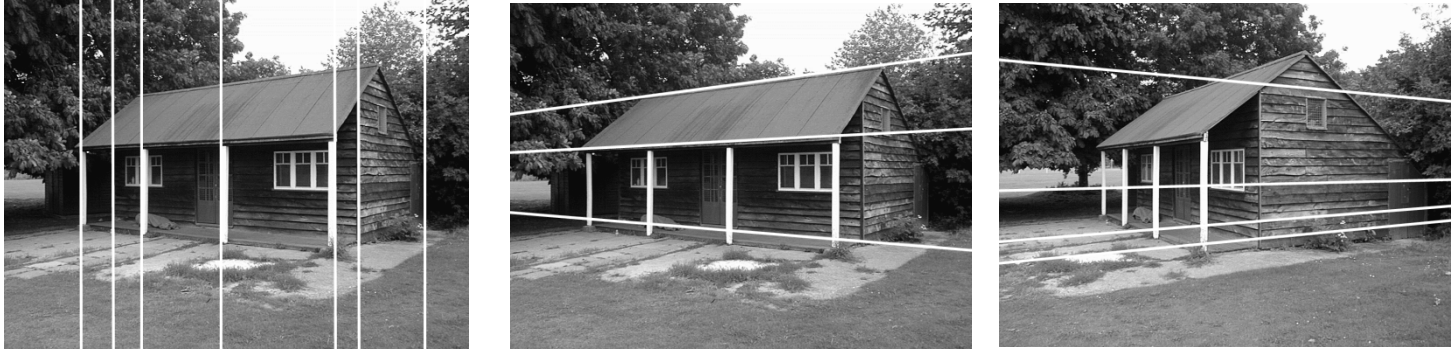
Similarity Ambiguity
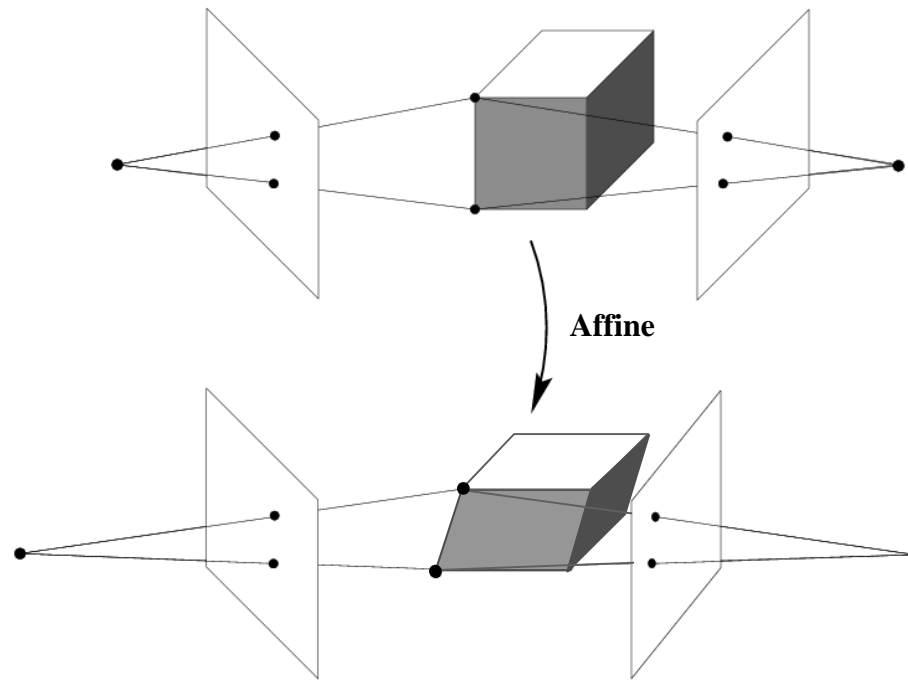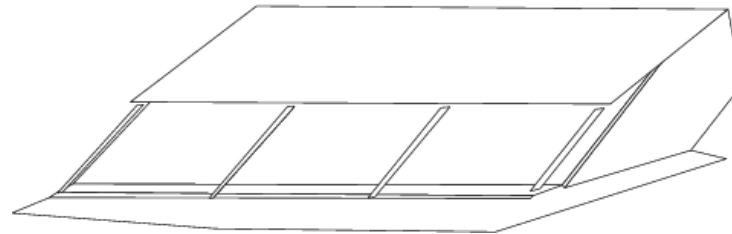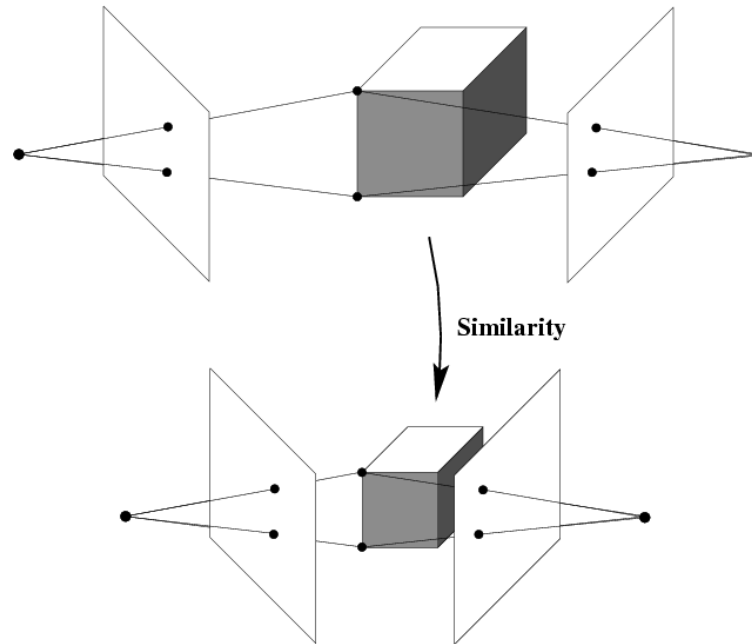

$$\mathbf{Q_s} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{PX} = \left(\mathbf{PQ_S^{-1}}\right)\left(\mathbf{Q_S X}\right)$$

Similarity Ambiguity
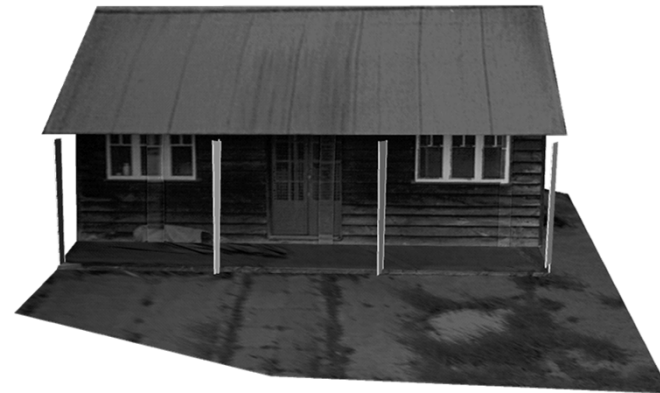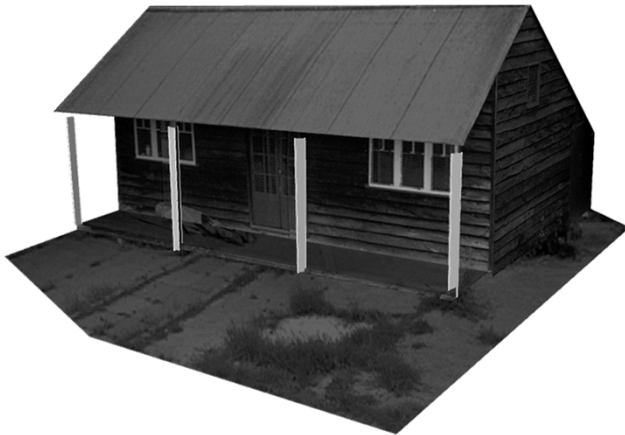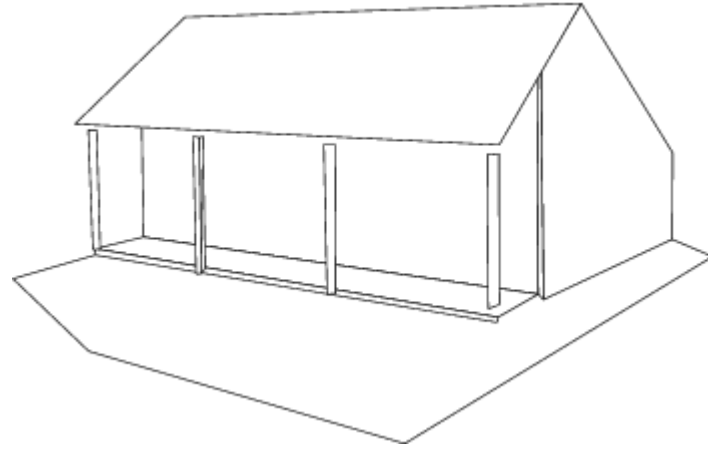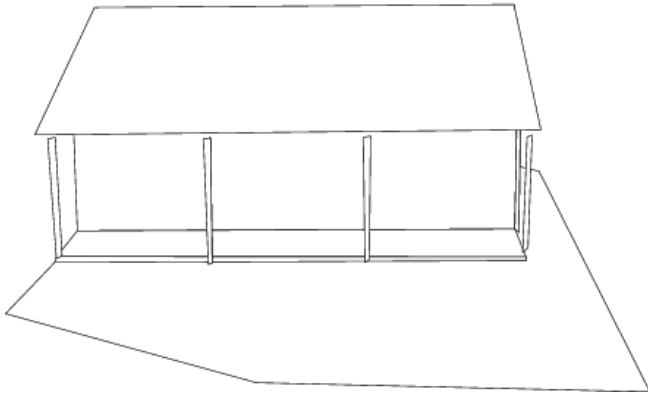
Structure from Motion: Affine Cameras

Structure from Motion

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

perspective · weak perspective · center at infinity

increasing focal length

increasing distance from camera
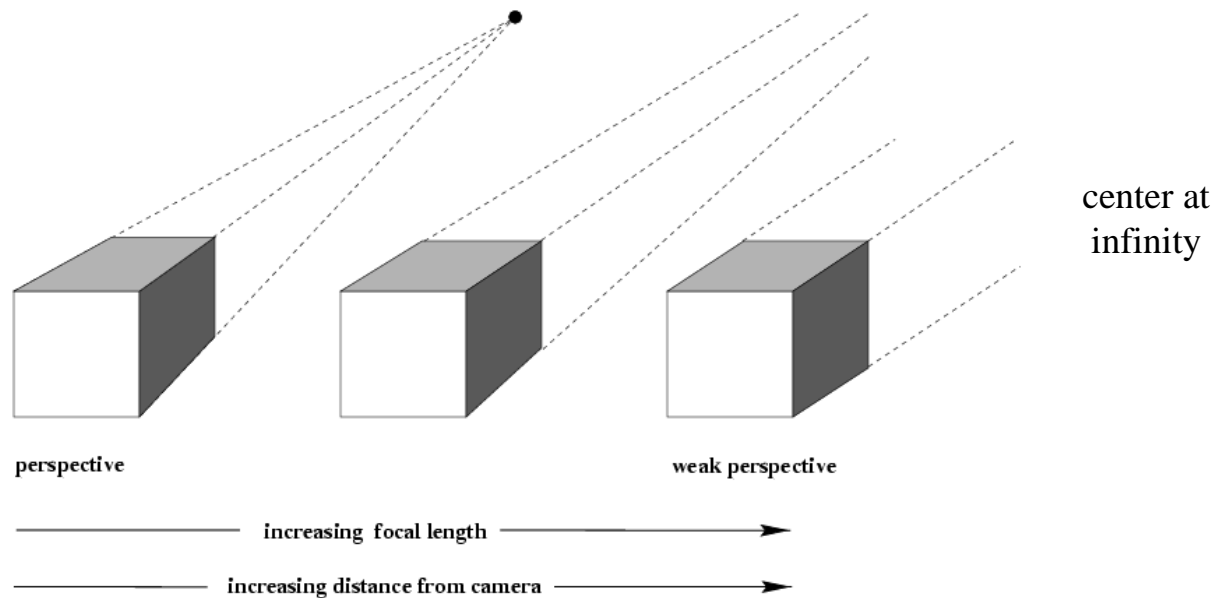
Orthographic Projection

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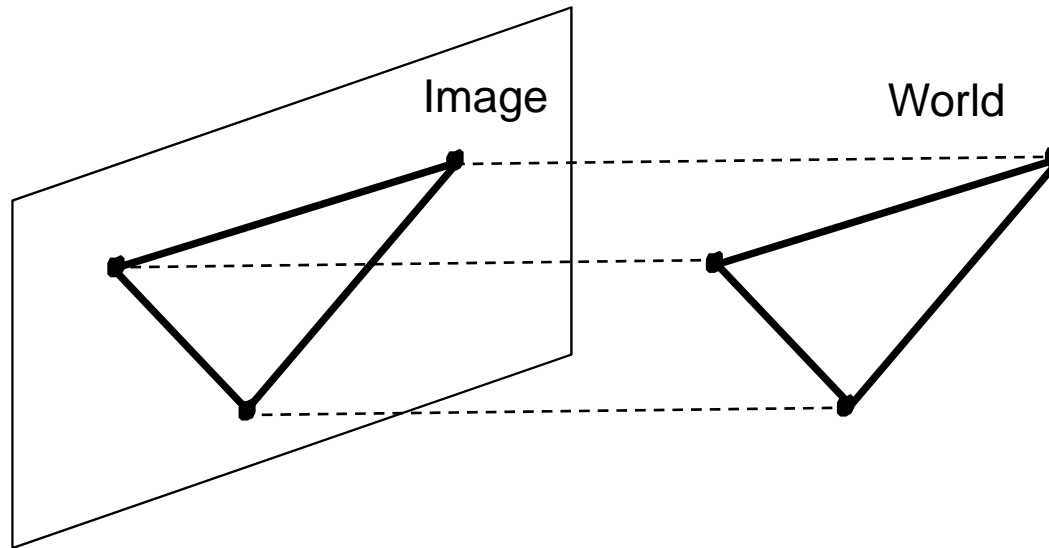
  - Projection matrix:

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

Affine Cameras

Orthographic Projection
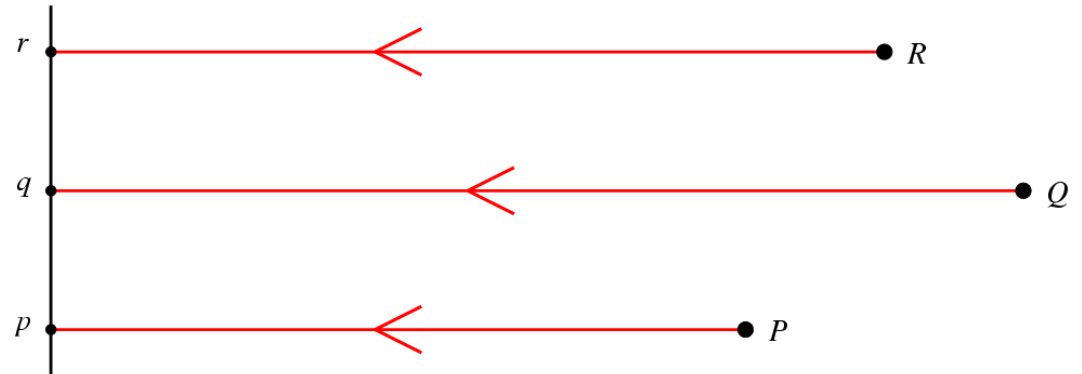
Parallel Projection
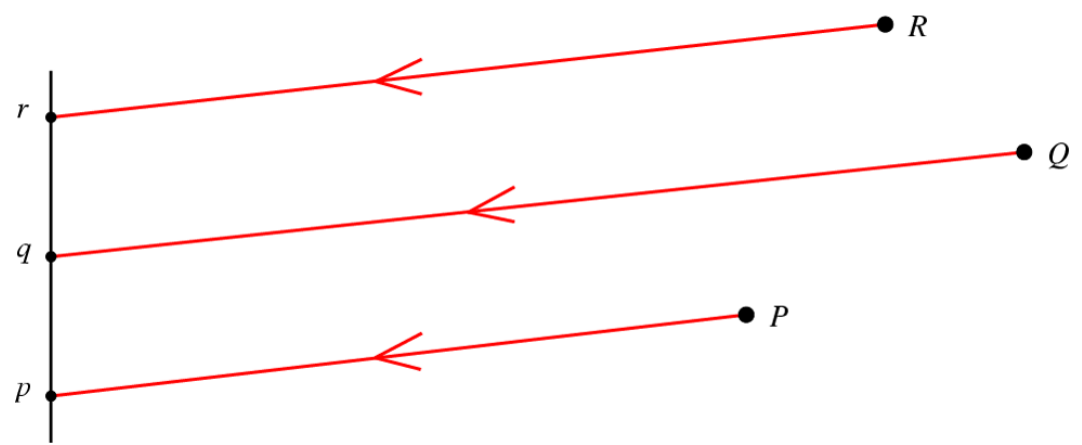
Affine Cameras

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

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

Affine projection is a linear mapping + translation in inhomogeneous coordinates

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

$\mathbf{x}$

$\mathbf{a}_2$

$\mathbf{a}_1$

$\mathbf{X}$

Projection of world origin

24
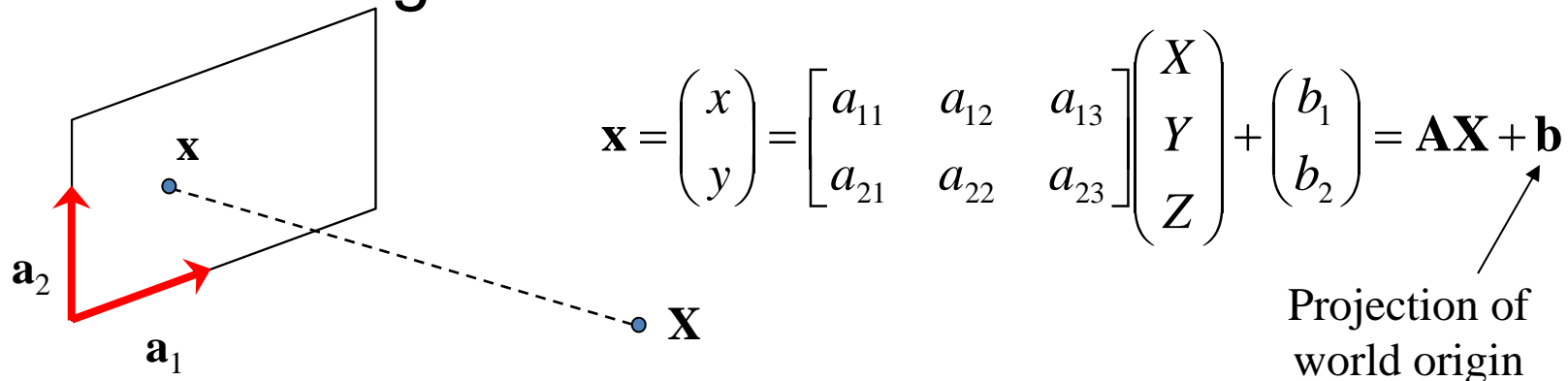
Affine Structure from Motion

- 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 $\mathbf{Q}$ (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 $2mn$ knowns and $8m + 3n$ unknowns (minus 12 dof for affine ambiguity)

- Thus, we must have $2mn >= 8m + 3n - 12$

- For two views, we need four point correspondences

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

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

Affine Structure from Motion

- $2mn$ data (measurement) matrix:

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix}$$

cameras $(2\,m)$

points (n)

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

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

points $(3 \times n)$

cameras
$(2\,m \times 3)$

The measurement matrix $\mathbf{D} = \mathbf{MS}$ must have rank 3!

Factorizing the Measurement Matrix

$2m$

Measurements = **Motion** × **Shape**

n

3

3

n

$$D = MS$$
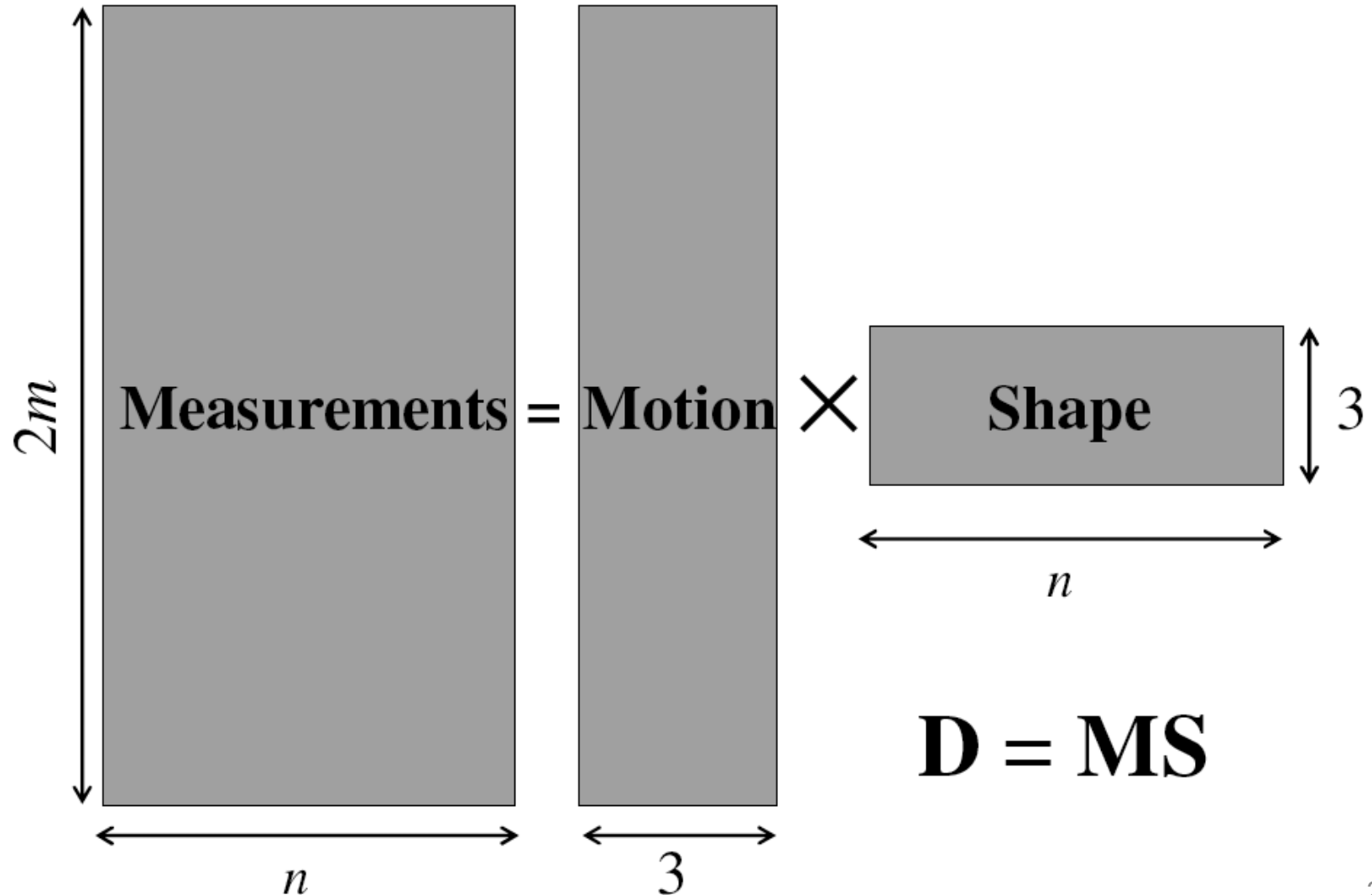
Factorizing the Measurement Matrix

- Singular value decomposition of D:
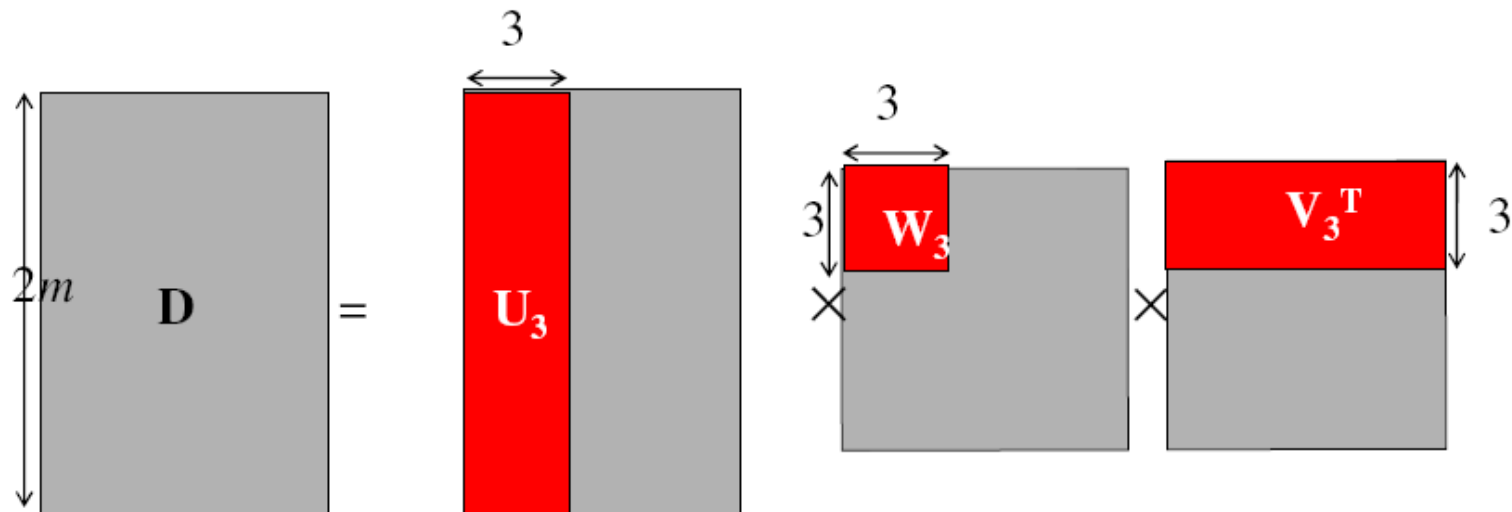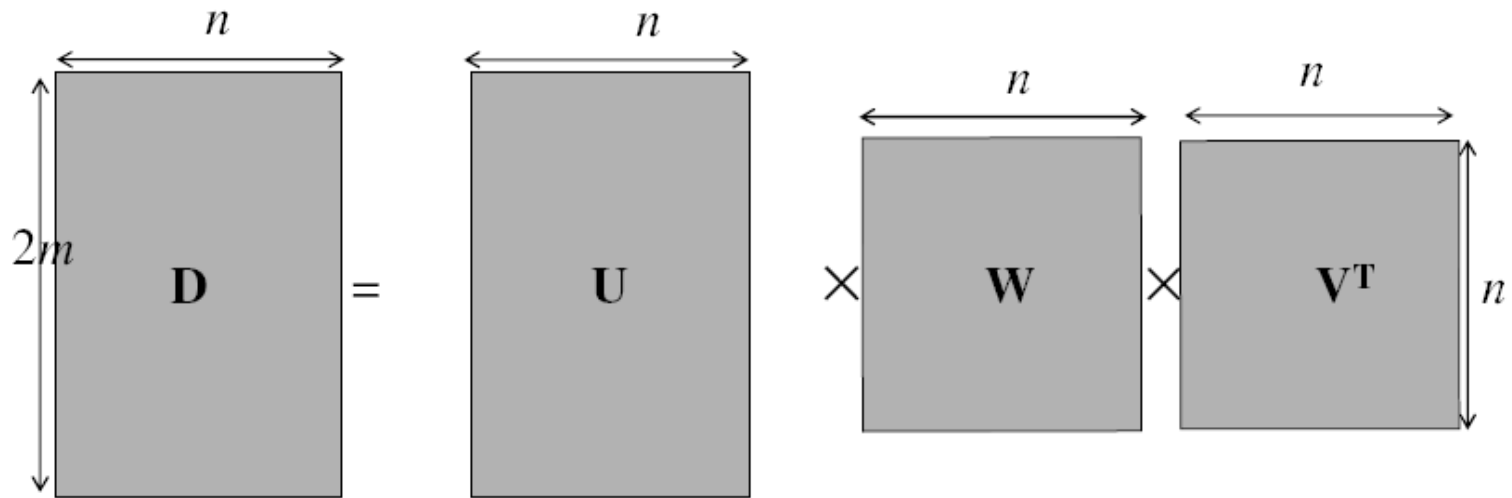
Factorizing the Measurement Matrix
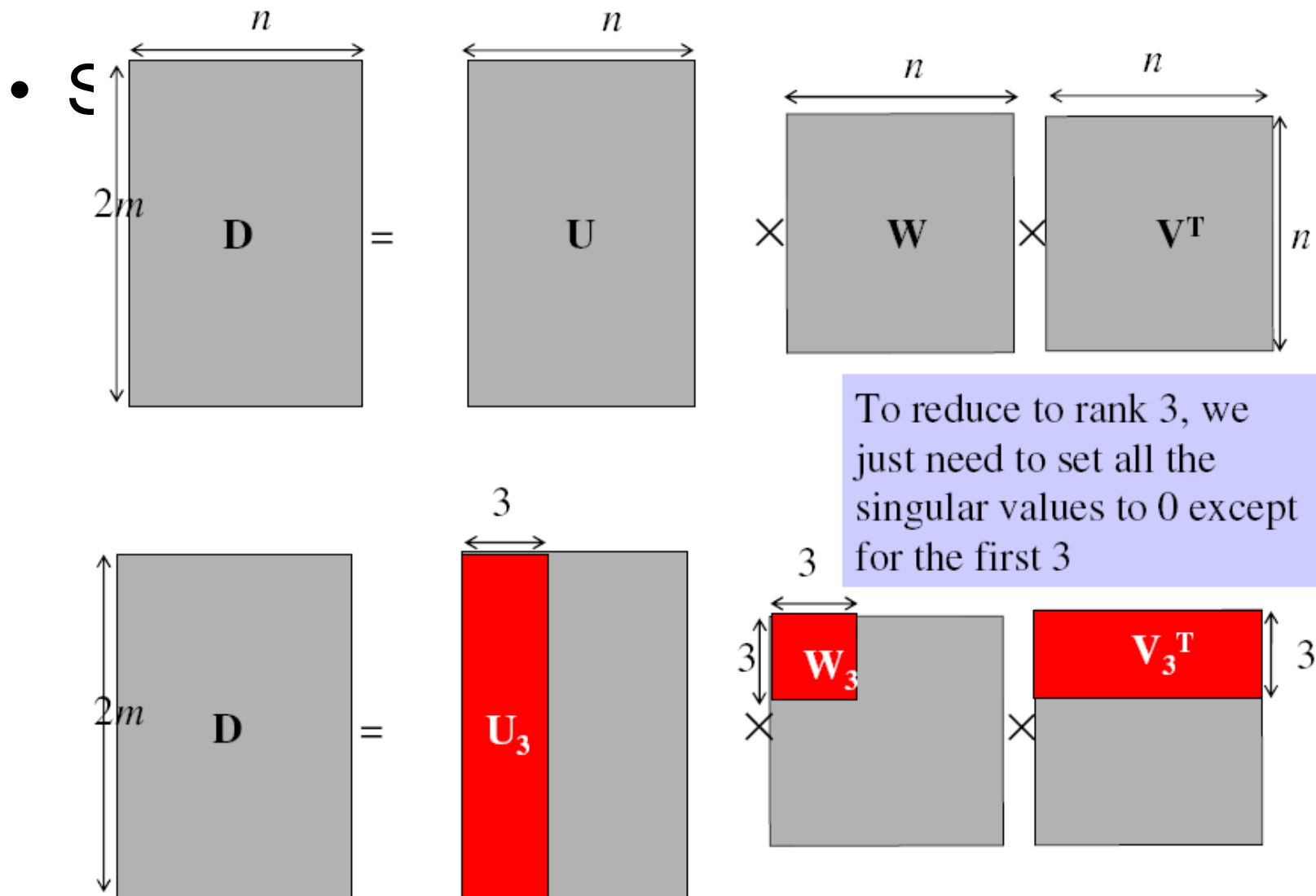
- S

n

n

n

n

$2m$

D = U × W × V^T n

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

3

3

3

3

$2m$

D = U_3 × W_3 × V_3^T 3

Factorizing the Measurement Matrix

$$2m \left\uparrow \boxed{D} \right\downarrow = \boxed{U_3} \times \boxed{W_3} \times \boxed{V_3^{T}}$$

Factorizing the Measurement Matrix

$2m$ $\left\{\vphantom{D}\right.$ **D** $= $ **U$_3$** $\times 3 \updownarrow$ **W$_3$** $\times$ **V$_3$^{T}** $\updownarrow 3$

3 ← → (under U$_3$)

3 (over W$_3$), n (over V$_3^T$)
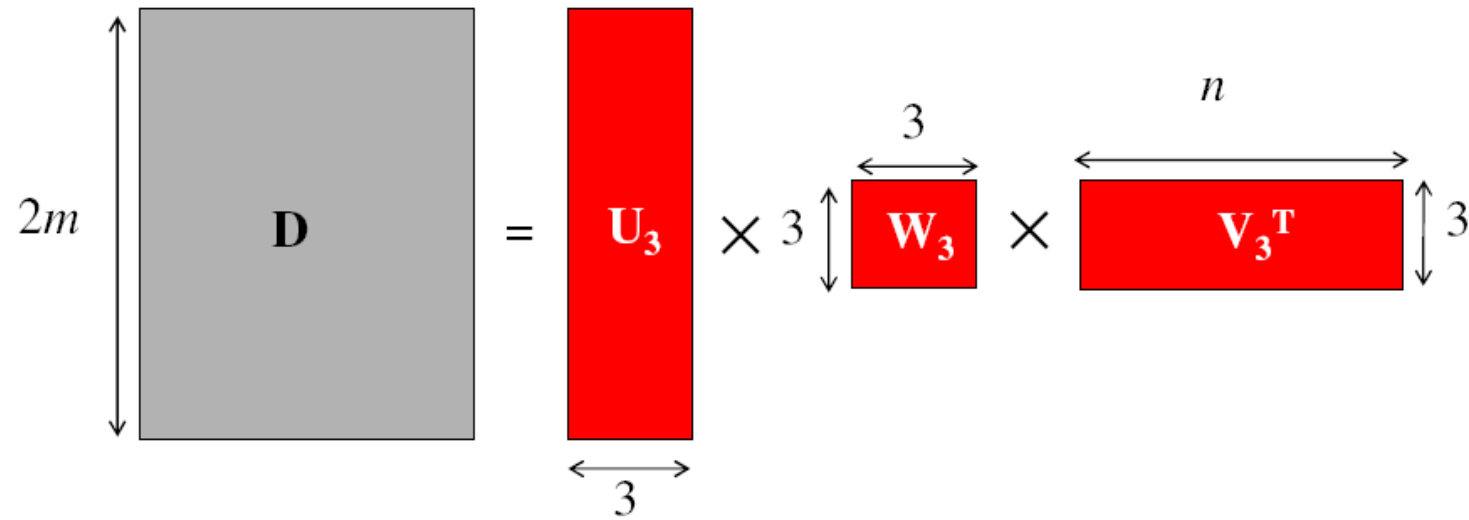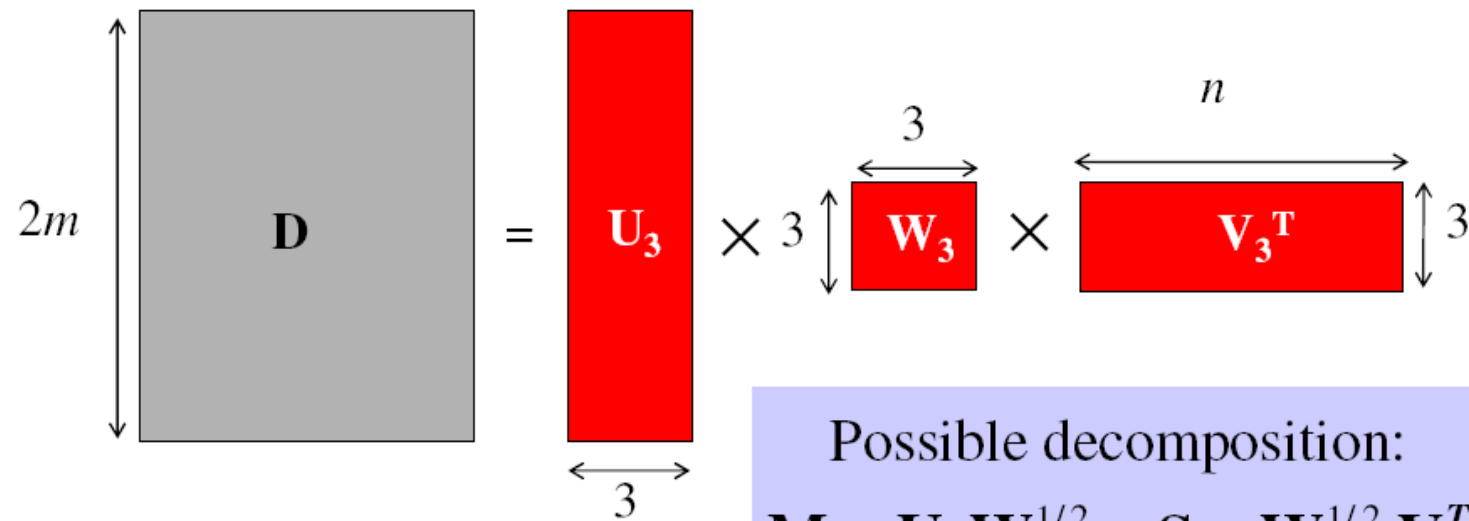
Possible decomposition:

$$M = U_3 W_3^{1/2} \qquad S = W_3^{1/2} V_3^T$$

D $=$ **M** $\times$ **S**
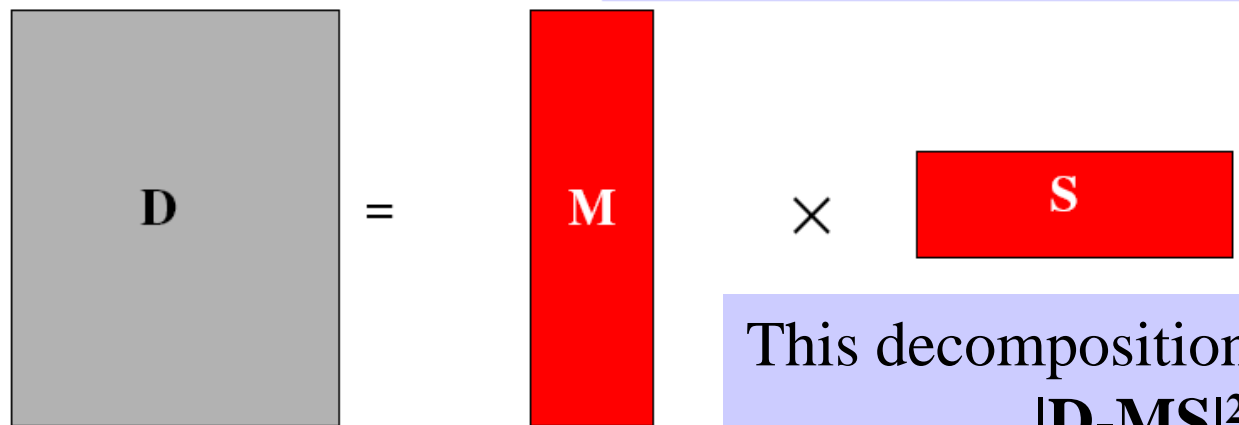
This decomposition minimizes $|D-MS|^2$

Affine Ambiguity

$$D = M \times S$$
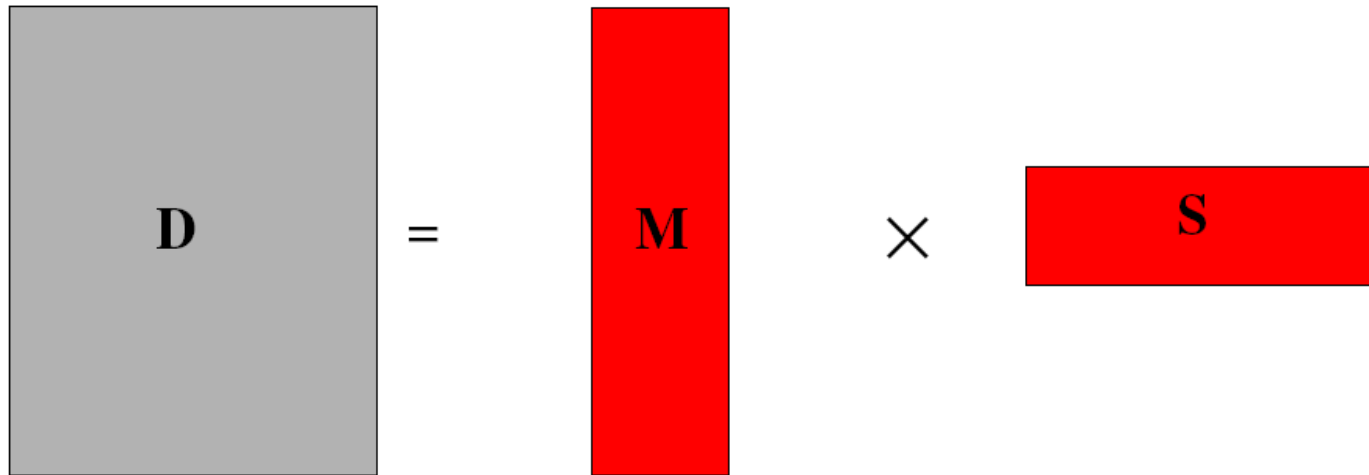
- The decomposition is not unique. We get the same D by using any 3×3 matrix C and applying the transformations M → MC, S →C^{-1}S

- 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)

Algorithm Summary

- Given: m images and n features $\mathbf{x}_{ij}$
- For each image i, center the feature coordinates
- Construct a $2m \times n$ measurement matrix $\mathbf{D}$:
  - 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 $\mathbf{D}$:
  - Compute SVD: $\mathbf{D} = \mathbf{U}\,\mathbf{W}\,\mathbf{V}^\mathsf{T}$
  - Create $\mathbf{U}_3$ by taking the first 3 columns of $\mathbf{U}$
  - Create $\mathbf{V}_3$ by taking the first 3 columns of $\mathbf{V}$
  - Create $\mathbf{W}_3$ by taking the upper left 3×3 block of $\mathbf{W}$
- Create the motion and shape matrices:
  - $\mathbf{M} = \mathbf{U}_3\mathbf{W}_3^{\frac{1}{2}}$ and $\mathbf{S} = \mathbf{W}_3^{\frac{1}{2}}\,\mathbf{V}_3^\mathsf{T}$ (or $\mathbf{M} = \mathbf{U}_3$ and $\mathbf{S} = \mathbf{W}_3\mathbf{V}_3^\mathsf{T}$)
- Eliminate affine ambiguity

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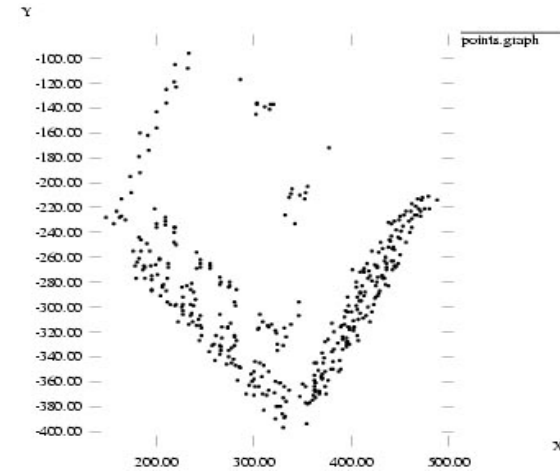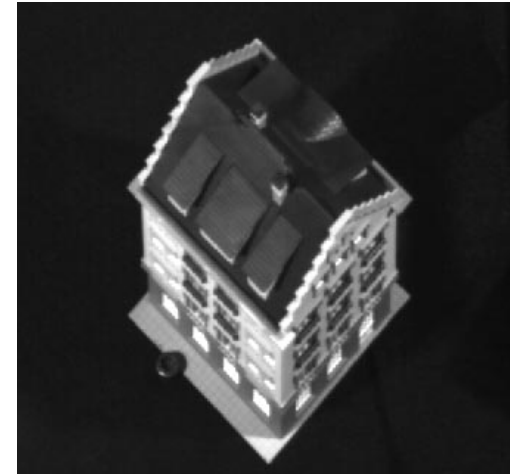
C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Structure from Motion: Perspective Cameras

Projective 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 $\mathbf{x}_{ij}$
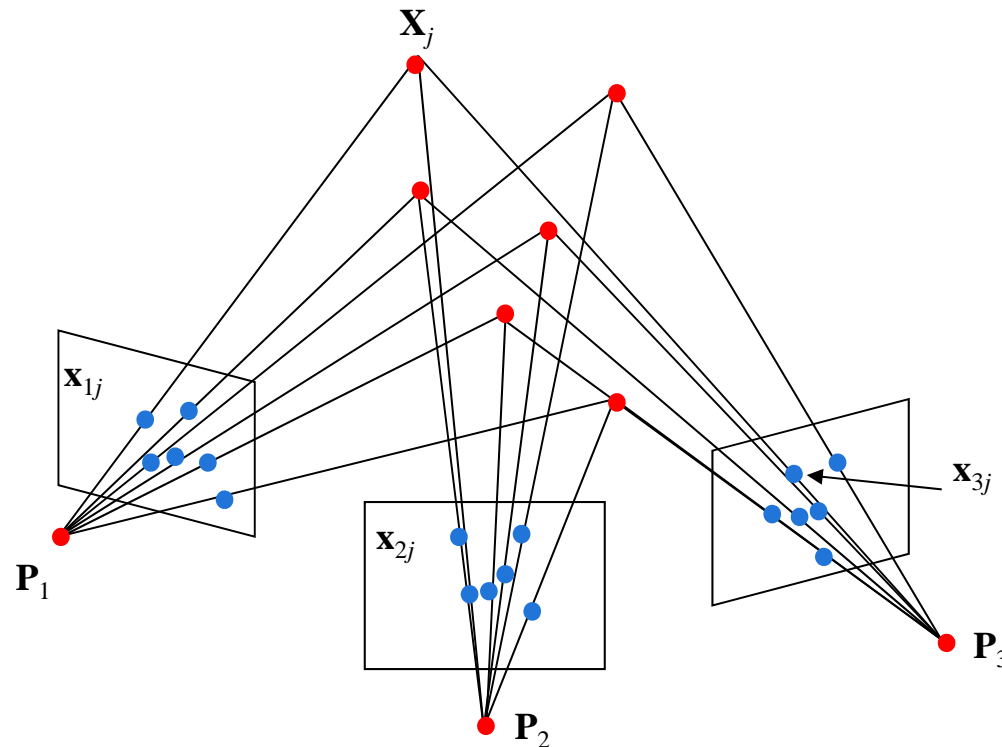
Projective Structure from Motion

- Given: m images of n fixed 3D points

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

- 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 $\mathbf{Q}$:

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

- We can solve for structure and motion when

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

- For two cameras, at least 7 points are needed

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

Sequential Structure from Motion

- Initialize motion from two images using fundamental matrix

- Initialize structure by triangulation

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

points

cameras

Sequential Structure from Motion

•Initialize motion from two images
using fundamental matrix

•Initialize structure by triangulation

•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

points

cameras

Sequential Structure from motion

•Initialize motion from two images using fundamental matrix

•Initialize structure by triangulation

•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

43
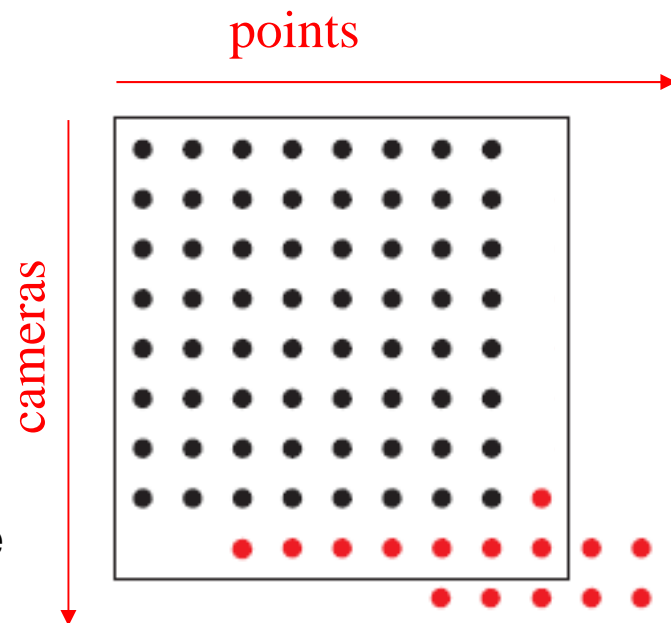
Bundle Adjustment

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

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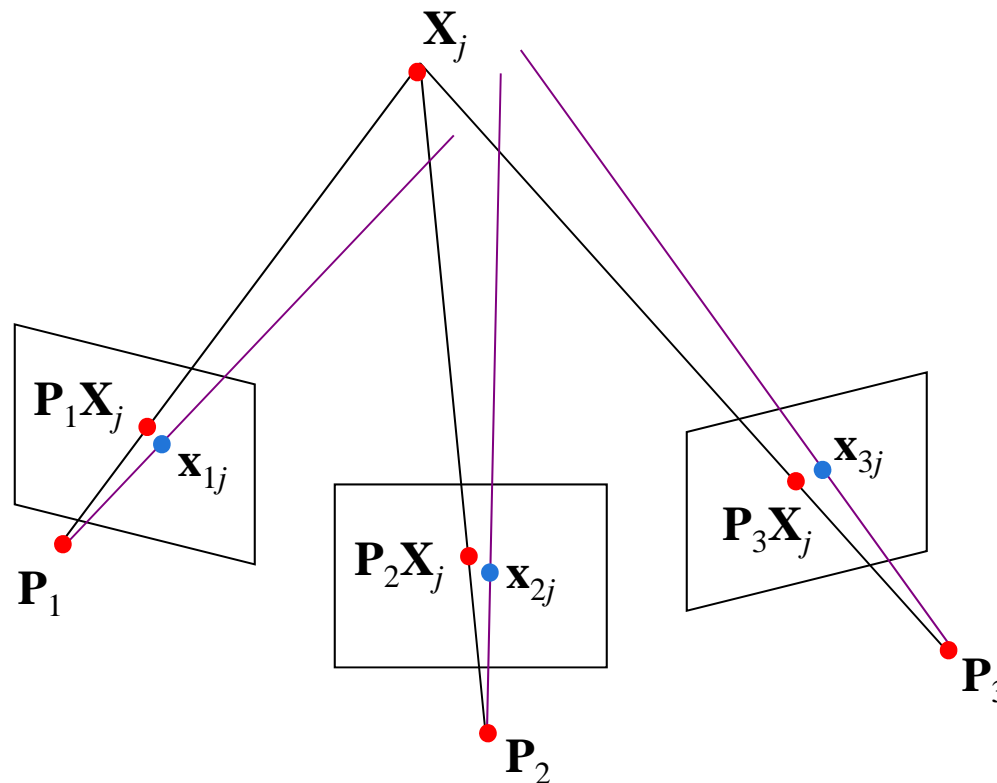
Self-calibration

- 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 $\mathbf{Q}$ 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
- Can use vanishing points

Triangulation

More formulations exist

Triangulation: Linear Solution

- Generally, rays C→x and C'→x' will not exactly intersect

- Can solve via SVD, finding a least squares solution to a system of equations

$$\mathbf{x} = \mathbf{PX}$$

$$\mathbf{x}' = \mathbf{P'X}$$
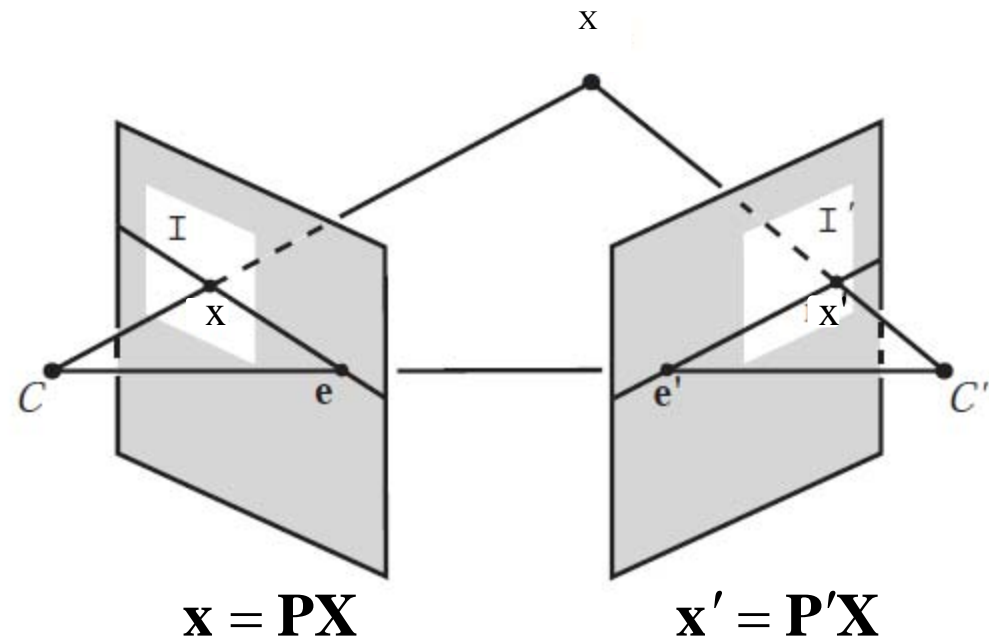
$$\mathbf{AX} = \mathbf{0} \quad \mathbf{A} = \begin{bmatrix} u\mathbf{p}_3^T - \mathbf{p}_1^T \\ v\mathbf{p}_3^T - \mathbf{p}_2^T \\ u'\mathbf{p}_3'^T - \mathbf{p}_1'^T \\ v'\mathbf{p}_3'^T - \mathbf{p}_2'^T \end{bmatrix}$$

From x×PX = 0 and x'×PX' = 0

47

Triangulation: Linear Solution

Given **P**, **P'**, **x**, **x'**

1. Precondition points and projection matrices
2. Create matrix **A**
3. [U, S, V] = svd(A)
4. X = V(:, end)

Pros and Cons

- Works for any number of corresponding images
- Not projectively invariant

$$\mathbf{x} = w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \mathbf{x}' = w\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{bmatrix} \qquad \mathbf{P}' = \begin{bmatrix} \mathbf{p}_1'^T \\ \mathbf{p}_2'^T \\ \mathbf{p}_3'^T \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} u\mathbf{p}_3^T - \mathbf{p}_1^T \\ v\mathbf{p}_3^T - \mathbf{p}_2^T \\ u'\mathbf{p}_3'^T - \mathbf{p}_1'^T \\ v'\mathbf{p}_3'^T - \mathbf{p}_2'^T \end{bmatrix}$$

Triangulation: Non-linear Solution

- Minimize projected error while satisfying $\widehat{x}'^T F \widehat{x} = 0$

$$cost(X) = dist(x, \widehat{x})^2 + dist(x', \widehat{x}')^2$$
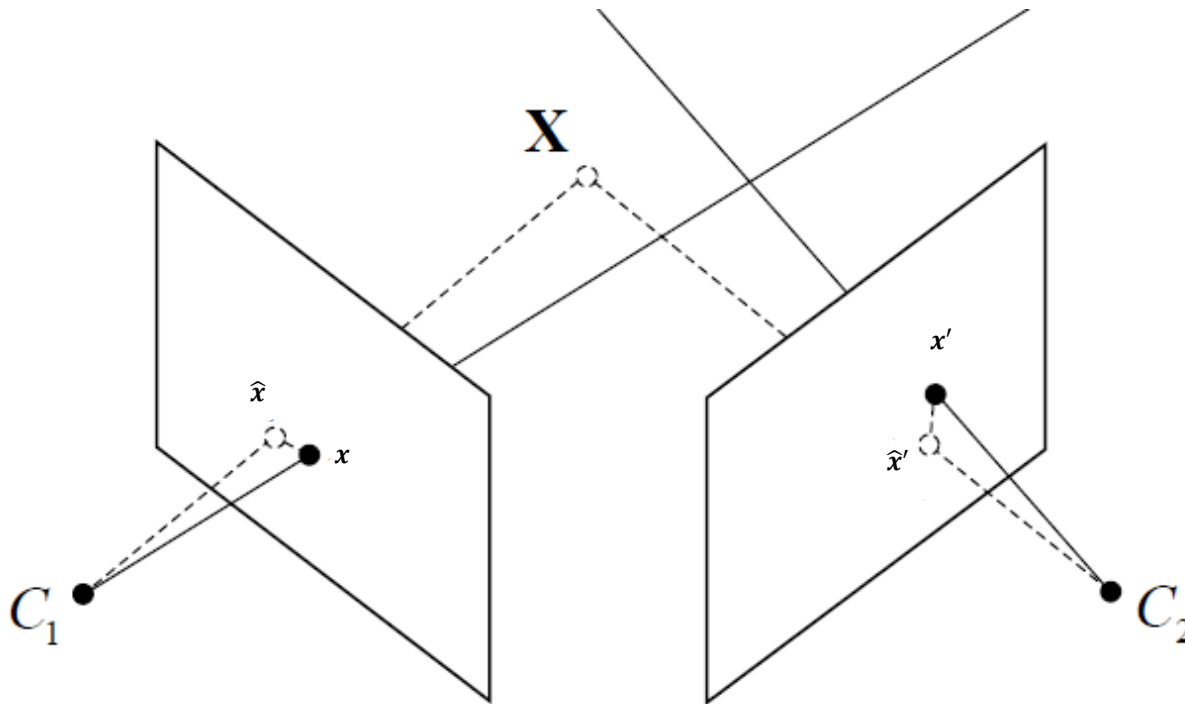
Triangulation: Non-linear Solution

- Minimize projected error while satisfying

$$\widehat{\boldsymbol{x}}'^{T} \boldsymbol{F}\widehat{\boldsymbol{x}}=0$$

$$cost(\boldsymbol{X}) = dist(\boldsymbol{x},\widehat{\boldsymbol{x}})^2 + dist(\boldsymbol{x}',\widehat{\boldsymbol{x}}')^2$$
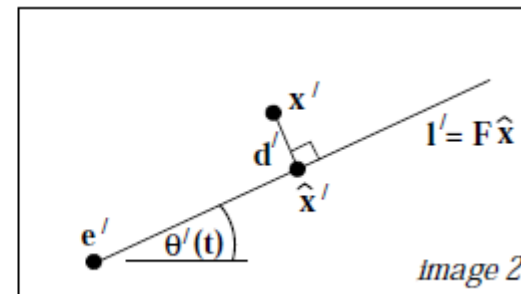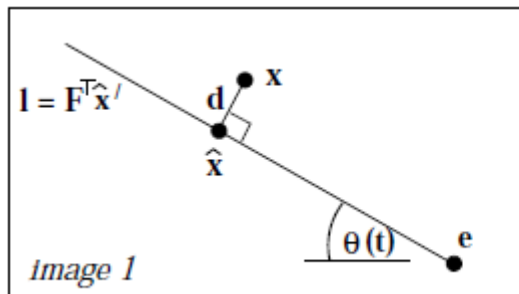
- Solution is a 6-degree polynomial of t, minimizing $d(\mathbf{x},\mathbf{l}(t))^2 + d(\mathbf{x}',\mathbf{l}'(t))^2$

Bundle Adjustment

Bundle Adjustment

- Refines a visual reconstruction to produce jointly optimal 3D structure and viewing parameters

- *'Bundle'* refers to the bundle of light rays leaving each 3D feature and converging on each camera center.

Reprojection Error

reprojection error $\|\mathbf{q}_{ij} - P(C_i, \mathbf{X}_j)\|$

objective function:

$$g(C, X) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} \|\mathbf{q}_{ij} - P(C_i, \mathbf{X}_j)\|^2$$
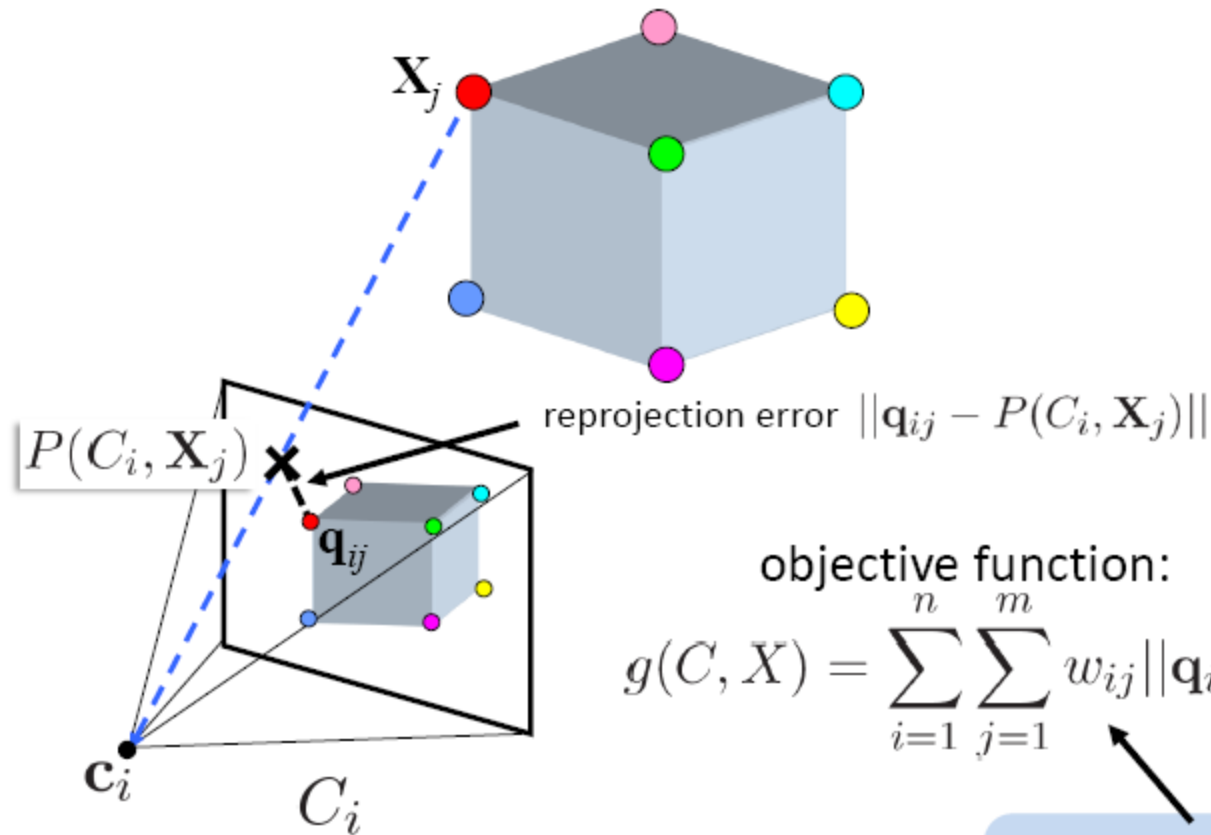
indicator variable:
1 if point *j* is visible in camera *i*
0 otherwise

Notation

- Structure and Cameras being parameterized by a single large vector **x**

- Small displacement in **x** represented by $\partial \boldsymbol{x}$

- Observations denoted by $\underline{z}$

- Predicted values at parameter value **x**, denoted by z = z(x)

- Residual prediction error, $\Delta z(\boldsymbol{x}) = \underline{z} - z(\boldsymbol{x})$

Objective Function

- Minimization of weighted sum of squared error (SSE) cost function:

$$f(\mathsf{x}) \equiv \tfrac{1}{2} \sum_i \triangle \mathsf{z}_i(\mathsf{x})^\top W_i \triangle \mathsf{z}_i(\mathsf{x}) \,, \qquad \triangle \mathsf{z}_i(\mathsf{x}) \equiv \underline{\mathsf{z}}_i - \mathsf{z}_i(\mathsf{x})$$

Optimization Techniques

- Gradient Descent Method
- Newton-Raphson Method
- Gauss – Newton Method
- Levenberg – Marquardt Method

Gradient Descent Method

- A first-order optimization algorithm.

- To find a local minimum of a function using gradient descent, one takes steps proportional to the *negative* of the gradient of the function at the current point.

- It is robust when x is far from optimum but has poor final convergence

While $k<k_{max}$

$$x_k = x_{k-1} - \lambda \nabla f(x_{k-1})$$

57
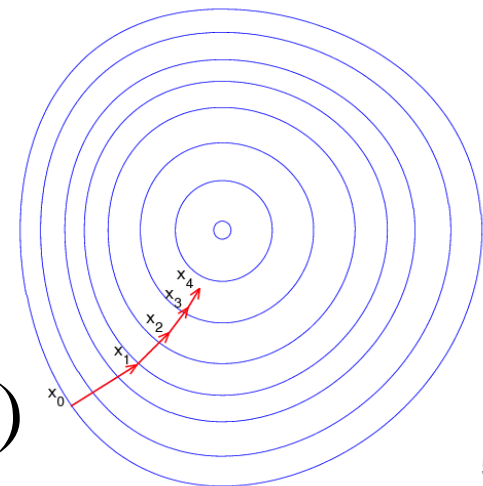
Newton – Raphson Method

- Second order optimization method
- Newton's method can often converge remarkably quickly, especially if the iteration begins "sufficiently near" the solution

$$f(x + \delta x) \approx f(x) + g^{\top}\delta x + \tfrac{1}{2}\delta x^{\top} H\,\delta x \qquad g \equiv \frac{df}{dx}(x) \qquad H \equiv \frac{d^2f}{dx^2}(x)$$

quadratic local model   gradient vector   Hessian matrix

$$\frac{df}{dx}(x + \delta x) \approx H\,\delta x + g \qquad \boxed{\delta x = -H^{-1}g}$$

Newton – Raphson Method

- For a quadratic function it converges in one iteration

- For other general function, its asymptotic convergence is quadratic

- The disadvantage of this method is the high computation complexity of H^{-1}

Gauss – Newton Method

- The Gauss-Newton algorithm is a method used to solve non-linear least squares problems

$$f(x) \equiv \tfrac{1}{2} \triangle z(x)^\top W \triangle z(x)$$

$$g \equiv \frac{df}{dx} = \triangle z^\top W J \qquad\qquad H \equiv \frac{d^2 f}{dx^2} = J^\top W J + \sum_i (\triangle z^\top W)_i \frac{d^2 z_i}{dx^2}$$

$$H \approx J^\top W J. \qquad\qquad \frac{d^2 z_i}{dx^2} \approx 0$$

$$(J^\top W J)\, \delta x = -J^\top W \triangle z$$

Gauss – Newton Method

- For well-parameterized bundle problems under an outlier-free least squares cost model evaluated near the cost minimum, the Gauss-Newton approximation is usually very accurate

Levenberg - Marquardt Algorithm

- LMA interpolates between the Gauss Newton algorithm (GNA) and gradient descent

- When it is far from the minimum it acts as a steepest descent and it performs Gauss Newton iterations when it is near the solution

$$(H + \lambda W)\, \delta x \ = \ -g$$

$$\lambda \gg 1 \Rightarrow Gradient\ Descent\ Method$$
$$\lambda < 1 \Rightarrow Gauss - Newton\ Method$$

General Implementation Issues

- Exploit the problem structure

  – See reduced camera matrices below

- Use factorization effectively

- Use stable local parameterizations

- Scaling and preconditioning

SfM in large areas, limited image overlap   Highly overlapping images in small area

Additional Material and Software

- Open Source Structure-from-Motion tutorial at CVPR 2015
  - http://www.kitware.com/cvpr2015-tutorial.html
- Advanced notes on bundle adjustment
- Tutorials on several popular open source SfM packages

"Visual Odometry for Ground Vehicle Applications" by David Nister, Oleg Naroditsky and James Bergen (2006)

Visual Odometry

- Focus on high accuracy and real-time performance

- One important conclusion: stereo cameras are necessary to avoid drift and enable long term deployment

VO Steps

- Detect Harris corners in each frame
- Do not track them using KLT, but extract in each frame separately
  - Use NCC in 11x11 windows

- Do bundle adjustment in sliding window mode often

Monocular Pipeline

1. Track features over a certain number of frames. Estimate the relative poses between three of the frames using the 5-point algorithm and preemptive RANSAC followed by iterative refinement.

2. Triangulate the observed feature tracks into 3D points using the first and last observation on each track and optimal triangulation according to directional error. If this is not the first time through the loop, estimate the scale factor between the present reconstruction and the previous camera trajectory with another preemptive RANSAC procedure. Put the present reconstruction in the coordinate system of the previous one.

3. Track for a certain additional number of frames. Compute the pose of the camera with respect to the known 3D points using the 3-point algorithm and preemptive RANSAC followed by iterative refinement.

4. Re-triangulate the 3D points using the first and last observations on their image track. Repeat from Step 3 a certain number of times.

5. Repeat from Step 1 a certain number of times.

6. Insert a firewall and repeat from Step 1.
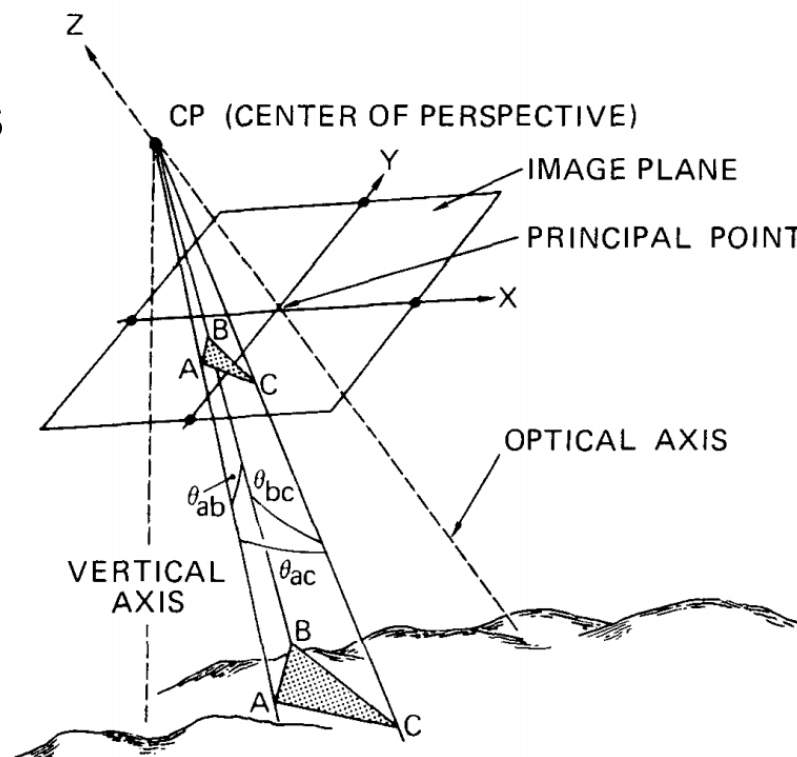
The 5-point Algorithm

- Computes relative camera pose given a minimal number of 5 correspondences
  - Up to 10 real solutions
- Intrinsics must be known
- Code available at http://vis.uky.edu/~stewe/FIVEPOINT/
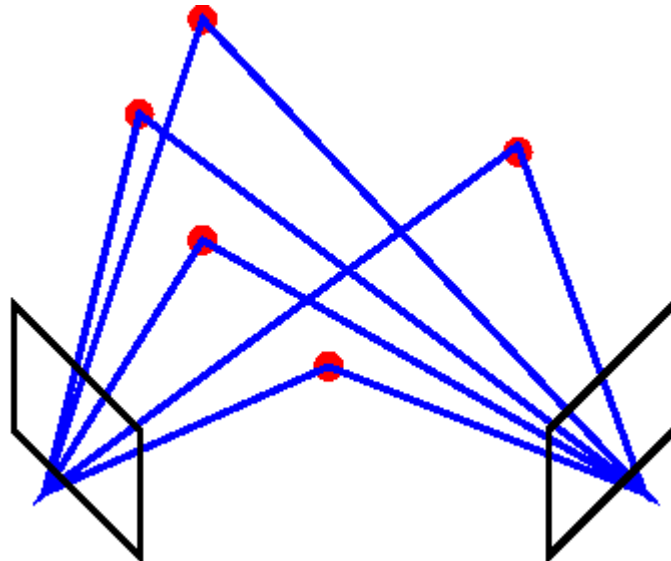
The 3-point Algorithm

- A.k.a. the Perspective 3 Point problem (Haralick et al. 2004)
- Estimate camera pose given images of three known 3D points
 - Up to 4 real solutions

Stereo Pipeline

1. Match feature points between the left and right images of the stereo pair. Triangulate the observed matches into 3D points.

2. Track features for a certain number of frames. Compute the pose of the stereo rig with preemptive RANSAC followed by iterative refinement. The 3-point algorithm (considering the left image) is used as the hypothesis generator. The scoring and iterative refinement are based on reprojection errors in both frames of the stereo pair.

3. Repeat from Step 2 a certain number of times.

4. Triangulate all new feature matches using the observations in the left and right images. Repeat from Step 2 a certain number of times.

5. Re-triangulate all 3D points to set up a firewall. Repeat from Step 2.

Advantages of Stereo System

- Fixed, known baseline sets and maintains global scale
- Avoids the difficult relative orientation step
  - Instead, performs triangulation followed by pose estimation

Results

- Remote controlled run in a parking lot. DGPS - Dark Blue.

- Wheel encoders fused with gyro- Medium Red.

- Visual odometry fused with gyro - Light Green

Parking Lot 1
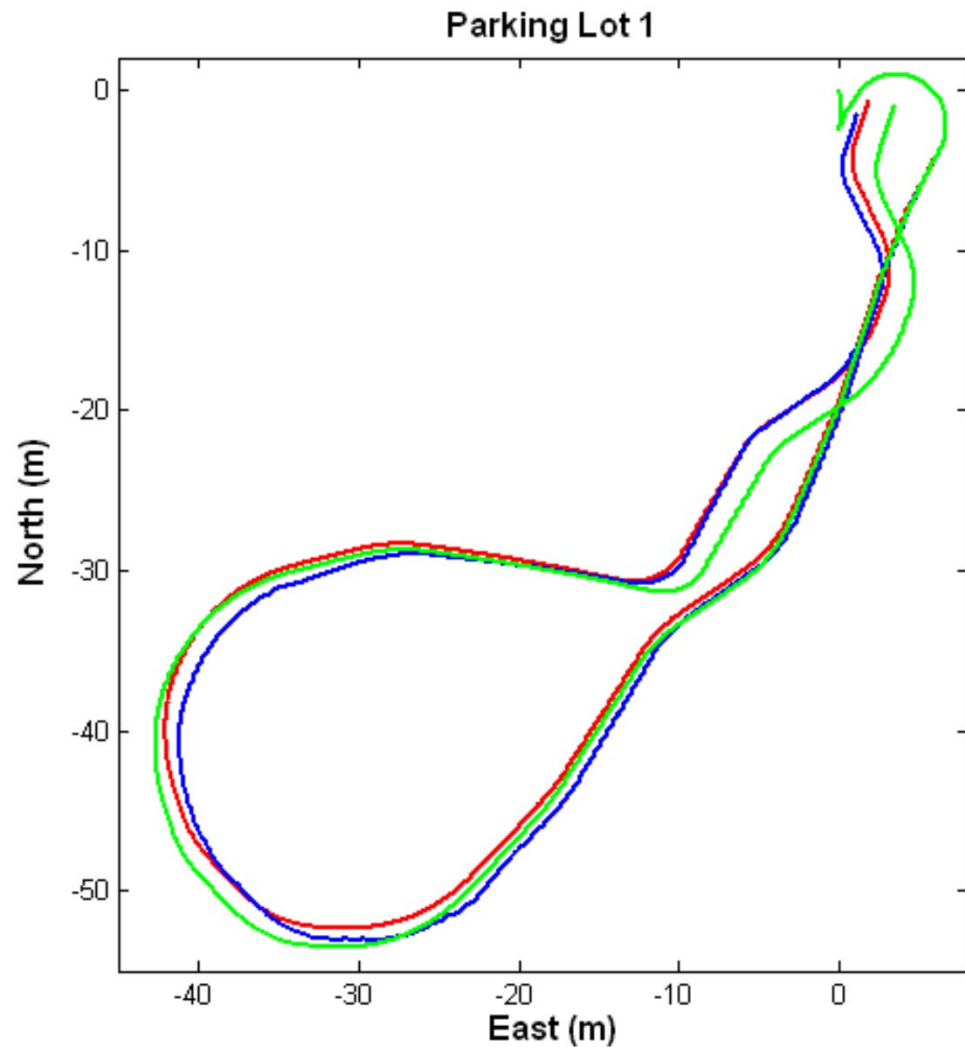
Parallel Tracking and Mapping (PTAM)

- Probably most reliable solution for limited spaces (Klein and Murray 2007)
  - http://www.robots.ox.ac.uk/~gk/PTAM/

- Tracking and Mapping are separated, and run in two parallel threads
- Mapping is based on keyframes, which are processed using batch techniques (Bundle Adjustment)
- The map is densely initialized from a stereo pair (5-Point Algorithm)
- New points are initialized with an epipolar search
- Large numbers (thousands) of points are mapped

The Map

- The map consists of a collection of M point features located in a world coordinate frame W
  - Each point feature represents a locally planar textured patch in the world
  - Each point also has a unit patch normal and a reference to the patch source pixels
- The map also contains N keyframes: these are snapshots taken by the handheld camera at various points in time
  - Each keyframe has an associated camera-centered coordinate frame
 - Each keyframe also stores a four level pyramid of greyscale images

Tracking

1. A new frame is acquired from the camera, and a prior pose estimate is generated from a motion model
2. Map points are projected into the image according to the frame's prior pose estimate
3. A small number (50) of the coarsest-scale features are searched for in the image
4. The camera pose is updated from these coarse matches
5. A larger number (1000) of points is re-projected and searched for in the image.
6. A final pose estimate for the frame is computed from all the matches found

Mapping
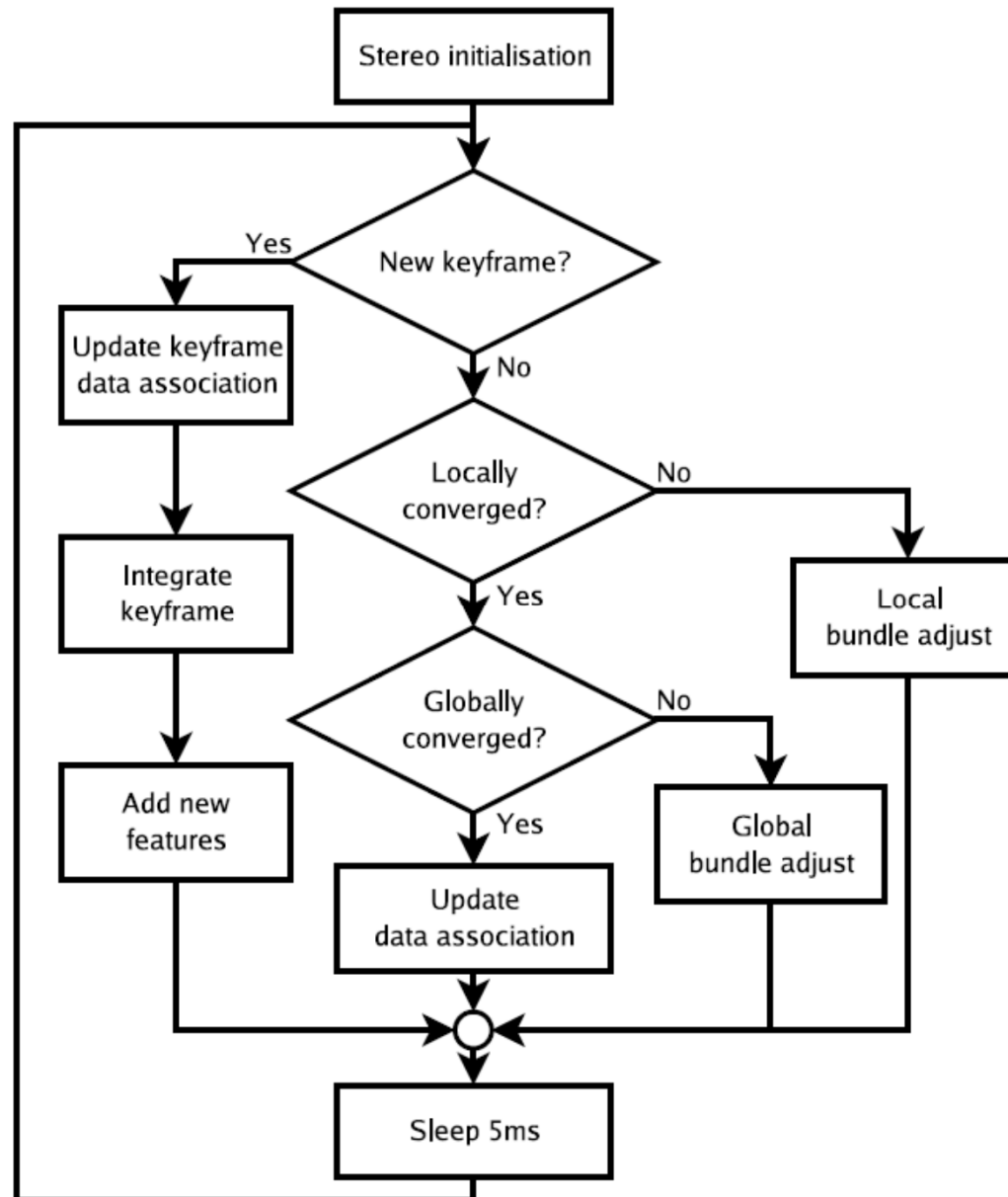
Results
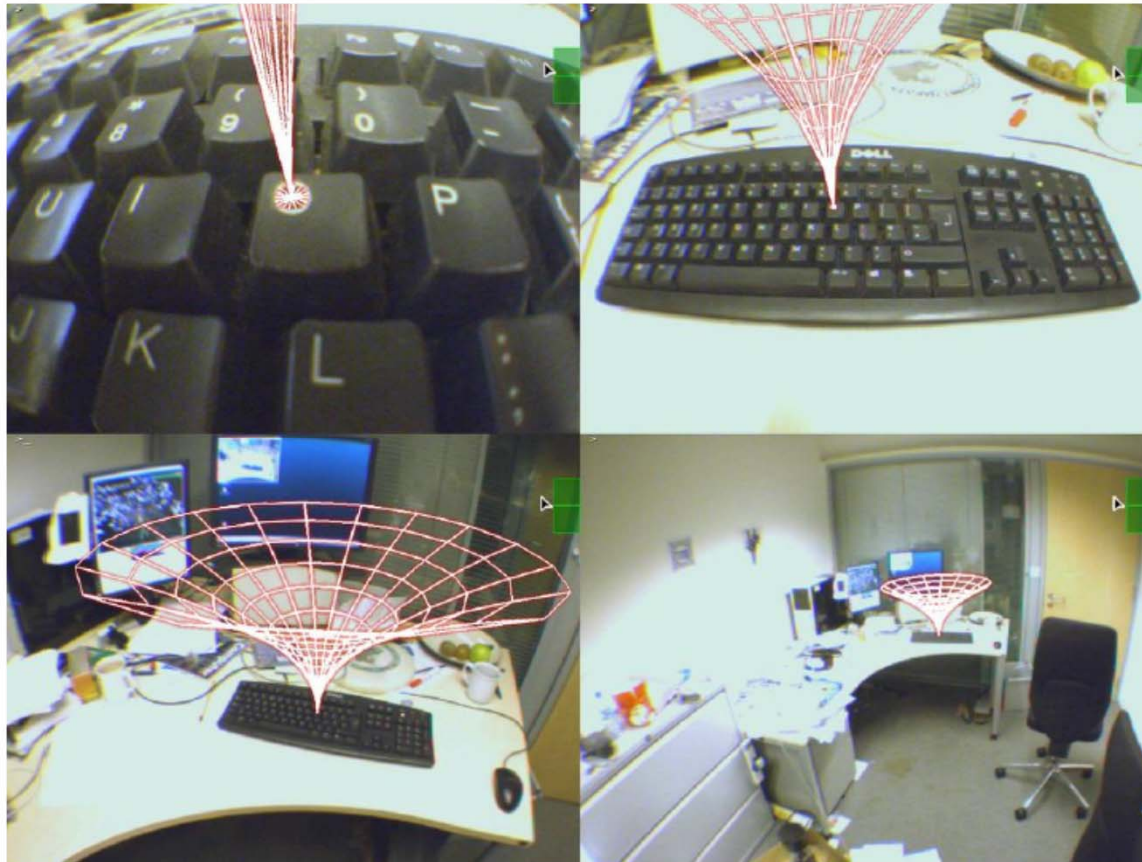
- PTAM can easily track across multiple scales
- Here, the map is initialized at the top-right scale
- The user moves closer in and places a label, which is still accurately registered when viewed from far away

Visual SLAM: Why Filter?

- H. Strasdat, J.M.M. Montiel, A.J. Davison (2012)

- The most accurate solution to off-line Structure from Motion (SFM): extract as much correspondence information as possible and perform batch optimization

- Sequential methods for live video streams must approximate this to fit within fixed computational bounds.

- Two quite different approaches that sparsify the problem in different ways:
  - Filtering methods marginalize out past poses and summarize the information gained over time with a probability distribution
  - Keyframe methods retain the optimization approach of global bundle adjustment, but select only a small number of past frames to process

Visual SLAM: Why Filter?

Filter   Keyframe BA
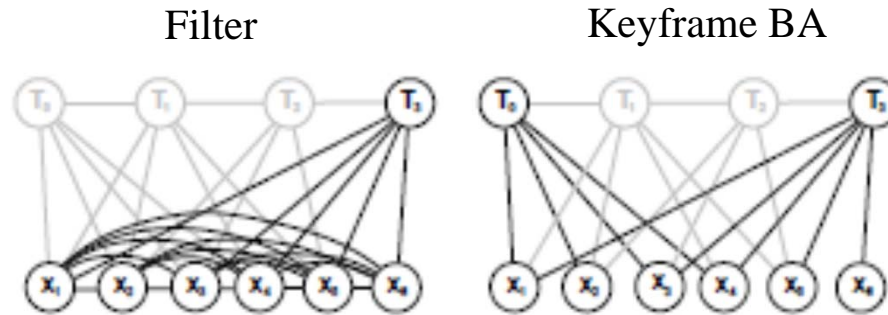
- ## Filtering:
  - All poses other than the current one are marginalized out after every frame
  - Features, which may be measured again in the future, are retained
  - The graph quickly becomes fully inter-connected, since every elimination of a past pose variable causes fill-in with new links between every pair of feature variables to which it was joined – poor scalability

- ## Bundle adjustment
  - Solve the graph from scratch time after time as is grows, but sparsify it by removing all but a small subset of past poses
  - The other poses, and all the measurements connected to them, are not marginalized out as in the filter, but simply discarded
  - The graph has more elements but it remains sparsely inter-connected
  - The ability to incorporate more feature measurements counters the information lost from the discarded frames

Visual SLAM: Why Filter?

- Key question: does it make sense to summarize the information gained from historic poses and measurements by joint probability distributions in state space and propagate these through time (filtering), or to discard some of those measurements in such a way that repeated optimization from scratch becomes feasible (keyframe BA)?

- In analysis considering both monocular and stereo SLAM on various different scenes and motion patterns: keyframe bundle adjustment outperforms filtering
  - It gives the highest accuracy per unit of computing time