Robert Collins
CSE486, Penn State

Lecture 25:

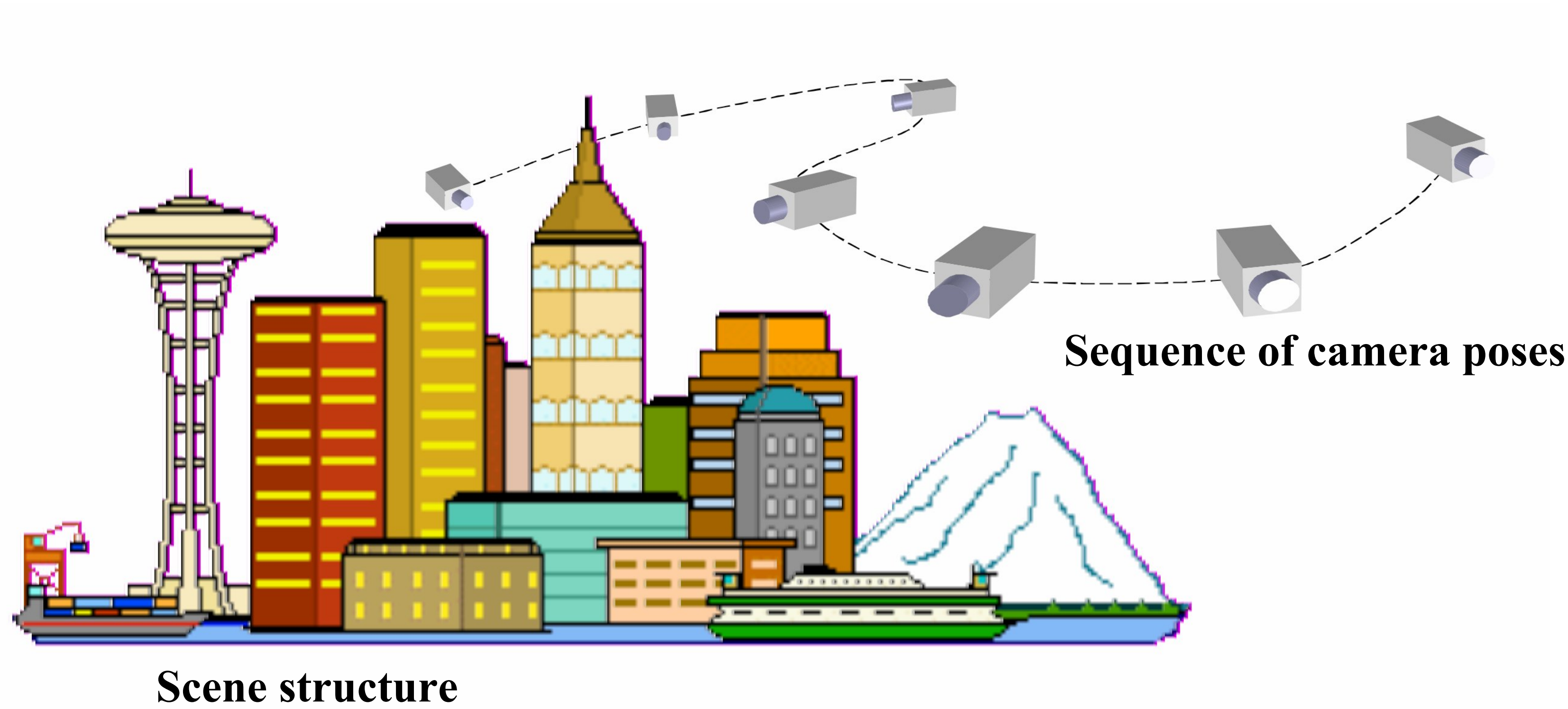
Structure from Motion

Robert Collins
CSE486, Penn State

Structure from Motion

Given a set of flow fields or displacement vectors from a moving camera over time, determine:

- the sequence of camera poses
- the 3D structure of the scene



SFM "Killer App"

Match Move

Track a set of feature points through a movie sequence

Deduce where the cameras are and the 3D locations of the points that were tracked

Render synthetic objects with respect to the deduced 3D geometry of the scene / cameras

Match Move Examples

“Harts’ War” and “Graham Kimpton” examples from www.realviz.com MatchMover Professional gallery. Copyrighted.

Factorization

Tomasi and Kanade, “Shape and Motion from Image Streams under Orthography,” *International Journal of Computer Vision (IJCV),* Vol 9, pp.137-154, 1992.

Goal: combine point correspondence information from multiple points over multiple frames to solve for scene structure and camera motion (structure from motion)

Approach: numerically stable approach based on using SVD to “factor” matrix of observed point positions.

Historical significance: until that time, most SFM work dealt with minimal configurations, and noise-free data. Factorization was one of the first “practical SFM algorithms”

Recall : World to Camera Transform

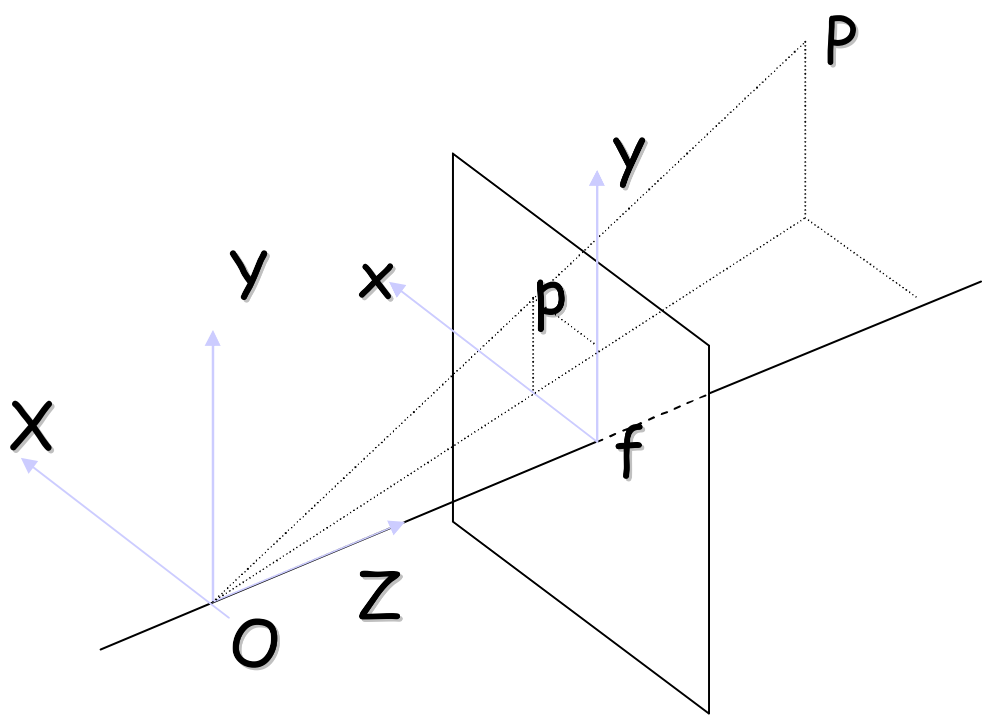
$$\mathbf{P^C = R\ (\ P^W - C\ )}$$

$$\begin{pmatrix} P^C_x \\ P^C_y \\ P^C_z \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P^W_x \\ P^W_y \\ P^W_z \\ 1 \end{pmatrix}$$

$$P^C = M_{ext} \bullet P^W$$

Robert Collins
CSE486, Penn State

Perspective Projection



$$x = f\,\frac{X}{Z}$$

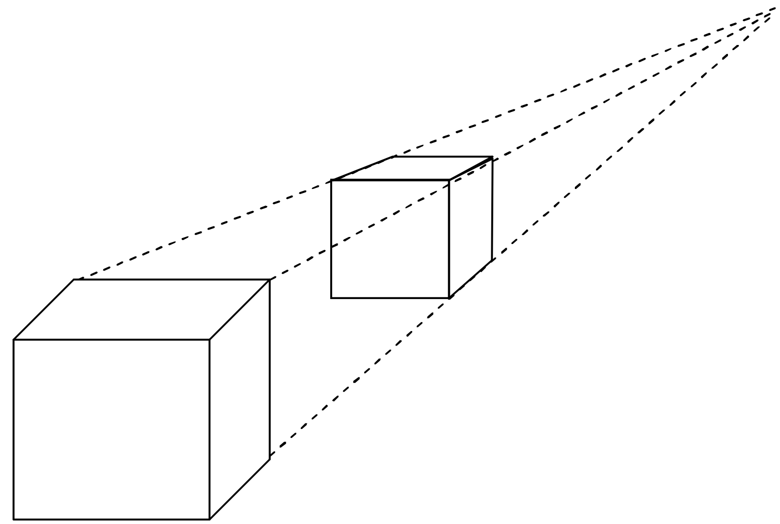
$$y = f\,\frac{Y}{Z}$$

- Non-linear equations
- Any point on the ray OP has image p !!

Perspective Projection

$$x = f\,\frac{X}{Z}$$

$$y = f\,\frac{Y}{Z}$$

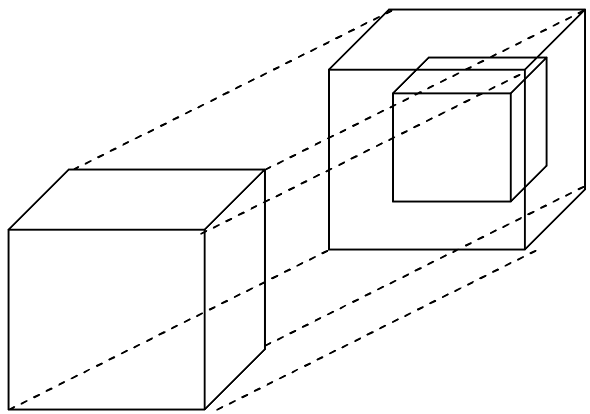


Perspective Projection : parallel lines appear to meet at a vanishing point; farther objects seem smaller

Simplification: Weak Perspective

$$x = \frac{f}{Z_o} X$$

$$y = \frac{f}{Z_o} Y$$

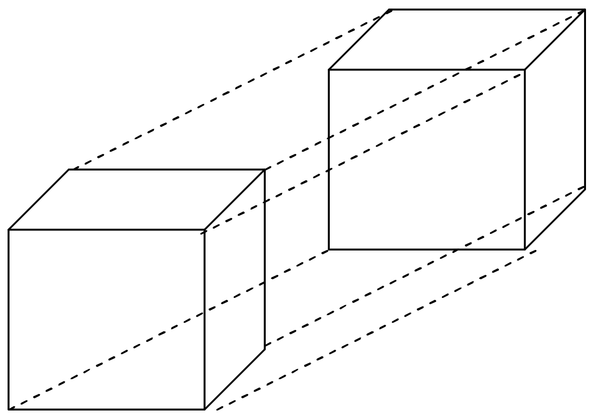


Weak perspective = Parallel projection (parallel lines remain parallel) + Scaling to simulate change in size due to object distance.

Simpler: Orthographic Projection

$$x = X$$

$$y = Y$$



Pure parallel projection. Highly simplified case where we even ignore the scaling due to distance.

Perspective Matrix Equation

(Camera Coordinates)

$$x = f\frac{X}{Z}$$

$$y = f\frac{Y}{Z}$$

Using homogeneous coordinates:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x = \frac{x'}{z'} \qquad y = \frac{y'}{z'}$$

Weak Perspective Approximation

$$x = \frac{f}{Z_o} X$$

$$y = \frac{f}{Z_o} Y$$

Using homogeneous coordinates:

$$\begin{bmatrix} x \\ y \\ \\ \end{bmatrix} = \begin{bmatrix} f/Z_0 & 0 & 0 & 0 \\ 0 & f/Z_0 & 0 & 0 \\ & & & \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Let's Consider Orthographic

$$x = X$$

$$y = Y$$

Using homogeneous coordinates:

$$\begin{bmatrix} x \\ y \\ \\ \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \\ \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Combine with External Params

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x^W \\ P_y^W \\ P_z^W \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x^W \\ P_y^W \\ P_z^W \\ 1 \end{pmatrix}$$

Combine with External Params

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cancel{0} \\ r_{21} & r_{22} & r_{23} & \cancel{0} \end{bmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{1} \end{pmatrix} \begin{pmatrix} P_x^W \\ P_y^W \\ P_z^W \\ 1 \end{pmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{bmatrix} \begin{pmatrix} P_x^W - c_x \\ P_y^W - c_y \\ P_z^W - c_z \end{pmatrix}$$

Orthographic: Algebraic Equation

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{array}{c} \mathbf{i}^{T} \\ \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{bmatrix} \\ \mathbf{j}^{T} \end{array} \left(\overset{\mathbf{P}}{\begin{bmatrix} P^{W}_{x} \\ P^{W}_{y} \\ P^{W}_{z} \end{bmatrix}} - \overset{\mathbf{T}}{\begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix}} \right)$$

$$x = \mathbf{i}^{T} (\mathbf{P} - \mathbf{T})$$

$$y = \mathbf{j}^{T} (\mathbf{P} - \mathbf{T})$$

Multiple Points, Multiple Frames

Notation (attack of the killer subscripts)

$$x = \mathbf{i}^T(\mathbf{P} - \mathbf{T})$$
$$y = \mathbf{j}^T(\mathbf{P} - \mathbf{T})$$

N points

$\mathbf{P}_1 \; \mathbf{P}_2 \; \ldots \; \mathbf{P}_j \; \ldots \; \mathbf{P}_N$

F frames

$\mathbf{i}_1 \; \mathbf{i}_2 \; \ldots \; \mathbf{i}_i \; \ldots \; \mathbf{i}_F$

$\mathbf{j}_1 \; \mathbf{j}_2 \; \ldots \; \mathbf{j}_i \; \ldots \; \mathbf{j}_F$

$\mathbf{T}_1 \; \mathbf{T}_2 \ldots \mathbf{T}_i \ldots \mathbf{T}_F$

$$x_{ij} = \mathbf{i}_i^T(\mathbf{P}_j - \mathbf{T}_i)$$
$$y_{ij} = \mathbf{j}_i^T(\mathbf{P}_j - \mathbf{T}_i)$$

Eq 8.31-8.32
T&V book

Robert Collins
CSE486, Penn State

Factorization Approach

$$x_{ij} = \mathbf{i}_i^T (\mathbf{P}_j - \mathbf{T}_i)$$

$$y_{ij} = \mathbf{j}_i^T (\mathbf{P}_j - \mathbf{T}_i)$$

N points

$\mathbf{P}_1 \ \mathbf{P}_2 \ \ldots \ \mathbf{P}_j \ \ldots \ \mathbf{P}_N$

(We want to recover these)

Note that absolute position of the set of points is something that cannot be uniquely recovered, so…

First Trick: set the origin of the world coordinate system to be the center of pass of the N points!

$$\frac{1}{N}\sum_{i=1}^{N} P_i = 0$$

Factorization Approach

$$x_{ij} = \mathbf{i}_i^{\mathbf{T}} (\mathbf{P}_j - \mathbf{T}_i)$$
$$y_{ij} = \mathbf{j}_i^{\mathbf{T}} (\mathbf{P}_j - \mathbf{T}_i)$$

Centroid at 0:

$$\frac{1}{N}\sum_{i=1}^{N} P_i = 0$$

Implication:

$$\bar{x}_{it} = \frac{1}{N}\sum_{i=1}^{N} i_t^T (P_i - T_t) = \frac{1}{N}\sum_{i=1}^{N} i_t^T P_i - \frac{1}{N}\sum_{i=1}^{N} i_t^T T_t = 0 - i_t^T T_t$$

Note: this is the center of mass of x coordinates in frame t

Robert Collins
CSE486, Penn State

Factorization Approach

$$\bar{x}_{ti} = \frac{1}{n}\sum_{i=1}^{n} i_t^T (P_i - T_t) = -i_t^T T_t$$

$$\bar{y}_{ti} = \frac{1}{n}\sum_{i=1}^{n} j_t^T (P_i - T_t) = -j_t^T T_t$$

Second Trick: subtract off the center of mass of the 2D points in each frame. (Centering)

$$x_{ij} = \mathbf{i}_i^{\mathbf{T}} (\mathbf{P}_j - \mathbf{T}_i)$$

$$y_{ij} = \mathbf{j}_i^{\mathbf{T}} (\mathbf{P}_j - \mathbf{T}_i)$$

$$\tilde{x}_{ti} = x_i - \bar{x}_{ti} = i_t^T P_i$$

$$\tilde{y}_{ti} = y_i - \bar{y}_{ti} = j_t^T P_i$$

Robert Collins
CSE486, Penn State

Factorization Approach

$$x_{ij} = \mathbf{i}_i^T (\mathbf{P}_j - \cancel{\mathbf{T}_i})$$

$$y_{ij} = \mathbf{j}_i^T (\mathbf{P}_j - \cancel{\mathbf{T}_i})$$

centering $\Rightarrow$

$$\tilde{x}_{ti} = x_i - \bar{x}_{ti} = i_t^T P_i$$

$$\tilde{y}_{ti} = y_i - \bar{y}_{ti} = j_t^T P_i$$

What have we accomplished so far?

1) Removed unknown camera locations from equations.

2) More importantly, we can now write everything As a big matrix equation…

Robert Collins
CSE486, Penn State

Factorization Approach

Form a matrix of centered image points.

2FxN

All N points in one frame

$$
\begin{pmatrix}
\tilde{x}_{11} & \tilde{x}_{12} & \tilde{x}_{13} & \ldots & \tilde{x}_{1N} \\
 & & \vdots & & \\
\tilde{x}_{F1} & \tilde{x}_{F2} & \tilde{x}_{F3} & \ldots & \tilde{x}_{FN} \\
\tilde{y}_{11} & \tilde{y}_{12} & \tilde{y}_{13} & \ldots & \tilde{y}_{1N} \\
 & & \vdots & & \\
\tilde{y}_{F1} & \tilde{y}_{F2} & \tilde{y}_{F3} & \ldots & \tilde{y}_{FN}
\end{pmatrix}
$$

Factorization Approach

Form a matrix of centered image points.

2FxN

$$
\begin{pmatrix}
\tilde{x}_{11} & \tilde{x}_{12} & \tilde{x}_{13} & \ldots & \tilde{x}_{1N} \\
 & & \vdots & & \\
\tilde{x}_{F1} & \tilde{x}_{F2} & \tilde{x}_{F3} & \ldots & \tilde{x}_{FN} \\
\tilde{y}_{11} & \tilde{y}_{12} & \tilde{y}_{13} & \ldots & \tilde{y}_{1N} \\
 & & \vdots & & \\
\tilde{y}_{F1} & \tilde{y}_{F2} & \tilde{y}_{F3} & \ldots & \tilde{y}_{FN}
\end{pmatrix}
$$

Tracking one point through all F frames

Factorization Approach

$$\tilde{x}_{it} = x_i - \bar{x}_{it} = i_t^T P_i$$

$$\tilde{y}_{it} = y_i - \bar{y}_{it} = j_t^T P_i$$

matrix of centered image points:

$$\begin{array}{ccccc} \textbf{2FxN} & & \textbf{2Fx3} & & \textbf{3xN} \\ \begin{pmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \tilde{x}_{13} & \dots & \tilde{x}_{1N} \\ & & \vdots & & \\ \tilde{x}_{F1} & & \mathbf{W} & . & \tilde{x}_{FN} \\ \tilde{y}_{11} & \tilde{y}_{12} & \tilde{y}_{13} & \dots & \tilde{y}_{1N} \\ & & \vdots & & \\ \tilde{y}_{F1} & \tilde{y}_{F2} & \tilde{y}_{F3} & \dots & \tilde{y}_{FN} \end{pmatrix} & = & \begin{pmatrix} i_1^T \\ \vdots \\ \mathbf{M} \\ j_1^T \\ \vdots \\ j_F^T \end{pmatrix} & & \begin{bmatrix} P_1 & \mathbf{S} & P_N \end{bmatrix} \end{array}$$

Factorization Approach

$$\underset{2F \times N}{\mathbf{W}} = \underset{2F \times 3}{\mathbf{M}} \; \underset{3 \times N}{\mathbf{S}}$$

Centered measurement matrix

“Motion” (camera rotation)

Structure (3D scene points)

Factorization Approach

$$\underset{2F \times N}{\mathbf{W}} = \underset{2F \times 3}{\mathbf{M}} \; \underset{3 \times N}{\mathbf{S}}$$

Rank Theorem:

The 2FxN centered observation matrix has at most rank 3.

Proof:

Trivial, using the properties:

- rank of mxn matrix is at most min(m,n)
- rank of A*B is at most min(rank(A),rank(B))

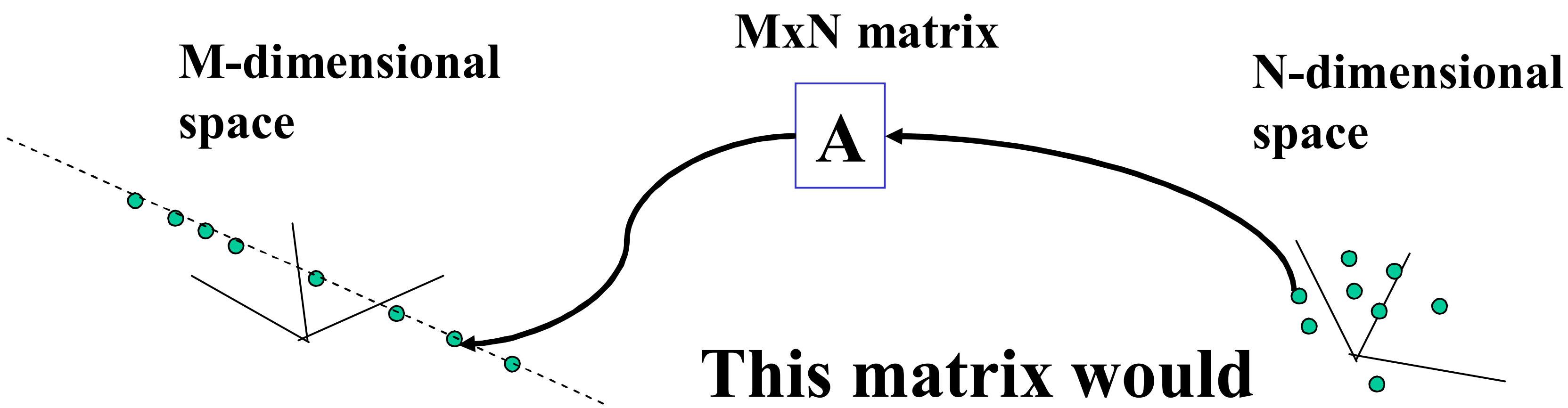
Robert Collins
CSE486, Penn State

Rank of a Matrix

What is rank of a matrix, anyways?

Number of columns (rows) that are linearly independent.

If matrix A is treated as a linear map, it is the <u>intrinsic</u> dimension of the space that is mapped into.



This matrix would have rank 1

Factorization Rank Theorem

Importance of rank theorem:

- Shows that video data is highly redundant
- Precisely quantifies the redundancy
- Suggests an algorithm for solving SFM

Factorization Approach

Form SVD of measurement matrix W

$$\underset{2F\times N}{\mathbf{W}} = \underset{2F\times 2F}{\mathbf{U}} \; \underset{2F\times N}{\mathbf{D}} \; \underset{N\times N}{\mathbf{V}^T}$$

Diagonal matrix with eigenvalues sorted in decreasing order:

$d_{11} >= d_{22} >= d_{33} >= \ldots$

Factorization Approach

Form SVD of measurement matrix W

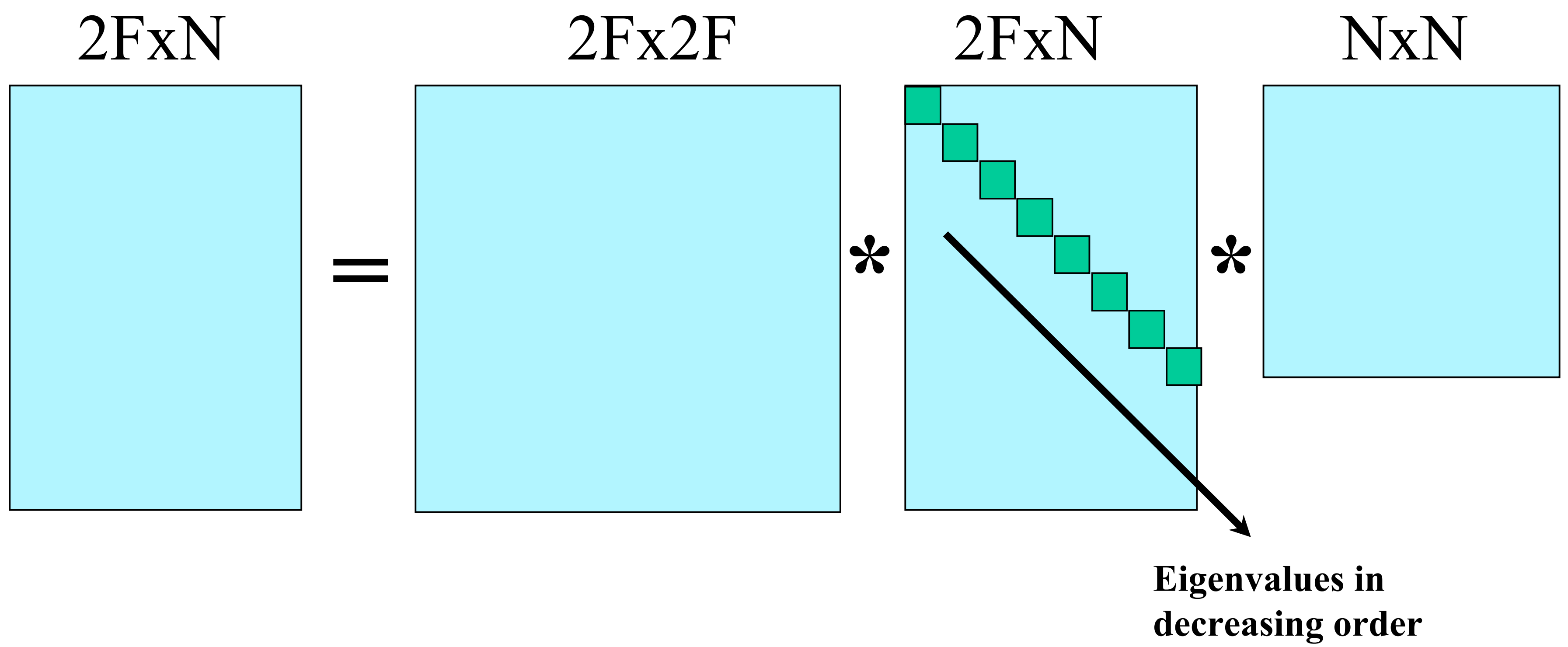
$$\underset{2F\times N}{\mathbf{W}} = \underset{2F\times 2F}{\mathbf{U}} \quad \underset{2F\times N}{\mathbf{D}} \quad \underset{N\times N}{\mathbf{V}^T}$$

Another useful rank property:

Rank of a matrix is equal to the number of nonzero eigenvalues.

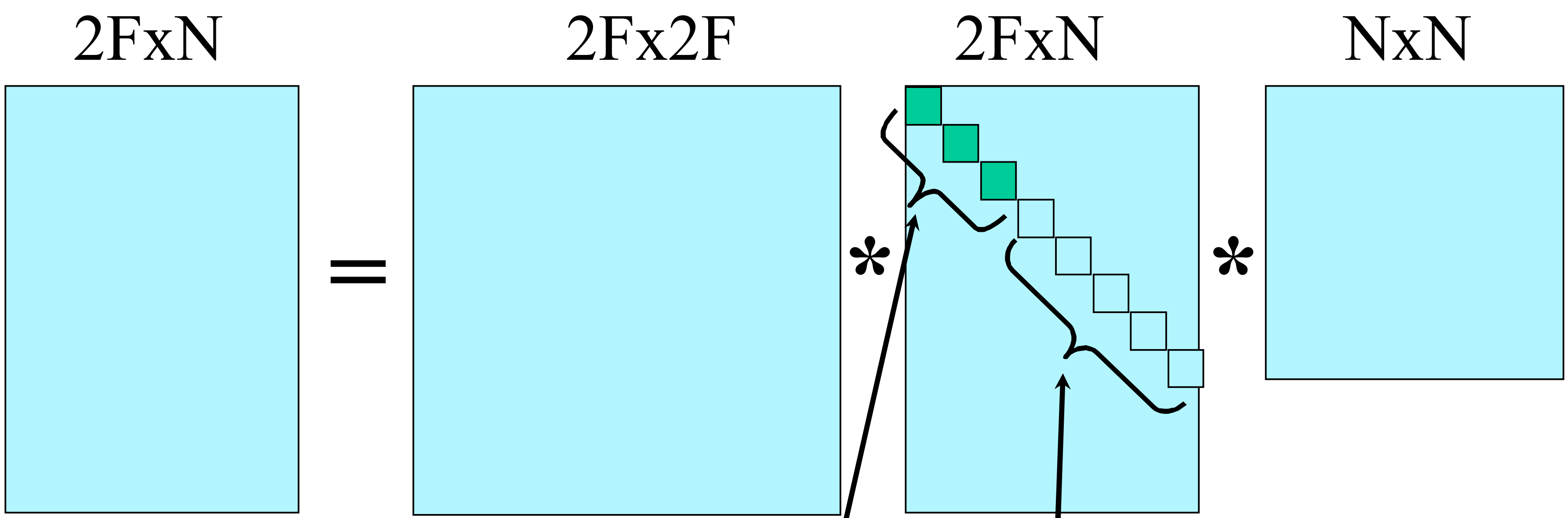
⇒ d_{11}, d_{22}, d_{33} are only nonzero eigenvalues (the rest are 0).

Factorization Approach



Robert Collins
CSE486, Penn State

Factorization Approach



Rank theorem says:

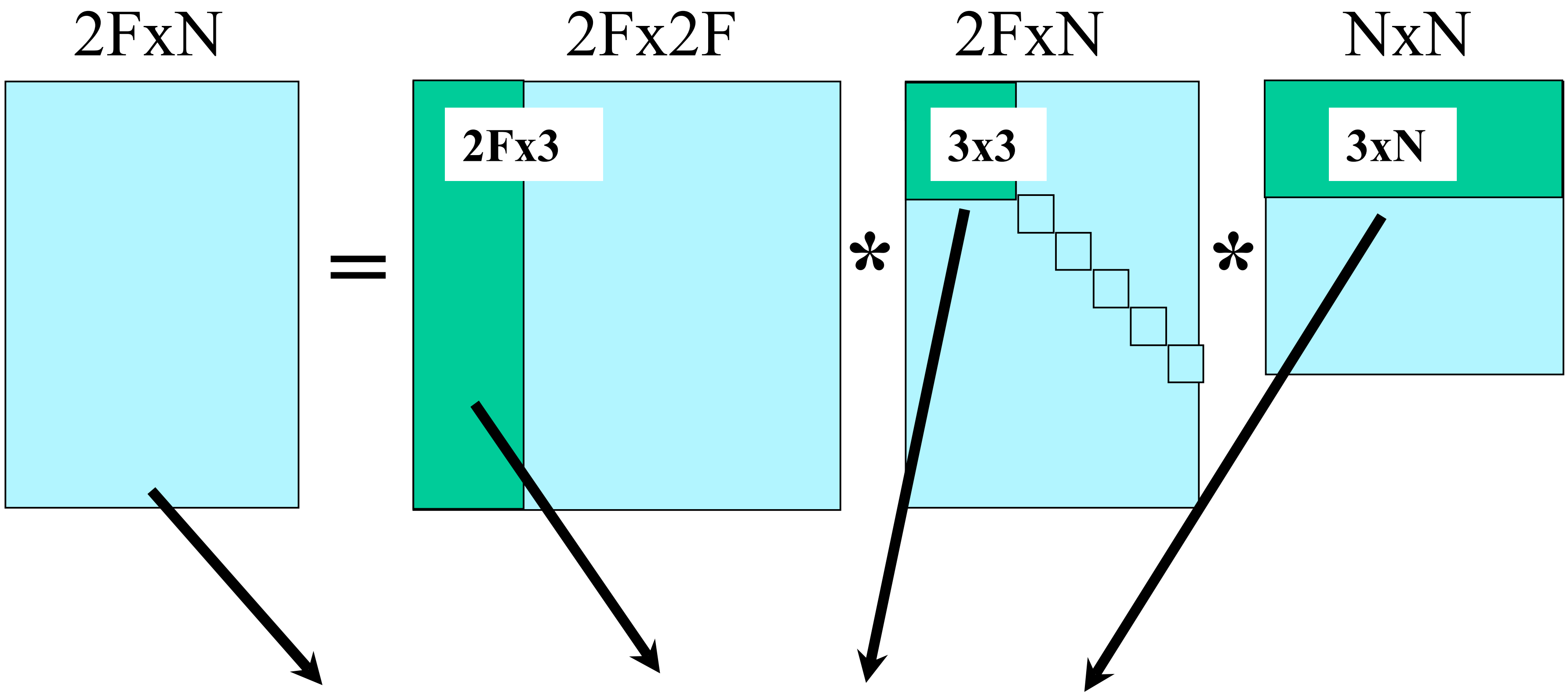
These should be zero

These 3 are nonzero

In practice, due to noise, there may be more than 3 nonzero eigenvalues, but rank theorem tells us to ignore all but the largest three.

Robert Collins
CSE486, Penn State

Factorization Approach



$$W = U' \; D' \; V'^T$$

Robert Collins
CSE486, Penn State

Factorization Approach

Observed image points

$$\mathbf{W} \overset{\text{SVD}}{=} \mathbf{U'}\ \mathbf{D'}\ \mathbf{V'}^{\mathbf{T}}$$

$$\underset{2F\text{x}N}{\mathbf{W}} = \underbrace{\mathbf{U'}\ \mathbf{D'}^{\mathbf{1/2}}}_{2F\text{x}3}\ \underbrace{\mathbf{D'}^{\mathbf{1/2}}\ \mathbf{V'}^{\mathbf{T}}}_{3\text{x}N}$$

$$\mathbf{W} = \mathbf{M}\ \mathbf{S}$$

Camera motion **Scene structure**

Annoying Details

$$\mathbf{W} = (\mathbf{U'} \ \mathbf{D'}^{1/2})(\mathbf{D'}^{1/2} \ \mathbf{V'}^T)$$

2FxN 2Fx3 3xN

$$\mathbf{W} = \mathbf{M} \ \mathbf{S}$$

Problems:

1) This is not a unique decomposition.

eg: $(\mathbf{M}\,\mathbf{Q})\,(\mathbf{Q}^{-1}\,\mathbf{S}) = \mathbf{M}\,\mathbf{Q}\,\mathbf{Q}^{-1}\,\mathbf{S} = \mathbf{M}\,\mathbf{S}$

2) i^T, j^T pairs (rows of M) are not necessarily orthogonal

Robert Collins
CSE486, Penn State

Solving the Annoying Details

Solution to both problems:

Solve for Q such that appropriate rows of M satisfy

$$\left.\begin{aligned} \hat{\mathbf{i}}_i^T Q Q^T \hat{\mathbf{i}}_i &= 1 \\ \hat{\mathbf{j}}_i^T Q Q^T \hat{\mathbf{j}}_i &= 1 \end{aligned}\right\} \text{unit vectors}$$

$$\hat{\mathbf{i}}_i^T Q Q^T \hat{\mathbf{j}}_i = 0 \qquad \text{orthogonal}$$

3N equations in 9 unknowns

But these are nonlinear equations

linearize and iterate

(see Exercise 8.8 in book for Newton's method)

(alternative approach is to use Cholesky decomposition – outside our scope)

Factorization Summary

Assumptions

- orthographic camera
- N non-coplanar points tracking in F>=3 frames

Form the centered measurement matrix $W=[\tilde{X} ; \tilde{Y}]$

- where $\tilde{x}_{ij} = x_{ij} - mx_j$
- where $\tilde{y}_{ij} = y_{ij} - my_j$
- mx_j and my_j are mean of points in frame i
- j ranges over set of points

Rank theorem: The centered measurement matrix has a rank of at most 3

Factorization Algorithm

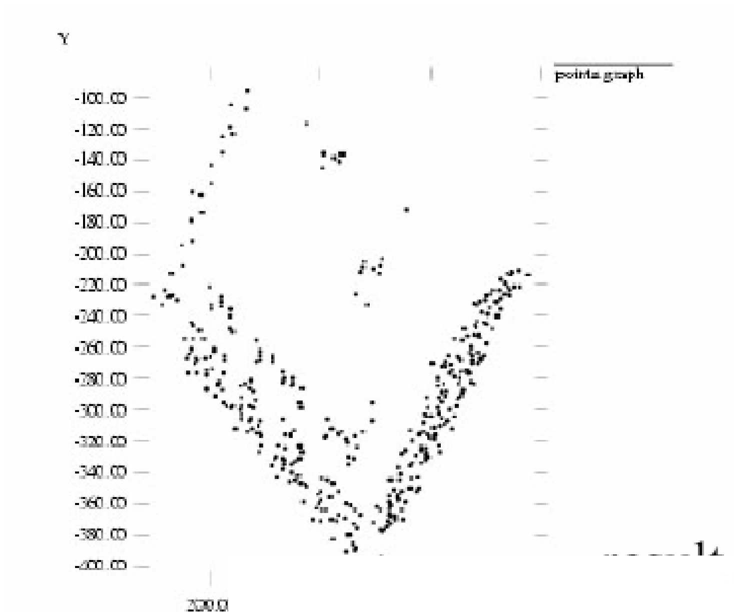
1) Form the centered measurement matrix W from N points tracked over F frames.
2) Compute SVD of $W = U\ D\ V^T$
   - U is 2Fx2F
   - D is 2FxN
   - V^T is NxN
3) Take largest 3 eigenvalues, and form
   - D' = 3x3 diagonal matrix of largest eigenvalues
   - U' = 2Fx3 matrix of corresponding column vectors from U
   - V'^T = 3xN matrix of corresponding row vectors from V^T
4) Define

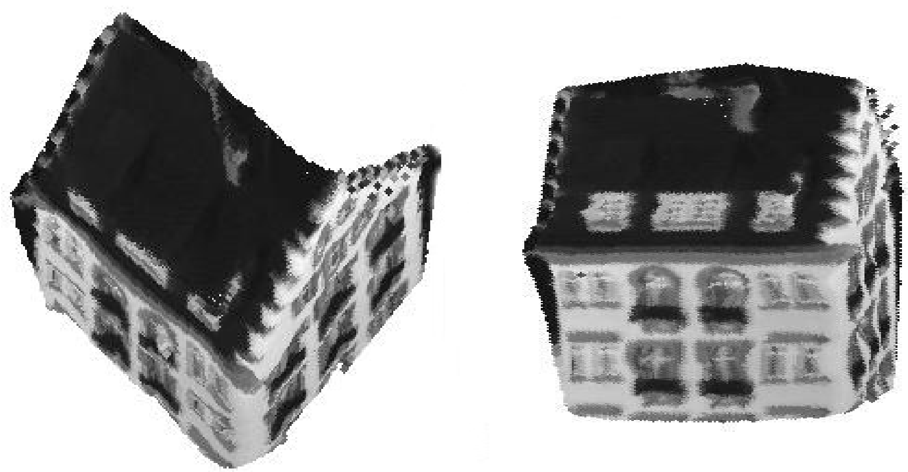
   $M = U'\ D'^{1/2}$ and $S = D'^{1/2}\ V'^T$
5) Solve for Q that makes appropriate rows of M orthogonal
6) Final solution is

   $M^* = M\ Q$ and $S^* = Q^{-1}\ S$

Sample Results

QuickTime™ and a
Cinepak decompressor
are needed to see this picture.





Sample Results

QuickTime™ and a
Cinepak decompressor
are needed to see this picture.