



MIT CSAIL

**6.869: Advances in Computer Vision**

**MIT**  
COMPUTER  
VISION

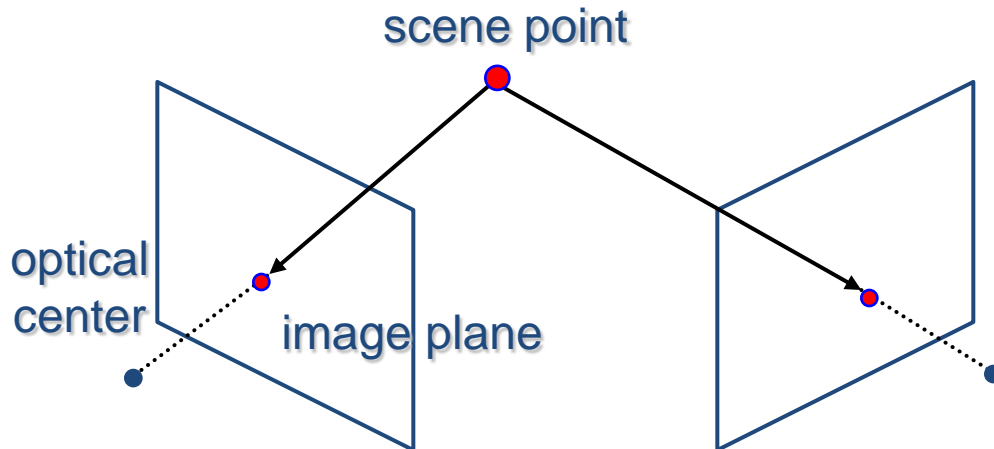
## Lecture 19

Structure from motion

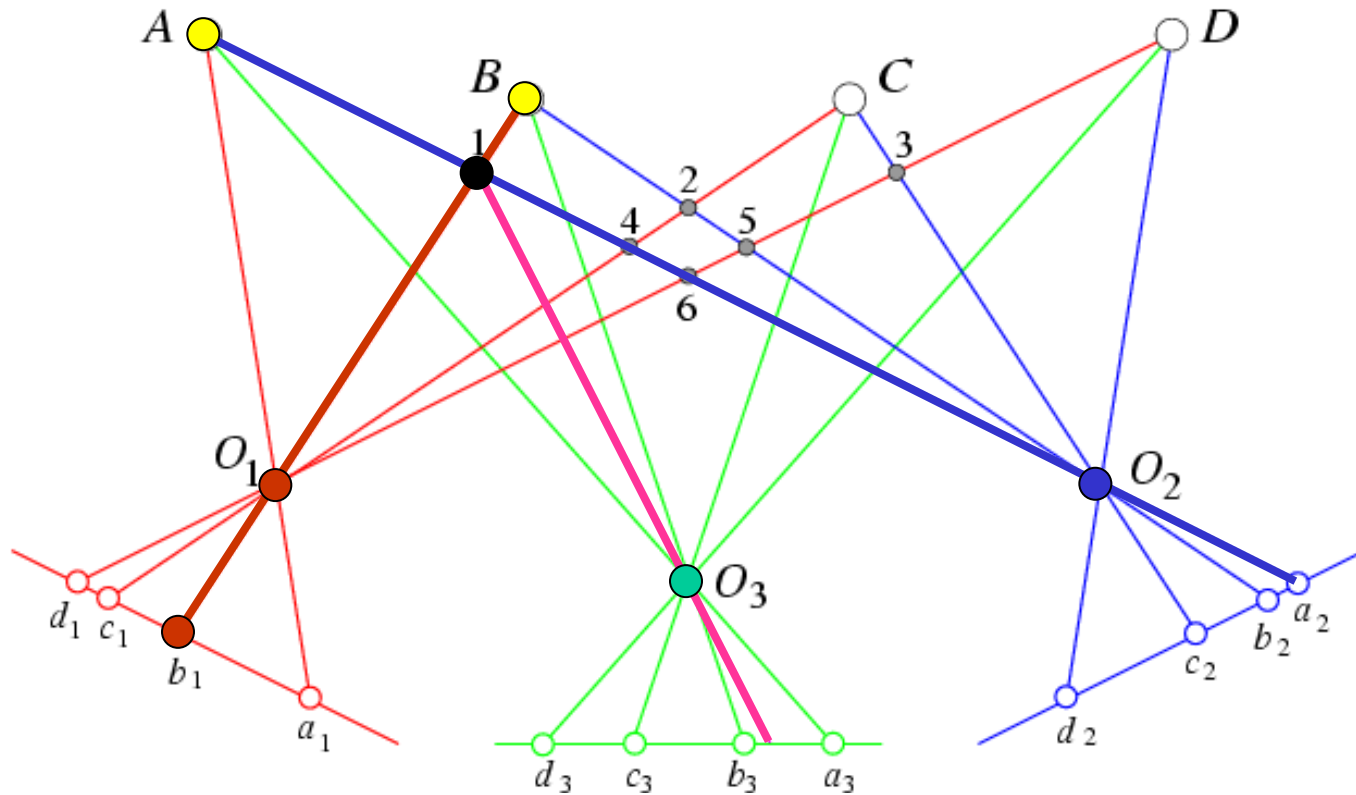
# Estimating depth with stereo



- **Stereo:** shape from disparities between two views
- We'll need to consider:
  - Info on camera pose ("calibration")
  - Image point correspondences



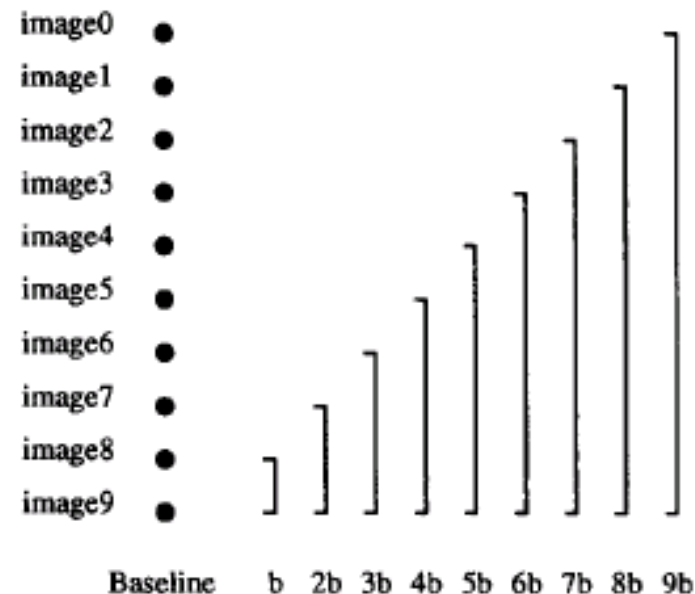
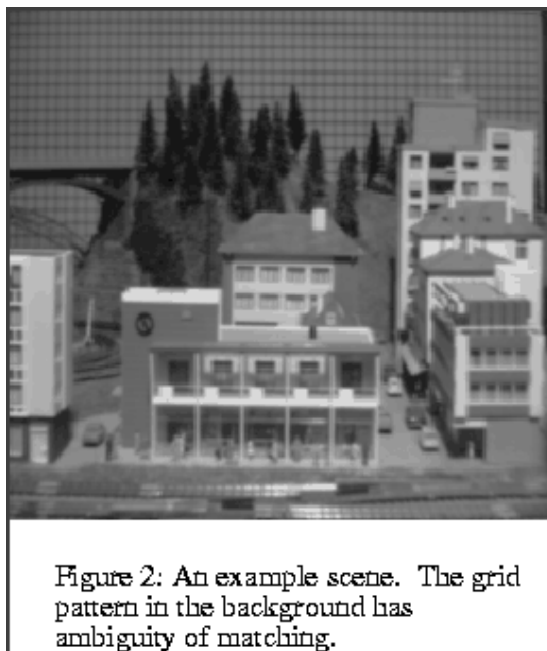
# Beyond two-view stereo



The third view can be used for verification

# Multiple-baseline stereo

- Pick a reference image, and slide the corresponding window along the corresponding epipolar lines of all other images, using **inverse depth** relative to the first image as the search parameter



M. Okutomi and T. Kanade, [“A Multiple-Baseline Stereo System,”](#) IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(4):353-363 (1993).

# Multiple-baseline stereo results

---



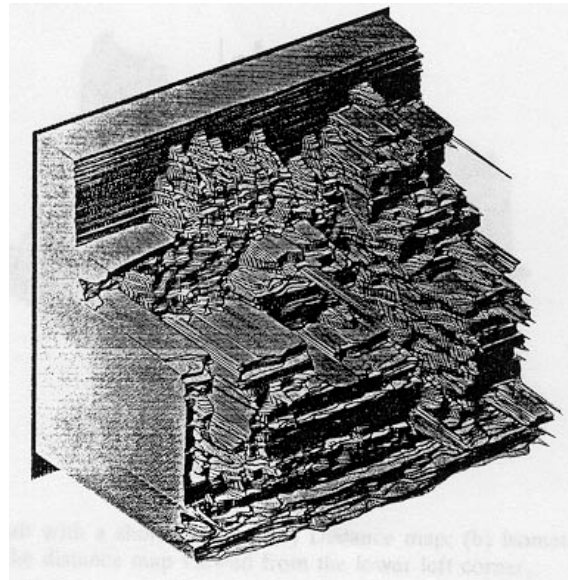
I1



I2



I10

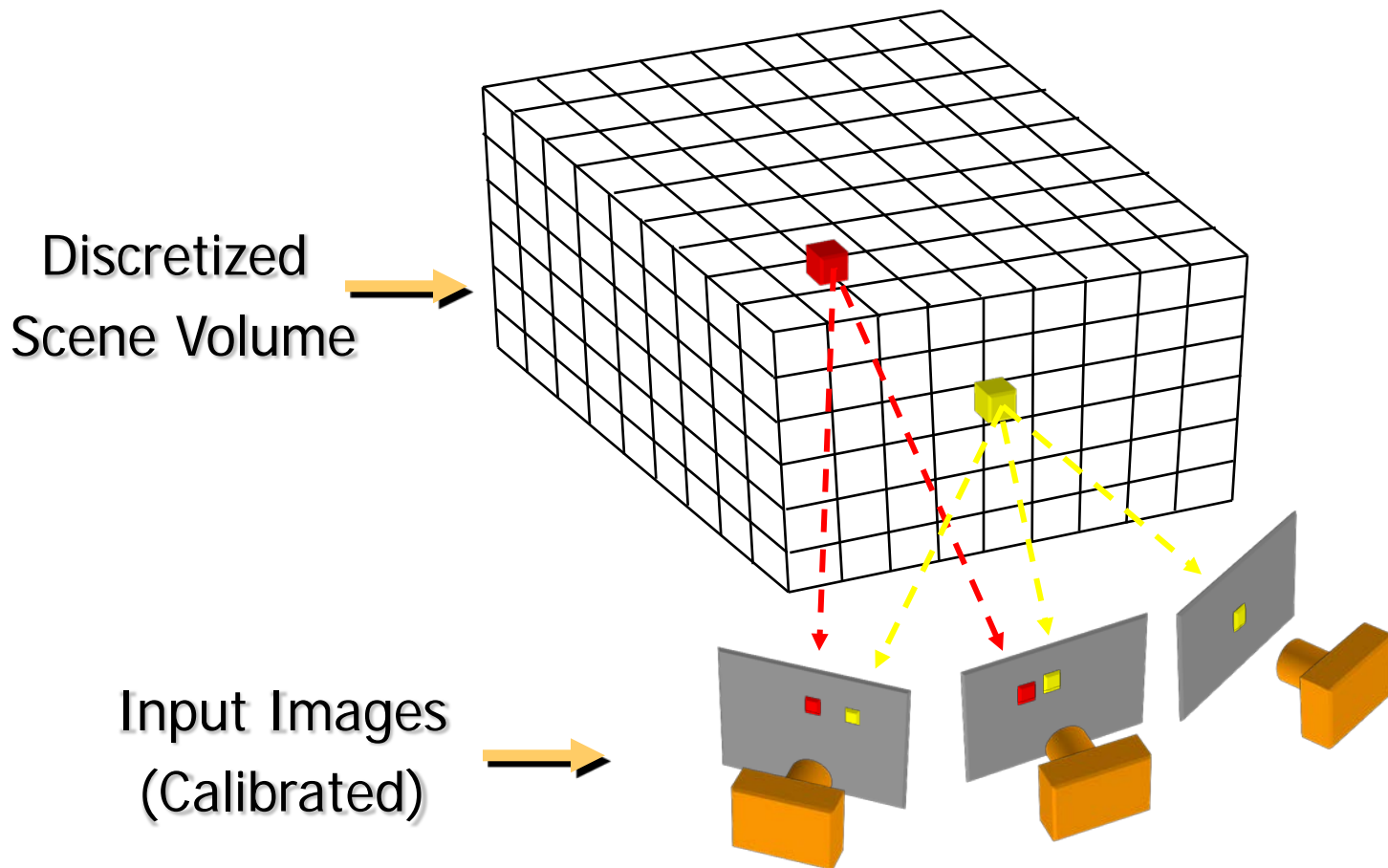


M. Okutomi and T. Kanade, [“A Multiple-Baseline Stereo System,”](#) IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(4):353-363 (1993).

Slide credit: S. Lazebnik

# Volumetric Stereo / Voxel Coloring

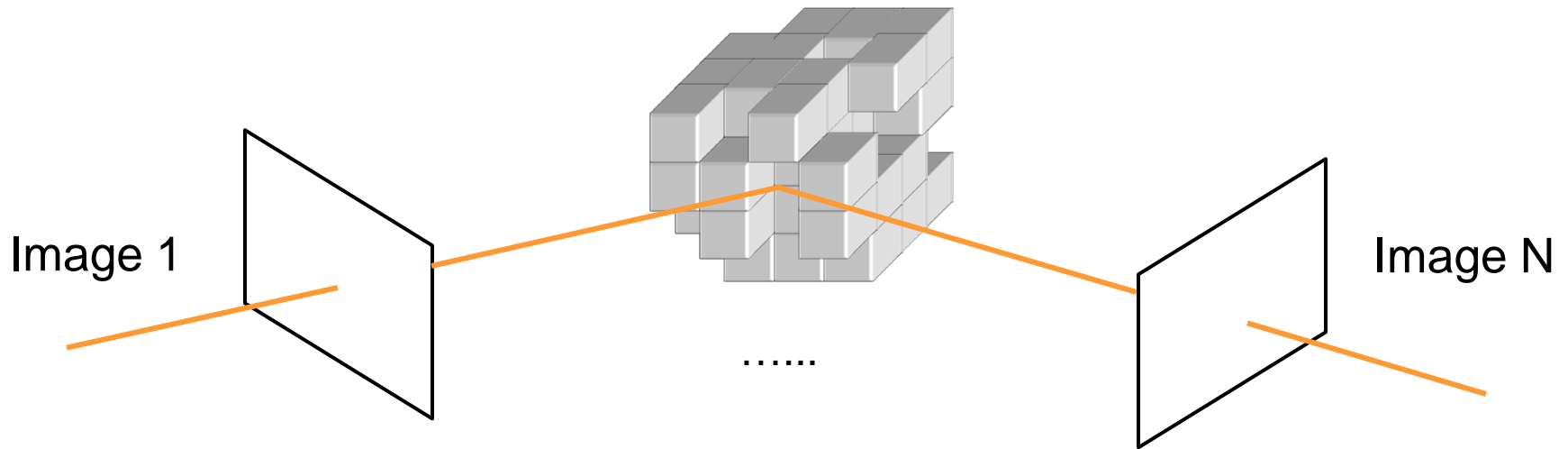
---



**Goal:** Assign RGB values to voxels in  $V$   
*photo-consistent* with images

# Space Carving

---



## Space Carving Algorithm

- Initialize to a volume  $V$  containing the true scene
- Choose a voxel on the outside of the volume
- Project to visible input images
- Carve if not photo-consistent
- Repeat until convergence



# Space Carving Results: African Violet

---



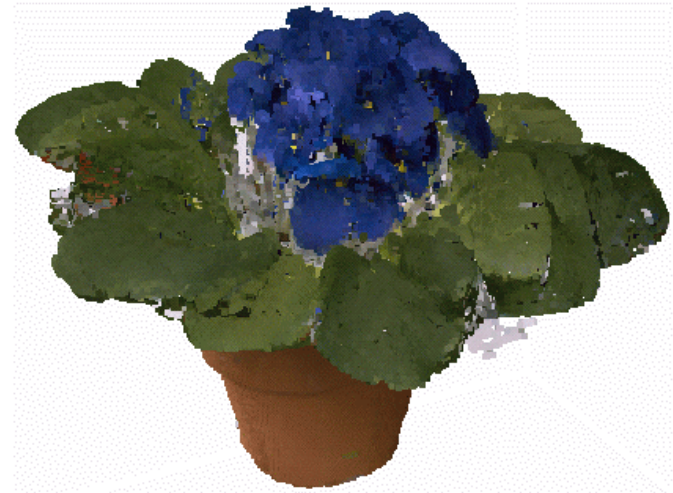
Input Image (1 of 45)



Reconstruction



Reconstruction



Reconstruction



# Space Carving Results: Hand

---



Input Image  
(1 of 100)

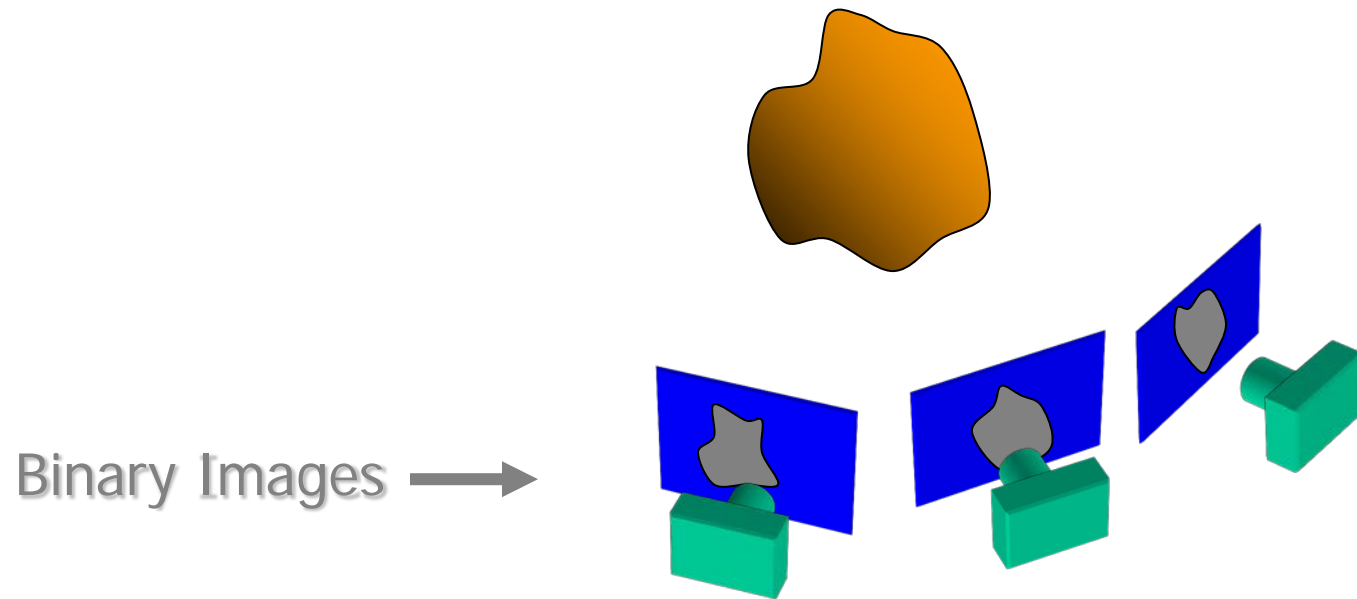


Views of Reconstruction

# Reconstruction from Silhouettes

---

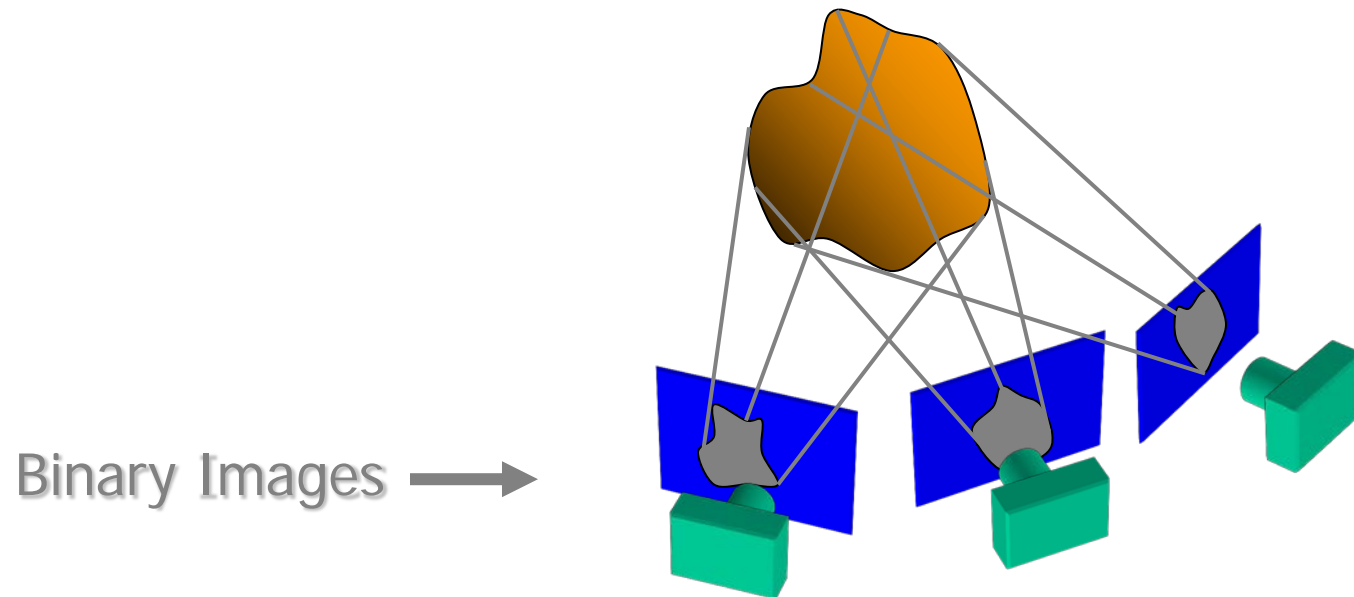
- The case of binary images: a voxel is photo-consistent if it lies inside the object's silhouette in all views



# Reconstruction from Silhouettes

---

- The case of binary images: a voxel is photo-consistent if it lies inside the object's silhouette in all views

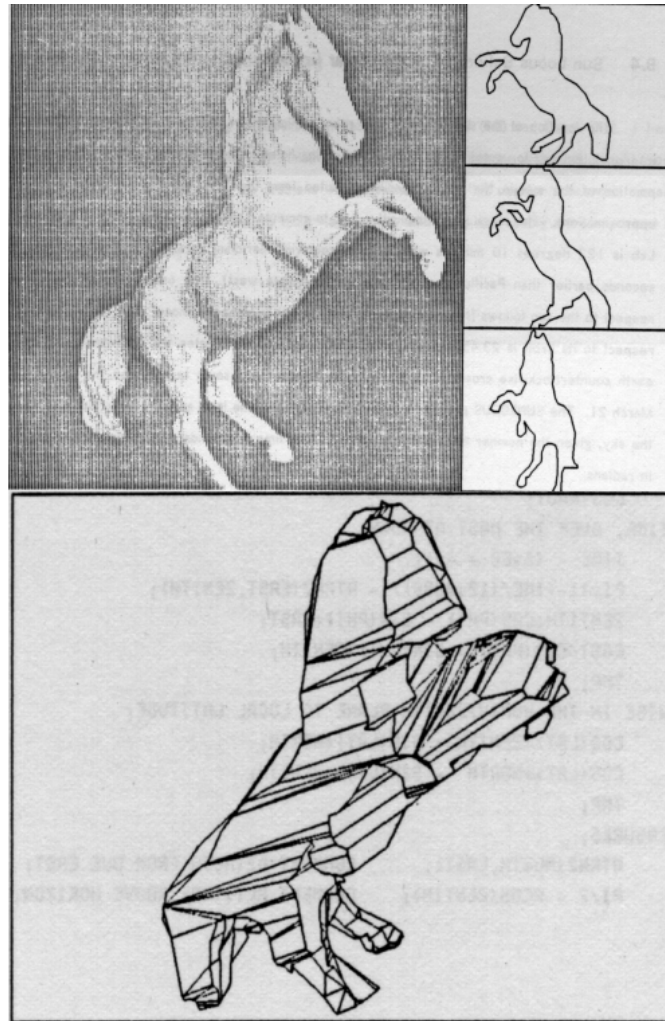


Finding the silhouette-consistent shape (*visual hull*):

- *Backproject* each silhouette
- Intersect backprojected volumes

# Volume intersection

---

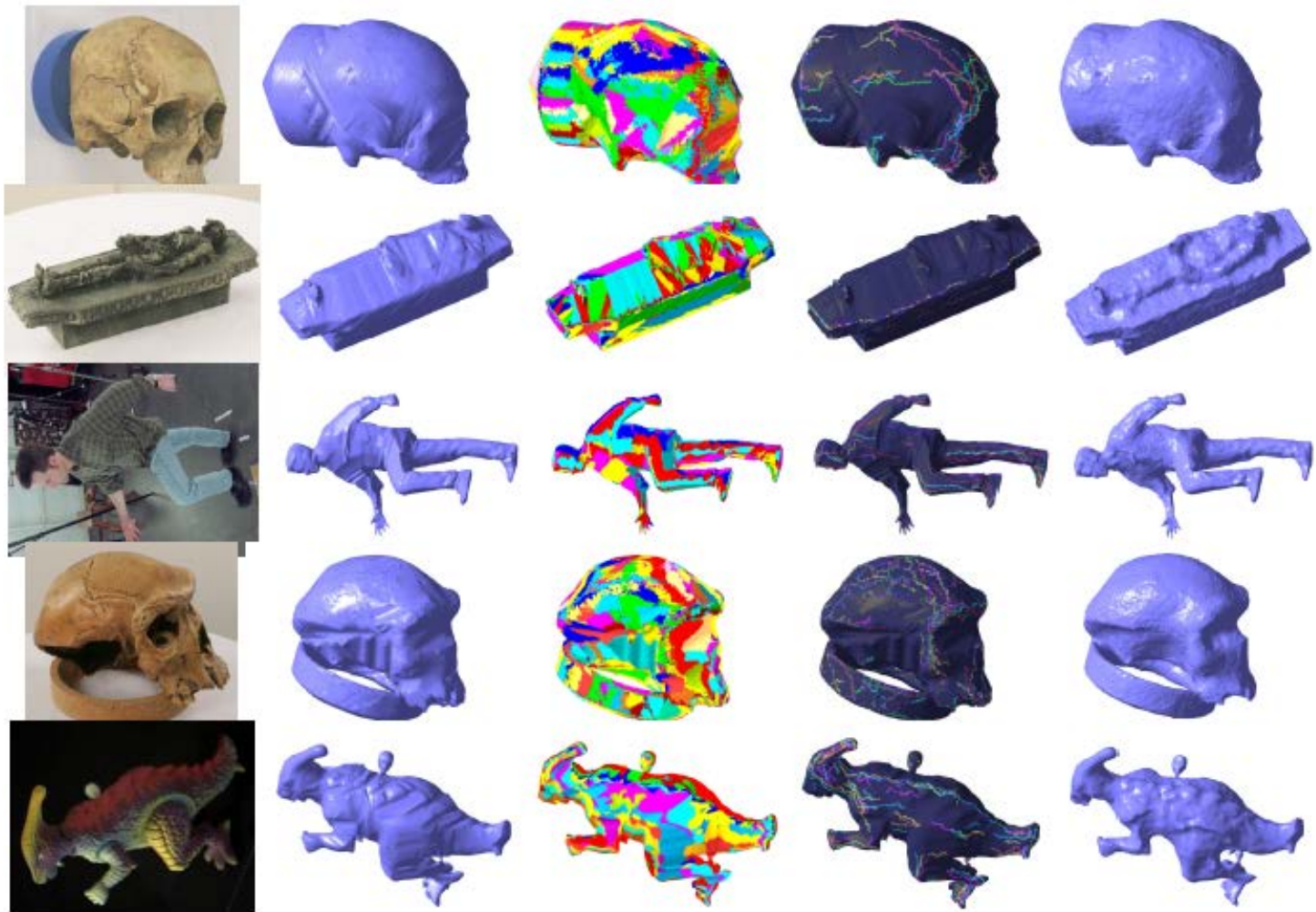


B. Baumgart, [\*Geometric Modeling for Computer Vision\*](#), Stanford Artificial Intelligence Laboratory, Memo no. AIM-249, Stanford University, October 1974.

Slide credit: S. Lazebnik

# Carved visual hulls

---



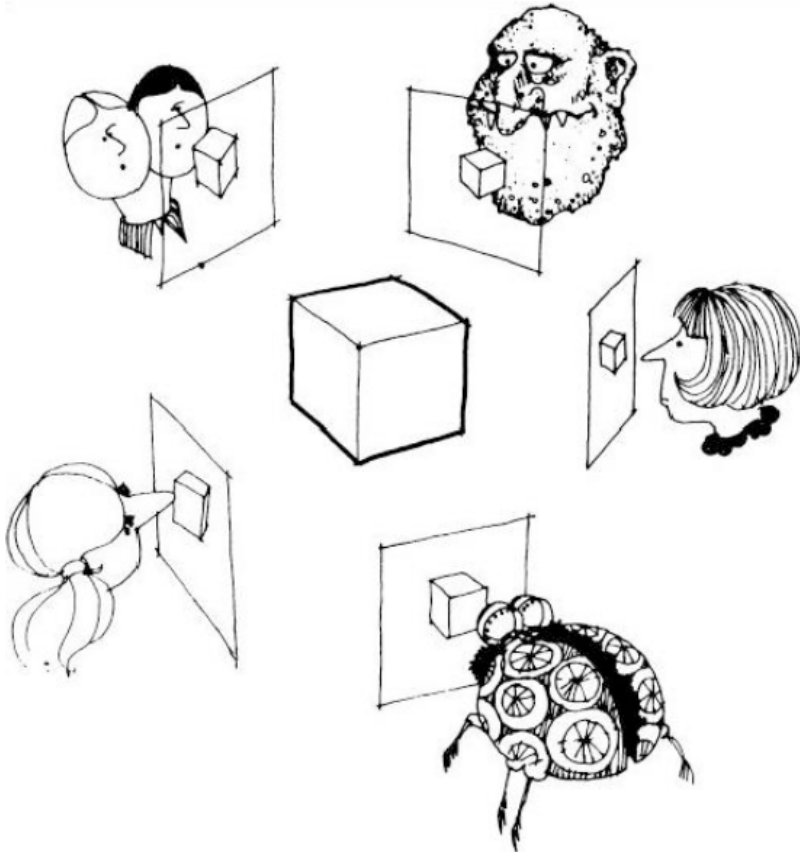
Yasutaka Furukawa and Jean Ponce, [Carved Visual Hulls for Image-Based Modeling](#), ECCV 2006.

Slide credit: S. Lazebnik

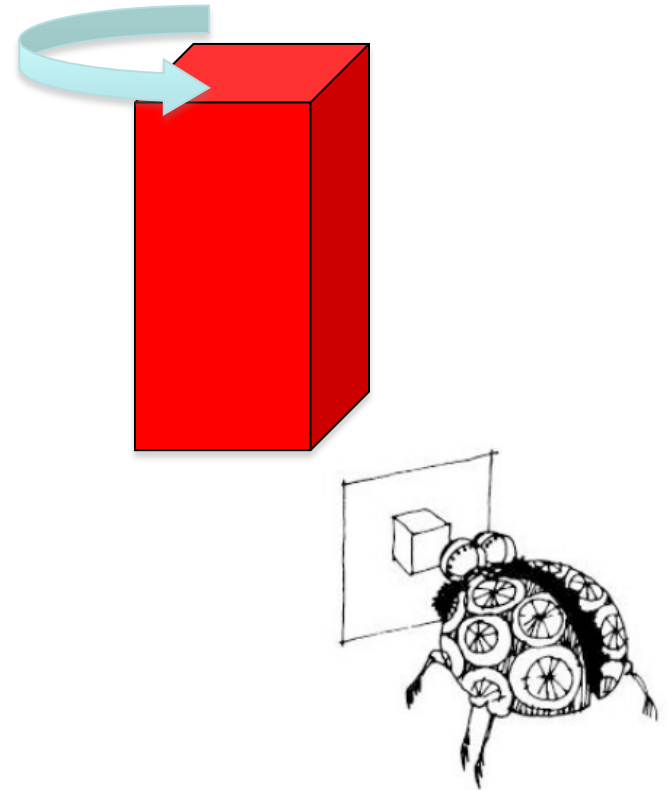


# Structure from motion

Multiple views of a static scene from different cameras

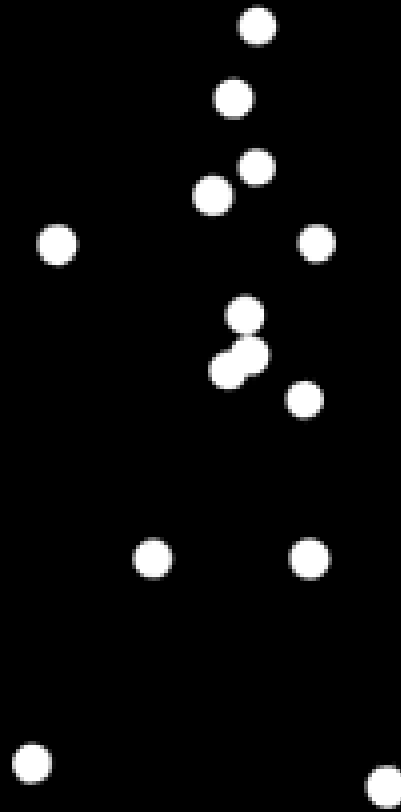


One camera, a moving object









# Point light walker

Gender and walking style:

<http://www.biomotionlab.ca/Demos/BMLwalker.html>

Inversion and other animals:

<http://www.biomotionlab.ca/Demos/scrambled.html>

# Material from Motion

Dan Kersten

[http://vision.psych.umn.edu/users/kersten/kersten-lab/demos/1\\_S\\_0001/1\\_S\\_0001.mov](http://vision.psych.umn.edu/users/kersten/kersten-lab/demos/1_S_0001/1_S_0001.mov)

# Examples



(a)



(b)



(c)



(d)

Figure 7.1: Some examples of structure from motion systems: (a–d) orthographic factorization (Tomasi and Kanade 1992); (e–f) using line matching (Schmid and Zisserman 1997). (g–k) incremental structure from motion (Snavely et al. 2006); 3D reconstructions produced by Snavely et al. (2006) for: (l) Trafalgar Square, (m) the Great Wall of China, and (c) the Old Town Square in Prague.

# Examples



(e)



(f)

Figure 7.1: Some examples of structure from motion systems: (a–d) orthographic factorization (Tomasi and Kanade 1992); (e–f) using line matching (Schmid and Zisserman 1997). (g–k) incremental structure from motion (Snavely et al. 2006); 3D reconstructions produced by Snavely et al. (2006) for: (l) Trafalgar Square, (m) the Great Wall of China, and (c) the Old Town Square in Prague.



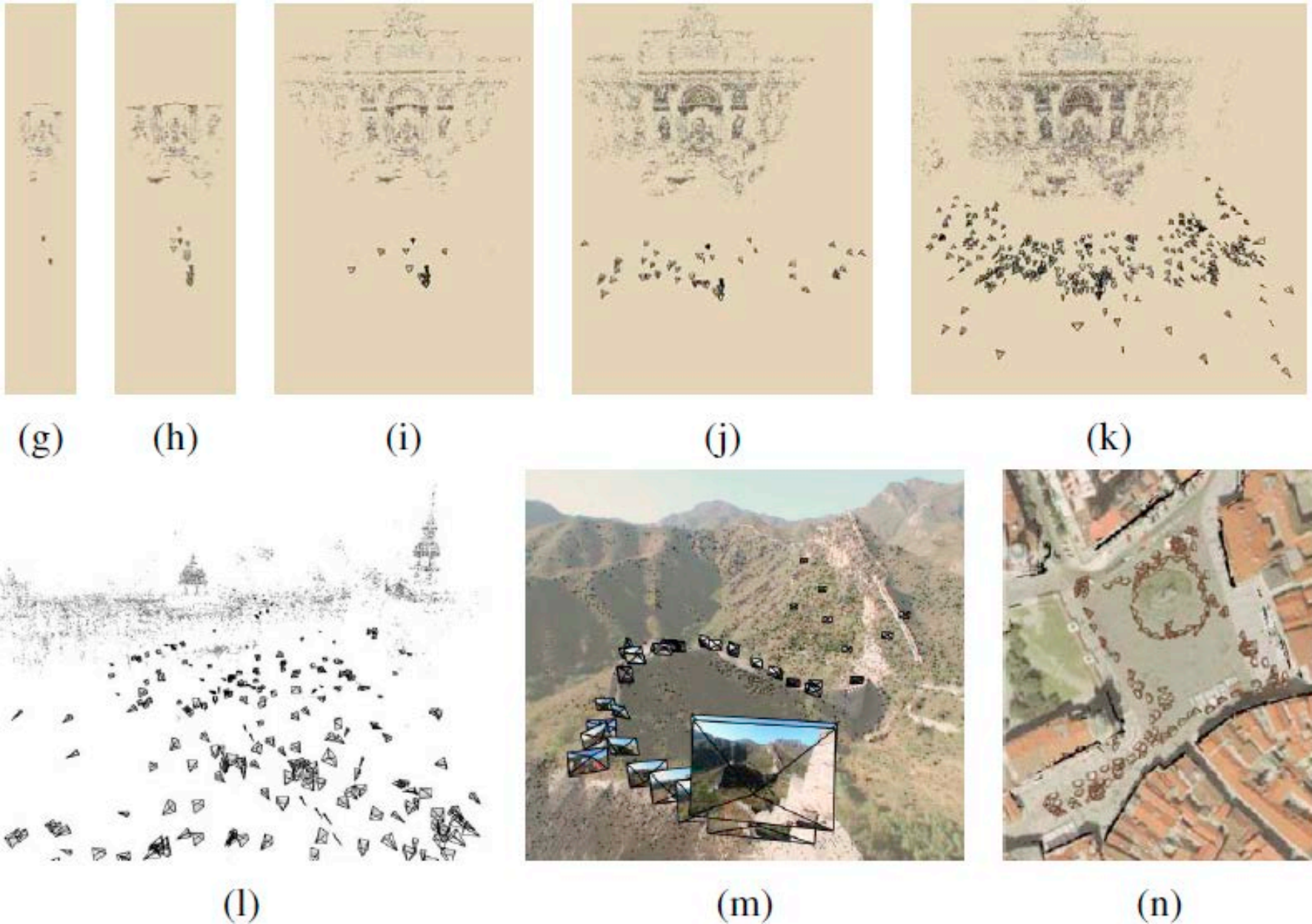


Figure 7.1: Some examples of structure from motion systems: (a–d) orthographic factorization (Tomasi and Kanade 1992); (e–f) using line matching (Schmid and Zisserman 1997). (g–k) incremental structure from motion (Snavely et al. 2006); 3D reconstructions produced by Snavely et al. (2006) for: (l) Trafalgar Square, (m) the Great Wall of China, and (n) the Old Town Square in Prague.

# KLT tracker

Kanade-Lucas-Tomasi Feature Tracker

Slide 23 from lecture 17

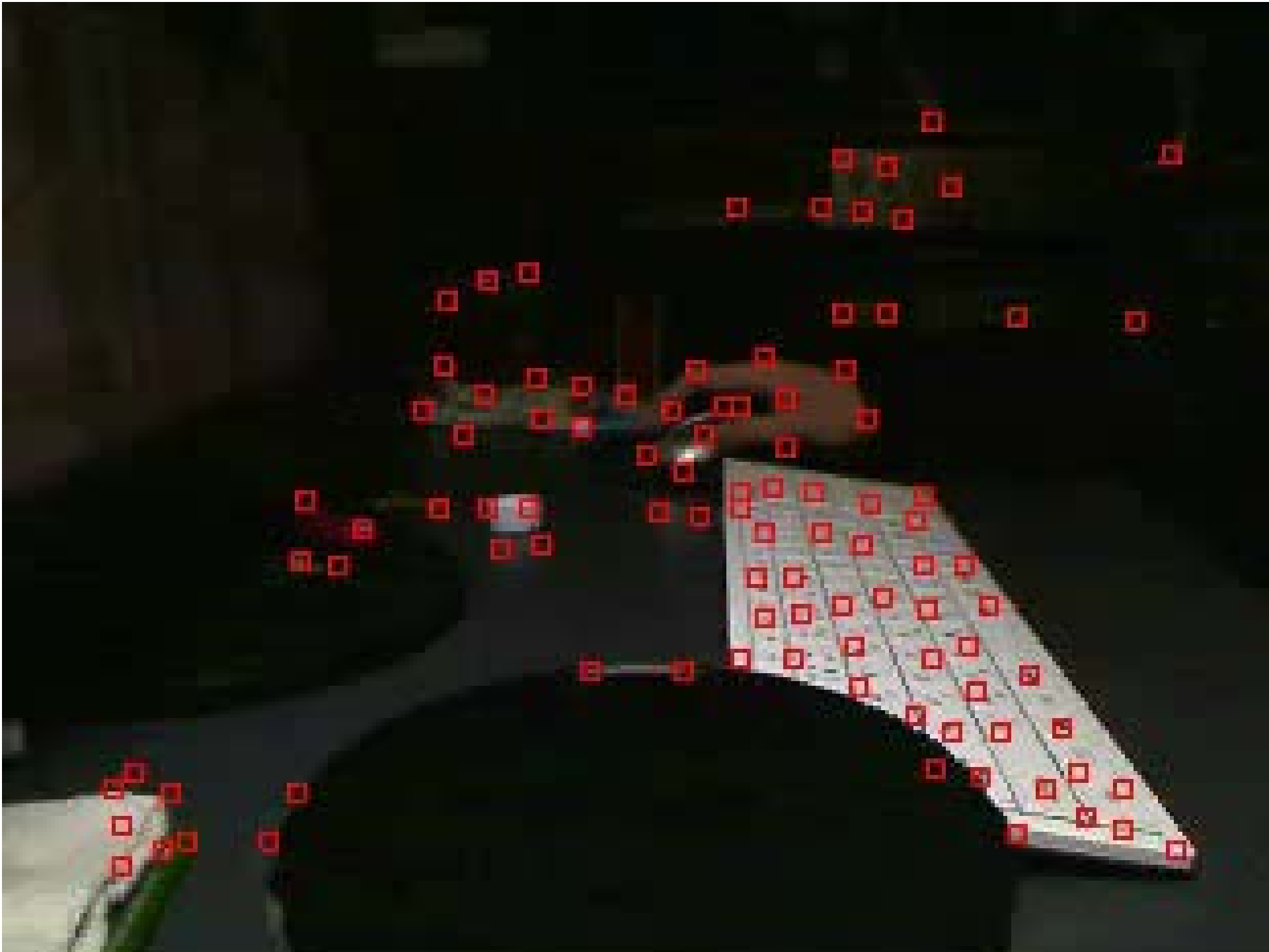
## Tracking reliable features

- Idea: no need to work on ambiguous region pixels (flat regions & line structures)
- Instead, we can track features and then propagate the tracking to ambiguous pixels
- Good features to track [Shi & Tomashi 94]

$$\begin{bmatrix} du \\ dv \end{bmatrix} = - \begin{bmatrix} \mathbf{I}_x^T \mathbf{I}_x & \mathbf{I}_x^T \mathbf{I}_y \\ \mathbf{I}_x^T \mathbf{I}_y & \mathbf{I}_y^T \mathbf{I}_y \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_x^T \mathbf{I}_t \\ \mathbf{I}_y^T \mathbf{I}_t \end{bmatrix}$$

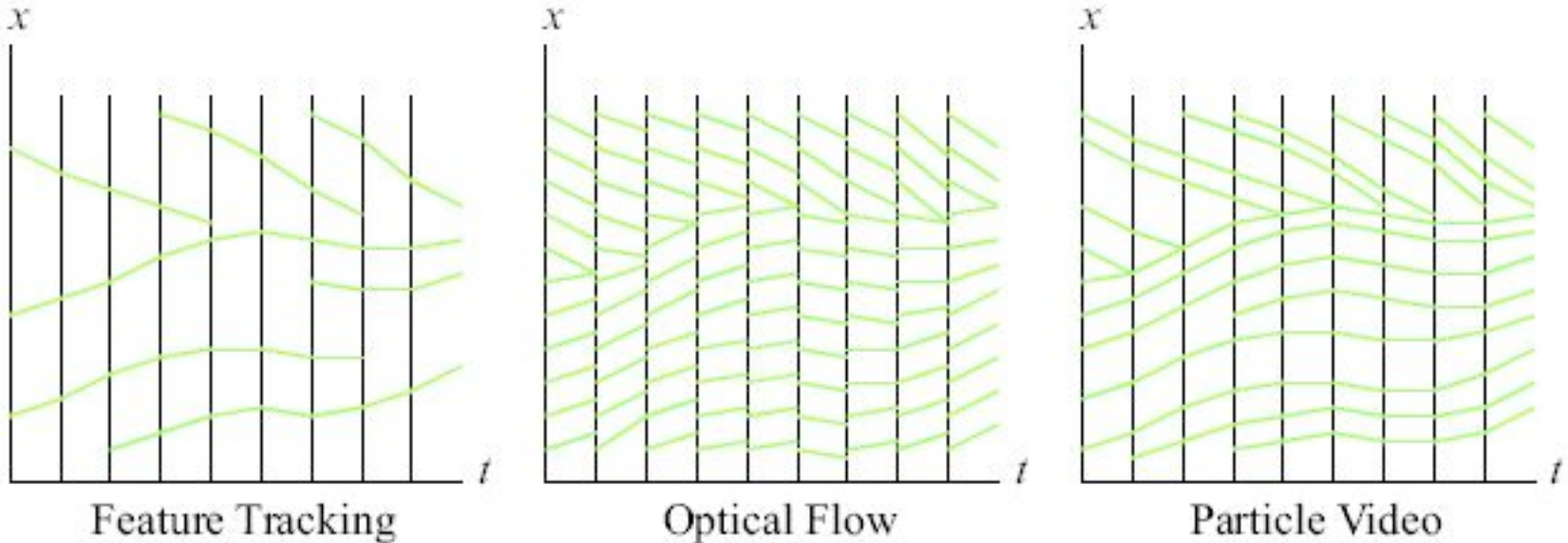
- Block matching + Lucas-Kanade refinement





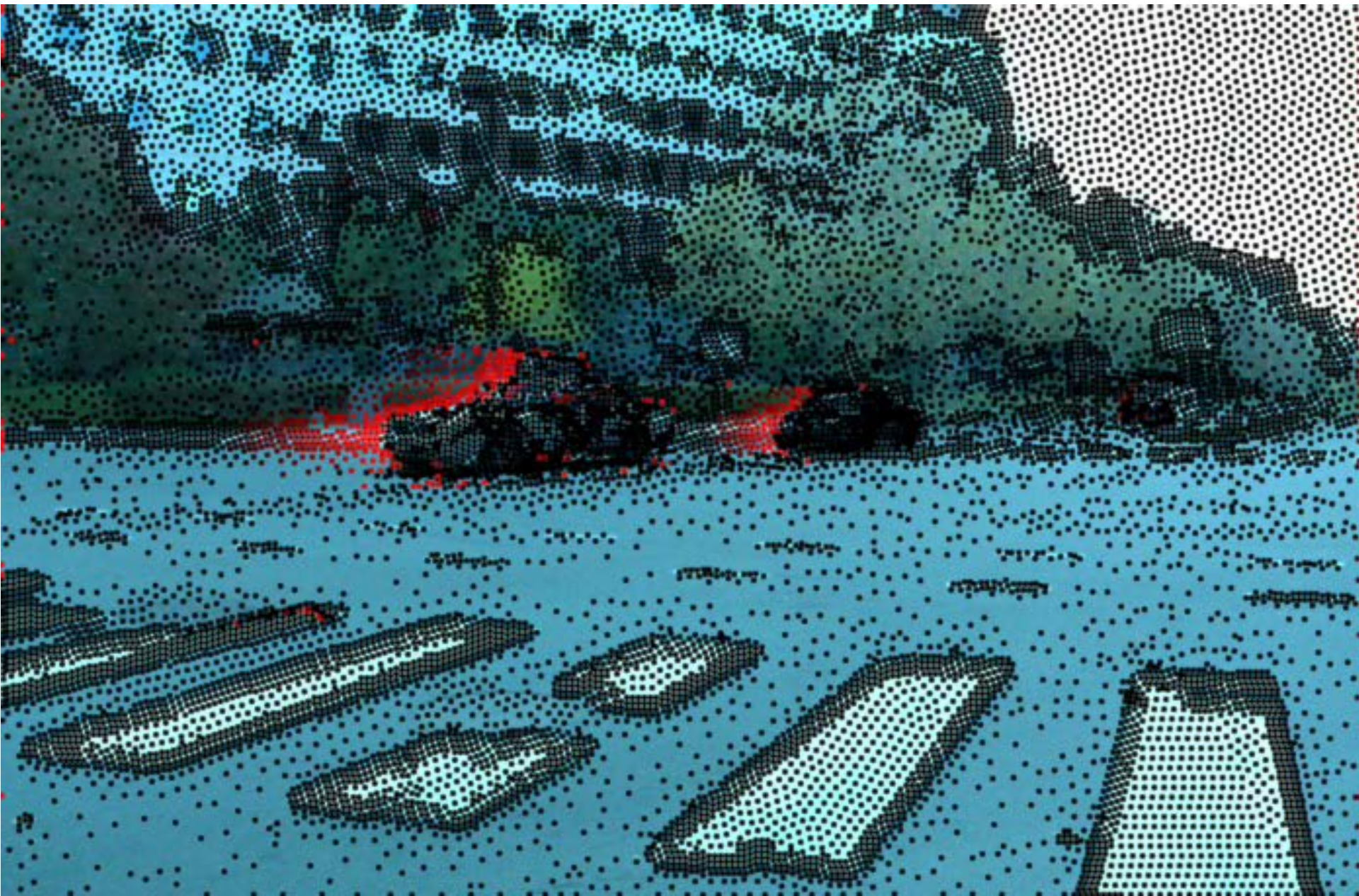
# Other trackers

Peter Sand and Seth Teller, Particle Video: Long-Range Motion Estimation using Point Trajectories, CVPR, 2006



Jepson, A.D., Fleet, D.J., El-Maraghi, T. (2003) Robust online appearance models for visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(10):1296-1311







## Shape and Motion from Image Streams under Orthography: a Factorization Method

CARLO TOMASI

*Department of Computer Science, Cornell University, Ithaca, NY 14850*

TAKEO KANADE

*School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213*

Received

### Abstract

Inferring scene geometry and camera motion from a stream of images is possible in principle, but is an ill-conditioned problem when the objects are distant with respect to their size. We have developed a *factorization method* that can overcome this difficulty by recovering shape and motion under orthography without computing depth as an intermediate step.

An image stream can be represented by the  $2F \times P$  measurement matrix of the image coordinates of  $P$  points tracked through  $F$  frames. We show that under orthographic projection this matrix is of rank 3.

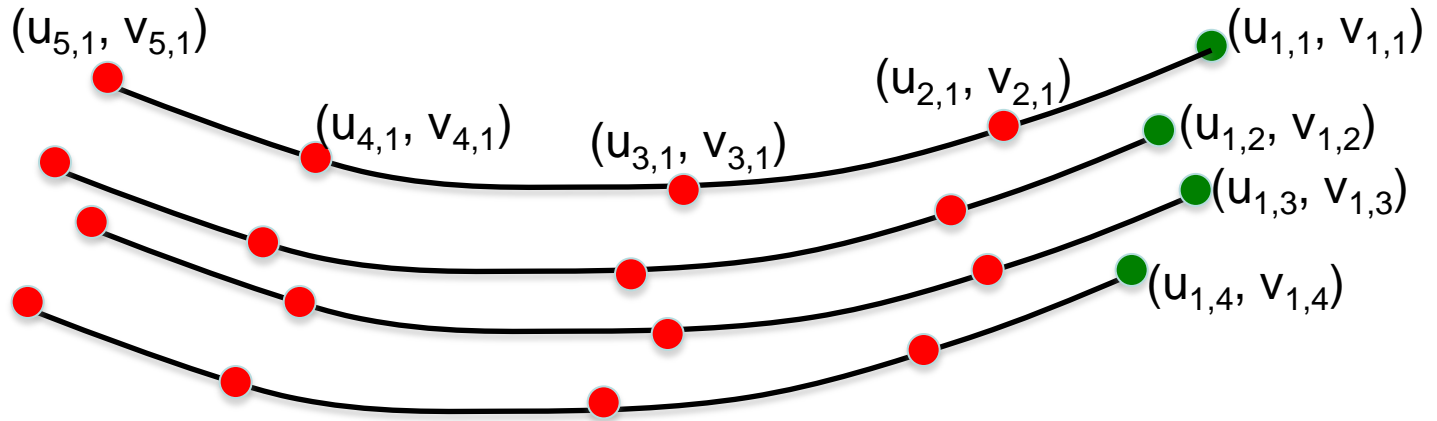
Based on this observation, the factorization method uses the singular-value decomposition technique to factor the measurement matrix into two matrices which represent object shape and camera rotation respectively. Two of the three translation components are computed in a preprocessing stage. The method can also handle and obtain a full solution from a partially filled-in measurement matrix that may result from occlusions or tracking failures.

The method gives accurate results, and does not introduce smoothing in either shape or motion. We demonstrate this with a series of experiments on laboratory and outdoor image streams, with and without occlusions.

We track P points in F frames

Each point has coordinates:  $(u_{fp}, v_{fp})$

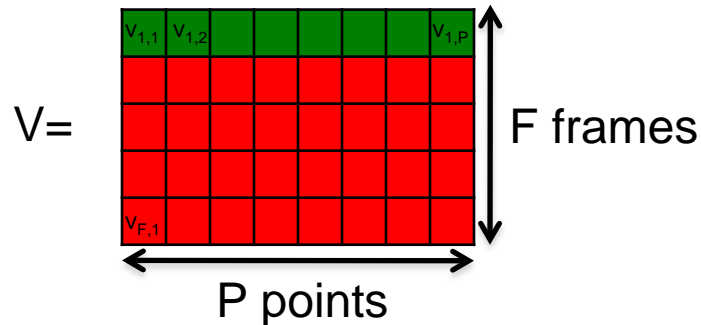
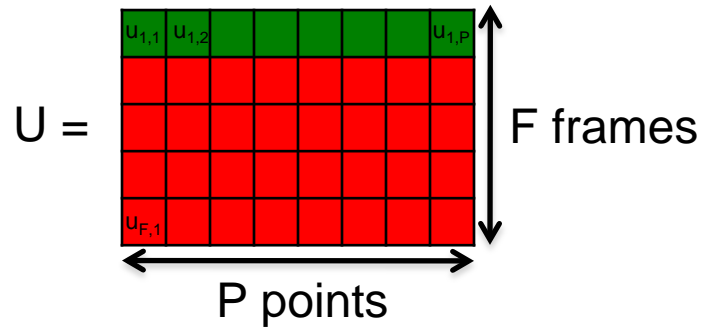
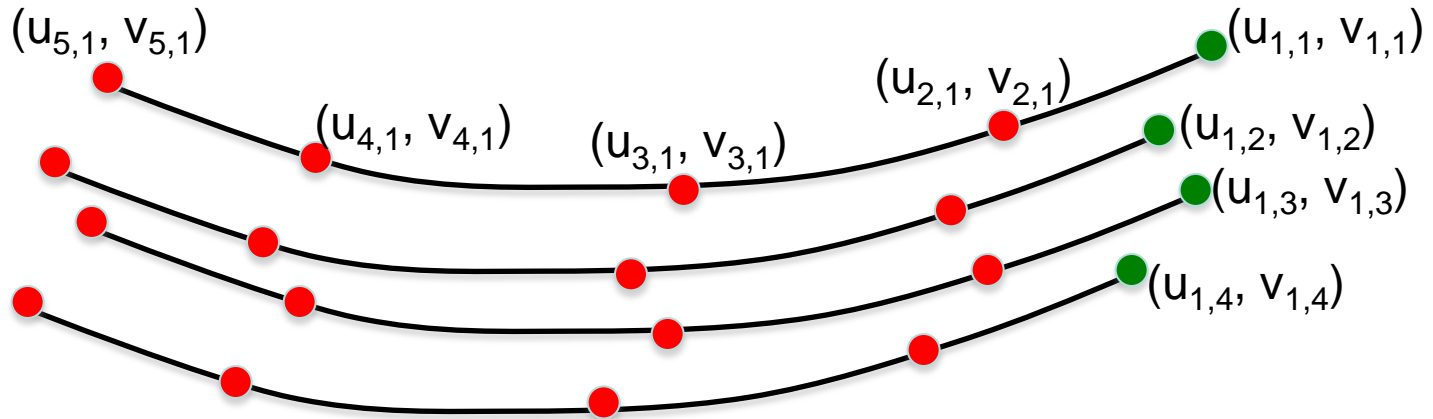
(I will use the notation from the IJCV 92 paper )



We track P points in F frames

Each point has coordinates:  $(u_{fp}, v_{fp})$

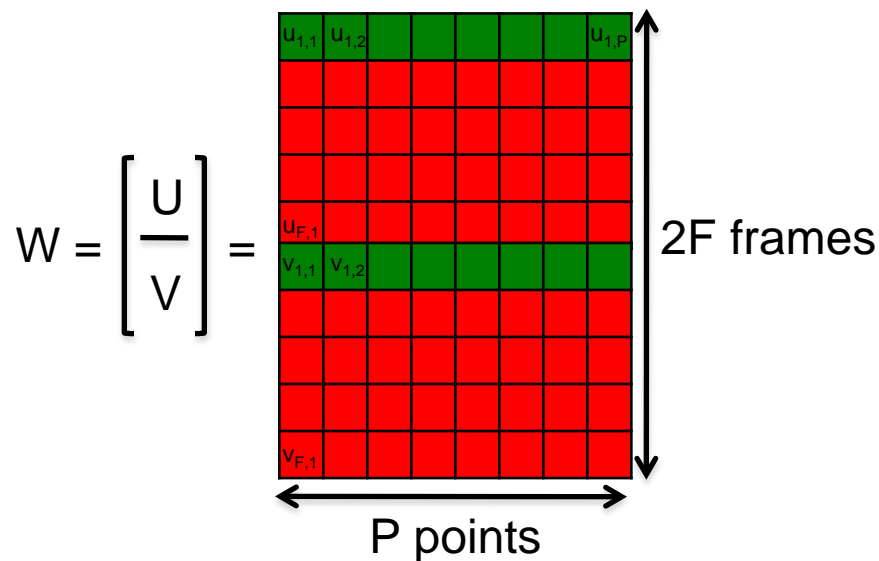
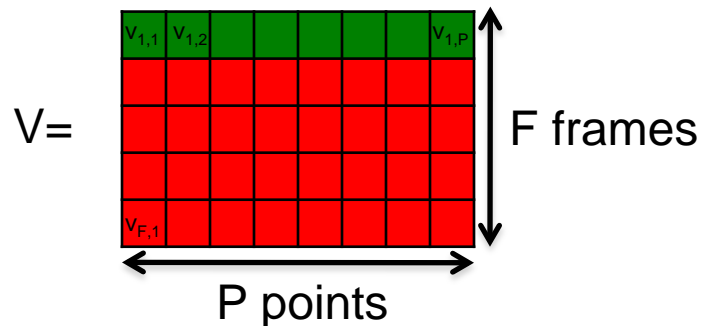
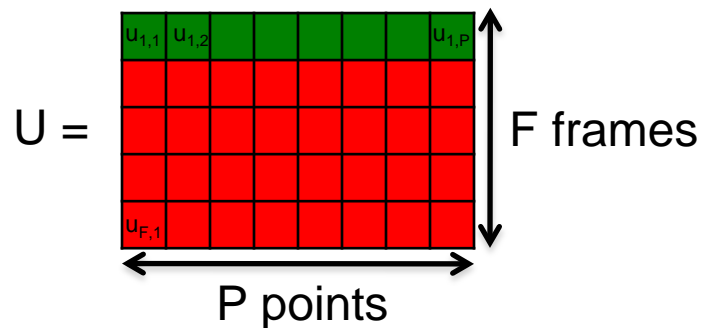
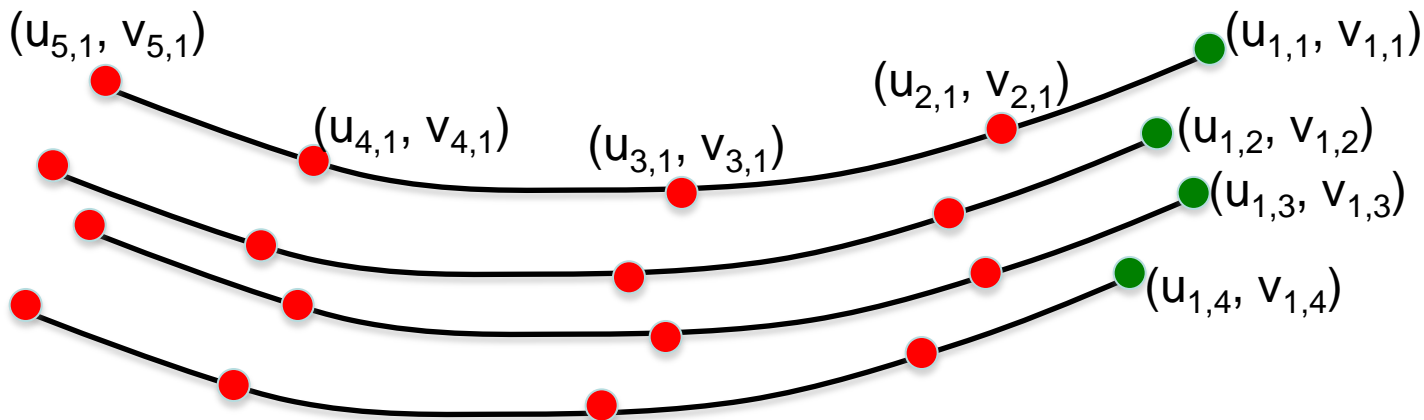
(I will use the notation from the IJCV 92 paper )



We track  $P$  points in  $F$  frames

Each point has coordinates:  $(u_{fp}, v_{fp})$

(I will use the notation from the IJCV 92 paper)



Measurement matrix

# Registered measurement matrix

$$W = \begin{bmatrix} U \\ V \end{bmatrix} = \begin{array}{|c|c|c|c|c|c|} \hline \color{green}u_{1,1} & \color{green}u_{1,2} & & & & \color{green}u_{1,P} \\ \hline \color{red}& \color{red}& \color{red}& \color{red}& \color{red}& \color{red} \\ \hline \color{red}& \color{red}& \color{red}& \color{red}& \color{red}& \color{red} \\ \hline \color{red}& \color{red}& \color{red}& \color{red}& \color{red}& \color{red} \\ \hline \color{red}u_{F,1} & \color{green}v_{1,1} & \color{green}v_{1,2} & & & \\ \hline \color{red}& \color{red}& \color{red}& \color{red}& \color{red}& \color{red} \\ \hline \color{red}& \color{red}& \color{red}& \color{red}& \color{red}& \color{red} \\ \hline \color{red}& \color{red}& \color{red}& \color{red}& \color{red}& \color{red} \\ \hline \color{red}v_{F,1} & \color{red}& \color{red}& \color{red}& \color{red}& \color{red} \\ \hline \end{array}$$

Measurement matrix

Centering: subtract from each row its mean

$$\tilde{U}_{fp} = U_{fp} - 1/P \sum_{p=1}^P U_{fp}$$

$$\tilde{V}_{fp} = V_{fp} - 1/P \sum_{p=1}^P V_{fp}$$

$$\tilde{W} = \begin{bmatrix} \tilde{U} \\ \tilde{V} \end{bmatrix} \quad \begin{array}{l} \text{Registered} \\ \text{measurement matrix} \end{array}$$

# The Rank Theorem

The matrix  $\tilde{W}$ , without noise, has at most rank 3.

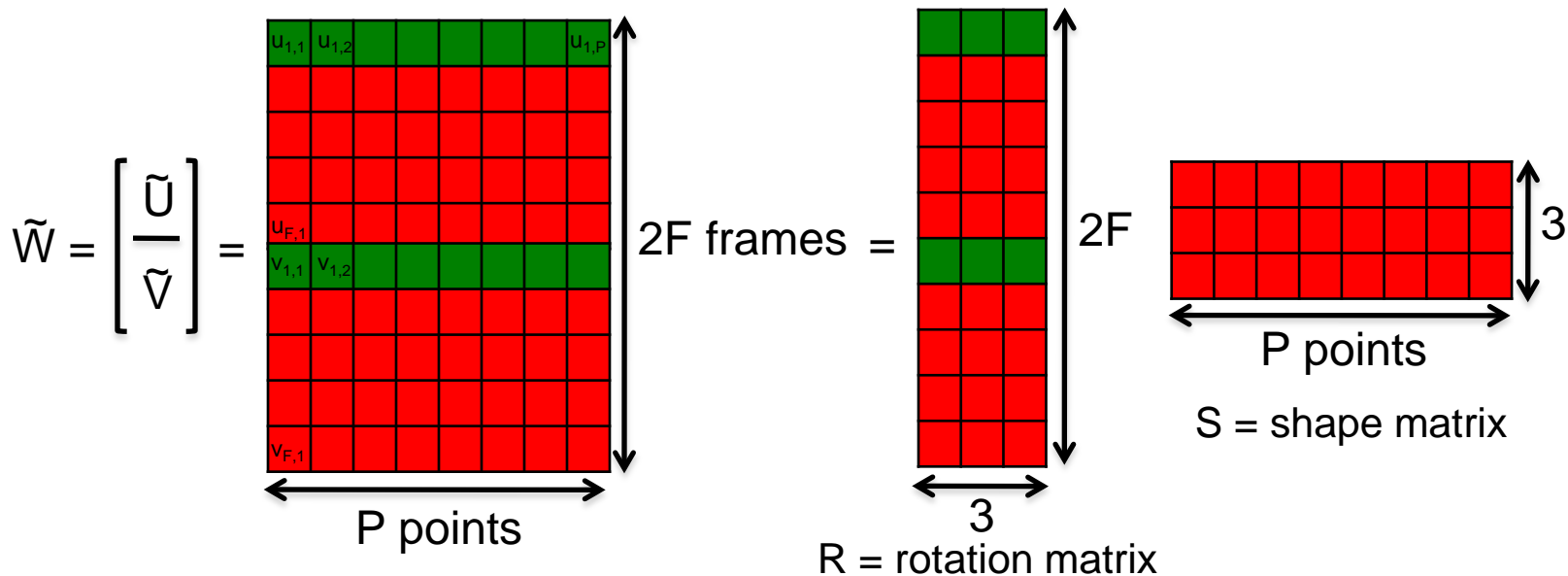
**Proof:** The matrix  $W$  can be decomposed as the product of two matrices, a rotation matrix  $R$  and a shape matrix  $S$ .



# The Rank Theorem

The matrix  $\tilde{W}$ , without noise, has at most rank 3.

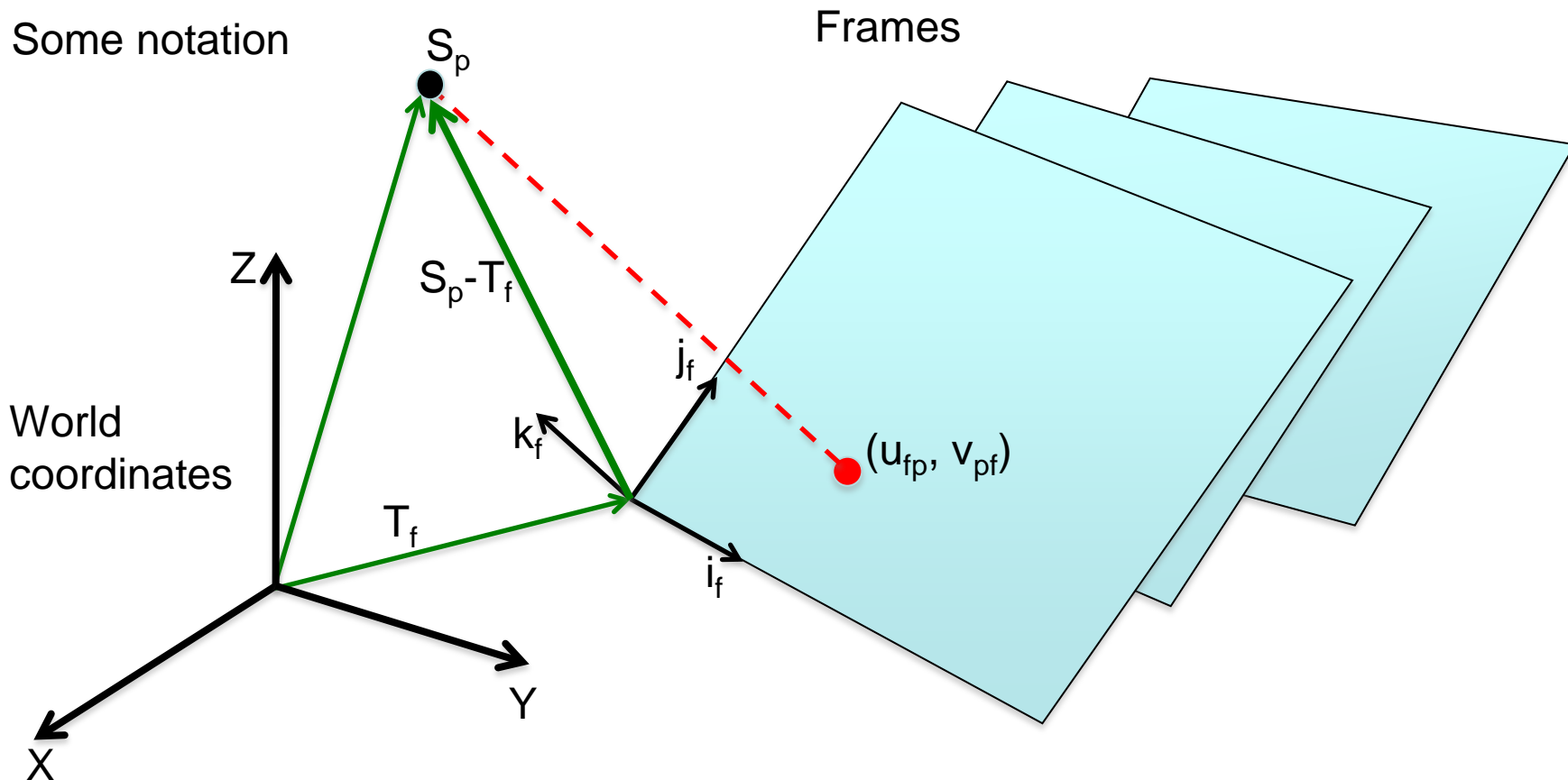
**Proof:** The matrix  $W$  can be decomposed as the product of two matrices, a rotation matrix  $R$  and a shape matrix  $S$ .



$\tilde{W}$  = Registered measurement matrix

# Proof

Some notation



$$u_{fp} = i_f^T (S_p - T_f)$$

$$v_{fp} = j_f^T (S_p - T_f)$$

Image coordinates

World coordinates

We define the origin of the world coordinates so that:

$$\frac{1}{P} \sum_{p=1}^P S_p = 0$$

# Proof

$$u_{fp} = i_f^T (S_p - T_f)$$

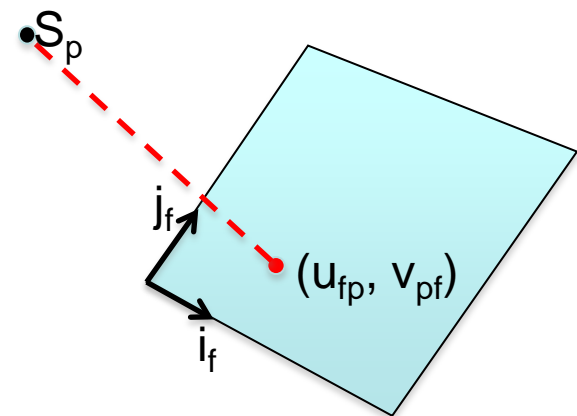
$$v_{fp} = j_f^T (S_p - T_f)$$

Image coordinates

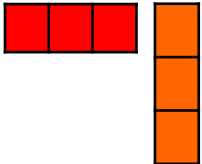
World coordinates

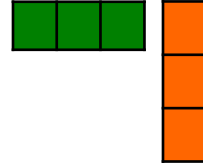
$$\begin{aligned} \tilde{u}_{fp} &= u_{fp} - 1/P \sum_{p=1}^P u_{fp} = \leftarrow u_{fp} = i_f^T (S_p - T_f) \\ &= i_f^T (S_p - 1/P \sum_{p=1}^P S_p) = \leftarrow 1/P \sum_{p=1}^P S_p = 0 \\ &= i_f^T S_p \end{aligned}$$

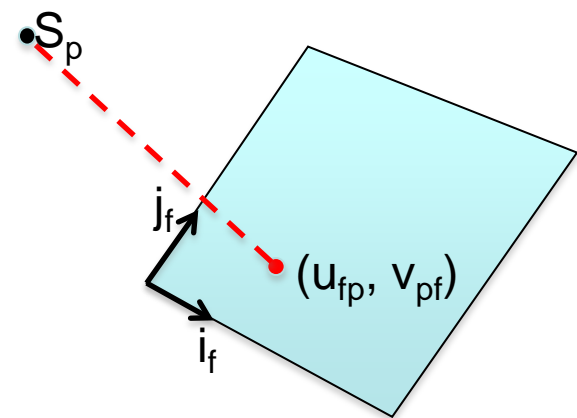
$$\tilde{v}_{fp} = j_f^T S_p$$



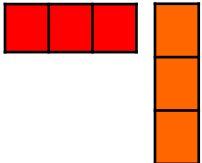
# Proof

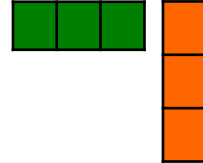
$$\tilde{u}_{fp} = \mathbf{i}_f^T \mathbf{S}_p$$


$$\tilde{v}_{fp} = \mathbf{j}_f^T \mathbf{S}_p$$




# Proof

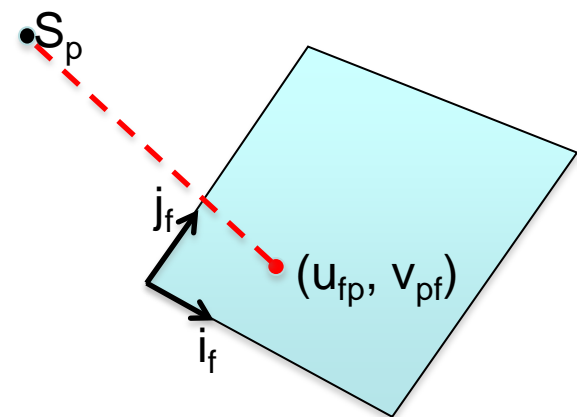
$$\tilde{u}_{fp} = \mathbf{i}_f^T \mathbf{S}_p$$


$$\tilde{v}_{fp} = \mathbf{j}_f^T \mathbf{S}_p$$


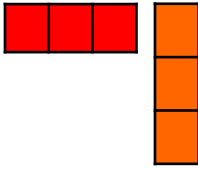
The registered measurement matrix ( $\tilde{W}$ ) can be decomposed as:

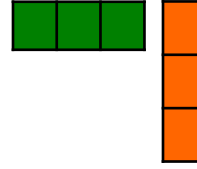
$$\tilde{W} = \begin{bmatrix} \tilde{U} \\ \tilde{V} \end{bmatrix} = \begin{array}{|c|c|c|c|c|} \hline u_{1,1} & u_{1,2} & & & u_{1,P} \\ \hline & & & & \\ \hline & & & & \\ \hline u_{F,1} & & & & \\ \hline v_{1,1} & v_{1,2} & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline v_{F,1} & & & & \\ \hline \end{array} \begin{array}{l} \updownarrow \\ 2F \text{ frames} \\ \updownarrow \end{array}$$

$\longleftrightarrow$  P points  $\longleftrightarrow$

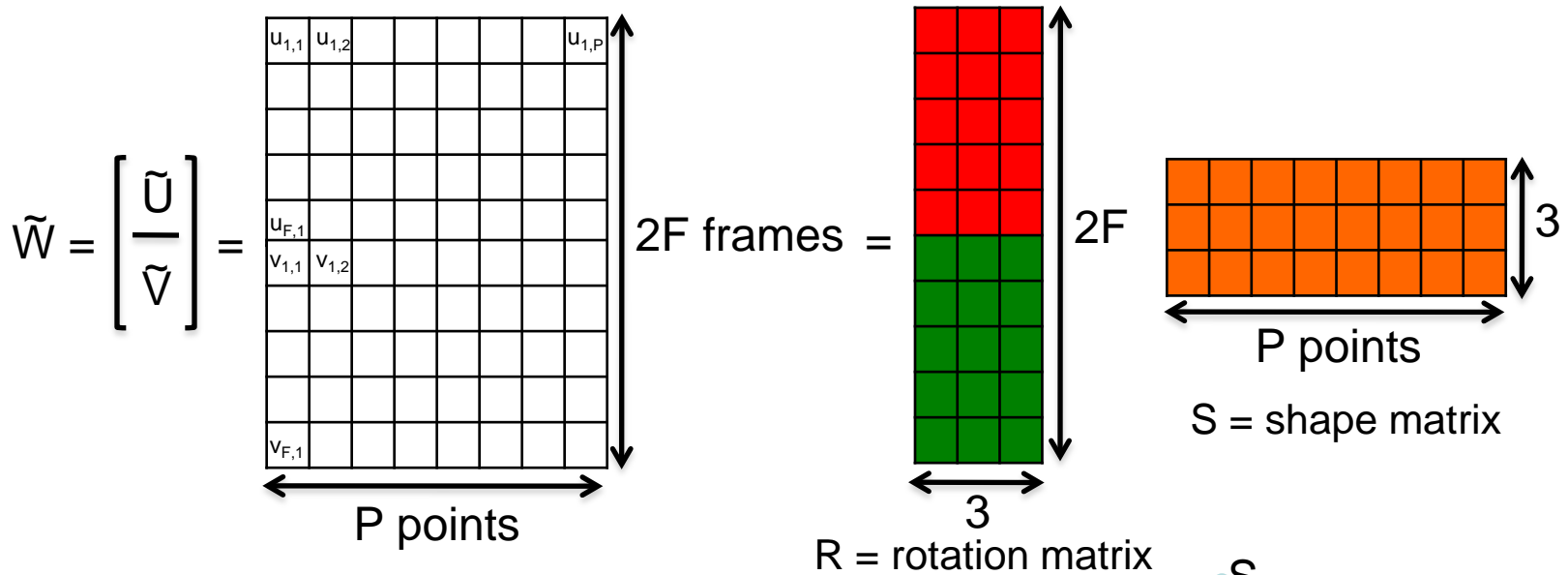


# Proof

$$\tilde{u}_{fp} = \mathbf{i}_f^T \mathbf{S}_p$$


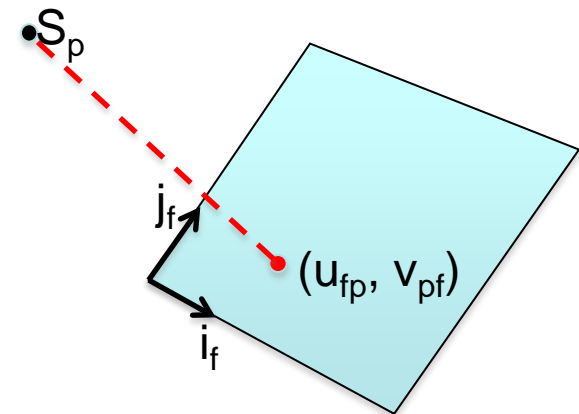
$$\tilde{v}_{fp} = \mathbf{j}_f^T \mathbf{S}_p$$


The registered measurement matrix ( $\tilde{W}$ ) can be decomposed as:



$$\tilde{W} = R S$$

$$\text{Rank}(\tilde{W}) \leq 3$$



# Factorization

Given the measurement matrix, we want to find the factorization:

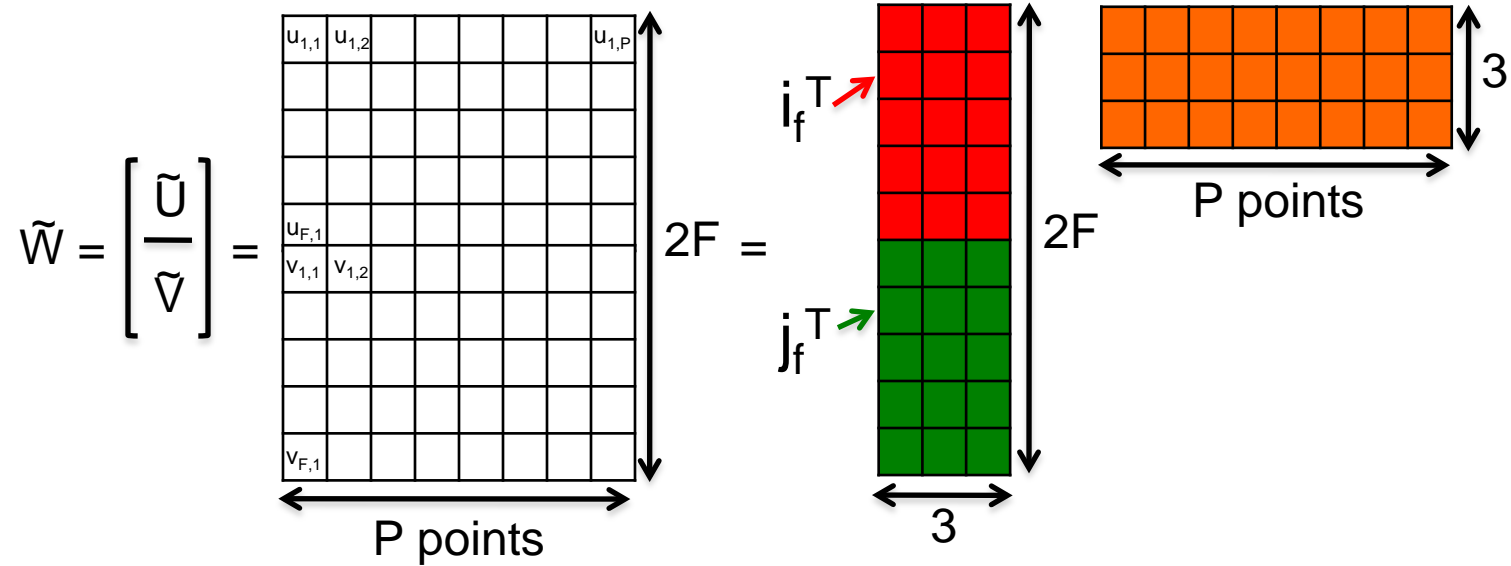
$$\tilde{W} = \begin{bmatrix} \tilde{U} \\ \tilde{V} \end{bmatrix} = \begin{array}{c} \begin{array}{cccc} u_{1,1} & u_{1,2} & & u_{1,P} \end{array} \\ \begin{array}{cccc} & & & \end{array} \\ \begin{array}{cccc} & & & \end{array} \\ \begin{array}{cccc} & & & \end{array} \\ \begin{array}{cccc} & & & \end{array} \\ \begin{array}{cccc} v_{1,1} & v_{1,2} & & \end{array} \\ \begin{array}{cccc} & & & \end{array} \\ \begin{array}{cccc} & & & \end{array} \\ \begin{array}{cccc} & & & \end{array} \\ \begin{array}{cccc} & & & \end{array} \\ \begin{array}{cccc} v_{F,1} & & & \end{array} \end{array} \begin{array}{l} \updownarrow \\ 2F \text{ frames} \\ \updownarrow \end{array} = \mathbf{R} \mathbf{S}$$

← P points →

**Problem:** this factorization is not unique. For any invertible matrix  $Q$ , then:

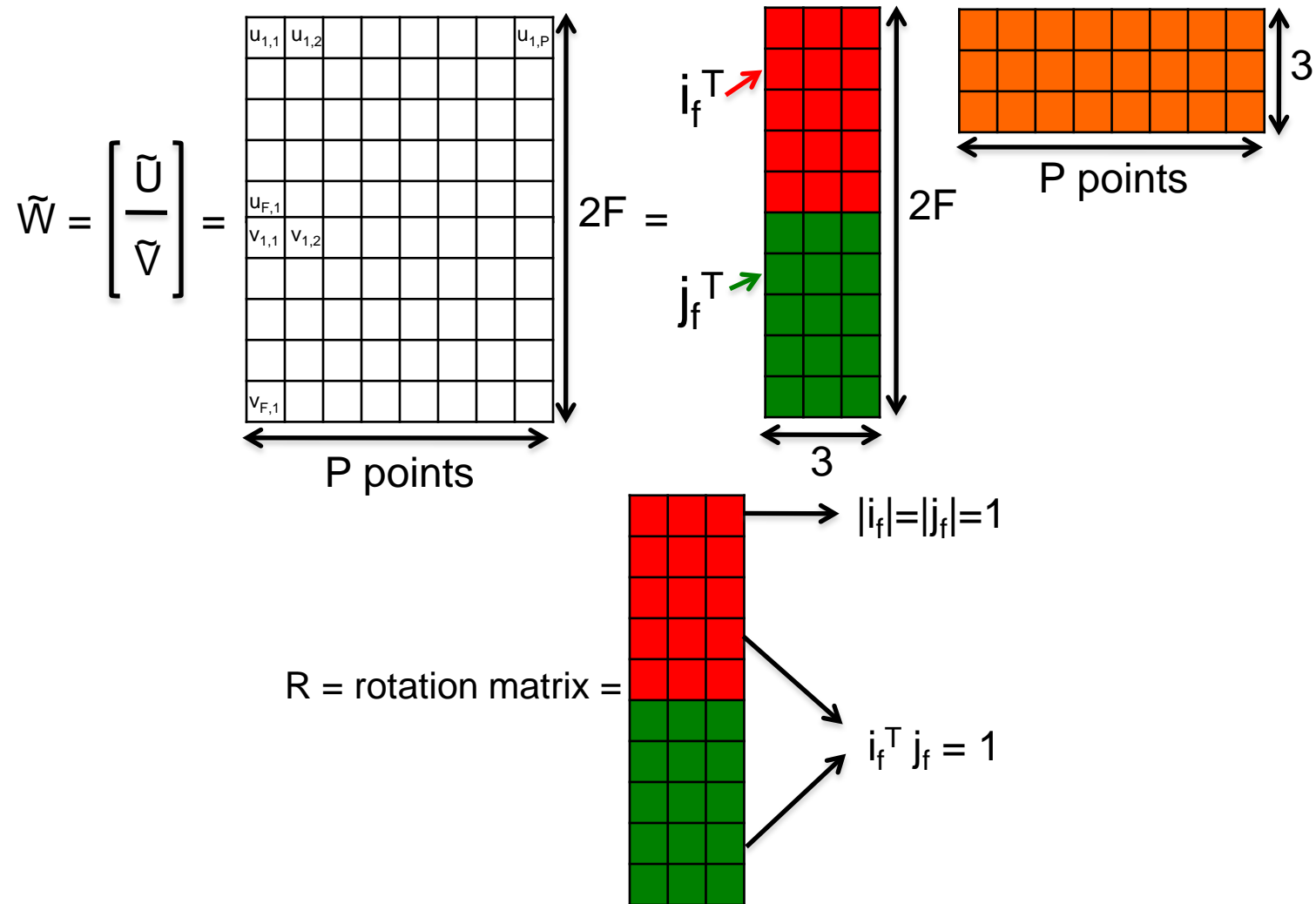
$$\tilde{W} = \mathbf{R} \mathbf{S} = \hat{\mathbf{R}} \mathbf{Q} \mathbf{Q}^{-1} \hat{\mathbf{S}} = (\hat{\mathbf{R}} \mathbf{Q}) (\mathbf{Q}^{-1} \hat{\mathbf{S}})$$

# Additional constraints





# Additional constraints



- The rows of  $R$  are unit norm
- Rows  $1 \dots F$  should be orthogonal to corresponding rows  $F+1 \dots 2F$

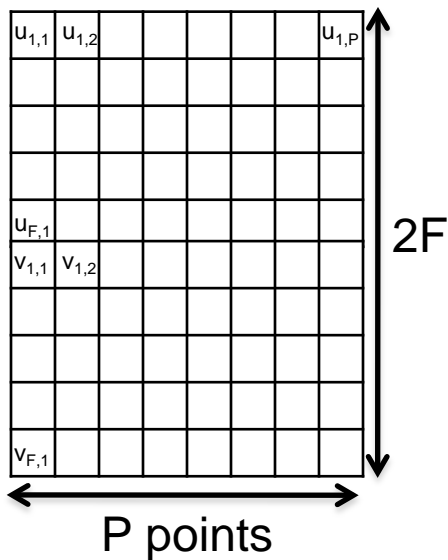
# Factorization algorithm

1. Use SVD to decompose  $\tilde{W}$  into rank 3 matrices
2. Impose constraints to find  $Q$

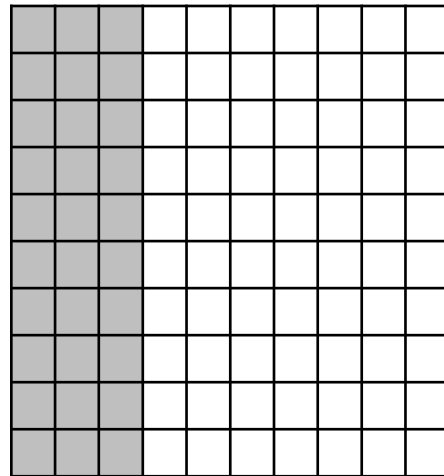
# Factorization algorithm

1. Use SVD to decompose  $\tilde{W}$  into rank 3 matrices
2. Impose constraints to find  $Q$

$$\tilde{W} = U D V^T$$

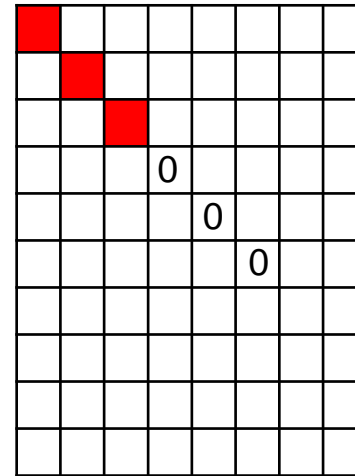


=



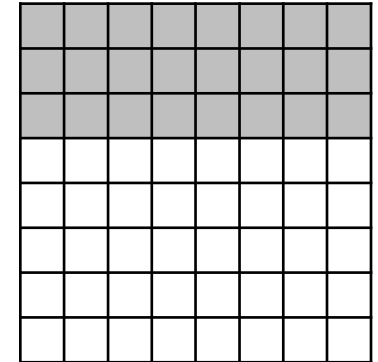
$2F \times 2F$

$U$



$2F \times P$

$D$



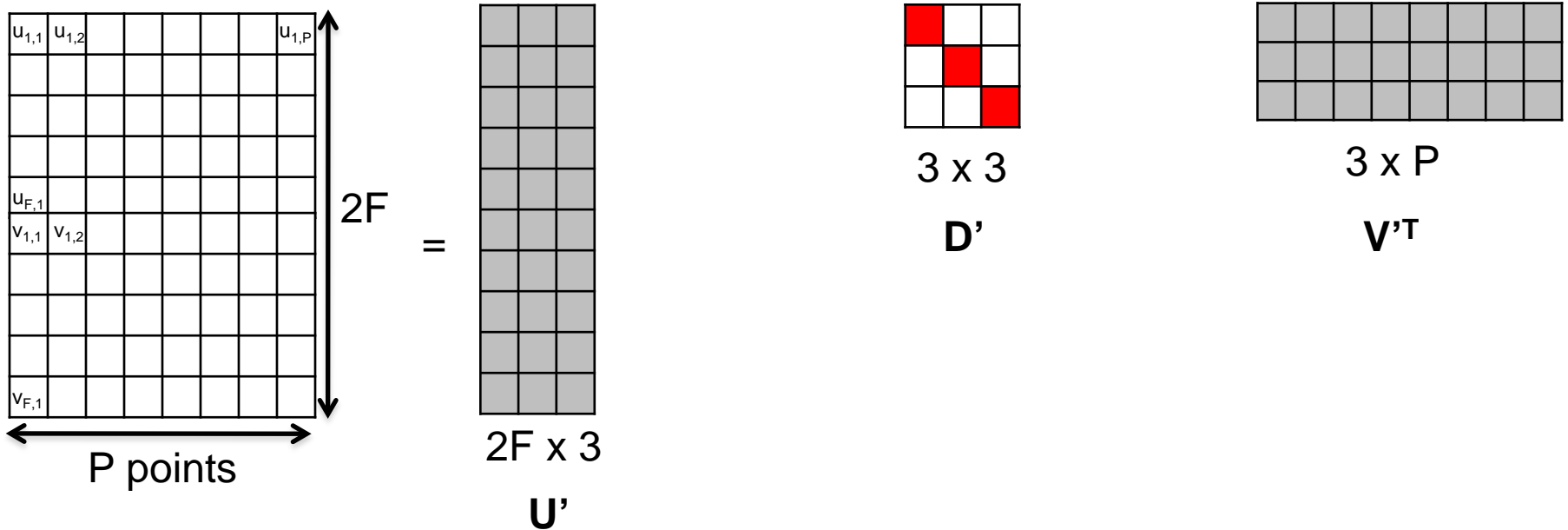
$P \times P$

$V^T$

# Factorization algorithm

1. Use SVD to decompose  $\tilde{W}$  into rank 3 matrices
2. Impose constraints to find  $Q$

$$\tilde{W} = U D V^T$$



$$\hat{R} = U' D'^{1/2} \quad \hat{S} = D'^{1/2} V'^T$$

$\hat{R}$  and  $\hat{S}$  are one of the possible decompositions. We need to add constraints to  $R$ .

# Factorization algorithm

1. Use SVD to decompose  $\tilde{W}$  into rank 3 matrices
2. Impose constraints to find Q

$$\hat{R} = U' D'^{1/2} \quad \hat{S} = D'^{1/2} V'^T$$

We look for Q such that the final rotation and shape matrices are:

$$R = \hat{R}Q \quad S = Q^{-1}\hat{S}$$

# Factorization algorithm

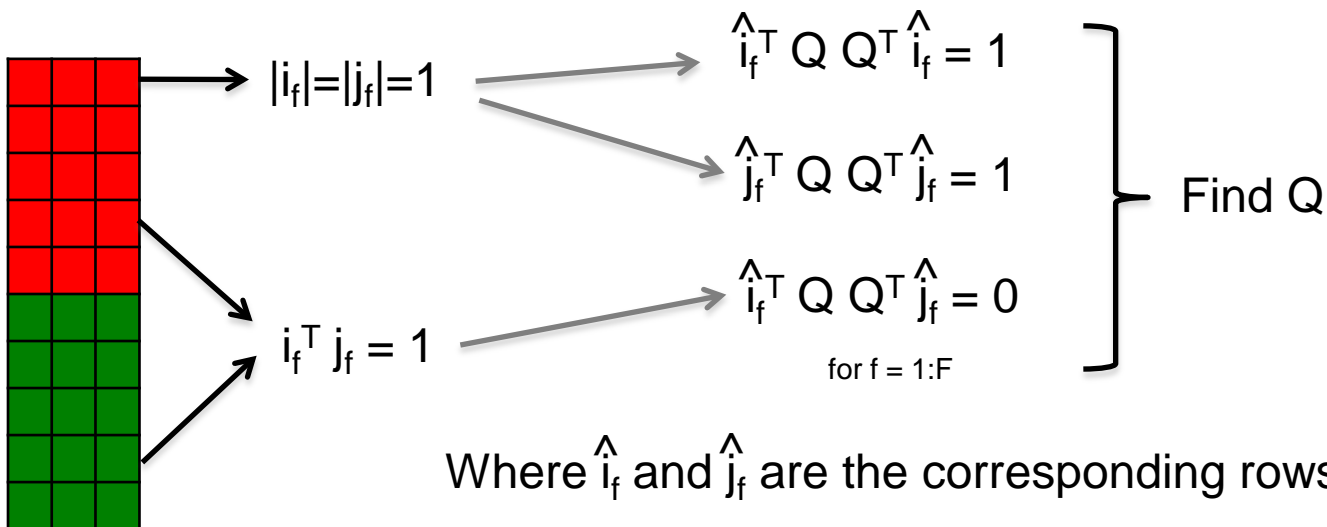
1. Use SVD to decompose  $\tilde{W}$  into rank 3 matrices
2. Impose constraints to find Q

$$\hat{R} = U' D'^{1/2} \quad \hat{S} = D'^{1/2} V'^T$$

We look for Q such that the final rotation and shape matrices are:

$$R = \hat{R}Q \quad S = Q^{-1}\hat{S}$$

We impose the constraints on the desired rotation matrix R:



# Solving for Q

**Algorithm 1:** Use non-linear solver and solve for Q

**Algorithm 2:** Solve linear system for C where  $C = QQ^T$

$$\hat{i}_f^T C \hat{i}_f = 1 \quad \text{for } f = 1:F$$

$$\hat{j}_f^T C \hat{j}_f = 1$$

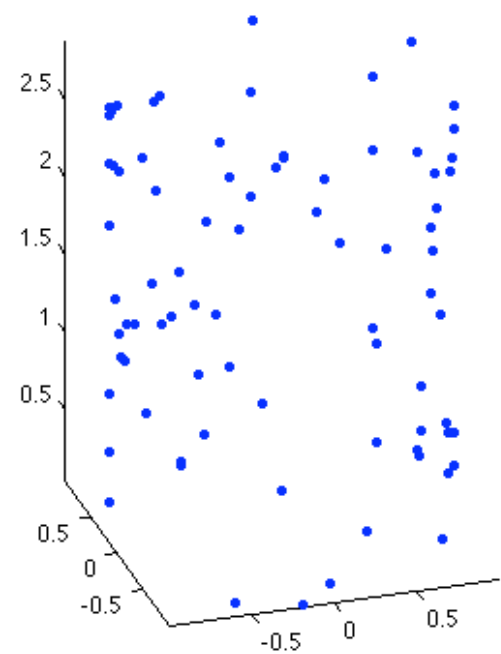
$$\hat{i}_f^T C \hat{j}_f = 0$$

Then, use Cholesky to factor C

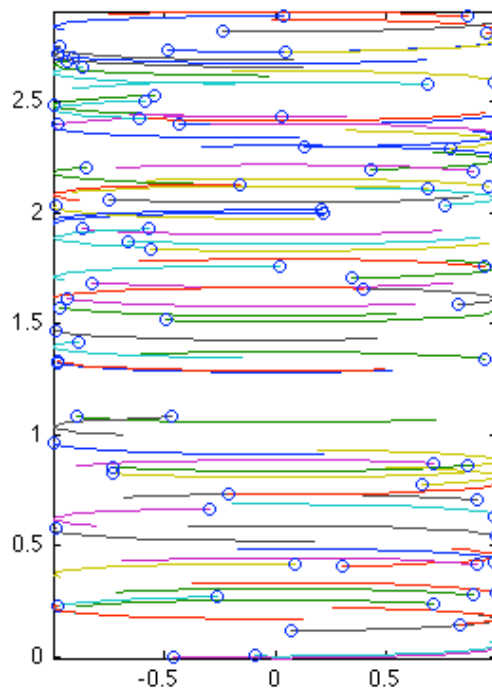
$$C = QQ^T$$

# Example: cylinder

3D data

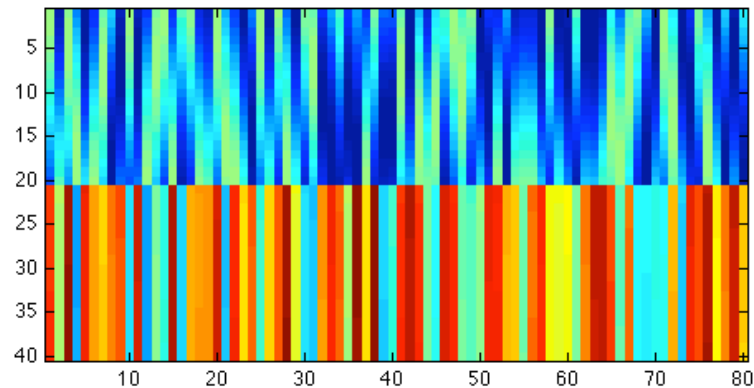


Observations:  
Image tracks

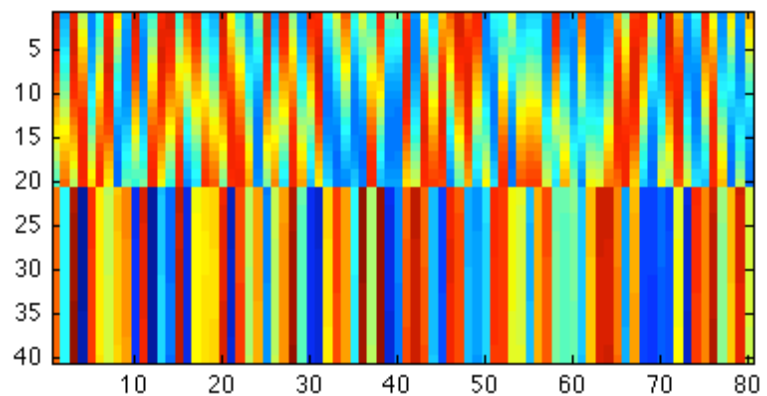


20 frames  
80 tracks

Measurement matrix  $W$



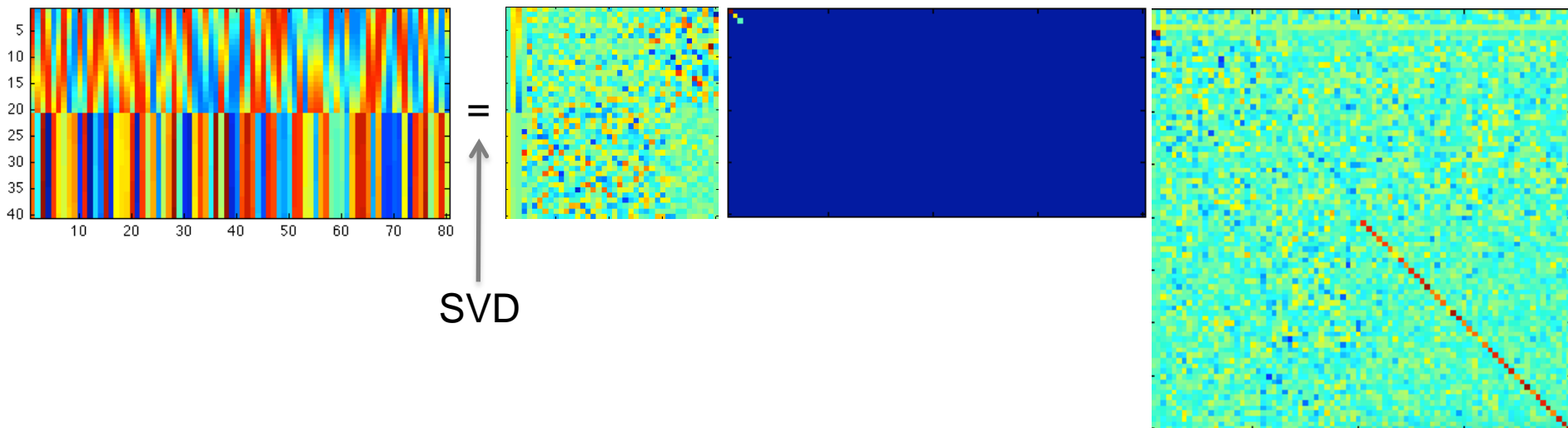
Registered measurement matrix  $\tilde{W}$





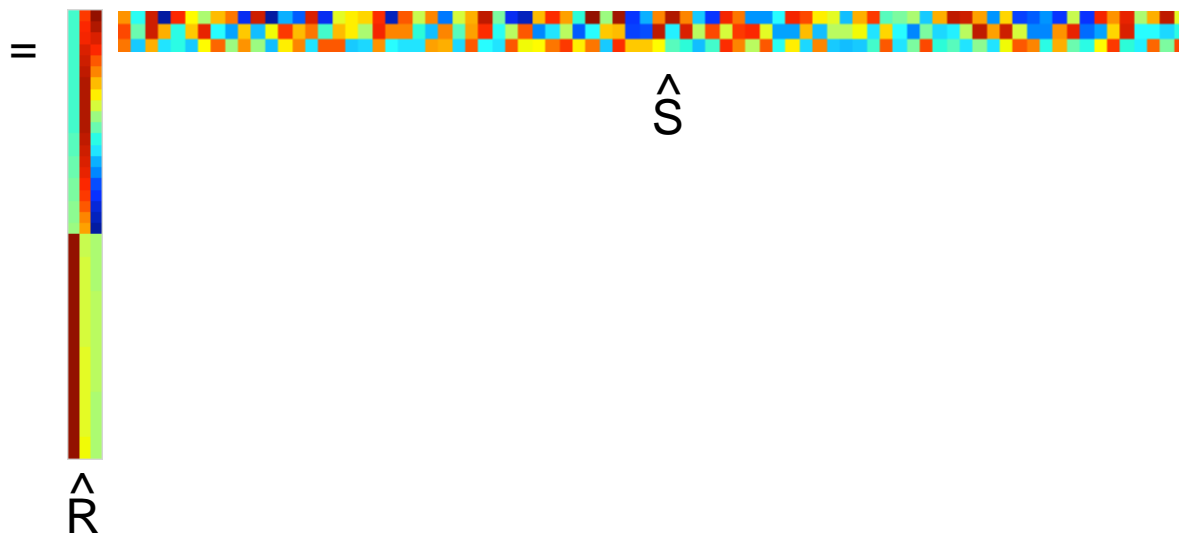
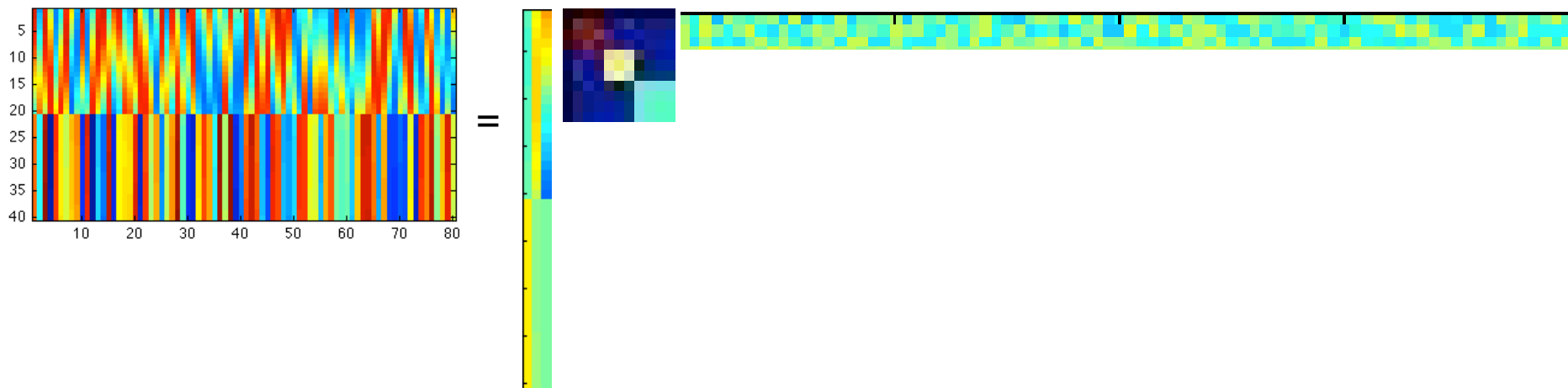
# Example: cylinder

Registered measurement matrix  $\tilde{W}$



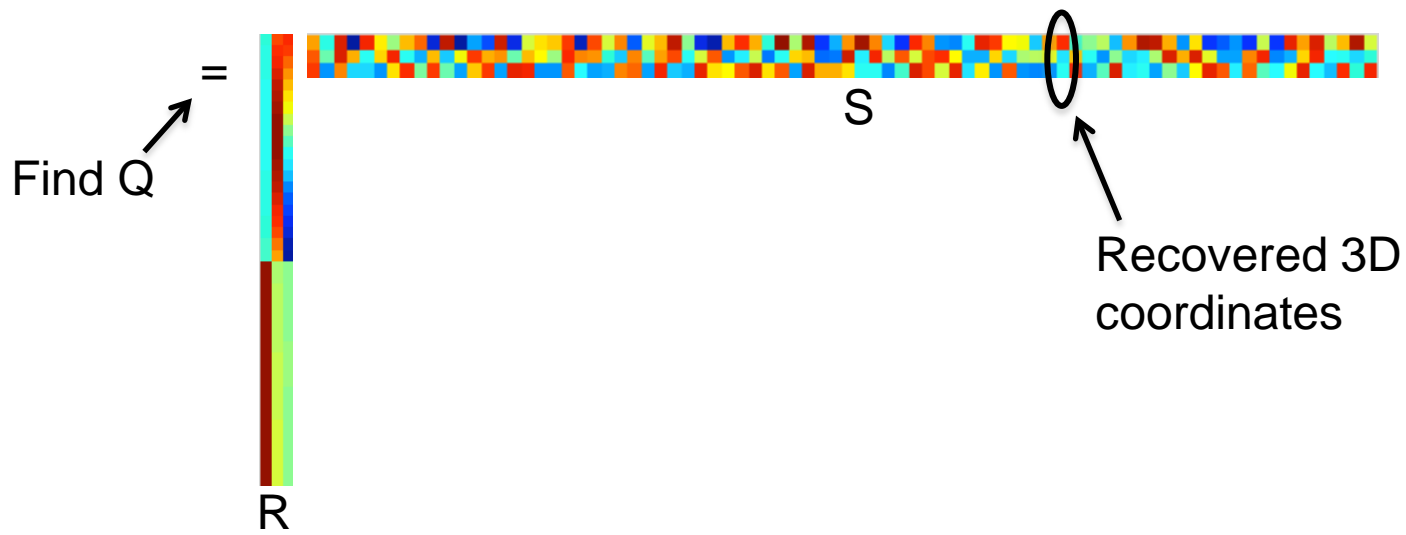
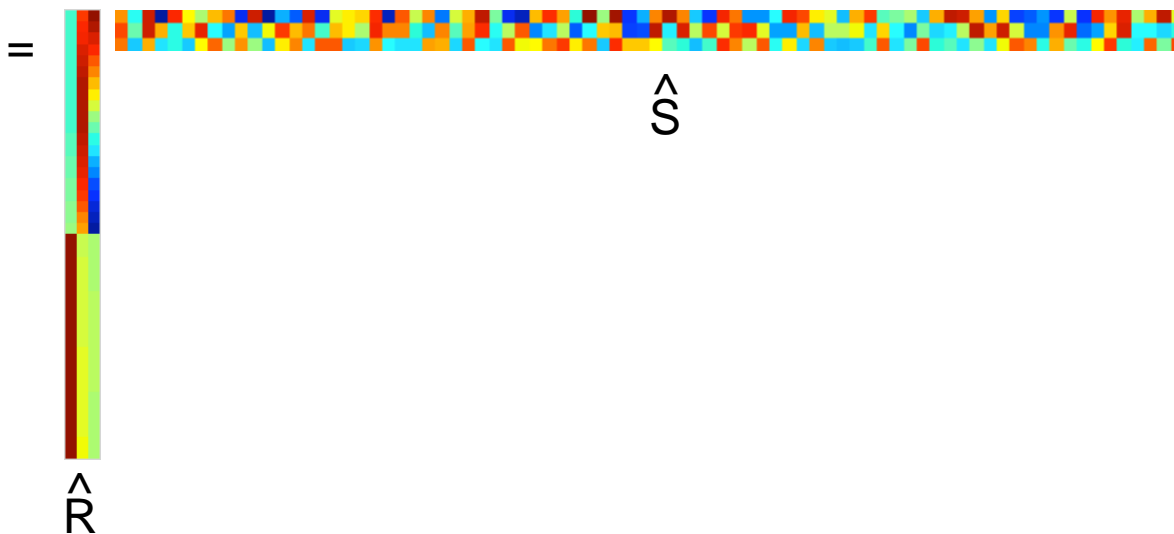
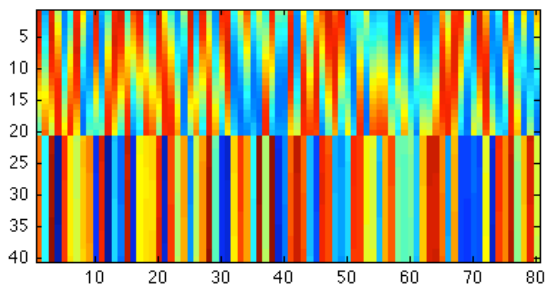
# Example: cylinder

Registered measurement matrix  $\tilde{W}$



# Example: cylinder

Registered measurement matrix  $\tilde{W}$



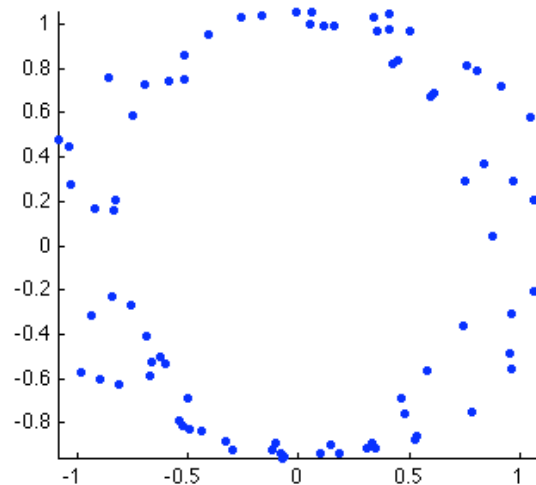
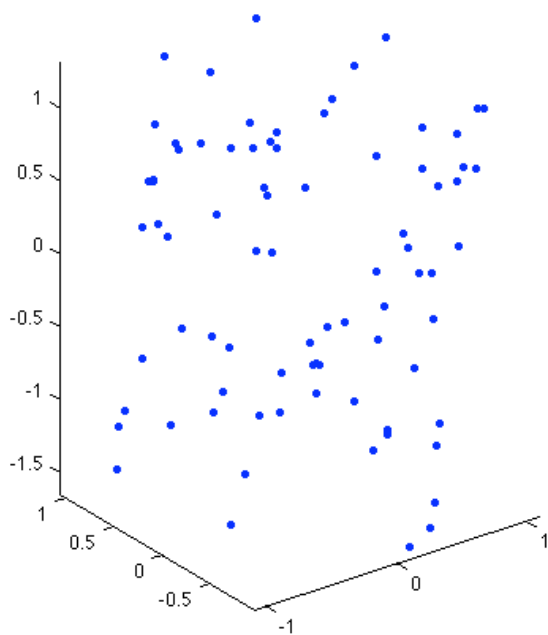
# Example: cylinder



S

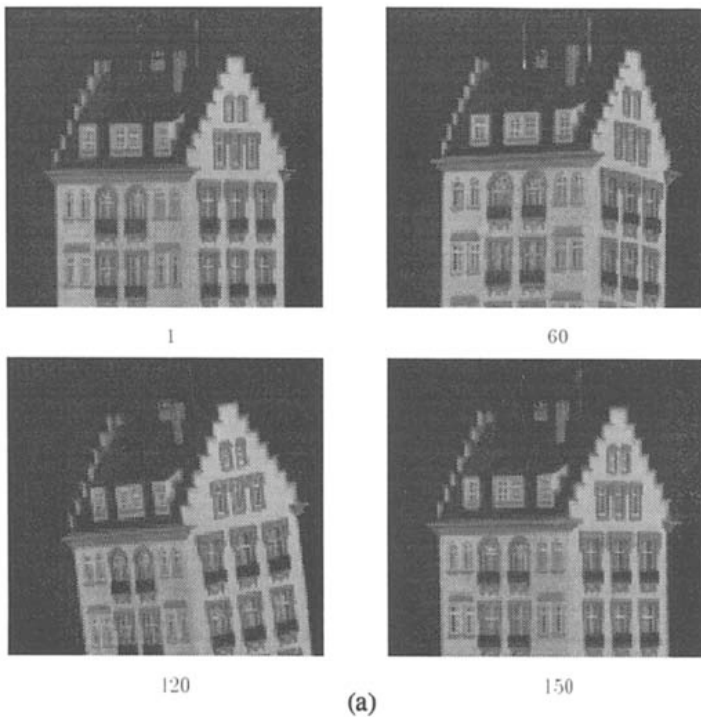


Recovered 3D  
coordinates

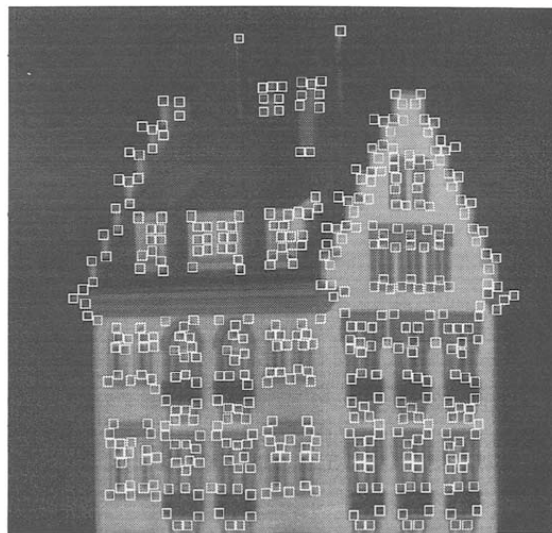


# Example

Input frames



Tracked points



Reconstructed top view



Fig. 2a. The "Hotel" stream: four of the 150 frames.

# Dealing with missing data

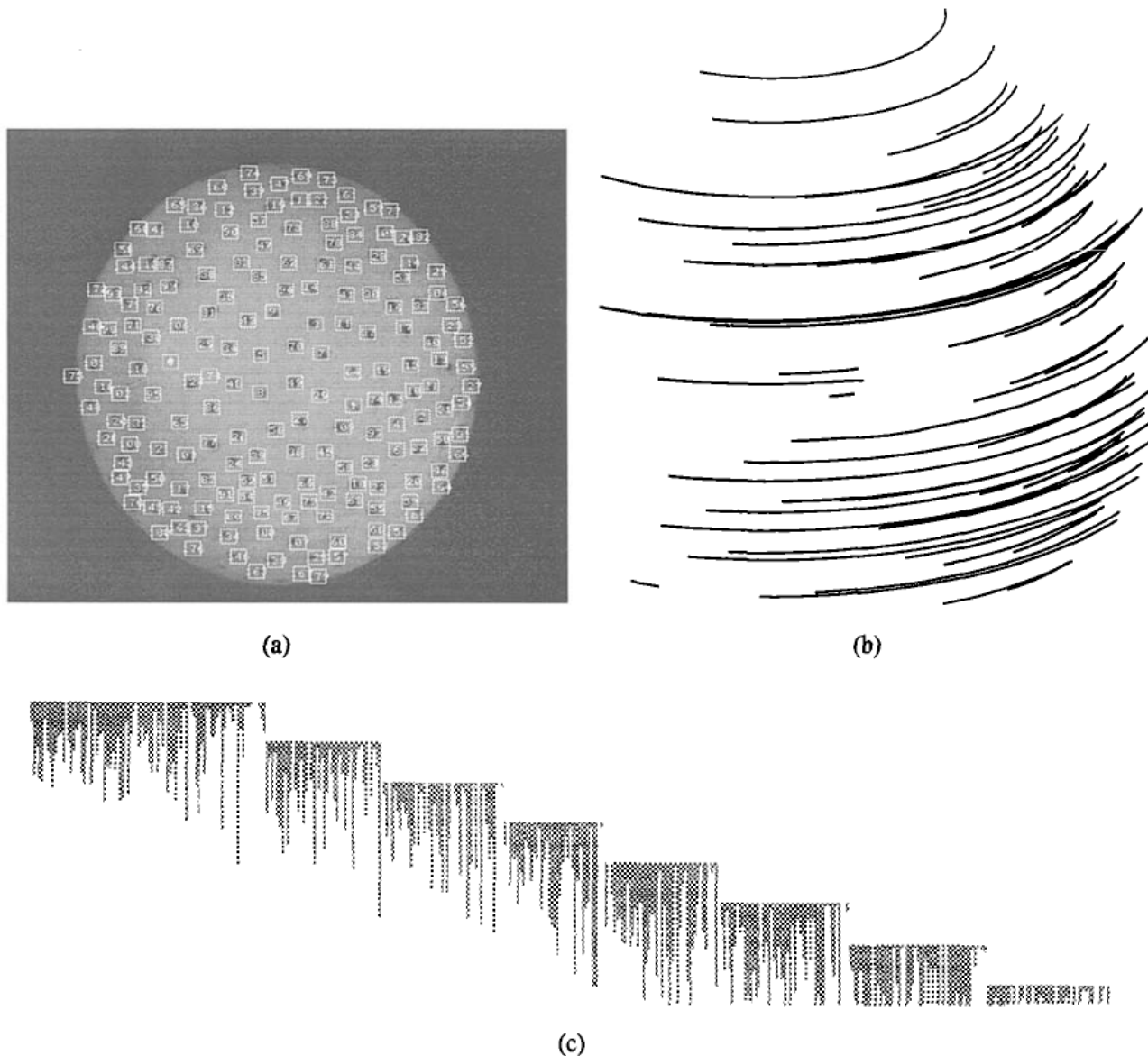


Fig. 9. The "Ball" stream : (a) the first frame, (b) tracks of 60 features, and (c) the fill matrix (shaded entries are known image coordinates).

# Dealing with missing data

Point  $p$  ( $u_{fp}, v_{fp}$ ) is missing in frame  $f$ .

Conditions for reconstruction:

- Point  $p$  is visible in three more frames  $f_1, f_2, f_3$
- And there are three more points visible in the four frames  $f, f_1, f_2, f_3$

Then we can recover the full shape matrix and the missing coordinates.

# Dealing with missing data

$$\mathbf{W} = \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ u_{41} & u_{42} & u_{43} & ? \\ v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & ? \end{bmatrix}$$

$$\mathbf{W}_{6 \times 4} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \end{bmatrix}$$

We can get the full shape matrix

$$\mathbf{W}_{8 \times 3} = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \\ u_{41} & u_{42} & u_{43} \\ v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \\ v_{41} & v_{42} & v_{43} \end{bmatrix}$$

We can get the full translation and rotation matrix





1



100

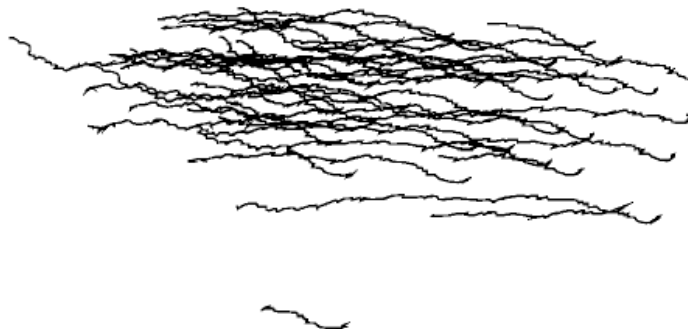


150

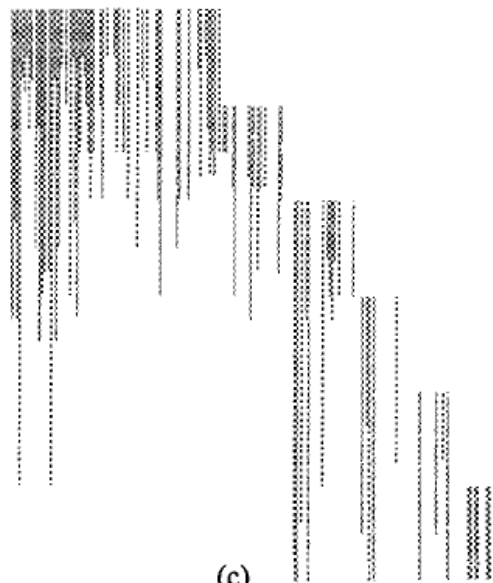


210

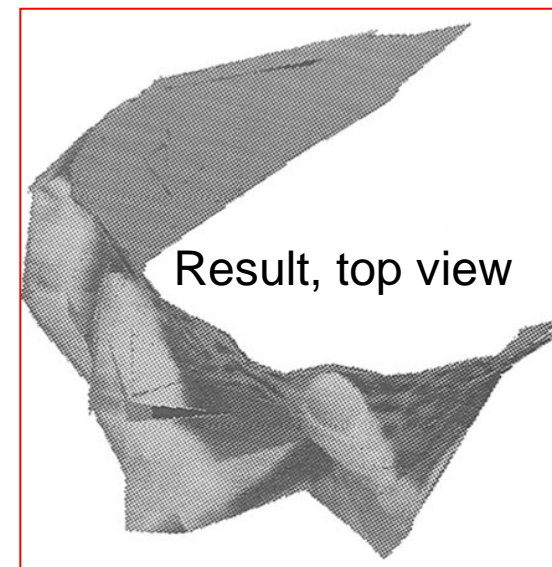
(a)



(b)



(c)



Result, top view

Fig. 12. The "Hand" stream: (a) four of the 240 frames, (b) tracks of 60 features, and (c) the fill matrix (shaded entries are known image coordinates).

# Perspective

## Euclidean Reconstruction: from Paraperspective to Perspective \*

Stéphane Christy and Radu Horaud

GRAVIR-IMAG & INRIA Rhône-Alpes  
46, avenue Félix Viallet 38031 Grenoble FRANCE

ECCV'96, volume II, pages 129–140

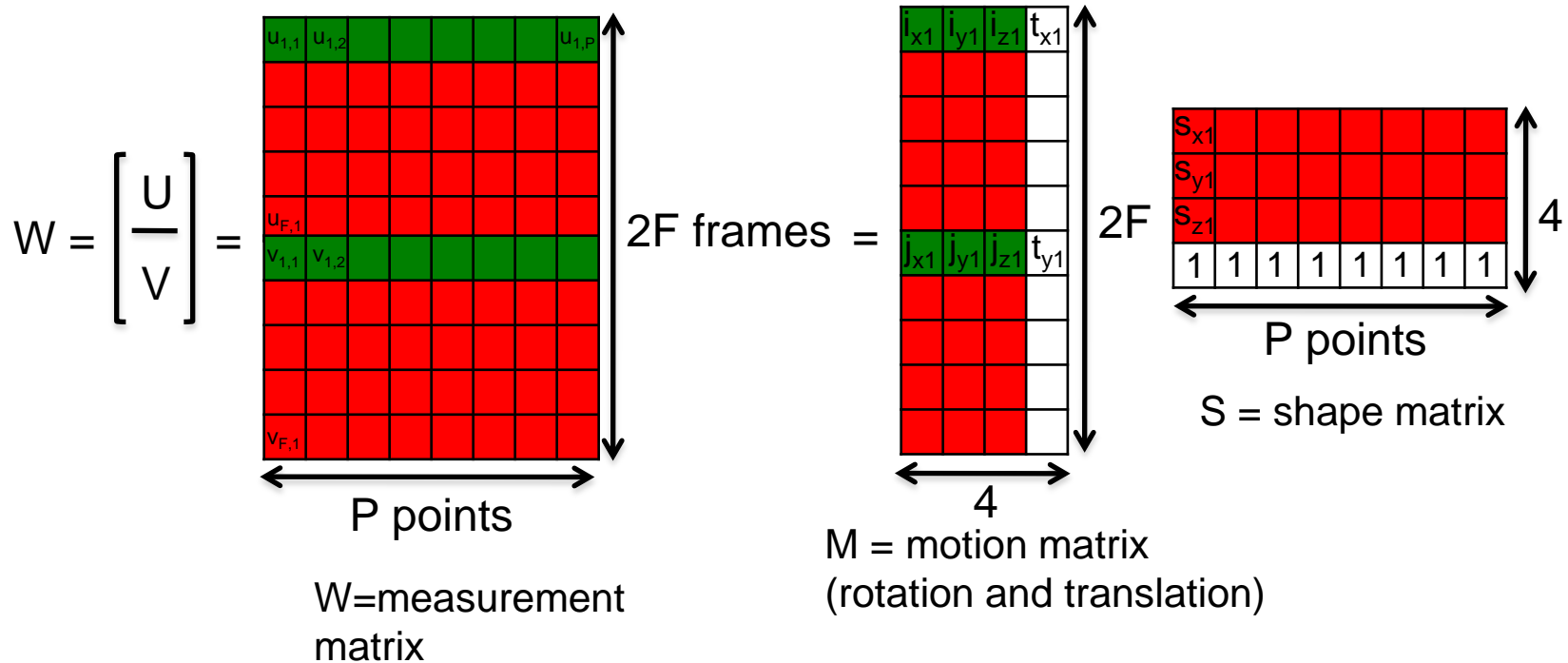
## **A factorization based algorithm for multi-image projective structure and motion**

PETER STURM AND BILL TRIGGS

to appear at ECCV'96, version of 16/01/96

# Factorization with translation

Let's not register de matrix  $W$  (do not remove the mean to each row)



$$W = M S$$

$$\text{Rank}(W) \leq 4$$

# Factorization with translation

Similar algorithm as before

1. Use SVD to decompose  $W$  into rank 4 matrices
2. Impose constraints to find  $Q$  (now add translation constraints)

$$W = U D V^T$$

$\hat{M}$  and  $\hat{S}$  are one of the possible decompositions.

$$\hat{M} = U' D'^{1/2}$$

$$\hat{S} = D'^{1/2} V'^T$$

We need to add constraints to  $M$ .

# Factorization with translation

Similar algorithm as before

1. Use SVD to decompose  $W$  into rank 4 matrices
2. Impose constraints to find  $Q$  (now add translation constraints)

$$W = MS = \hat{M}QQ^{-1}\hat{S} = (\hat{M}Q)(Q^{-1}\hat{S})$$

$$M = \hat{M}Q \quad S = Q^{-1}\hat{S}$$

$$Q = [Q_R \mid Q_t]$$

Constraints:

$$\hat{i}_f^T Q_R Q_R^T \hat{i}_f = 1$$

$$\hat{j}_f^T Q_R Q_R^T \hat{j}_f = 1$$

$$\hat{i}_f^T Q_R Q_R^T \hat{j}_f = 0$$

for  $f = 1:F$

} Find  $Q_R$

$$Q_t = D^{-1/2} U^T w$$

$w$  is the mean of the rows of  $W$

# Multi-body factorization

*Proc. R. Soc. Lond. B.* **203**, 405–426 (1979)

*Printed in Great Britain*

## The interpretation of structure from motion

BY S. ULLMAN

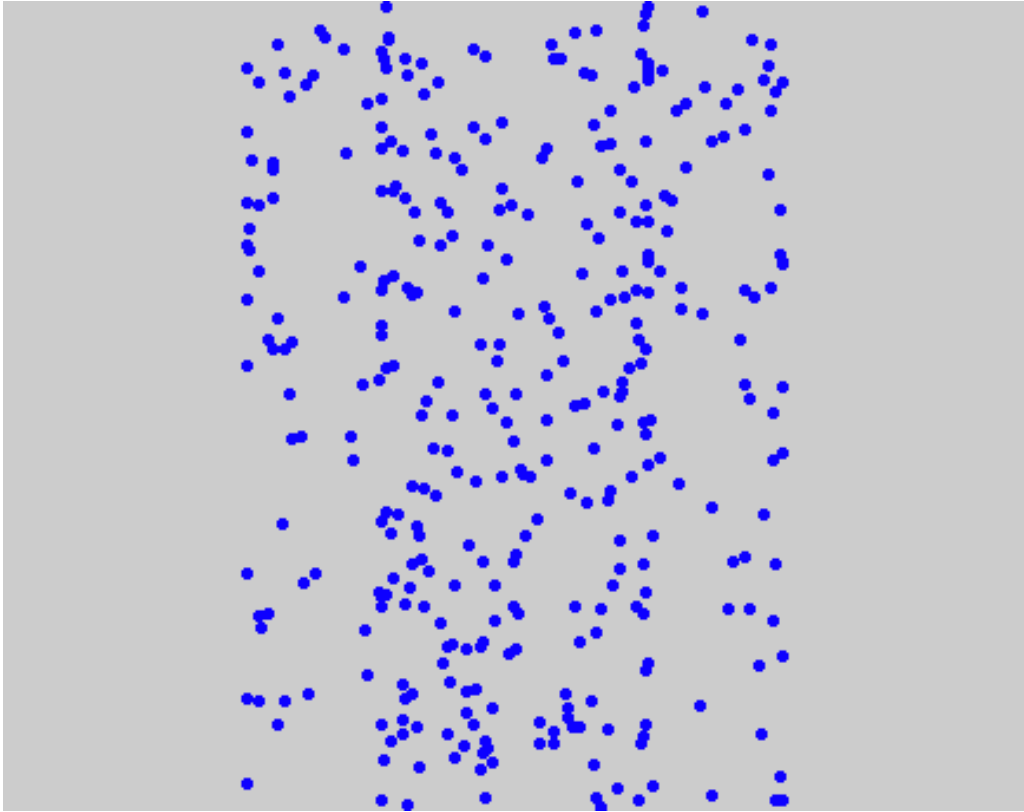
*Artificial Intelligence Laboratory, Massachusetts Institute of Technology,  
545 Technology Square (Room 808), Cambridge, Massachusetts 02139 U.S.A.*

*(Communicated by S. Brenner, F.R.S. – Received 20 April 1978)*

The interpretation of structure from motion is examined from a computational point of view. The question addressed is how the three dimensional structure and motion of objects can be inferred from the two dimensional transformations of their projected images when no three dimensional information is conveyed by the individual projections.

The following scheme is proposed: (i) divide the image into groups of four elements each; (ii) test each group for a rigid interpretation; (iii) combine the results obtained in (ii). It is shown that this scheme will correctly decompose scenes containing arbitrary rigid objects in motion, recovering their three dimensional structure and motion. The analysis is based primarily on the 'structure from motion' theorem which states that the structure of four non-coplanar points is recoverable from three orthographic projections. The interpretation scheme is extended to cover perspective projections, and its psychological relevance is discussed.





# Multi-body factorization

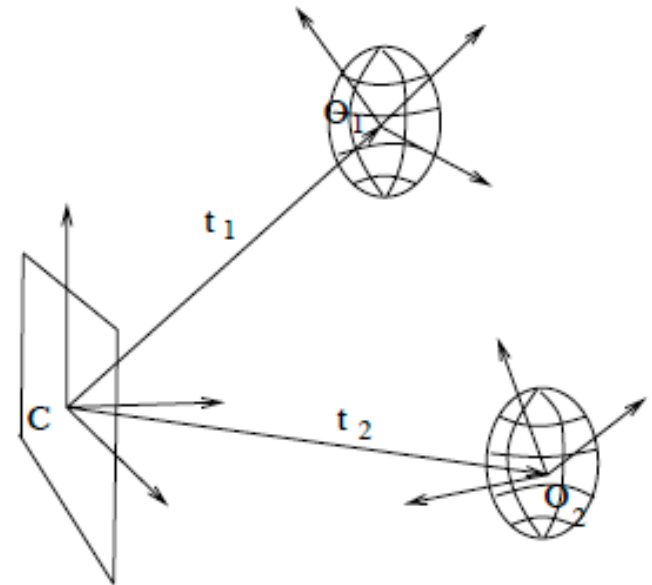
## A Multi-body Factorization Method for Motion Analysis

João Costeira      Takeo Kanade

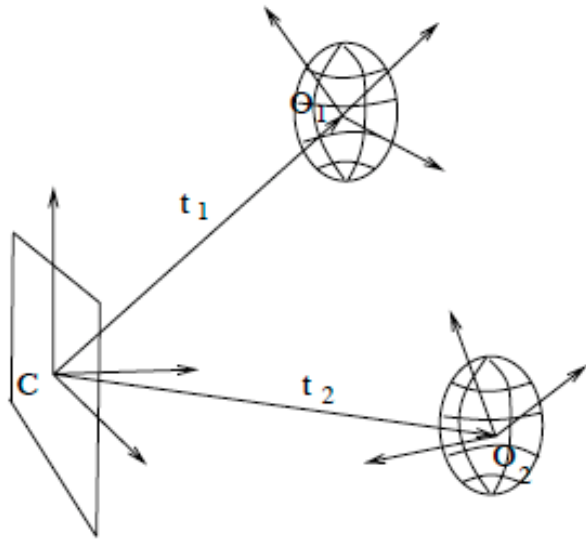
**CMU-CS-TR-94-220**

School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213

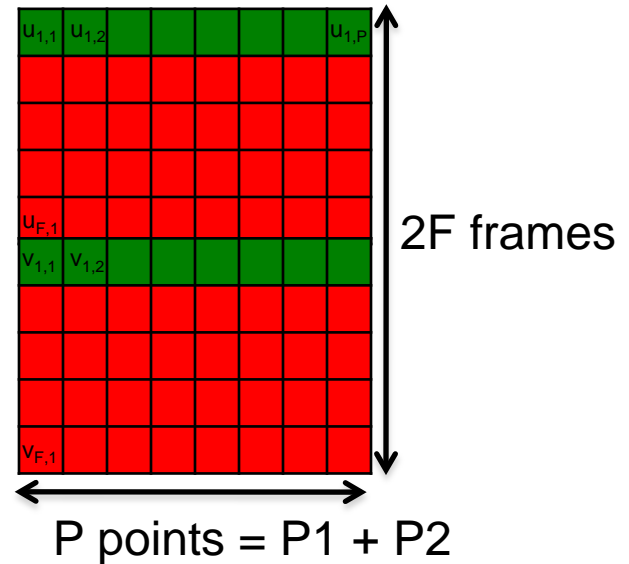
September 30, 1994



# Multi-body factorization



$$W = \begin{bmatrix} U \\ V \end{bmatrix} =$$



Let's assume we know which points belong to each object and we sort them so that:

$$W^* = [W_1 \mid W_2]$$

# Multi-body factorization

- Let's assume we know which points belong to each object and we sort them so that:

$$W^* = [W_1 \mid W_2]$$

- Each measurement matrix can be decomposed as:

Motion-shape factorization:  $W_1 = M_1 S_1 = (\hat{M}_1 A_1)(A_1^{-1} \hat{S}_1)$

SVD:  $W_1 = U_1 D_1 V_1^T$

- In this canonical form, the measurement matrix is:

$$W^* = [M_1 \mid M_2] \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix}$$

$W^*$  will have rank at most 8

# Multi-body factorization

Challenge: we do not know which features belong to each object.

Shape interaction matrix:

First, we compute SVD of the measurement matrix

$$W = UDV^T$$

We compute  $r = \text{rank}(W)$  and keep the first  $r$  rows of  $V^T$

$$Q = VV^T$$

# Multi-body factorization

## Shape interaction matrix

$$Q = VV^T$$

- Let's assume we know which points belong to each object and we sort them so that:

$$\begin{aligned} Q^* &= V^* V^{*T} \\ &= S^{*T} A^{*T} \Sigma^* A^* S^* \\ &= S^{*T} (A^{*-1} \Sigma^{*-1} A^{*-T})^{-1} S^* \\ &= S^{*T} \left[ (A^{*-1} \Sigma^{*-1/2} V^{*T}) (V^* \Sigma^{*-1/2} A^{*-T}) \right]^{-1} S^* = S^{*T} (S^* S^{*T})^{-1} S^* \\ &= \begin{bmatrix} S_1^T & 0 \\ 0 & S_2^T \end{bmatrix} \begin{bmatrix} \Lambda_1^{-1} & 0 \\ 0 & \Lambda_2^{-1} \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \\ &= \begin{bmatrix} S_1^T \Lambda_1^{-1} S_1 & 0 \\ 0 & S_2^T \Lambda_2^{-1} S_2 \end{bmatrix}. \end{aligned}$$

$Q^*$  has block structure

# Multi-body factorization

$Q^*$  has block structure

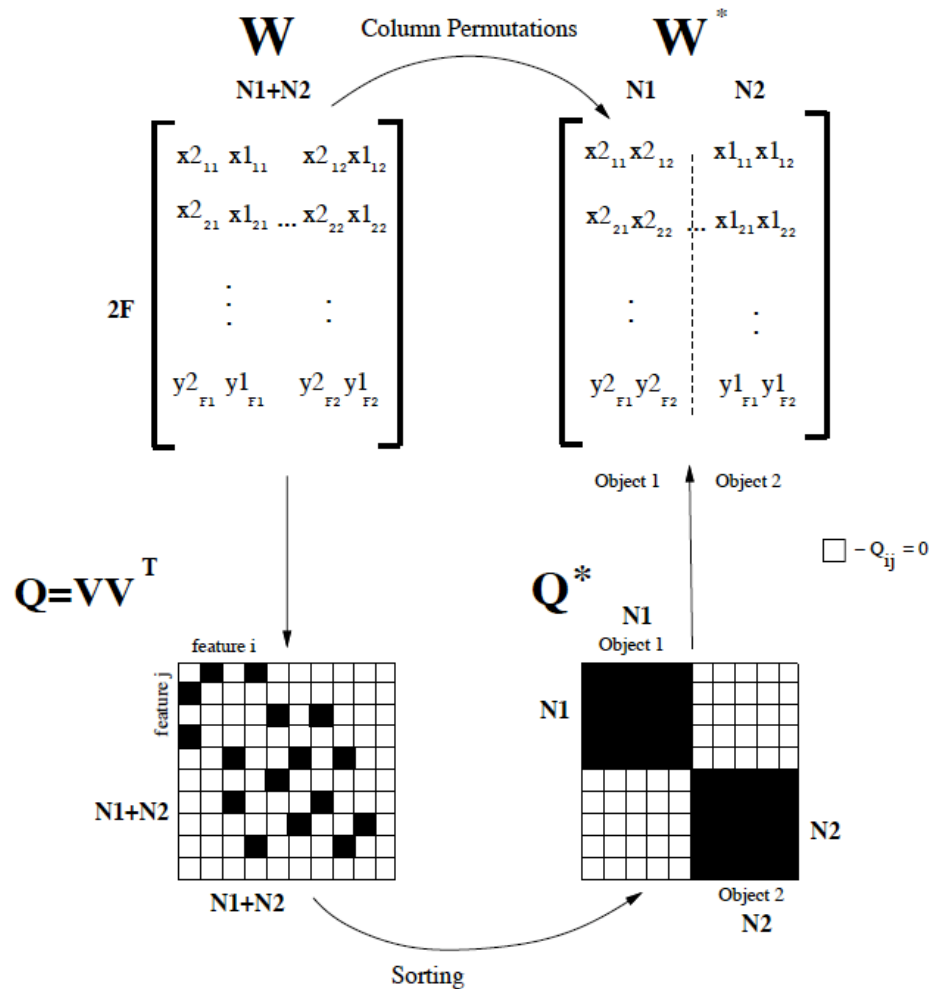
$$Q^* = \begin{bmatrix} \mathbf{S}_1^T \mathbf{\Lambda}_1^{-1} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2^T \mathbf{\Lambda}_2^{-1} \mathbf{S}_2 \end{bmatrix}$$

$$Q_{ij}^* = \begin{cases} \mathbf{s}_{1_i}^T \mathbf{\Lambda}_1^{-1} \mathbf{s}_{1_j} & \text{if feature trajectory } i \text{ and } j \text{ belong to object 1} \\ \mathbf{s}_{2_i}^T \mathbf{\Lambda}_2^{-1} \mathbf{s}_{2_j} & \text{if feature trajectory } i \text{ and } j \text{ belong to object 2} \\ 0 & \text{if feature trajectory } i \text{ and } j \text{ belong to different objects.} \end{cases}$$

# Multi-body factorization

If we do not know the segmentation  $Q$  will be unsorted.

We can swap rows and columns until it becomes block diagonal. Then, use same permutations to sort  $W$ .





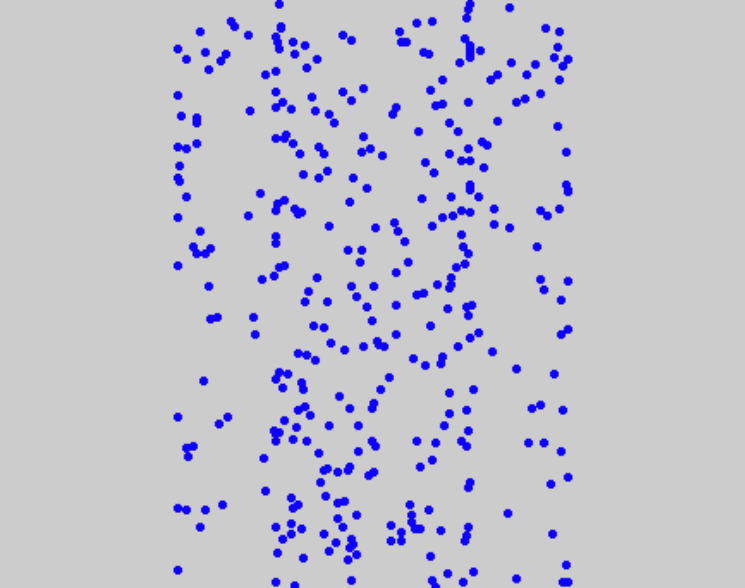
# Multi-body factorization

The whole algorithm of the multi-body factorization method is now summarized as:

1. Extract and track features in the input image sequence and create matrix  $\mathbf{W}$
2. Compute  $r = \text{rank}(\mathbf{W})$
3. Decompose matrix  $\mathbf{W}$  using SVD
4. Compute shape interaction matrix  $\mathbf{Q}$  using the first  $r$  rows of  $\mathbf{V}^T$
5. Block-diagonalize  $\mathbf{Q}$
6. Permute matrix  $\mathbf{V}^T$  into submatrices, each corresponding to a single object
7. Compute  $\mathbf{A}_i$  for each object, and thus its shape and motion.

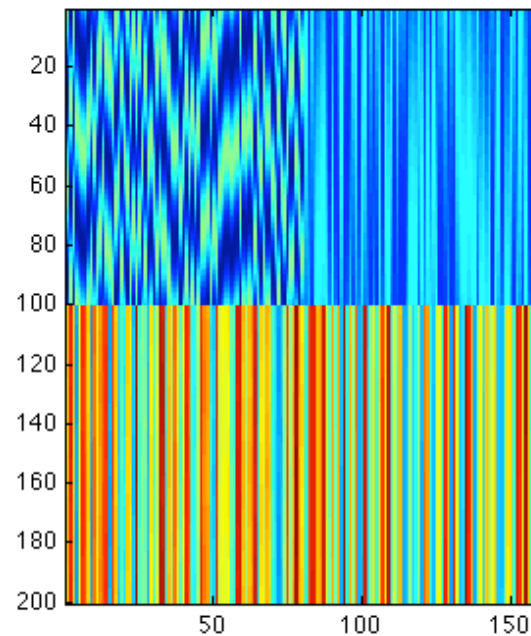
This clustering algorithm is related to spectral clustering.

See “Segmentation using eigenvectors: a unifying view”, Weiss, ICCV’99

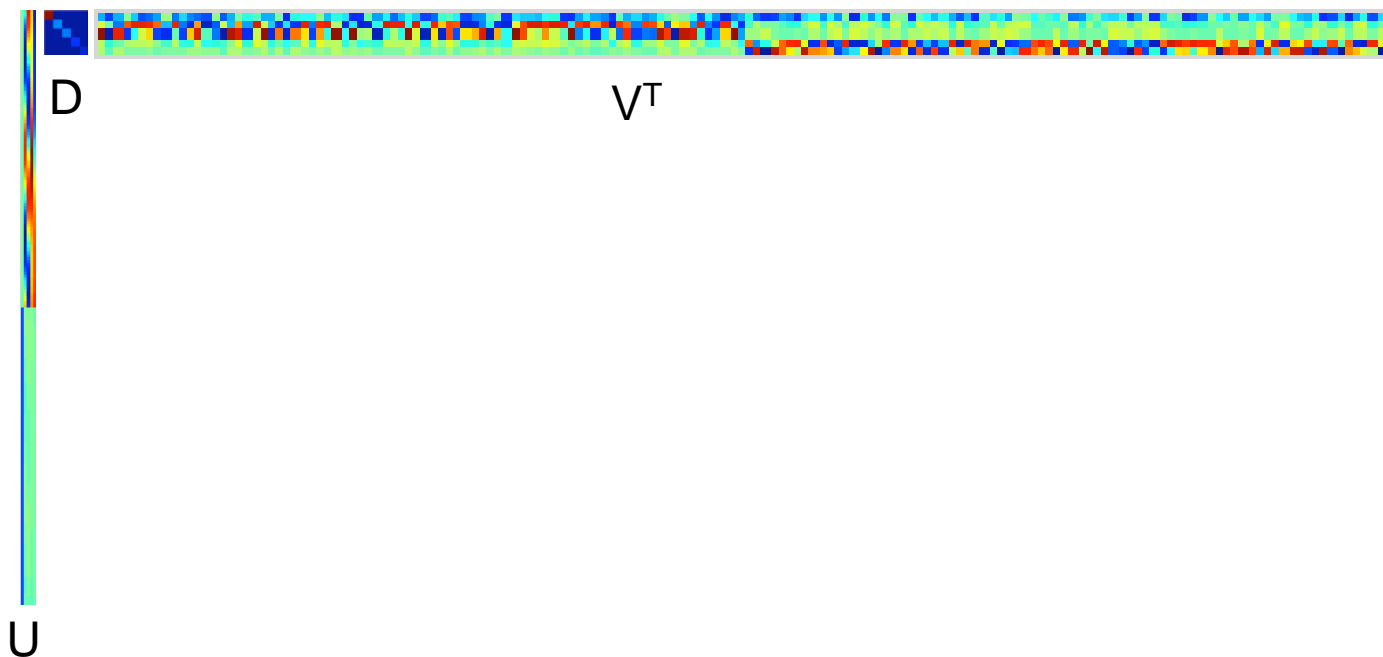


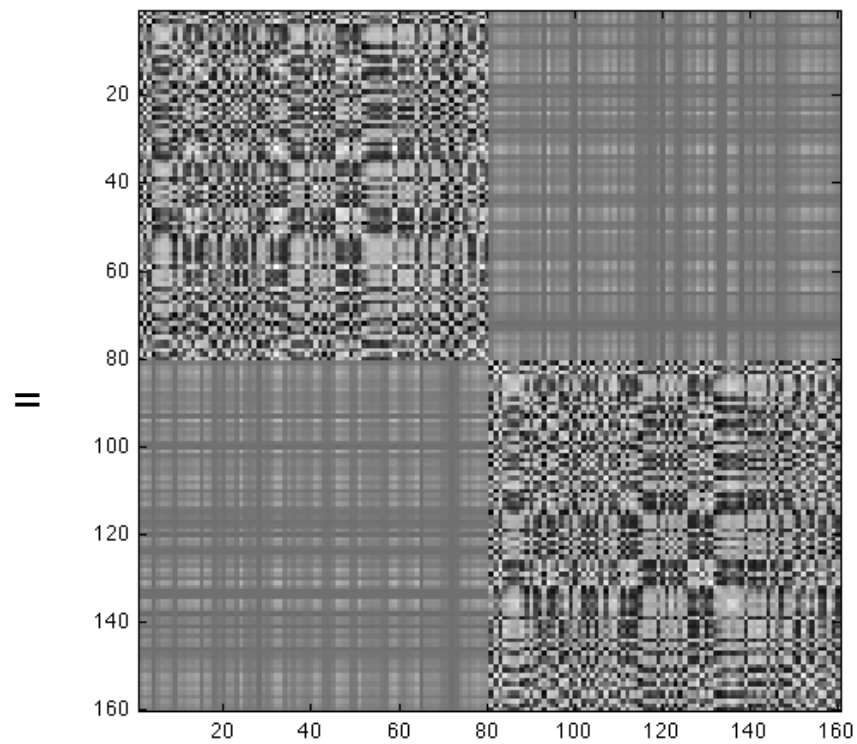
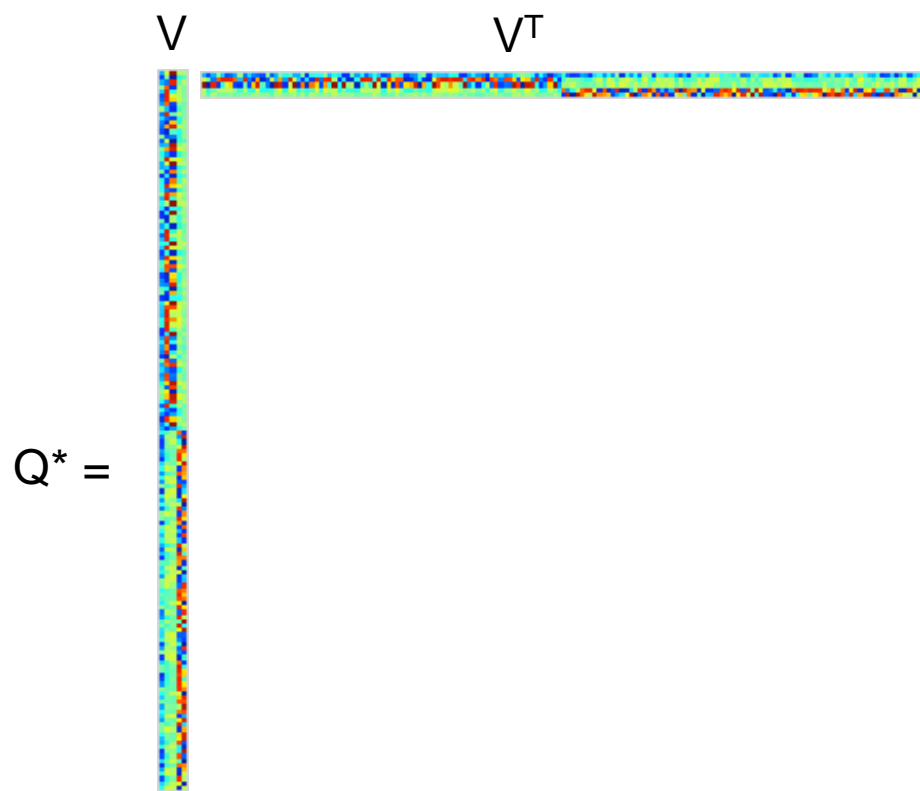
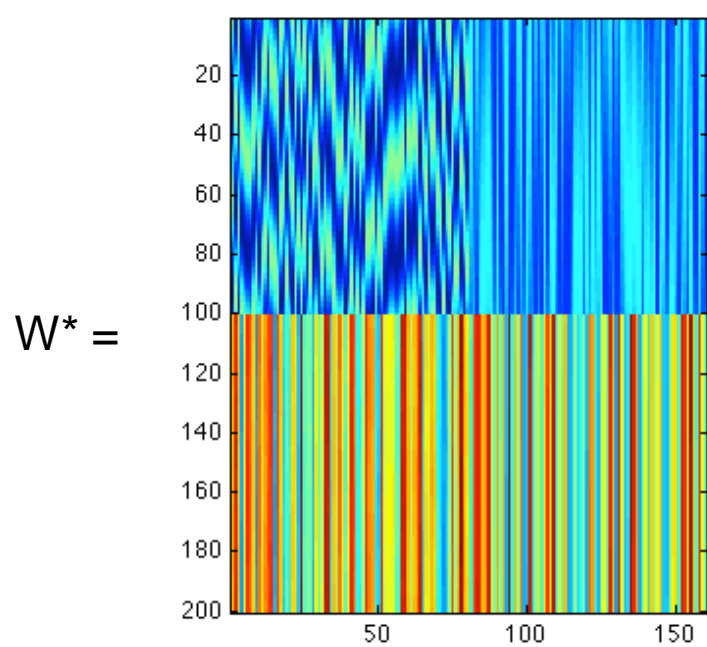
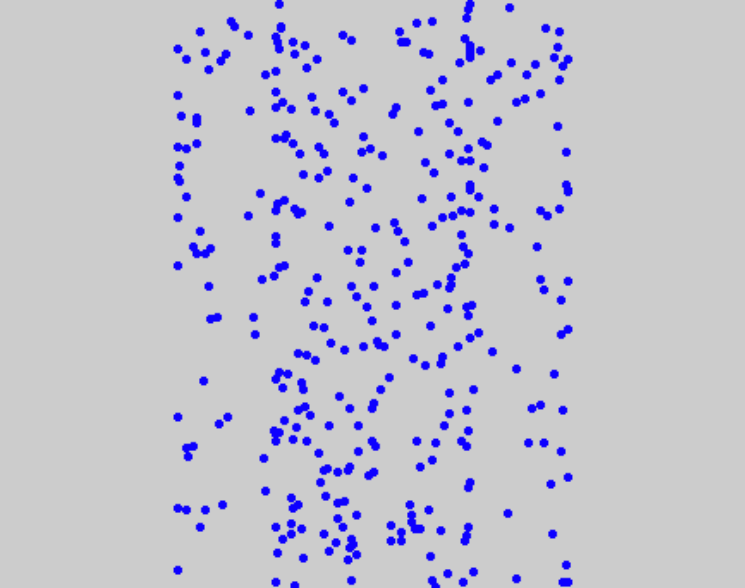
100 frames  
80 points per cylinder

$W^* =$

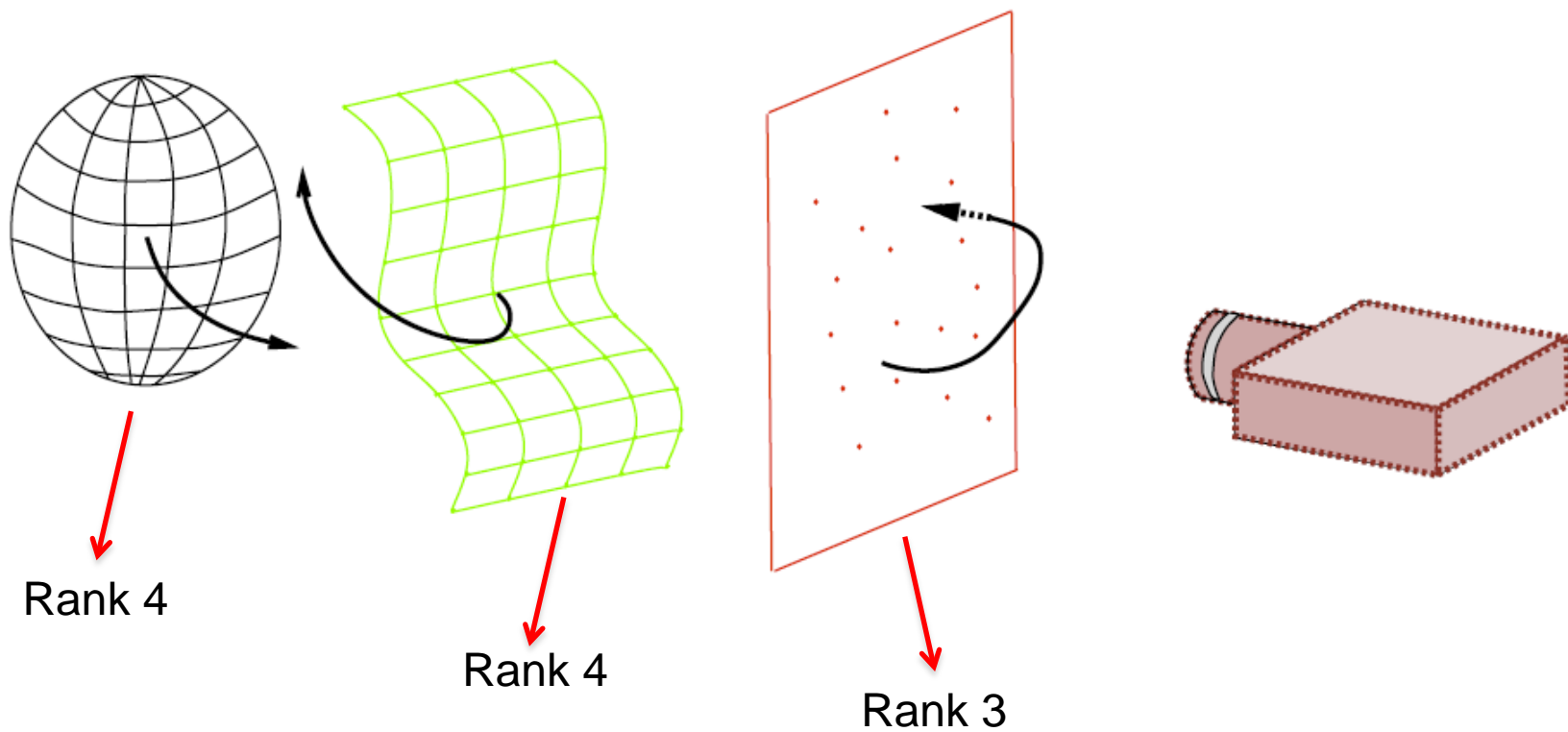


$W^* =$





# Example



# Example

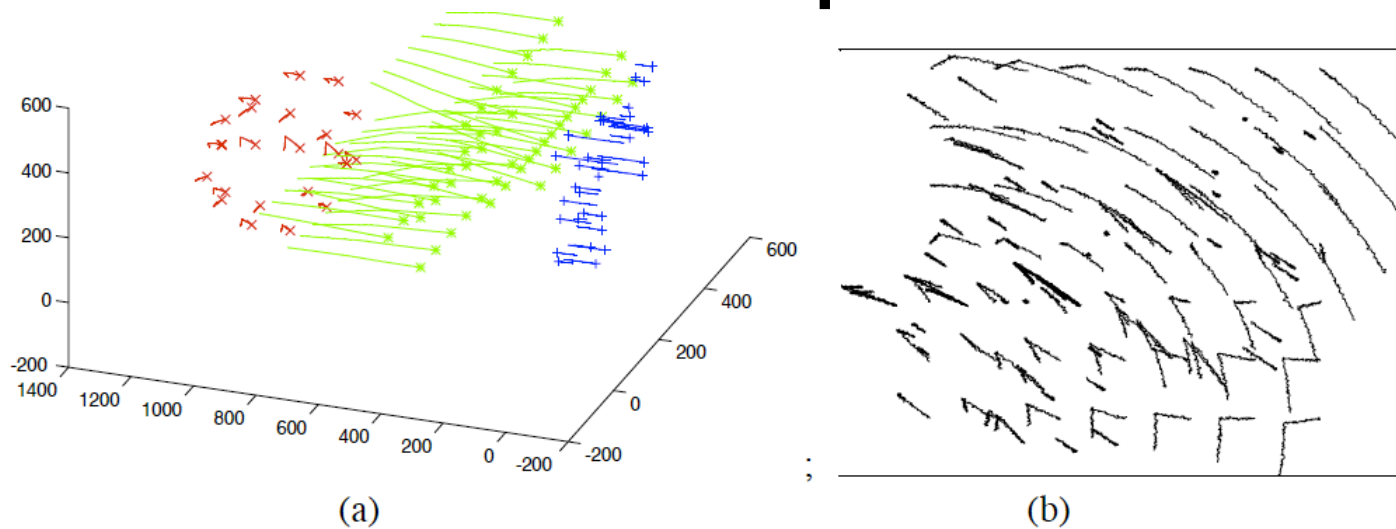


Figure 5: (a) 3D trajectories of the points and (b) noisy image tracks

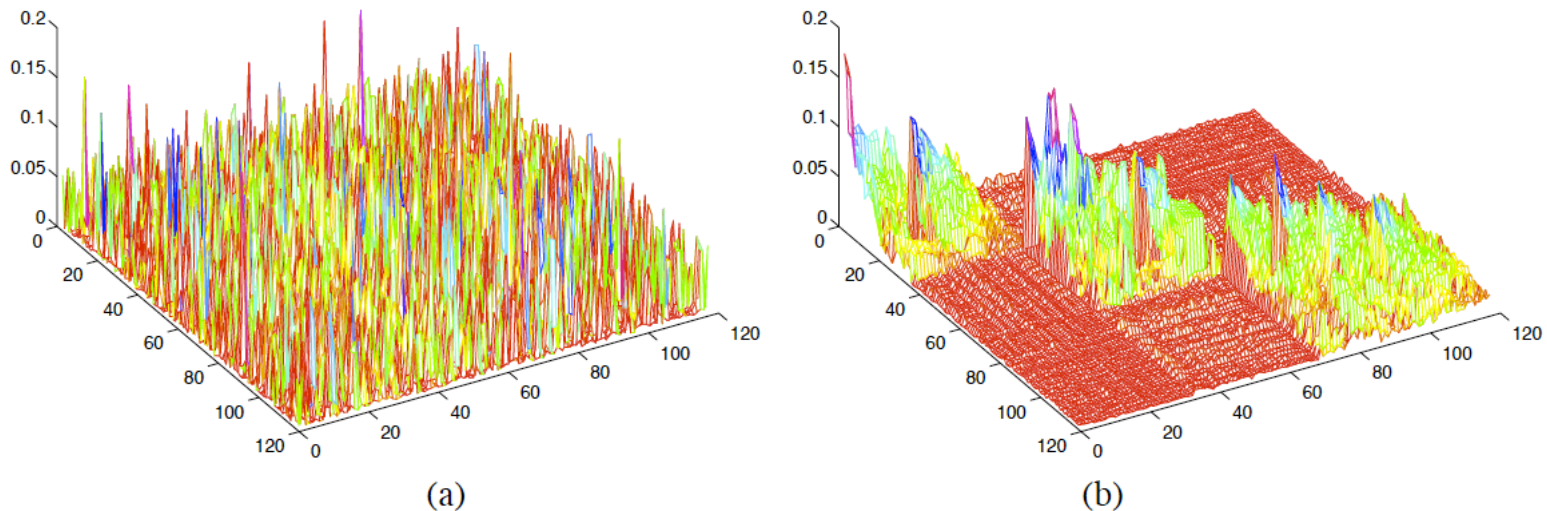
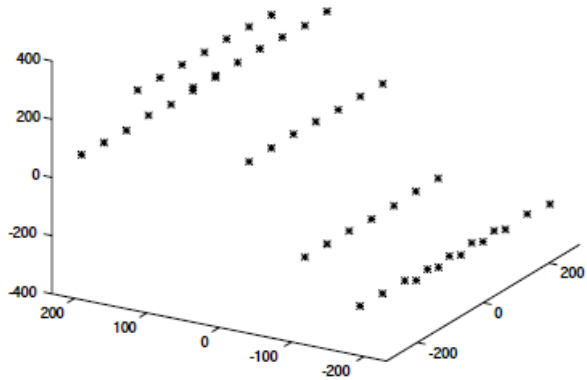
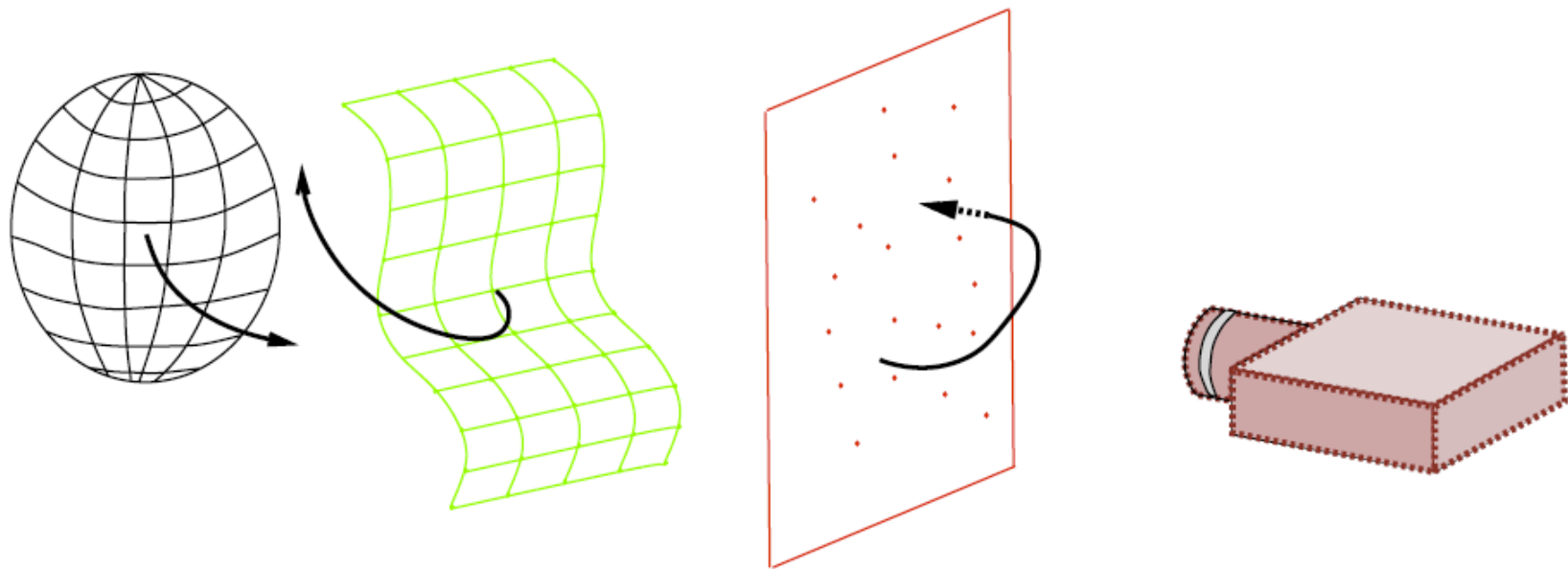
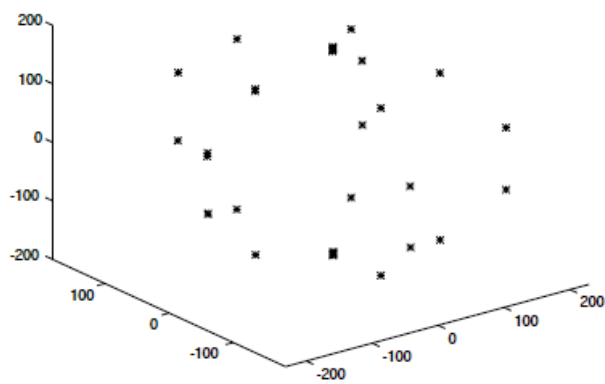


Figure 6: The shape interaction matrix for the synthetic scene with three transparent objects: (a) Unsorted matrix  $Q$ , and (b) sorted matrix  $Q^*$ .

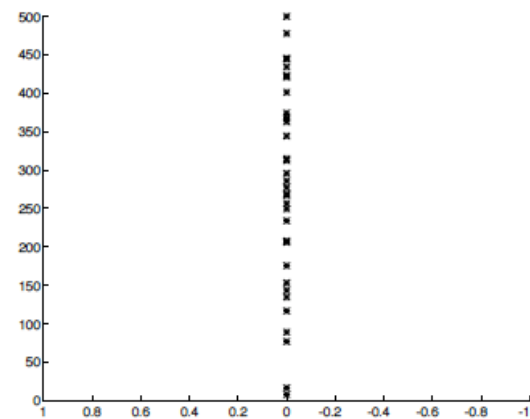
# Example



(a)



(b)



(c)

Figure 7: Recovered shape of the objects

# Multi-body factorization

## Pros:

- No dependency on the type of motions
- No need to know number of objects
- No need to know shapes

## Cons:

- Sensitivity to noise
- Need full trajectories. Dealing with occlusions will be delicate.

# Multibody Factorization with Uncertainty and Missing Data Using the EM Algorithm

Amit Gruber and Yair Weiss  
School of Computer Science and Engineering  
The Hebrew University of Jerusalem  
Jerusalem, Israel 91904

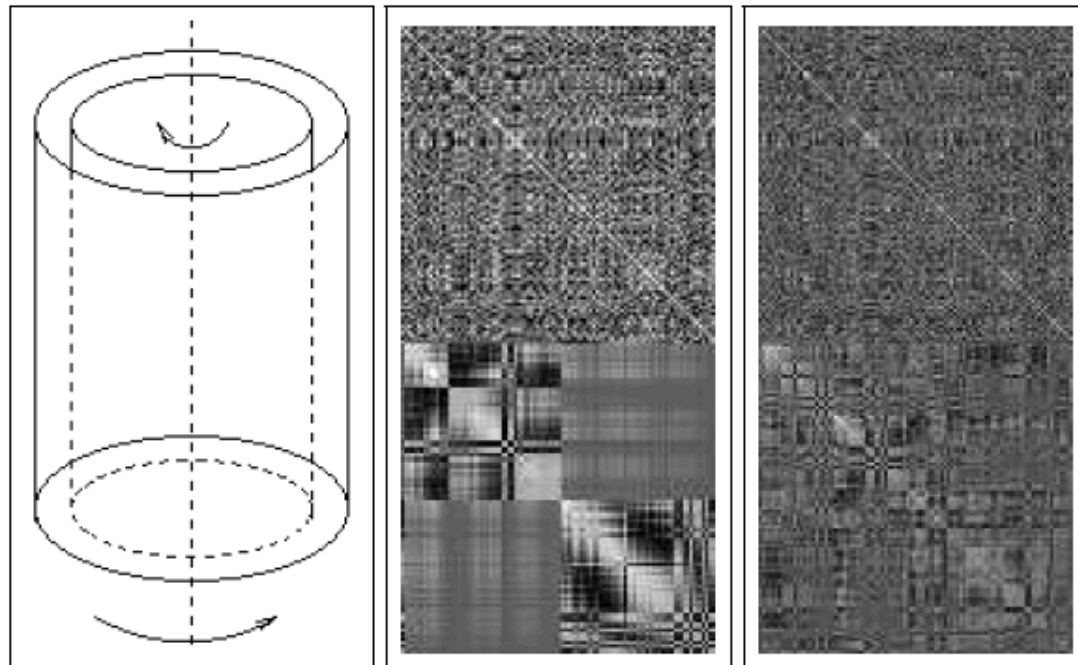


Figure 1: **a.** Ullman's [12] co-axial transparent cylinders demonstration. **b.** Costeira-Kanade [1] factorized matrix for noise free input. Top row is unsorted matrix, bottom is sorted. **c.** Costeira-Kanade factorized matrix for noisy input (unsorted on top row, sorted on bottom).



# Temporal Factorization Vs. Spatial Factorization

Lihi Zelnik-Manor<sup>1</sup> and Michal Irani<sup>2</sup>

<sup>1</sup> California Institute of Technology, Pasadena CA, USA,  
lihi@caltech.edu,

WWW home page: <http://www.vision.caltech.edu/lihi>

<sup>2</sup> Weizmann Institute of Science, Rehovot, Israel

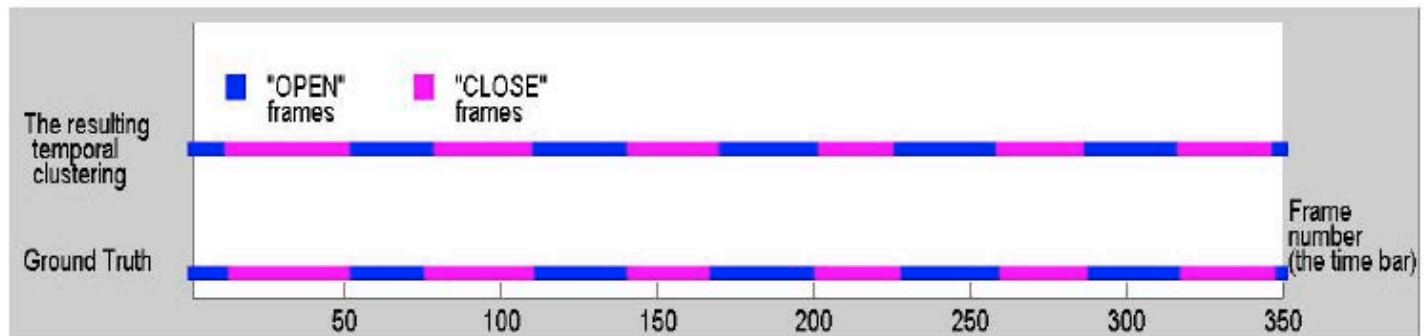
Given a video clip of a dynamic scene:

1. Track reliable feature points along the entire sequence.
2. Place each trajectory into a column vector and construct the correspondence matrix  $W = \begin{bmatrix} X \\ Y \end{bmatrix}$  (see Eq. (1))
3. Apply any of the existing algorithms for column clustering (e.g., “multi-body factorization” of [6, 7, 9]), but to the matrix  $W^T$  (instead of  $W$ ).

# Temporal factorization

	Spatial Factorization	Temporal Factorization
Apply clustering to	$W^T W$	$W W^T$
Data dimensionality	$N \times N$	$F \times F$
Data type	Points (columns)	Frames (rows)
Cluster by	Consistent motions	Consistent shapes

(a) Temporal factorization result:



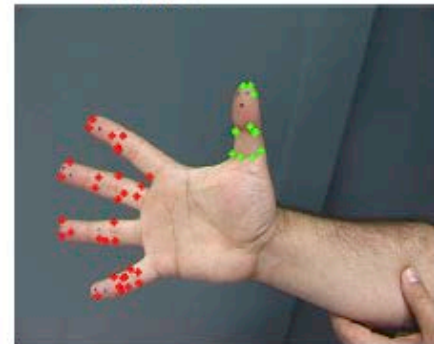
(b) Example frame from the "OPEN" cluster



(c) Example frame from the "CLOSE" cluster



(d) Spatial factorization result



# Non-Rigid Structure from Locally-Rigid Motion

Jonathan Taylor

Allan D. Jepson

Kiriakos N. Kutulakos

Department of Computer Science  
University of Toronto

The key idea is to first solve many local 3-point, N-view rigid problems independently, providing a "soup" of specific, plausibly rigid, 3D triangles. Triangles from this soup are then grouped into bodies, and their depth flips and instantaneous relative depths are determined.

The main advantage here is that the extraction of 3D triangles requires only very weak assumptions:

- (1) deformations can be locally approximated by near-rigid motion of three points (i.e., stretching not dominant) and
- (2) local motions involve some generic rotation in depth.

# Non-Rigid Structure By Locally-Rigid Motion

Jonathan Taylor, Allan Jepson, Kiriakos Kutulakos

CVPR 2010

# Bundle adjustment

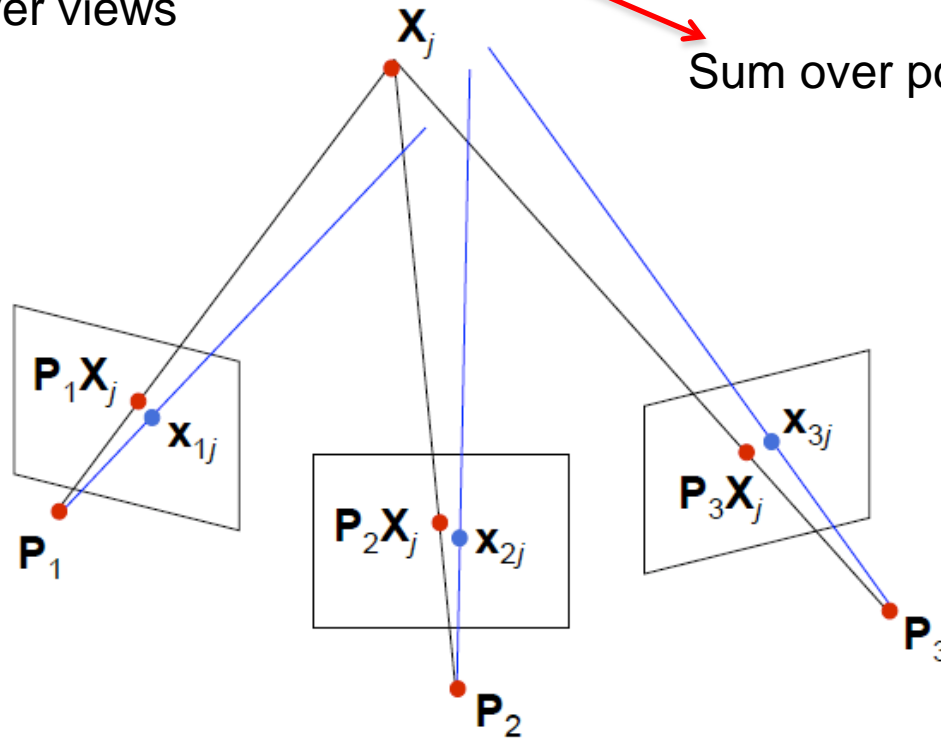
Recover structure and motion by minimization of measurement error.

Given the image coordinates  $x_{ij}$ , find the camera matrices  $P_i$  and the 3D locations of the points  $X_j$

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

Sum over views

Sum over points



# Bundle Adjustment in the Large

Sameer Agarwal<sup>1\*</sup>, Noah Snavely<sup>2</sup>, Steven M. Seitz<sup>3</sup>, and Richard Szeliski<sup>4</sup>



# Complex motion





## Dynamic Textures

GIANFRANCO DORETTO

*Computer Science Department, University of California, Los Angeles, CA 90095*

doretto@cs.ucla.edu

ALESSANDRO CHIUSO

*Dipartimento di Ingegneria dell' Informazione, Università di Padova, Italy 35131*

chiuso@dei.unipd.it

YING NIAN WU

*Statistics Department, University of California, Los Angeles, CA 90095*

ywu@stat.ucla.edu

STEFANO SOATTO

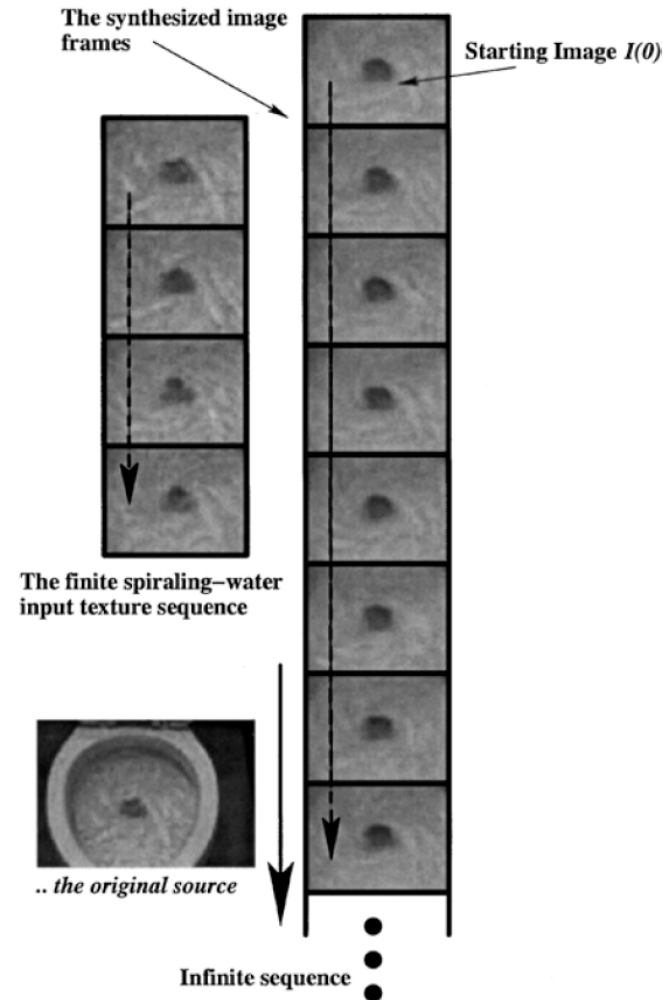
*Computer Science Department, University of California, Los Angeles, CA 90095*

soatto@ucla.edu

*Received May 1, 2001; Revised February 21, 2002; Accepted July 3, 2002*

$$\begin{cases} x(t+1) = Ax(t) + v(t) & v(t) \sim \mathcal{N}(0, Q); & x(0) = x_0 \\ y(t) = Cx(t) + w(t) & w(t) \sim \mathcal{N}(0, R) \end{cases}$$

image







Copyright (c) UCLA, G. Doretto and S. Soatto, 2002

Original

Synthesized

Copyright (c) UCLA, G. Doretto and S. Soatto, 2002



Original

Synthesized



Copyright (c) UCLA, G. Doretto and S. Soatto, 2002

Original

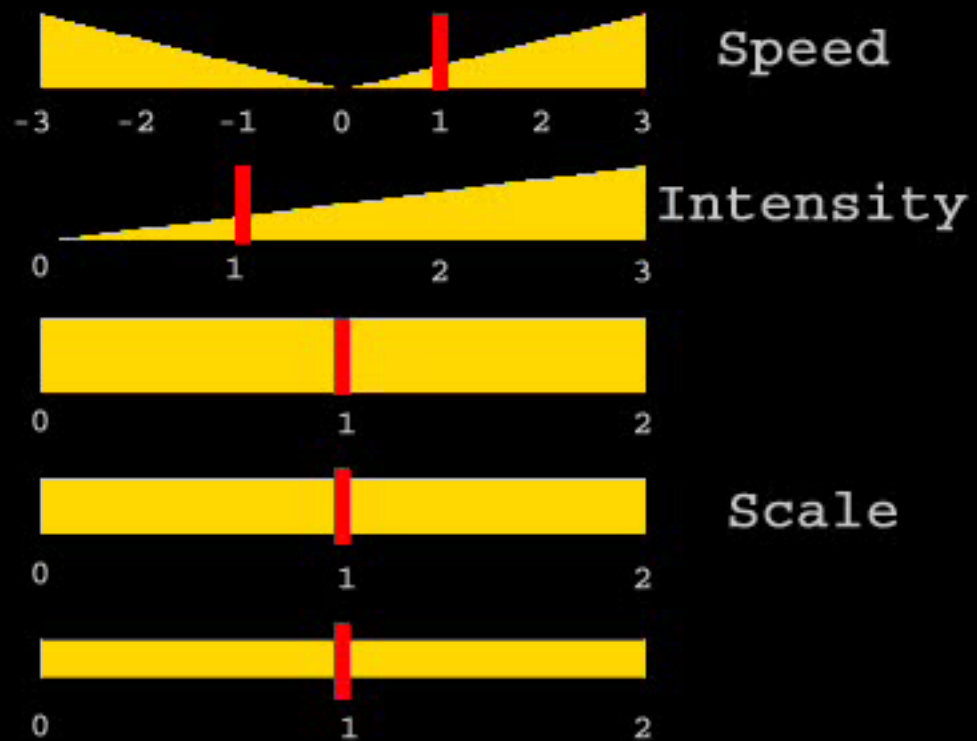
Synthesized

The training sequence is full of highlights and this makes the learning procedure much more difficult.

Copyright (c) UCLA, G. Doretto and S. Soatto, 2002



Copyright (c) UCLA, G. Doretto and S. Soatto, 2002



Copyright (c) UCLA, G. Doretto and S. Soatto, 2002

