Two-view geometry (cont’d)

Multi-view geometry



Three questions:

(i) Correspondence geometry: Given an image point x in the first view, how does this constrain the position of the corresponding point x' in the second image?

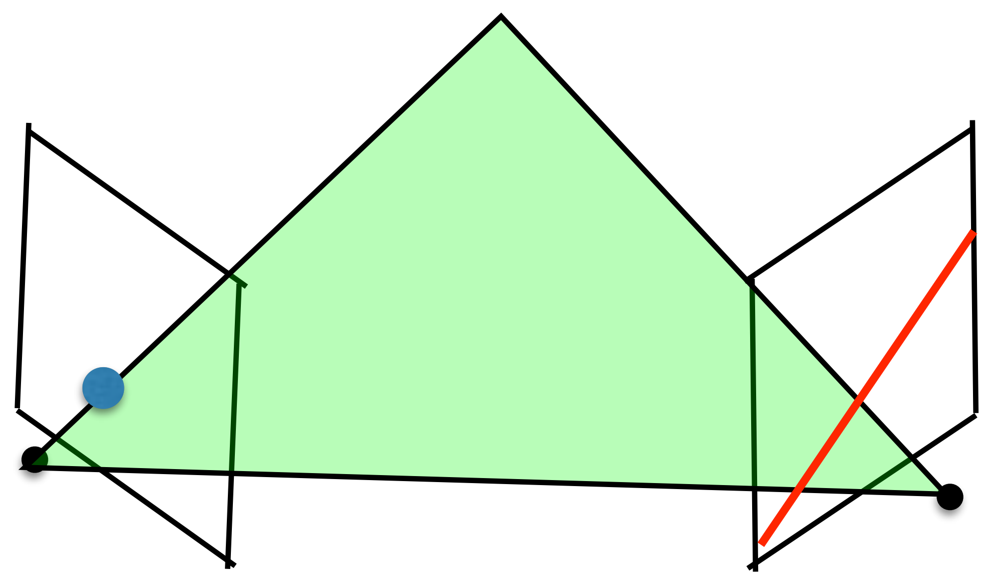
(ii) Camera geometry (motion): Given a set of corresponding image points $\{x_i \leftrightarrow x'_i\}$, $i=1,\ldots,n$, what are the cameras P and P' for the two views?

(iii) Scene geometry (structure): Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras P, P', what is the position of (their pre-image) X in space?

Outline

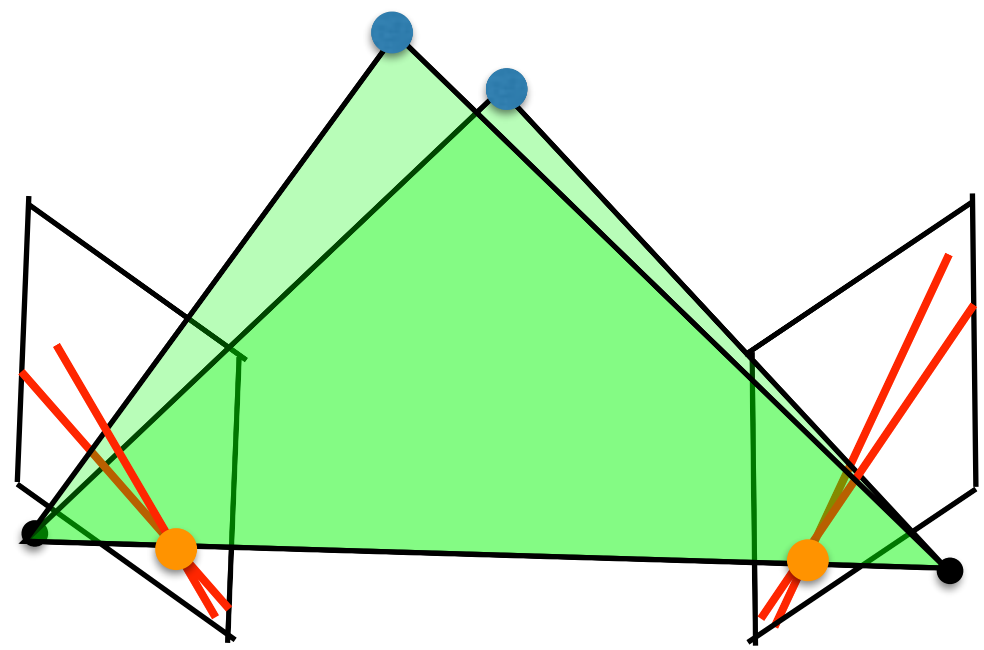
- 2-view geometry
- essential matrix, fundamental matrix
- properties
- estimation

Mathematical formulation



Goal: given point in left image, we want to compute the equation of the line on the right image

Definitions



How do epipolar lines change when we double distance between two cameras?

Epipolar plane: plane defined by 2 camera centers & candidate 3D point (green)

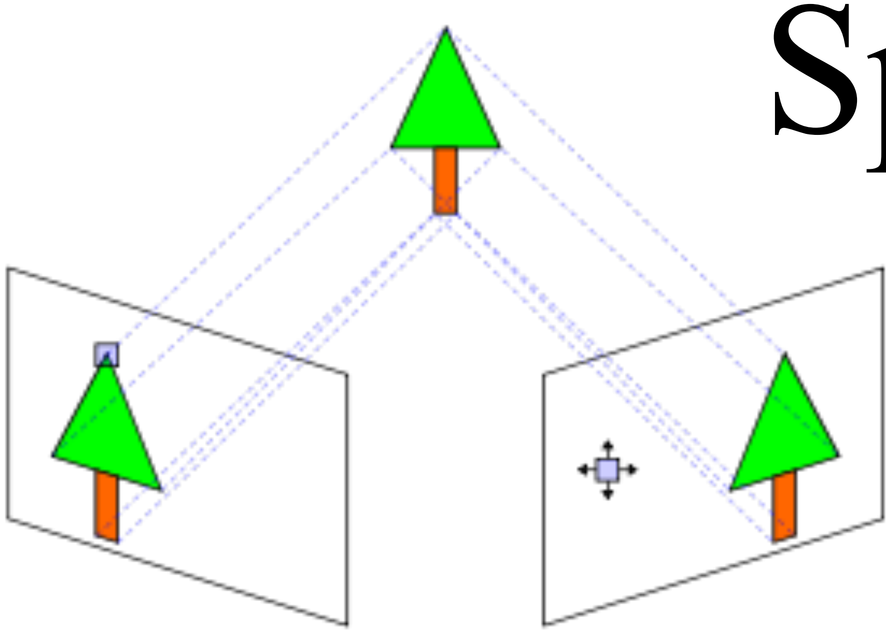
(also defined by 2 camera centers any 1 points in either image plane)

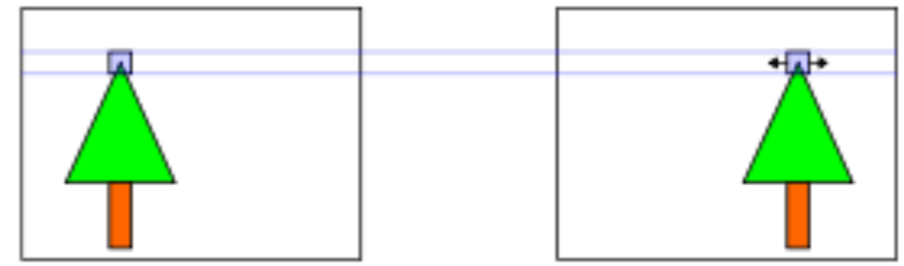
Epipolar lines: intersection of epipolar plane and image planes (red)

Epipoles: projection of camera center 1 in camera 2 (& vice versa) (orange)

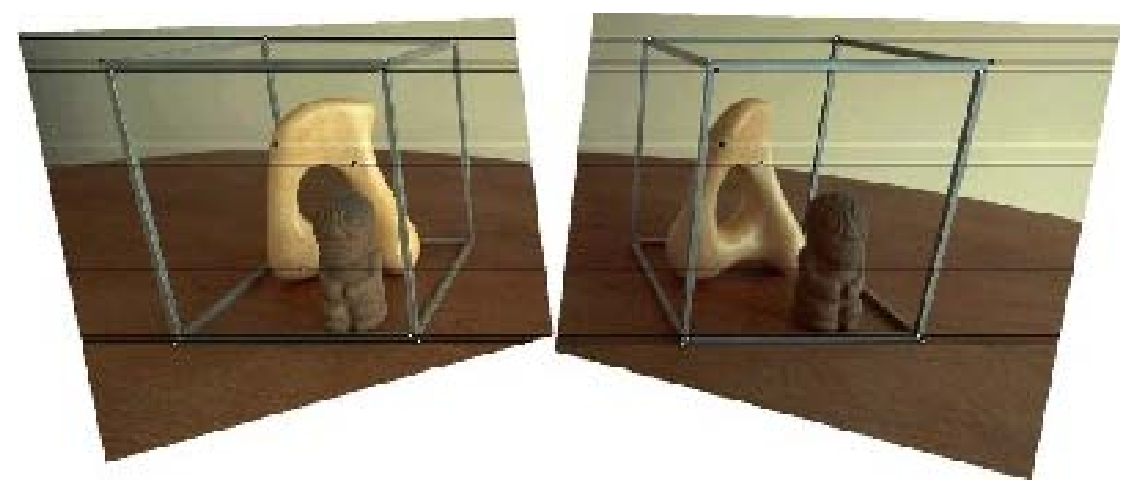
(set of all epipolar lines intersect at the epipoles)

Special case





Stereo Pair

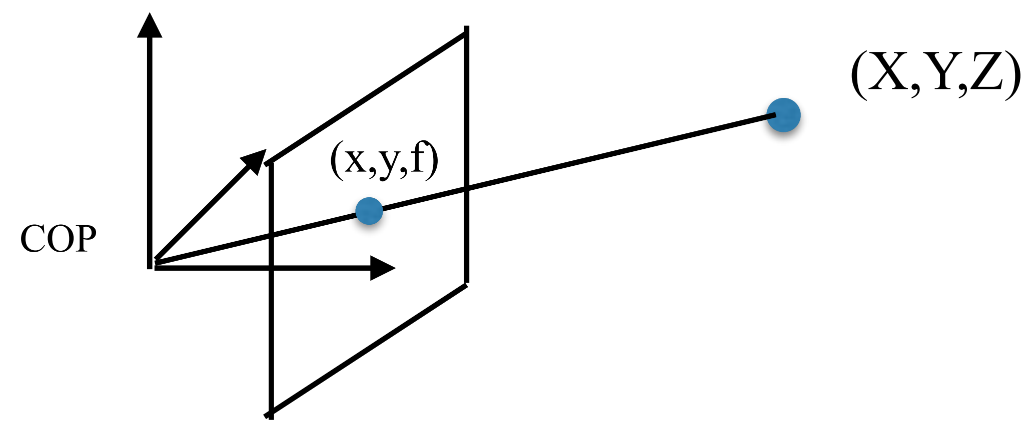


Rectified Stereo Pair

Rectify a stero pair with a homograpy transformation

Epipolar geometry is purely determined by camera extrinsics and camera instrinics

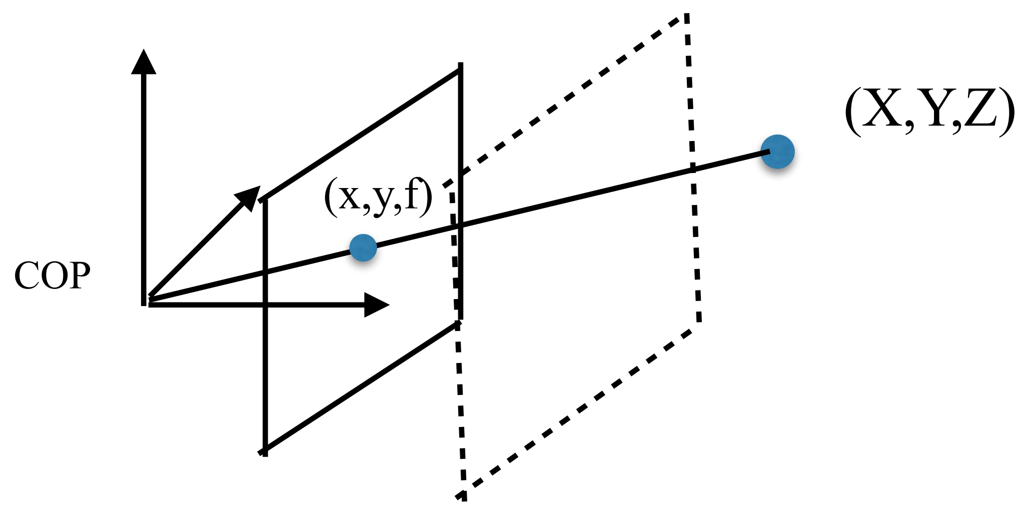
Projecting from camera coordinate system to image coordinates



$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\lambda \mathbf{x} = K\mathbf{X}$$

Projecting from camera coordinate system to *normalized* image coordinates



If K is known, work with warped image

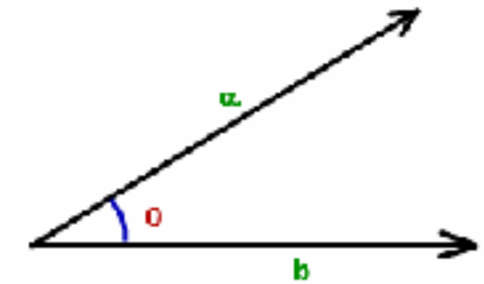
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\lambda \mathbf{x}' = \mathbf{X}$$

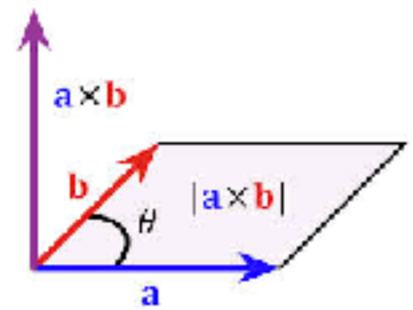
To simplify notation, we'll use **x** instead of **x'**

Recall

Dot product: $\mathbf{a} \cdot \mathbf{b} = ||\mathbf{a}|| \;\; ||\mathbf{b}|| cos\theta$



Cross product: $\mathbf{a} \times \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \sin\theta \; \mathbf{n}$

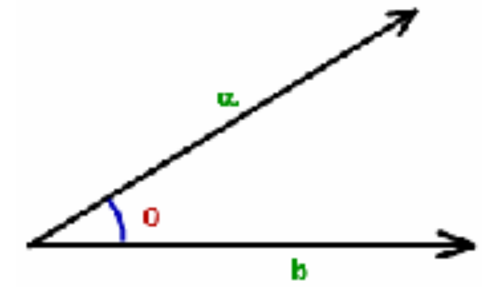


Cross product matrix: $\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \equiv \hat{\mathbf{a}}\mathbf{b}$

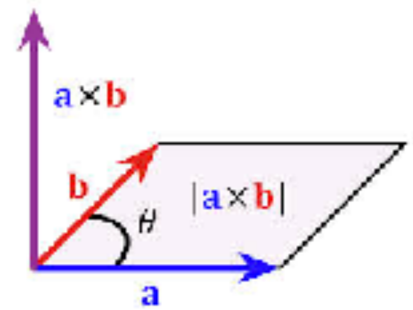
Important property (skew symmetric): $\hat{\mathbf{a}}^T = -\hat{\mathbf{a}}$

Recall

Dot product: $\mathbf{a} \cdot \mathbf{b} = ||\mathbf{a}|| \ \ ||\mathbf{b}|| cos\theta$



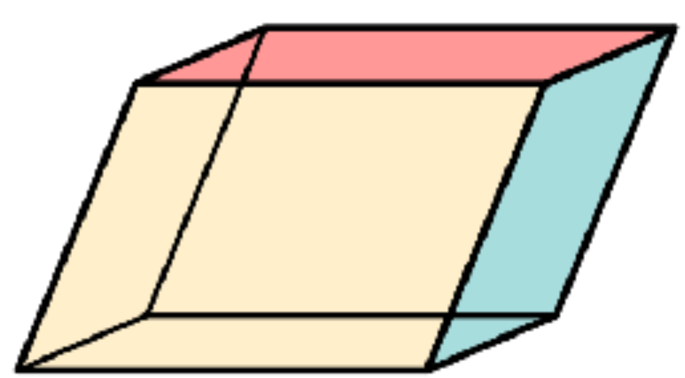
Cross product: $\mathbf{a} \times \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \sin\theta \; \mathbf{n}$



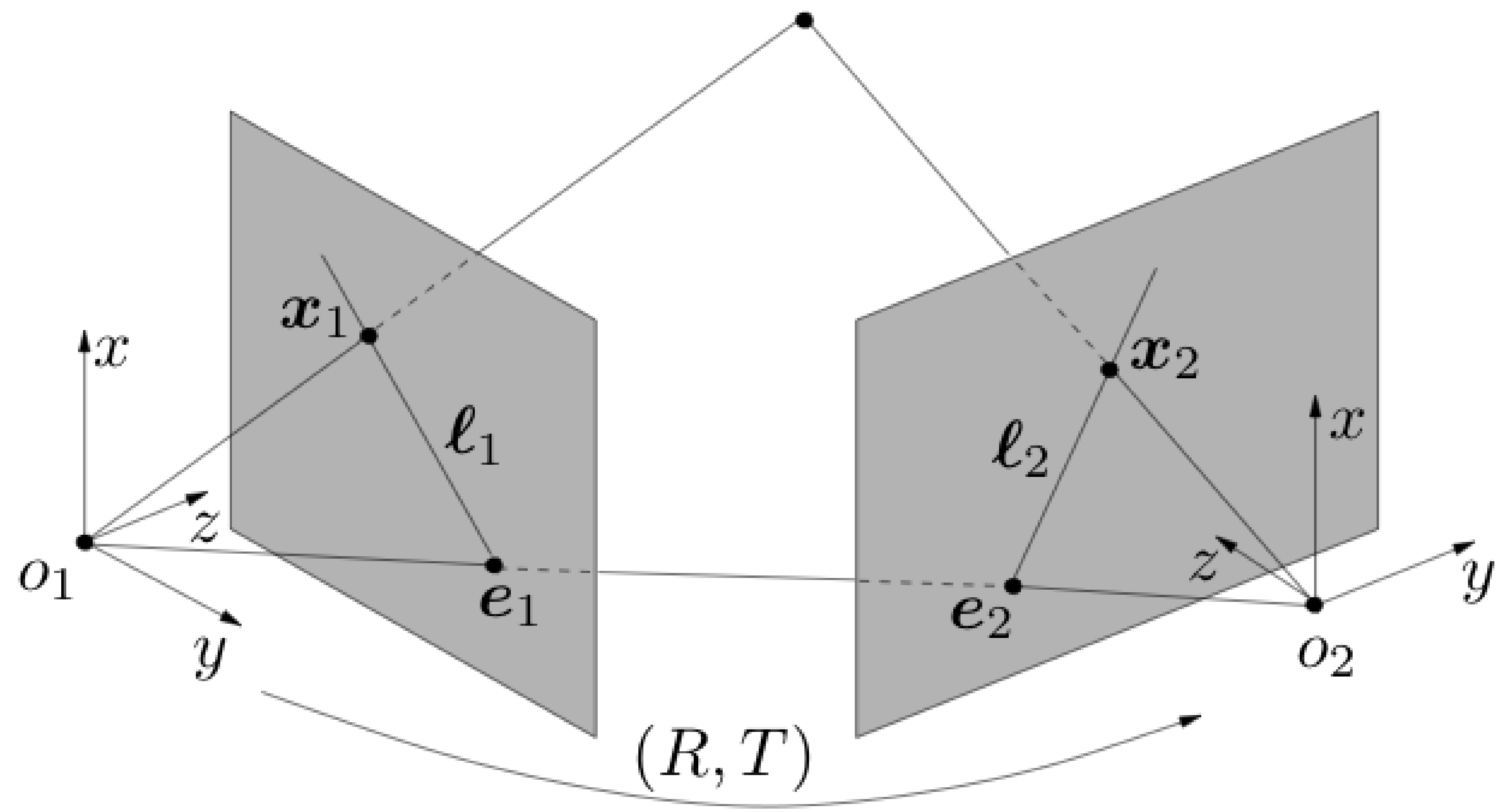
Cross product matrix: $\mathbf{a} \times \mathbf{b} = \hat{\mathbf{a}}\mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$

$\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$ = volume of parallelpiped

= 0 for coplanar vectors



Calibrated 2-view geometry



$$\boxed{\boldsymbol{X}_2 = R\boldsymbol{X}_1 + \boldsymbol{T}}$$

$$\boldsymbol{X}_1 = \lambda_1 \boldsymbol{x}_1, \quad \boldsymbol{X}_2 = \lambda_2 \boldsymbol{x}_2$$

Epipolar geometry

$$\boxed{\boldsymbol{X}_2 = R\boldsymbol{X}_1 + \boldsymbol{T}}$$

$$\boldsymbol{X}_1 = \lambda_1 \boldsymbol{x}_1, \quad \boldsymbol{X}_2 = \lambda_2 \boldsymbol{x}_2$$

$$\lambda_2 \boldsymbol{x}_2 = R\lambda_1 \boldsymbol{x}_1 + \boldsymbol{T}$$

Take (left) cross product of both sides with T

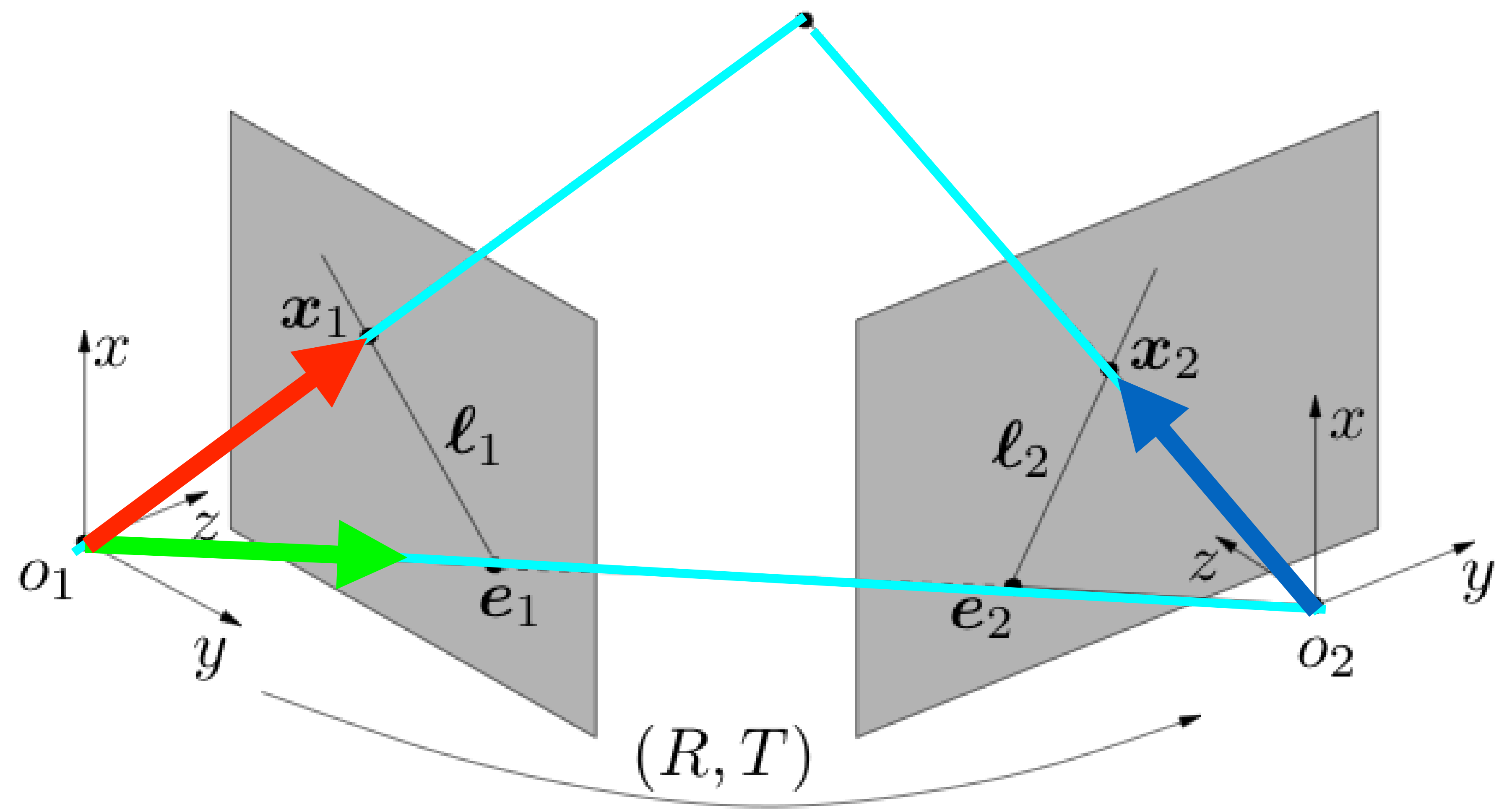
$$\lambda_2 \widehat{T} \boldsymbol{x}_2 = \widehat{T} R \lambda_1 \boldsymbol{x}_1 + \underbrace{\widehat{T} T}_{=\mathbf{0}}$$

Take (left) dot product of both sides with $\boldsymbol{x}_2$

$$\lambda_2 \underbrace{\boldsymbol{x}_2^\top \widehat{T} \boldsymbol{x}_2}_{=0} = \boldsymbol{x}_2^\top \widehat{T} R \lambda_1 \boldsymbol{x}_1$$

$$\boldsymbol{x}_2^\top \widehat{T} R \boldsymbol{x}_1 = 0$$

Geometric derivation



Simply the coplanar constraint applied to 3 vectors from camera 2's coordinate system

$$\mathbf{x}_2 \cdot (\mathbf{T} \times R\mathbf{x}_1) = 0$$

Epipolar geometry

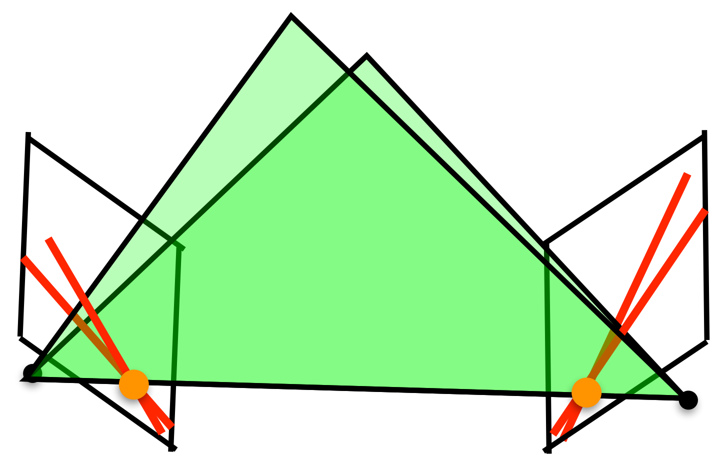
$$\boldsymbol{x}_2^\top \widehat{T} R \boldsymbol{x}_1 = 0$$

$$\boxed{\boldsymbol{x}_2^\top E \boldsymbol{x}_1 = 0}$$

E is known as the *essential* matrix

Fundamental matrix

(Faugeras and Luong, 1992)



In uncalibrated case, we need to account for camera intrinsics:

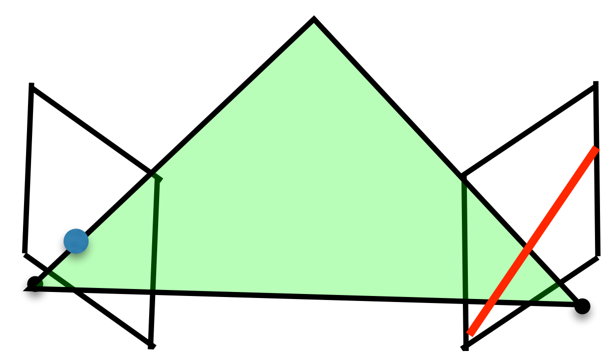
$$\lambda \mathbf{x} = K\mathbf{X}$$

$$E = \hat{T}R$$

$$F = {K_2}^{-T} E K_1^{-1}$$

Essential matrix

$$\boxed{\boldsymbol{x}_2^\top E \boldsymbol{x}_1 = 0}$$



$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = E \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

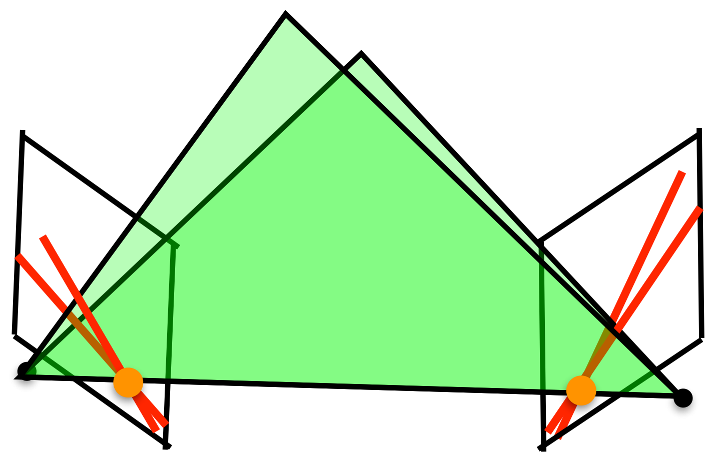
$$a x_2 + b y_2 + c = 0$$

Maps a (x1,y1) point from left image to line in right image (and vice versa)

But how is this different from a Homography (also a 3X3 matrix)?

Epipoles

$$\boxed{\boldsymbol{x}_2^\top E \boldsymbol{x}_1 = 0}$$



We'll write epipolar lines as 3-vectors: $\mathbf{l_2} = E\mathbf{x_1}$

Note that all epipolar lines in an image plane intersect at the epipole. Equivalently, the epipole has a distance of zero from every epipolar line: $\boldsymbol{e}_2^\top \boldsymbol{l}_2 = 0, \forall \boldsymbol{x}_1$, and similarly $\boldsymbol{e}_1^\top \boldsymbol{l}_1 = 0, \forall \boldsymbol{x}_2$.

For this to hold true, $\boldsymbol{e}_2^\top E$ and $E\boldsymbol{e}_1$ must be zero vectors, i.e.,

$$\boldsymbol{e}_2^\top E = \mathbf{0}, \qquad E\boldsymbol{e}_1 = \mathbf{0}$$

Thus $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are vectors in the right and left null space of E, respectively, i.e., the left and right singular vectors of E with singular value 0.

Outline

- 2-view geometry
- essential matrix, fundamental matrix
- properties
- estimation

Overview

Fundamental matrices:

$$\mathbf{x_2}^T F \mathbf{x_1} = 0$$

8 DOFs because of scale ambiguity

Rank 2

Essential matrices:

$$\boxed{\boldsymbol{x}_2^\top E \boldsymbol{x}_1 = 0}$$

$$\boldsymbol{x}_2^\top \widehat{T} R \boldsymbol{x}_1 = 0$$

More-or-less behaves like a cross-product (skew symmetric matrix)

Properties (essential matrix)

https://en.wikipedia.org/wiki/Essential_matrix#Properties_of_the_essential_matrix

Q. How many DOFs are needed to specify an essential matrix?

3 (rotations) + 2 (translation direction)

Q. Can any 3x3 matrix be an essential matrix?

No…

E is the product of a rotation and skew-symmetric matrix

Singular values of E = (sigma,sigma,0)

[rotations do not effect singular values]

Q. Given E, can we uniquely recover R,t?

Almost. It is unique up to easy-to-deal with symmetries

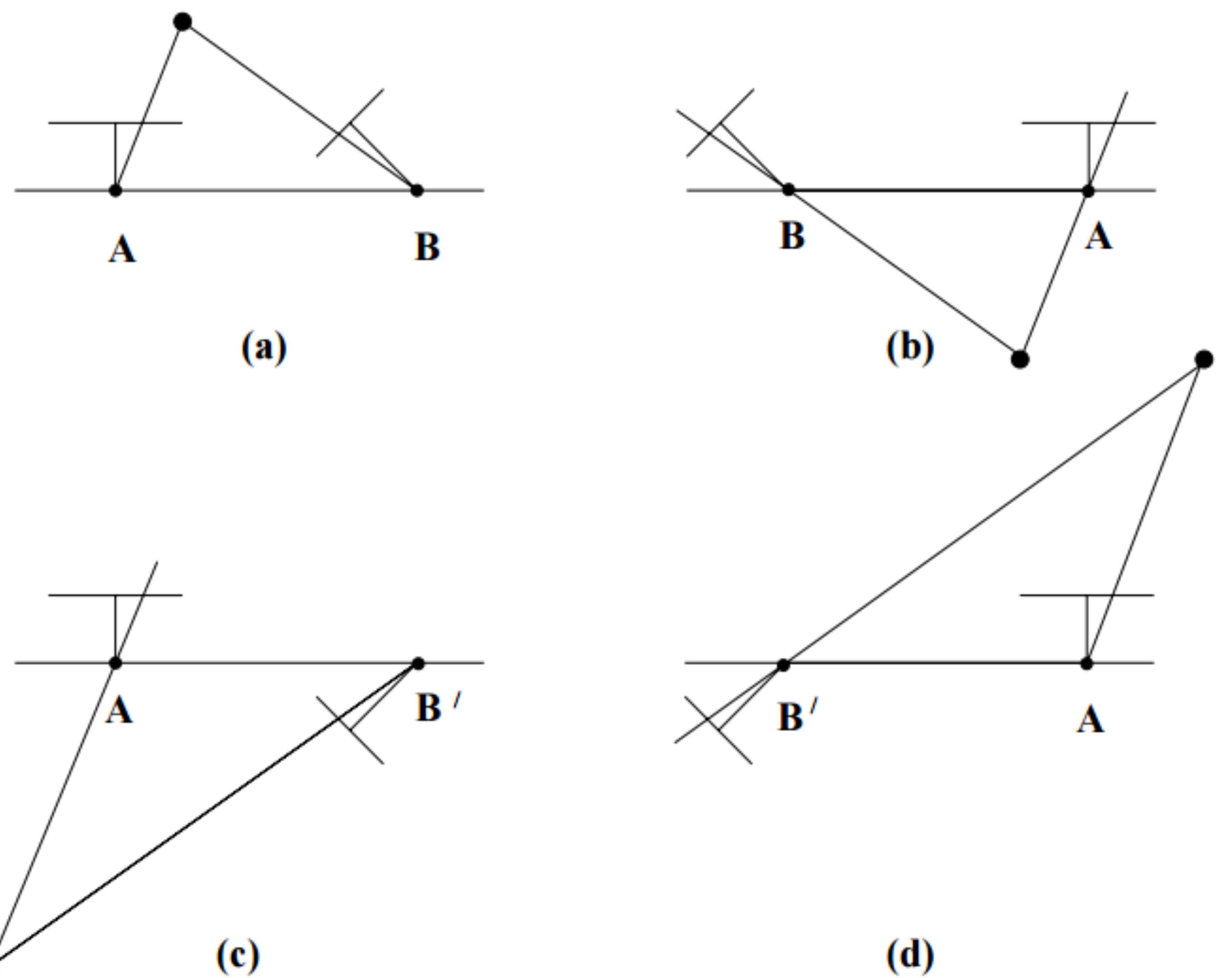
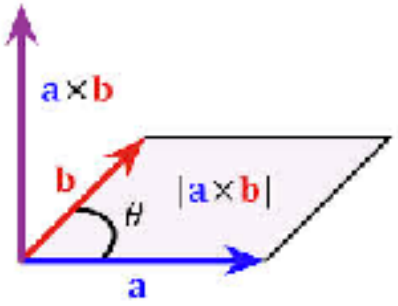


Fig. 8.12. **The four possible solutions for calibrated reconstruction from E.** *Between the left and right sides there is a baseline reversal. Between the top and bottom rows camera B rotates 180° about the baseline. Note, only in (a) is the reconstructed point in front of both cameras.*

Background: SVDs of skew symmetric matrics

Any skew-symmetric matrice ($A = -A^T$) can be thought of as a cross-product

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \equiv \hat{\mathbf{a}}\mathbf{b}$$



SVD of a skew-symmetric matrix:

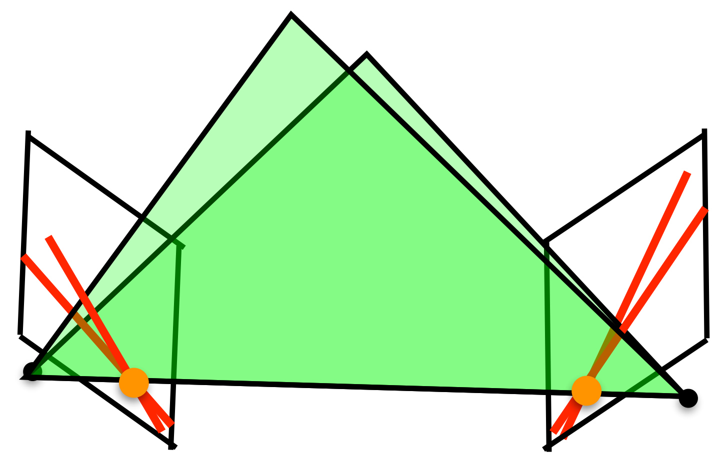
$$\hat{\mathbf{a}} = \begin{bmatrix} -\mathbf{e}_2 & \mathbf{e}_1 & \mathbf{e}_3 \end{bmatrix} \begin{bmatrix} ||a|| & 0 & 0 \\ 0 & ||a|| & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{bmatrix}$$

where $\mathbf{e}_3 = \mathbf{a} \,/\, ||\mathbf{a}||$

One singular value is 0 and the other two = $||\mathbf{a}||$

$$\hat{\mathbf{a}} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \begin{bmatrix} ||a|| & 0 & 0 \\ 0 & ||a|| & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{bmatrix}$$

Recovering T,R from E

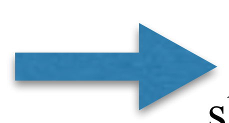


1. Universal scale ambiguity

Doubling T results in same epipolar lines

Let's fix $||\mathbf{T}|| = 1$

[notation switch]



unit-vector: $\hat{\mathbf{t}}$

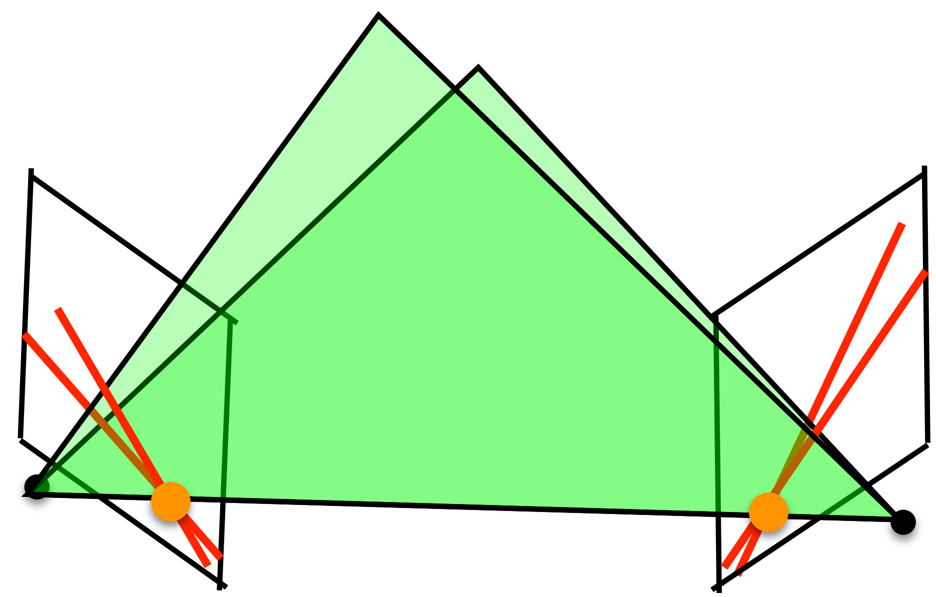
skew-symmetric matrix: $\hat{\mathbf{t}}_{\times}$

Numerous methods for recovering **t**,R from E exist:
SVD, Louget-Higgen's alg, etc.

Recovering T from E

SVD-based approach for noise-free E (Szeliski Chap 7.2)

$$\boxed{\boldsymbol{x}_2^\top E \boldsymbol{x}_1 = 0}$$



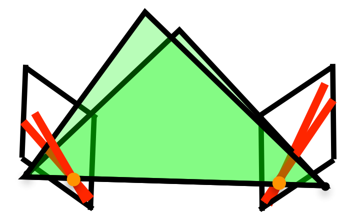
Take (left-handside) cross product of E = $[t]_x$R with t

$$\hat{\boldsymbol{t}}^T \boldsymbol{E} = 0.$$

Implies that translation vector = epipole in right image (in homogenous coordinates)

Recovering T from E

SVD-based approach for noise-free E (Szeliski Chap 7.2)



$$\boldsymbol{E} = [\hat{\boldsymbol{t}}]_\times \boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \hat{\boldsymbol{t}} \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_0^T \\ \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \end{bmatrix}$$

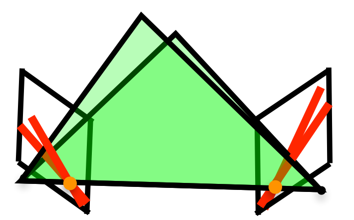
Set translation direction = smallest left singular vector of E

But we can't distinguish E from -E, so we only know direction up to a sign

Aside: $\mathbf{v}_2$ = epipole in left image

Recovering R from E

SVD-based approach (Szeliski Chap 7.2)



Recall skew-symmetric decomposition (for unit-norm vector)

$$[\hat{\boldsymbol{t}}]_\times = \boldsymbol{S}\boldsymbol{Z}\boldsymbol{R}_{90^\circ}\boldsymbol{S}^T = \begin{bmatrix} \boldsymbol{s}_0 & \boldsymbol{s}_1 & \hat{\boldsymbol{t}} \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & \\ 1 & 0 & \\ & & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_0^T \\ \boldsymbol{s}_1^T \\ \hat{\boldsymbol{t}}^T \end{bmatrix}$$

$$\boldsymbol{E} = [\hat{\boldsymbol{t}}]_\times \boldsymbol{R} = \boldsymbol{S}\boldsymbol{Z}\boldsymbol{R}_{90^\circ}\boldsymbol{S}^T\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

By matching orthogonal and diagonal matrices, S = U, Z = Sigma

$$\boldsymbol{R}_{90^\circ}\boldsymbol{U}^T\boldsymbol{R} = \boldsymbol{V}^T$$

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{R}_{90^\circ}^T\boldsymbol{V}^T$$

$$\boldsymbol{R} = \pm\boldsymbol{U}\boldsymbol{R}_{\pm 90^\circ}^T\boldsymbol{V}^T$$

Generate 4 possible rotations and keep 2 with determinant = 1 (non-reflections)

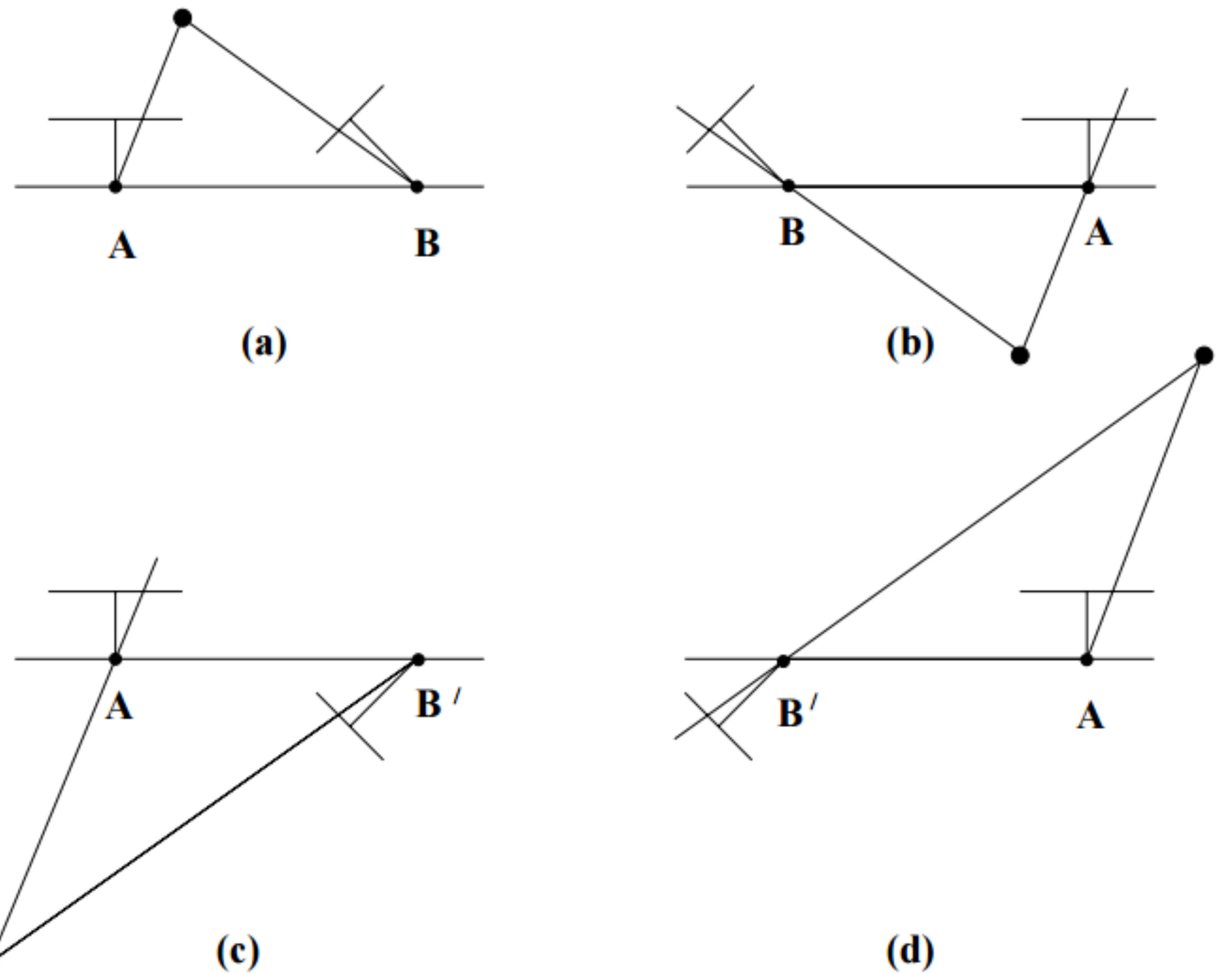
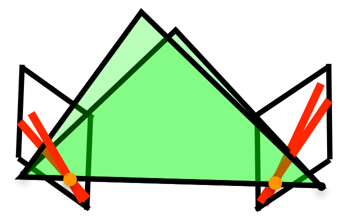


Fig. 8.12. **The four possible solutions for calibrated reconstruction from E.** *Between the left and right sides there is a baseline reversal. Between the top and bottom rows camera B rotates* 180° *about the baseline. Note, only in (a) is the reconstructed point in front of both cameras.*

Properties (fundamental matrix)



$$\mathbf{x_2}^T K_2^{-T} \hat{T} R K_1^{-1} \mathbf{x_1} = 0$$

$$\boxed{\mathbf{x_2}^T F \mathbf{x_1} = 0}$$

Q. How many DOFs are needed to specify F?

8 = 9 - 1 (for scale)

Q. Can any 3x3 matrix be a fundamental matrix?

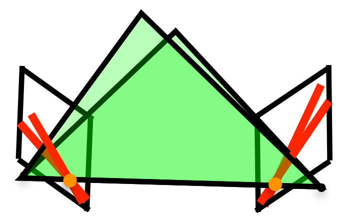
No! epipoles are still in the null space, implying rank(F) = 2

Proof: Let $\mathbf{e}_2 = K_2 \mathbf{T}$

$\mathbf{e}_2^T F = 0$

(similar argument for $\mathbf{e}_1$; c.f. Invitation to 3D Vision, Chap 6.2)

Properties (fundamental matrix)

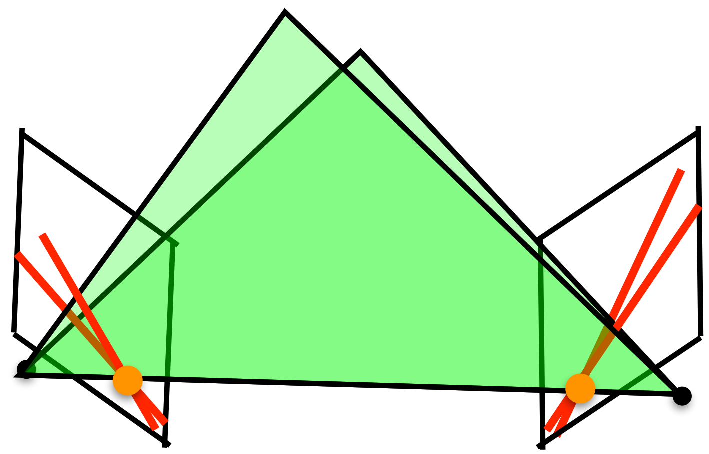


$$\text{F}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \boldsymbol{e}_1 \end{bmatrix} \begin{bmatrix} \sigma_0 & & \\ & \sigma_1 & \\ & & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_0^T \\ \boldsymbol{v}_1^T \\ \boldsymbol{e}_0^T \end{bmatrix}$$

Two non-zero singular values are not (in general) equal

Singular vectors with zero singular valur are the eipoles

Essential and Fundamental Matrices



$$E = \hat{T}R$$

$$\mathbf{x}_2^T E \mathbf{x}_1 = 0$$

$$E = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \mathbf{e}_2 \end{bmatrix} \begin{bmatrix} \sigma & & \\ & \sigma & \\ & & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_0^T \\ \mathbf{v}_1^T \\ \mathbf{e}_1^T \end{bmatrix}$$

"Proof": properties of skew-symmetric matrices

$$F = K_2^{-T} E K_1^{-1}$$

$$\mathbf{x}_2^T F \mathbf{x}_1 = 0$$

$$F = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \mathbf{e}_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_0^T \\ \mathbf{v}_1^T \\ \mathbf{e}_1^T \end{bmatrix}$$

Proof: scale ambiguity

where e1, e2 are epipoles in right and left images

Formal characterizations

Ma et al, An Invitation to 3D Vision

Theorem 5.1 (Characterization of the essential matrix). *A non-zero matrix $E \in \mathbb{R}^{3\times3}$ is an essential matrix* if and only if *E has a singular value decomposition (SVD): $E = U\Sigma V^T$ with*

$$\Sigma = diag\{\sigma, \sigma, 0\}$$

for some $\sigma \in \mathbb{R}_+$ and $U, V \in SO(3)$.

Remark 6.1. *Characterization of the fundamental matrix. A non-zero matrix $F \in \mathbb{R}^{3\times3}$ is a fundamental matrix if F has a singular value decomposition (SVD): $E = U\Sigma V^T$ with*

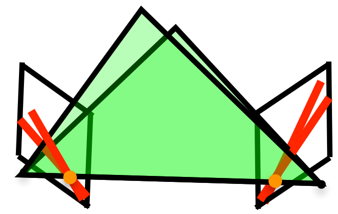
$$\Sigma = diag\{\sigma_1, \sigma_2, 0\}$$

for some $\sigma_1, \sigma_2 \in \mathbb{R}_+$.

Outline

- 2-view geometry
- essential matrix, fundamental matrix
- properties
- estimation

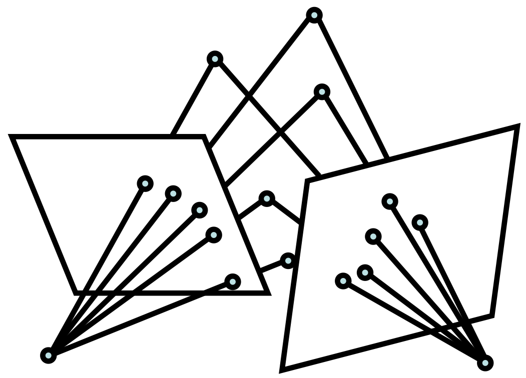
Estimation (fundamental matrix)



Assume we have a corrsponding pair of points: in noise-free case….

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = 0 \iff \begin{bmatrix} xx' & xy' & x & yx' & yy' & y & x' & y' & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

Estimation (fundamental matrix)

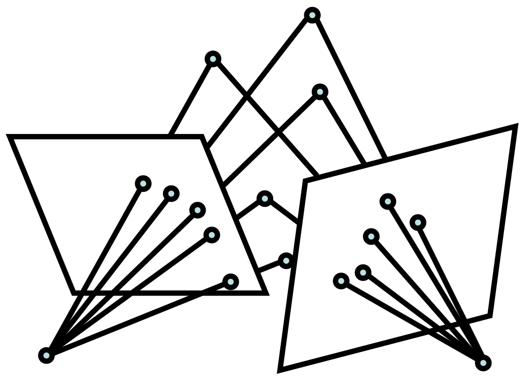


Given m point correspondences (x_i,y_i) and (x'_i,y'_i):

$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx'_m & x_my'_m & x_m & y_mx'_m & y_my'_m & y_m & x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

$$AF(:) = 0$$

Estimation (fundamental matrix)



Given m point correspondences (x_i,y_i) and (x'_i,y'_i):

$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx'_m & x_my'_m & x_m & y_mx'_m & y_my'_m & y_m & x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

AF(:) = 0

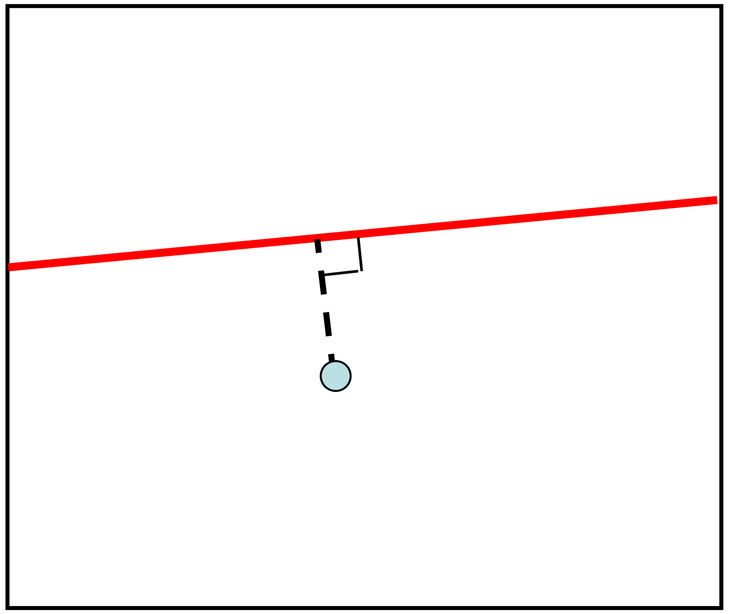
noisy case: $\min_{||F||=1} ||AF(:)||^2 = \min_F \sum_i (\mathbf{x}_i^T F \mathbf{x}'_i)^2$

Is this reasonable error to minimize?

Recall: distance of point from a line

https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line

$$\operatorname{distance}(ax + by + c = 0, (x_0, y_0)) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}.$$



${\mathbf{x}'}_i^T F \mathbf{x}_i$ is *scaled* euclidean distance of (x'_i, y'_i) from line defined by (x_i, y_i)

The eight-point algorithm

- Meaning of error $\sum_{i=1}^{N}(x_i^T F x_i')^2$:

  sum of squared distances between points x_i and epipolar lines Fx'_i (or points x'_i and epipolar lines $F^T x_i$) multiplied by a scale factor

- Nonlinear approach: minimize

$$\sum_{i=1}^{N}\left[\mathrm{d}^2(x_i, F x_i') + \mathrm{d}^2(x_i', F^T x_i)\right]$$

8-point algorithm

Longuet-Higgens

Given m point correspondences…

$$\begin{bmatrix} x_1x_1' & x_1y_1' & x_1 & y_1x_1' & y_1y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx_m' & x_my_m' & x_m & y_mx_m' & y_my_m' & y_m & x_m' & y_m' & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

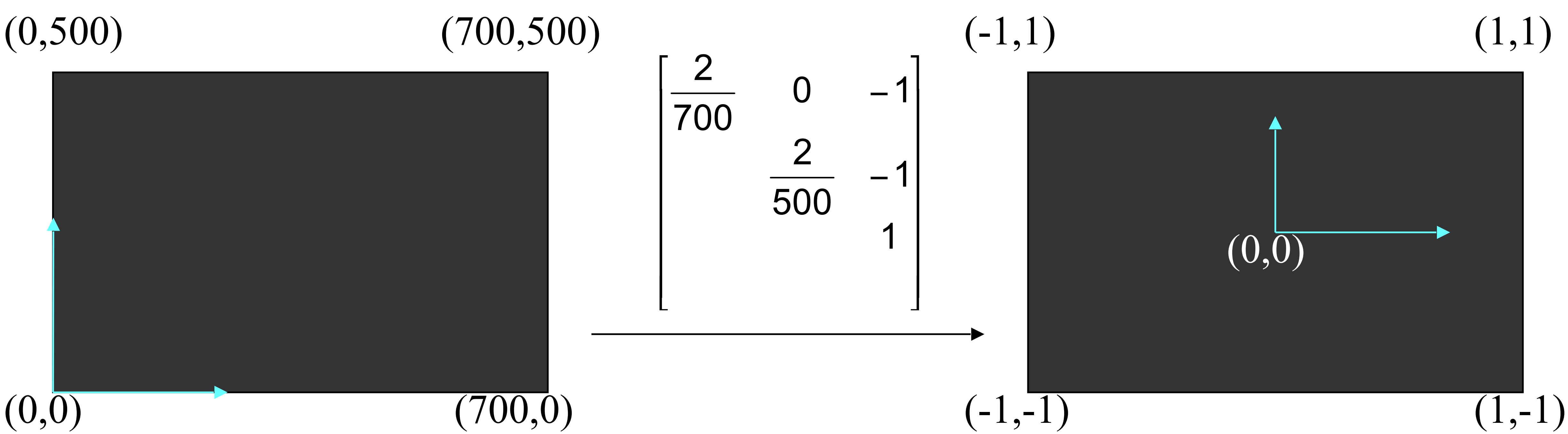
~10000 ~10000 ~100 ~10000 ~10000 ~100 ~100 ~100 1

! Orders of magnitude difference
Between column of data matrix
$\rightarrow$ least-squares yields poor results

“In Defence of the 8-point Algorithm”

(Hartley, PAMI´97)

Transform image to [-1,1]x[-1,1]



$$\begin{bmatrix} \frac{2}{700} & 0 & -1 \\ & \frac{2}{500} & -1 \\ & & 1 \\ & & \end{bmatrix}$$

SVD now produces good results

Final “annoying” issue

Least squares solution won’t produce F that satisfies rank 2
(or rank-2 E with 2 identical singular values)

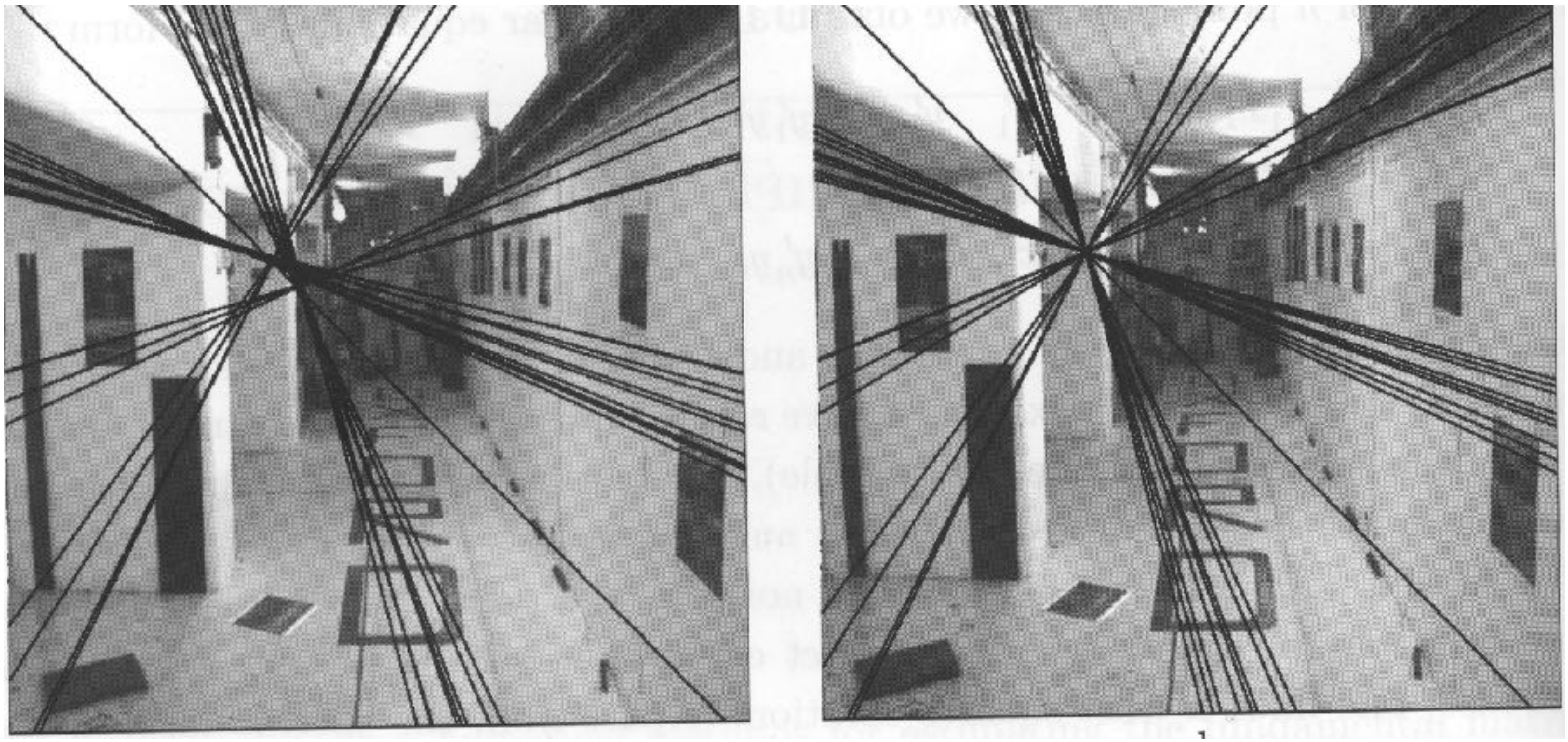
Solution: find the closest F/E (Frebonius norm) with SVD

$$X = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T$$

Closest fundamental matrix: set sigma3 = 0

Closest essential matrix: set sigma3 = 0, sigma = .5*(sigma1+sigma2)

Rank-2 Fundamental Matrix



7-point algorithm

Since F are rank-difficient, we can estimate them with m=7 correspondences

$$\begin{bmatrix} x_1x_1' & x_1y_1' & x_1 & y_1x_1' & y_1y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx_m' & x_my_m' & x_m & y_mx_m' & y_my_m' & y_m & x_m' & y_m' & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

AF(:)=0

Idea: search for null vector of A_{Mx9} that satisfies additional contraints (reshaped 3x3 matrix has 0 singular value)

1) A is rank 7. Find 2 vectors that span *null space* of A, F_1 and F_2.

2) Find alpha such that Determinant(alpha*F1 + (1-alpha)*F2) = 0

[3rd order polynomial in alpha with at least one real solution]

Aside: what if cameras are calibrated?

Turns out we only need 5 points, but need to find roots to 10th degree polynomial

[Nister 04]

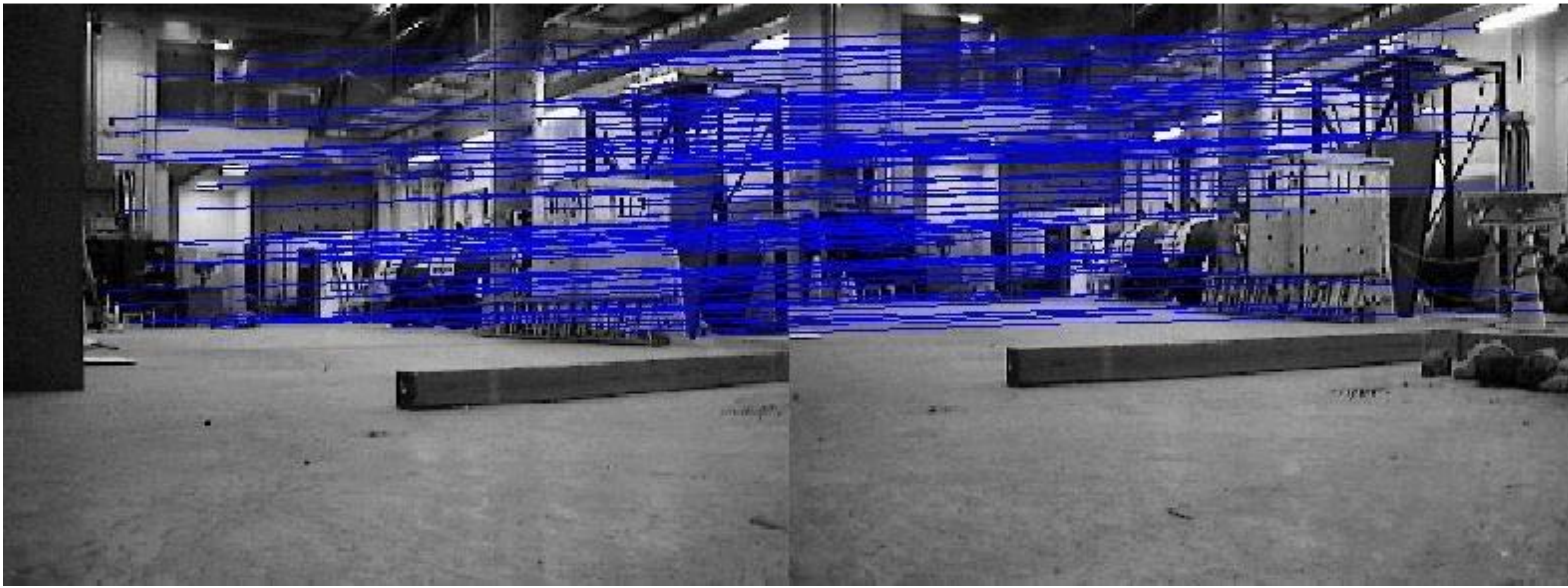
Recall: RANSAC

RANSAC loop:

1. Select feature pairs (at random)
2. Compute transformation T (exact)
3. Compute *inliers* (point matches where $|p_i' - \boldsymbol{T}\, p_i|^2 < \varepsilon$)
4. Keep largest set of inliers

5. Re-compute least-squares estimate of transformation **T** using all of the inliers

Fundamental matrix estimaiton with RANSAC



Outline

- 2-view geometry
- essential matrix, fundamental matrix
- properties
- estimation
- stereo

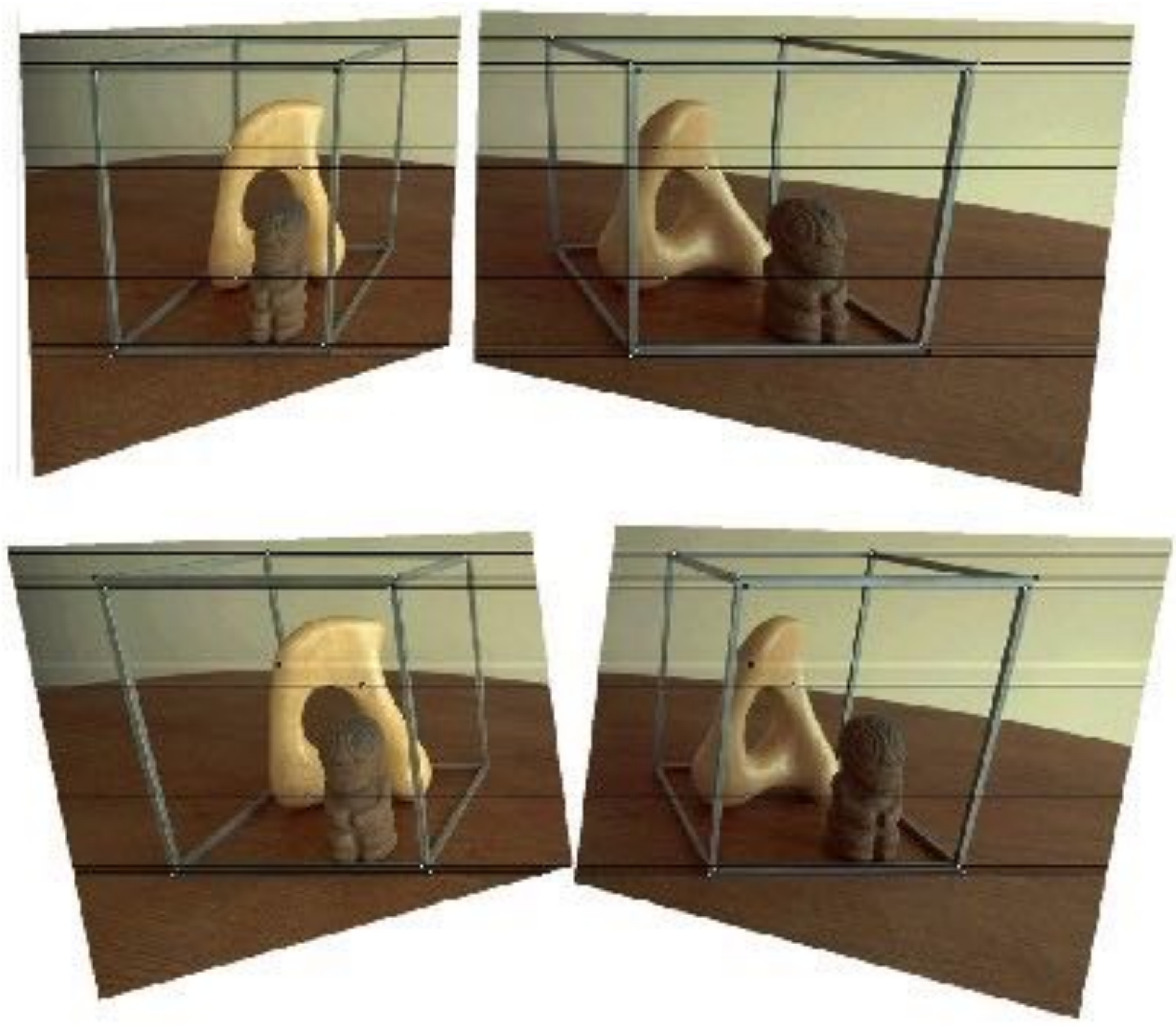
Three questions:

(i) **Correspondence geometry:** Given an image point x in the first view, how does this constrain the position of the corresponding point x' in the second image?

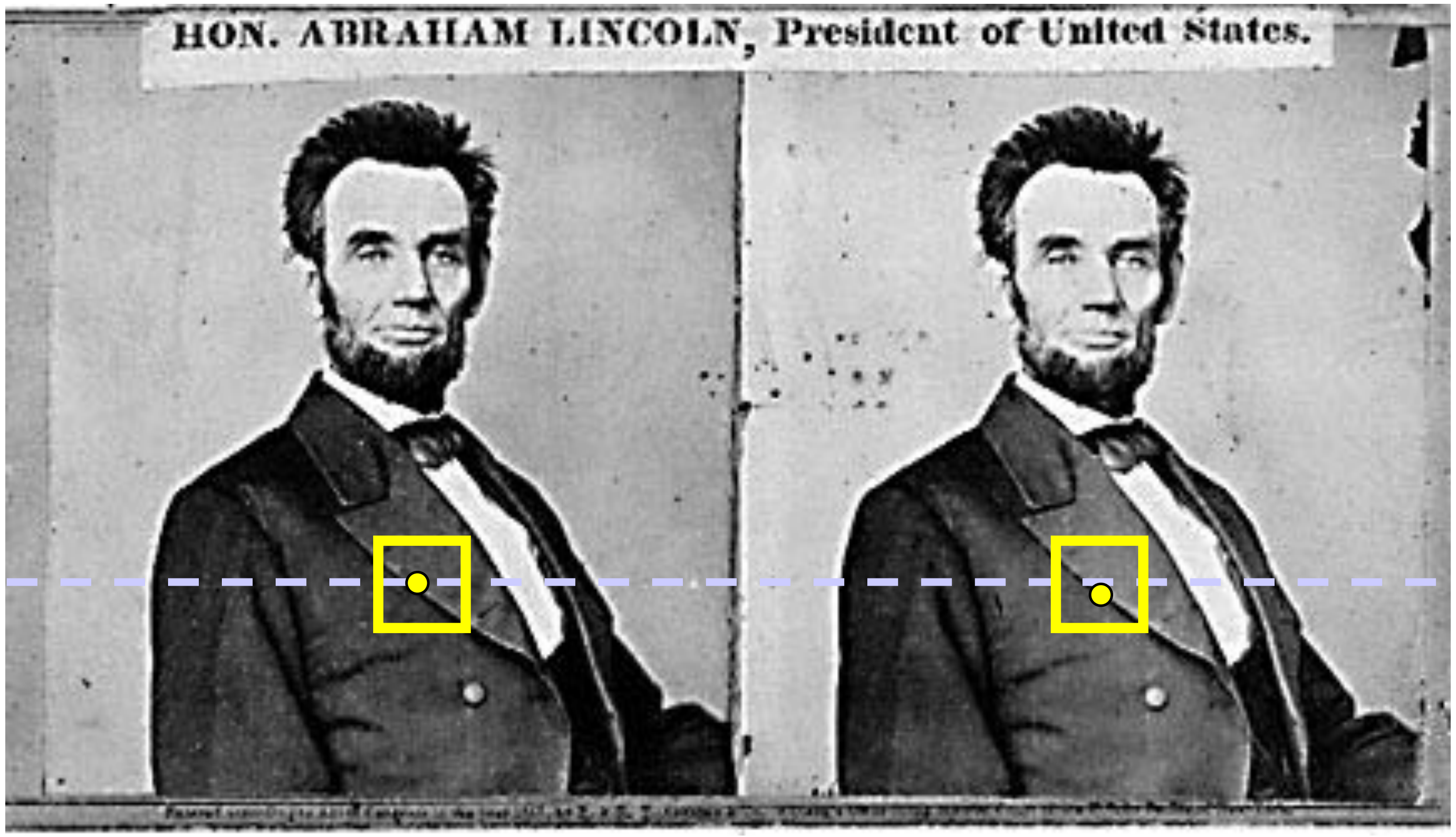
(ii) **Camera geometry (motion):** Given a set of corresponding image points $\{x_i \leftrightarrow x'_i\}$, $i=1,\dots,n$, what are the cameras P and P' for the two views?

(iii) **Scene geometry (structure):** Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras P, P', what is the position of (their pre-image) X in space?

Stereo



Basic Stereo Algorithm



For each epipolar line

For each pixel in the left image

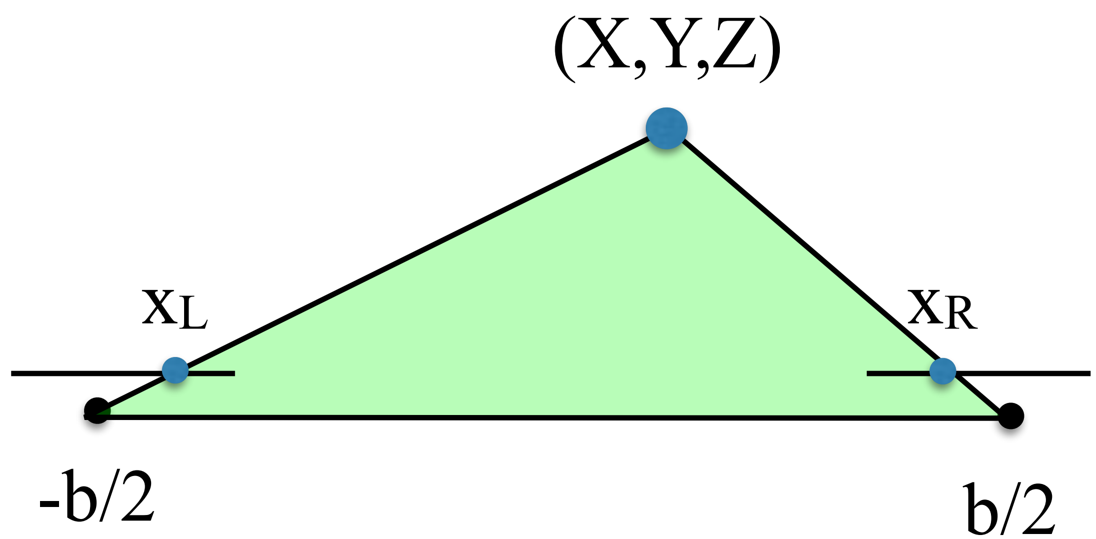
- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Improvement: match ***windows***

- (Normalized) Correlation, Sum of Squared Difference (SSD), Sum of Absolute Differences (SAD), etc…

Triangulation for Rectified Stereo Pairs

Top-down view where world coordinates are centered between cameras



$$\frac{x_L}{f} = \frac{X + b/2}{Z} \qquad \frac{x_R}{f} = \frac{X - b/2}{Z} \qquad \frac{y_L}{f} = \frac{y_R}{f} = \frac{Y}{Z}$$

$$\Rightarrow \quad X = \frac{b(x_L + x_R)}{2(x_L - x_R)} \qquad Y = \frac{b(y_L + y_R)}{2(x_L - x_R)} \qquad Z = \frac{bf}{(x_L - x_R)}$$

$d = x_L - x_R = \dfrac{bf}{Z}$ is the **disparity** between corresponding left and right image points

- inverse proportional to depth Z
- disparity increases with baseline b

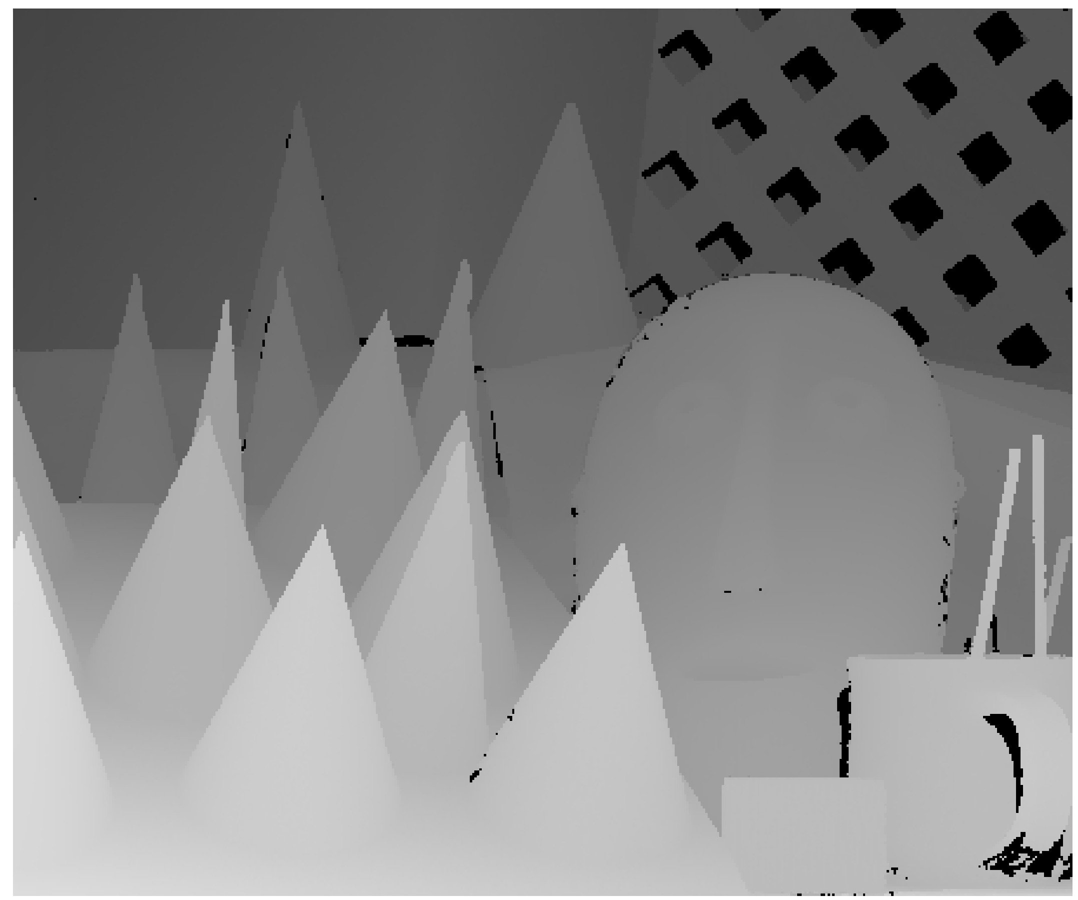
Disparity Maps

$$d = x_L - x_R = \frac{bf}{Z}$$

Left image

Right image

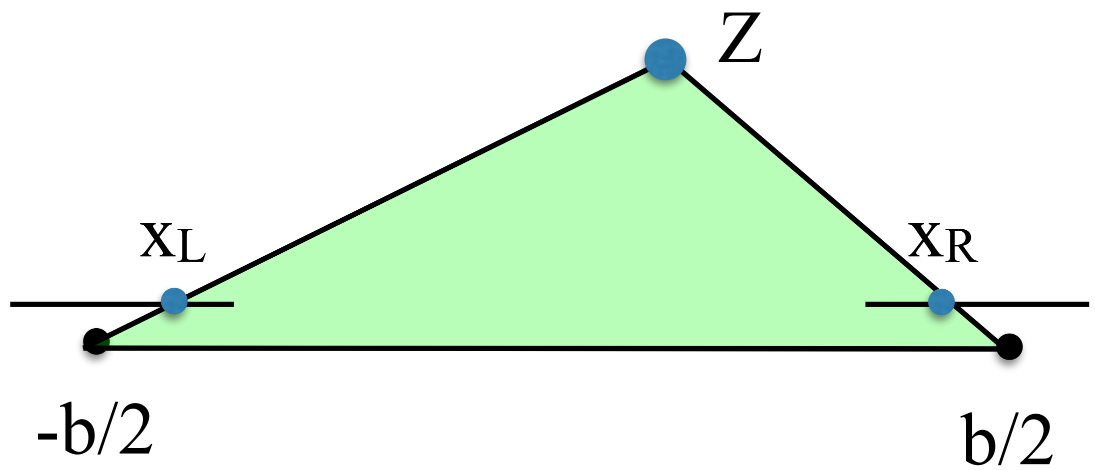
Disparity values (0-64)



Note how disparity is larger (brighter) for closer surfaces.

If we double the size of scene geometry and baseline, what happens to disparity?

Numerical stability



$$d = x_L - x_R = \frac{bf}{Z}$$

Scene + camera variables: Z,f,b

Dependant variable: d = function(Z,f,b)

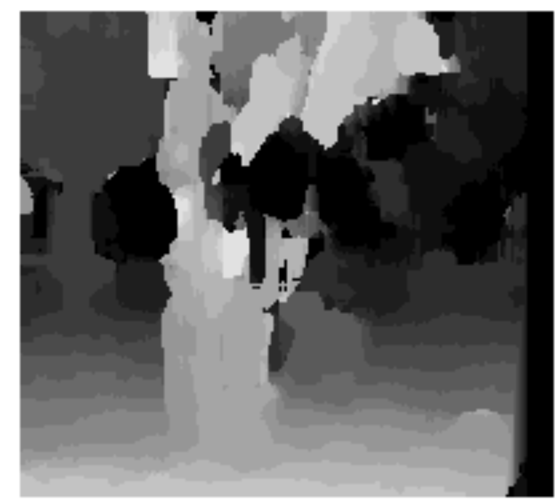
How do we characterize the error in depth Z given an error in disparity d, in terms of scene + camera?

$$Z = \frac{bf}{x_L - x_R} = \frac{bf}{d} \quad \Rightarrow \quad \frac{\partial Z}{\partial d} = -\frac{bf}{d^2} = -\frac{Z^2}{bf}$$

1. Error increases quadratically with depth (hard to reconstruct far away points)
2. Error inversely proportional to baseline (larger baselines increase numerical stability)

Disparity maps (in practice)

Small matching window
(better localization)



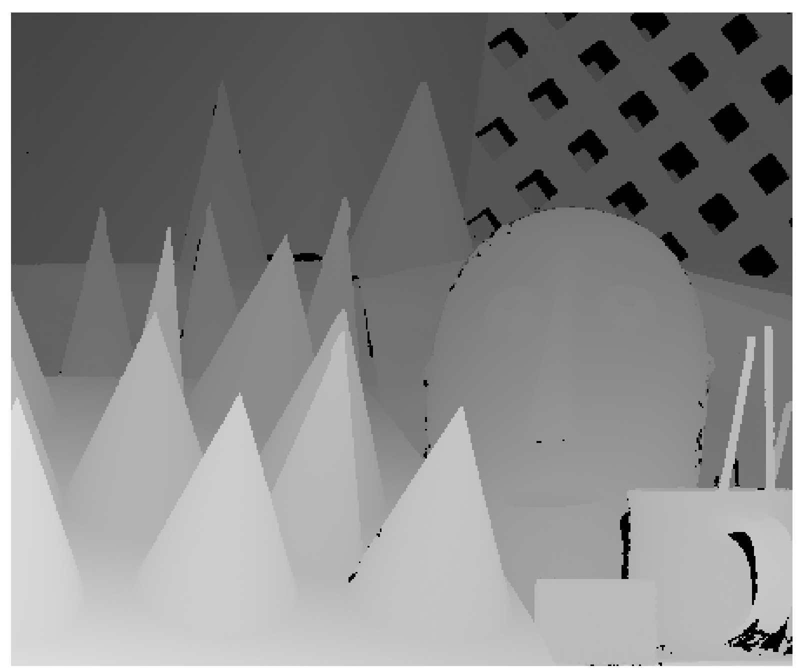
Large matching window
(better detection)

Variational stereo

Penalize differences in nearby disparities (a “1-d” flow problem!)

$$\min_{\mathrm{u,v}} \mathrm{E_{intensity}} + \mathrm{E_{smooth}}$$

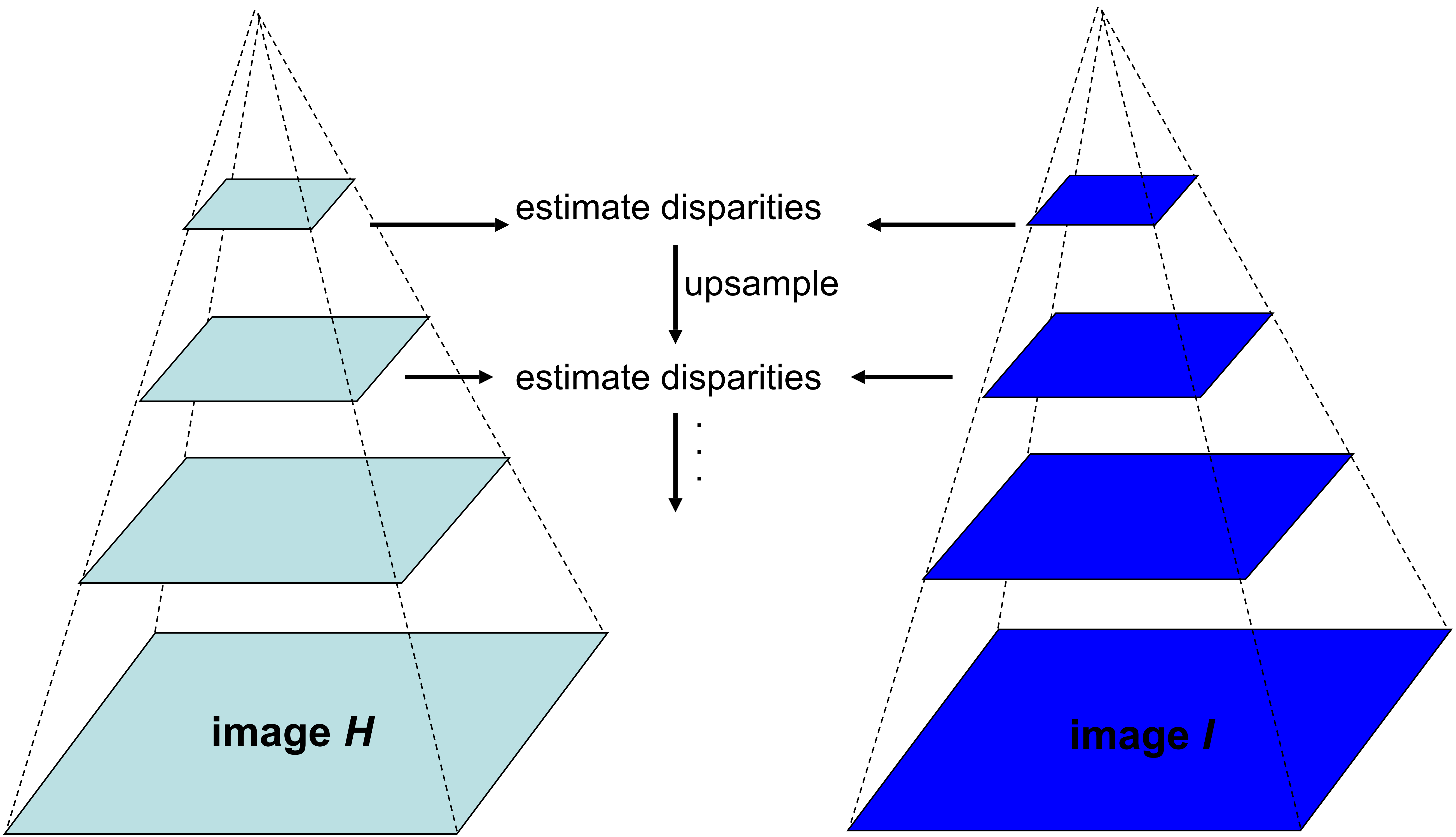
$$E_{intensity}(d) = \int\int (I_2(x + d(x,y), y) - I_1(x,y))^2 dx dy \qquad E_{smooth}(d) = \int\int ||\nabla d(x,y)||^2 dx dy$$



1. Linearlize $\mathrm{E_{intensity}}$ term and solve with least squares

2. Add robust error terms $\rho(\cdot)$ to handle discontinuties

Coarse-to-fine stereo



Gaussian pyramid of image *H*

Gaussian pyramid of image *I*

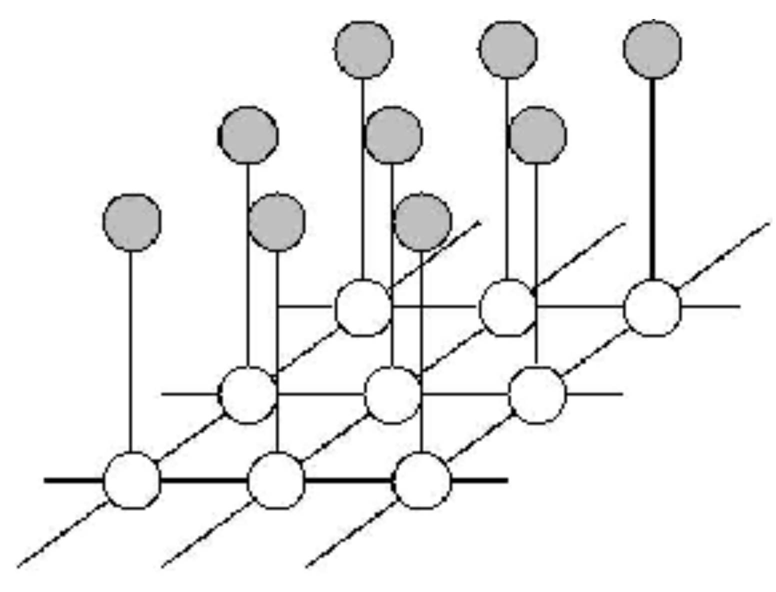
Discrete disparity estimation

$$z \in \{-5 \ldots 5\}$$

$$\phi_i(z_i) = \rho(||I_2(x_i + z_i, y_i) - I(x_i, y_i)||)$$

$$\psi_{ij}(z_i, z_j) = \rho(z_i - z_j)$$

$$E(z) = \sum_{i \in V} \phi_i(z_i) + \sum_{ij \in E} \psi_{ij}(z_i, z_j)$$

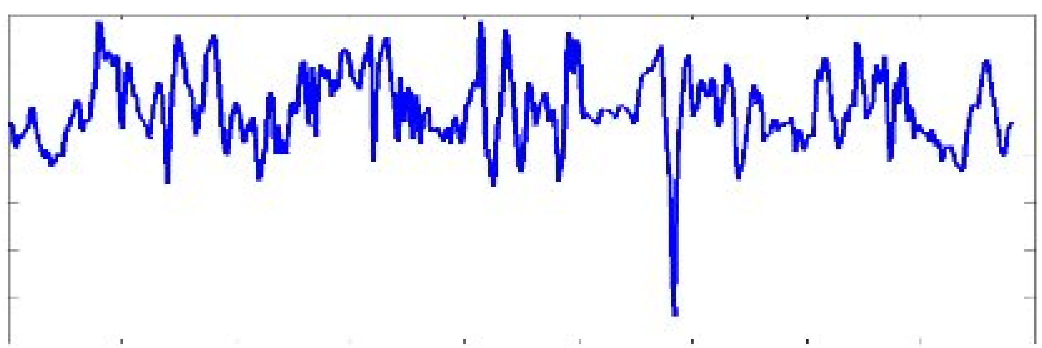


Solve with GraphCuts

Special case: single-scan-line consistency

Left Image

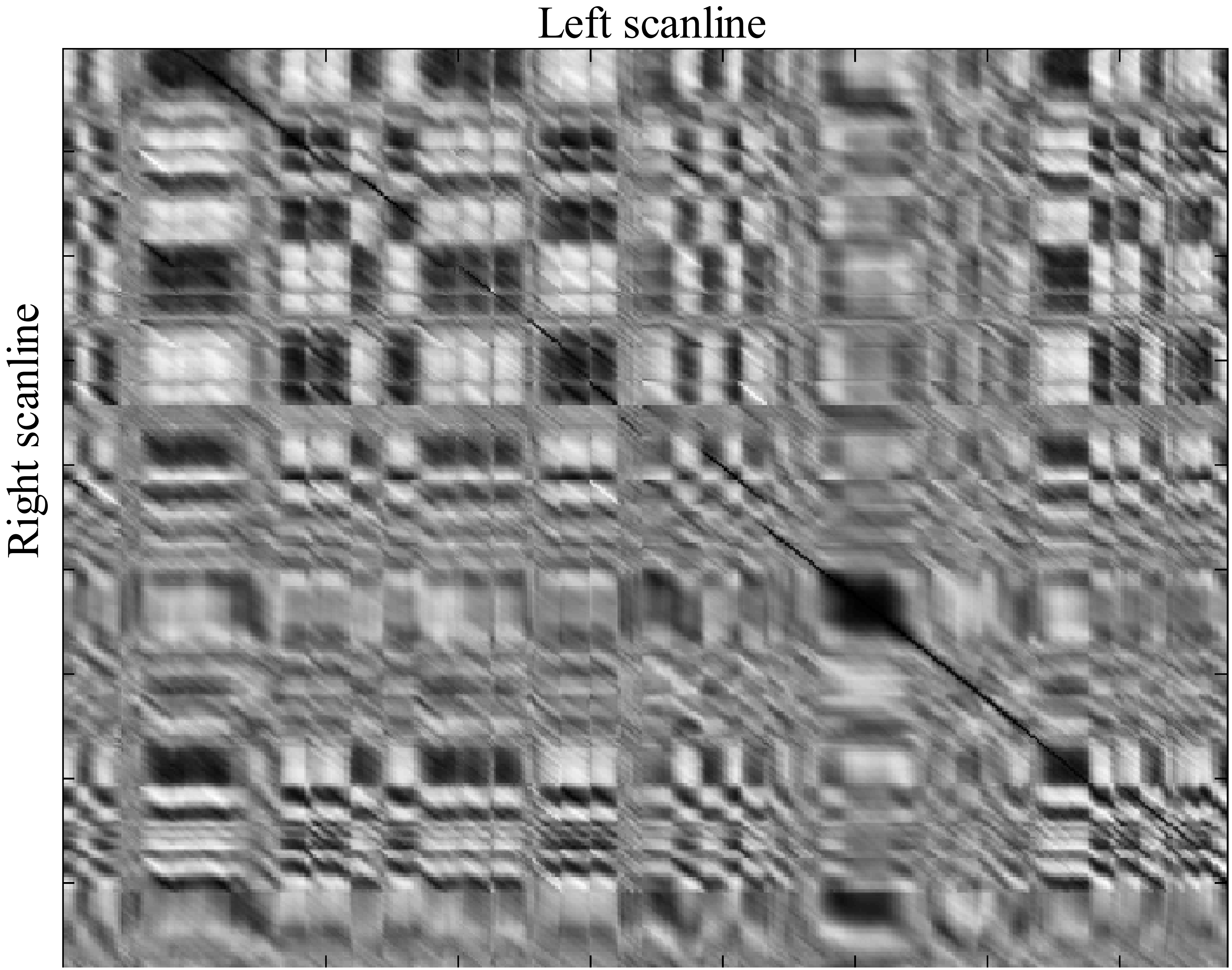
Right Image



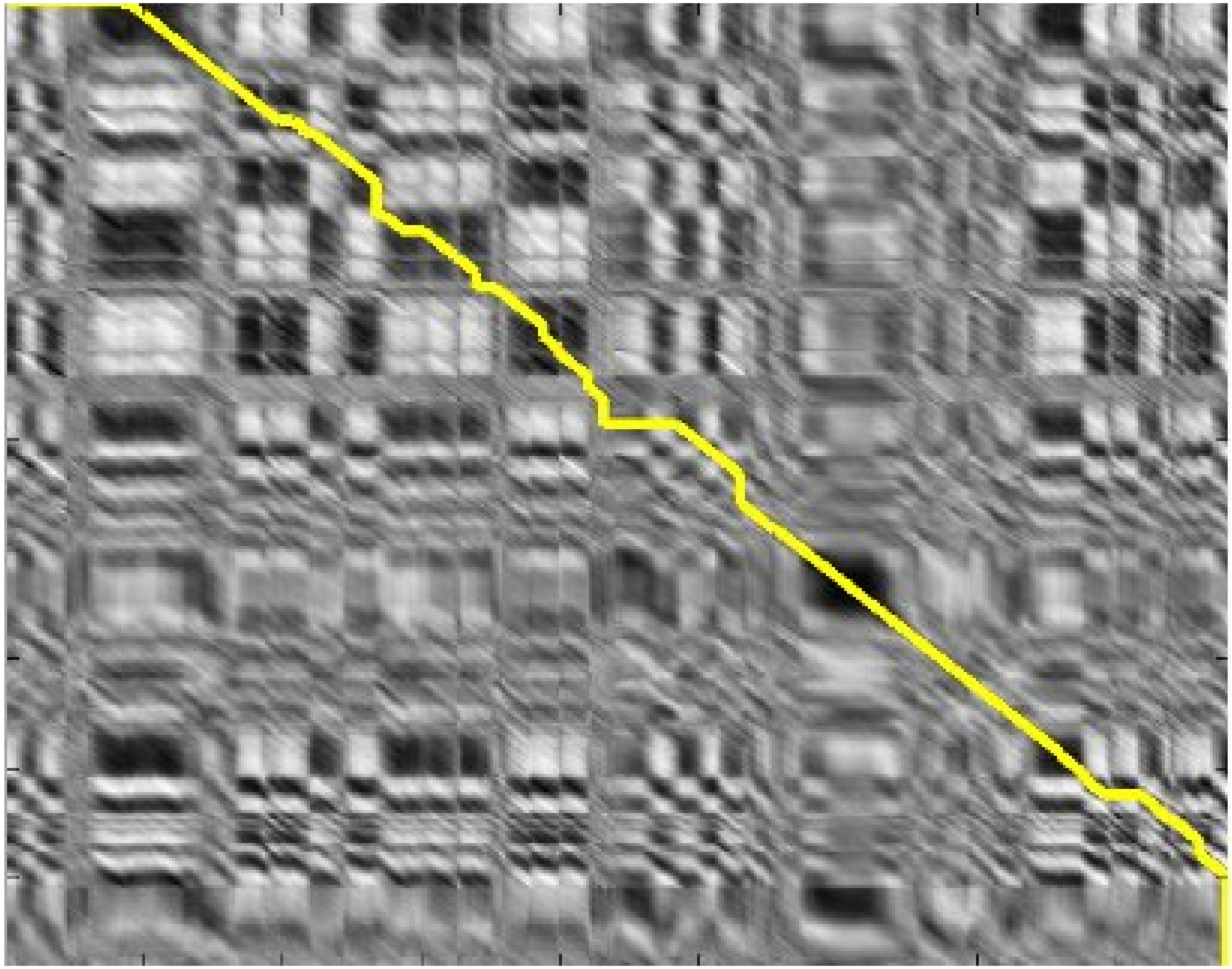
Dissimilarity Values

(1-NCC) or SSD

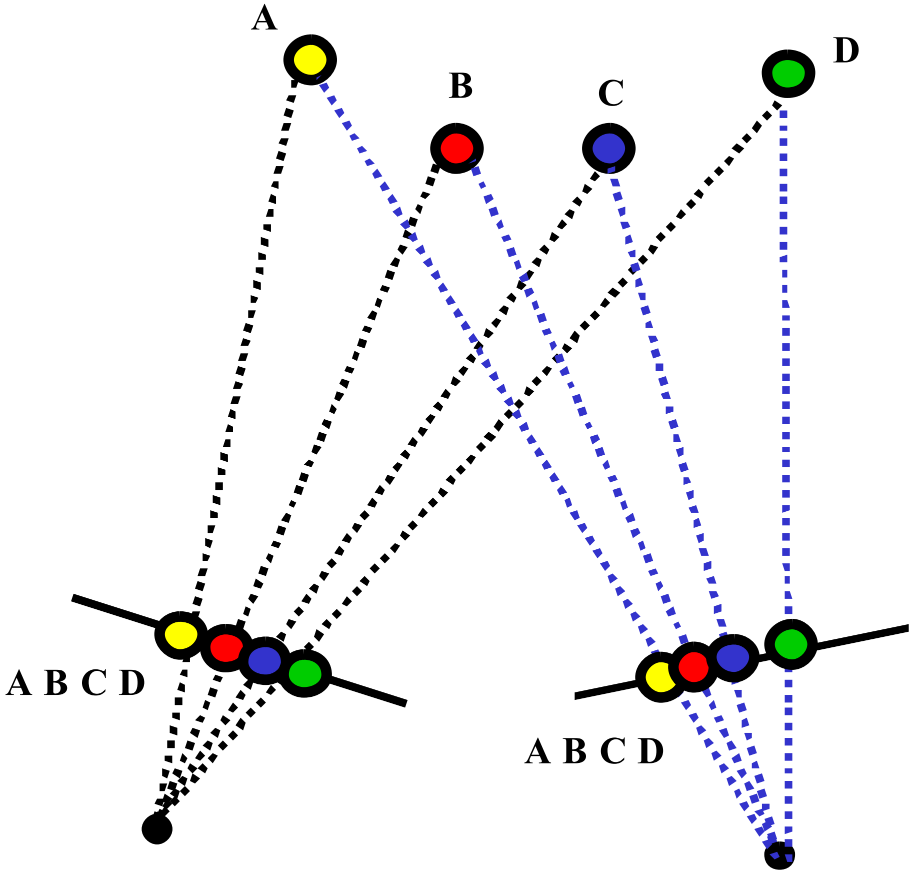
Disparity Space Image (DSI)



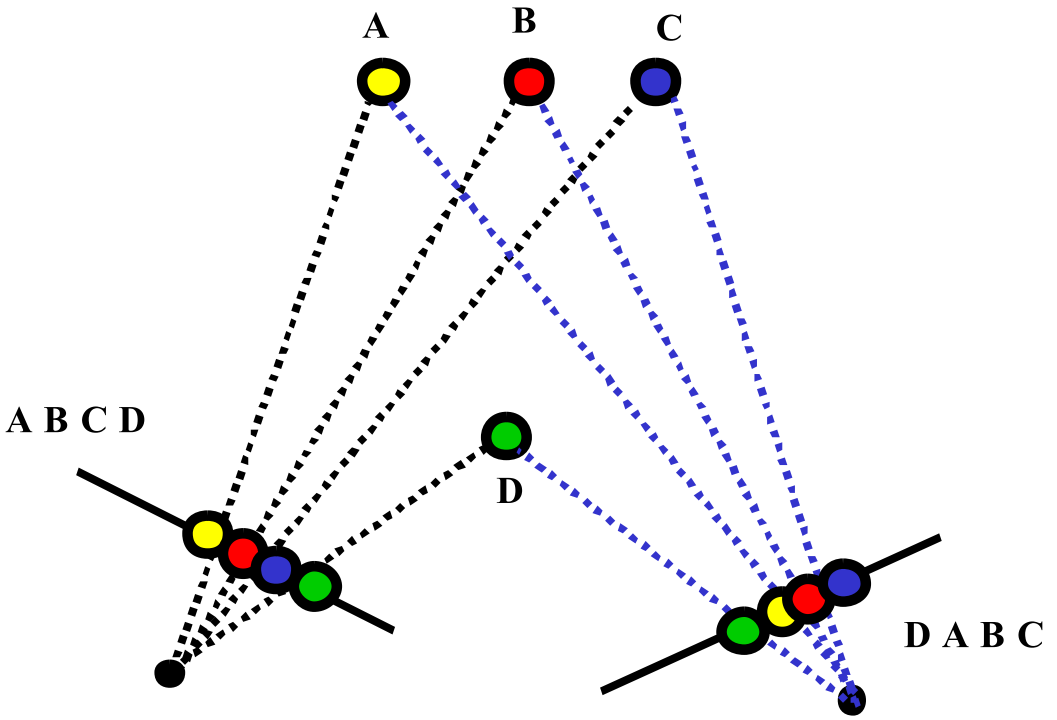
Representing the cost of all scanline correspondences



Ordering Constraint

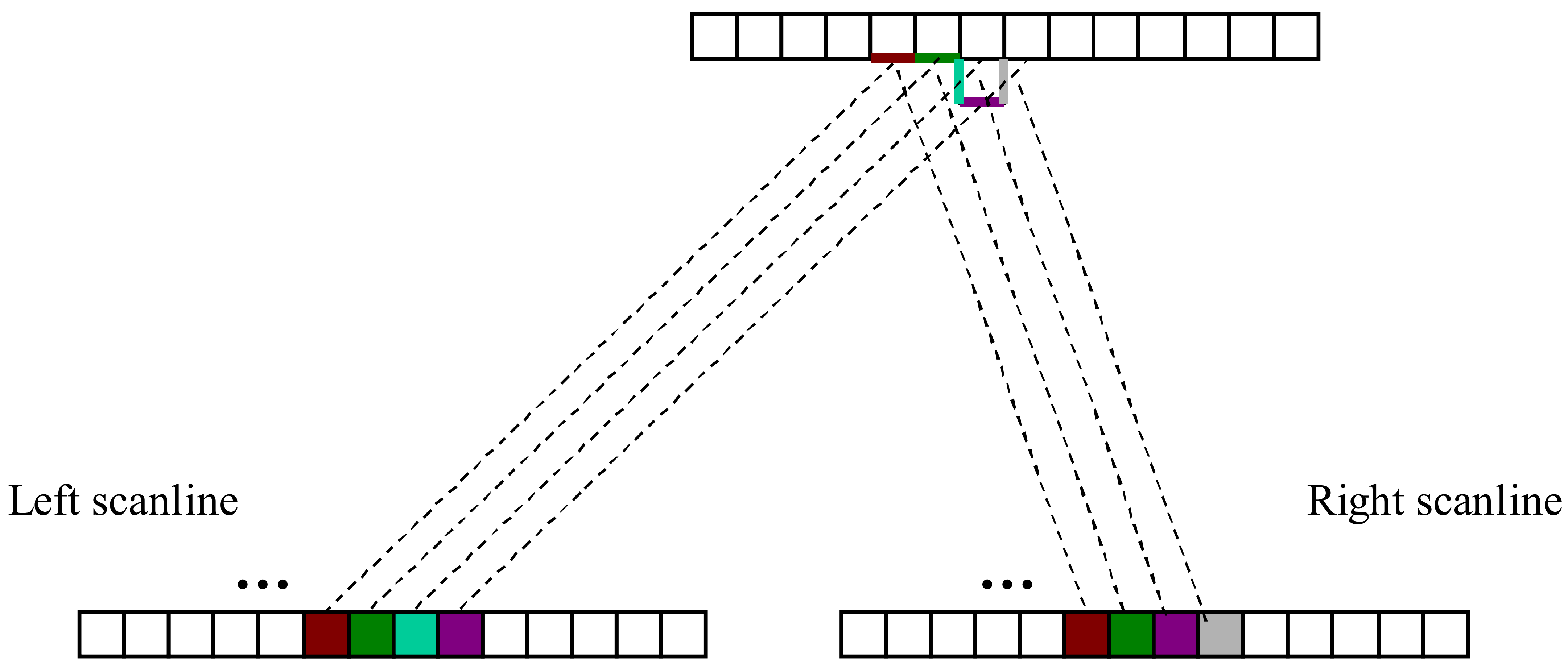


Ordering constraint…

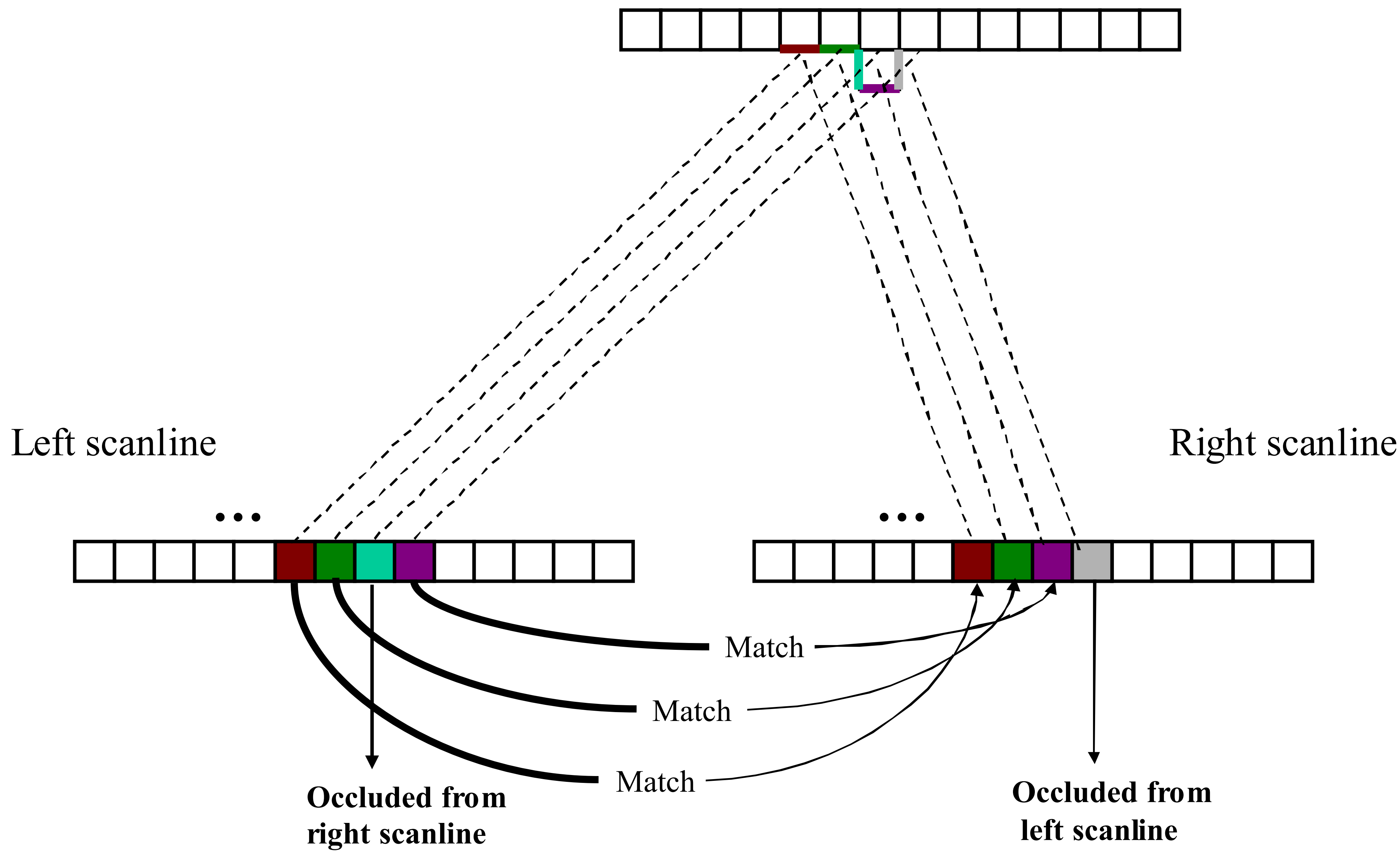


…and its failure

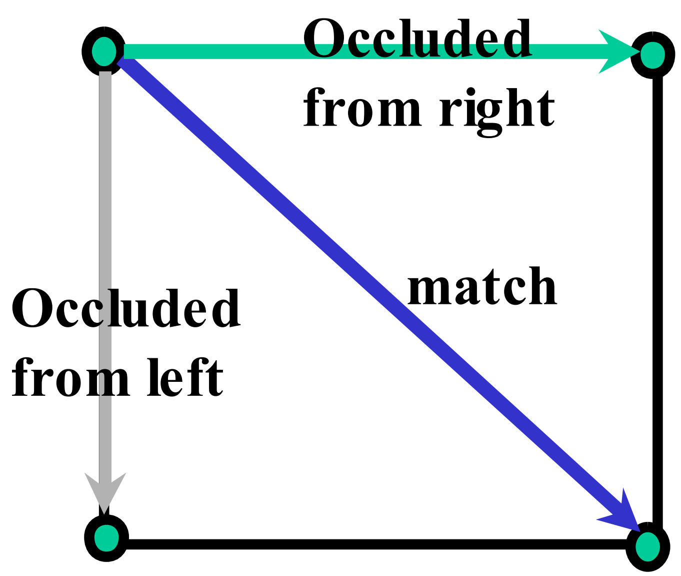
Occlusions

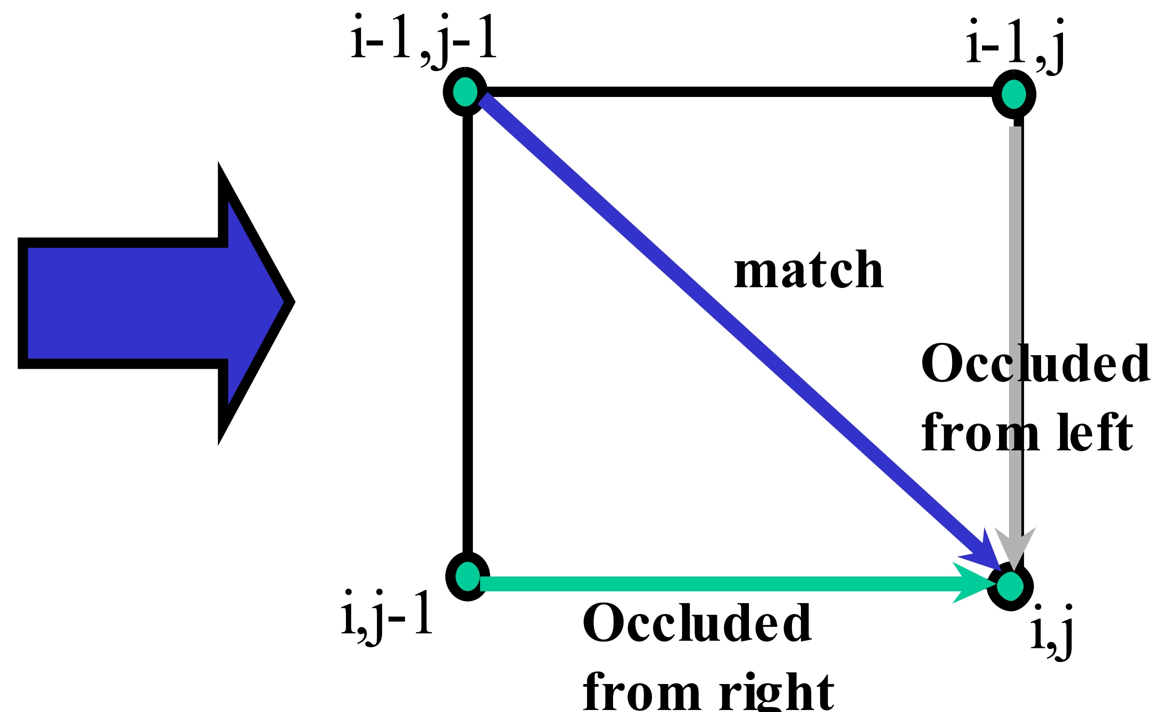


Occlusions


Left scanline
Right scanline
…
…
Match
Match
Match
Occluded from right scanline
Occluded from left scanline

Compute partial scanline costs





Three cases:

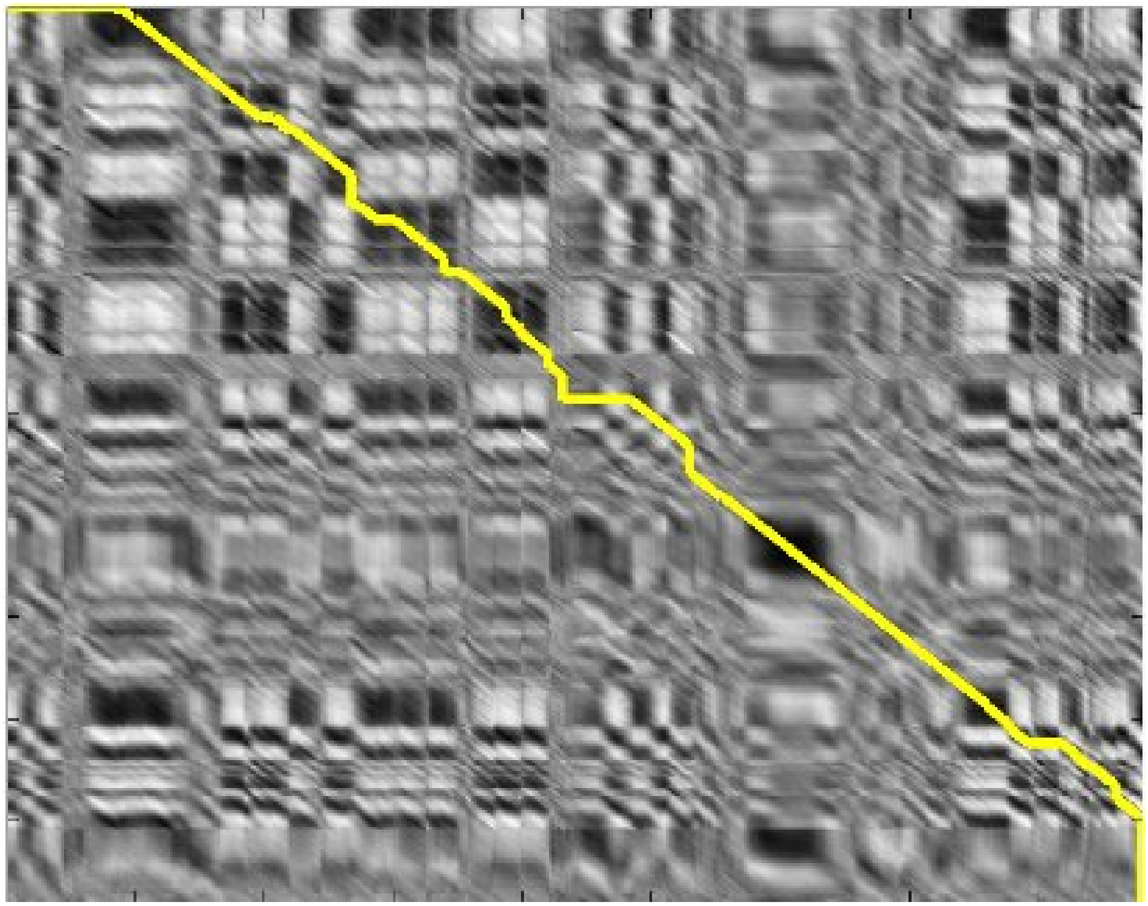
- Matching patches. Cost = dissimilarity score
- Occluded from right. Cost is some constant value.
- Occluded from left. Cost is some constant value.

$$
\begin{aligned}
C(i,j) = \min([\, & C(i-1,j-1) + \text{dissimilarity}(i,j) \\
& C(i-1,j) + \text{occlusionConstant}, \\
& C(i,j-1) + \text{occlusionConstant}]);
\end{aligned}
$$

Cox, Hingorani, Rao, Maggs, "A Maximum Likelihood Stereo Algorithm," Computer Vision and Image Understanding, Vol 63(3), May 1996, pp.542-567.

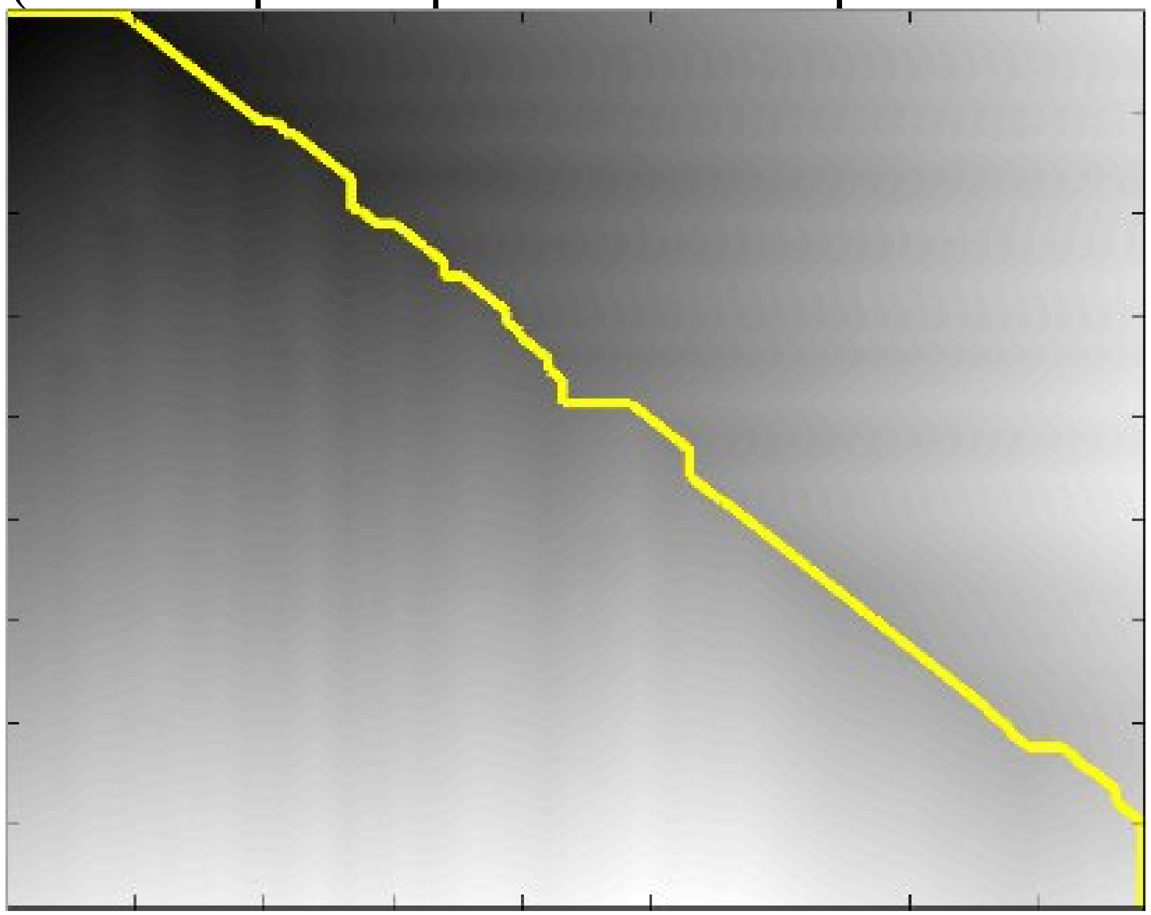
Dynamic Programming

DSI



DP cost matrix

(cost of optimal path from each point to END)

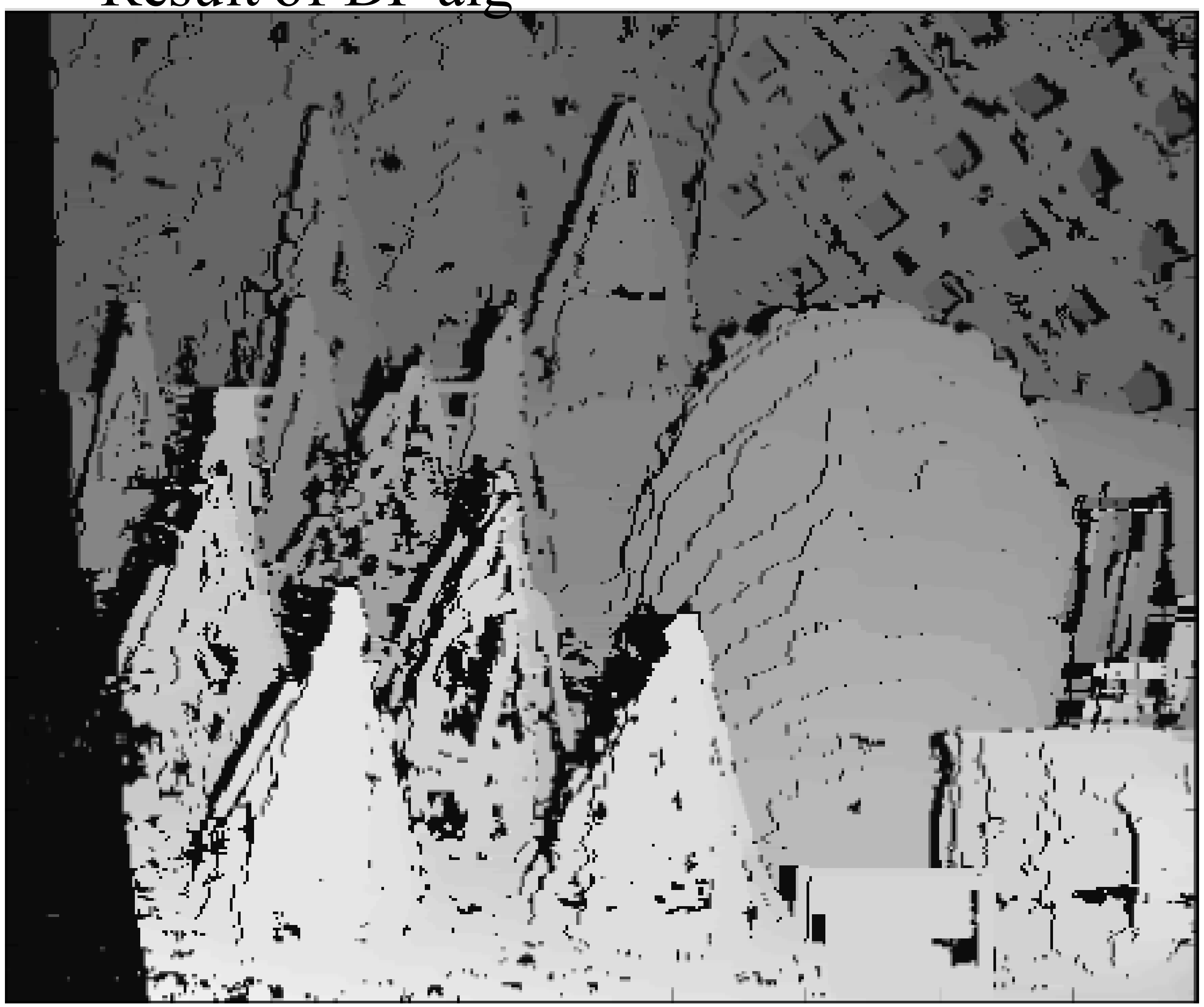


Each pixel in DSI is now marked with a disparity value or occlusion label

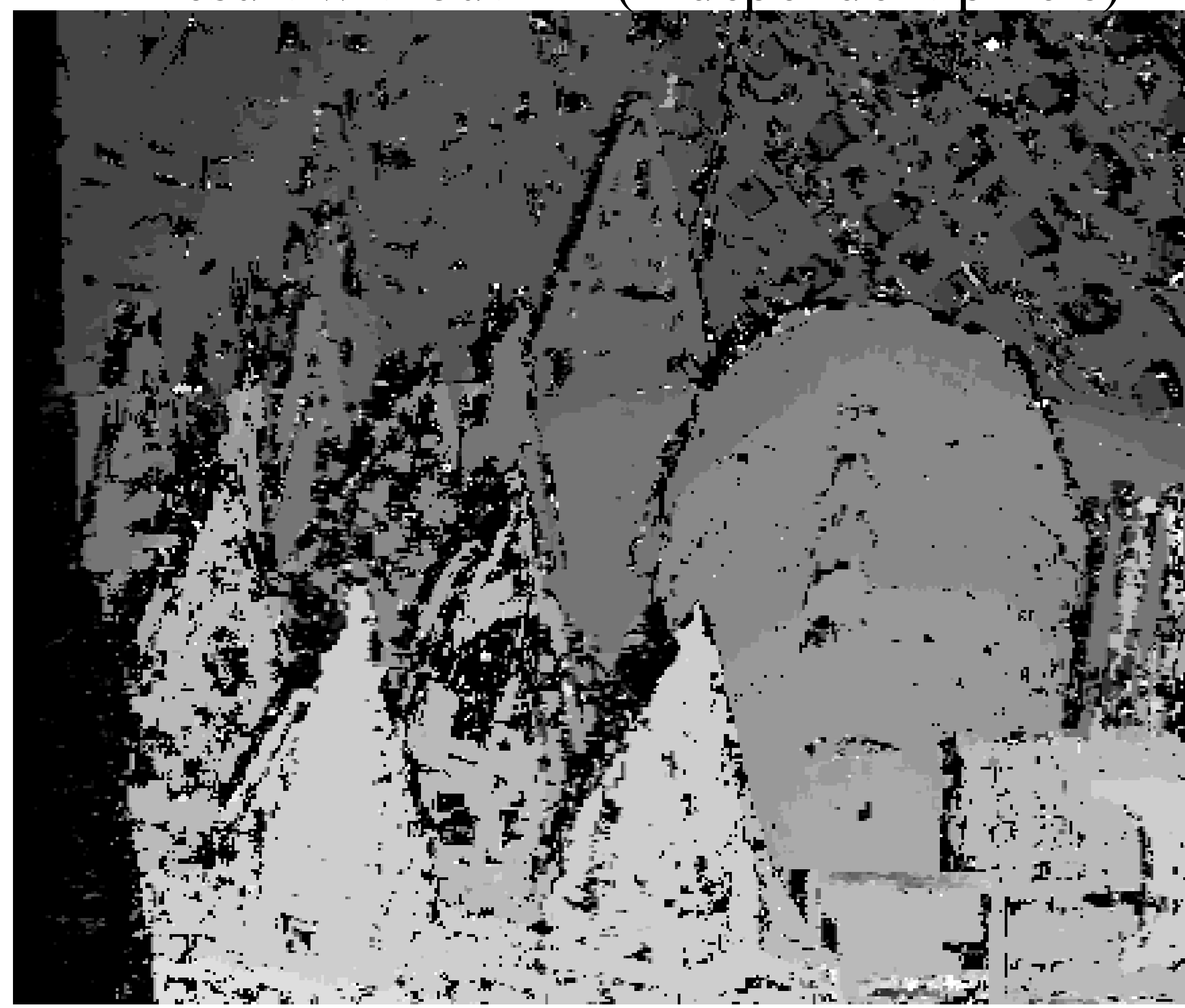
In practice, enforce upper bound on disparity by computing diagonal band of DSI

Results

Result of DP alg



Result without DP (independent pixels)



Result of DP alg. Black pixels = occluded.

Stereo evaluation: http://vision.middlebury.edu/stereo/

vision.middlebury.edu

stereo • mview • MRF • flow

Stereo Evaluation • Datasets • Code • Submit

Daniel Scharstein • Richard Szeliski

Welcome to the Middlebury Stereo Vision Page, formerly located at www.middlebury.edu/stereo. This website accompanies our taxonomy and comparison of two-frame stereo correspondence algorithms [1]. It contains:

- An on-line evaluation of current algorithms
- Many stereo datasets with ground-truth disparities
- Our stereo correspondence software
- An on-line submission script that allows you to evaluate your stereo algorithm in our framework

How to cite the materials on this website:
We grant permission to use and publish all images and numerical results on this website. If you report performance results, we request that you cite our paper [1]. Instructions on how to cite our datasets are listed on the datasets page. If you want to cite this website, please use the URL "**vision.middlebury.edu/stereo/**".

References:

[1] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7-42, April-June 2002.
Microsoft Research Technical Report MSR-TR-2001-81, November 2001.

Stereo—best algorithms

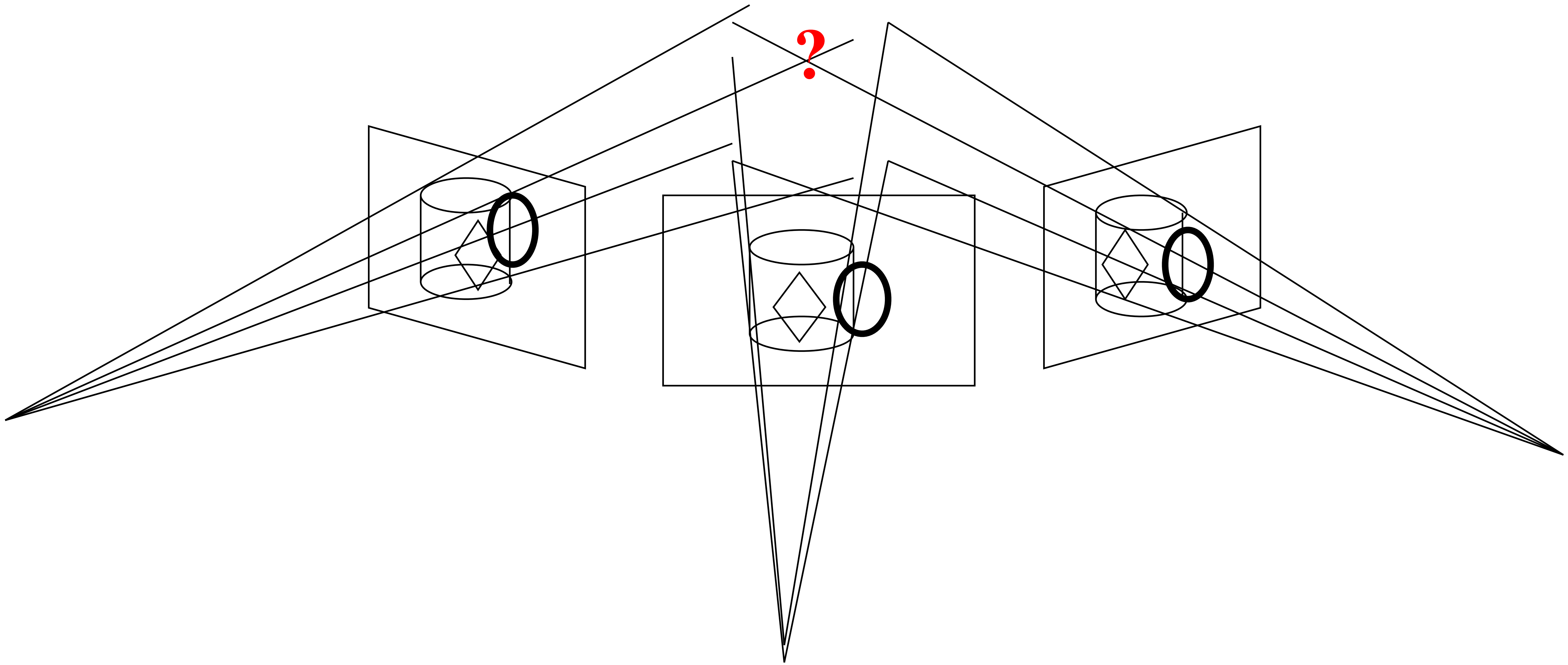
Error Threshold = 1		Sort by nonocc				Sort by all				Sort by disc	
Error Threshold...											
Algorithm	Avg. Rank	Tsukuba ground truth			Venus ground truth			Teddy ground truth			Cones ground truth
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc

Algorithm	Avg. Rank	Tsukuba nonocc	Tsukuba all	Tsukuba disc	Venus nonocc	Venus all	Venus disc	Teddy nonocc	Teddy all	Teddy disc	Cones nonocc	Cones all	Cones disc
AdaptingBP [17]	2.8	1.11 6	1.37 3	5.79 7	**0.10** 1	0.21 2	**1.44** 1	4.22 4	7.06 2	11.8 4	**2.48** 1	7.92 2	**7.32** 1
DoubleBP2 [35]	2.9	**0.88** 1	**1.29** 1	**4.76** 1	0.13 3	0.45 5	1.87 5	3.53 2	8.30 3	**9.63** 1	2.90 3	8.78 8	7.79 2
DoubleBP [15]	4.9	0.88 2	1.29 2	4.76 2	0.14 5	0.60 13	2.00 7	3.55 3	8.71 5	9.70 2	2.90 4	9.24 11	7.80 3
SubPixDoubleBP [30]	5.6	1.24 10	1.76 13	5.98 8	0.12 2	0.46 6	1.74 4	**3.45** 1	8.38 4	10.0 3	2.93 5	8.73 7	7.91 4
AdaptOvrSegBP [33]	9.9	1.69 22	2.04 21	5.64 6	0.14 4	**0.20** 1	1.47 2	7.04 14	11.1 7	16.4 11	3.60 11	8.96 10	8.84 10
SymBP+occ [7]	10.8	0.97 4	1.75 12	5.09 4	0.16 6	0.33 3	2.19 8	6.47 8	10.7 6	17.0 14	4.79 24	10.7 21	10.9 20
PlaneFitBP [32]	10.8	0.97 5	1.83 14	5.26 5	0.17 7	0.51 8	1.71 3	6.65 9	12.1 13	14.7 7	4.17 20	10.7 20	10.6 19
AdaptDispCalib [36]	11.8	1.19 8	1.42 4	6.15 9	0.23 9	0.34 4	2.50 11	7.80 19	13.6 21	17.3 17	3.62 12	9.33 12	9.72 15
Segm+visib [4]	12.2	1.30 15	1.57 5	6.92 18	0.79 21	1.06 18	6.76 22	5.00 5	**6.54** 1	12.3 5	3.72 13	8.62 6	10.2 17
C-SemiGlob [19]	12.3	2.61 29	3.29 24	9.89 27	0.25 12	0.57 10	3.24 15	5.14 6	11.8 8	13.0 6	2.77 2	8.35 4	8.20 5
SO+borders [29]	12.8	1.29 14	1.71 9	6.83 15	0.25 13	0.53 9	2.26 9	7.02 13	12.2 14	16.3 9	3.90 15	9.85 16	10.2 18
DistinctSM [27]	14.1	1.21 9	1.75 11	6.39 11	0.35 14	0.69 16	2.63 13	7.45 18	13.0 17	18.1 19	3.91 16	9.91 18	8.32 7
CostAggr+occ [39]	14.3	1.38 17	1.96 17	7.14 19	0.44 16	1.13 19	4.87 19	6.80 11	11.9 10	17.3 16	3.60 10	8.57 5	9.36 13
OverSegmBP [26]	14.5	1.69 23	1.97 18	8.47 24	0.51 18	0.68 15	4.69 18	6.74 10	11.9 12	15.8 8	3.19 8	8.81 9	8.89 11
SegmentSupport [28]	15.1	1.25 11	1.62 7	6.68 13	0.25 11	0.64 14	2.59 12	8.43 24	14.2 22	18.2 20	3.77 14	9.87 17	9.77 16
RegionTreeDP [18]	15.7	1.39 19	1.64 8	6.85 16	0.22 8	0.57 10	1.93 6	7.42 17	11.9 11	16.8 13	6.31 30	11.9 27	11.8 23
EnhancedBP [24]	16.6	0.94 3	1.74 10	5.05 3	0.35 15	0.86 17	4.34 17	8.11 22	13.3 19	18.5 22	5.09 27	11.1 23	11.0 21

Outline

- 2-view geometry
- essential matrix, fundamental matrix
- properties
- estimation
- stereo
- multiview stereo

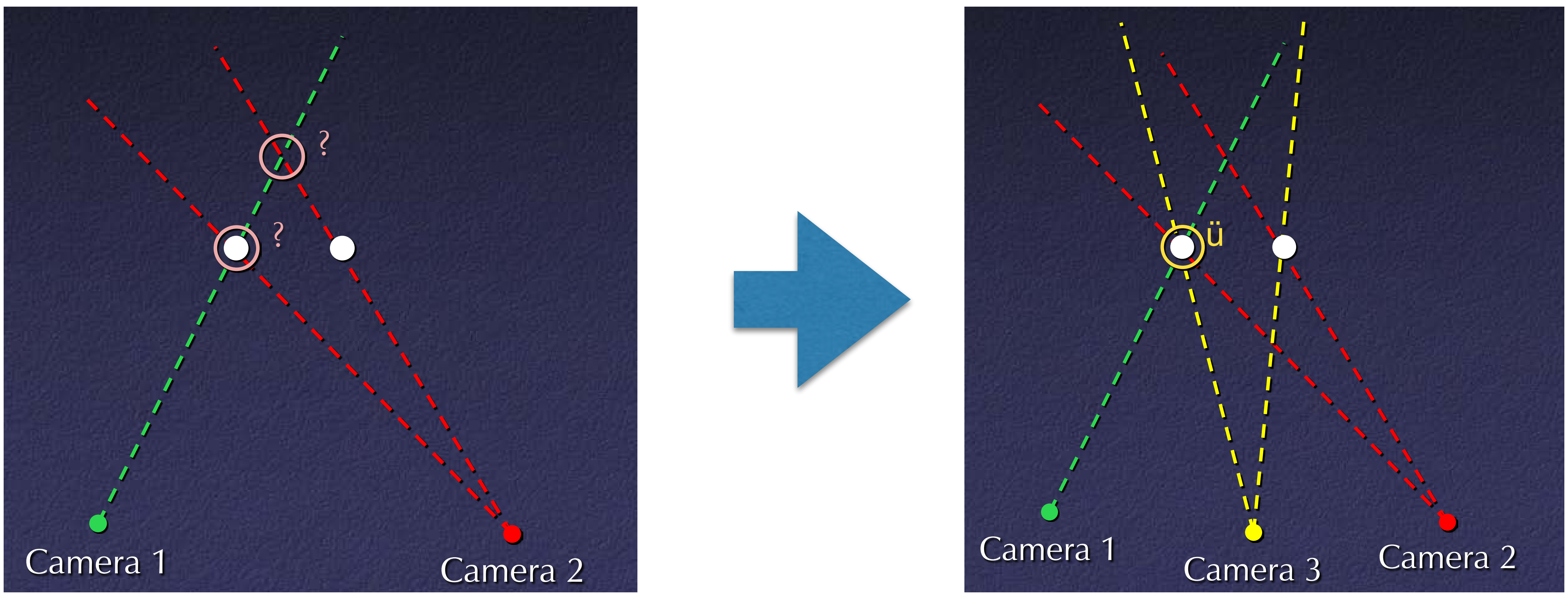
Dense multi view stereo



- Reconstruct the 3D position of the points corresponding to (all the) pixels in a set of images.
- Key assumption: We know the relative position, orientation, K, of all the cameras.
- **Number of cameras >> 2**

Trinocular stereo (version 0)

1. Pick 2 views, find correspondences
2. For each matching pair, reconstruct 3D point
3. If can't find correspondence near projected location, reject



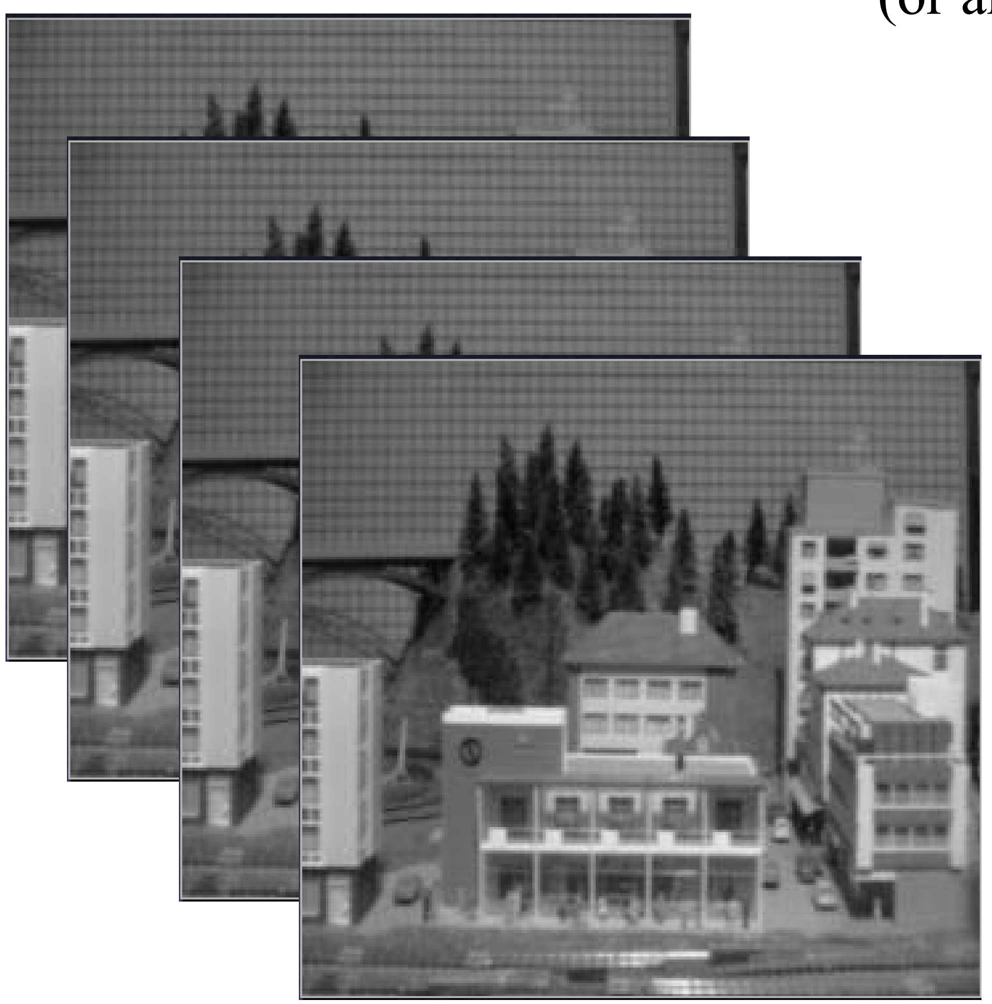
Version 1: generalize 3x3 fundmamental matrix to a 3x3x3 trifocal tensor

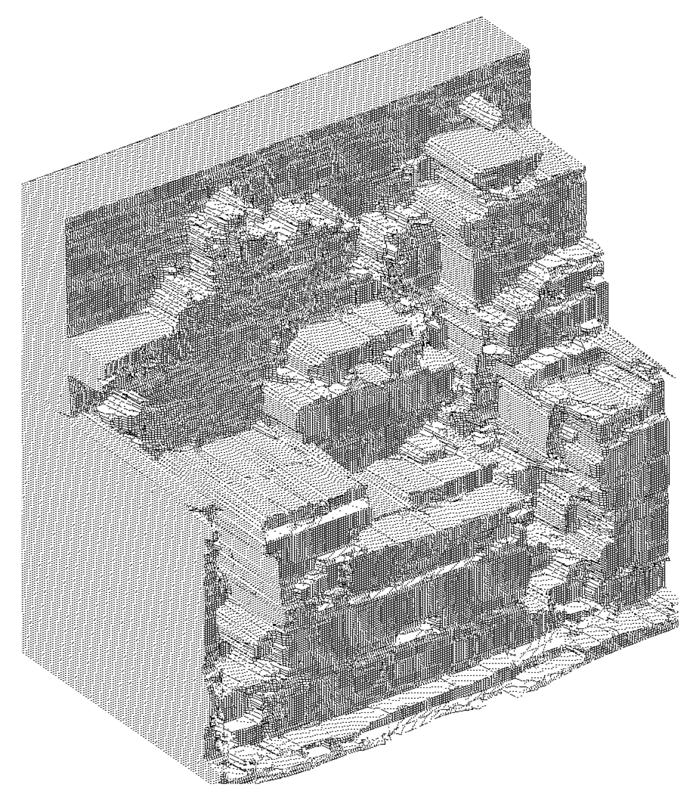
(constraints points and lines across 3 images)

Multiview stereo (version 0)

-Pick one reference view
-For each point and for each candidate depth

- keep depths with low SSD error in all other views
  (or any *photoconsistency* measure)



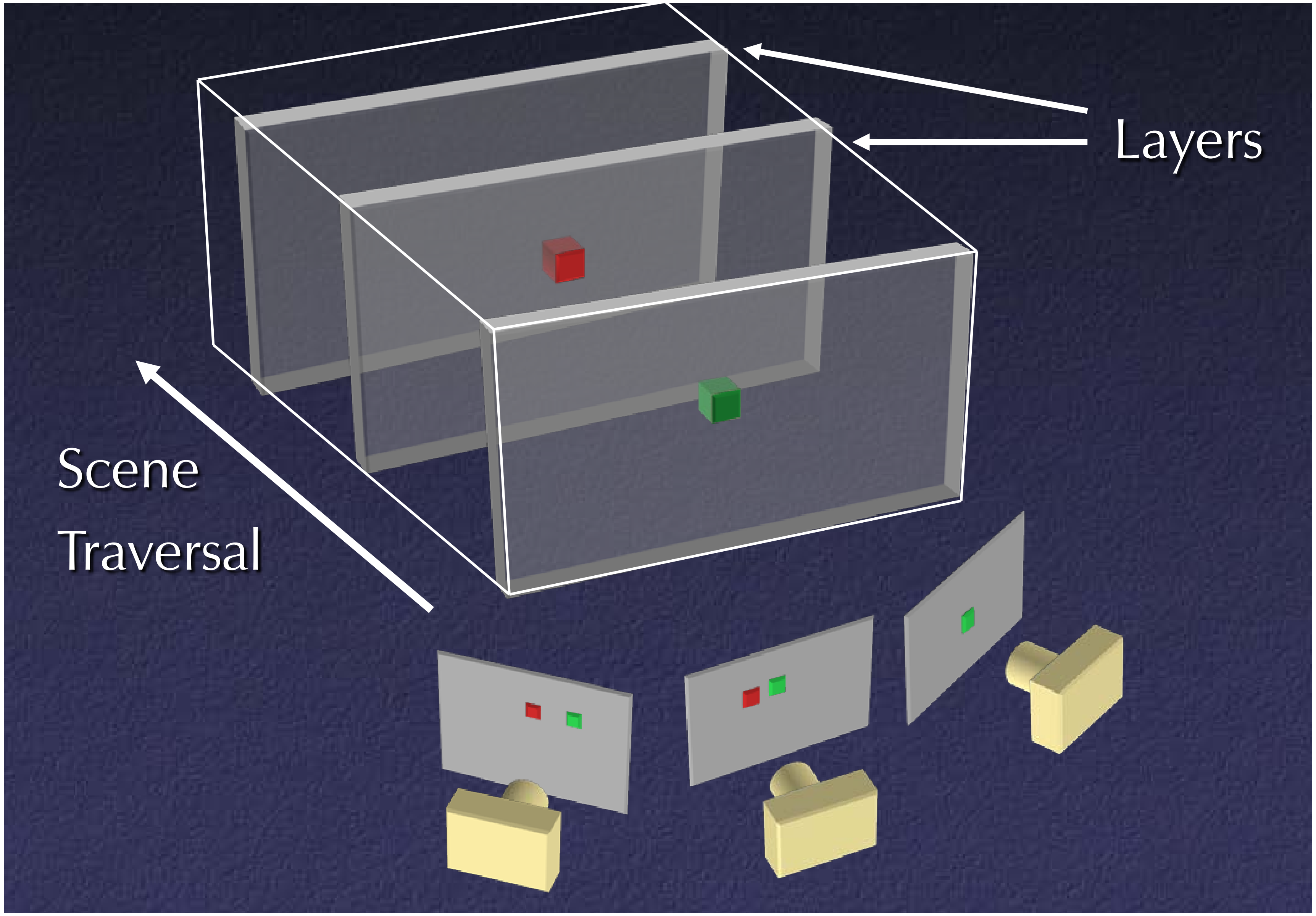


Problem: not all points are visible in all other views (occlusion and visibility major nuisance!)

Multiview stereo (version 1)

Hypothesize depths in a “smart” order where occluding points are found *first*

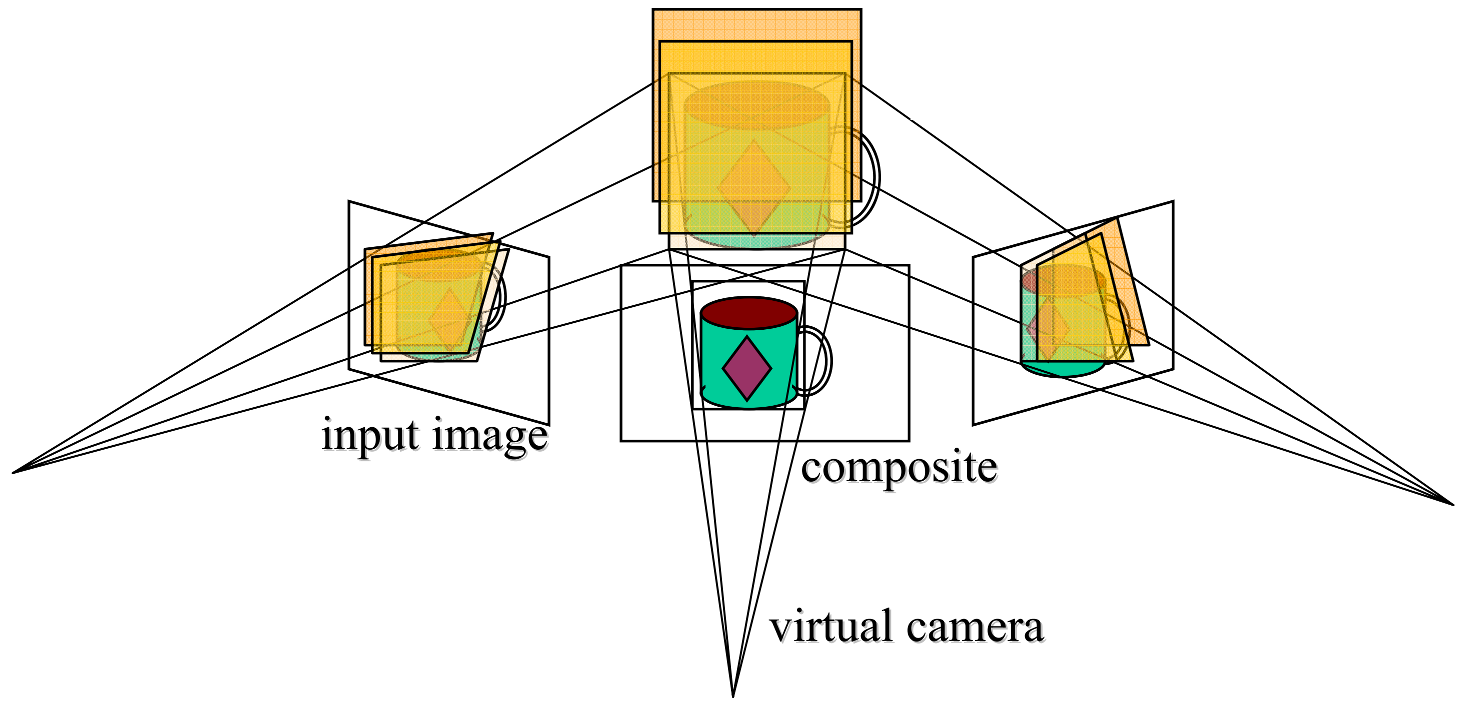
Use knowledge of occluding points to smartly select view for photoconsistency check



Store photoconsistent color in a 3D voxel grid (don’t need a reference image)

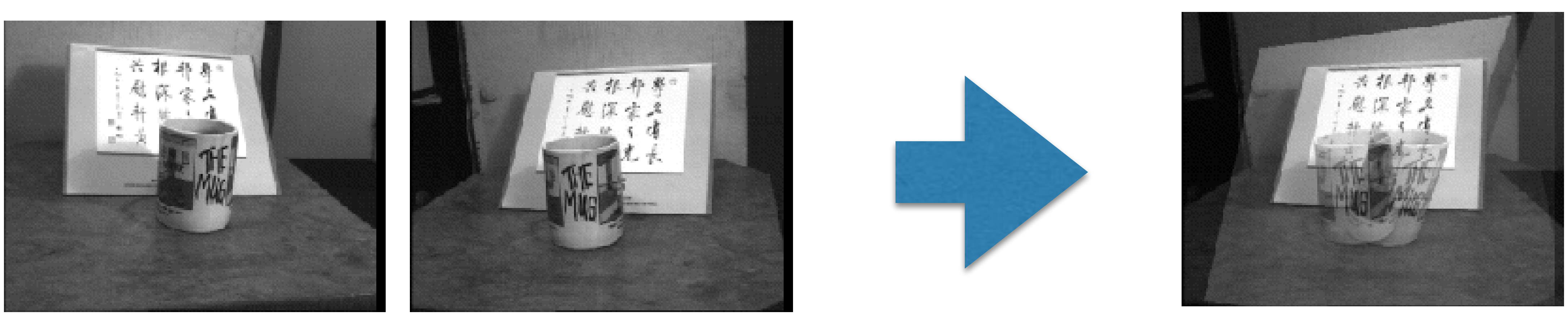
Reconstuct shape *and* appearance

Speedup: plane sweeps



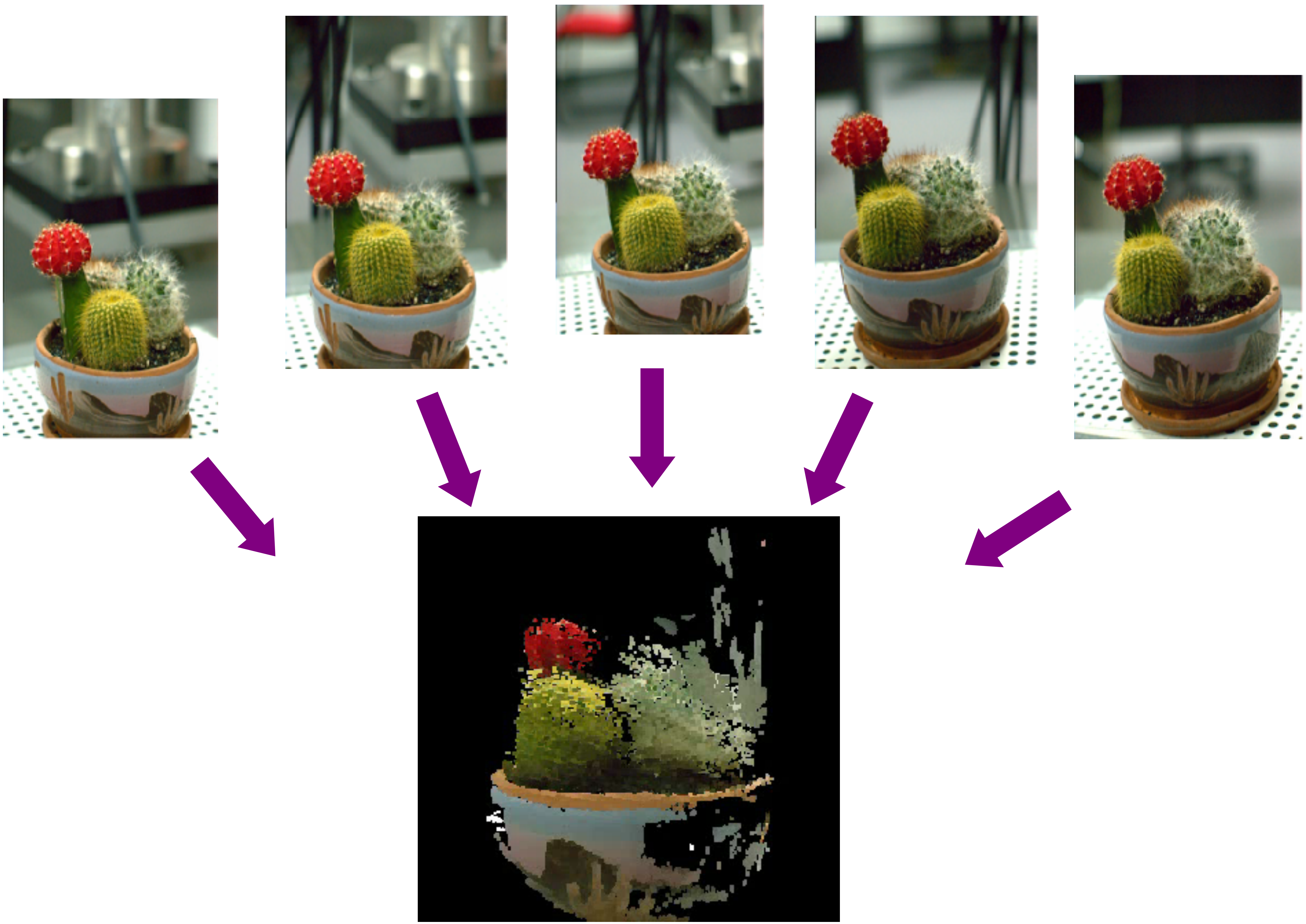
Validate voxels in a plane by computing their appearance in a virtual view using all N cameras

Keep track of image-specific occlusion masks

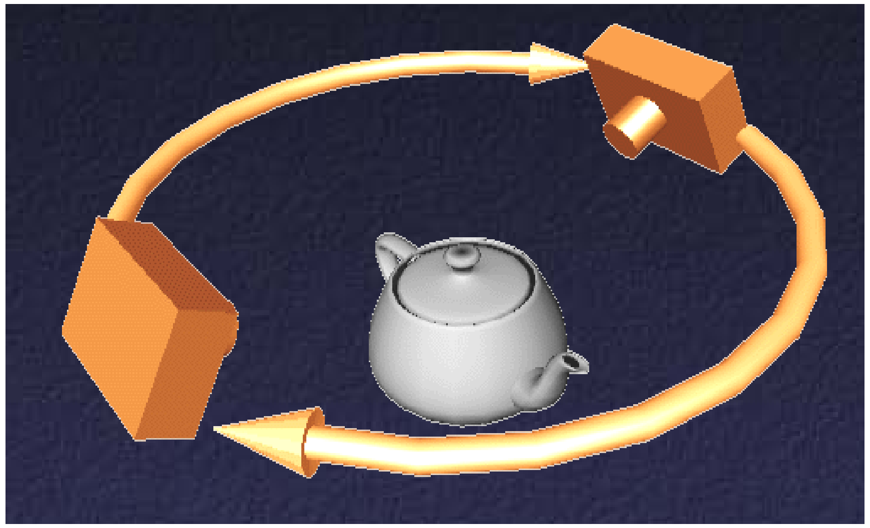


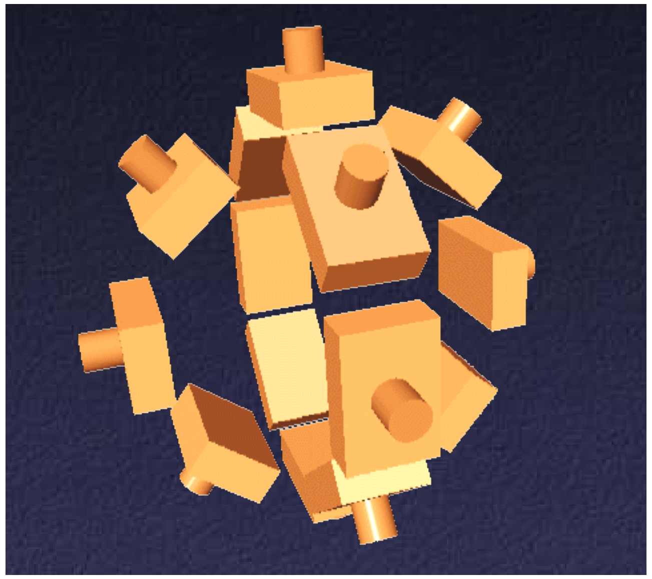
What is the transformation that warps image N to virtual view?

Voxel coloring



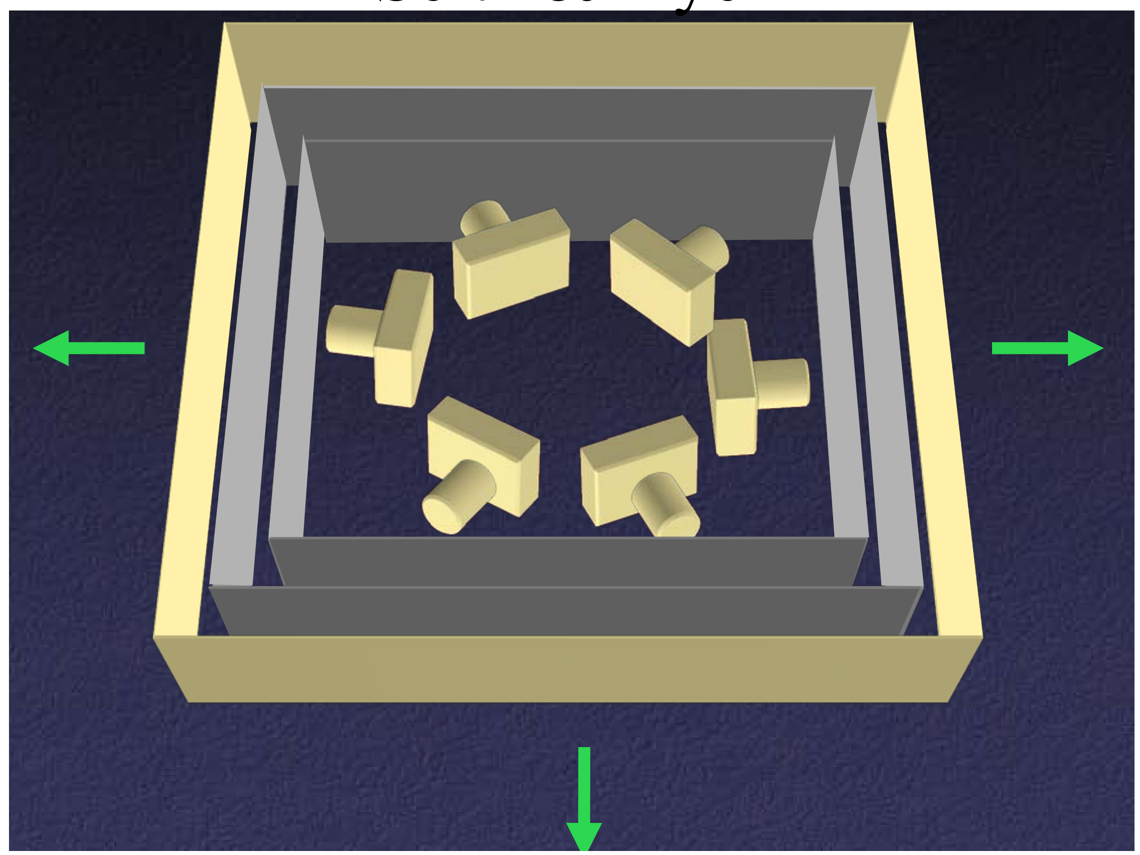
What about other camera steups?





Panoramic depth ordering

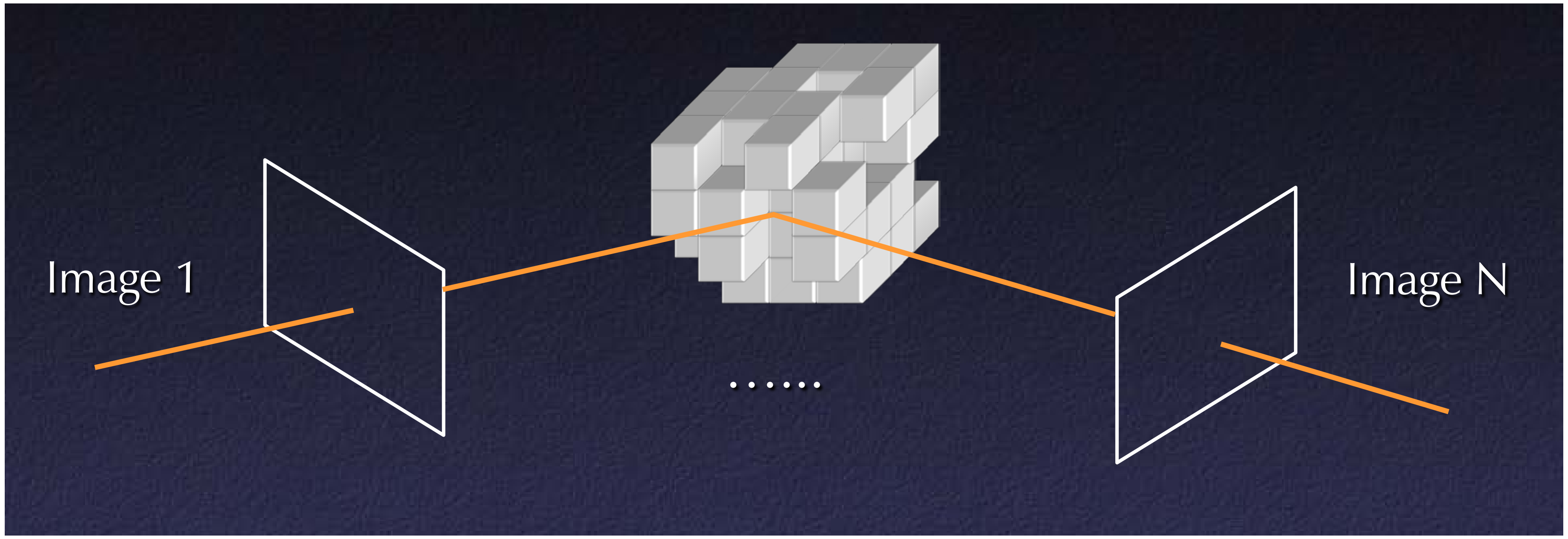
Seitz & Dyer



Layers radiate inwardly/outwardly

Space carving

Kutulakos & Seitz



Initialize voxel grid to all '1's

Repeatedly choose a voxel on current surface:

Project to visible images

Carve out if not photoconsistent

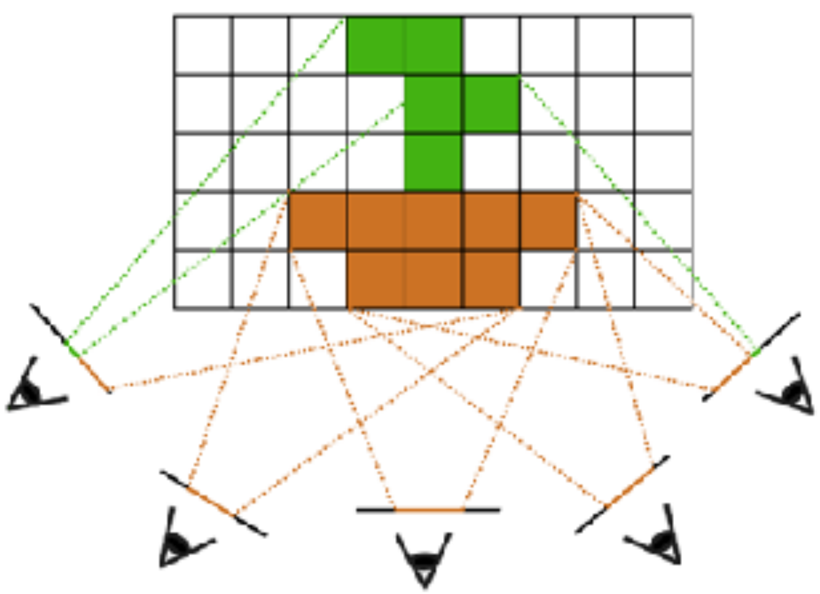
Convergence

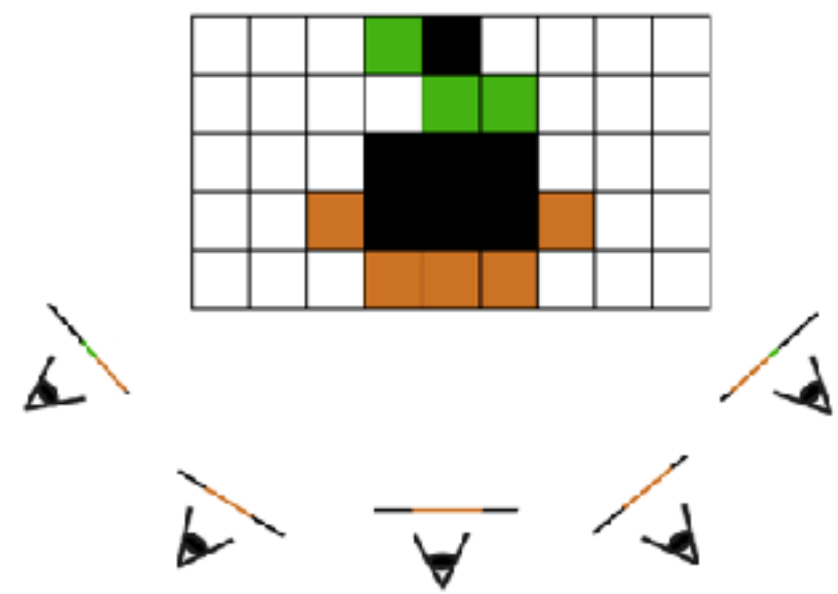
Consistency Property

- The resulting shape is photo-consistent
  > all inconsistent points are removed

Convergence Property

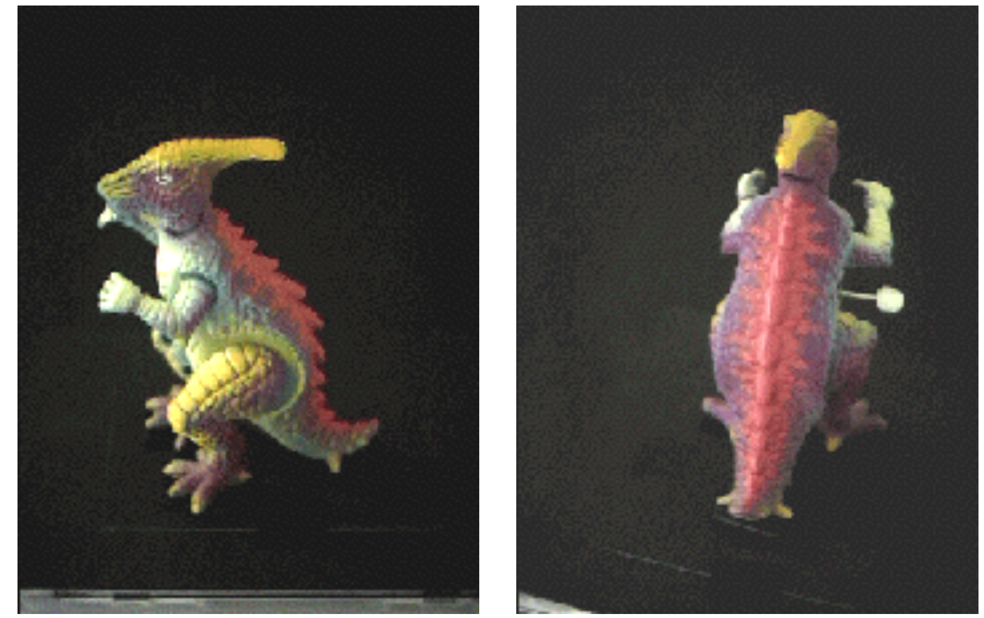
- Carving converges to a non-empty shape
  > a point on the true scene is *never* removed



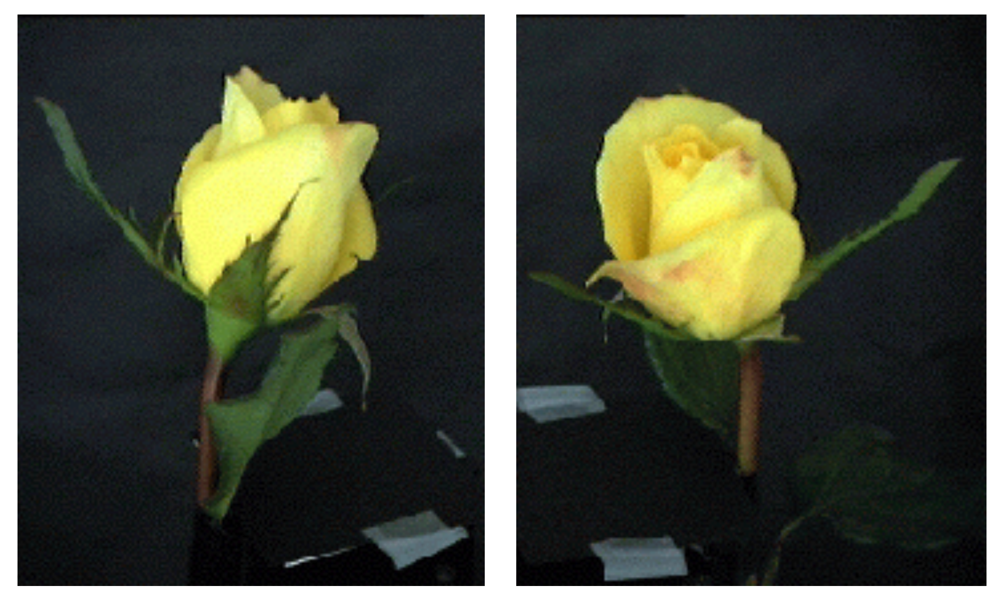


Calibrated Image Acquisition

Calibrated Turntable



Selected Dinosaur Images

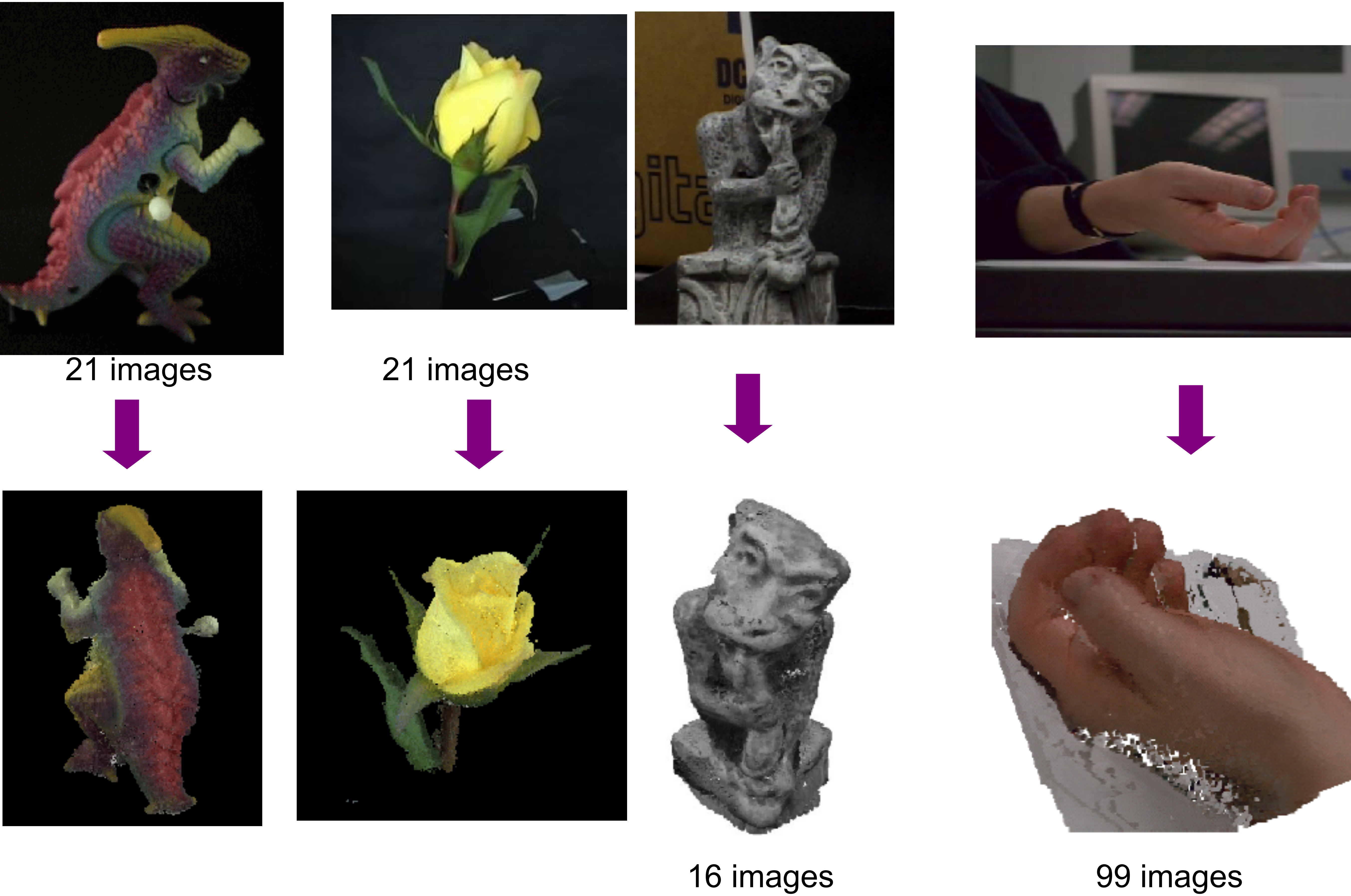


Selected Flower Images

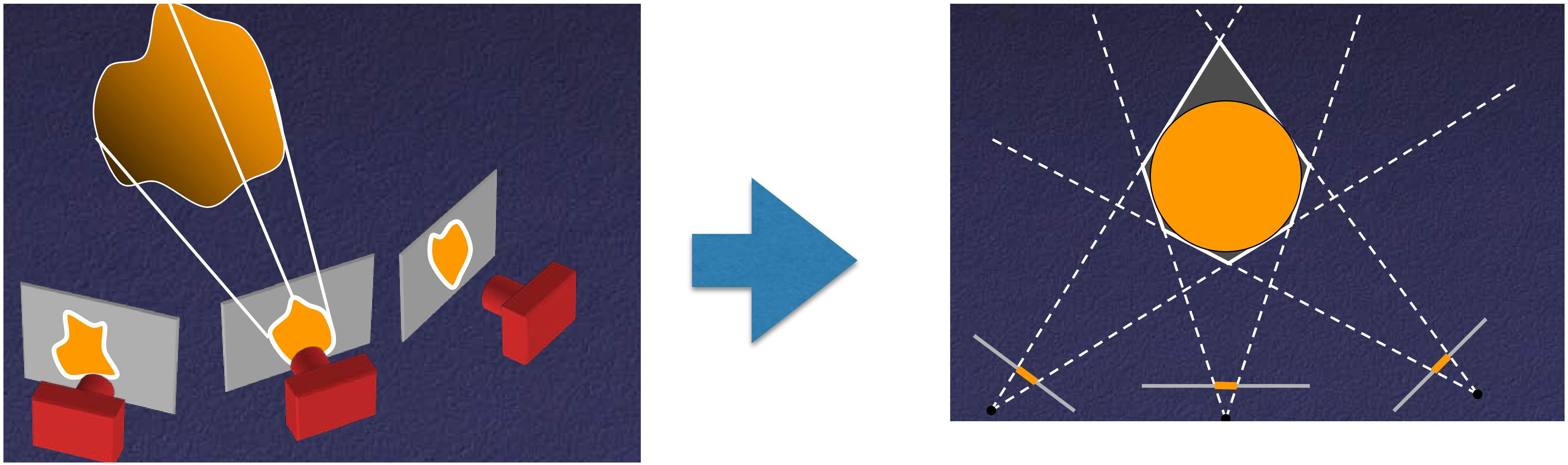
Voxel Coloring Results

Dinosaur Reconstruction
72 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI

Flower Reconstruction
70 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI


21 images
21 images
16 images
99 images

Silhoette carving



Backproject binary silhouettes and find intersection

In limit of infinite cameras, this will produce convex hull reconstruction of object

Outline

- essential matrix, fundamental matrix

  (point-to-line correspondence, SVD properties)
- stereo

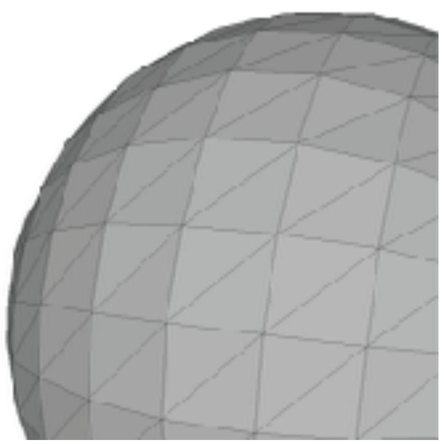
  (variational, discrete graph labelling, dynamic programming)
- multiview

  (volumetric models, visibility reasoning, patch-based methods)

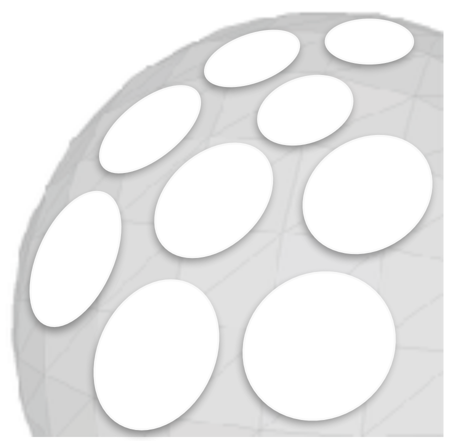
Long-standing leader

Accurate, Dense, and Robust Multi-View Stereopsis

Yasutaka Furukawa and Jean Ponce, *Fellow, IEEE*

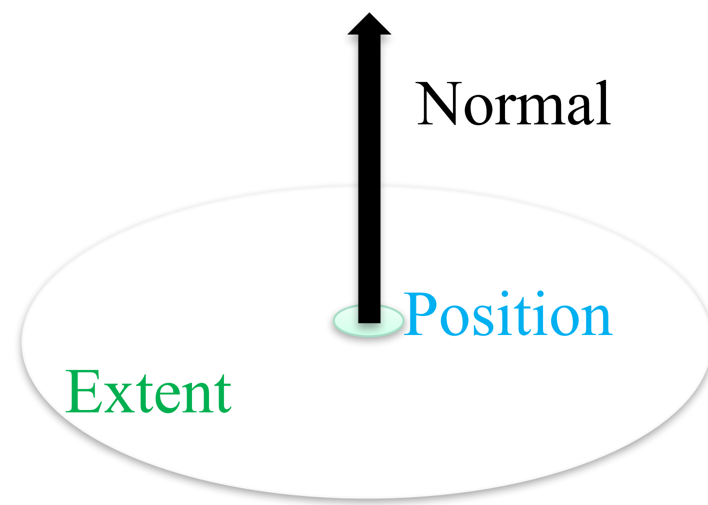


Mesh



Patch

Easier to approximate
surface by dense set of
local planar patches



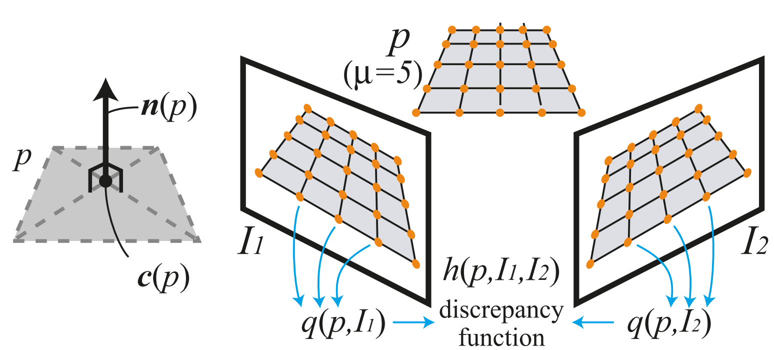
Patch-based Multiview Stereo (PMVS)

Pipeline: feature detection

Find sparse matches over pairs of images (using interest points + matching)
Triangulate to find sparse 3D points {p}

Pipeline: patch optimization

At each point p, estimate normal N(p) and visibility $V_i(p)$ in each image using photoconsistency check (NCC over ~9x9 pixels)



Pipeline: patch expansion

Expand set of points {p} by looking for hypothesizing 2D neighbors in visible images, backprojecting, and verifying photoconsistency

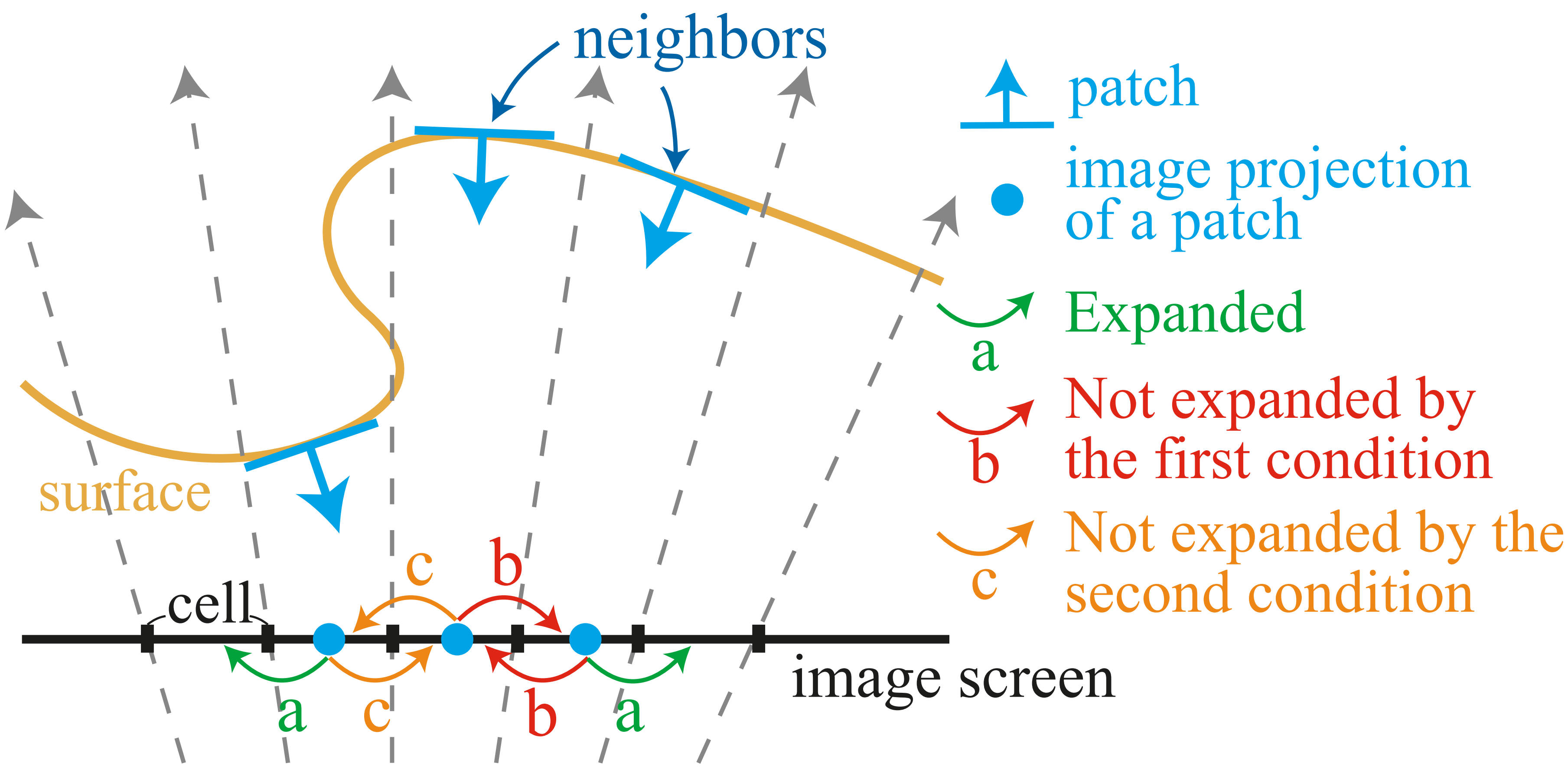


Fig. 5. (a) Given an existing patch, an expansion procedure is performed to generate new ones for the neighboring empty image cells in its visible images. The expansion procedure is not performed for an image cell (b) if there already exists a neighboring patch reconstructed there, or (c) if there is a depth discontinuity when viewed from the camera. See text for more details.

Pipeline: filter out outlier patches

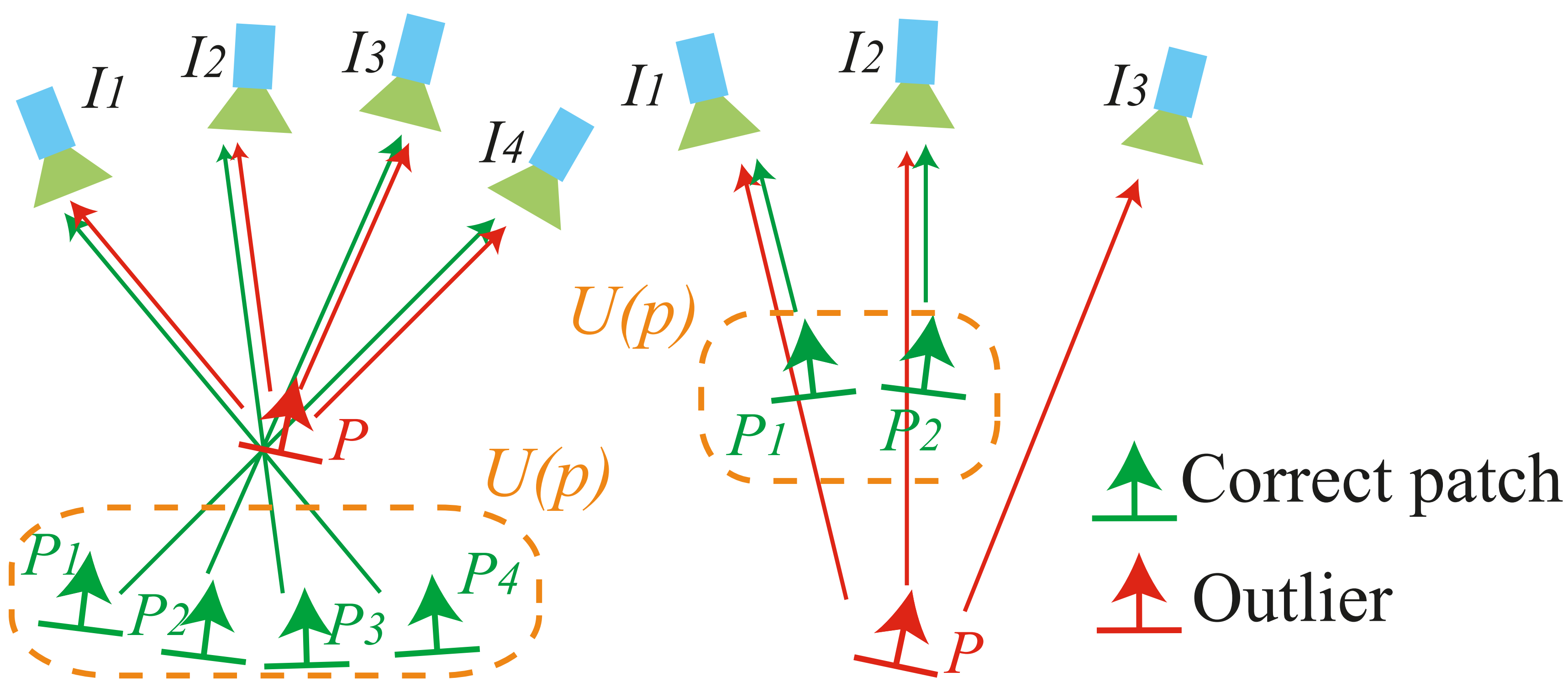
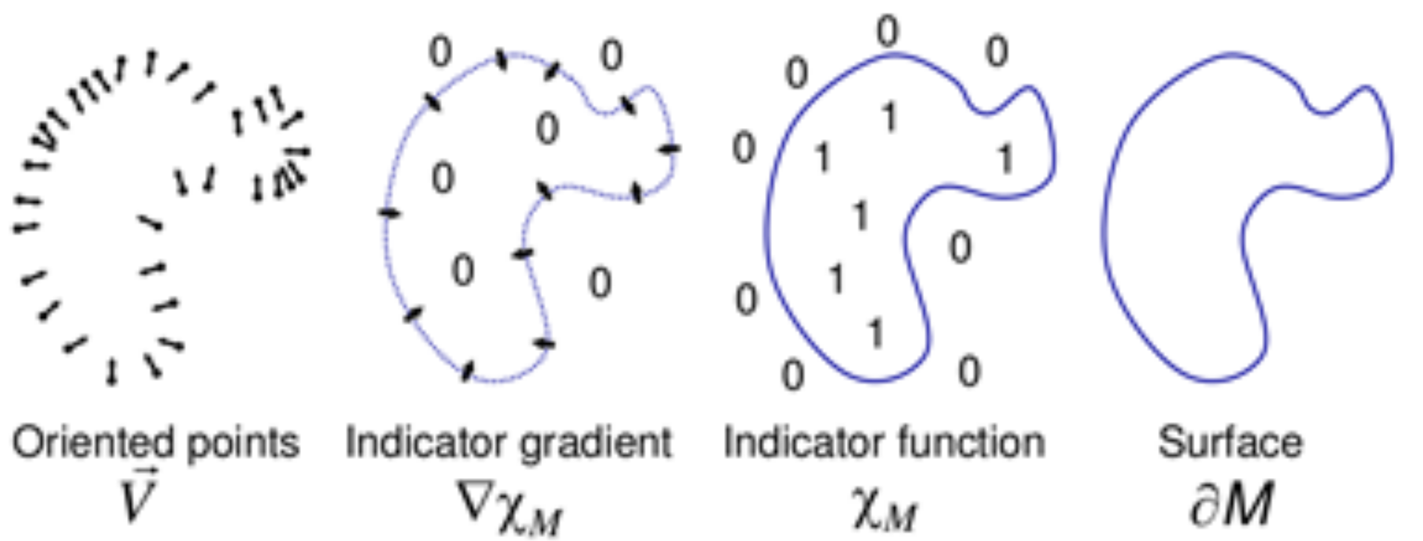


Fig. 7. The first filter enforces global visibility consistency to remove outliers (red patches). An arrow pointing from p_i to I_j represents a relationship $I_j \in V(p_i)$. In both cases (left and right), $U(p)$ denotes a set of patches that is inconsistent in visibility information with p.

Pipeline: construct mesh

Convert set of 3D patches (*surfel* model) into polygonal mesh



Represent surface implicitly using a volumetric signed distance function
Solve differential equation that equates gradients of function to normals

Results

Outline

- essential matrix, fundamental matrix

  (point-to-line correspondence, SVD properties)
- stereo

  (variational, discrete graph labelling, dynamic programming)
- multiview

  (volumetric models, visibility reasoning, patch-based methods)