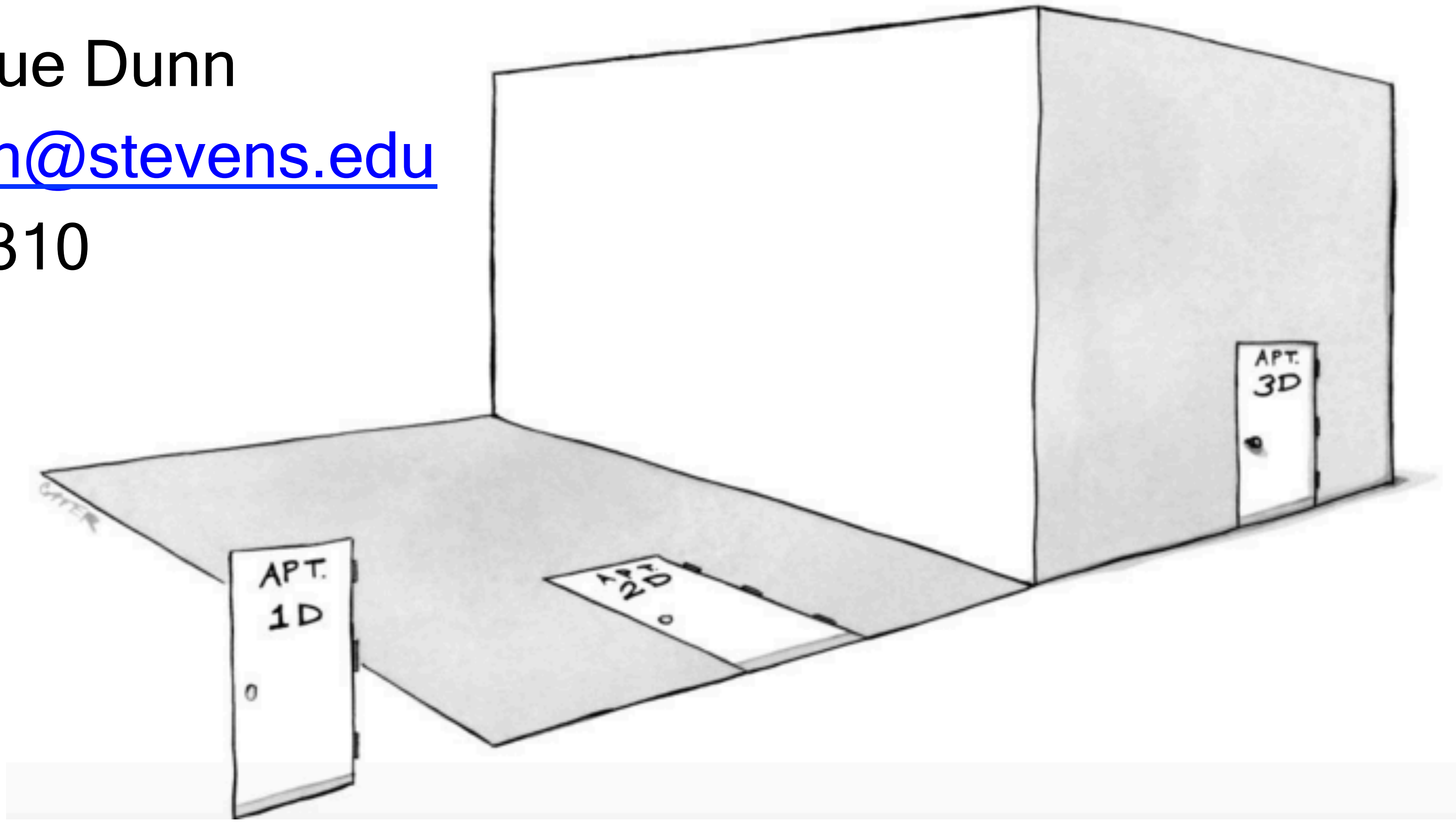
CS 532: 3D Computer Vision
Lecture 3

Enrique Dunn
edunn@stevens.edu
Lieb 310



Course TA

Andy Wiggins

<awiggins@stevens.edu>

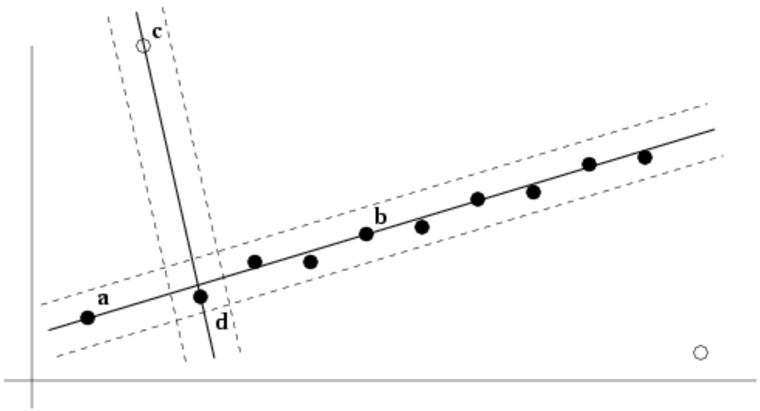
Office hours:
Lieb lounge on Wednesdays &Thursdays 2pm-4pm

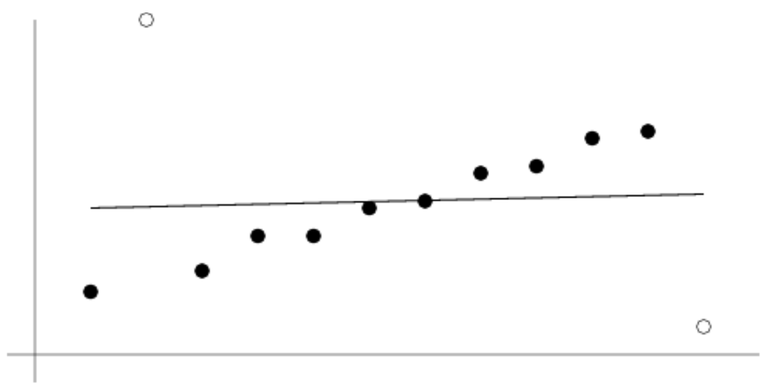
RANSAC

Slides by R. Hartley, A. Zisserman and M. Pollefeys

Robust Estimation

- What if set of matches contains gross outliers?





RANSAC

<u>Objective</u>

Robust fit of model to data set S which contains outliers

<u>Algorithm</u>

(i) Randomly select a sample of *s* data points from S and instantiate the model from this subset.

(ii) Determine the set of data points S_i which are within a distance threshold *t* of the model. The set S_i is the consensus set of samples and defines the inliers of S.

(iii) If the subset of S_i is greater than some threshold *T*, re-estimate the model using all the points in S_i and terminate

(iv) If the size of S_i is less than *T*, select a new subset and repeat the above.

(v) After *N* trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i

How Many Samples?

Choose N so that, with probability p, at least one random sample is free from outliers. e.g. p=0.99

$$\left(1-\left(1-e\right)^{s}\right)^{N}=1-p$$

$$N=\log\left(1-p\right)/\log\left(1-\left(1-e\right)^{s}\right)$$

Sampling Inlier Data point $1-e$

Sampling All-Inlier Set $\left(1-e\right)^{s}$

Sampling Contaminated Set $1-\left(1-e\right)^{s}$

	proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Acceptable Consensus Set

- Typically, terminate when inlier ratio reaches expected ratio of inliers

$$T = (1 - e)n$$

Adaptively Determining the Number of Samples

e is often unknown a priori, so pick worst case, e.g. 50%, and adapt if more inliers are found, e.g. 80% would yield e=0.2

- N=∞, *sample_count* =0
- While N >*sample_count* repeat
  - Choose a sample and count the number of inliers
  - Set e=1-(number of inliers)/(total number of points)
  - Recompute N from e
  - Increment the *sample_count* by 1
- Terminate

$$\left(N = \log\left(1-p\right) / \log\left(1-\left(1-e\right)^{s}\right)\right)$$

Other robust algorithms

- RANSAC maximizes number of inliers
- LMedS minimizes median error

- Not recommended: case deletion, iterative least-squares, etc.

Automatic Computation of H

Objective

Compute homography between two images

Algorithm

(i) **Interest points:** Compute interest points in each image

(ii) **Putative correspondences:** Compute a set of interest point matches based on some similarity measure

(iii) **RANSAC robust estimation:** Repeat for N samples

(a) Select 4 correspondences and compute H

(b) Calculate the distance $d_\perp$ for each putative match

(c) Compute the number of inliers consistent with H ($d_\perp < t$)

Choose H with most inliers

(iv) **Optimal estimation:** re-estimate H from all inliers by minimizing ML cost function with Levenberg-Marquardt

(v) Guided matching: Determine more matches using prediction by computed H

Optionally iterate last two steps until convergence

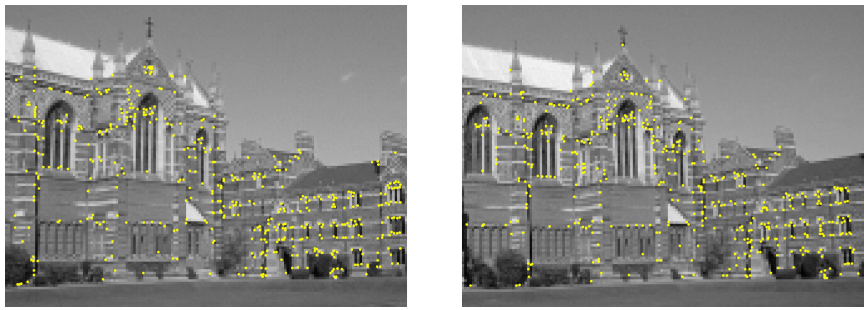
Determine Putative Correspondences

- Compare interest points
  Similarity measure:
  – SAD, SSD, ZNCC in small neighborhood

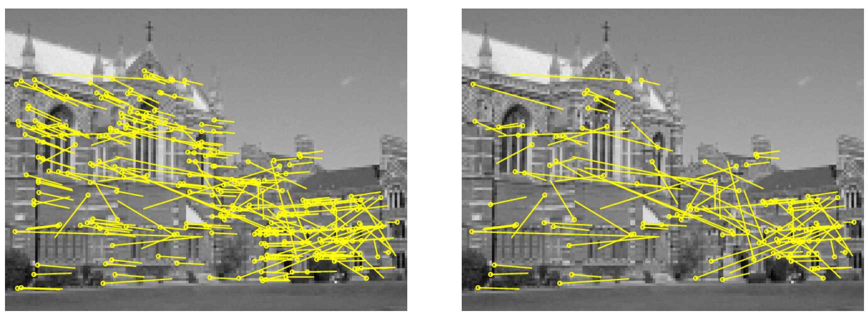
- If motion is limited, only consider interest points with similar coordinates

Example: robust computation

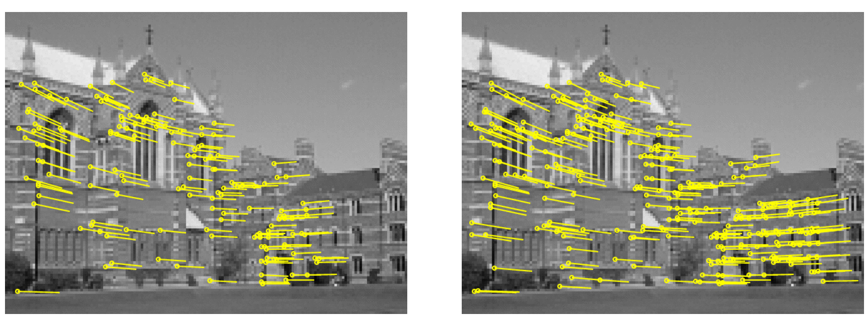
#in	1-e	adapt. *N*
6	2%	20M
10	3%	2.5M
44	16%	6,922
58	21%	2,291
73	26%	911
151	56%	43



Interest points
(500/image)
(640x480)



Putative correspondences (268)
(Best match,SSD<20)
Outliers (117)
(t=1.25 pixel; 43 iterations)



Inliers (151)

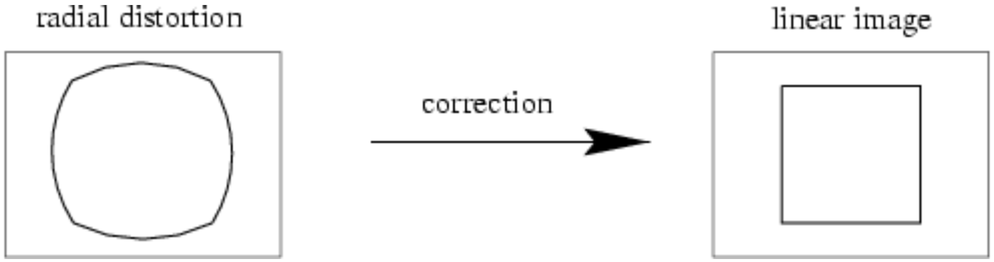
Final inliers (262)

Radial Distortion and Undistortion

Slides by R. Hartley, A. Zisserman and M. Pollefeys

Radial Distortion

short and long focal length



Typical Undistortion Model

Correction of distortion

$$\hat{x} = x_c + L(r)(x - x_c) \qquad \hat{y} = y_c + L(r)(y - y_c)$$

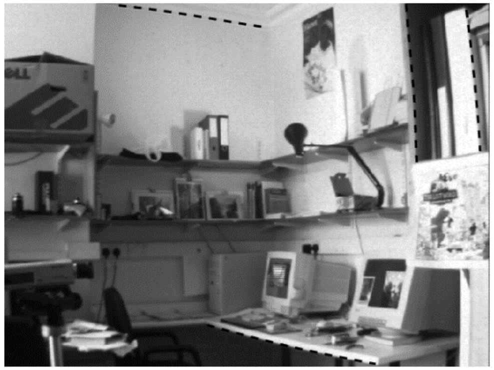
Choice of the distortion function and center

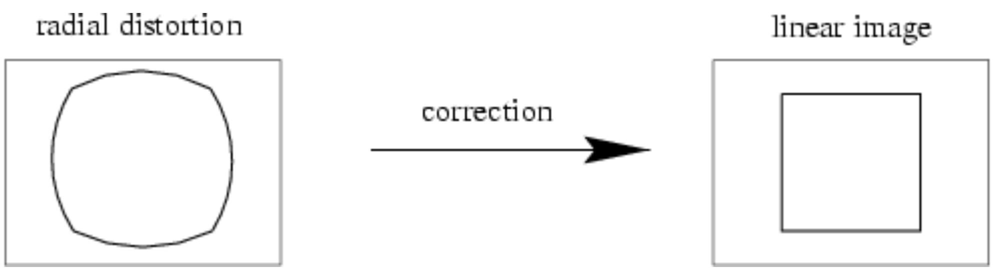
$$x = x_o + (x_o - c_x)(K_1 r^2 + K_2 r^4 + \ldots)$$
$$y = y_o + (y_o - c_y)(K_1 r^2 + K_2 r^4 + \ldots)$$

$$r = (x_o - c_x)^2 + (y_o - c_y)^2 \ .$$

Computing the parameters of the distortion function
(i) Minimize with additional unknowns
(ii) Straighten lines
(iii) …

Why Undistort?





$$(\tilde{x}, \tilde{y}, 1)^\top = [\mathtt{I} \mid \mathbf{0}]\mathbf{X}_{\text{cam}}$$

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = L(\tilde{r}) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}$$

Two-View Geometry

Slides by R. Hartley, A. Zisserman and M. Pollefeys

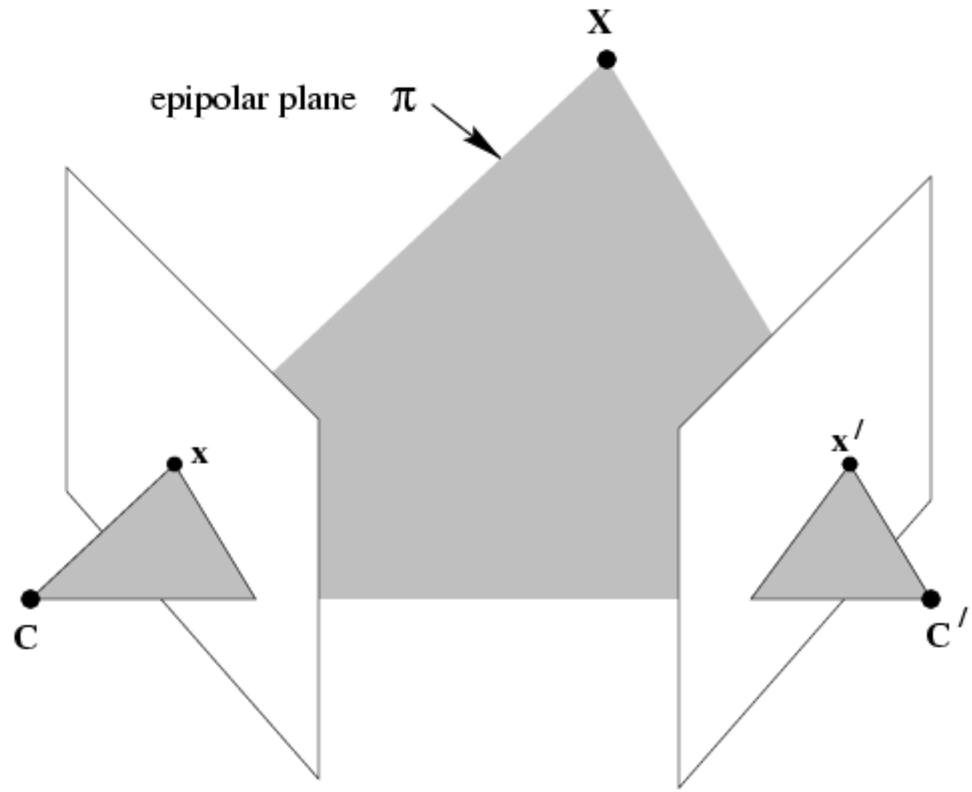
Three questions:

(i) **Correspondence geometry:** Given an image point x in the first image, 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?

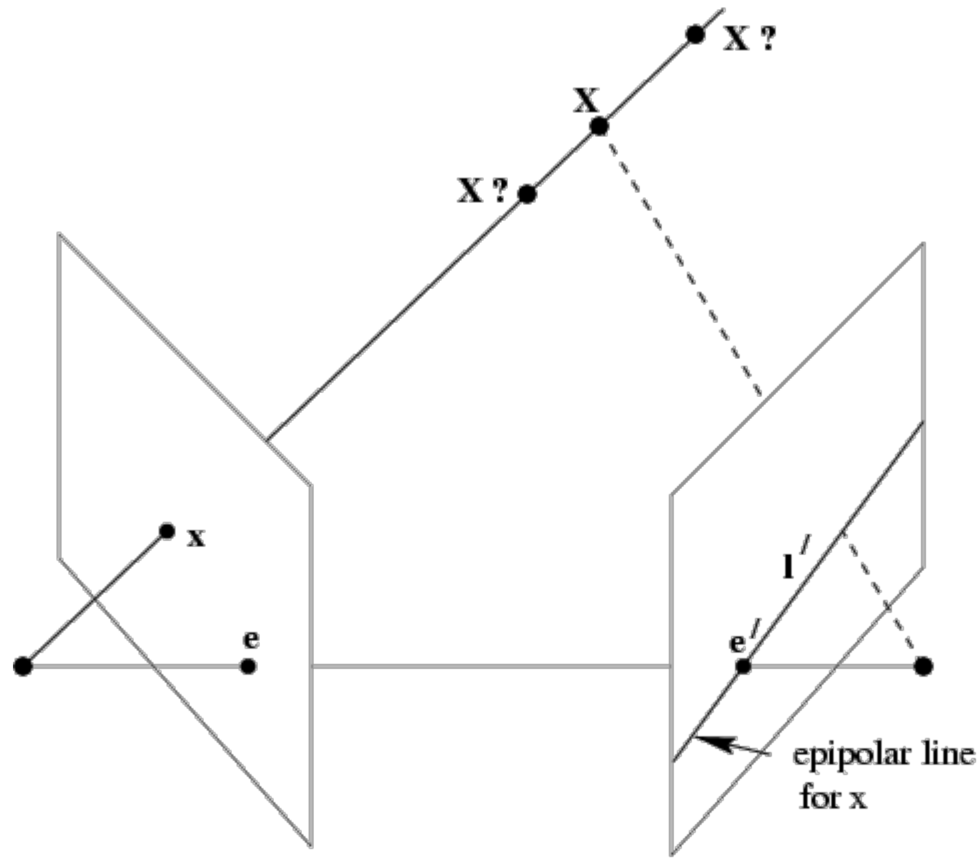
(iii) **Scene geometry (structure):** Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras P, P', what is the position of (their pre-image) X in space?

The Epipolar Geometry



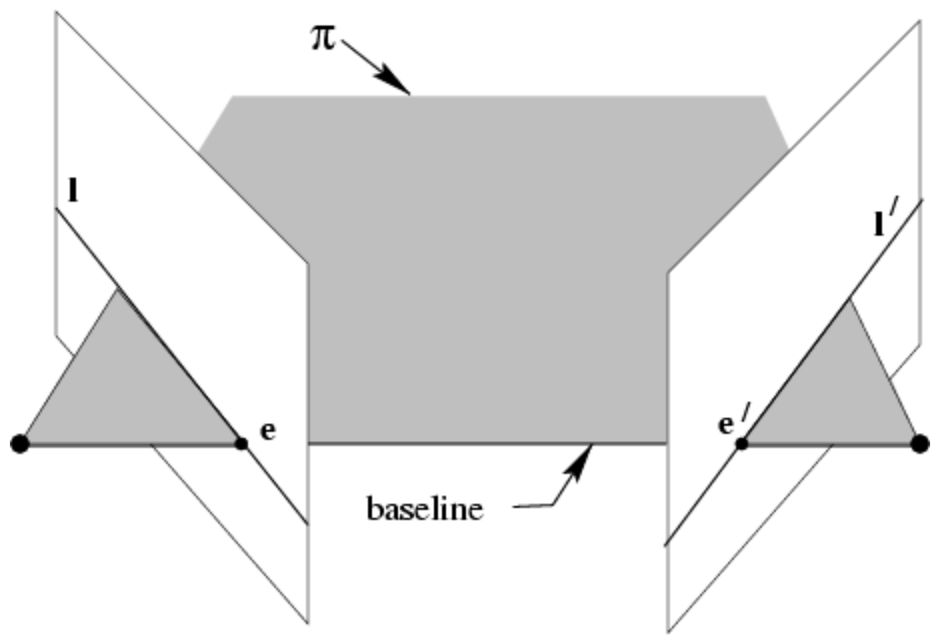
C, C', x, x' and X are coplanar

The Epipolar Geometry



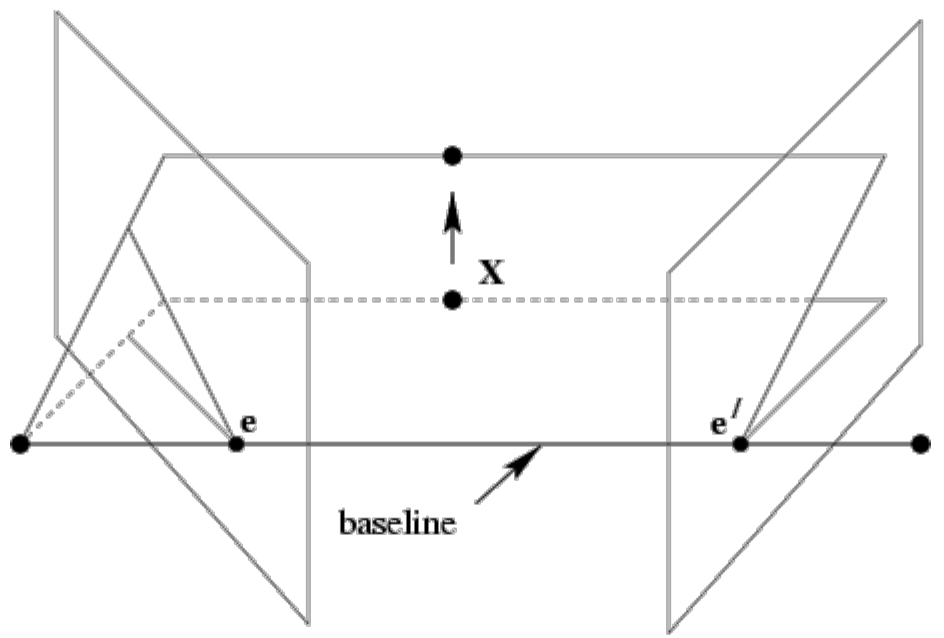
What if only C,C',x are known?

The Epipolar Geometry



All points on π project on l and l’

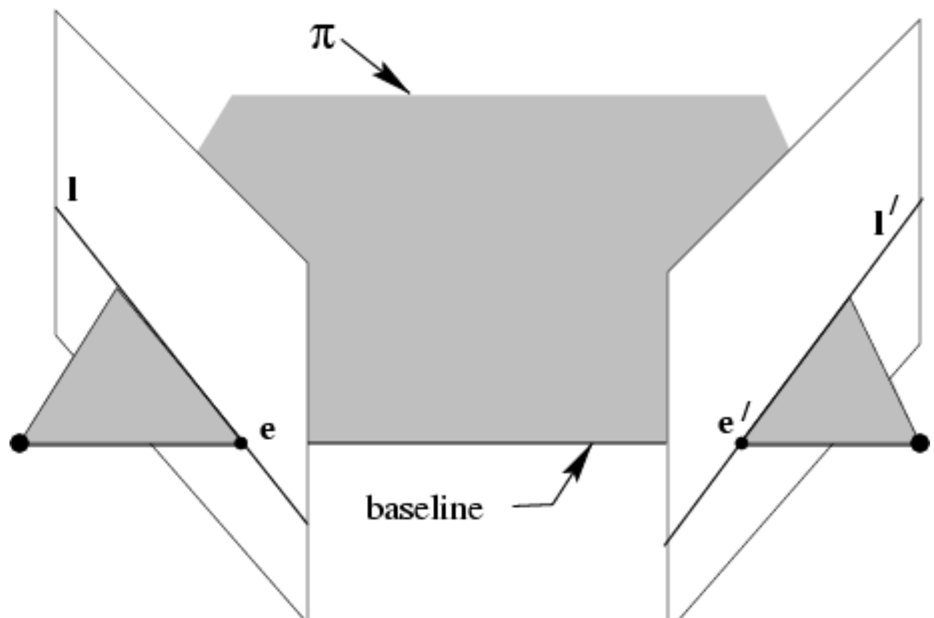
The Epipolar Geometry



Family of planes π and lines l and l'
Intersection in e and e'

The Epipolar Geometry

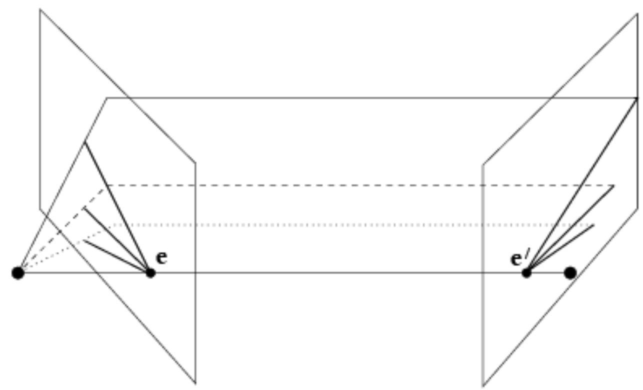
epipoles e, e’
= intersection of baseline with image plane
= projection of projection center in other image
= vanishing point of camera motion direction

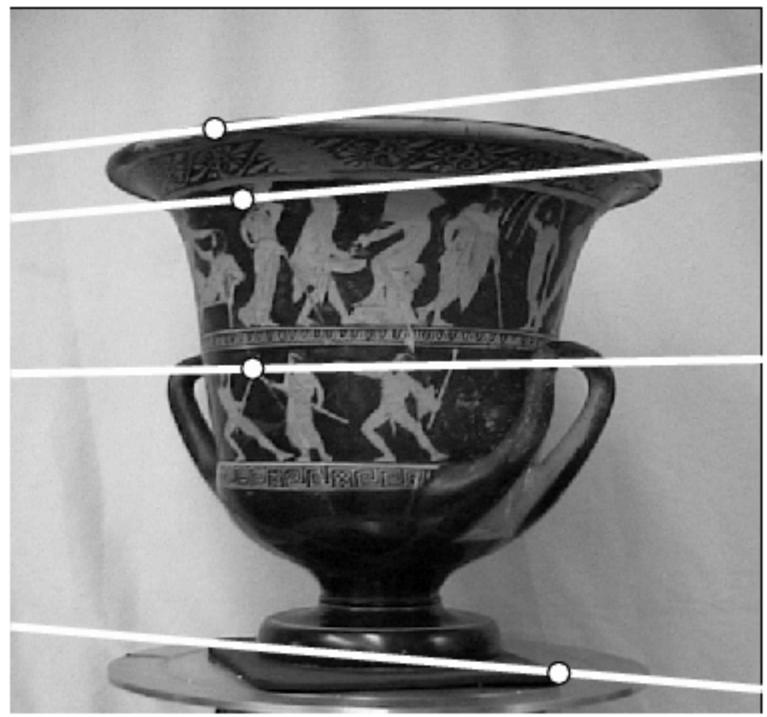


an epipolar plane = plane containing baseline (1-D family)

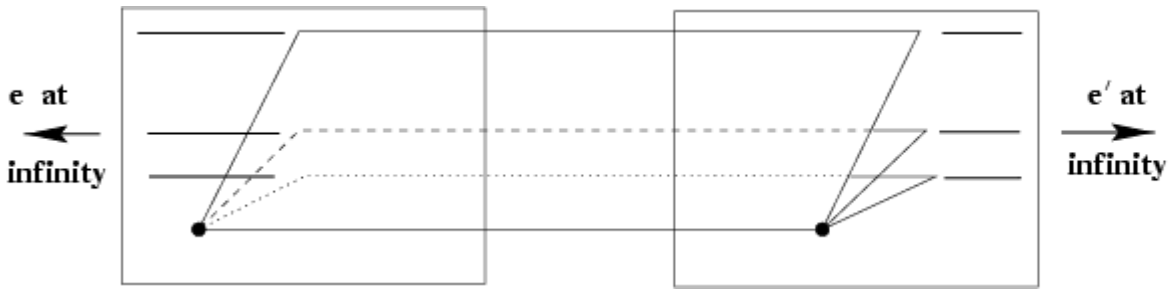
an epipolar line = intersection of epipolar plane with image
(always come in corresponding pairs)

Example: Converging Cameras



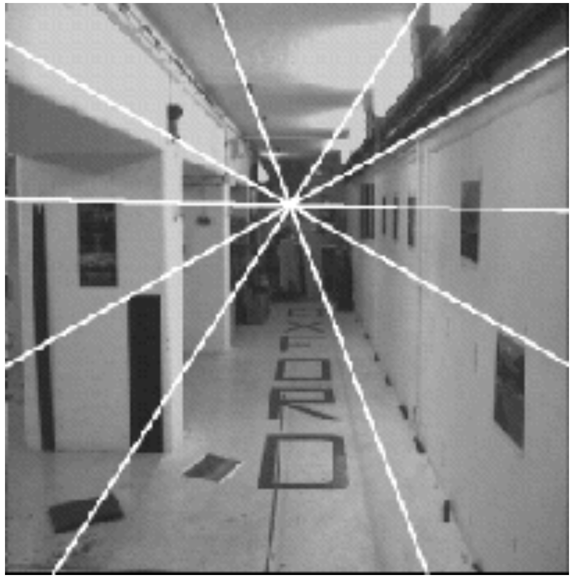


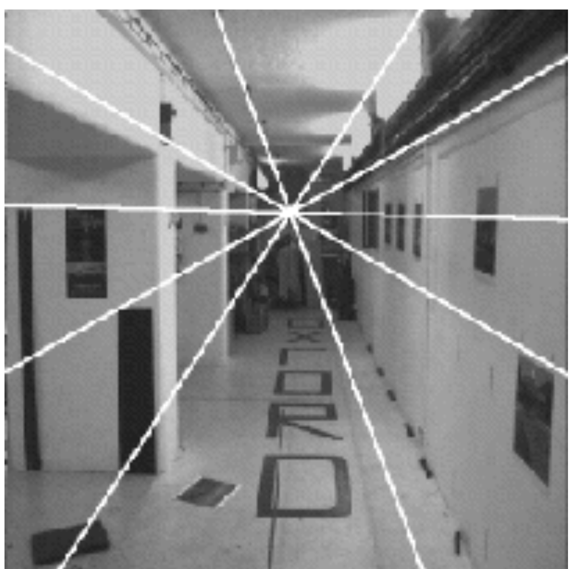
Example: Motion Parallel to Image Plane

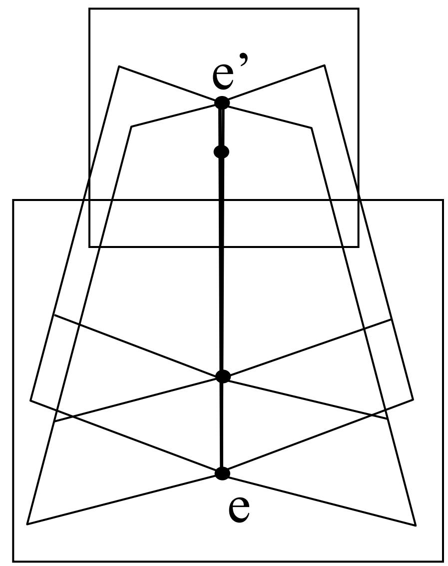


(simple for stereo → rectification)

Example: Forward Motion







The Fundamental Matrix F

algebraic representation of epipolar geometry

$$x \mapsto l'$$

we will see that mapping is a (singular) correlation
(i.e. projective mapping from points to lines)
represented by the fundamental matrix F

The Fundamental Matrix F

correspondence condition

The fundamental matrix satisfies the condition that for any pair of corresponding points x↔x’ in the two images

$$x'^T F x = 0 \qquad \left(x'^T l' = 0\right)$$

The Fundamental Matrix F

F is the unique 3x3 rank 2 matrix that satisfies $x'^T F x=0$ for all $x \leftrightarrow x'$

- **(i) Transpose:** if F is fundamental matrix for (P,P'), then F^T is fundamental matrix for (P',P)
- **(ii) Epipolar lines:** $l'=Fx$ & $l=F^T x'$
- **(iii) Epipoles:** on all epipolar lines, thus $e'^T F x=0, \forall x \Rightarrow e'^T F=0$, similarly $Fe=0$
- **(iv) F** has 7 d.o.f. , i.e. 3x3-1(homogeneous)-1(rank2)
- **(v) F** is a correlation, projective mapping from a point x to a line $l'=Fx$ (not a proper correlation, i.e. not invertible)

Two View Geometry Computation: Linear Algorithm

For every match (m,m´): $\mathrm{x}'^{\mathrm{T}}\mathrm{F}\mathrm{x} = 0$

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0$$

separate known from unknown

$$[x'x, x'y, x', y'x, y'y, y', x, y, 1][f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^{\mathrm{T}} = 0$$

(data) (unknowns)

(linear)

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathrm{f} = 0$$

$$\mathrm{Af} = 0$$

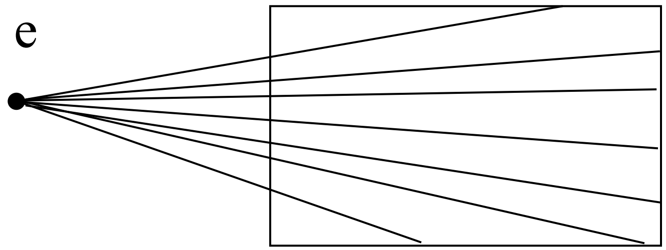
Benefits from having F

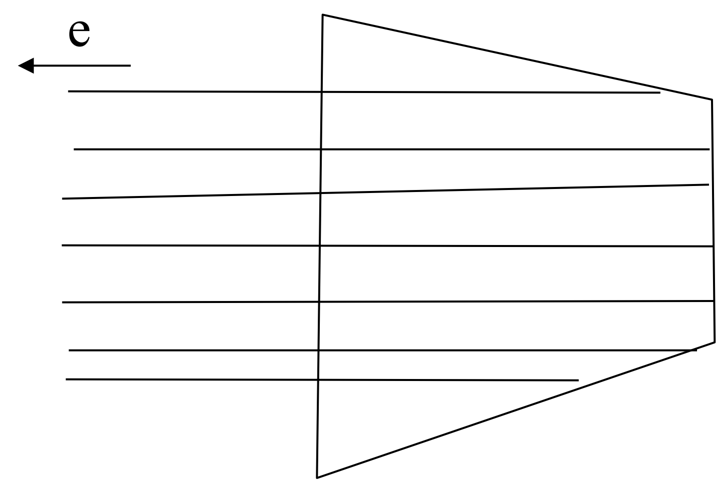
- Given a pixel in one image, the corresponding pixel has to lie on epipolar line
- Search space reduced from 2-D to 1-D

Image Pair Rectification

simplify stereo matching
by warping the images

Apply projective transformation so that epipolar lines
correspond to horizontal scanlines





$$\begin{bmatrix}1\\0\\0\end{bmatrix} = He$$

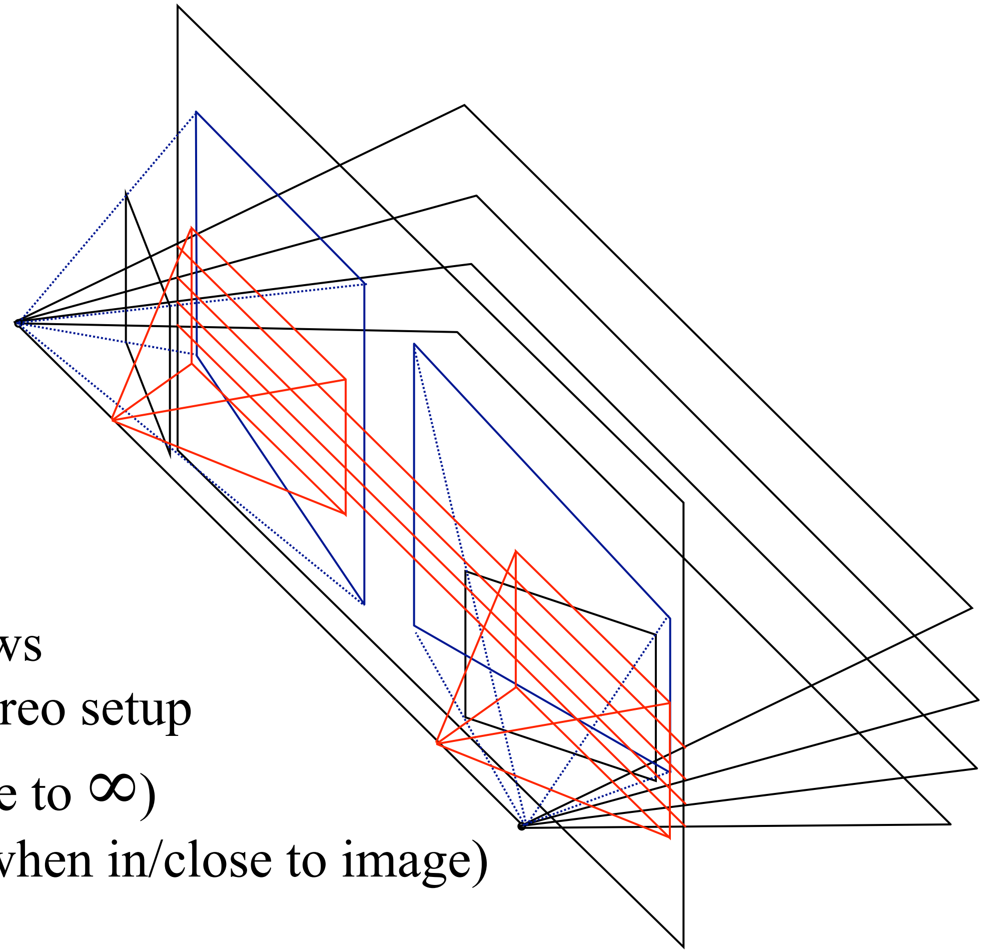
map epipole e to (1,0,0)

try to minimize image distortion

problem when epipole in (or close to) the image

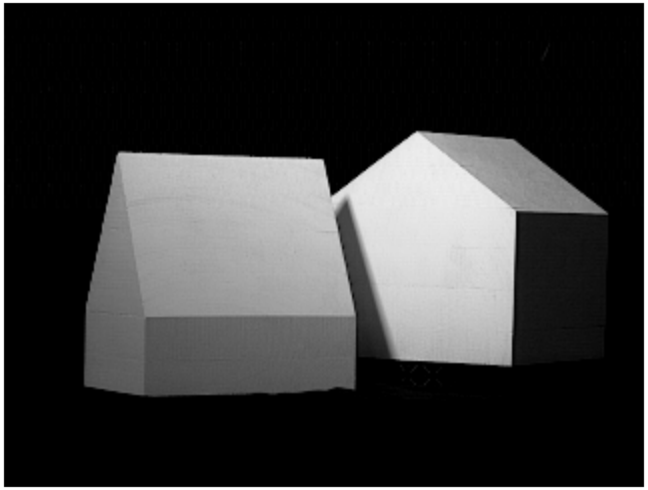
Planar Rectification

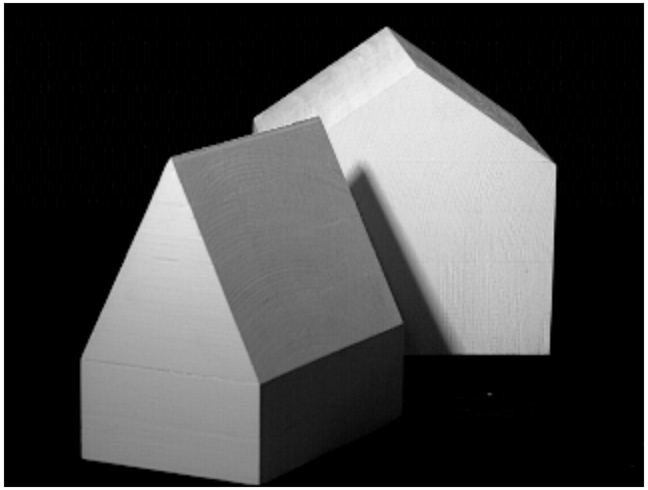
(standard approach)

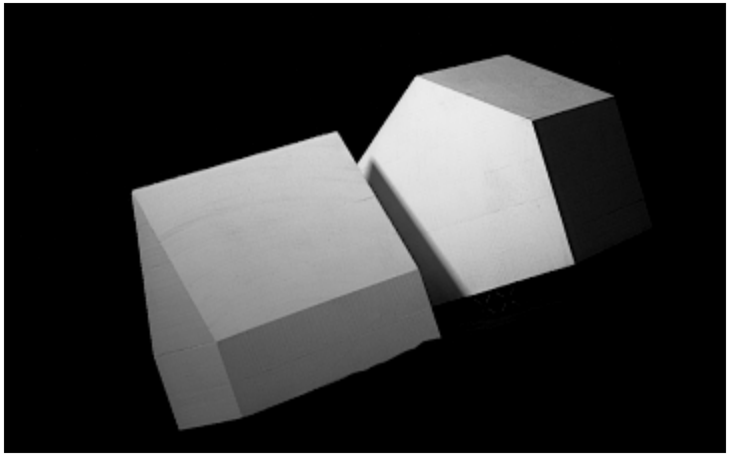


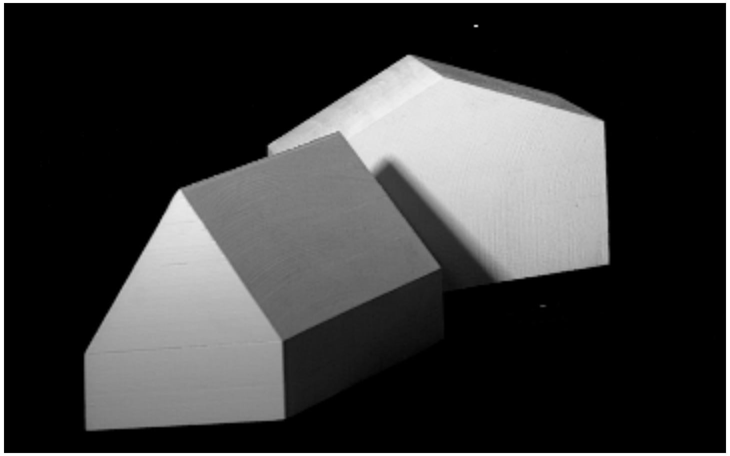
Bring two views
to standard stereo setup

(moves epipole to ∞)
(not possible when in/close to image)









The Essential Matrix

~fundamental matrix for calibrated cameras (remove K)

$$E = [t]_\times R = R[R^T t]_\times \qquad [\mathbf{a}]_\times \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

$$\hat{x}'^T E \hat{x} = 0 \qquad \left(\hat{x} = K^{-1}x;\ \hat{x}' = K^{-1}x'\right)$$

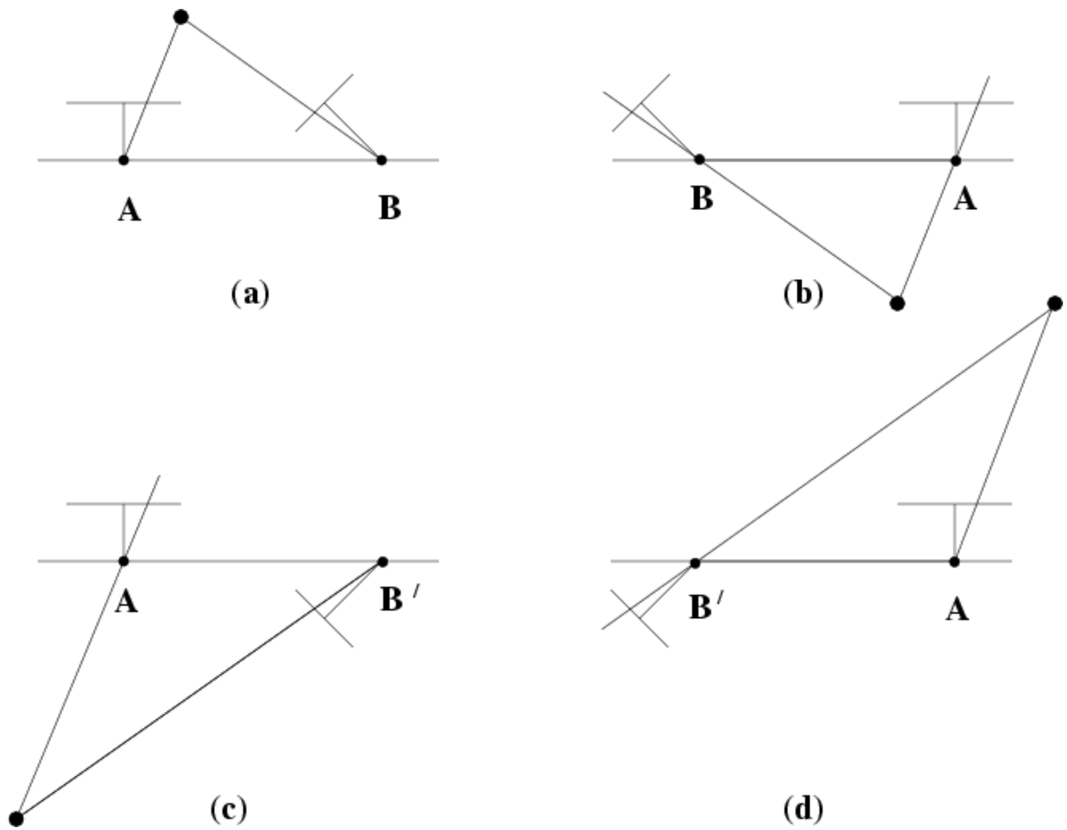
$$E = K'^T F K$$

5 d.o.f. (3 for R; 2 for t up to scale)

E is an essential matrix if and only if two singular values are equal (and the third=0)

$$E = U\mathrm{diag}(1,1,0)V^T$$

Four Possible Solutions from E



Given E and setting the first camera matrix P = [I | 0], there are four possible solutions for P’ (only one solution, however, where a reconstructed point is in front of both cameras)

Fundamental Matrix Estimation

Epipolar Geometry: Basic Equation

$$\mathrm{x'}^{\mathrm{T}}\,\mathrm{Fx} = 0$$

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0$$

separate known from unknown

$$[x'x, x'y, x', y'x, y'y, y', x, y, 1][f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^{\mathrm{T}} = 0$$

(data) (unknowns)

(linear)

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathrm{f} = 0$$

$$\mathrm{Af} = 0$$

The Singularity Constraint

$$e'^{T} F = 0 \qquad Fe = 0 \qquad \det F = 0 \qquad \text{rank } F = 2$$

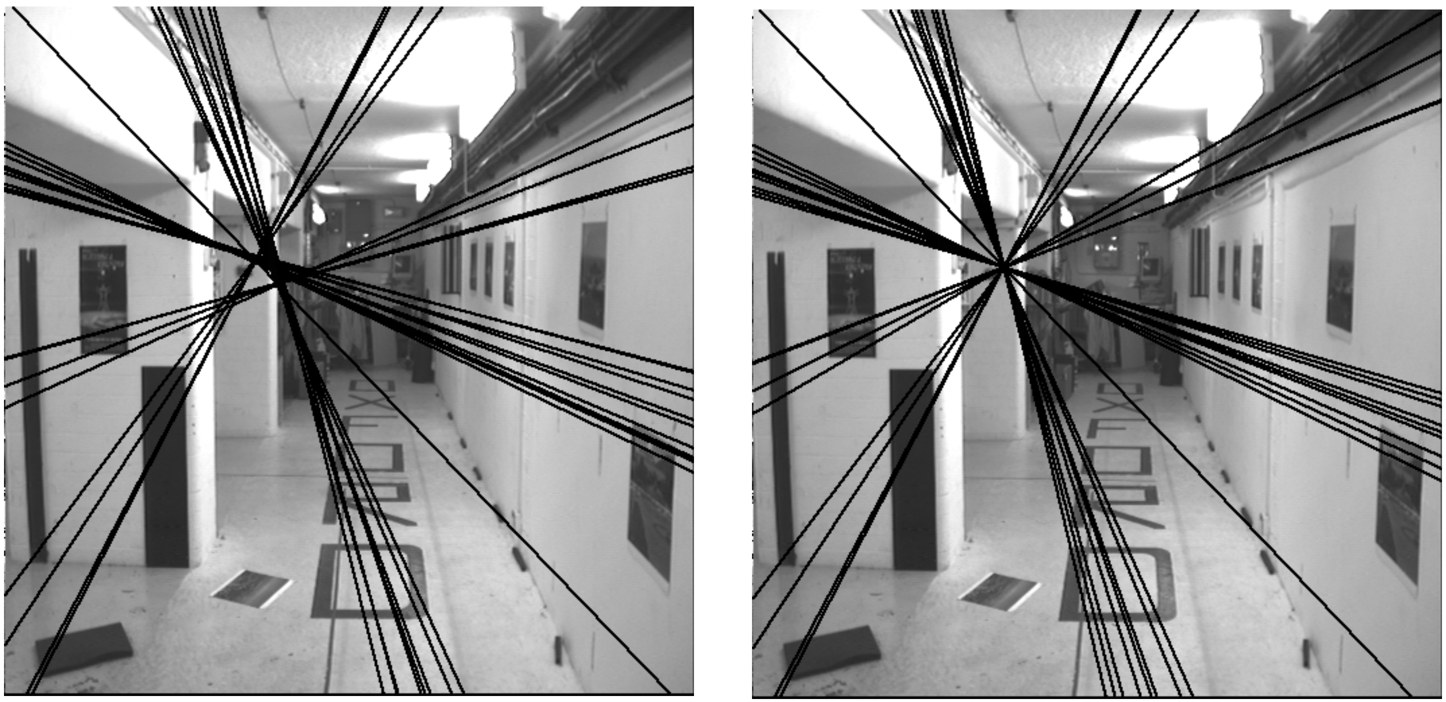
SVD from linearly computed F matrix (rank 3)

$$F = U \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} V^T = U_1 \sigma_1 V_1^T + U_2 \sigma_2 V_2^T + U_3 \sigma_3 V_3^T$$

Compute closest rank-2 approximation $\min \|F - F'\|_F$

$$F' = U \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix} V^T = U_1 \sigma_1 V_1^T + U_2 \sigma_2 V_2^T$$

The Singularity Constraint



The NOT Normalized 8-point Algorithm

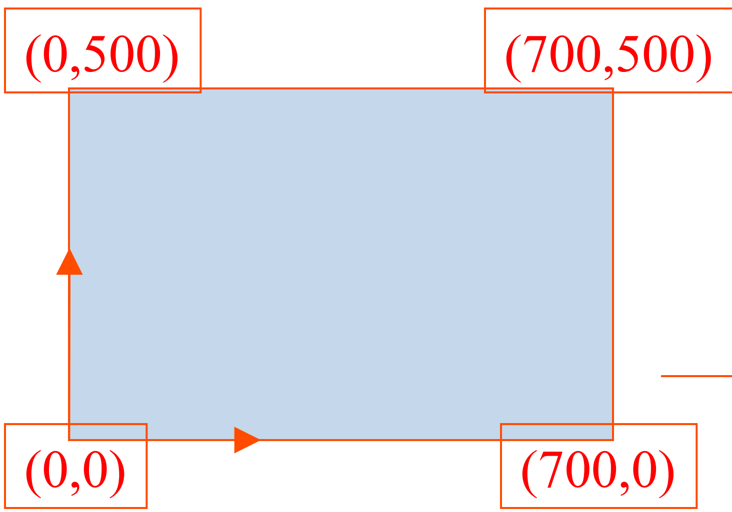
$$\begin{array}{c}\begin{bmatrix} x_1x_1' & y_1x_1' & x_1' & x_1y_1' & y_1y_1' & y_1' & x_1 & y_1 & 1 \\ x_2x_2' & y_2x_2' & x_2' & x_2y_2' & y_2y_2' & y_2' & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_nx_n' & y_nx_n' & x_n' & x_ny_n' & y_ny_n' & y_n' & x_n & y_n & 1 \end{bmatrix} \\ \begin{array}{ccccccccc} \sim 10000 & \sim 10000 & \sim 100 & \sim 10000 & \sim 10000 & \sim 100 & \sim 100 & \sim 100 & 1 \end{array}\end{array} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

!

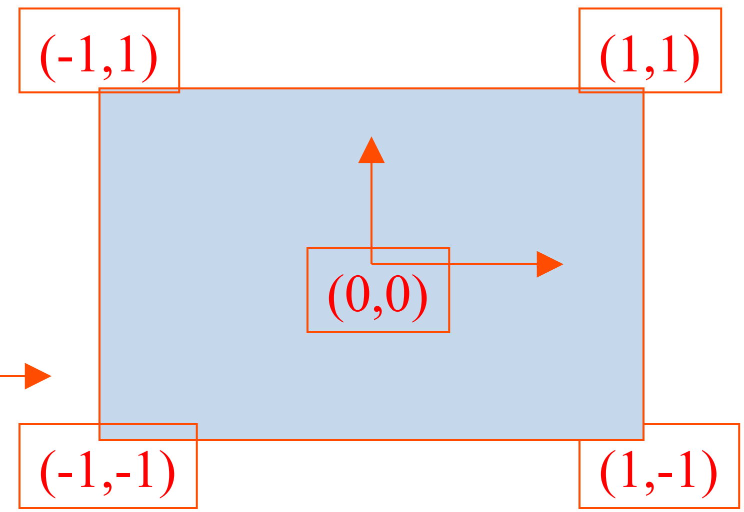
Orders of magnitude difference
between column of data matrix
$\rightarrow$ least-squares yields poor results

The Normalized 8-point Algorithm

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

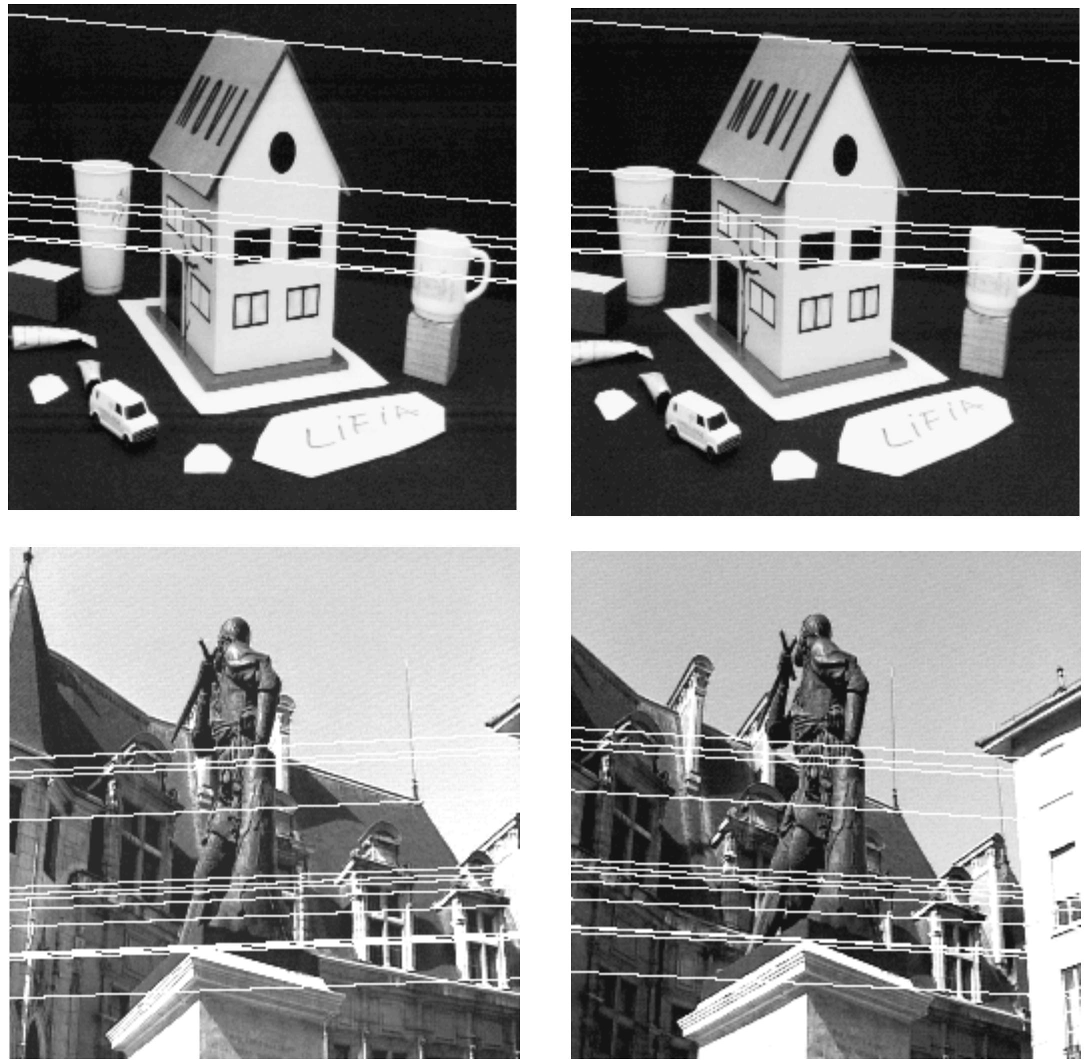


$$\begin{bmatrix} \frac{2}{700} & 0 & -1 \\ & \frac{2}{500} & -1 \\ & & 1 \end{bmatrix}$$



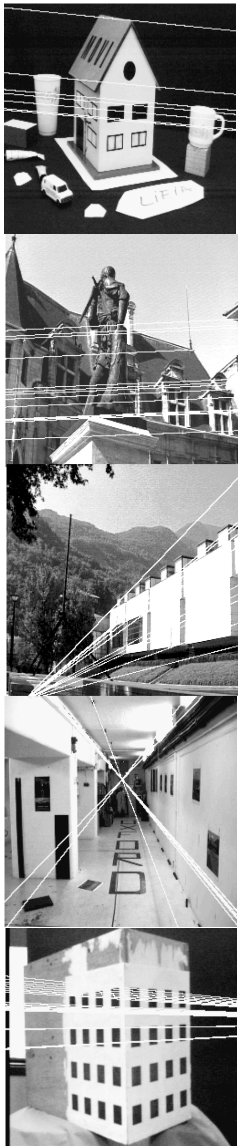
normalized least squares yields good results

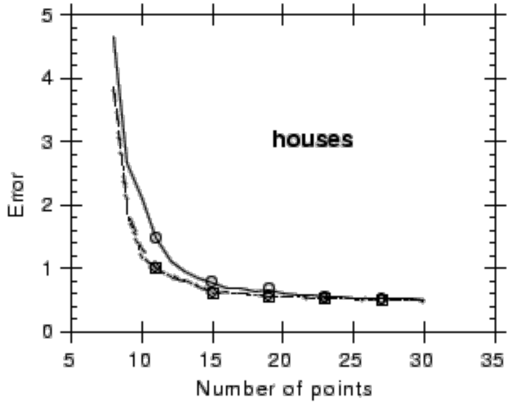
Some Experiments

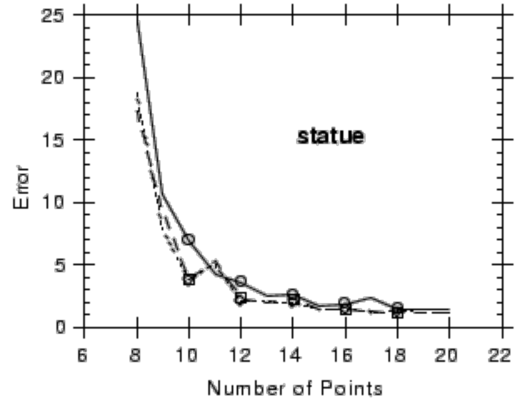


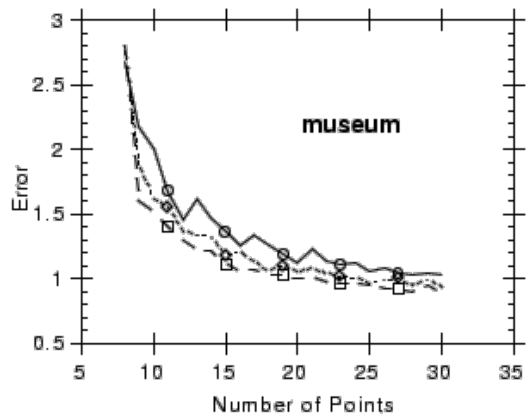
Some Experiments

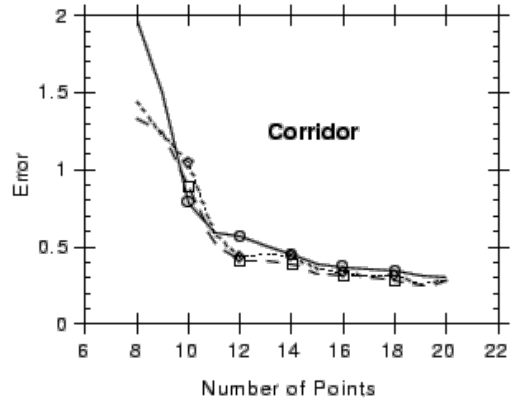
Some Experiments

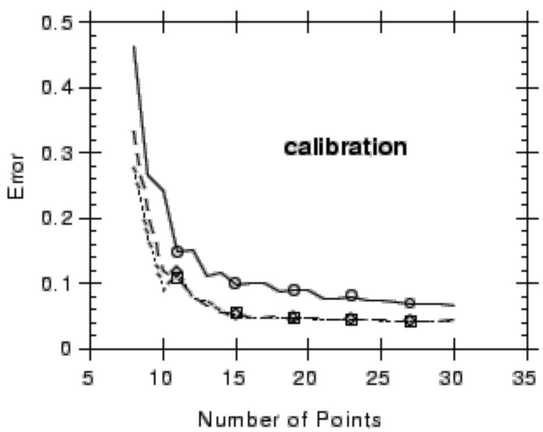












Residual error:

$$\sum_i d(\mathrm{x}'_i, F\mathrm{x}_i)^2 + d(\mathrm{x}_i, \mathrm{F}^{\mathrm{T}}\mathrm{x}'_i)^2$$

(for all points!)

Recommendations:

1. Do not use unnormalized algorithms

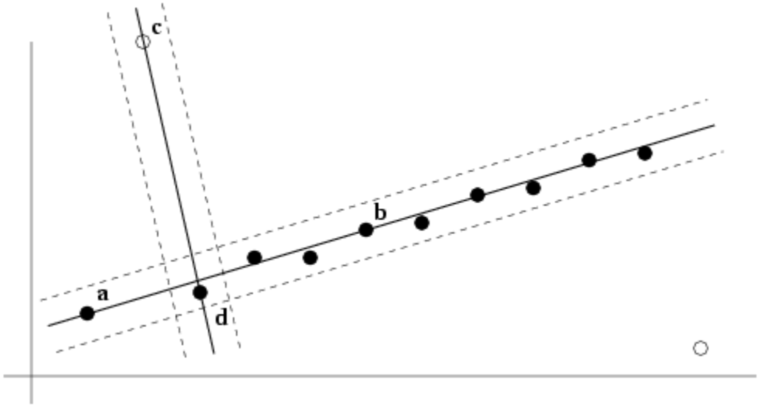
2. Quick and easy to implement: 8-point normalized

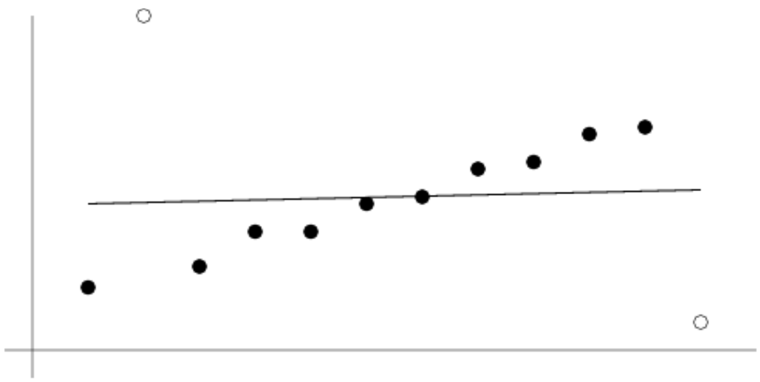
3. Better: enforce rank-2 constraint during minimization

4. Best: Maximum Likelihood Estimation
   (minimal parameterization, sparse implementation)

Robust Estimation

- What if set of matches contains gross outliers?





RANSAC

<u>Objective</u>

Robust fit of model to data set S which contains outliers

<u>Algorithm</u>

(i) Randomly select a sample of *s* data points from S and instantiate the model from this subset.

(ii) Determine the set of data points S_i which are within a distance threshold *t* of the model. The set S_i is the consensus set of samples and defines the inliers of S.

(iii) If the subset of S_i is greater than some threshold *T*, re-estimate the model using all the points in S_i and terminate

(iv) If the size of S_i is less than *T*, select a new subset and repeat the above.

(v) After *N* trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i

How many samples?

- Choose t so probability for inlier is α (e.g. 0.9)
  - Or empirically
- Choose N so that, with probability *p*, at least one random sample is free from outliers. e.g. *p* =0.99

$$\left(1-\left(1-e\right)^{s}\right)^{N}=1-p$$

$$N=\log\left(1-p\right)/\log\left(1-\left(1-e\right)^{s}\right)$$

	proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Acceptable consensus set?

- Typically, terminate when inlier ratio reaches expected ratio of inliers

$$T = (1 - e)n$$

Adaptively determining the number of samples

e is often unknown a priori, so pick worst case, e.g. 50%, and adapt if more inliers are found, e.g. 80% would yield e=0.2

- $N=\infty$, *sample_count* =0
- While N >*sample_count* repeat
  - Choose a sample and count the number of inliers
  - Set e=1-(number of inliers)/(total number of points)
  - Recompute N from e
  - Increment the *sample_count* by 1
- Terminate

$$N = \log(1 - p) / \log\left(1 - (1 - e)^s\right)$$

RANSAC for F Estimation

Step 1. Extract features

Step 2. Compute a set of potential matches

Step 3. do

Step 3.1 select minimal sample (i.e. 7 matches)

Step 3.2 compute solution(s) for F

(generate hypothesis)

Step 3.3 determine inliers (verify hypothesis)

until p(#*inliers*,#*samples*)>95% or 99%

Step 4. Compute F based on all inliers

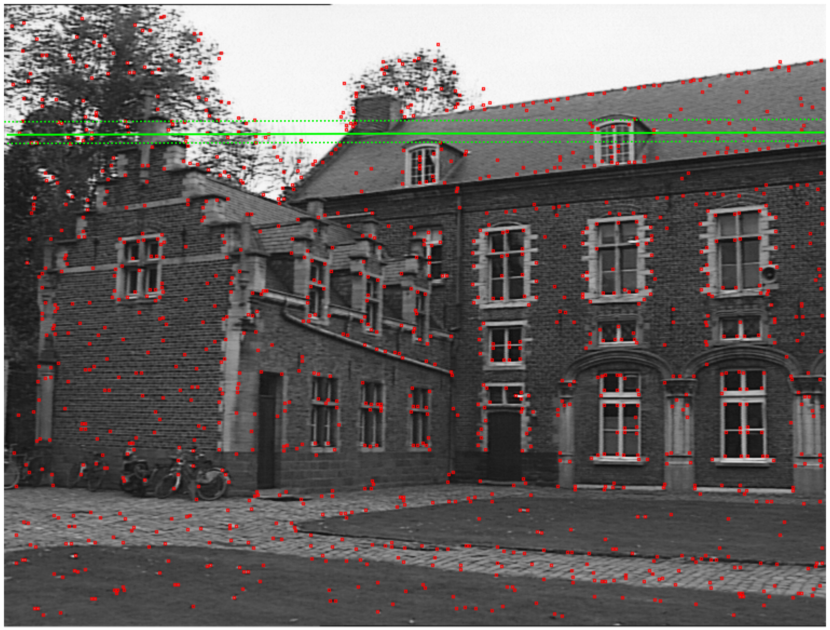
Step 5. Look for additional matches

Step 6. Refine F based on all correct matches

$$p = 1 - \left(1 - \left(\frac{\#inliers}{\#matches}\right)^7\right)^{\#samples}$$

#inliers	90%	80%	70%	60%	50%
#samples	5	13	35	106	382

Finding more matches

restrict search range to neighborhood of epipolar line
(±1.5 pixels)
relax disparity restriction (along epipolar line)

Degenerate Cases

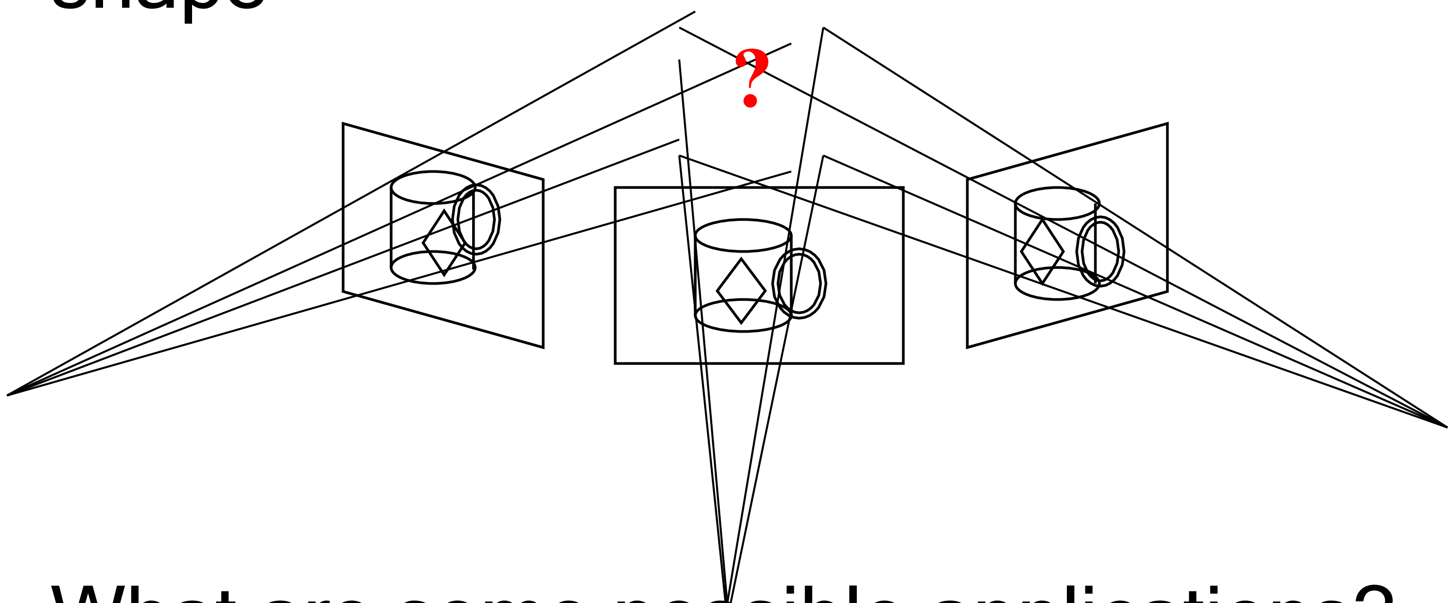
- Degenerate cases
  - Planar scene
  - Pure rotation
- No unique solution
  - Remaining DOF filled by noise
  - Use simpler model (e.g. homography)
- Model selection (Torr et al., ICCV´98, Kanatani, Akaike)
  - Compare H and F according to expected residual error (compensate for model complexity)

Stereo Matching

Slides by Rick Szeliski, Pascal Fua and P. Mordohai

Stereo Matching

- Given two or more images of the same scene or object, compute a representation of its shape



- What are some possible applications?

Stereo Matching

- Given two or more images of the same scene or object, compute a representation of its shape

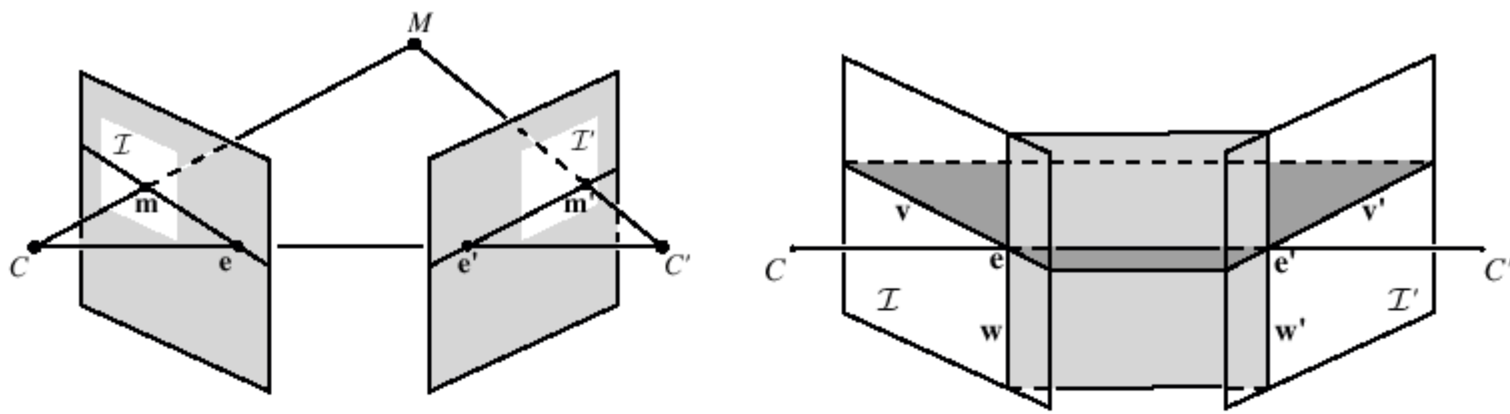
- What are some possible representations?
  - depth maps
  - volumetric models
  - 3D surface models
  - planar (or offset) layers

Stereo Matching

- What are some possible algorithms?
  - match “features” and interpolate
  - match edges and interpolate
  - match all pixels with windows (coarse-fine)
  - use optimization:
    - iterative updating
    - dynamic programming
    - energy minimization (regularization, stochastic)
    - graph algorithms

Rectification

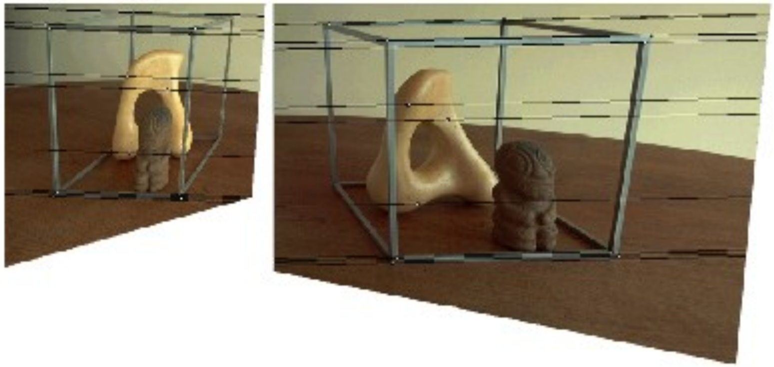
- Project each image onto same plane, which is parallel to the baseline
- Resample lines (and shear/stretch) to place lines in correspondence, and minimize distortion



- Take rectification for granted in this course

Rectification

(a) Original image pair overlayed with several epipolar lines.

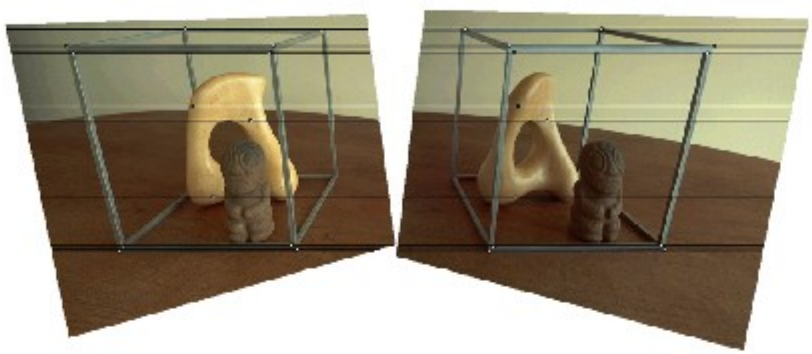


(b) Image pair transformed by the specialized projective mapping $\mathbf{H}_p$ and $\mathbf{H}'_p$. Note that the epipolar lines are now parallel to each other in each image.

BAD!

Rectification

(c) Image pair transformed by the similarity $\mathbf{H}_r$ and $\mathbf{H}'_r$. Note that the image pair is now rectified (the epipolar lines are horizontally aligned).



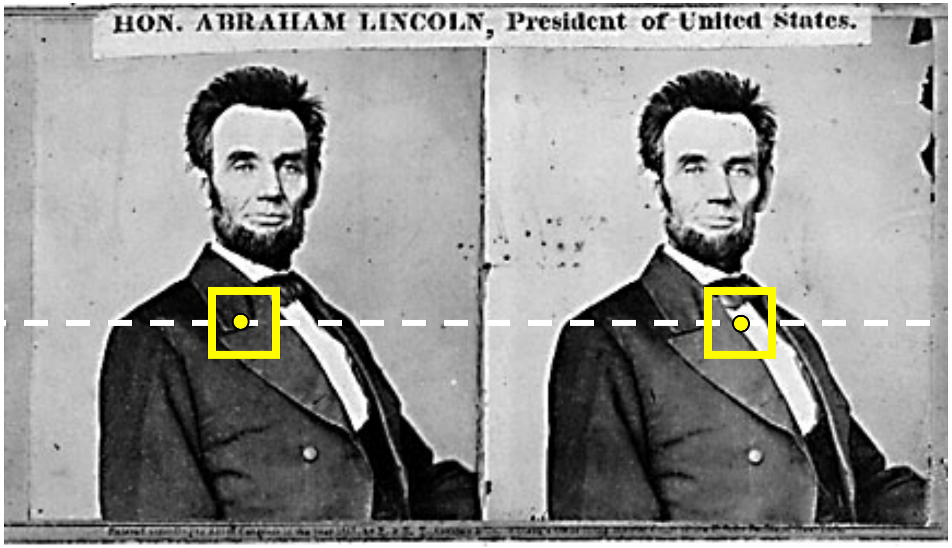
(d) Final image rectification after shearing transform $\mathbf{H}_s$ and $\mathbf{H}'_s$. Note that the image pair remains rectified, but the horizontal distortion is reduced.

GOOD!

Finding Correspondences

- Apply feature matching criterion at *all* pixels simultaneously
- Search only over epipolar lines (many fewer candidate positions)

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***

Disparity

- Disparity d is the difference between the x coordinates of corresponding pixels in the left and right image

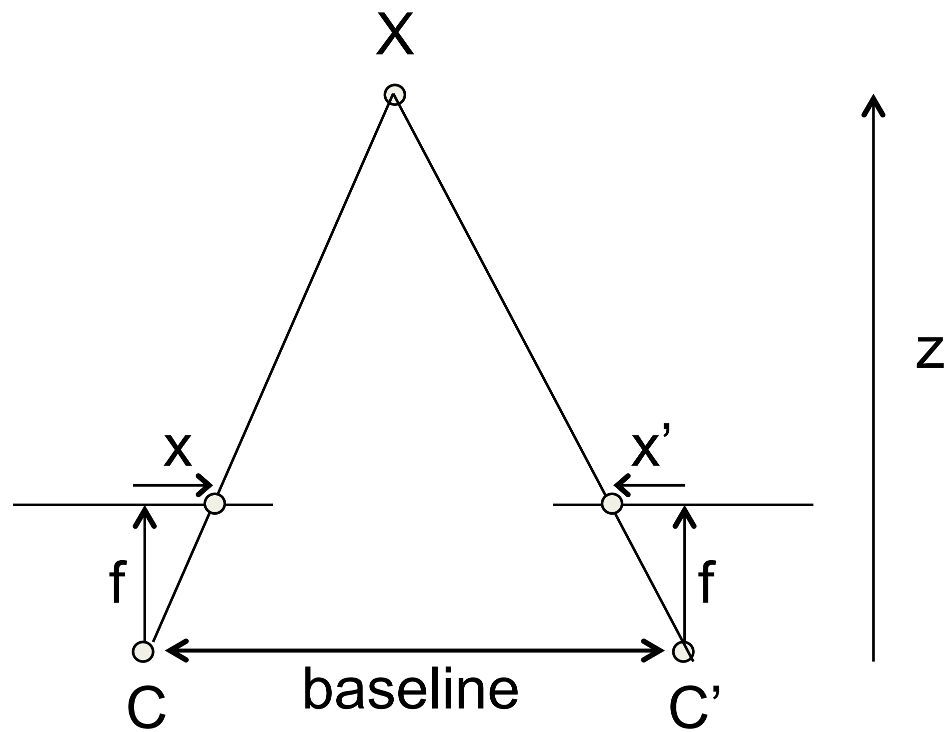
$$d=x_L-x_R$$

- Disparity is inversely proportional to depth

$$Z=bf/d$$

Stereo Reconstruction

$$Z = bf/d$$

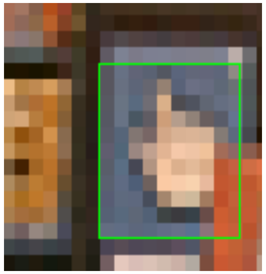


Finding Correspondences

- How do we determine correspondences?
  - *block matching* or *SSD* (sum squared differences)

$$SSD = \sum_{[i,j] \in R} (f(i,j) - g(i,j))^2$$

  - *d* is the *disparity* (horizontal motion)

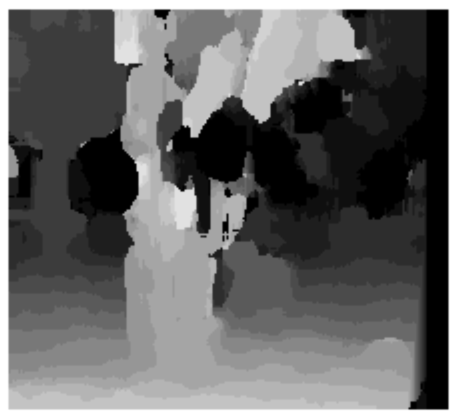


- How big should the neighborhood be?

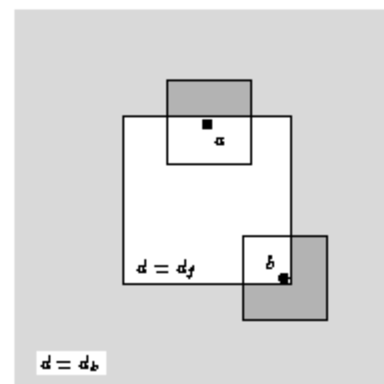
Neighborhood size

- Smaller neighborhood: more details
- Larger neighborhood: fewer isolated mistakes

w = 3



w = 20



Challenges

- Ill-posed inverse problem
  - Recover 3-D structure from 2-D information
- Difficulties
  - Uniform regions
  - Half-occluded pixels
  - Repeated patterns

Pixel Dissimilarity

- Sum of Squared Differences of intensities (SSD)

$$SSD = \sum_{[i,j]\in R} (f(i,j) - g(i,j))^2$$

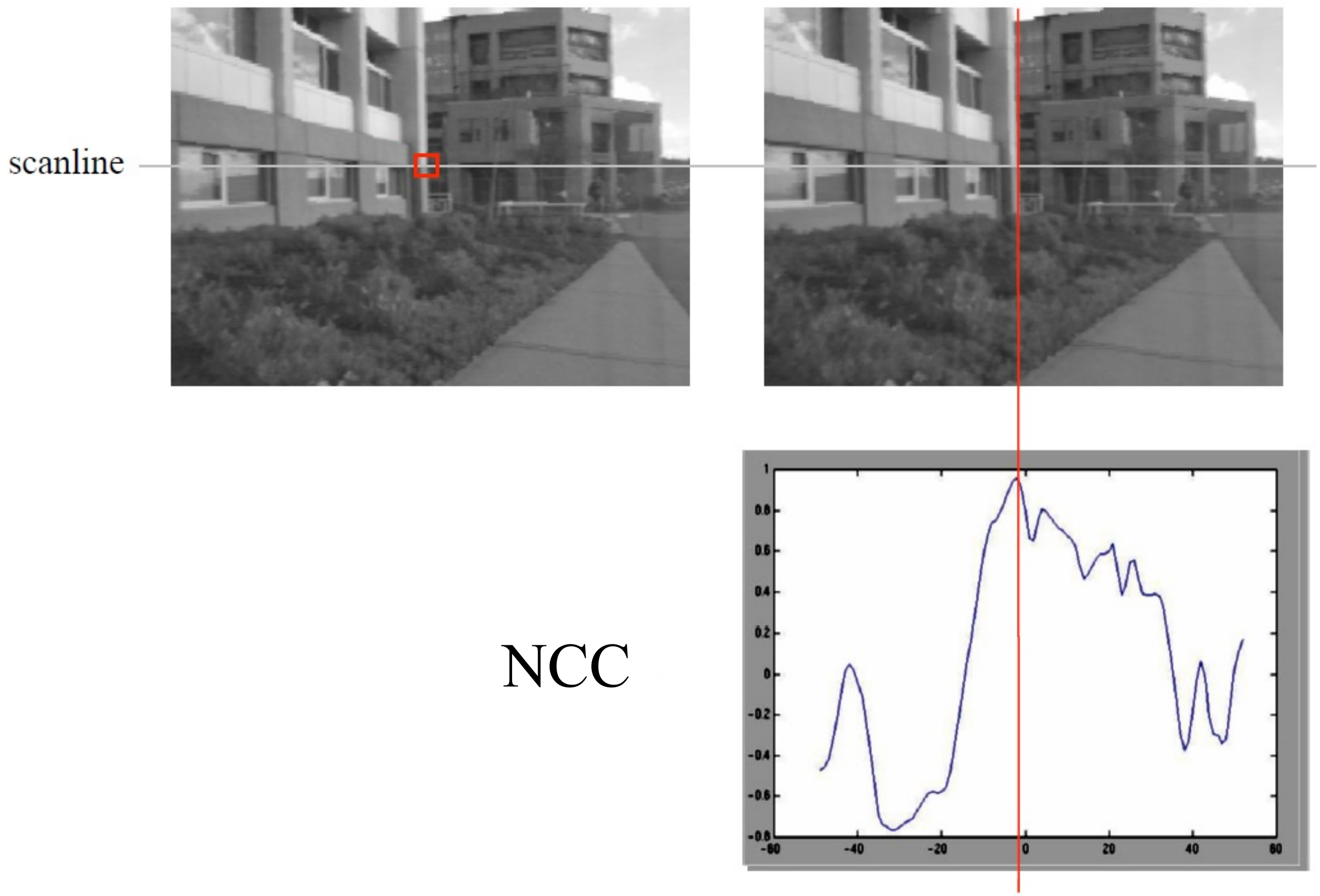
- Sum of Absolute Differences of intensities (SAD)

$$SAD = \sum_{[i,j]\in R} |f(i,j) - g(i,j)|$$

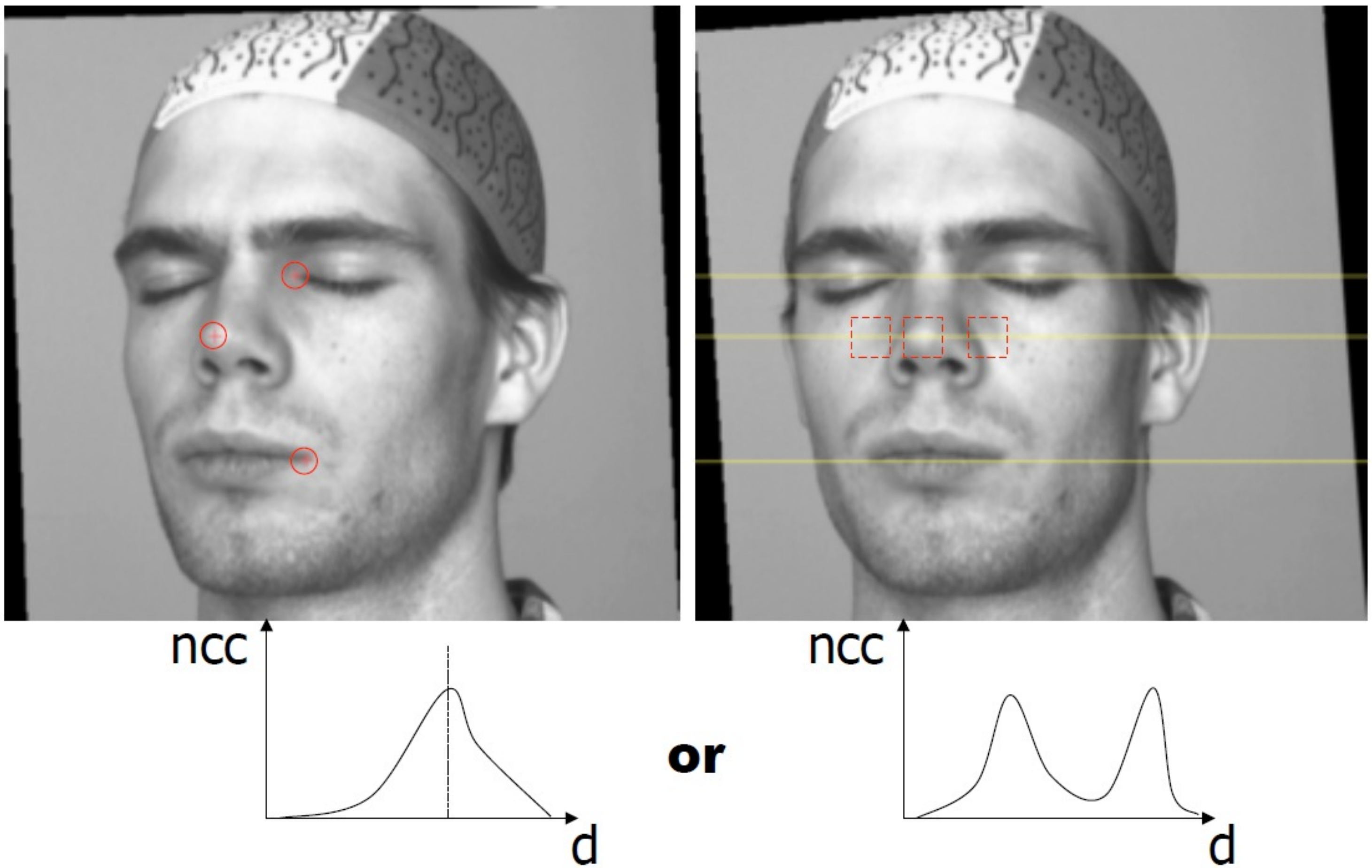
- Zero-mean Normalized Cross-correlation (NCC)

$$NCC(x,y,d) = \frac{\sum_{i\in W}(I_L(x_i,y_i) - \mu_L)(I_R(x_i - d, y_i) - \mu_R)}{\sigma_L \sigma_R}$$

Cost/Score Curve

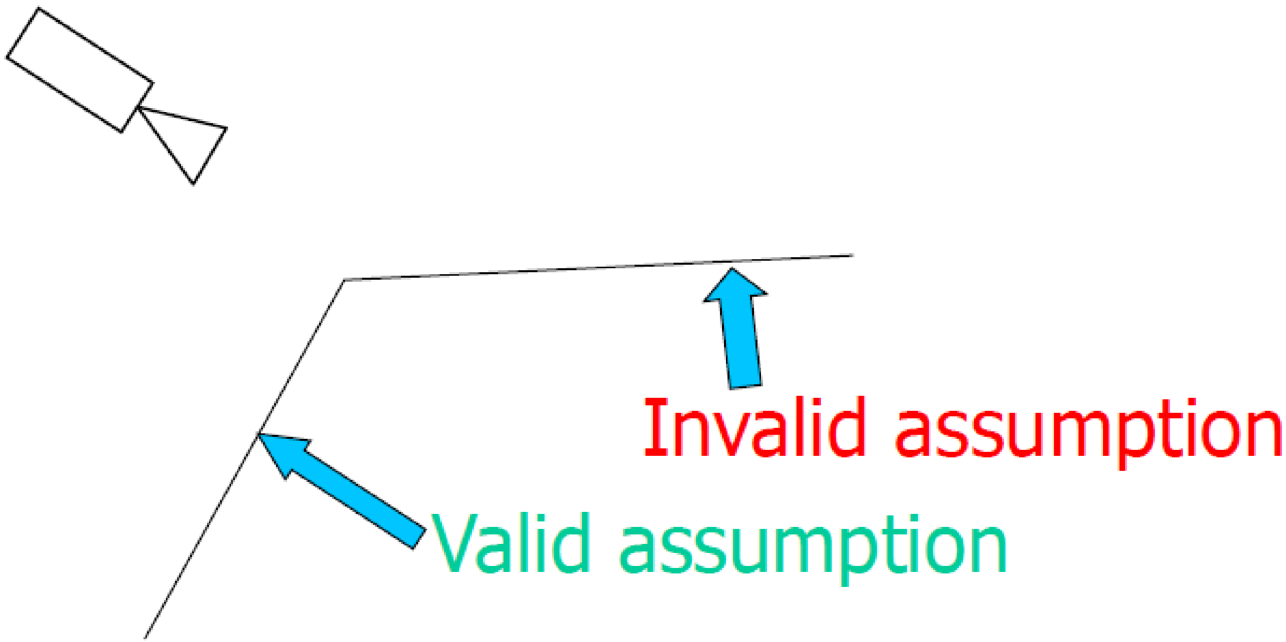


Cost/Score Curve



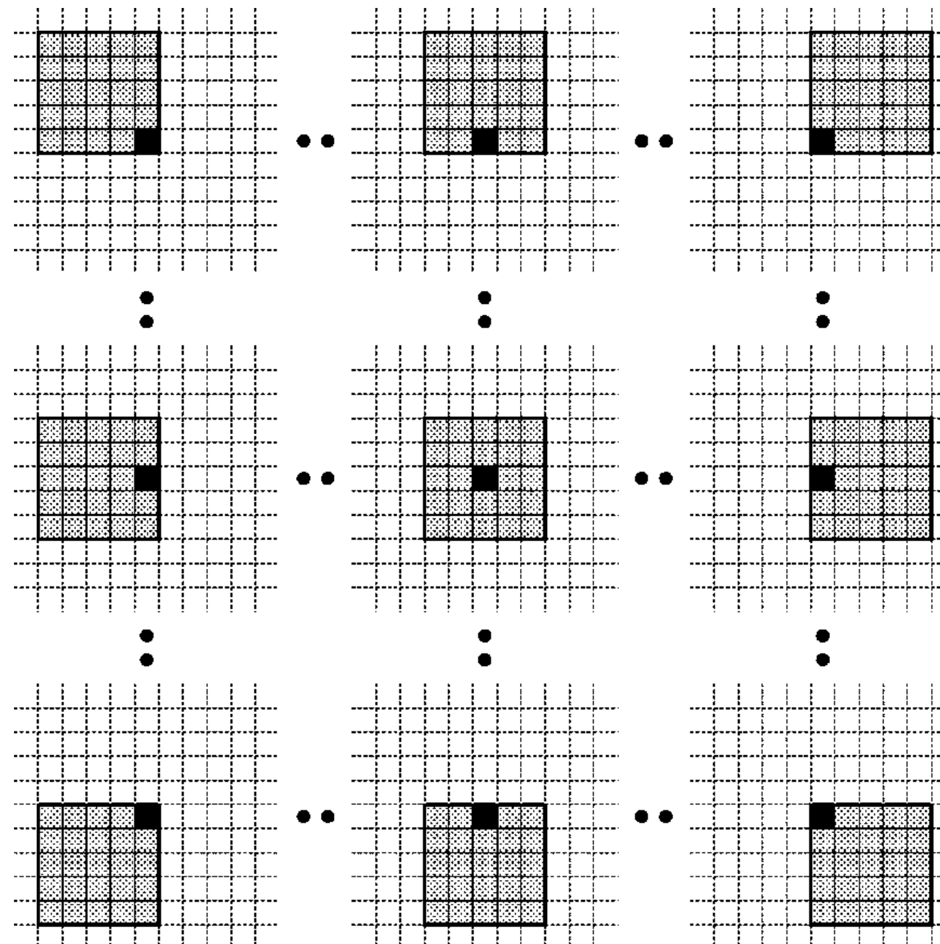
Fronto-Parallel Assumption

- The disparity is assumed to be the same in the entire matching window
  - equivalent to assuming constant depth



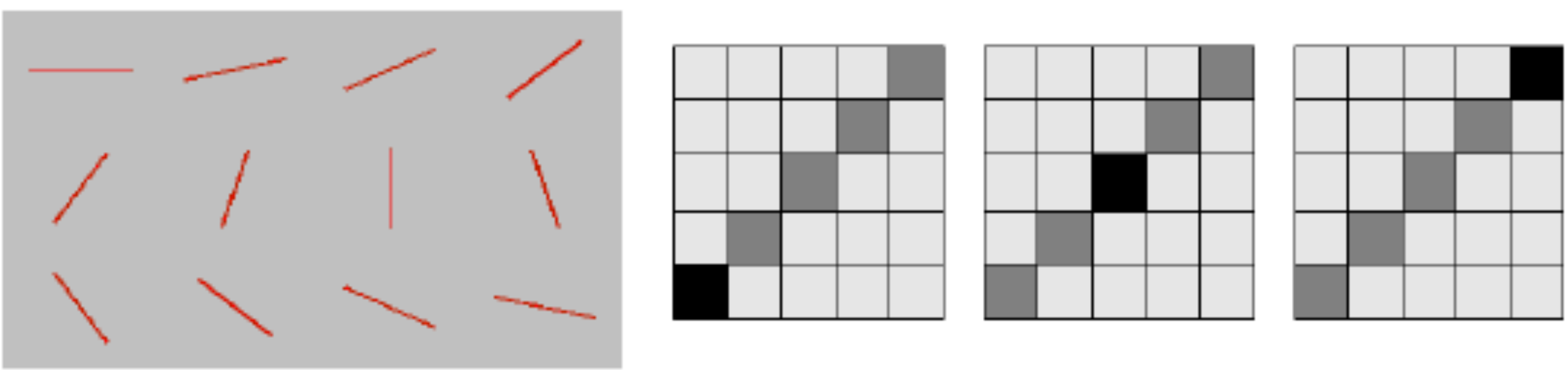
Shiftable Windows

- Avoid having using matching windows that straddle two surfaces
  - Disparity will not be constant for all pixels
- Shift the window around the reference pixel
  - Keep the one with min cost (max NCC)



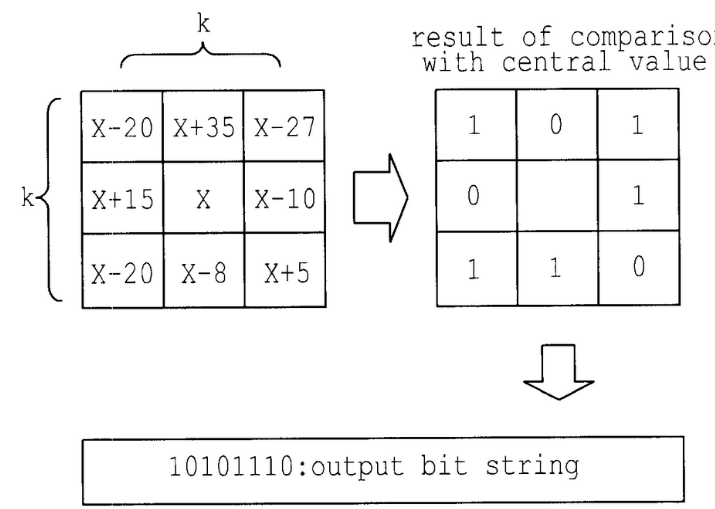
Rod-shaped Filters

- Instead of square windows aggregate cost in rod-shaped shiftable windows
- Search for one that minimizes the cost (assume that it is an iso-disparity curve)



Alternative Dissimilarity Measures

- Rank and Census transforms
- Rank transform:
  - Define window containing R pixels around each pixel
  - Count the number of pixels with lower intensities than center pixel in the window
  - Replace intensity with rank (0..R-1)
  - Compute SAD on rank-transformed images
- Census transform:
  - Use bit string, defined by neighbors, instead of scalar rank
- Robust against illumination changes



Locally Adaptive Support

Apply weights to contributions of neighboring pixels according to similarity and proximity

(a) left support window (b) right support window (c) color difference between (a) and (b)

Locally Adaptive Support

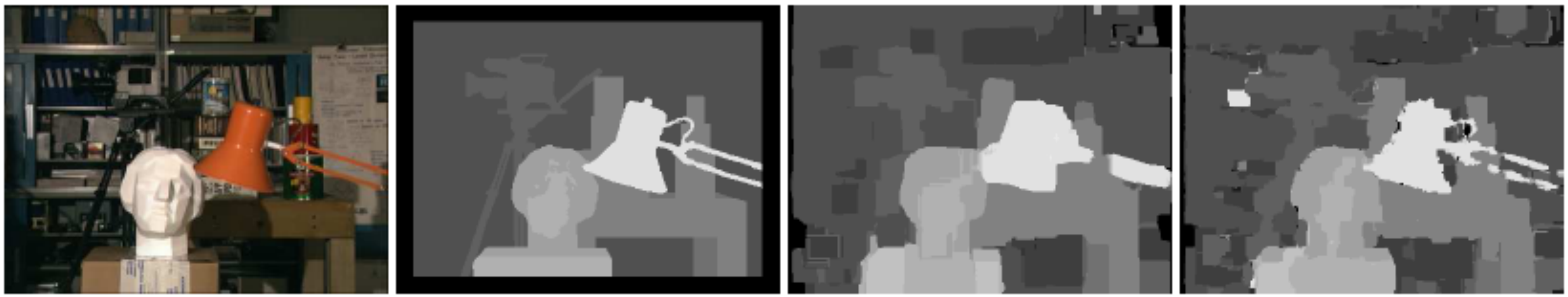
- Similarity in CIE Lab color space:

$$\Delta c_{pq} = \sqrt{(L_p - L_q)^2 + (a_p - a_q)^2 + (b_p - b_q)^2}$$

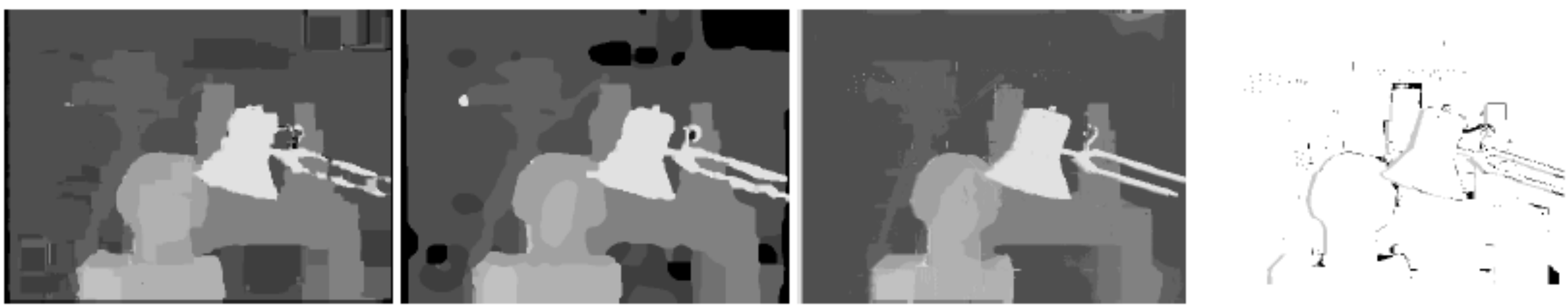
- Proximity: Euclidean distance

- Weights: $w(p,q) = k \cdot \exp\left(-(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p})\right)$

Locally Adaptive Support: Results



(a) left image (b) ground truth (c) shiftable win. [7] (d) compact win. [3]



(e) variable win. [4] (f) Bay. diff. [19] (g) our result (h) bad pixels (error > 1)