3D Computer Vision II

Two View Geometry
Part 3 – Fundamental Matrix Computation

Nassir Navab

based on a course given at UNC by Marc Pollefeys & the book “Multiple View Geometry” by Hartley & Zisserman

November 09, 2010

Outline – Two-View Geometry

- Epipolar Geometry
- 3D Reconstruction
- Fundamental Matrix Computation

Reminder – 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,…,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 Fundamental Matrix F

Algebraic representation of Epipolar Geometry:

$$x \mapsto l'$$

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

Points From Lines and Vice-versa

- Intersections of lines

  The intersection of two lines l and l' is $\mathrm{x} = \mathrm{l} \times \mathrm{l}' = [l]_x l' = -[l']_x l$

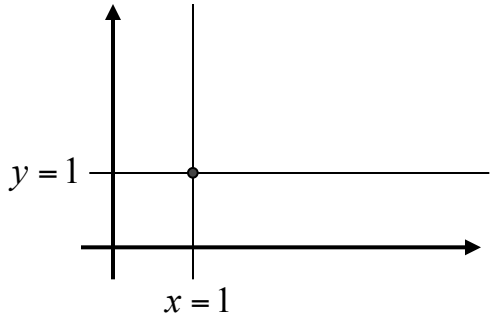
- Line joining two points

  The line through two points x and x' is $\mathrm{l} = \mathrm{x} \times \mathrm{x}' = [\mathrm{x}]_x \mathrm{x}' = -[\mathrm{x}']_x \mathrm{x}$

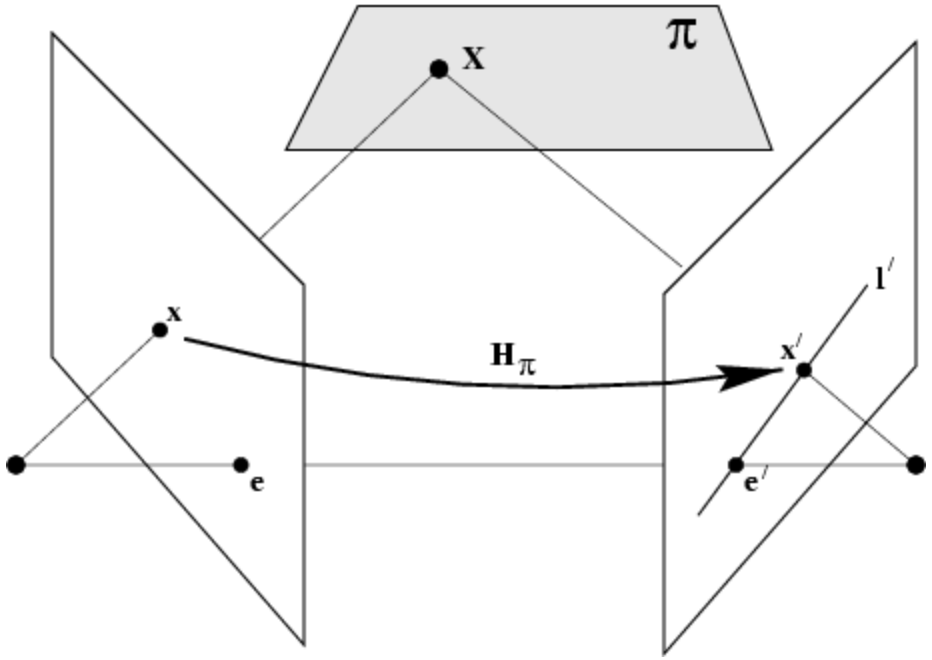
- Anti-symmetric Matrix

$$[l]_x = \begin{bmatrix} 0 & -l_3 & l_2 \\ l_3 & 0 & -l_1 \\ -l_2 & l_1 & 0 \end{bmatrix}$$

Example



The Fundamental Matrix F – Geometric Derivation



$$x' = H_\pi x$$

$$l' = e' \times x' = [e']_\times H_\pi x = Fx$$

mapping from 2-D to 1-D family (rank 2)

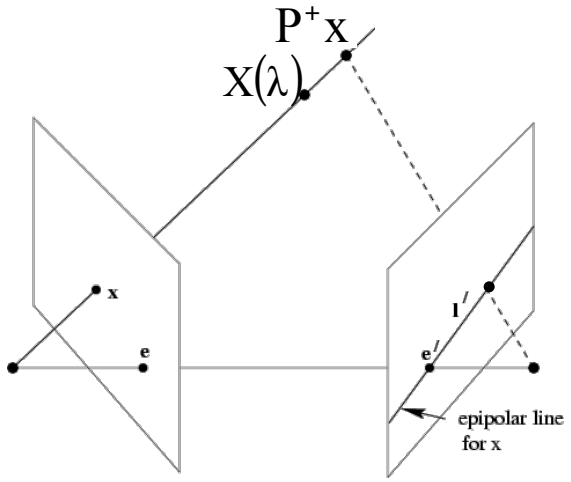
The Fundamental Matrix F

Algebraic derivation:

$$X(\lambda) = (1-\lambda)C + \lambda P^{+}x \qquad \left(P^{+}P = I\right)$$

$$l = P'C \times P'P^{+}x$$

$$F = \left[e'\right]_{\times} P'P^{+}$$



(note: doesn't work for C=C' ⇒ F=0)

The Fundamental Matrix F

Correspondence condition:

The fundamental matrix satisfies the condition that for any pair of corresponding points $x \leftrightarrow x'$ 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 $x'^{T}Fx=0$ 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: $l'=Fx$ & $l=F^{T}x'$

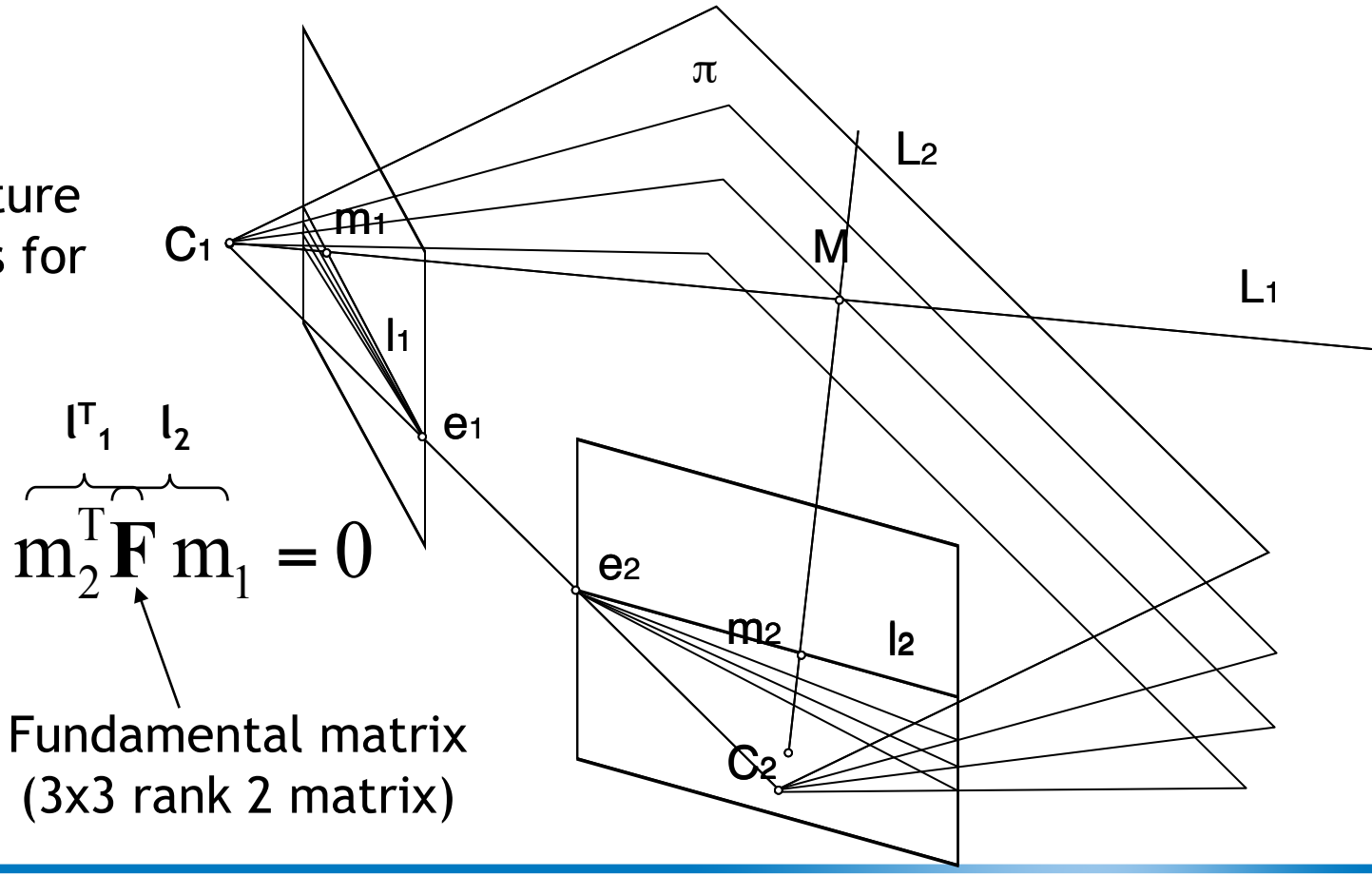
(iii) Epipoles: on all epipolar lines, thus $e'^{T}Fx=0$, $\forall x$ $\Rightarrow e'^{T}F=0$, similarly $Fe=0$

(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)

Epipolar Geometry

Underlying structure in set of matches for rigid scenes



$$\underbrace{l'^{T}_{1} \quad l_2}_{} \qquad m_2^T \mathbf{F} m_1 = 0$$

Fundamental matrix
(3x3 rank 2 matrix)

Epipolar Geometry

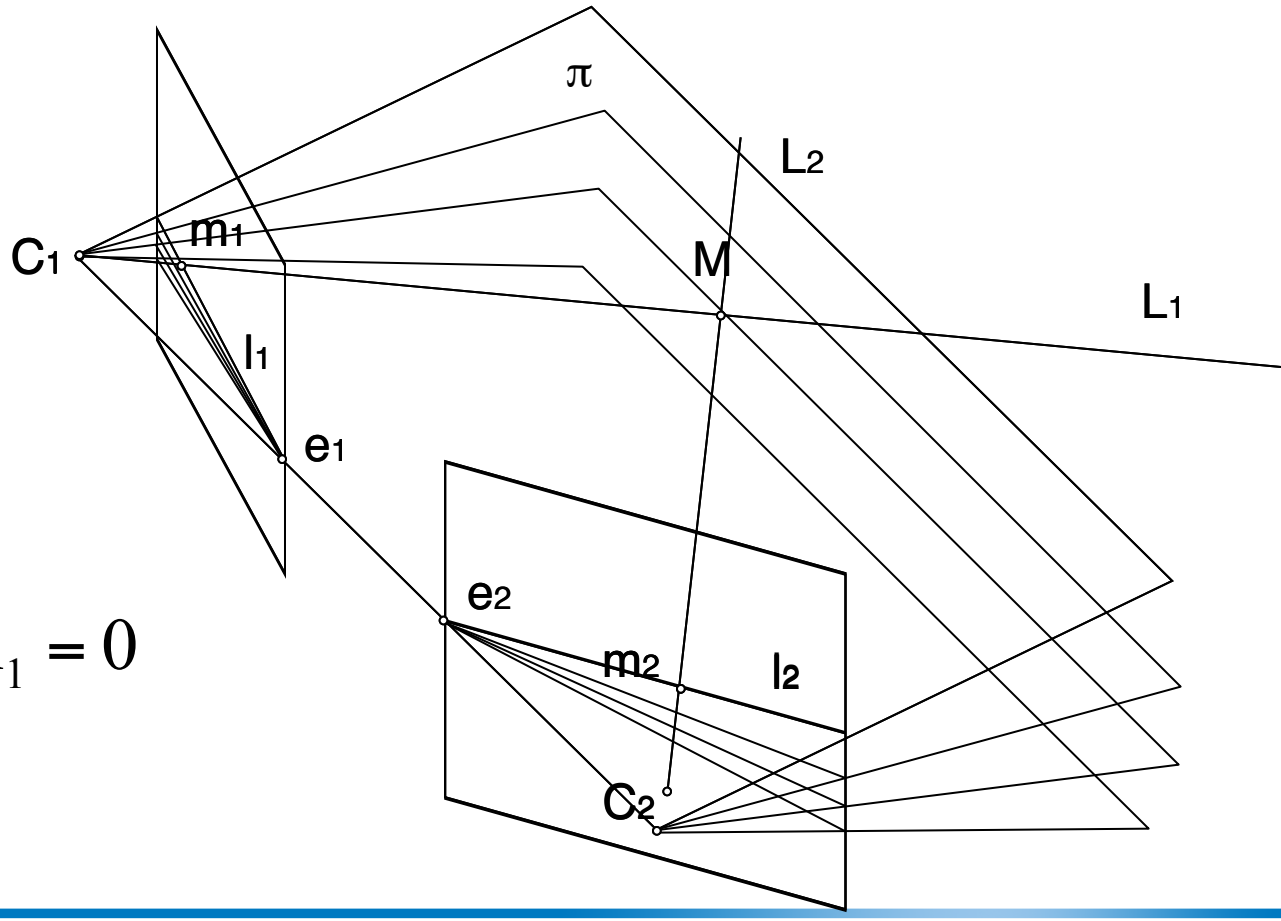
$$l_1 = [e_1]_\times x_1$$

$$l_2 = P_2^{+T} P_1^T l_1$$

$$\pi = P_1^T l_1$$

$$x_2^T l_2 = 0$$

$$x_2^T P_2^{+T} P_1^T [e_1]_\times x_1 = 0$$



Epipolar Geometry

Canonical representation:

$$P = [I \mid 0] \quad P' = [[e']_\times F + e' v^T \mid \lambda e']$$

1. Computable from corresponding points
2. Simplifies matching
3. Allows to detect wrong matches
4. Related to calibration

Epipolar Geometry – Basic Equations

$$\mathbf{x}'^{\mathrm{T}} \mathrm{F}\mathbf{x} = 0$$

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

... separate knowns from unknowns:

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

(data)

(unknowns)
(linear)

Epipolar Geometry – Basic Equations

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

$$\mathrm{A}\mathrm{f} = 0$$

The Singularity Constraint

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

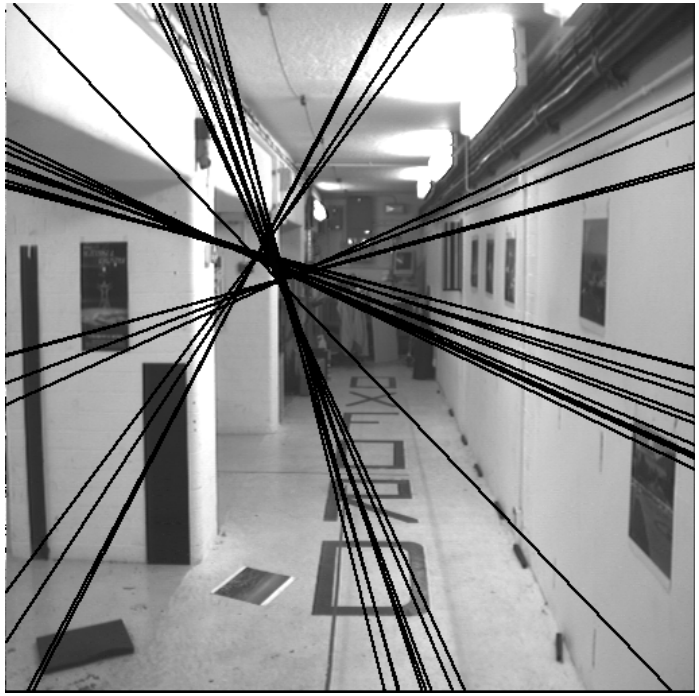
SVD from linearly computed F matrix (rank 3)

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

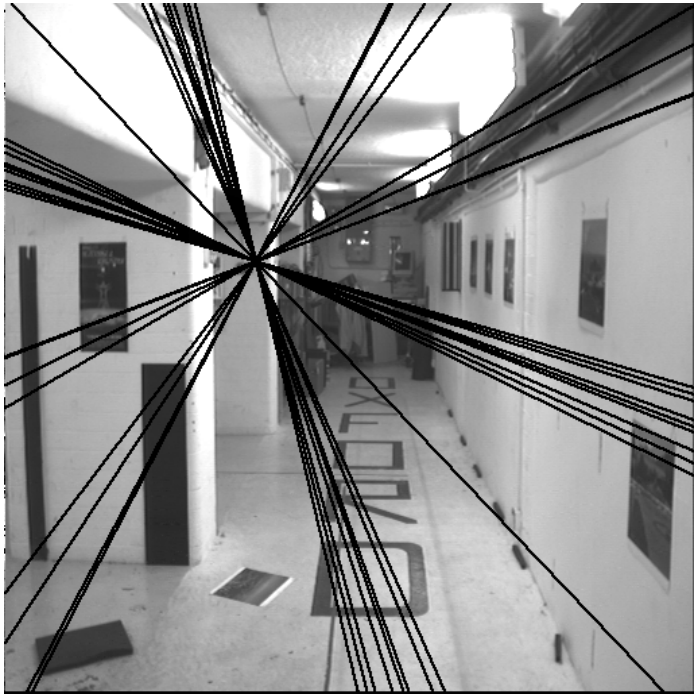
Compute closest rank-2 approximation $\min \|F - F'\|_F$

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

The Singularity Constraint



Non singular **F**



Forcing singularity using the SVD method

The Minimum Case – 7 Point Correspondences

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_7 x_7 & x'_7 y_7 & x'_7 & y'_7 x_7 & y'_7 y_7 & y'_7 & x_7 & y_7 & 1 \end{bmatrix} \mathrm{f} = 0$$

$$\mathrm{A} = \mathrm{U}_{7\mathrm{x}9}\mathrm{diag}(\sigma_1,\ldots,\sigma_7,0,0)\mathrm{V}_{9\mathrm{x}9}{}^{\mathrm{T}}$$

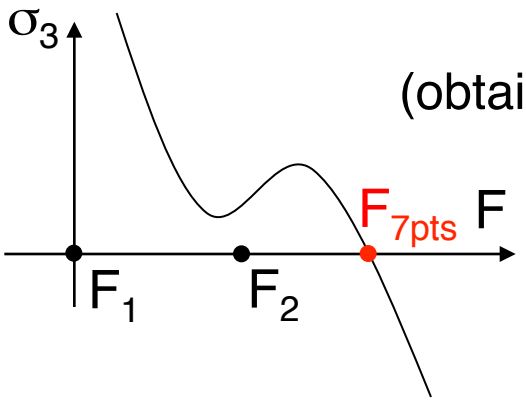
$$\Rightarrow \mathrm{A}[\mathrm{V}_8\mathrm{V}_9] = 0_{9\mathrm{x}2}$$

$$\mathrm{x}'^{\mathrm{T}}_i(\mathrm{F}_1 + \lambda\mathrm{F}_2)\mathrm{x}_i = 0, \forall i = 1\ldots7$$

One parameter family of solutions

But $F_1+\lambda F_2$ not automatically rank 2

The Minimum Case – Impose Rank 2



(obtain 1 or 3 solutions)

$$\det(F_1 + \lambda F_2) = a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0$$ (cubic equation)

One or three solutions (only real solutions are acceptable)

The 8-Point Algorithm

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

The Unnormalized 8-Point Algorithm

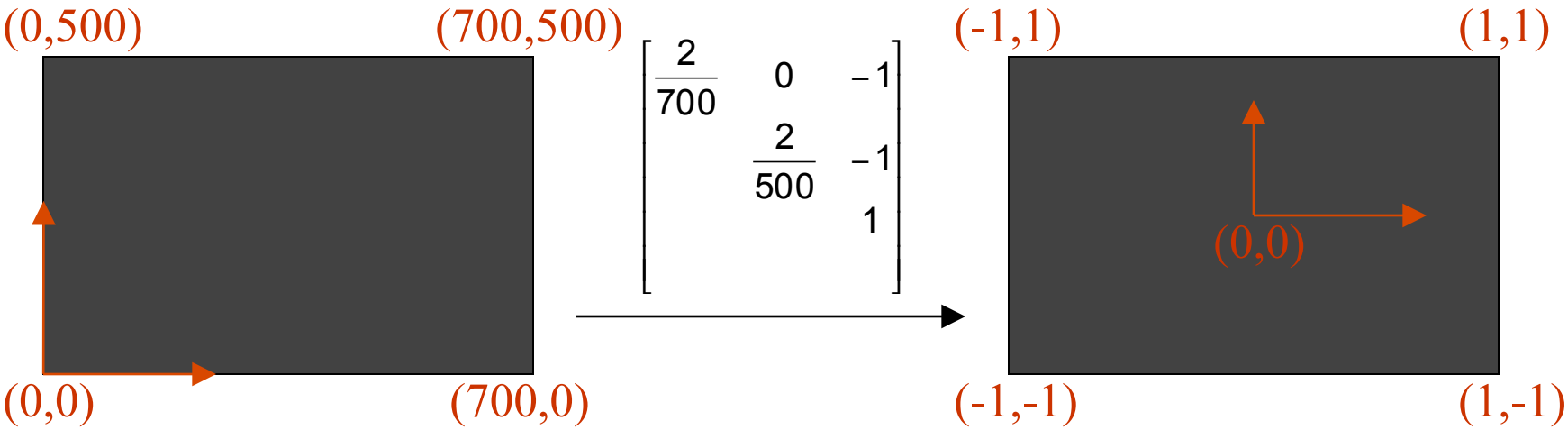
$$\begin{bmatrix} x_1x_1' & y_1x_1' & x_1' & x_1y_1' & y_1y_1' & y_1' & x_1 & y_1 & 1 \\ x_2x_2' & y_2x_2' & x_2' & x_2y_2' & y_2y_2' & y_2' & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_nx_n' & y_nx_n' & x_n' & x_ny_n' & y_ny_n' & y_n' & x_n &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

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

Orders of magnitude difference
between columns of data matrix
→ least-squares yields poor results

The Normalized 8-Point Algorithm

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



Least squares yields good results (Hartley, PAMI´97)

Algebraic Minimization

Possible to iteratively minimize algebraic distance subject to det F=0 (see book if interested).

$$\mathrm{F} = \mathrm{M}[\mathrm{e}]_{\mathrm{x}} \Rightarrow f = Em$$

where $$E = \begin{bmatrix} [\mathrm{e}]_{\mathrm{x}} & 0 & 0 \\ 0 & [\mathrm{e}]_{\mathrm{x}} & 0 \\ 0 & 0 & [\mathrm{e}]_{\mathrm{x}} \end{bmatrix}$$

Minimize $\|AEm\|$ subject to $\|Em\| = 1$

Geometric Distance

- Gold standard
- Sampson error
- Symmetric epipolar distance

Gold Standard

Maximum Likelihood Estimation (= least-squares for Gaussian noise)

$$\sum_i d(\mathrm{x}_i, \hat{\mathrm{x}}_i)^2 + d(\mathrm{x}'_i, \hat{\mathrm{x}}'_i)^2 \quad \text{subject to} \quad \hat{\mathrm{x}}'^{\mathrm{T}} \mathrm{F} \hat{\mathrm{x}} = 0$$

Initialize: normalized 8-point, (P,P ‘) from F, reconstruct X_i v

Parameterize:

$$\mathrm{P} = [\mathrm{I} \,|\, 0], \mathrm{P}' = [\mathrm{M} \,|\, \mathrm{t}], \mathrm{X}_i \quad \text{(overparametrized)}$$

$$\hat{\mathrm{x}}_i = \mathrm{P}\mathrm{X}_i, \hat{\mathrm{x}}_i = \mathrm{P}'\mathrm{X}_i$$

Minimize cost using Levenberg-Marquardt
(preferably sparse LM, see book)

Gold Standard

Alternative, minimal parametrization (with a=1)

$$\mathbf{F} = \begin{bmatrix} a & b & -ax-by \\ c & d & -cx-dy \\ -ax'-cy' & -bx'-dy' & F_{33} \end{bmatrix}$$

$$where\ F_{33} = (ax+by)x'+(cx+dy)y'$$ (note (x,y,1) and (x ‘,y ‘,1) are epipoles)

problems:

- a=0 → pick largest of a,b,c,d to fix
- epipole at infinity → pick largest of x,y,w and of x’,y’,w’

4x3x3=36 parametrizations!

reparametrize at every iteration, to be sure

First-order Geometric Error (Sampson Error)

$$\sum e^T \left(JJ^T\right)^{-1} e \qquad \sum \frac{e^T e}{JJ^T}$$

(one eq./point $\Rightarrow$ JJ^T scalar)

$$e = \sum x'^T Fx = 0 \qquad \frac{\partial e}{\partial x_i} = x'^T F \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$JJ^T = \left(x'^T F\right)_1^2 + \left(x'^T F\right)_2^2 + \left(Fx\right)_1^2 + \left(Fx\right)_2^2$$

$$\sum \frac{e^T e}{JJ^T} \qquad \sum \frac{x'^T Fx}{\left(x'^T F\right)_1^2 + \left(x'^T F\right)_2^2 + \left(Fx\right)_1^2 + \left(Fx\right)_2^2}$$

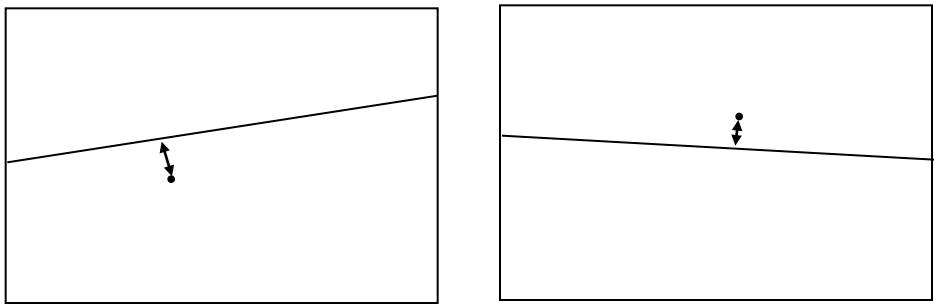
(problem if some x is located at epipole)

advantage: no subsidiary variables required

Symmetric Epipolar Error

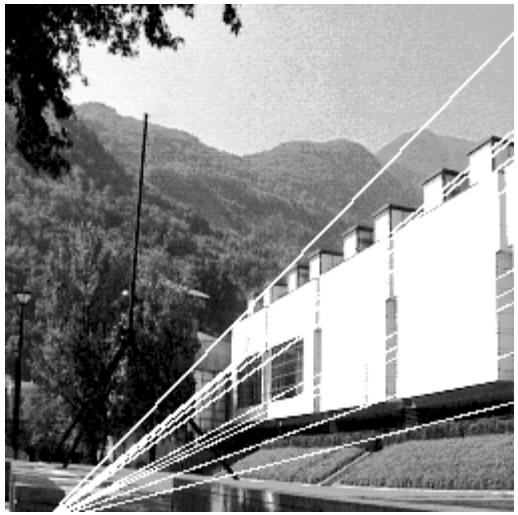
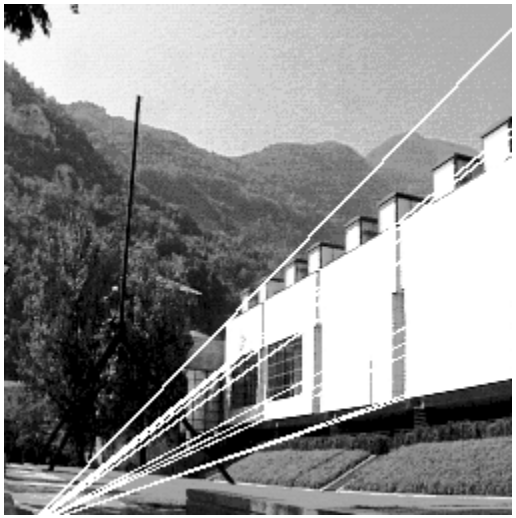
$$\sum_i d\left(\mathrm{x}'_i, F\mathrm{x}_i\right)^2 + d\left(\mathrm{x}_i, \mathrm{F}^{\mathrm{T}}\mathrm{x}'_i\right)^2$$

$$= \sum \mathrm{x}'^{\mathrm{T}}\,\mathrm{Fx}\left(\frac{1}{\left(\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\right)_1^2 + \left(\mathrm{x}'^{\mathrm{T}}\,\mathrm{F}\right)_2^2} + \frac{1}{\left(\mathrm{Fx}\right)_1^2 + \left(\mathrm{Fx}\right)_2^2}\right)$$

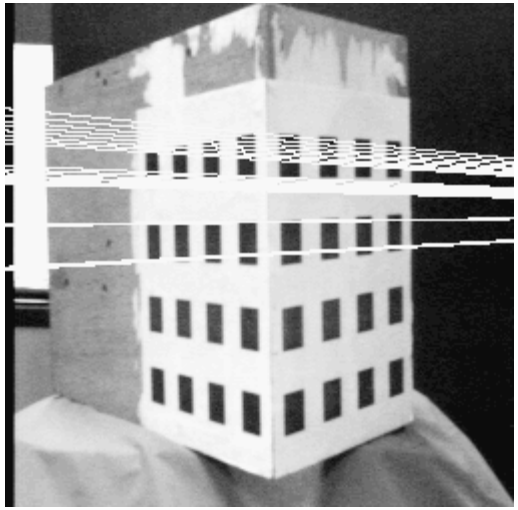


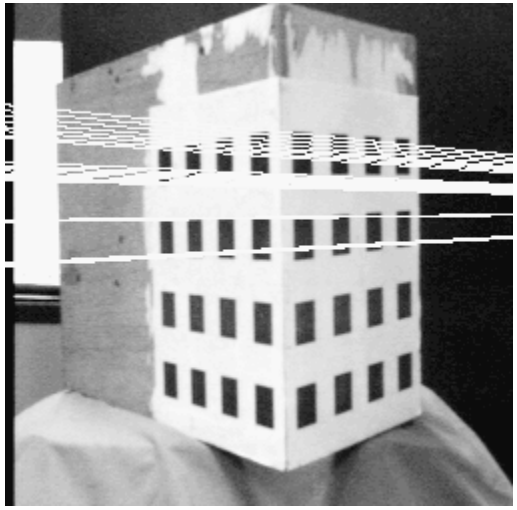
Some Experiments

Some Experiments

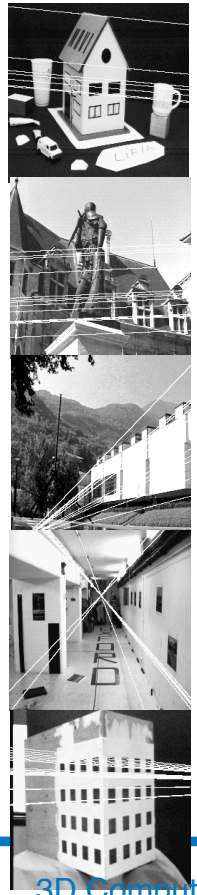



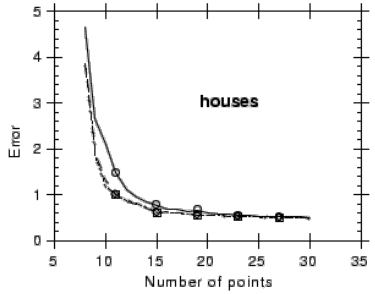
Some Experiments

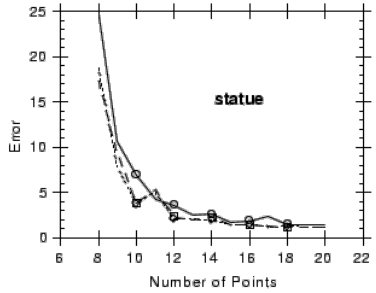


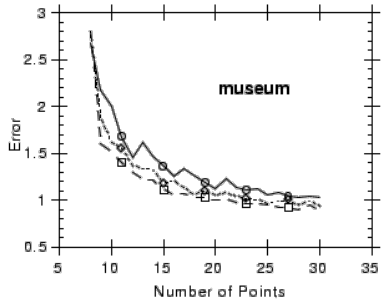


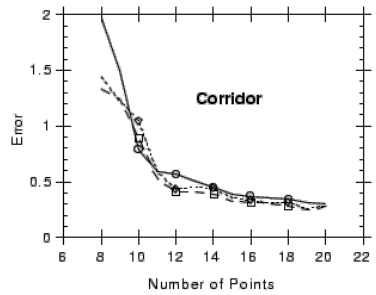
Some Experiments

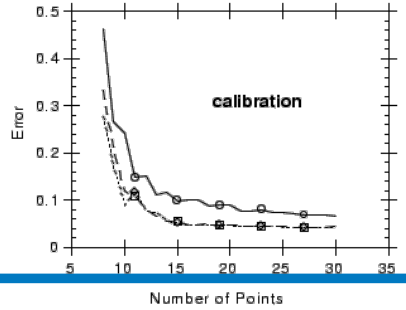












Residual error:

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

(for all points!)

Recommendations

- Do not use unnormalized algorithms
- Quick and easy to implement: 8-point normalized
- Better: enforce rank-2 constraint during minimization
- Best: Maximum Likelihood Estimation
  (minimal parameterization, sparse implementation)

Special Case

Enforce constraints for optimal results:

- Pure translation (2dof):

$$F = [e']_\times$$

- Planar motion (6dof),
- Calibrated case (5dof)

The Envelope of Epipolar Lines

What happens to an epipolar line if there is noise?

Monte Carlo

n=10

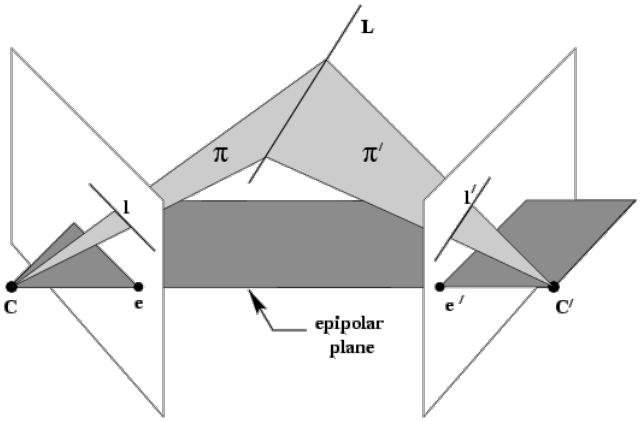
n=15

n=25

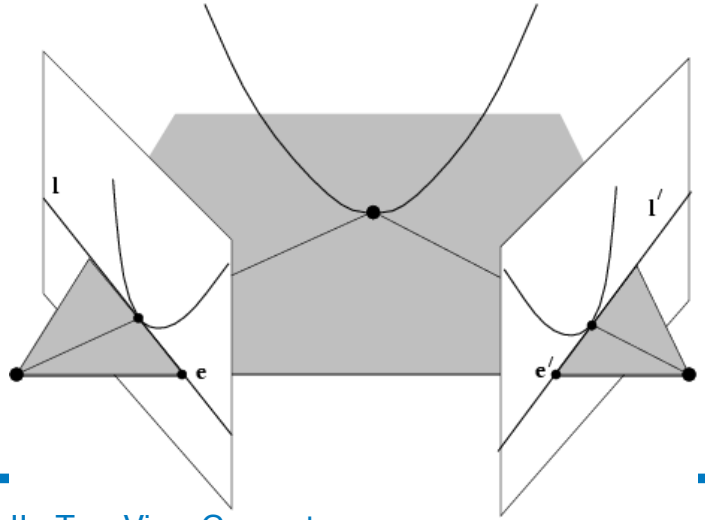
n=50

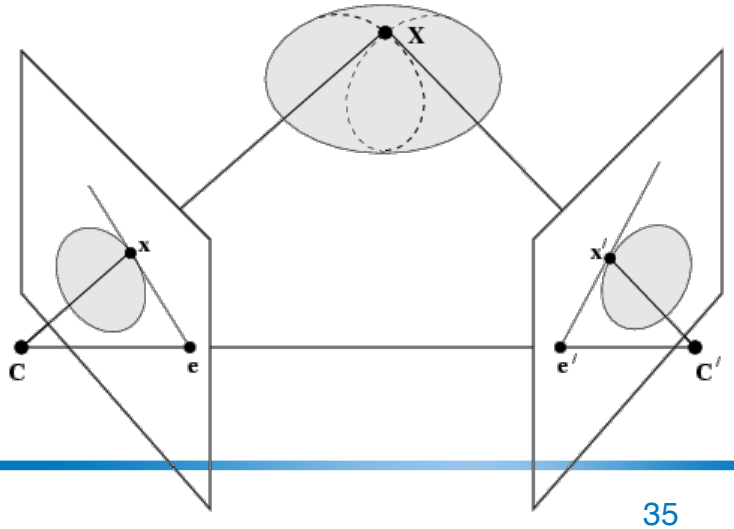
Other Entities

Lines give no constraint for two view geometry (but will for three and more views)



Curves and surfaces yield some constraints related to tangency





Automatic Computation of F

(i) Interest points
(ii) Putative correspondences
(iii) RANSAC
(iv) Non-linear re-estimation of F
(v) Guided matching

Repeat (iv) and (v) until stable

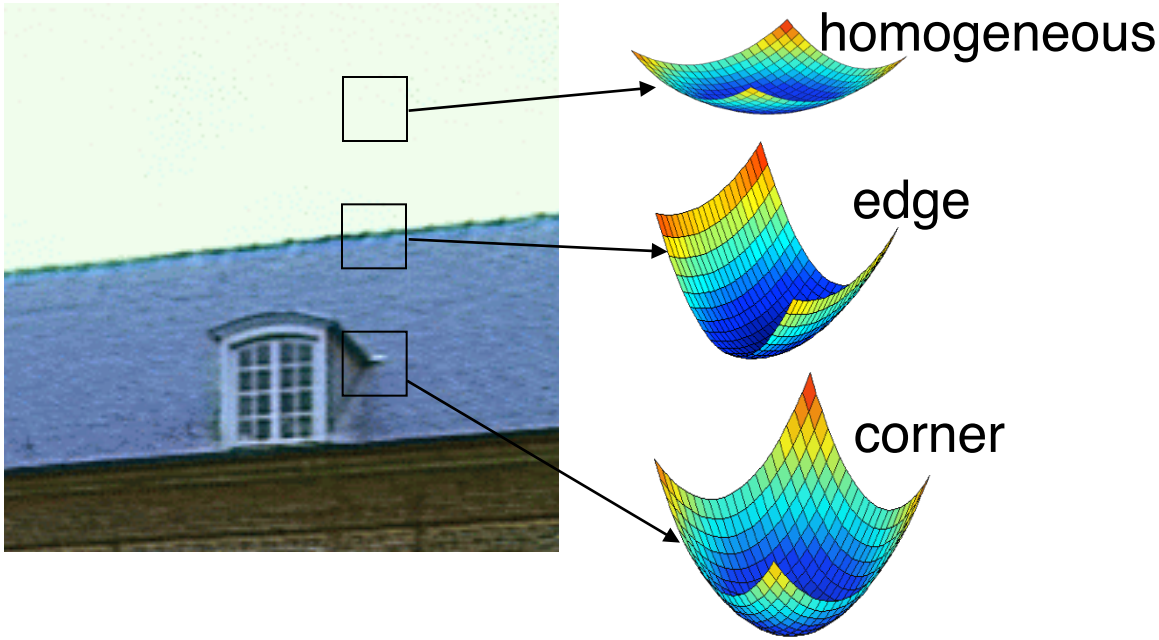
Feature Points

- Extract feature points to relate images
- Required properties:
  - Well-defined
    (i.e. neigboring points should all be different)
  - Stable across views
    (i.e. same 3D point should be extracted as feature for neighboring viewpoints)

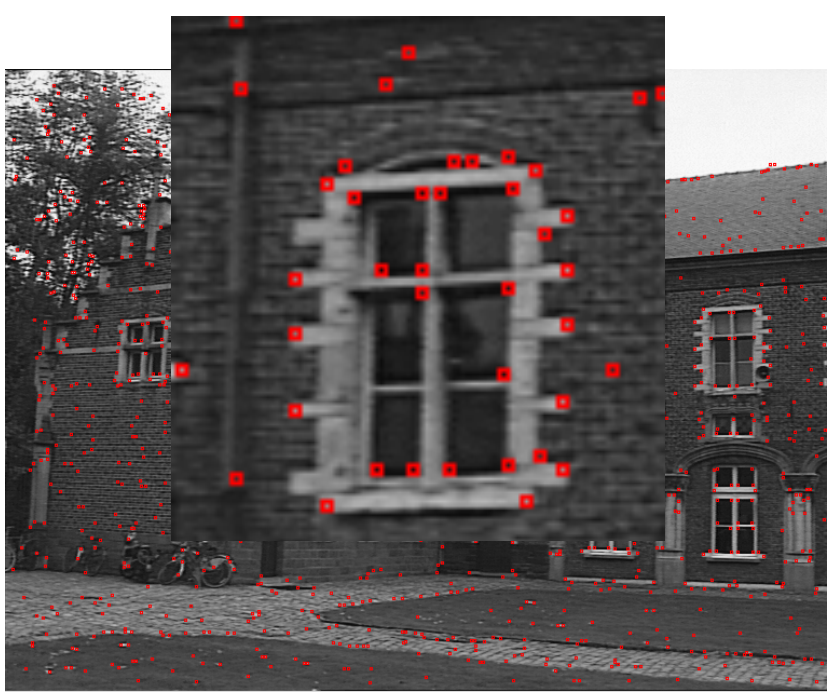
Feature Points

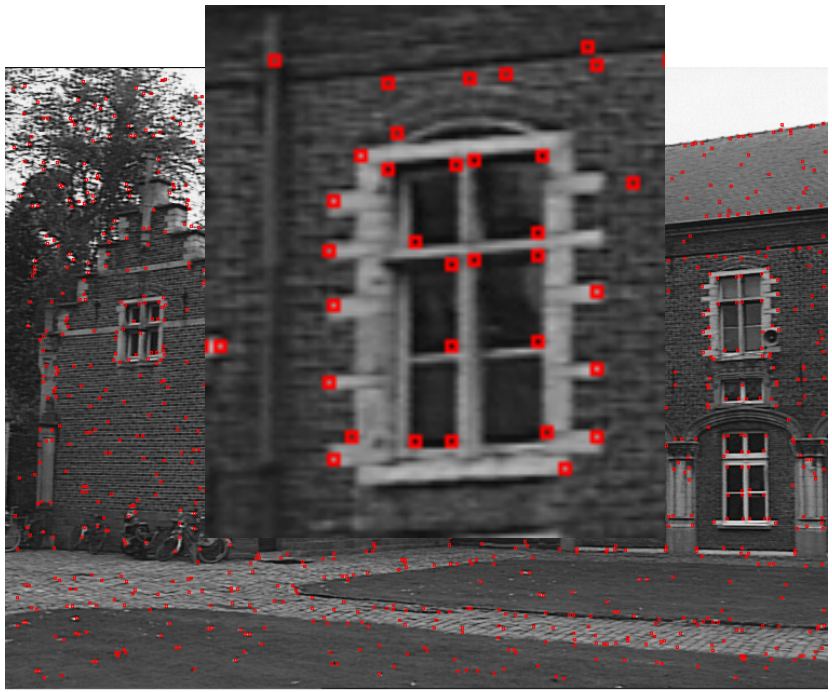
(e.g. Harris & Stephens ‘88; Shi & Tomasi ‘94)

Find points that differ as much as possible
from all neighboring points



Feature Points



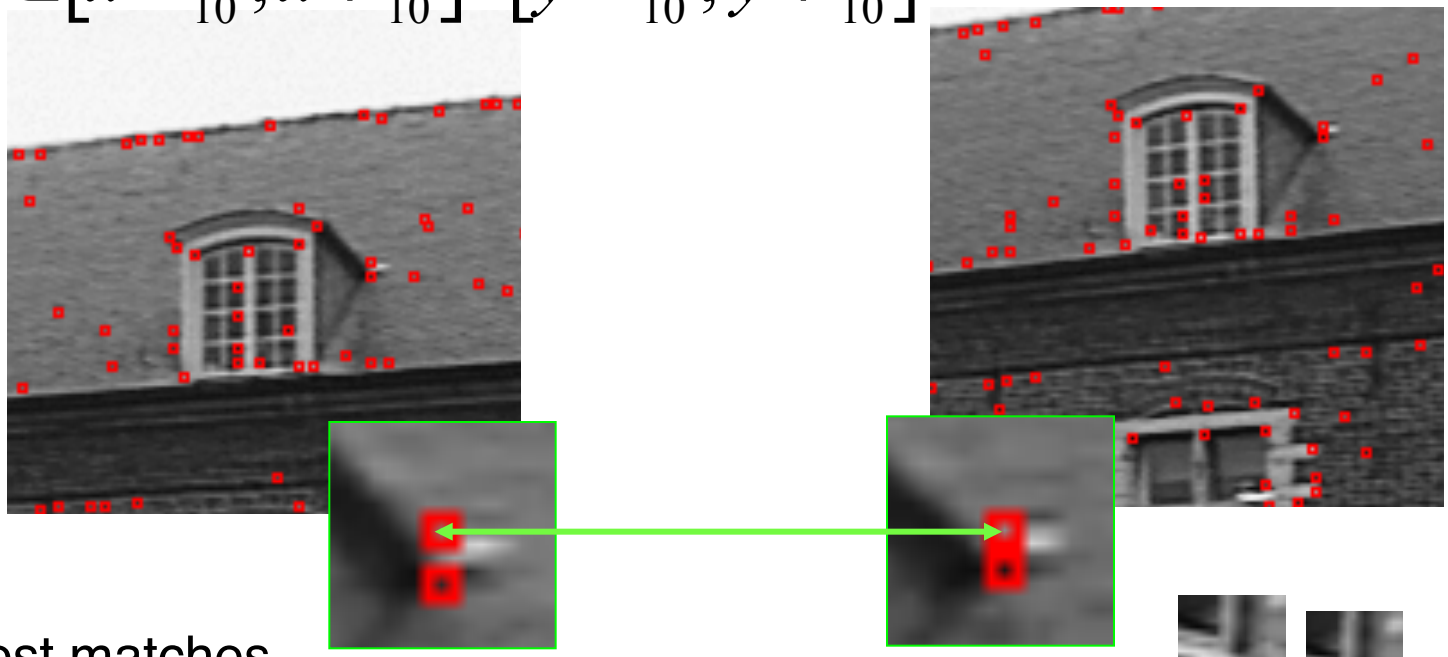


Select strongest features **(e.g. 1000/image)**

Feature Matching

Evaluate NCC for all features with similar coordinates

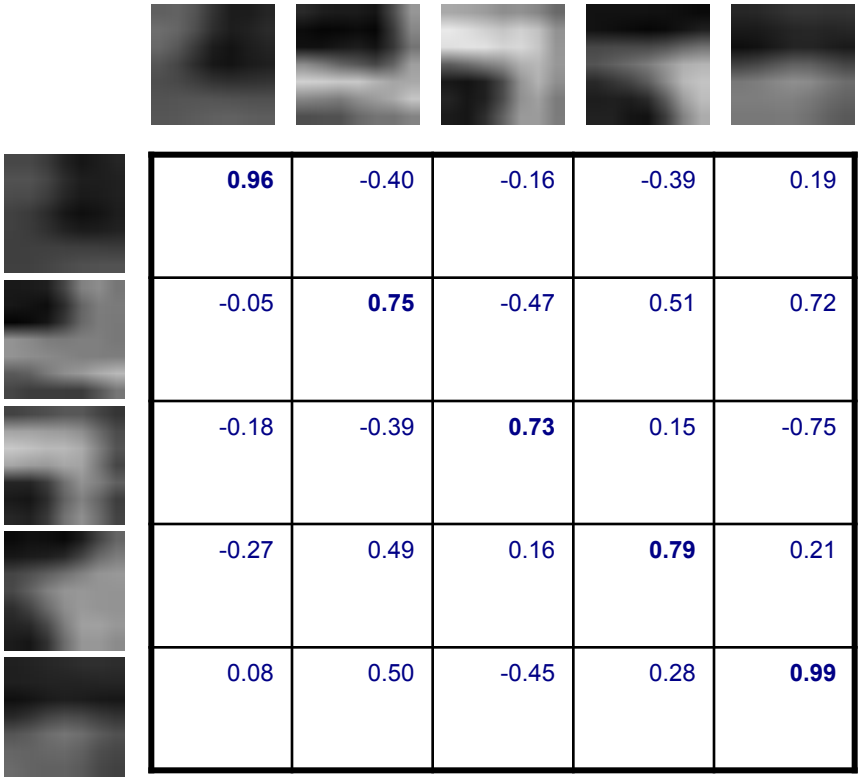
$$\text{e.g.}\ (x', y') \in \left[x - \tfrac{w}{10}, x + \tfrac{w}{10}\right] \times \left[y - \tfrac{h}{10}, y + \tfrac{h}{10}\right]$$



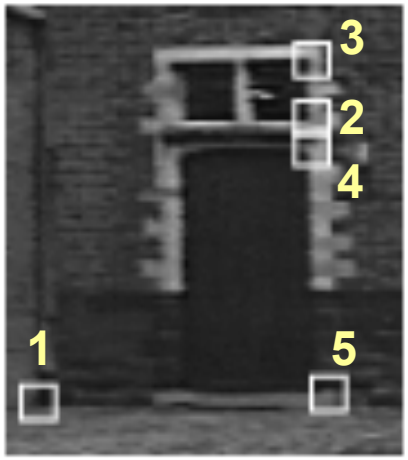
Keep mutual best matches

Still many wrong matches!

Feature Example



0.96	-0.40	-0.16	-0.39	0.19
-0.05	**0.75**	-0.47	0.51	0.72
-0.18	-0.39	**0.73**	0.15	-0.75
-0.27	0.49	0.16	**0.79**	0.21
0.08	0.50	-0.45	0.28	**0.99**



Gives satisfying results
for small image motions

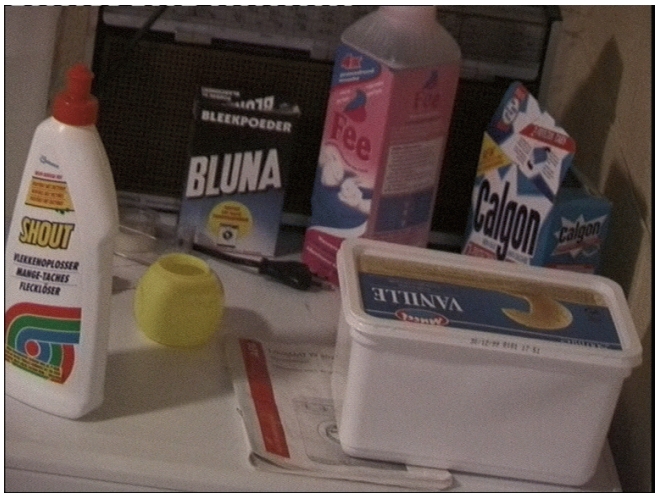
Wide-baseline Matching

- Requirement to cope with larger variations between images
  - Translation, rotation, scaling
  - Foreshortening

  } geometric transformations

  - Non-diffuse reflections
  - Illumination

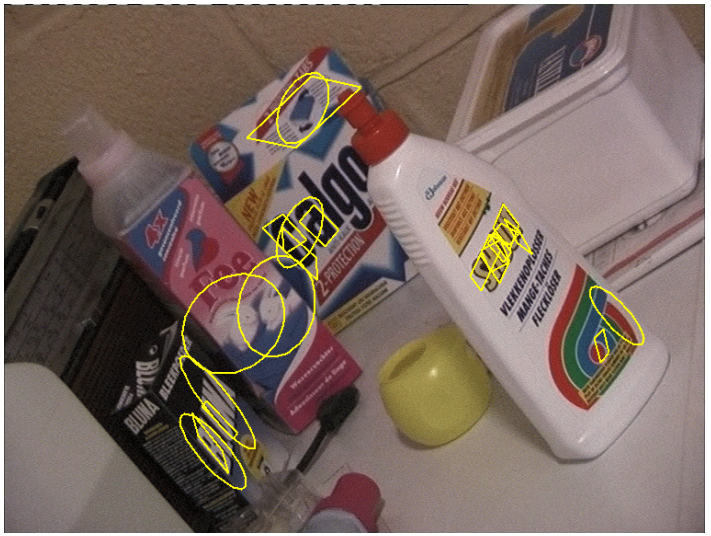
  } photometric changes



Wide-baseline Matching

(Tuytelaars and Van Gool BMVC 2000)

Wide baseline matching for two different region types



RANSAC

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
  Step 3.3 determine inliers
until $\Gamma(\#inliers,\#samples)<95\%$

generate hypothesis

verify hypothesis

Step 4. Compute F based on all inliers
Step 5. Look for additional matches
Step 6. Refine F based on all correct matches

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

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

Finding More Matches

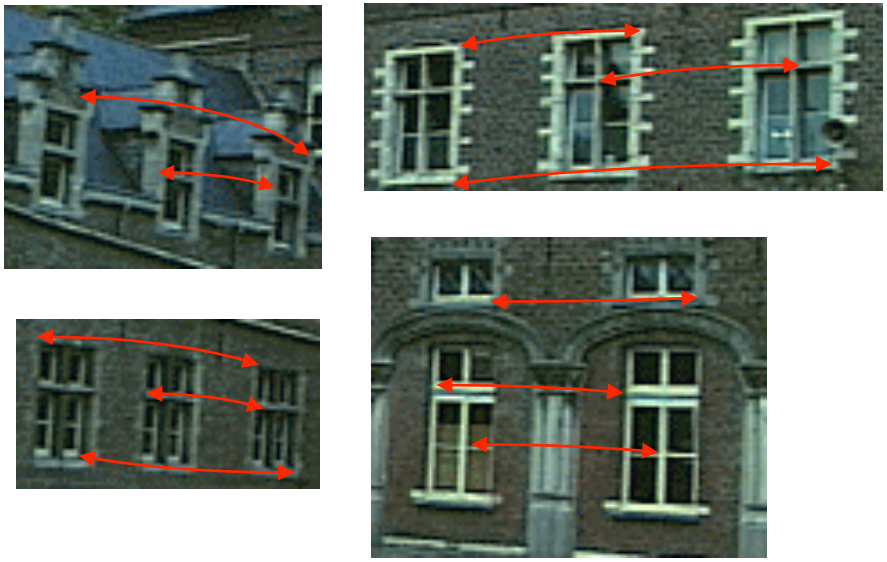
Restrict search range to neighborhood of epipolar line
(±1.5 pixels)
Relax disparity restriction (along epipolar line)

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)

More Problems

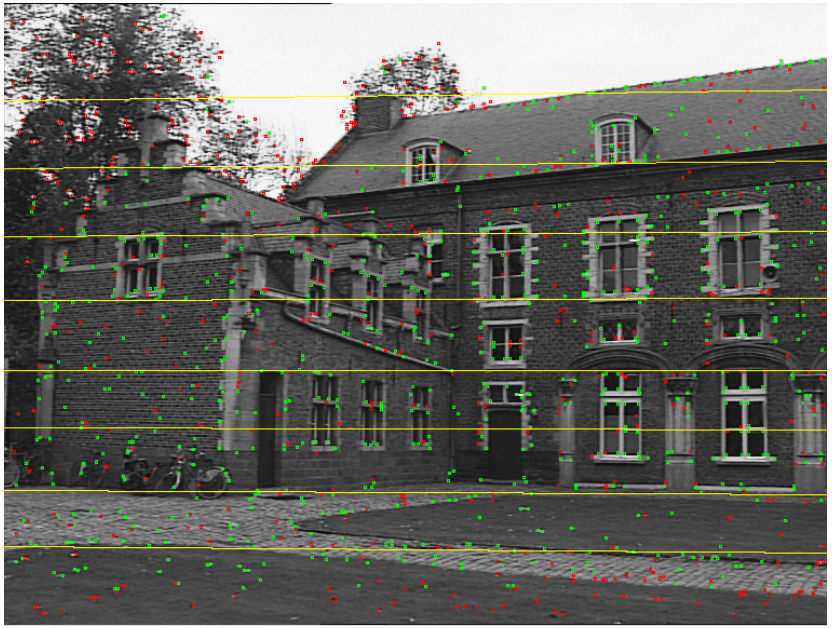
- Absence of sufficient features (no texture)
- Repeated structure ambiguity



•Robust matcher also finds support for wrong hypothesis
• Solution: detect repetition

(Schaffalitzky and Zisserman, BMVC '98)

Two-view Geometry

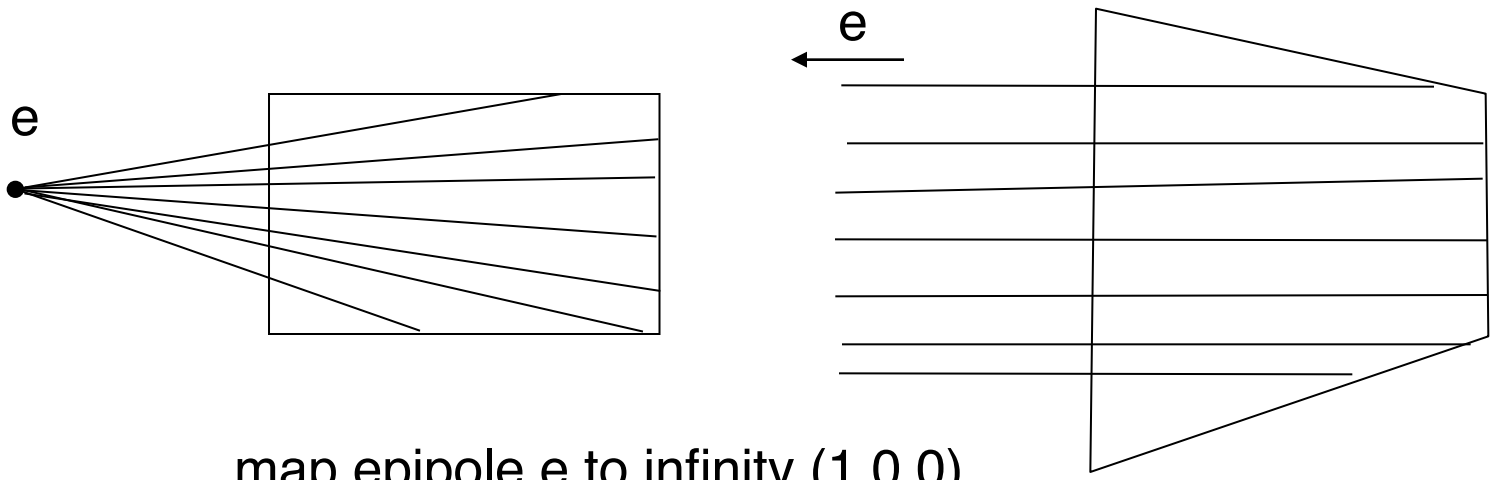


Geometric relations between two views are fully described by recovered 3x3 matrix **F.**

Image Rectification

Simplify stereo matching by warping the images.

Apply projective transformation so that epipolar lines correspond to horizontal scanlines



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

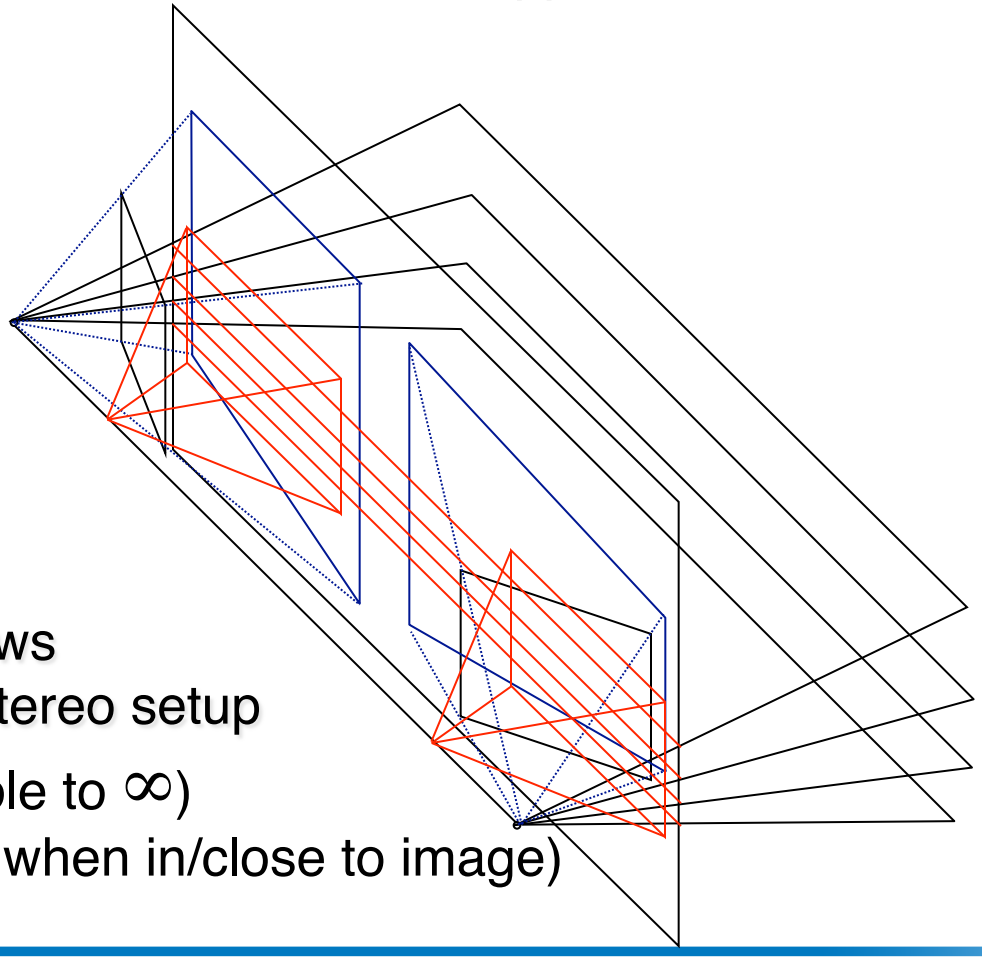
map epipole e to infinity (1,0,0)

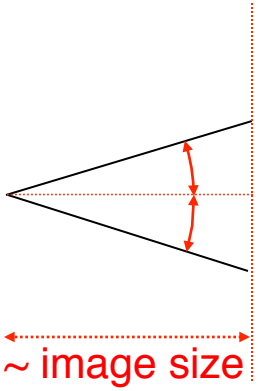
try to minimize image distortion

problem when epipole in (or close to) the image

Planar Rectification

(standard approach)



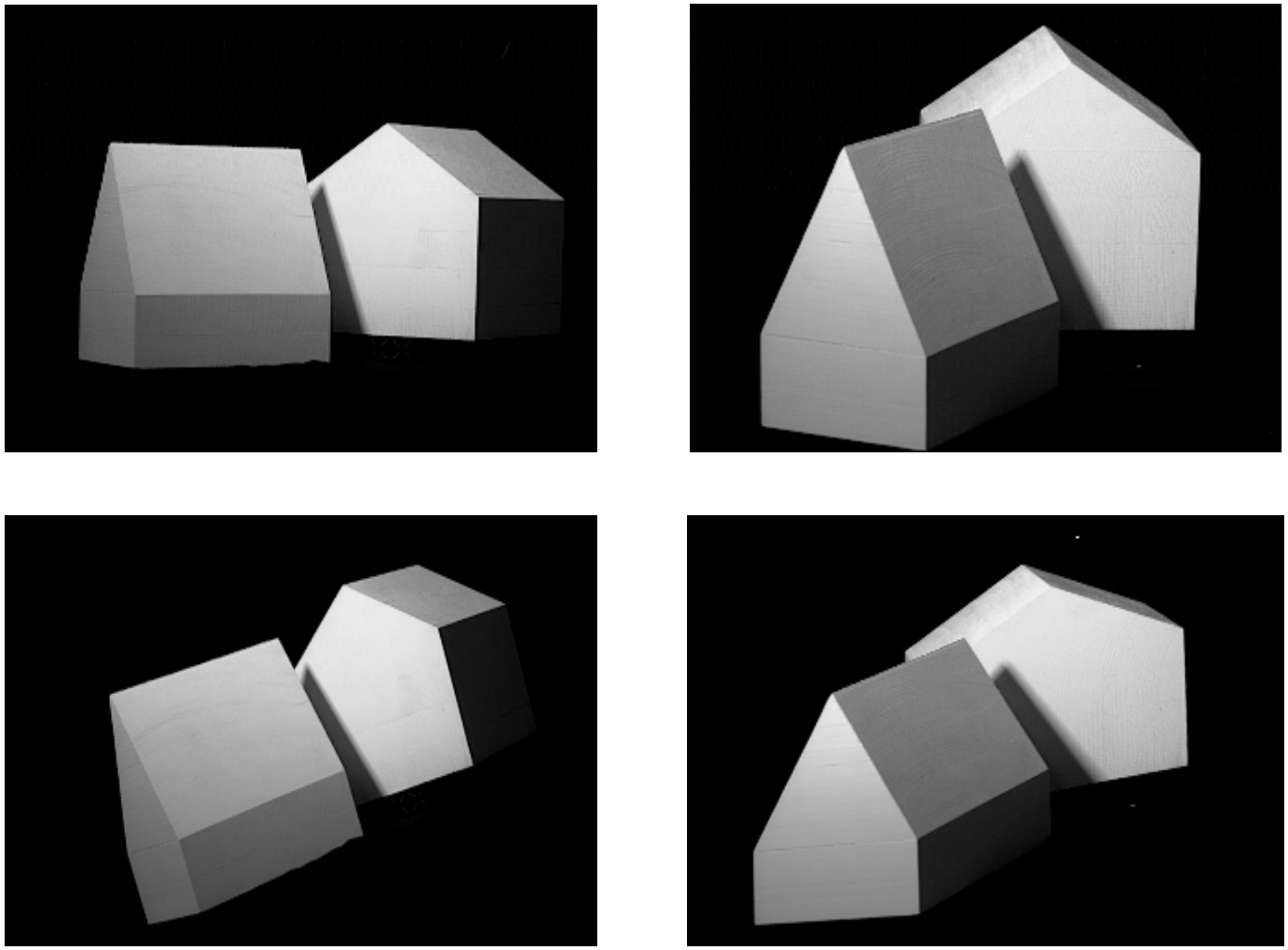


(calibrated)

Distortion minimization
(uncalibrated)

Bring two views
to standard stereo setup

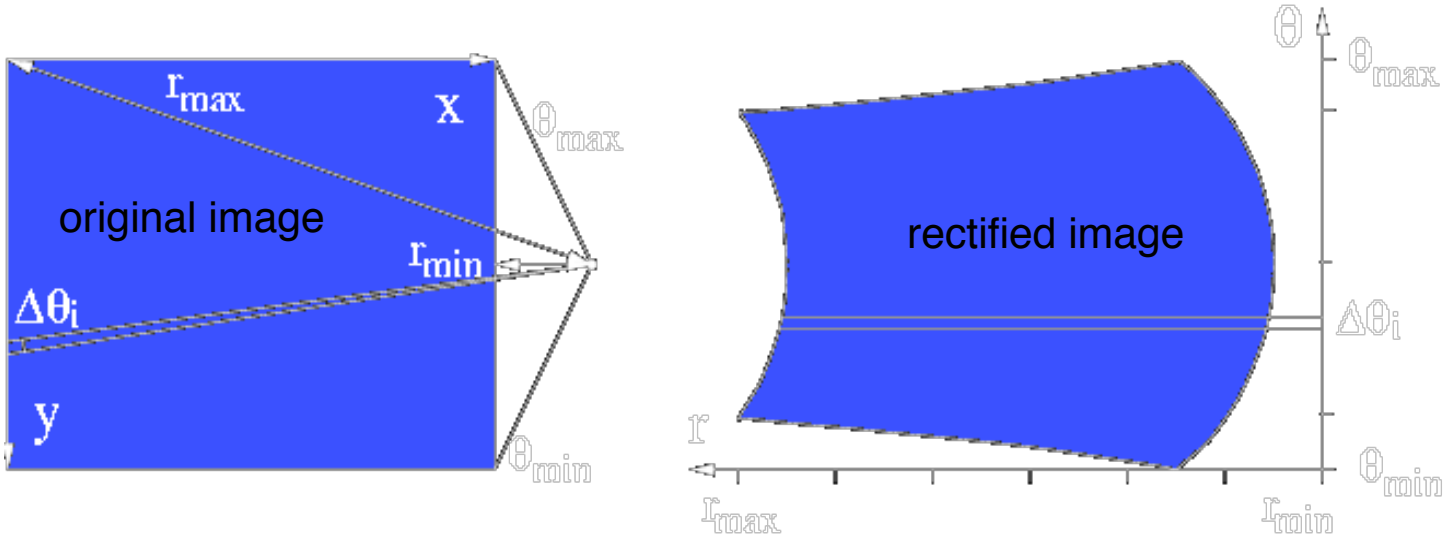
(moves epipole to ∞)
(not possible when in/close to image)



Polar Rectification

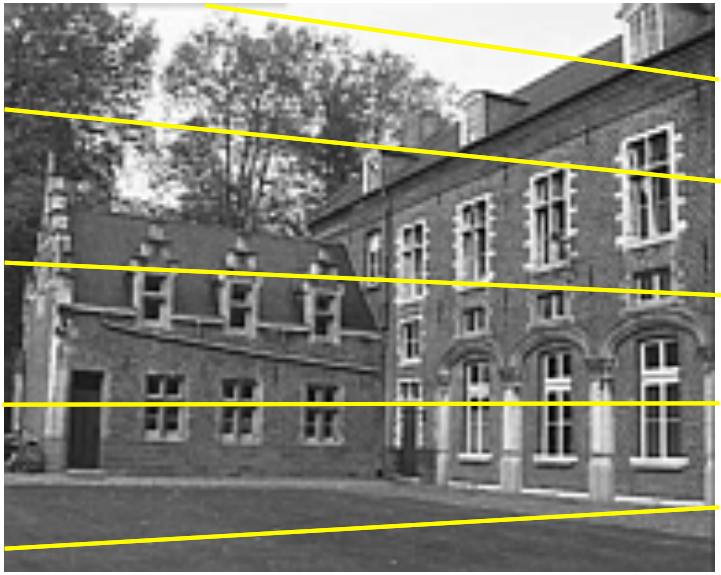
(Pollefeys et al. ICCV'99)

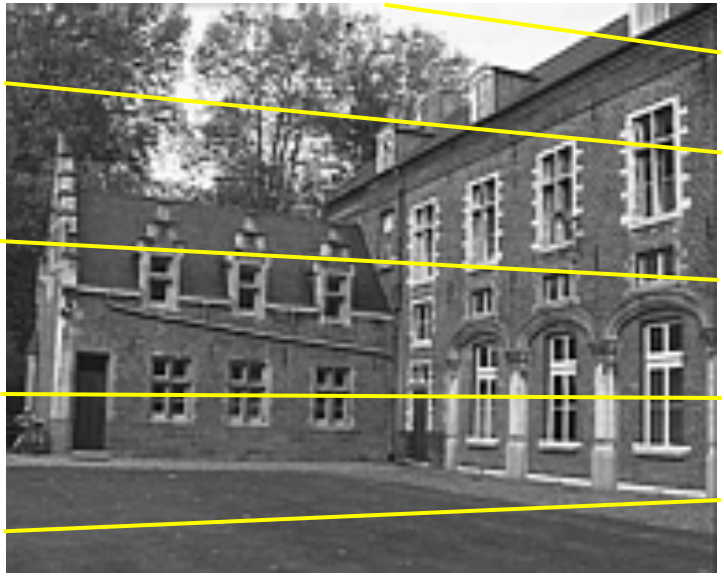
Polar re-parameterization around epipoles
Requires only (oriented) epipolar geometry
Preserve length of epipolar lines
Choose $\Delta\theta$ so that no pixels are compressed



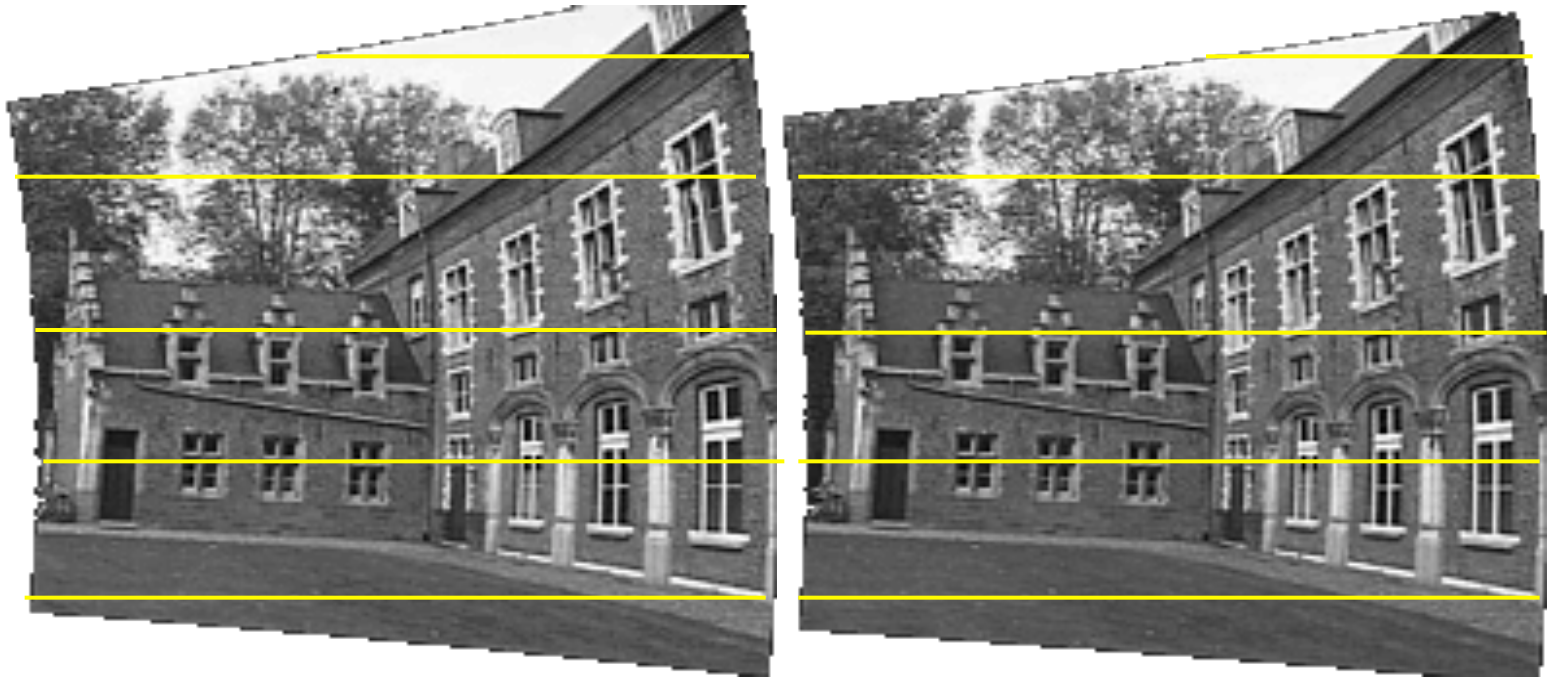
Works for all relative motions
Guarantees minimal image size

Polar Rectification :: Example

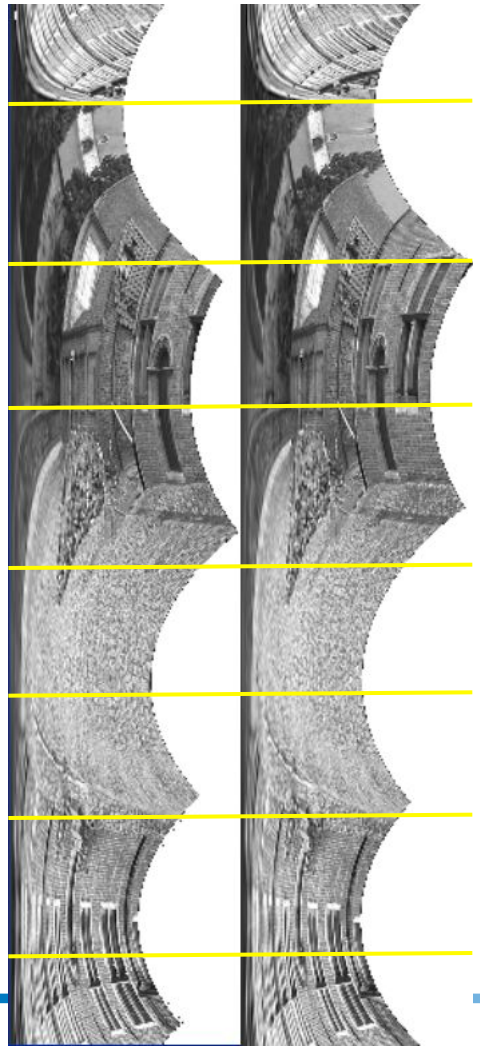




Polar Rectification :: Example



Example :: Béguinage of Leuven



Does not work with standard
Homography-based approaches

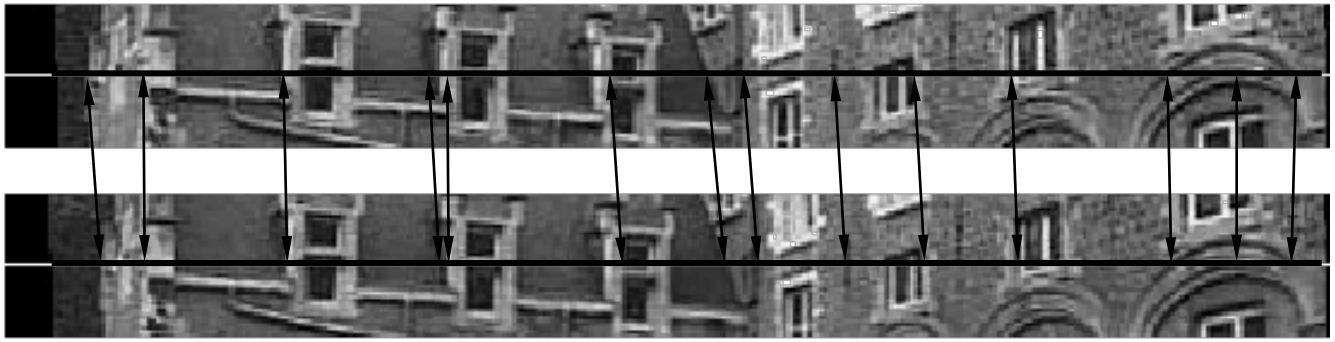
Example :: Béguinage of Leuven

Stereo Matching

- attempt to match every pixel
- use additional constraints

Exploiting Motion and Scene Constraints

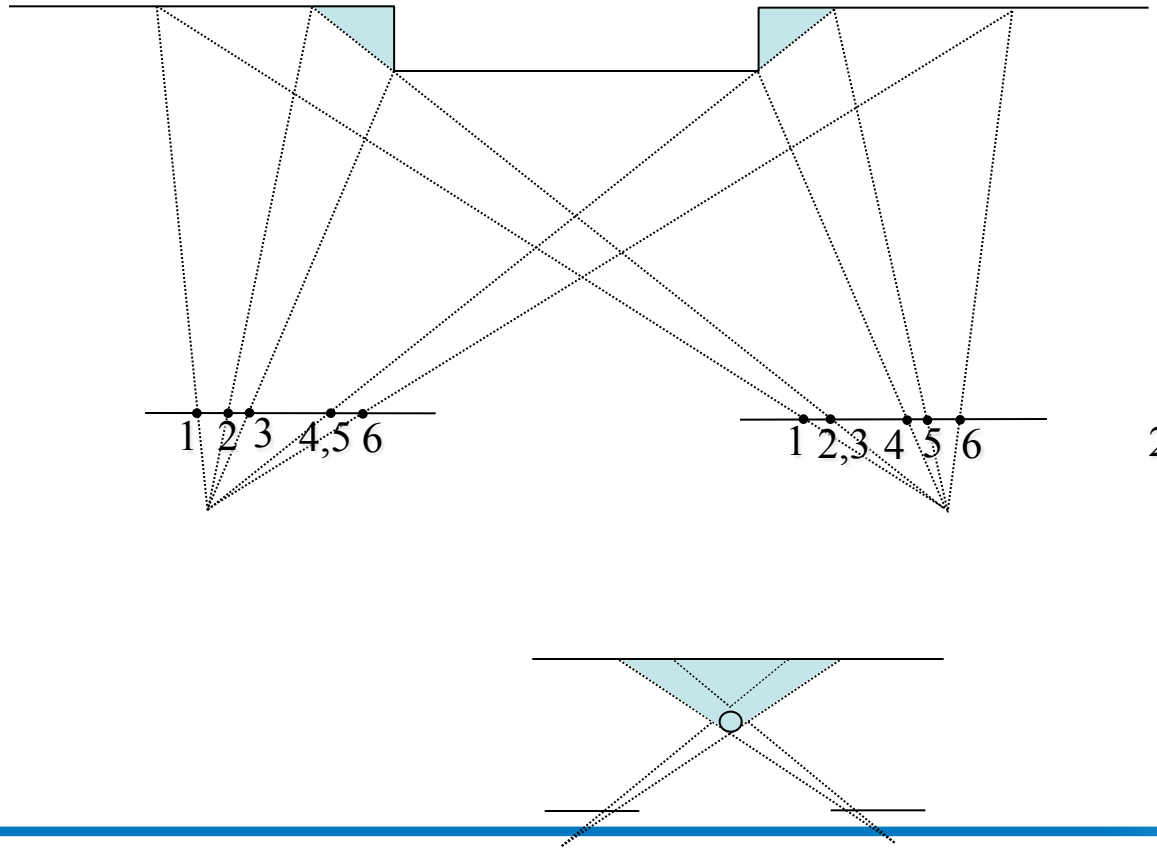
- **Epipolar constraint** (through rectification)



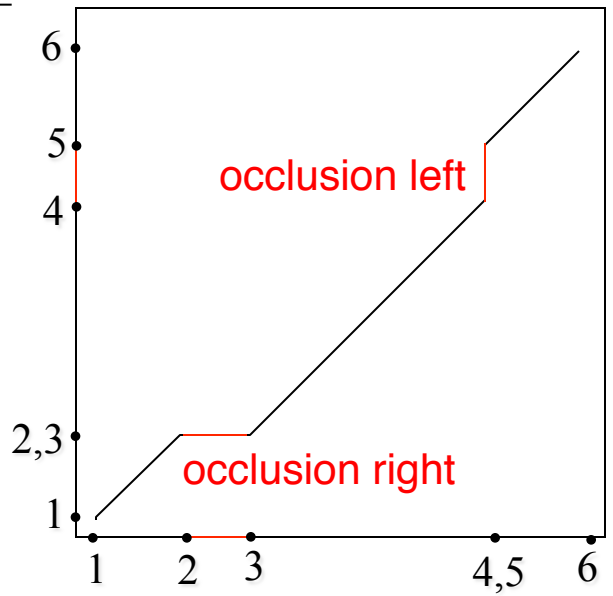
- Ordering constraint
- Uniqueness constraint
- Disparity limit
- Disparity continuity constraint

Ordering Constraint

surface slice



surface as a path

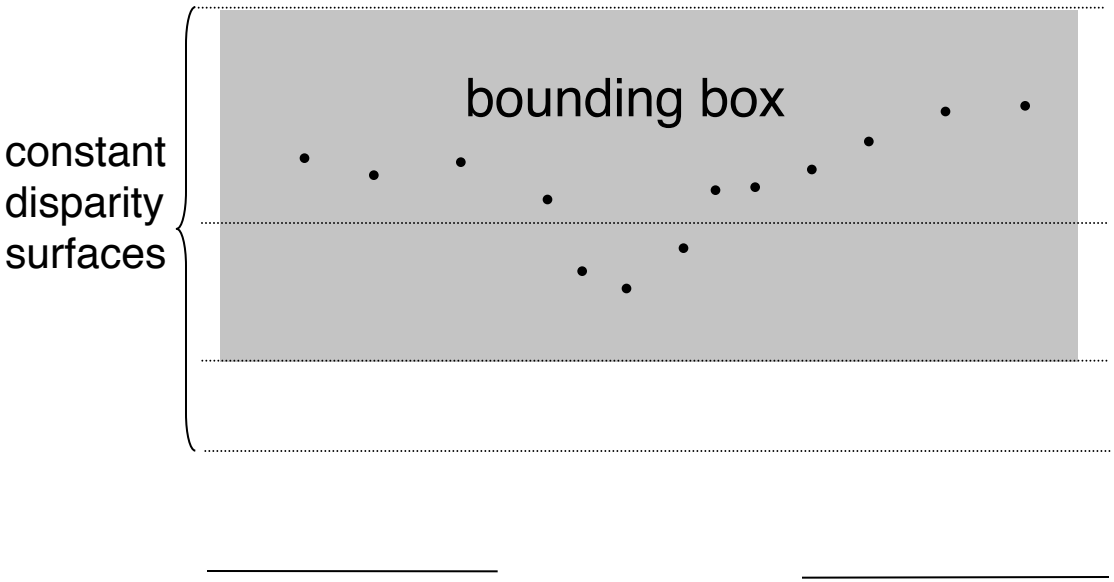


Uniqueness Constraint

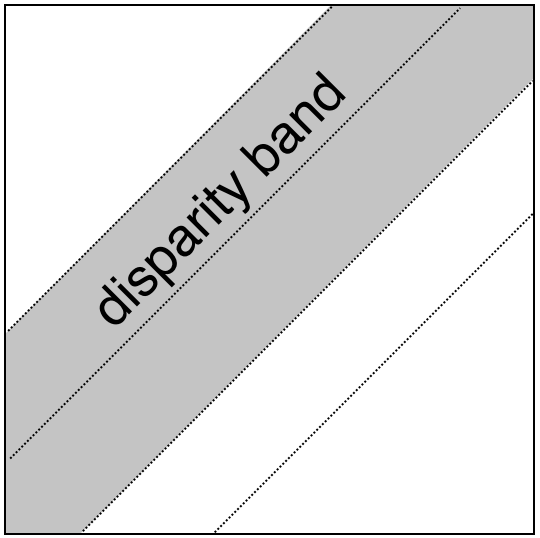
- In an image pair each pixel has at most one corresponding pixel
  - In general one corresponding pixel
  - In case of occlusion there is none

Disparity Constraint

surface slice



surface as a path



use reconsructed features
to determine bounding box

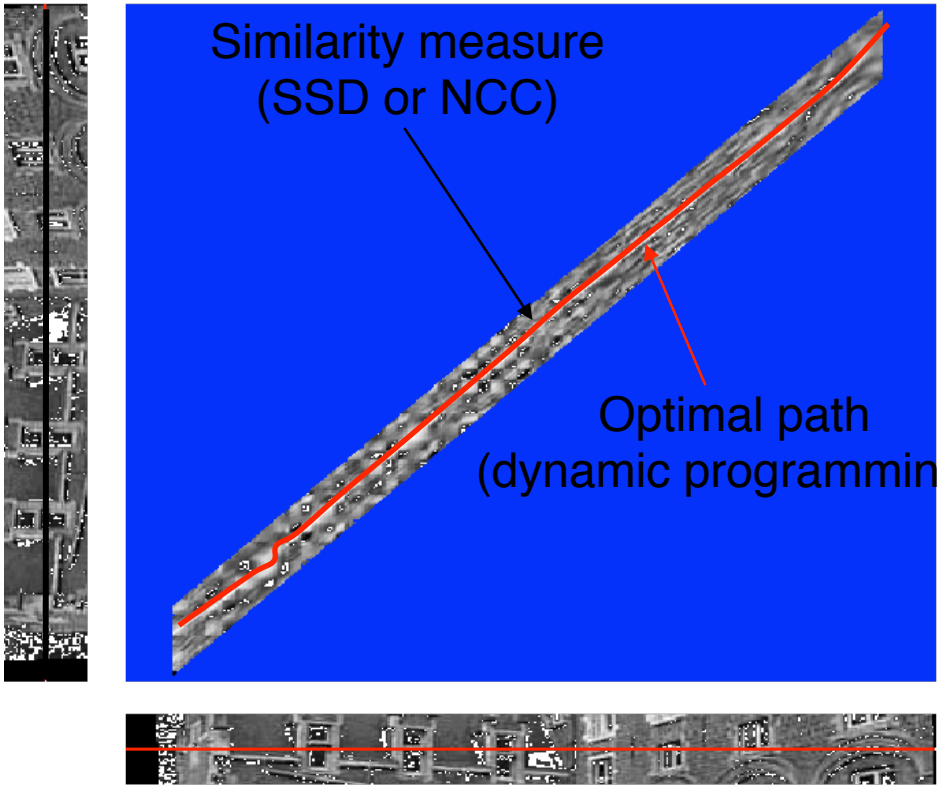
Disparity Continuity Constraint

- Assume piecewise continuous surface

$\Rightarrow$ piecewise continuous disparity

  - In general disparity changes continuously
  - discontinuities at occluding boundaries

Stereo Matching



Constraints
- epipolar
- ordering
- uniqueness
- disparity limit
- disparity gradient limit

Trade-off
- Matching cost (data)
- Discontinuities (prior)

(Cox et al. CVGIP' 96; Koch' 96; Falkenhagen´97;
Van Meerbergen,Vergauwen,Pollefeys,VanGool IJCV '02)

Disparity Map

image I(x,y)

Disparity map D(x,y)

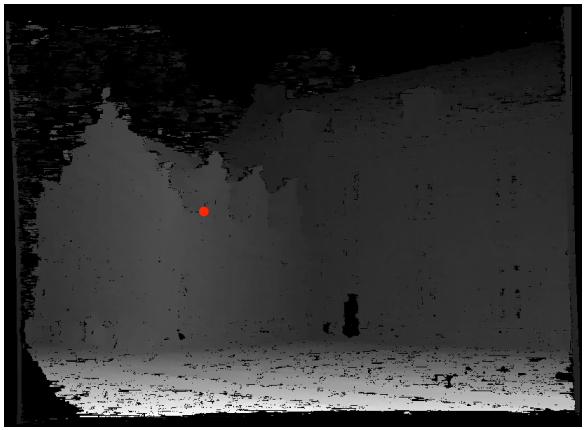


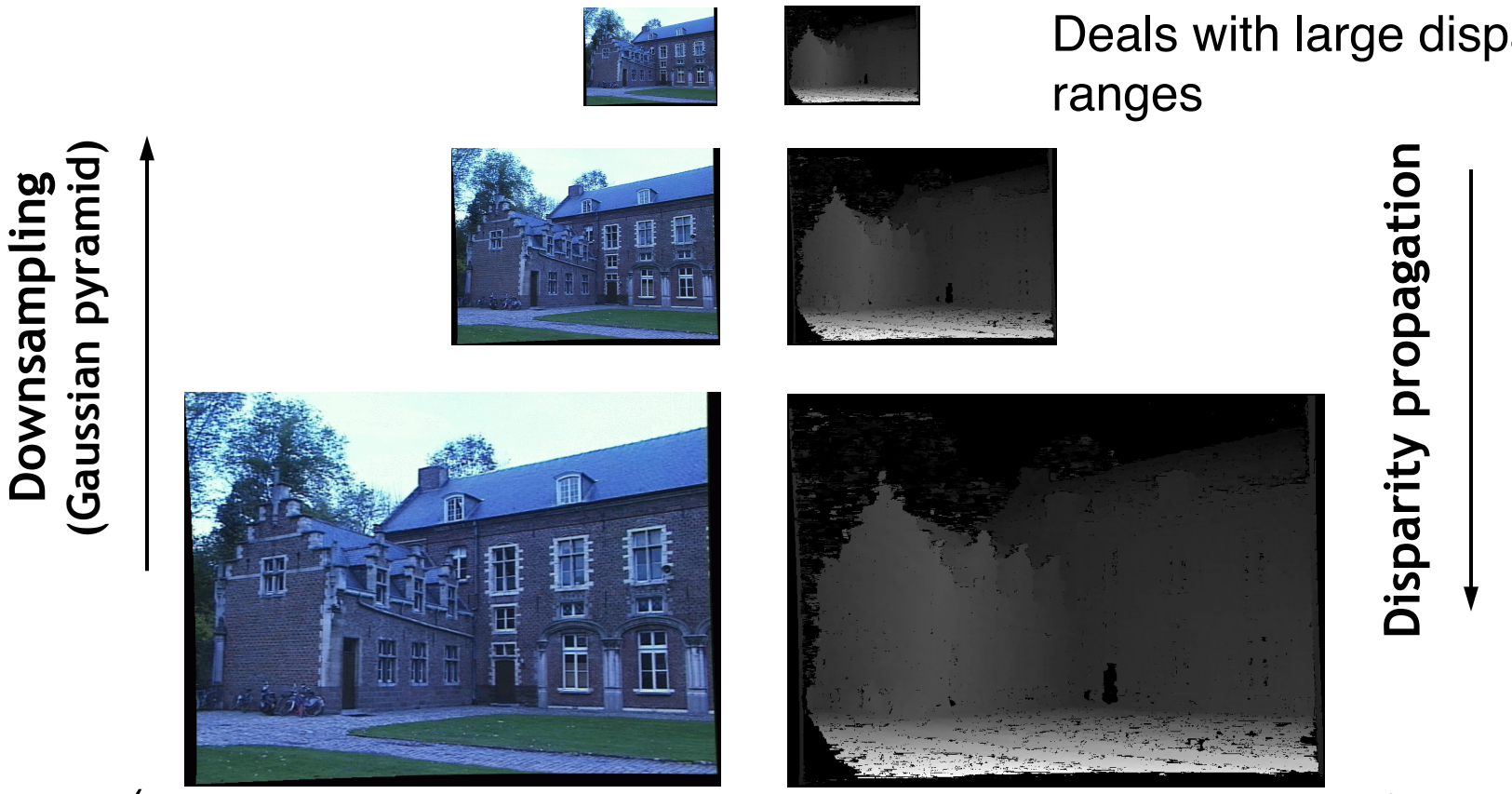
image I´(x´,y´)

$$(x´,y´)=(x+D(x,y),y)$$

Hierarchical Stereo Matching

Allows faster computation

Deals with large disparity ranges



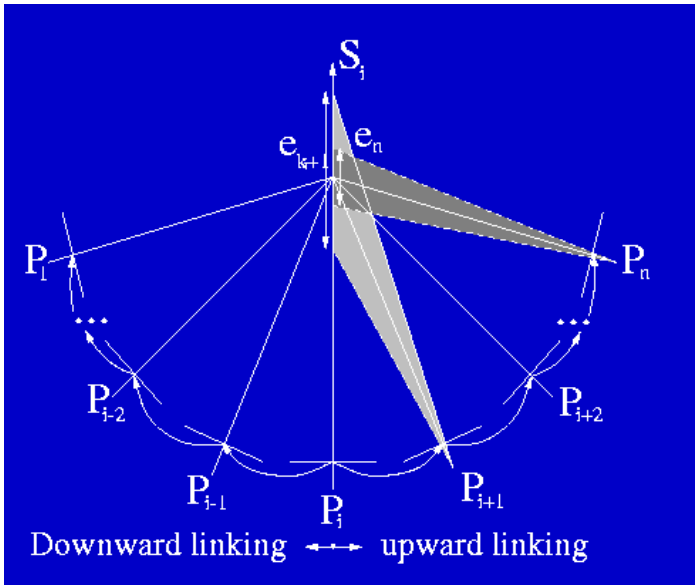
(Falkenhagen´97;Van Meerbergen,Vergauwen,Pollefeys,VanGool IJCV ‘02)

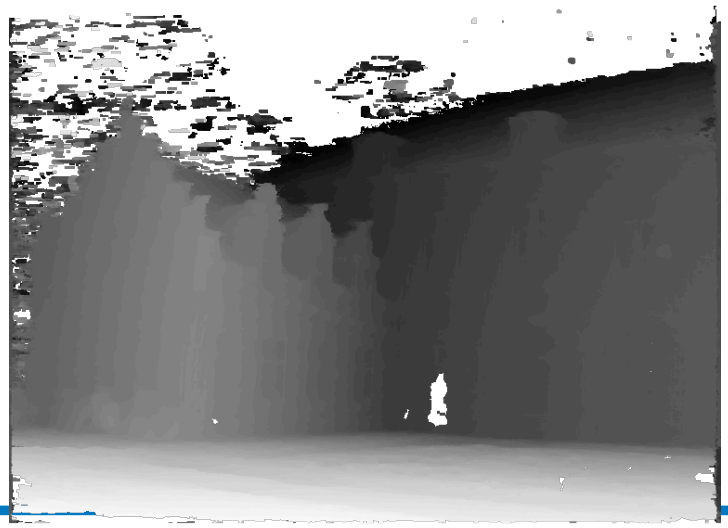
Example: Reconstruct Image from Neighboring Images

Multi-view Depth Fusion

(Koch, Pollefeys and Van Gool. ECCV '98)

- Compute depth for every pixel of reference image
  - Triangulation
  - Use multiple views
  - Up- and down sequence
  - Use Kalman filter





Allows to compute robust texture