Multi View Geometry (Spring '08)

Computing F

Prof. Kyoung Mu Lee
SoEECE, Seoul National University

Basic equation

- Given a correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$
- The basic incidence relation is $\mathbf{x}'^{\mathrm{T}}\mathbf{F}\mathbf{x} = 0$
- It can be written

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

- Or

$$\underbrace{[x'x, x'y, x', y'x, y'y, y', x, y, 1]}_{\text{(data)}}\underbrace{[f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^{\mathrm{T}}}_{\text{(unknowns)}} = 0$$

- For *n* correspondences

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

$$\mathbf{Af} = 0 \qquad \text{(linear)}$$

Basic equation

- Solution is determined up to scale
- Need 8 equations ⟶ 8points
- 8 points ⟶ unique solution
- > 8 points ⟶ least-square solution

- Least-square solution
  - (i) Form equations $\mathbf{Af} = 0$
  - (ii) Take SVD: $\mathbf{A} = \mathbf{UDV}^T$
  - (iii) Solution is the last column of **V** (corresponding to smallest e-value)
  - (iv) Minimizes $\|\mathbf{Af}\|$ subject to $\|\mathbf{f}\| = 1$

The singularity constraint

- Note **F** has rank 2: *det* **F** =0

$$\mathbf{e}'^{\mathrm{T}}\mathbf{F} = 0 \qquad \mathbf{Fe} = 0$$

- SVD from linearly computed **F** matrix (rank 3)

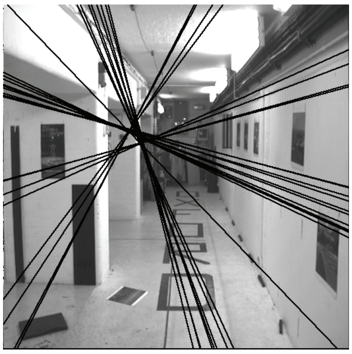
$$\mathbf{F} = \mathbf{U}\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix}\mathbf{V}^{\mathrm{T}} = \mathbf{U}_1\sigma_1\mathbf{V}_1^{\mathrm{T}} + \mathbf{U}_2\sigma_2\mathbf{V}_2^{\mathrm{T}} + \mathbf{U}_3\sigma_3\mathbf{V}_3^{\mathrm{T}}$$

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

$$\mathbf{F}' = \mathbf{U}\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix}\mathbf{V}^{\mathrm{T}} = \mathbf{U}_1\sigma_1\mathbf{V}_1^{\mathrm{T}} + \mathbf{U}_2\sigma_2\mathbf{V}_2^{\mathrm{T}}$$

- Replace **F** by **F'**

The singularity constraint



Uncorrected **F**

Corrected **F**

The minimum case – 7 point correspondences

- **F** has 9 entries but it is defined only up to scale
- Singularity condition *det* ***F****=0* gives a further constraint
- **F** has 3 rows ⟶ *det* ***F****=0* is a cubic constraint
- **F** has only 7 DOF
- Thus, it is possible to solve for **F** from just 7 point correspondences

The minimum case – 7 point correspondences

- Computation of **F** from 7 points correspondences

(i) Form the 7x9 set of equations

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

(ii) System has 2-dimensional solution set

(iii) By using SVD, the general solution is of the form

$$\mathbf{A} = \mathbf{U}_{7x7}\mathrm{diag}(\sigma_1,\ldots,\sigma_7,0,0)\mathbf{V}_{9x9}^{T}$$

$$\Rightarrow \mathbf{A}[\mathbf{v}_8 \mathbf{v}_9] = \mathbf{0}_{9x2} \qquad \mathbf{f} = \mathbf{f}_1 + \lambda \mathbf{f}_2,\ \mathbf{f}_1 = \mathbf{v}_8, \mathbf{f}_2 = \mathbf{v}_9$$

(iv) In matrix term

$$\mathbf{F} = \mathbf{F}_1 + \lambda \mathbf{F}_2$$

The minimum case – 7 point correspondences

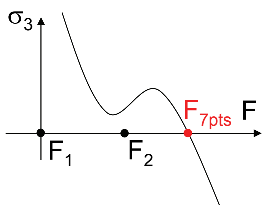
(v) Condition *det* **F** =0 gives cubic variations in λ

$$\det(\mathbf{F}_1 + \lambda \mathbf{F}_2) = a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0$$

$$\det(\mathbf{F}_1 + \lambda \mathbf{F}_2) = \det \mathbf{F}_2 \det(\mathbf{F}_2^{-1}\mathbf{F}_1 + \lambda \mathbf{I}) = 0$$

Compute possible λ as eigenvalues of $\mathbf{F}_2^{-1}\mathbf{F}_1$
(only real solutions are potential solutions)

(vi) Either one of three real solutions



The NOT normalized 8-point algorithm

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

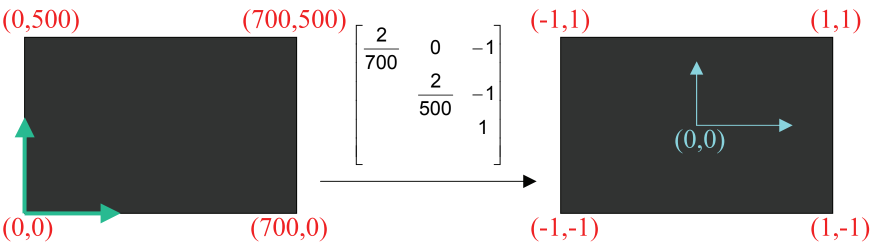
~10000 ~10000 ~100 ~10000 ~10000 ~100 ~100 ~100 1

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

- Two steps:
  - (i) linear solution: Solve **Af =0** to find **F**
  - (ii) Constraint enforcement: Replace **F** by **F'**
- Raw 8-point algorithm performs badly in presence of noise and sensitive to origin and scale

Data normalization

- Data must be translated and scaled to canonical coordinate frame
- Transform image to *~[-1,1]x[-1,1]*



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

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

The normalized 8-point algorithm

- Normalized 8-point algorithm:
  (i) Normalization: Transform the image coordinates:

$$\hat{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i$$
$$\hat{\mathbf{x}}'_i = \mathbf{T}'\mathbf{x}'_i$$

  (ii) Solution: Compute Fundamental matrix from the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$

$$\hat{\mathbf{x}}'^T_i \hat{\mathbf{F}}_i \hat{\mathbf{x}}_i = 0$$

  (iii) Singularity constraint: Find the closest singular $\hat{\mathbf{F}}'$ to $\hat{\mathbf{F}}$
  (iv) Denormalization: $\mathbf{F} = \mathbf{T}'^T \hat{\mathbf{F}}' \mathbf{T}$

Algebraic minimization

- Enforcing the singularity constraint
  - ✓ SVD method minimizes
  - ✓ Simple and rapid
  - ✓ Not optimal
  - ✓ Treats all entries of **F** equally
  - ✓ However, some entries of **F** are more tightly constrained by the data

- It is possible to iteratively minimize algebraic distance subject to *det* **F**=0

Minimizing of Geometric error

- Algebraic error vector $\|\mathbf{Af}\|$ has no clear geometric meaning
- Should be minimizing geometric quantities
- Errors derived from incorrect measurements of match points
- The distances from the epipolar lines can be used

- Gold standard
- Sampson error
- Symmetric epipolar distance

Gold standard (ML) method

- Maximum Likelihood Estimation
  - ✓ Assumes a Gaussian distributed noise
  - ✓ Measured correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$
  - ✓ Estimated correspondences $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$
  - ✓ least-squares for Gaussian noise

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

  - ✓ Simultaneous estimation of **F** and $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$

Gold standard (ML) method

- Minimizing the Gold-Standard error function
  - ✓ Initialize: normalized 8-point, (**P**,**P'**) from **F**, reconstruct $\mathbf{X}_i$

$$\mathbf{P} = [\mathbf{I} \mid \mathbf{0}], \mathbf{P}' = [\mathbf{M} \mid \mathbf{t}], \mathbf{X}_i$$

$$\mathbf{P}' = [\mathbf{M} \mid \mathbf{t}] = [[\mathbf{t}]_\times \mathbf{F} \mid \mathbf{t}] = [[\mathbf{e}']_\times \mathbf{F} \mid \mathbf{e}']$$
$$\mathbf{F} = [\mathbf{t}]_\times \mathbf{M} = [\mathbf{e}']_\times \mathbf{M}$$

  - ✓ Compute $\hat{\mathbf{x}}_i = \mathbf{P}\mathbf{X}_i, \hat{\mathbf{x}}'_i = \mathbf{P}'\mathbf{X}_i$
  - ✓ Parameterize: $\mathbf{P}' = [\mathbf{M} \mid \mathbf{t}]$ and $\mathbf{X}_i$
  - ✓ Minimize the cost function using sparse LM method
  - ✓ Total of *3n+12* parameters
    - 12 for **P'**
    - 3 for each point $\mathbf{X}_i$
  - ✓ Once **P'** is found, compute $\hat{\mathbf{F}} = [\mathbf{t}]_\times \mathbf{M}$

Parameterization of F

- Estimation of **F** may be done by parameter minimization
- Parameterize **F** such that *det* **F** =0
- Various parameterizations can be used

- Overparameterization
  - ✓ Write $\mathbf{F} = [\mathbf{t}]_\times \mathbf{M}$
  - ✓ 3 parameters for **t** and 9 for **M**

- Epipolar parameterization

$$\mathbf{F} = \begin{bmatrix} a & b & \alpha a + \beta b \\ c & d & \alpha c + \beta d \\ e & f & \alpha e + \beta f \end{bmatrix}$$

  - ✓ $(\alpha, \beta, -1)$ is an epipole
  - ✓ To achieve a minimum set of parameters, set one of the elements (say *f*) to 1

Parameterization of F

- Both epipoles as parameters

$$F = \begin{bmatrix} a & b & \alpha a + \beta b \\ c & d & \alpha c + \beta d \\ \alpha' a + \beta' c & \alpha' b + \beta' d & \alpha' \alpha a + \alpha' \beta b + \beta' \alpha c + \beta' \beta d \end{bmatrix}$$

✓ $(\alpha, \beta, -1)$ and $(\alpha', \beta', -1)$ are the two epipoles

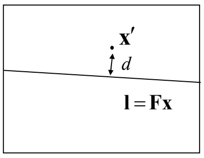
✓ Can set one of *a,b,c,d* to 1

✓ To avoid singularities, switch between different choices of two columns and two rows to use as basis

✓ Total of 36 (4x3x3) parameterizations

Epipolar distance

- Epipolar distance: the distance of point **x**' to epipolar line **Fx**
- Let $\mathbf{Fx} = (\lambda, \mu, \nu)^T$ and $\mathbf{x}' = (x', y', 1)^T$



- Then the distance

$$d(\mathbf{x}', \mathbf{Fx}) = \mathbf{x}'^T \mathbf{Fx} (\lambda^2 + \mu^2)^{-1/2}$$
$$= \frac{\mathbf{x}'^T \mathbf{Fx}}{\left((\mathbf{Fx})_1^2 + (\mathbf{Fx})_2^2\right)^{1/2}}$$

- Total cost function:

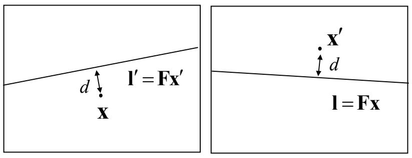
$$\sum_i d(\mathbf{x}'_i, \mathbf{Fx}_i)^2 = \sum_i \frac{\left(\mathbf{x}_i'^T \mathbf{Fx}_i\right)^2}{(\mathbf{Fx}_i)_1^2 + (\mathbf{Fx}_i)_2^2}$$

- Minimize the cost function over parameterization of **F**

Symmetric epipolar distance error

- Sum of epipolar distances in both images
- Symmetric cost function:

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



- Total cost function:

$$\sum_i d(\mathbf{x}'_i, \mathbf{F}\mathbf{x}_i)^2 + d(\mathbf{x}_i, \mathbf{F}^T\mathbf{x}'_i)^2$$
$$= \sum_i (\mathbf{x}'^T \mathbf{F}\mathbf{x})^2 \left(\frac{1}{(\mathbf{x}'^T\mathbf{F})_1^2 + (\mathbf{x}'^T\mathbf{F})_2^2} + \frac{1}{(\mathbf{F}\mathbf{x})_1^2 + (\mathbf{F}\mathbf{x})_2^2} \right)$$

- Problems:
  - ✓ Points near the epipole have a disproportionate influence
  - ✓ Small deviation in point makes big difference to epipole line

Experimental evaluation of the algorithms

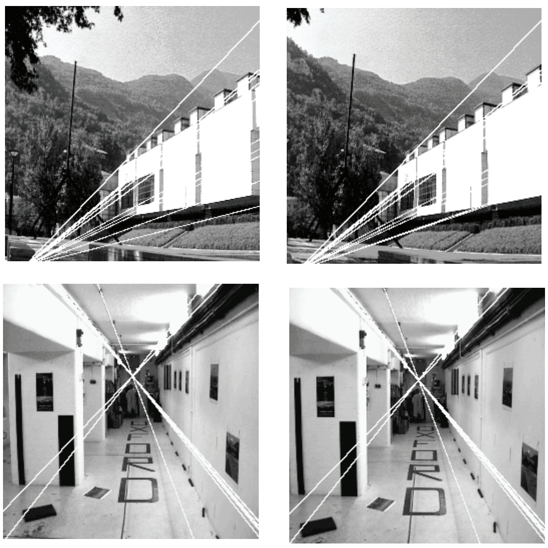
- Three algorithms:
  - ✓ Normalized 8-point algorithm
  - ✓ Minimization of algebraic error with singularity constraint
  - ✓ Gold Standard algorithm

- Experimental procedure
  - ✓ Find matched points (*N*) in image pair
  - ✓ Select *n* matched point at random
  - ✓ Compute **F**
  - ✓ Compute epipolar distance for all points
  - ✓ Repeat 100 times for each *n* and collect statistics

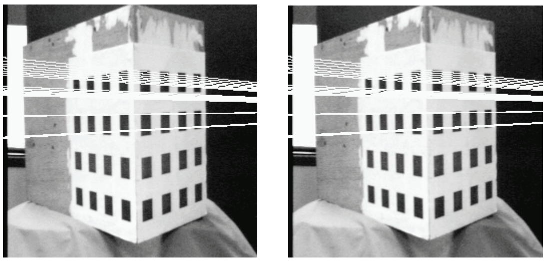
- Residual Error: $\frac{1}{N}\sum_i^N \left(d(\mathbf{x}'_i, \mathbf{F}\mathbf{x}_i)^2 + d(\mathbf{x}_i, \mathbf{F}^T\mathbf{x}'_i)^2 \right)$

Some experiments:

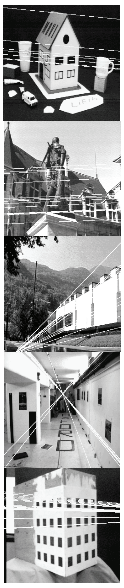
Some experiments:

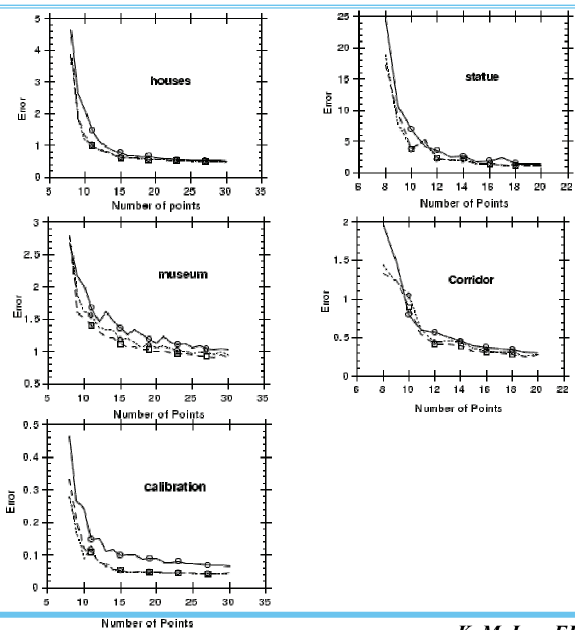


Some experiments:



Some experiments:





Recommendations:

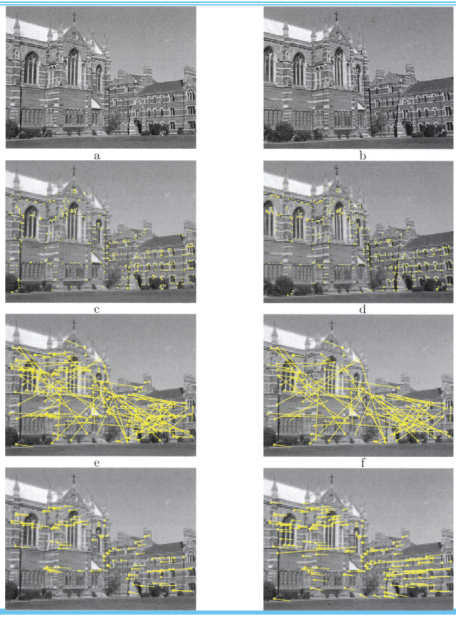
- Do not use unnormalized algorithms
- Quick and easy to implement: 8-point normalized algorithm
- Better: enforce rank-2 constraint during minimization
- Best: Gold Standard algorithm (ML)
  (minimal parameterization, sparse implementation)

Automatic computation of F

- Algorithm based on RANSAC

(i) Find **interest points** in each image
(ii) Determine **putative correspondences** (using cross-correlation & proximity)
(iii) RANSAC:
  Repeat
  (a) Select random sample of 7/8 correspondences
  (b) Compute F
  (c) Measure support (# of inliers)
(iv) Re-estimation of **F** from inliers using MLE
(v) Guided matching: generate additional matches
  (repeat (iv) and (v) until stable)

Automatic computation of F



Interest points

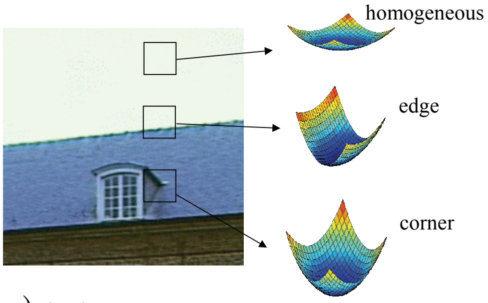
- How to find good interest points?
- Required properties:
  - ✓ Uniqueness/Dissimilarity to neighboring points
  - ✓ Stability across views (view invariant)

- Harris corner point (Harris & Stephens, AVC 88)
- SUSAN (Smallest Univalue Segment Assimilating Nucleus) (www.fmrib.ox.ac.uk/~steve/susan)
- CSS (Curvature Scale Space) (Mokhtarian, PAMI 98)

Feature points

- Harris corner points
  - ✓ Dissimilarity to neighbors

$$D(\Delta x, \Delta y) = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \mathbf{M} [\Delta x \quad \Delta y]$$



$$\mathbf{M} = \iint_W \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} w(x, y) dxdy$$

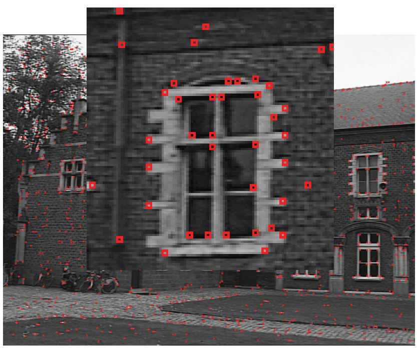
$\mathbf{M}$ should have large eigenvalues

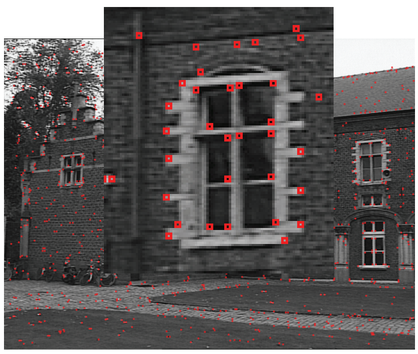
  - ✓ Corner response function

$$R = \det \mathbf{M} - k(\operatorname{trace} \mathbf{M})^2$$

Feature points

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





Putative correspondence points

- Window-based correlation matching
- Use SSD (Sum of Squared Difference) or NCC (Normalized Cross Correlation) with search area

$$S(d_x, d_y) \in \left[x - \tfrac{w}{10}, x + \tfrac{w}{10}\right] \times \left[y - \tfrac{h}{10}, y + \tfrac{h}{10}\right]$$

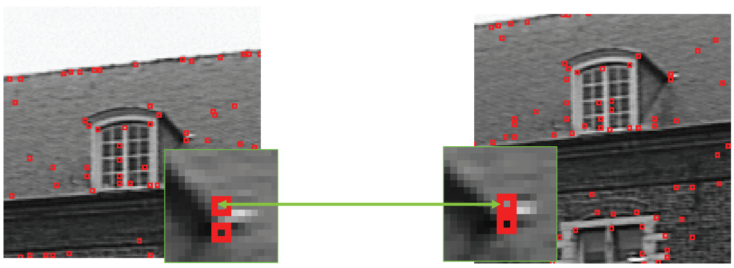
$$SSD = \iint_W \left[I_1(x,y) - I_2(x',y')\right]^2 \omega(x,y)\,dxdy$$

$$NCC = \frac{\iint_W \left(I_1(x,y) - \bar{I}_1\right)\left(I_2(x',y') - \bar{I}_2\right)\omega(x,y)\,dxdy}{\sqrt{\iint_W \left(I_1(x,y) - \bar{I}_1\right)^2 \omega(x,y)\,dxdy}\sqrt{\iint_W \left(I_2(x',y') - \bar{I}_2\right)^2 \omega(x,y)\,dxdy}}$$

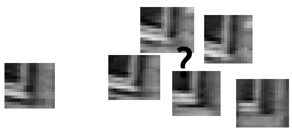
where

$$\bar{I}_1 = \iint_W I_1(x,y)\,dxdy, \quad \bar{I}_2 = \iint_W I_2(x',y')\,dxdy, \quad (x',y') = (x + d_x, y + d_y)$$

Putative correspondence points

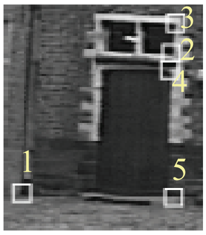


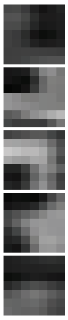
- Keep mutual best matches
- Still many wrong matches (10-50%), but enough to estimate **F**



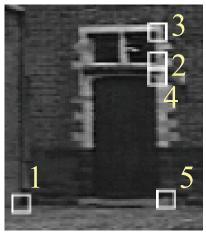
Feature example

- Gives satisfying results for small image motions





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**

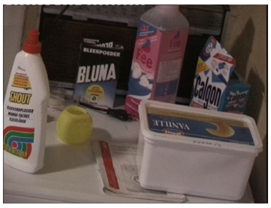


Correspondences consistent with epipolar lines

- Use RANSAC
- Obtain correspondences
- Guided matching by epipolar line
- Typically, final number of matches is about 200-250, with distance error of ~0.2 pixels

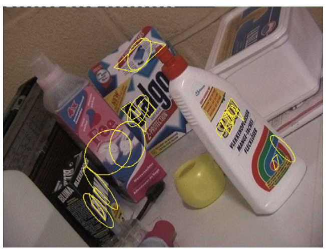
Wide-baseline matching

- Harris corner with window correlation-based matching does not working in wide view variation case
  - ✓ Translation, rotation, scaling } geometric transformations
  - ✓ Foreshortening
  - ✓ Non-diffuse reflections } photometric changes
  - ✓ Illumination
- Need robust feature matching scheme



Wide-baseline matching

- Similarity invariant matching:
  - ✓ Lowe (SIFT) ICCV 99
  - ✓ Schmid & Mohr PAMI 97
- Affine invariant matching:
  - ✓ Mikolajczyk & Schmid, ECCV 02, CVPR 03
  - ✓ Tuytelaars & Van Gool, IJCV 01
  - ✓ Schaffalitzky & Ziserman ECCV 02
  - ✓ J. Matas et al. BMVC 02

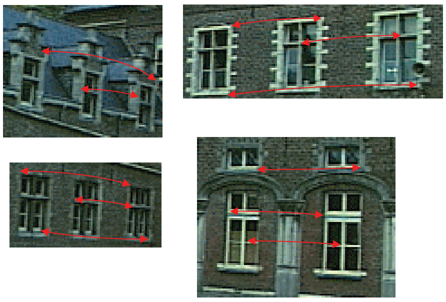


Finding more matches

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

Problems:

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



Two-view geometry

- The recovered Fundamental **F** matrix encodes the full geometric relations between two views

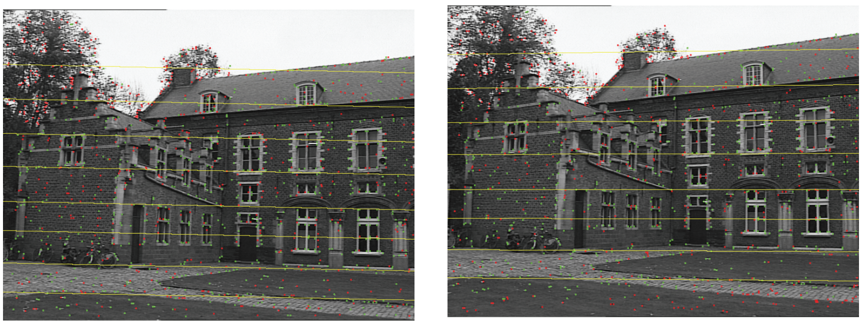
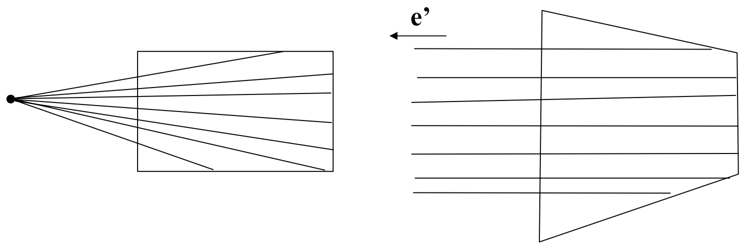


Image rectification

- Make epipolar lines in two images are horizontal (standard stereo setup)
- Which projective transformation?



- Consider the right image
  - ✓ map epipoles **e**' to infinity (-1,0,0) (1,0,0)
  - ✓ while minimizing image distortion
  - ✓ problem when epipole in (or close to) the image

Image rectification

- Steps for rectification of the right image

$$\mathbf{H}' = \mathbf{G}'\mathbf{R}'\mathbf{T}'$$

- **T'**: translation a point of interest $\mathbf{x}'_0$ to the origin and **e'** to $\hat{\mathbf{e}}'$
- **R'**: rotation taking $\hat{\mathbf{e}}'$ to *(-f,0,1)^T* *(f,0,1)^T*
- **G'**: mapping taking *(-f,0,1)^T* to infinity *(-1,0,0)^T* *(1,0,0)^T*

Image rectification

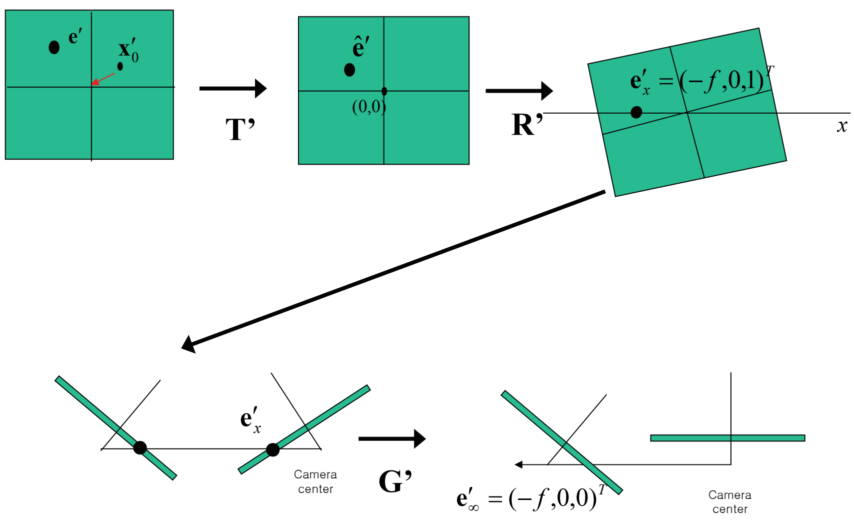


Image rectification

- How to map $\mathbf{e}'_x = (-f, 0, 1)^T$ to infinity point $\mathbf{e}'_\infty = (-f, 0, 0)^T$?
- Let the transformation matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/f & 0 & 1 \end{bmatrix}$$

$$(f, 0, 1)^T \rightarrow (f, 0, 0)^T$$

- Then

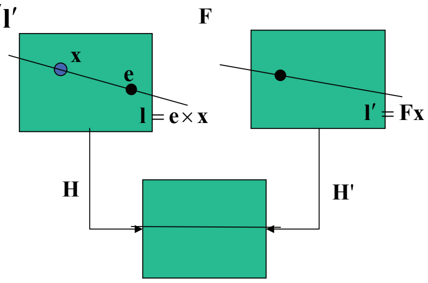
$$\mathbf{G}\mathbf{e}'_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/f & 0 & 1 \end{bmatrix} \begin{bmatrix} -f \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -f \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{G}\mathbf{e}'_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix} \begin{bmatrix} f \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix}$$

- Note for $|x/f| < 1$

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ 1 \end{bmatrix} = \mathbf{G} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 + x/f \end{bmatrix} = \begin{bmatrix} x(1 - x/f + \cdots) \\ y(1 - x/f + \cdots) \\ 1 \end{bmatrix}, \; J = \frac{\partial(\hat{x}, \hat{y})}{\partial(x, y)} \approx \begin{bmatrix} 1 - 2x/f & 0 \\ -y/f & 1 - x/f \end{bmatrix}$$

Image rectification

- How to rectify the left image to match up the epipolar lines?
- Match constraint: $\mathbf{H}^{-T}\mathbf{l} = \mathbf{H}'^{-T}\mathbf{l}'$
- The matched transform:
  - ✓ Given $\mathbf{F} = [\mathbf{e}']_\times \mathbf{M}$



$$\mathbf{H}^{-T}(\mathbf{e} \times \mathbf{x}) = \mathbf{H}'^{-T}(\mathbf{F}\mathbf{x}), \; \forall \mathbf{x}$$

$$\Rightarrow [\mathbf{H}\mathbf{e}]_\times \mathbf{H} = [\mathbf{H}'\mathbf{e}']_\times \mathbf{H}'\mathbf{M}$$

$$\mathbf{H} = (\mathbf{I} + \mathbf{H}'\mathbf{e}'\mathbf{a}^T)\mathbf{H}'\mathbf{M} = \mathbf{H}_A \mathbf{H}_0$$

where $\mathbf{H}_A = \mathbf{I} + \mathbf{H}'\mathbf{e}'\mathbf{a}^T = \mathbf{I} + (-1, 0, 0)^T \mathbf{a}^T = \begin{bmatrix} -a & -b & -c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\mathbf{H}_0 = \mathbf{H}'\mathbf{M}$

$$\begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Image rectification

- Which $\mathbf{H}_A$?
  - ✓ Find $\mathbf{H}_A$ that minimizes the disparity

$$\min \sum_i d\left(\mathbf{H}_A \mathbf{H}_0 \mathbf{x}_i, \mathbf{H}'\mathbf{x}'_i\right)^2$$

$$= \min \sum_i d\left(\mathbf{H}_A \hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i\right)^2$$

$$= \min \sum_i \left(a\hat{x}_i + b\hat{y}_i + c + \hat{x}'_i\right)^2 \qquad \min \sum_i \left(a\hat{x}_i + b\hat{y}_i + c - \hat{x}'_i\right)^2$$

Rectification algorithm outline

- Rectification Algorithm:

(i) Find correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$

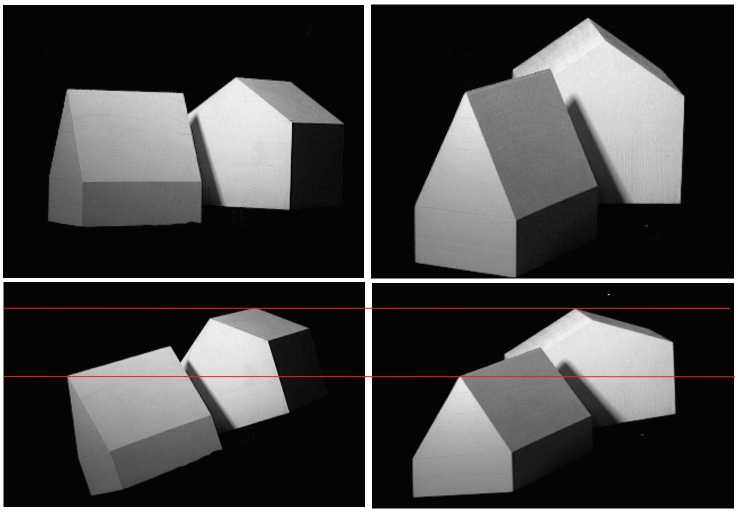
(ii) Compute **F** and **e**, **e'**

(iii) Select **H'** that transforms e' to $(\pm 1,0,0)^T$ for the second image

(iv) Find the matching transform **H** for the first image that minimizes the disparity

(v) Resample the first and second images according to **H** and **H'**, respectively

Rectification example



Rectification example