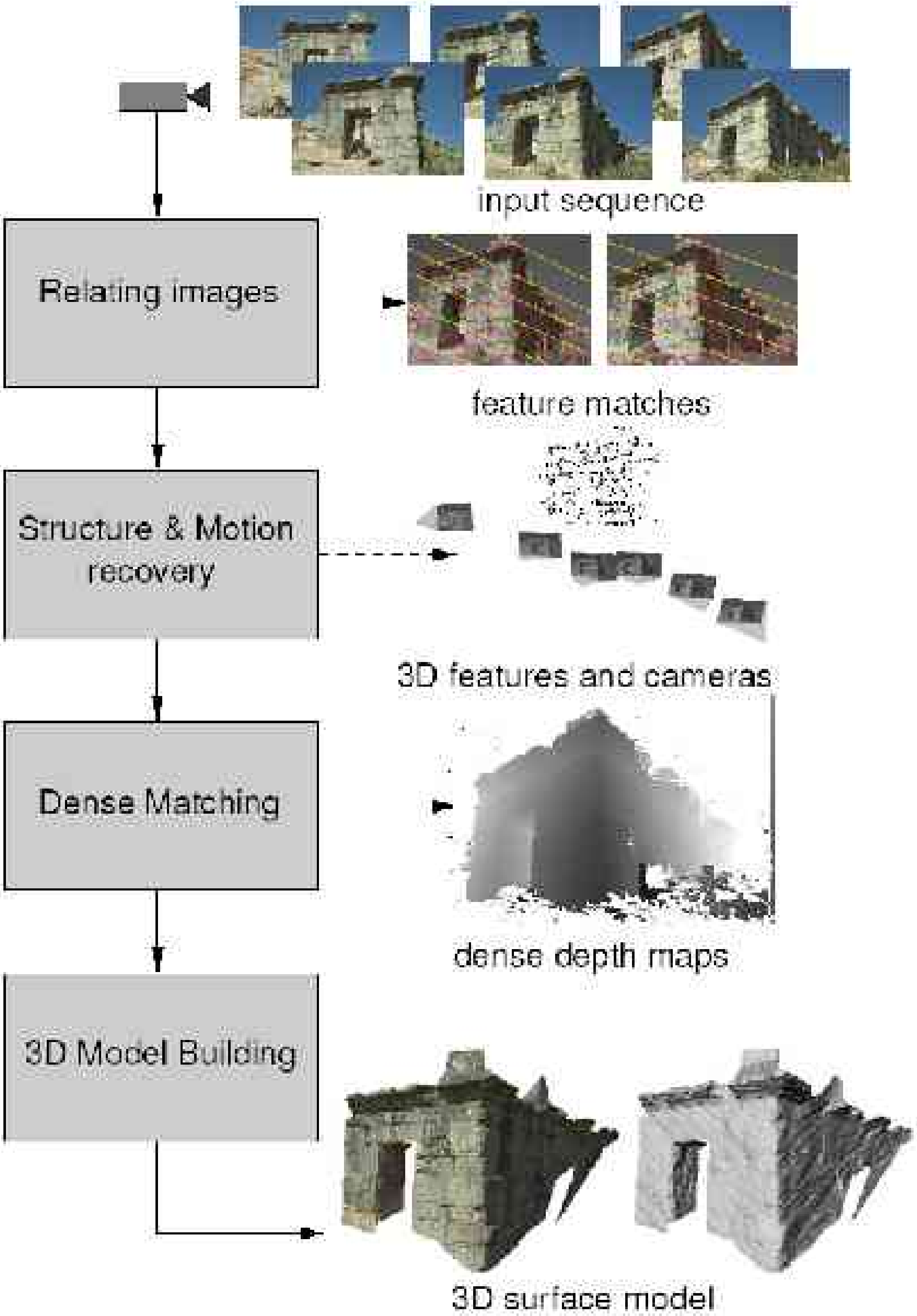
Automatic Creation of 3D Models From Uncalibrated Image Sequences

Jason Repko
University of North Carolina at Chapel Hill

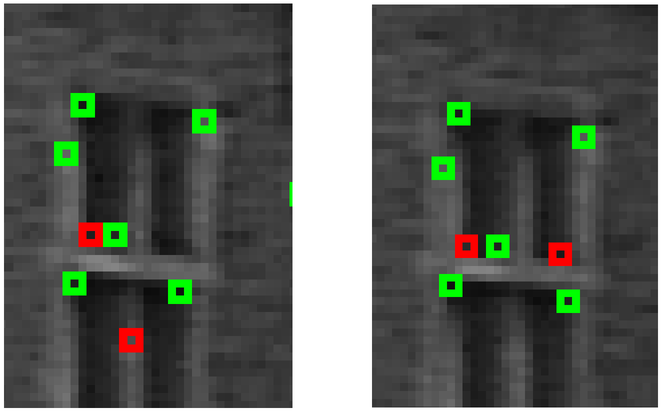
Slides used in this presentation taken from the course notes for 3D Photography taught by Marc Pollefeys in the Fall '05

(Pollefeys et al. '98)



Feature matching vs. tracking

Image-to-image correspondences are key to passive triangulation-based 3D reconstruction



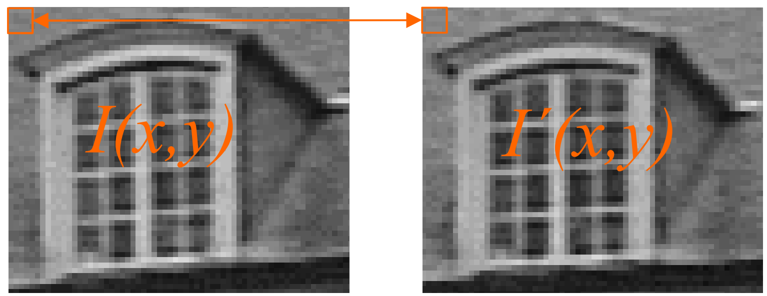
Extract features independently and then match by comparing descriptors

Extract features in first images and then try to find same feature back in next view

What is a good feature?

Comparing image regions

Compare intensities pixel-by-pixel



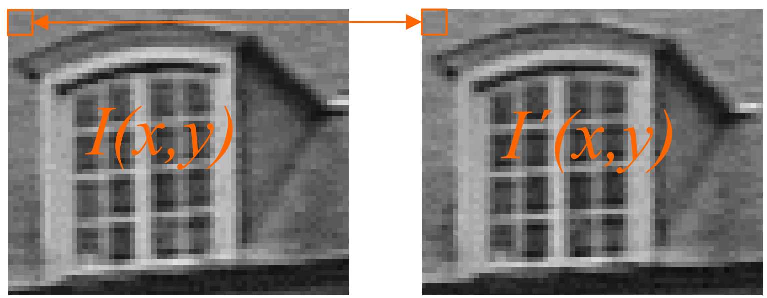
Dissimilarity measures

Sum of Square Differences

$$SSD = \iint_W \left[I'(x,y) - I(x,y) \right]^2 dx\,dy$$

Comparing image regions

Compare intensities pixel-by-pixel



Similarity measures

Zero-mean Normalized Cross Correlation

$$NCC = \frac{N(I', I)}{\sqrt{N(I', I')N(I, I)}}$$

$$N(A, B) = \iint_W \left(A(x,y) - \overline{A}\right)\left(B(x,y) - \overline{B}\right) dxdy$$

Feature points

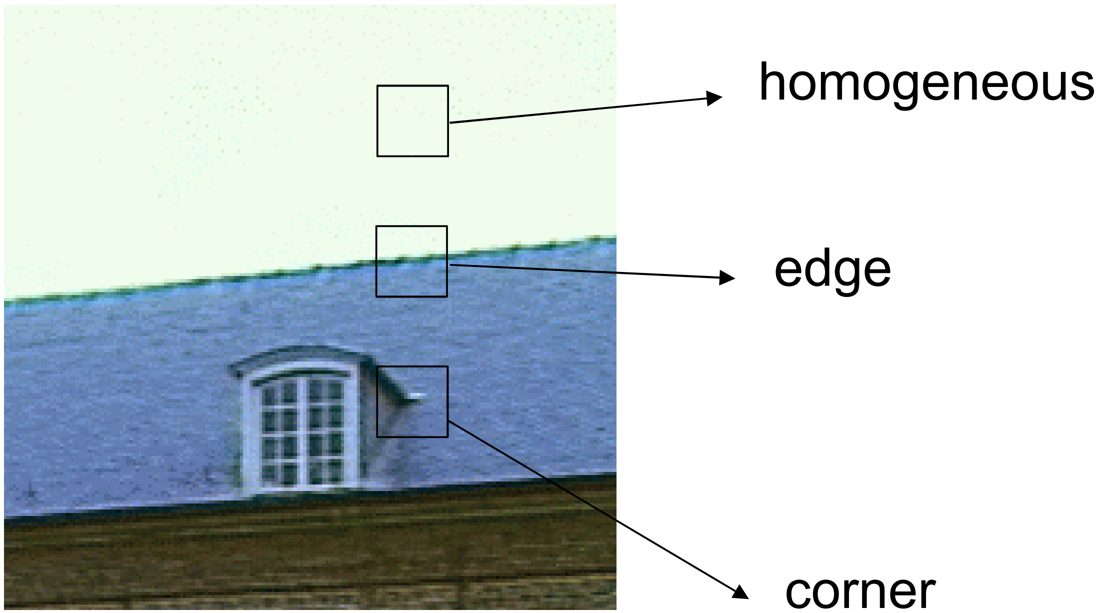
- Required properties:
  - Well-defined

    (i.e. neigboring points should all be different)
  - Stable across views

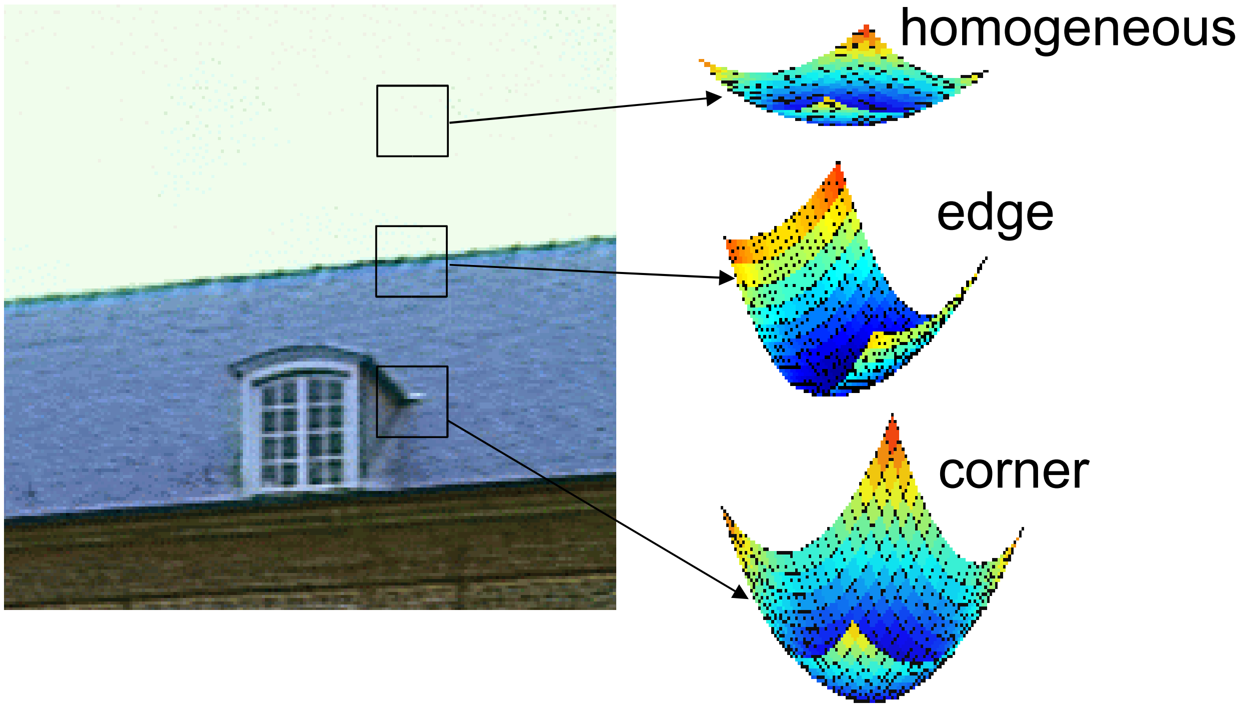
    (i.e. same 3D point should be extracted as feature for neighboring viewpoints)

Feature point extraction

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

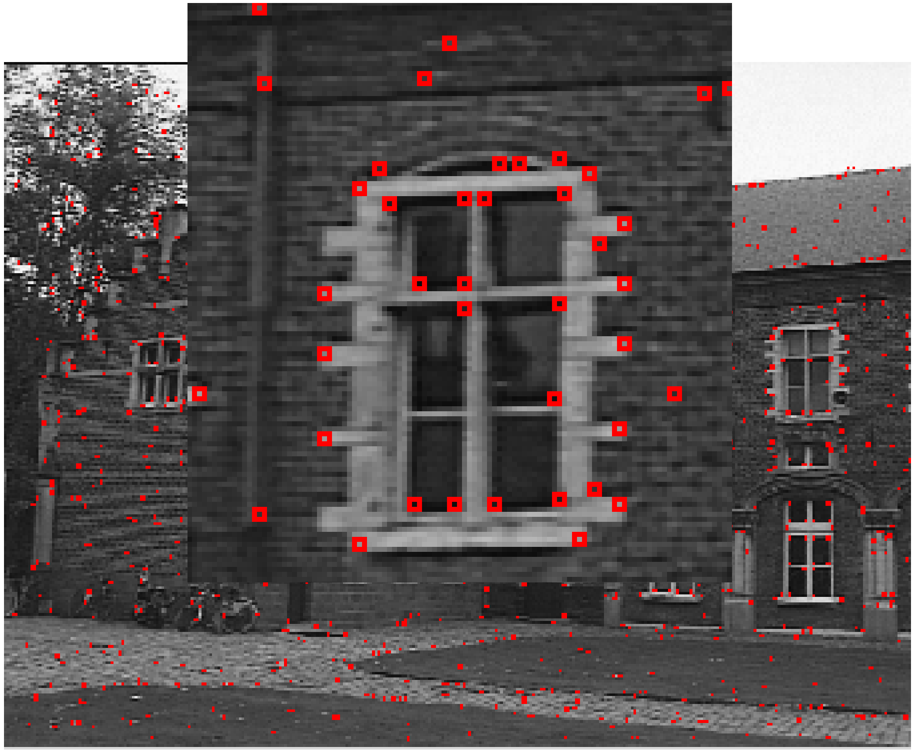


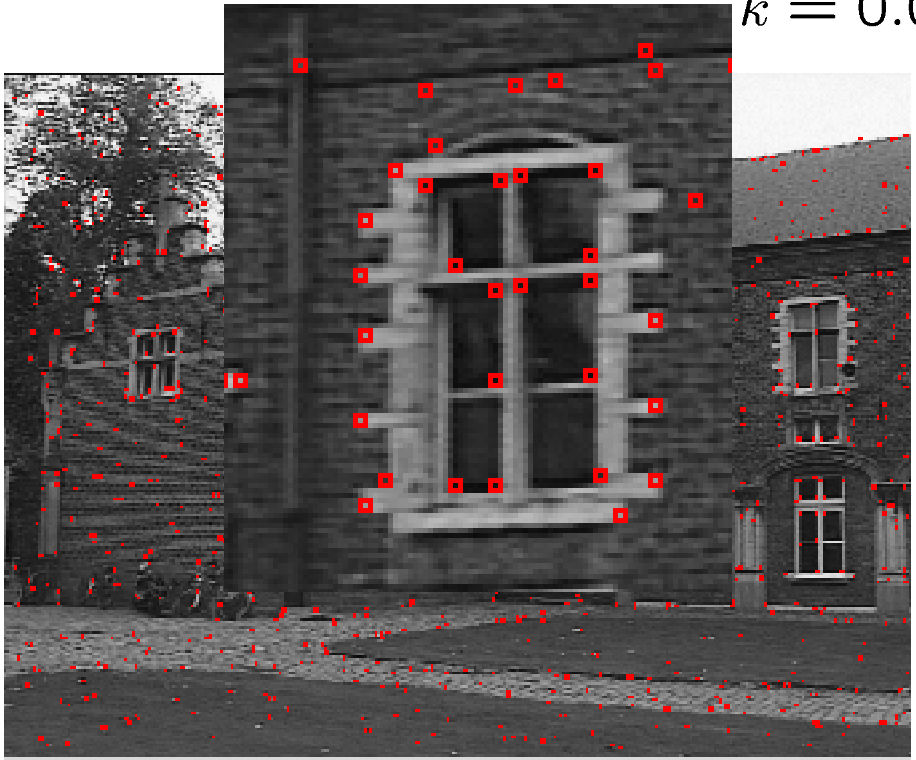
Feature point extraction



Harris corner detector

- Use small local window: $w(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$ $\sigma = 0.7$
- Maximize „cornerness“: $R = \det \mathrm{M} - k\,(\mathrm{trace}\mathrm{M})^2$ $k = 0.04$

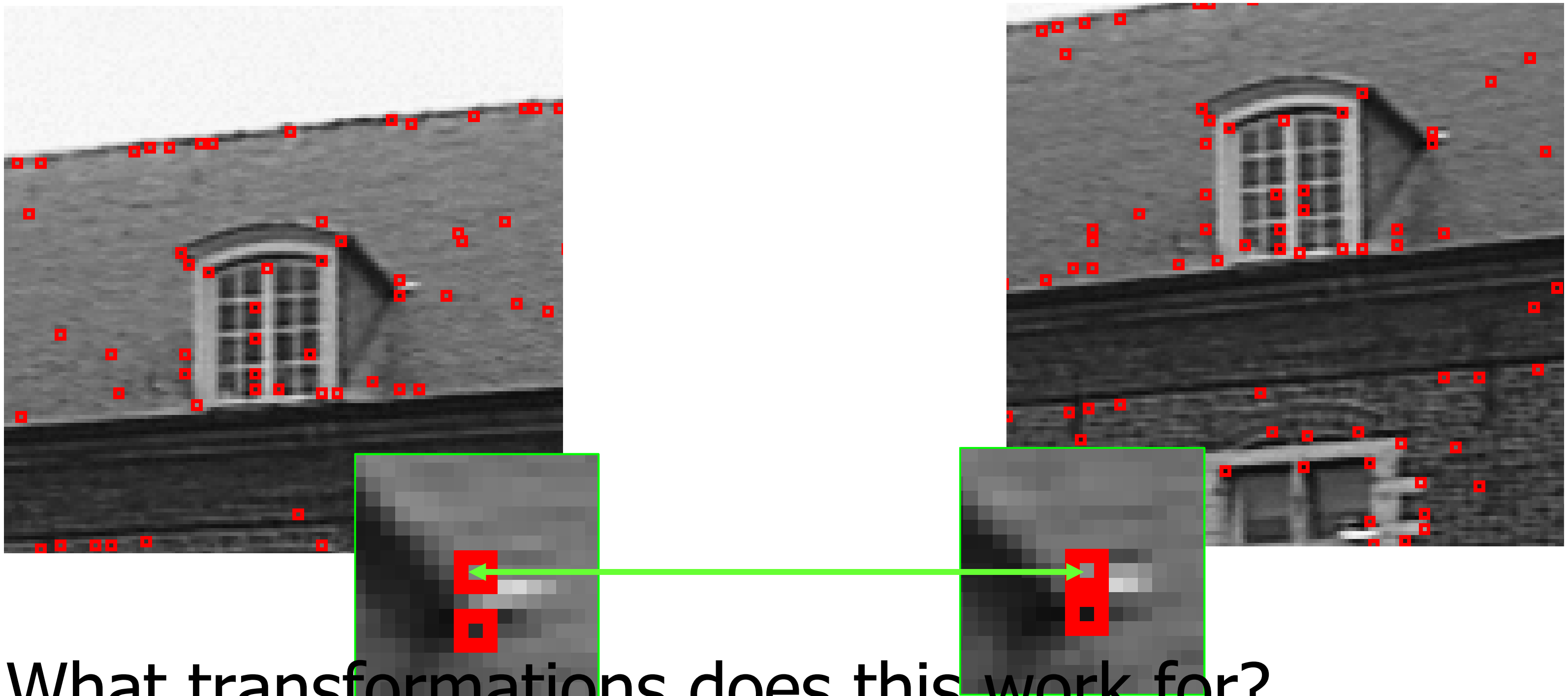




- Only use local maxima, subpixel accuracy through second order surface fitting
- Select strongest features over whole image and over each tile (e.g. 1000/image, 2/tile)

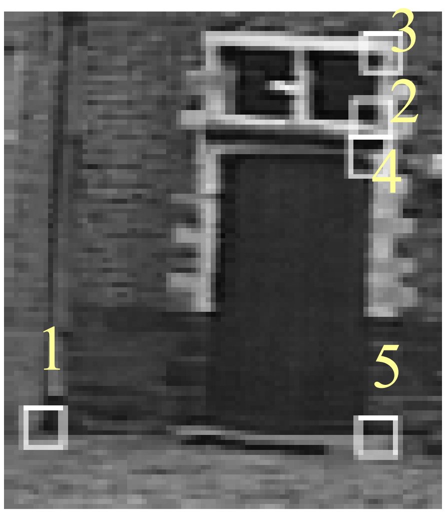
Simple matching

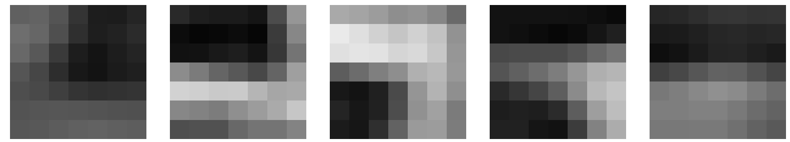
- for each corner in image 1 find the corner in image 2 that is most similar (using SSD or NCC) and vice-versa
- Only compare geometrically compatible points
- Keep mutual best matches

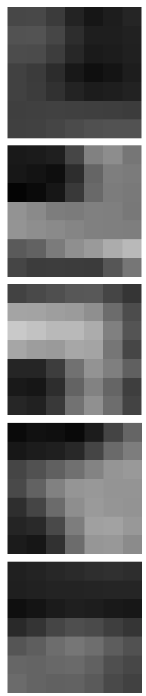


What transformations does this work for?

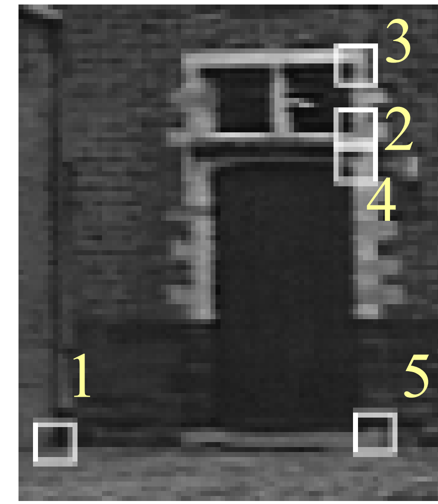
Feature matching: example







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



What transformations does this work for?

What level of transformation do we need?

Wide baseline matching

- Requirement to cope with larger variations between images
  - Translation, rotation, scaling
  - Foreshortening
  - Non-diffuse reflections
  - Illumination

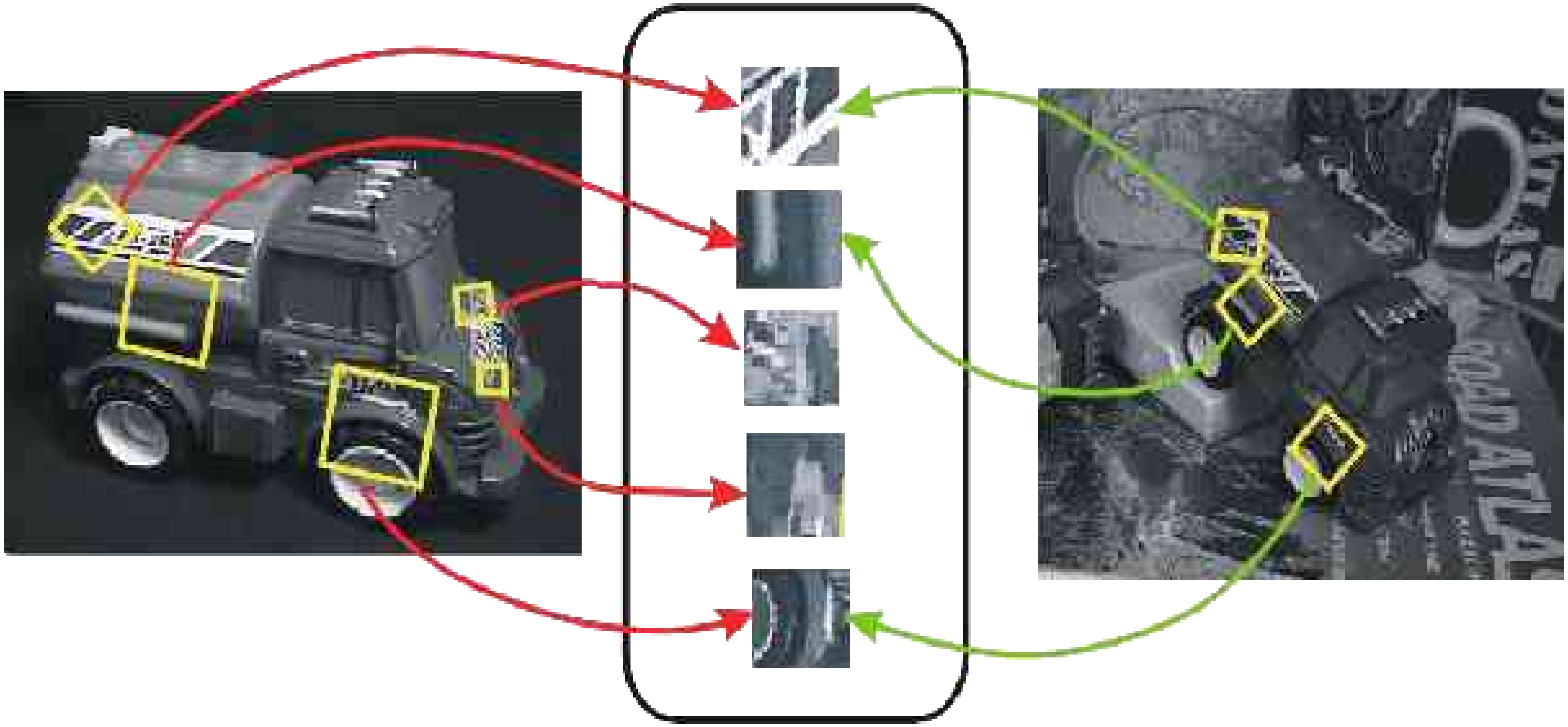
geometric transformations (Translation, rotation, scaling; Foreshortening)

photometric changes (Non-diffuse reflections; Illumination)

Lowe's SIFT features

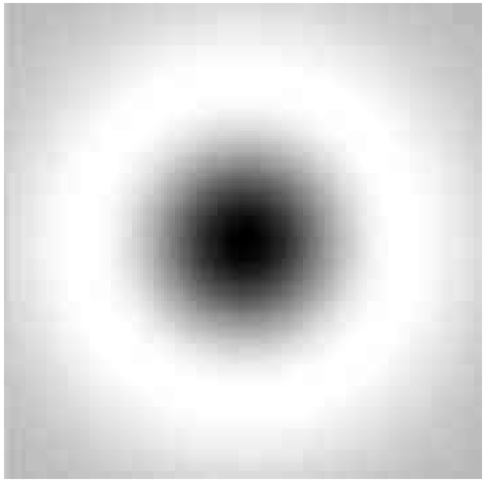
(Lowe, ICCV99)

Recover features with position, orientation and scale



Position

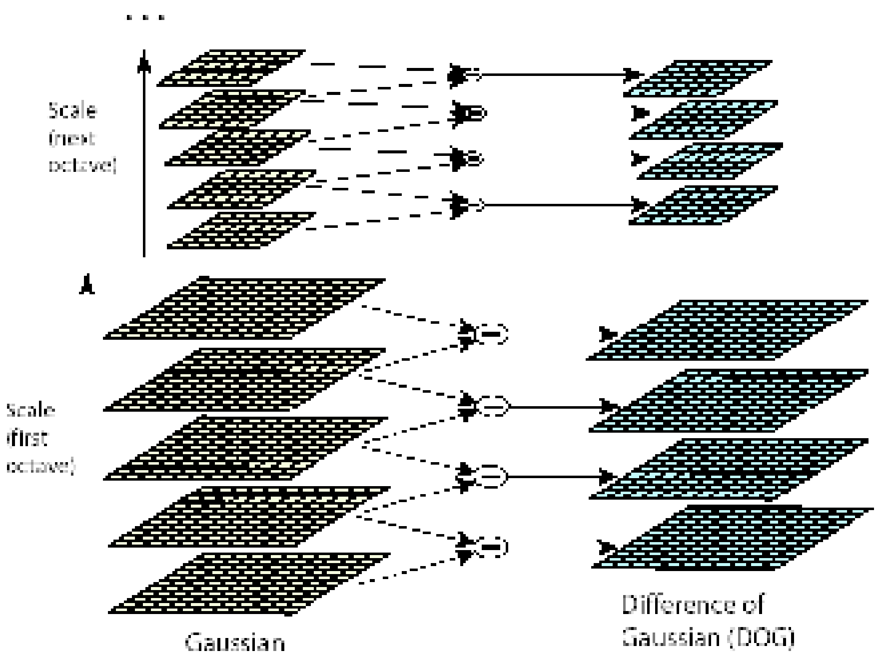
- Look for strong responses of DOG filter (Difference-Of-Gaussian)
- Only consider local maxima

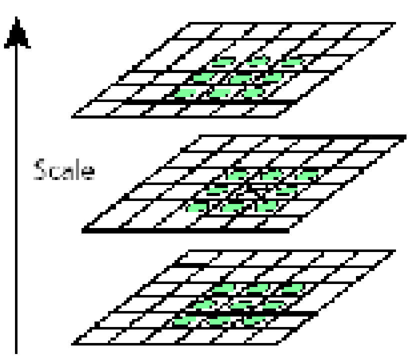


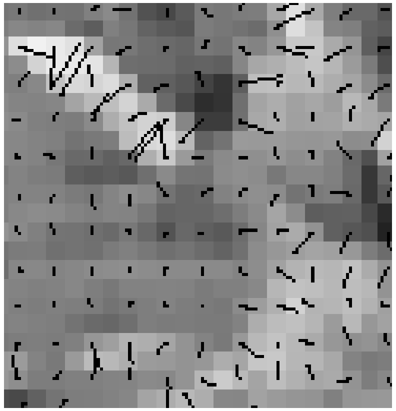
$$\mathsf{DOG}(x,y) = \frac{1}{k} e^{-\frac{x^2+y^2}{(k\sigma)^2}} - e^{-\frac{x^2+y^2}{\sigma^2}} \qquad k = \sqrt{2}$$

Scale

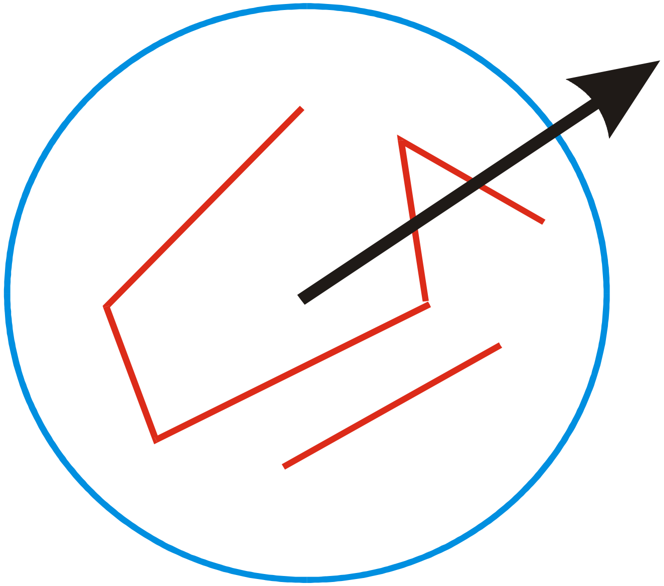
- Look for strong responses of DOG filter (Difference-Of-Gaussian) over scale space
- Only consider local maxima in both position and scale
- Fit quadratic around maxima for subpixel



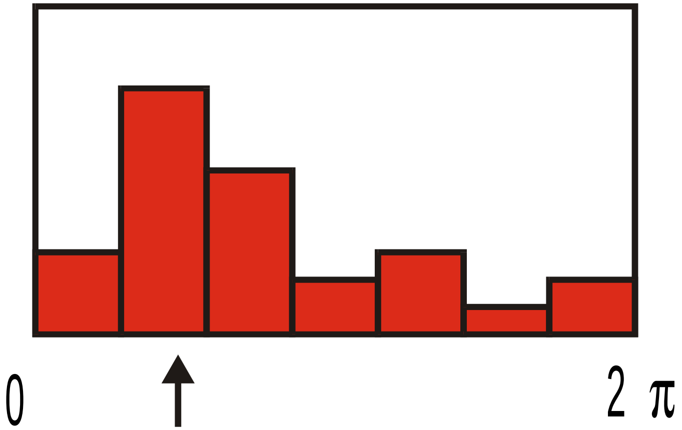




Orientation



- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)



Minimum contrast and "cornerness"

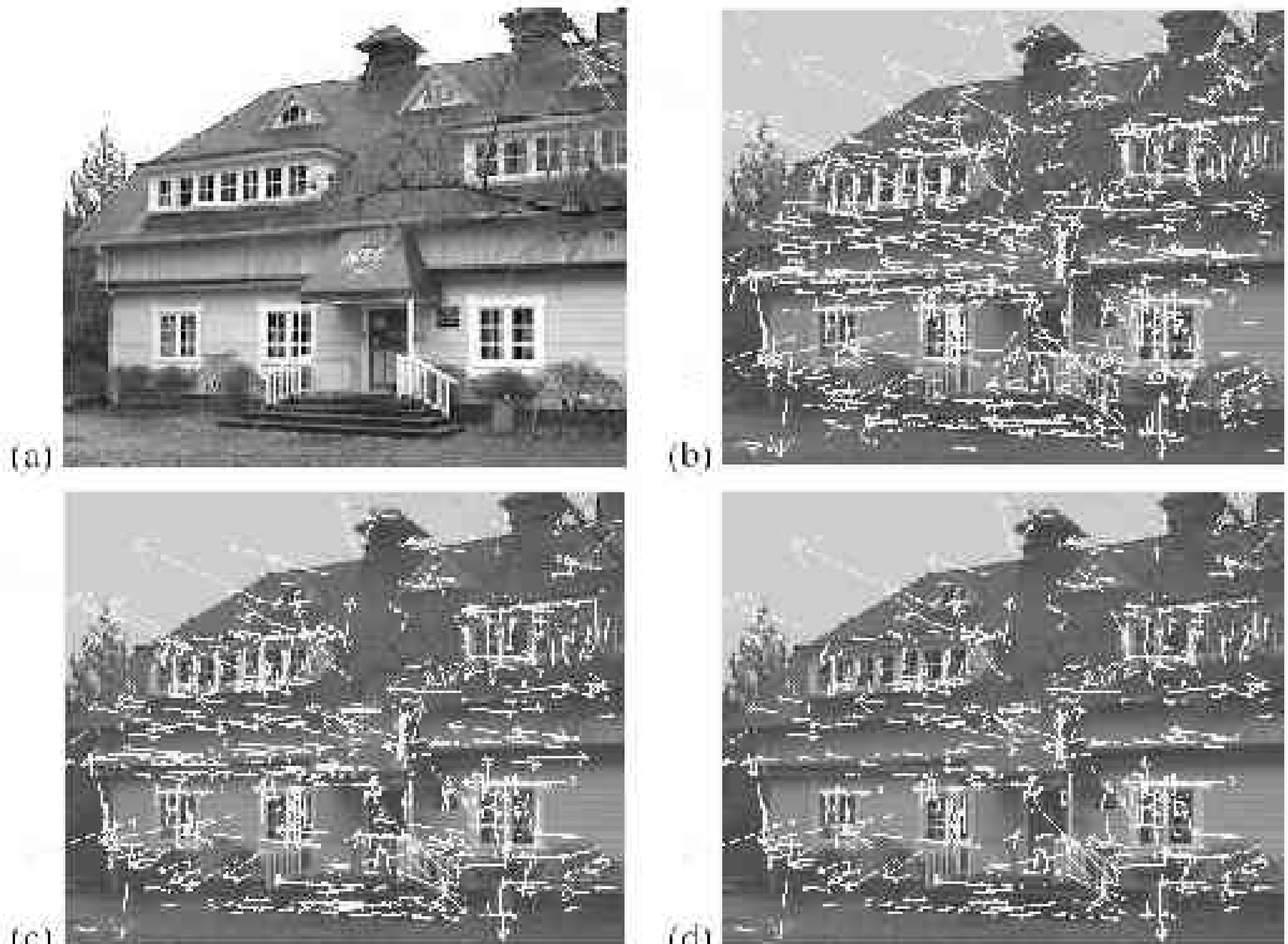
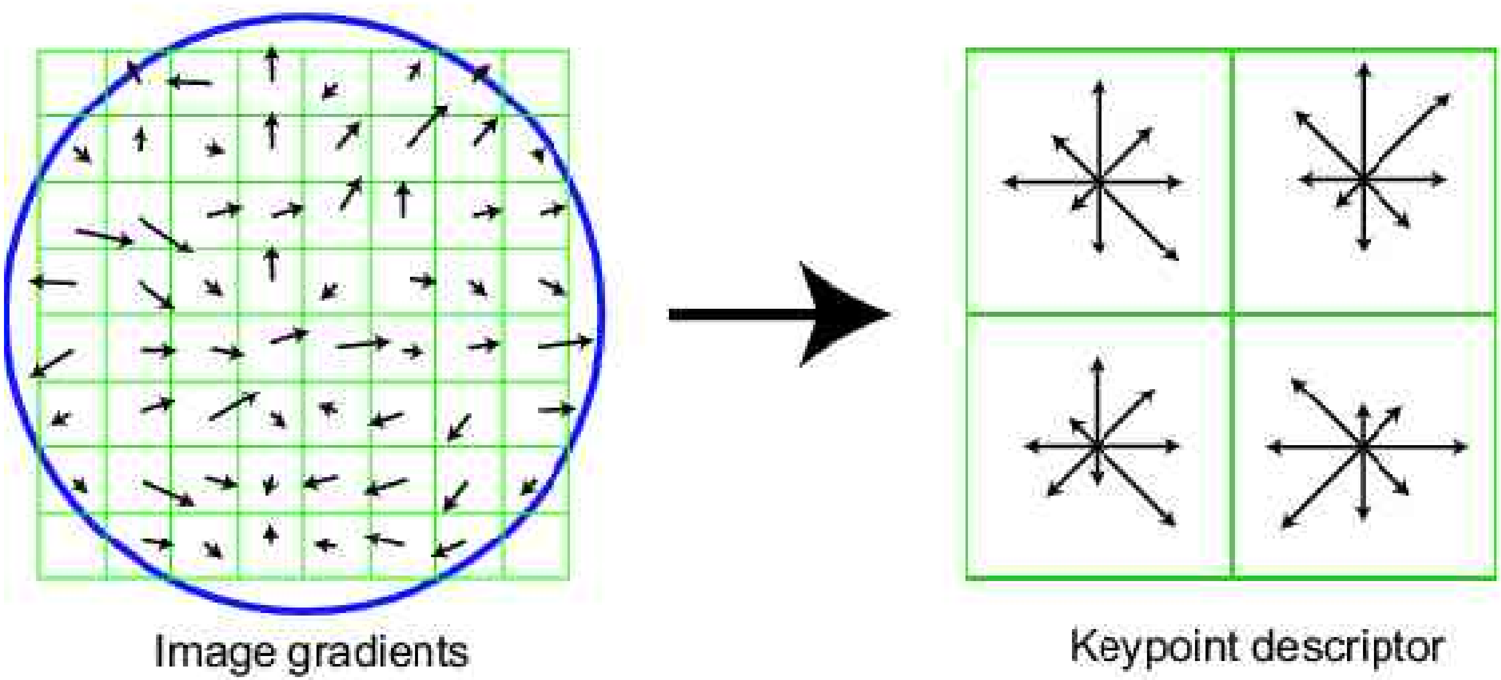


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principle curvatures.

SIFT descriptor

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions



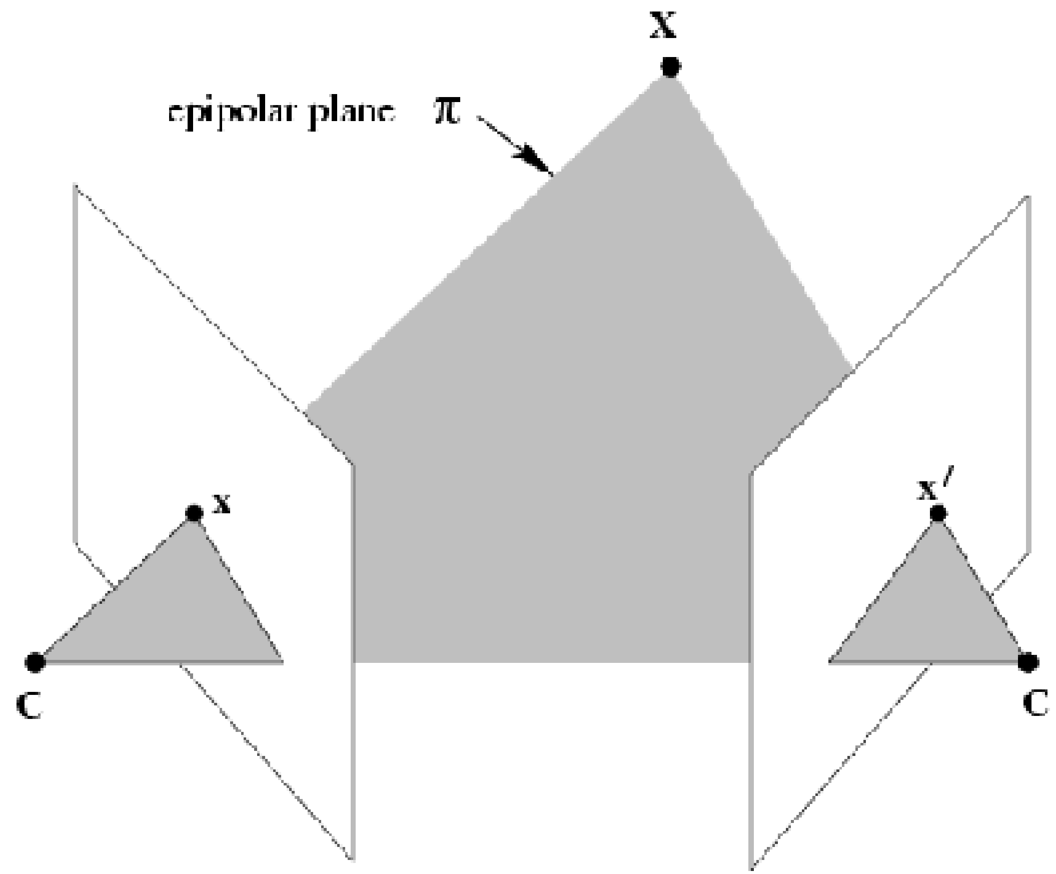
Three questions:

(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?

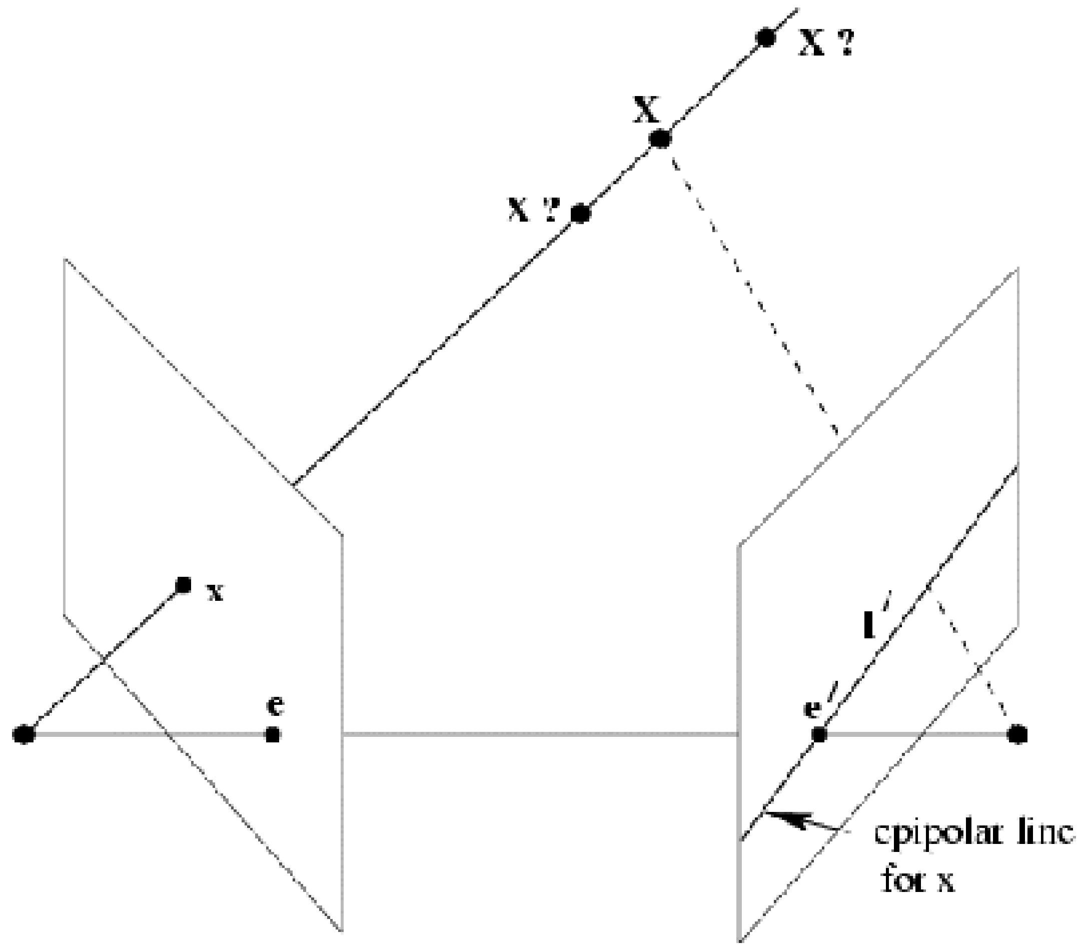
(iii) Scene geometry (structure): Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras P, P', what is the position of (their pre-image) X in space?

The epipolar geometry



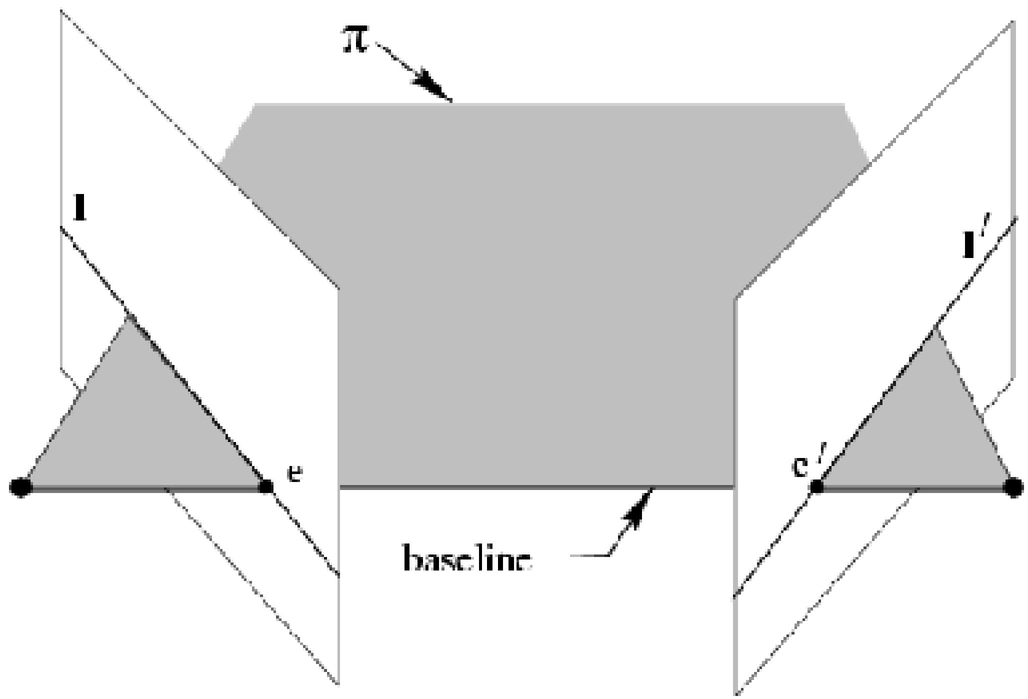
C, C', x, x' and X are coplanar

The epipolar geometry



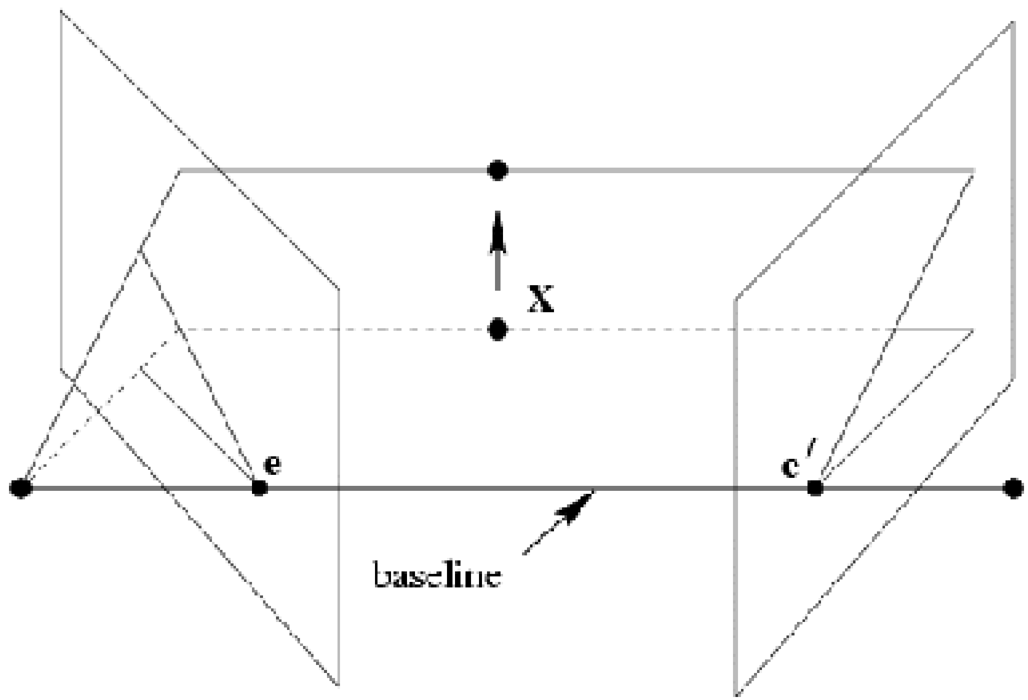
What if only C,C',x are known?

The epipolar geometry



All points on π project on $\mathbf{l}$ and $\mathbf{l}$'

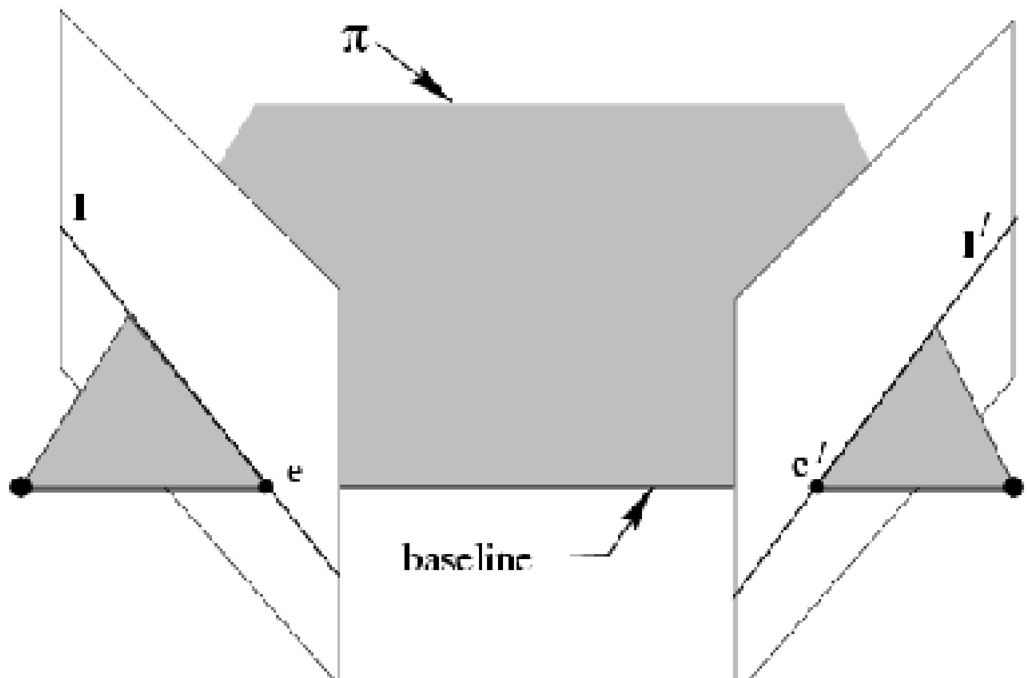
The epipolar geometry



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

The epipolar geometry

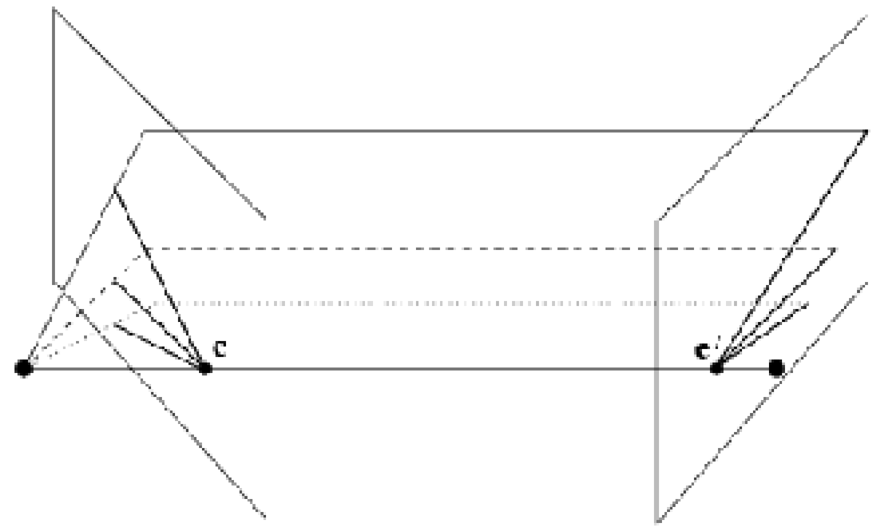
epipoles e,e’
= intersection of baseline with image plane
= projection of projection center in other image
= vanishing point of camera motion direction

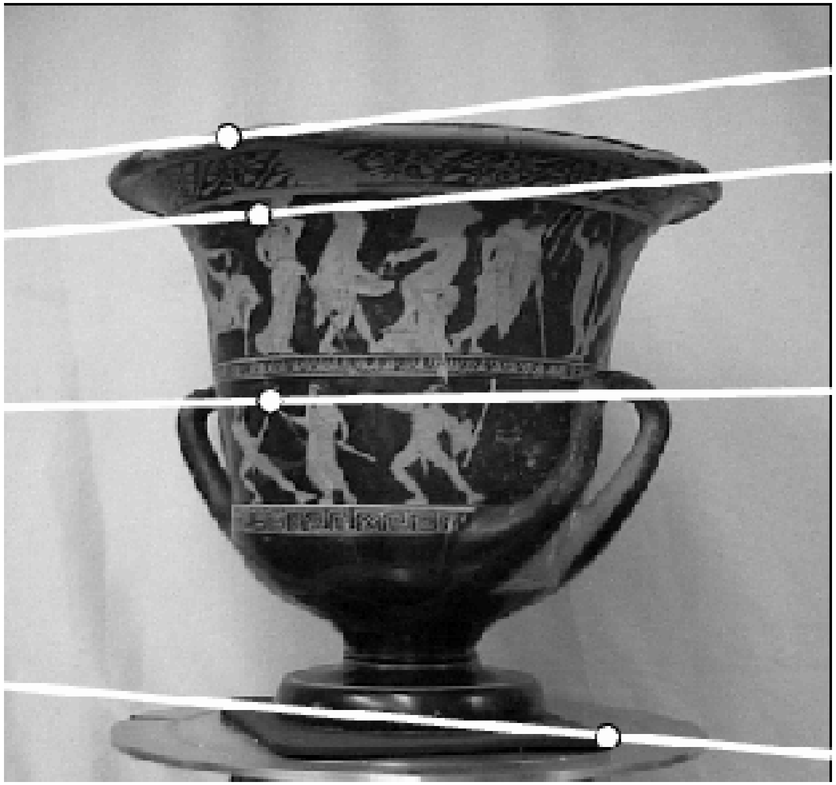


an epipolar plane = plane containing baseline (1-D family)

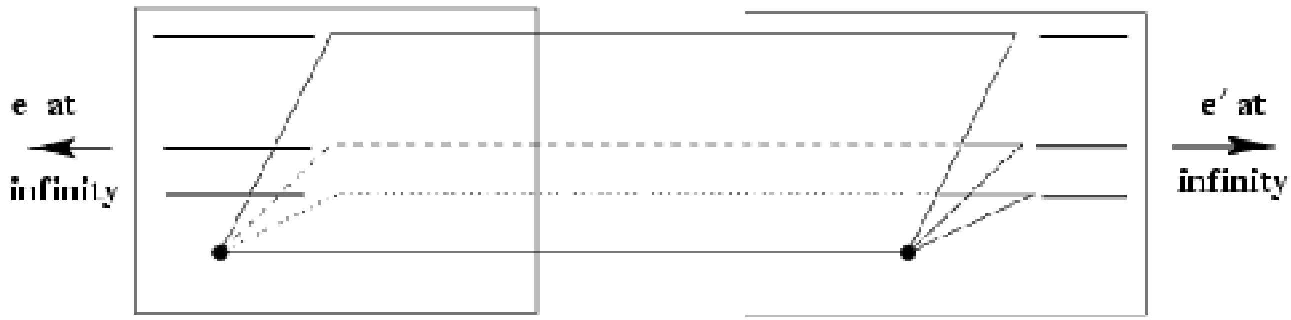
an epipolar line = intersection of epipolar plane with image
(always come in corresponding pairs)

Example: converging cameras



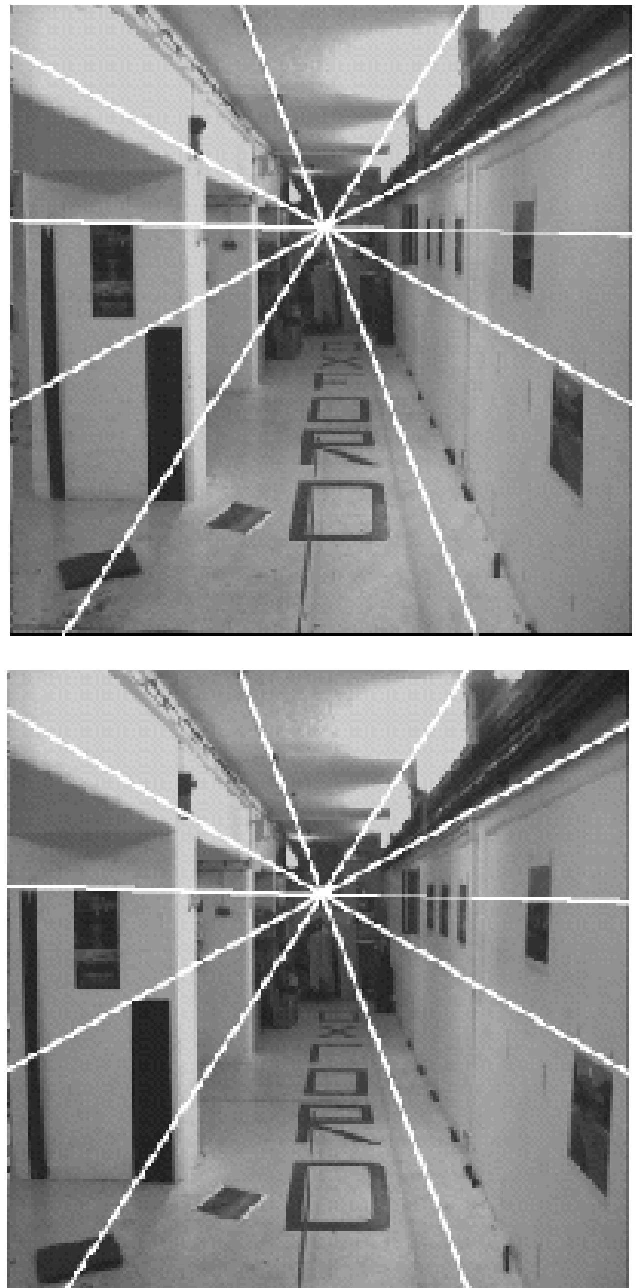


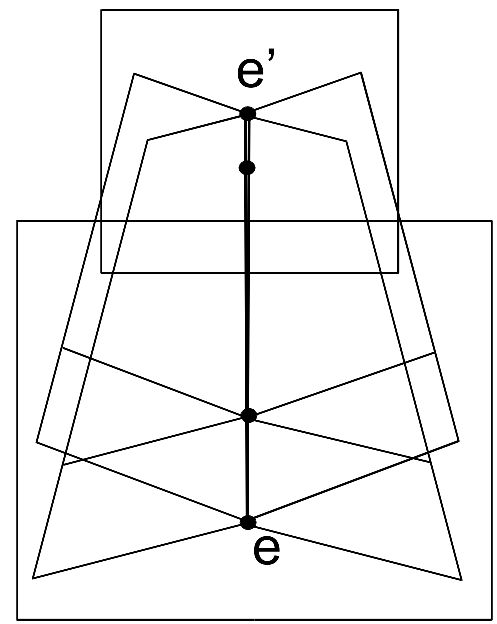
Example: motion parallel with image plane



(simple for stereo → rectification)

Example: forward motion





The fundamental matrix F

algebraic representation of epipolar geometry:

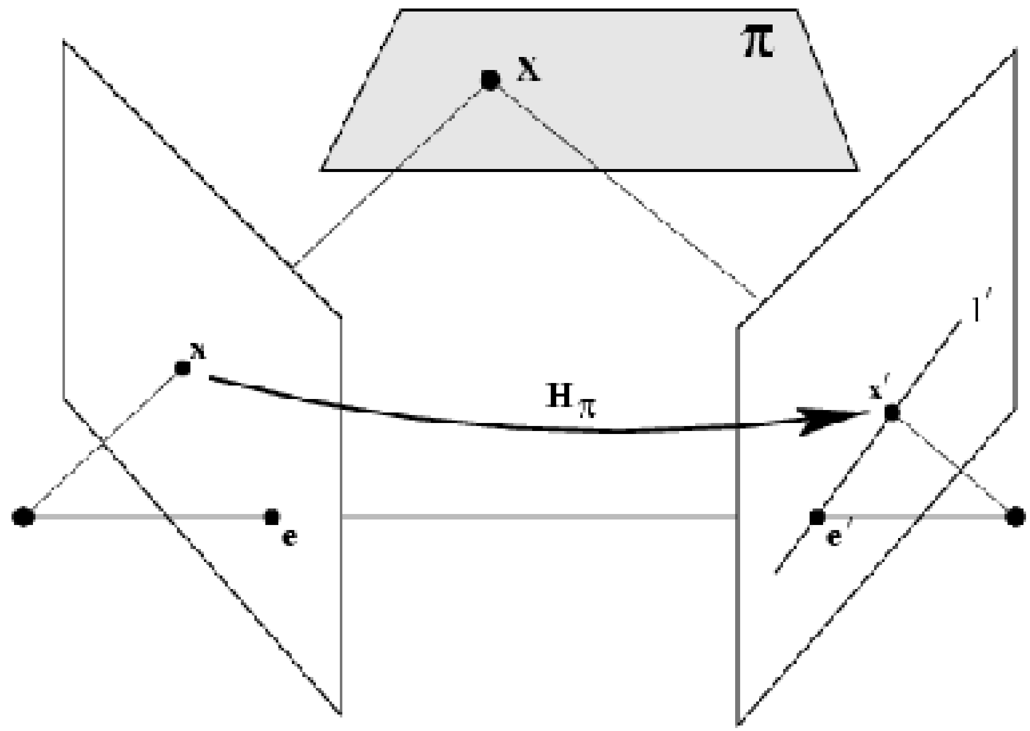
$l' \sim F x$

$x'^T F x = 0$

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

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

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:** $l'=Fx$ & $l=F^{T}x'$

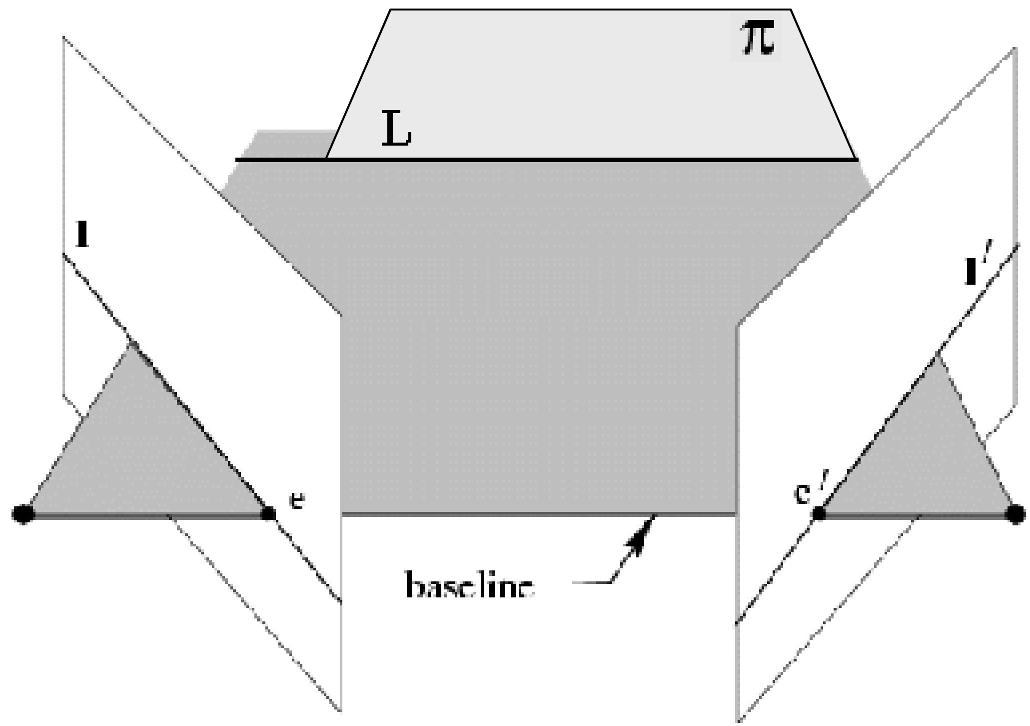
(iii) **Epipoles:** on all epipolar lines, thus $e'^{T}Fx=0, \forall x \Rightarrow e'^{T}F=0$, similarly $Fe=0$

(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)

The fundamental matrix F

relation to homographies



$$[e']_\times H_\pi = F \qquad l' = H_\pi^{-T} l \qquad e' = H_\pi e$$

valid for all plane homographies

Epipolar geometry: basic equation

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

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

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)

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

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

the NOT normalized 8-point algorithm

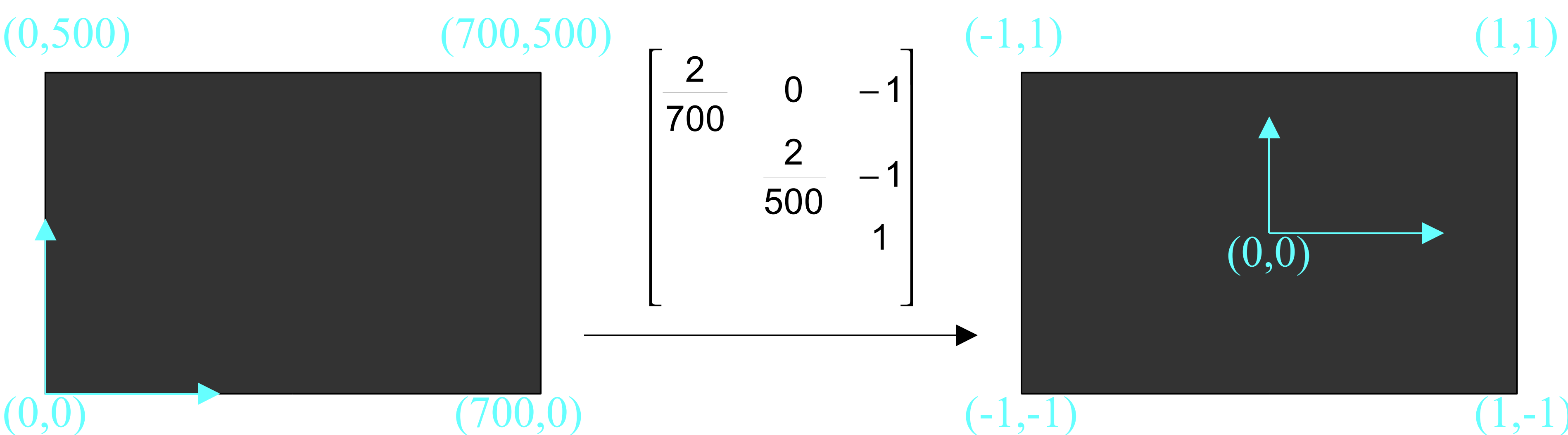
$$\begin{bmatrix} x_1 x_1' & y_1 x_1' & x_1' & x_1 y_1' & y_1 y_1' & y_1' & x_1 & y_1 & 1 \\ x_2 x_2' & y_2 x_2' & x_2' & x_2 y_2' & y_2 y_2' & y_2' & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n x_n' & y_n x_n' & x_n' & x_n y_n' & y_n y_n' & y_n' & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

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

! Orders of magnitude difference between column of data matrix
→ 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 (Hartley, PAMI´97)

Epipolar geometry computation: robust estimation (RANSAC)

Step 1. Extract features
Step 2. Compute a set of potential matches
Step 3. do

Step 3.1 select minimal sample
Step 3.2 compute solution(s) for F
Step 3.3 count inliers, if not promising stop

(generate hypothesis)

(verify hypothesis)

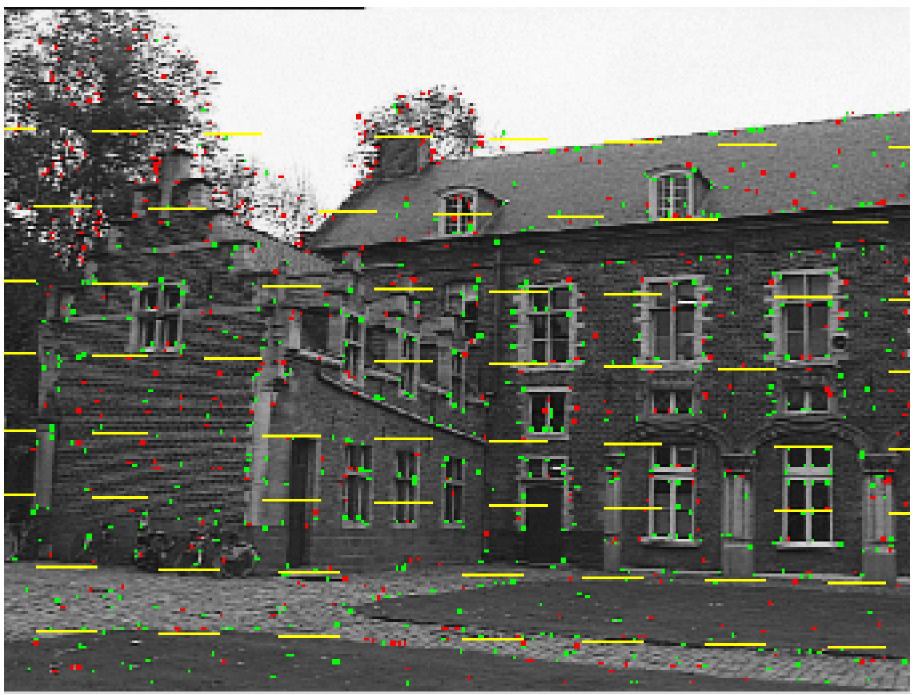
until $\Gamma(\#inliers,\#samples)<95\%$

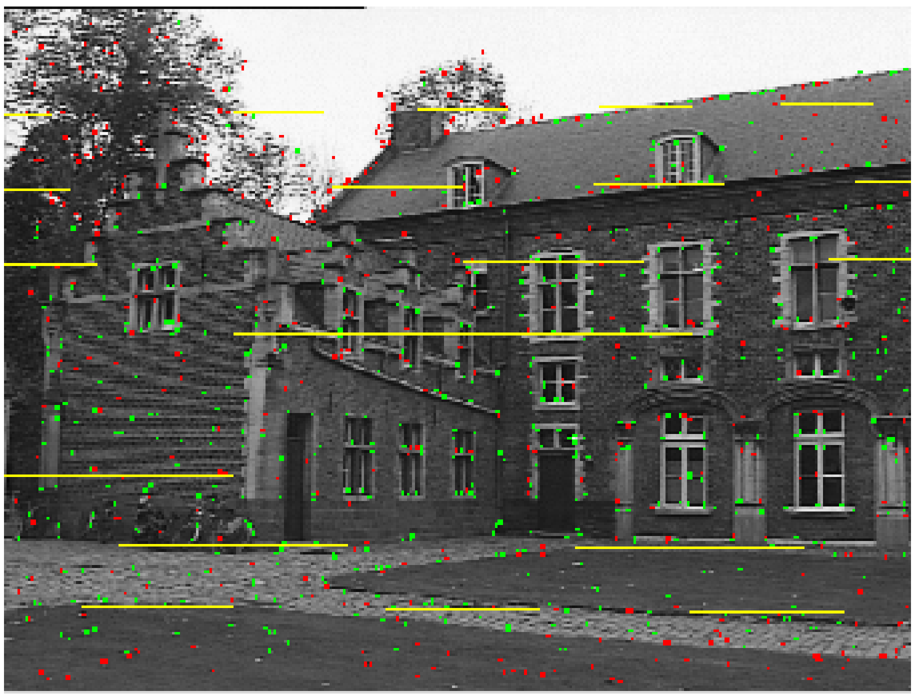
$$\Gamma = 1-\left(1-\left(\frac{\#inliers}{\#matches}\right)^{7}\right)^{\#samples}$$

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

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

Epipolar geometry computation





geometric relations between two views is fully described by recovered 3x3 matrix F

Cameras given F

Possible choice:

$$P = [I \mid 0] \quad P' = [[e']_\times F \mid e']$$

$$F = [e']_\times P'P^+ = [e']_\times [[e']_\times F \mid e'] \begin{bmatrix} I \\ 0 \end{bmatrix}$$

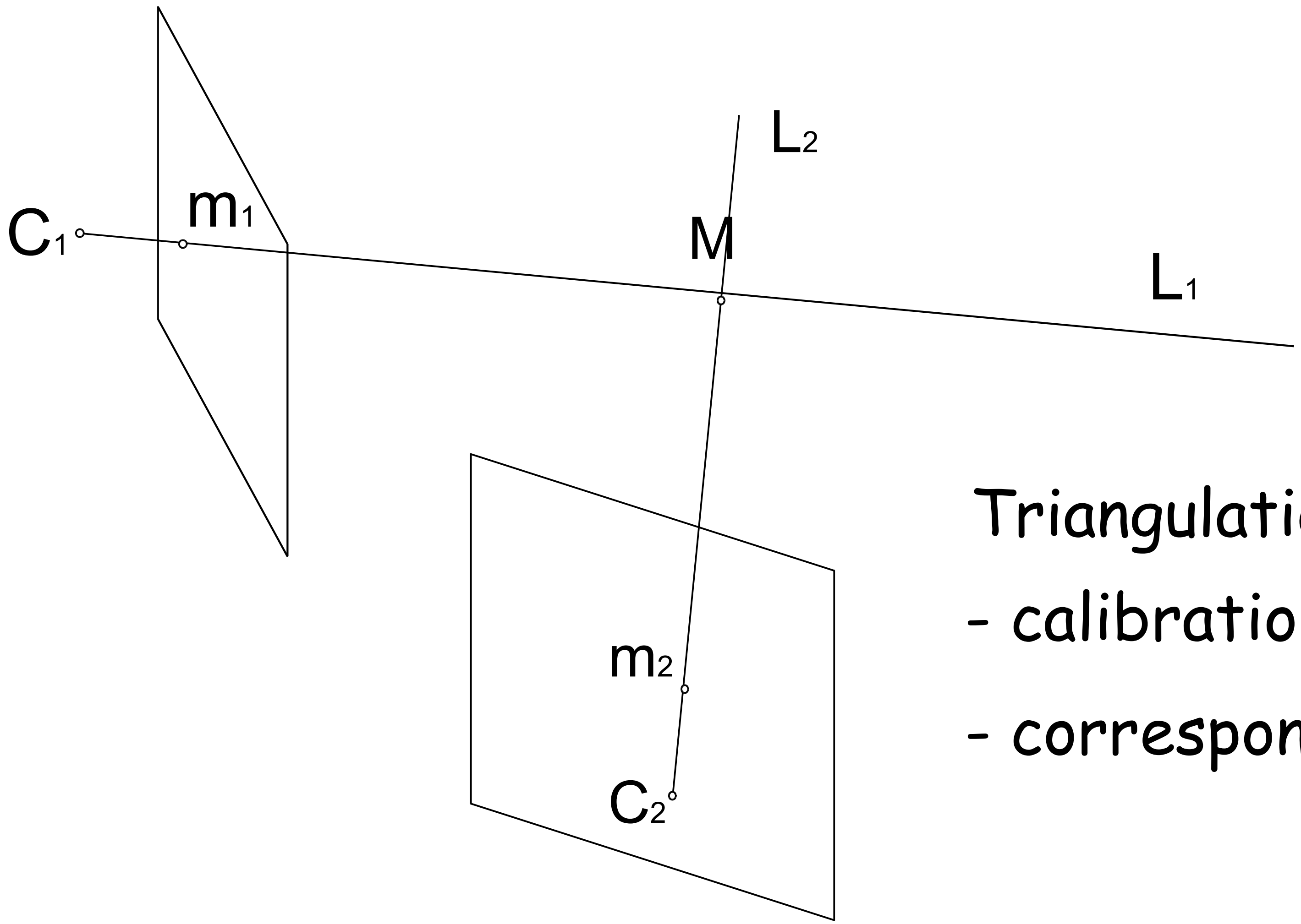
$$\left([e']_\times [e']_\times = e'.e'^T - \left(e'^T.e' \right) I \right)$$

$$= \left(e'.e'^T - \left(e'^T.e' \right) \right) F = \lambda F$$

representation:

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

Triangulation



Triangulation

- calibration
- correspondences

Structure and motion recovery

- Sequential approach
- Initialize motion from two images
- Initialize structure
- For each additional view
  - Determine pose of camera
  - Refine and extend structure
- Refine structure and motion

Initial projective camera motion

- Choose P and P′ compatible with F

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_{3\times3} & 0_3 \end{bmatrix}$$

$$\mathbf{P}' = \begin{bmatrix} e' \times \mathbf{F} + e' a^{T} & e' \end{bmatrix}$$

(reference plane;arbitrary)

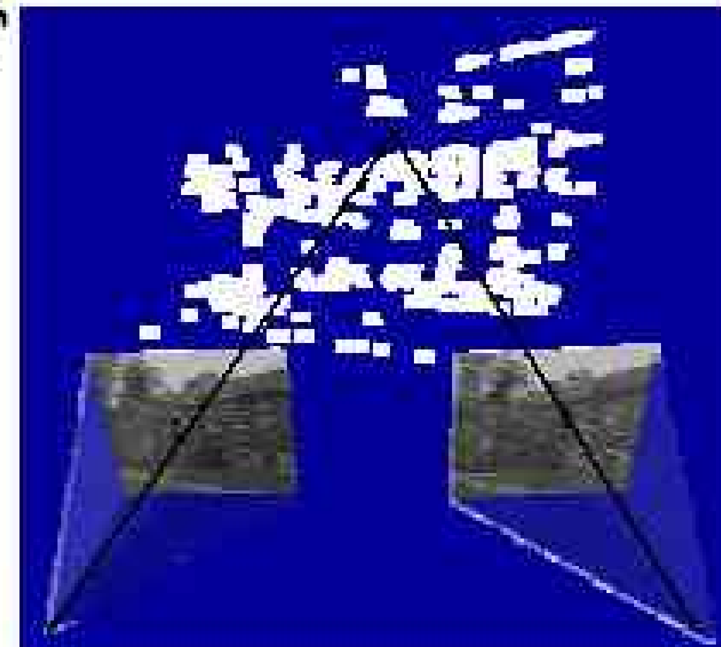
Reconstruction up to projective ambiguity

(Faugeras´92,Hartley´92)

- Initialize motion
- Initialize structure
- For each additional view
  - Determine pose of camera
  - Refine and extend structure
- Refine structure and motion

Initializing projective structure

- Reconstruct matches in projective frame
- by minimizing the reprojection error



$$D(\mathbf{m}_1, \mathbf{P}_1\mathbf{M})^2 + D(\mathbf{m}_2, \mathbf{P}_2\mathbf{M})^2$$

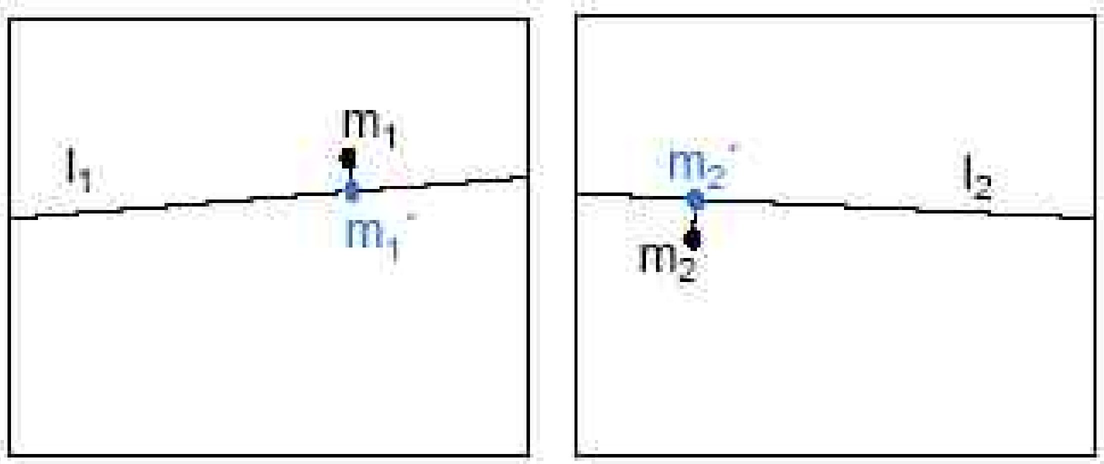
Non-iterative optimal solution

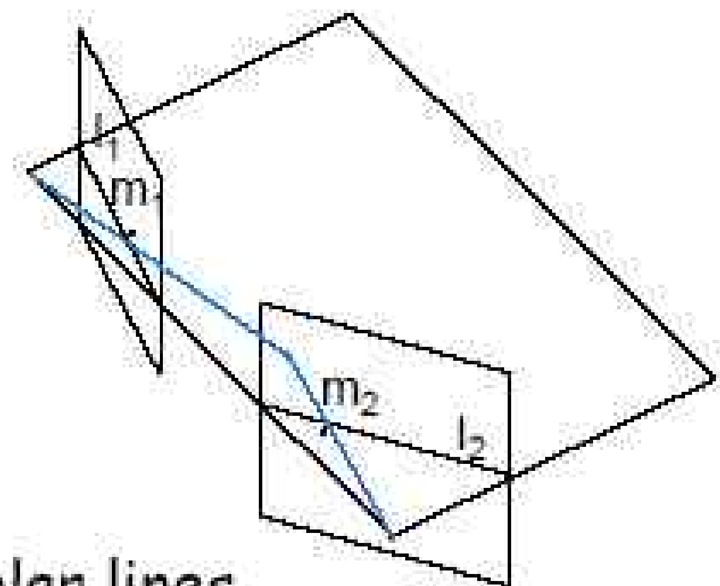
(see Hartley&Sturm,CVIU'97)

- Initialize motion
- Initialize structure
- For each additional view
  - Determine pose of camera
  - Refine and extend structure
- Refine structure and motion

Optimal 3D point in epipolar plane

- Given an epipolar plane, find best 3D point for (m_1, m_2)





Select closest points (m_1', m_2') on epipolar lines
Obtain 3D point through exact triangulation
Guarantees minimal reprojection error (given this epipolar plane)

Initializing projective structure

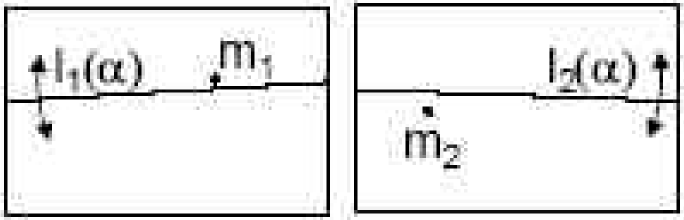
- Reconstruct matches in projective frame by minimizing the reprojection error

$$D(\mathbf{m}_1, \mathbf{P}_1\mathbf{M})^2 + D(\mathbf{m}_2, \mathbf{P}_2\mathbf{M})^2 \quad \textbf{3DOF}$$

- Non-iterative method (Hartley and Sturm, CVIU´97)

Determine the epipolar plane for reconstruction

$$D(\mathbf{m}_1, \mathbf{l}_1(\alpha))^2 + D(\mathbf{m}_2, \mathbf{l}_2(\alpha))^2 \quad \text{(polynomial of degree 6)}$$



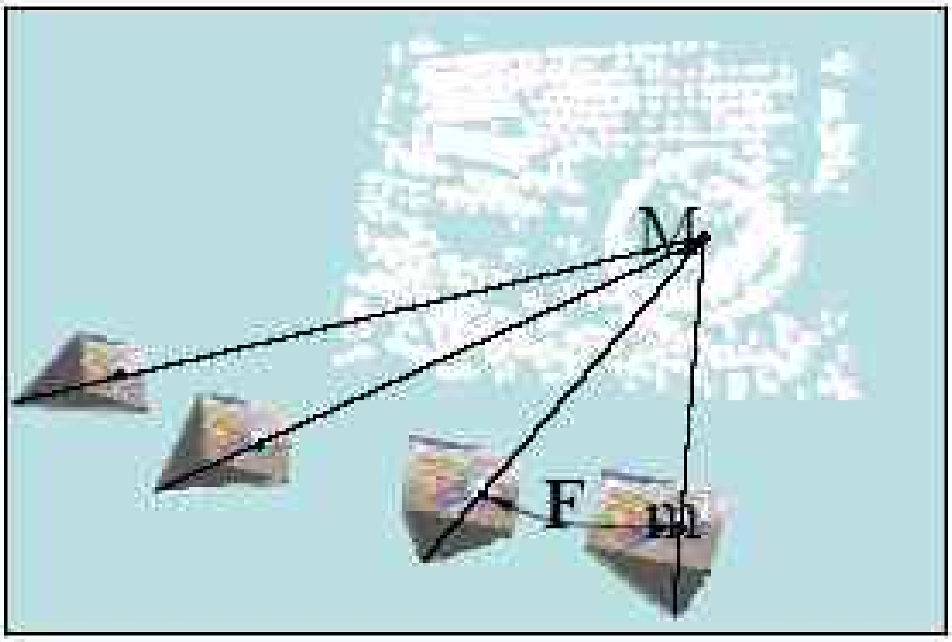
1DOF

Reconstruct optimal point from selected epipolar plane

Projective pose estimation

- Infere 2D-3D matches from 2D-2D matches
- Compute pose from $\mathrm{m} \sim \mathbf{P}\mathrm{M}$ (RANSAC,6pts)

$$\begin{bmatrix} \mathrm{M}^{\top} & 0 & \mathrm{M}^{\top}x \\ 0 & \mathrm{M}^{\top} & \mathrm{M}^{\top}y \end{bmatrix} \mathbf{p} = 0$$



Inliers:

$$\exists \mathrm{M}\, \forall \mathrm{m}_i \; D\left(\mathbf{P}_i \mathrm{M}, \mathrm{m}_i\right) < D_{\mathrm{in}}$$

- Initialize motion
- Initialize structure
- For each additional view
  - Determine pose of camera
  - Refine and extend structure
- Refine structure and motion

Refining and extending structure

- Refining structure

$$\frac{1}{P_3\widetilde{M}}\begin{bmatrix}P_3x - P_1 \\ P_3y - P_1\end{bmatrix}M = 0$$ (Iterative linear)

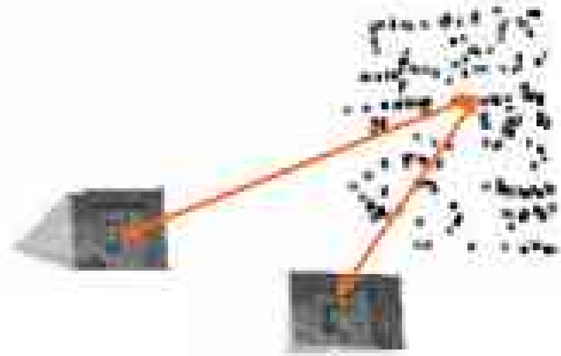
- Extending structure

  Triangulation (Hartley&Sturm,CVIU´97)

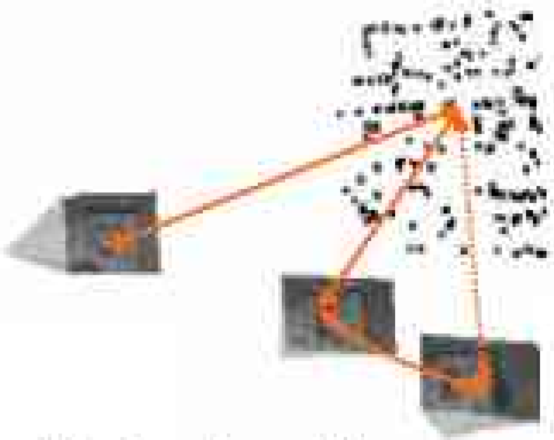
- Initialize motion
- Initialize structure
- For each additional view
  - Determine pose of camera
  - Refine and extend structure
- Refine structure and motion

Sequential Structure and Motion Computation

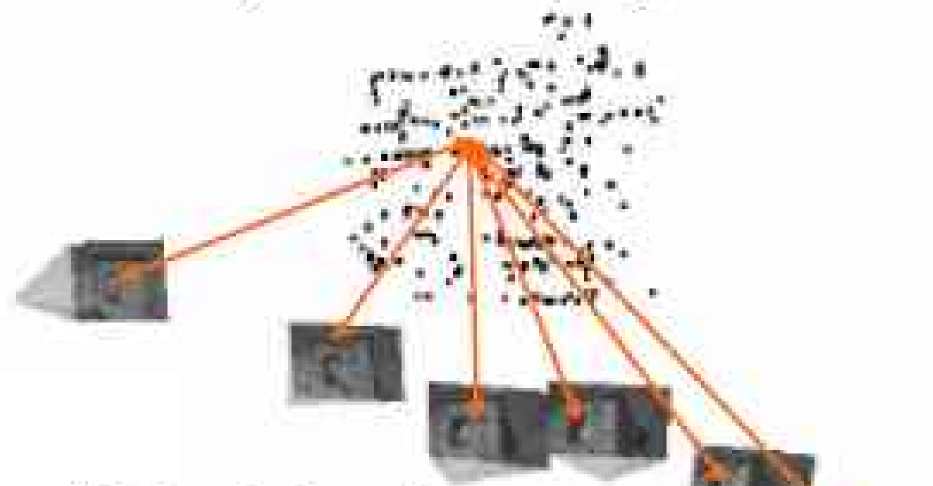
Initialize Motion
(P_1,P_2 compatibel with F)



Initialize Structure
(minimize reprojection error)



Extend motion
(compute pose through matches seen in 2 or more previous views)



Extend structure
(Initialize new structure, refine existing structure)

Refining structure and motion

- Minimize reprojection error

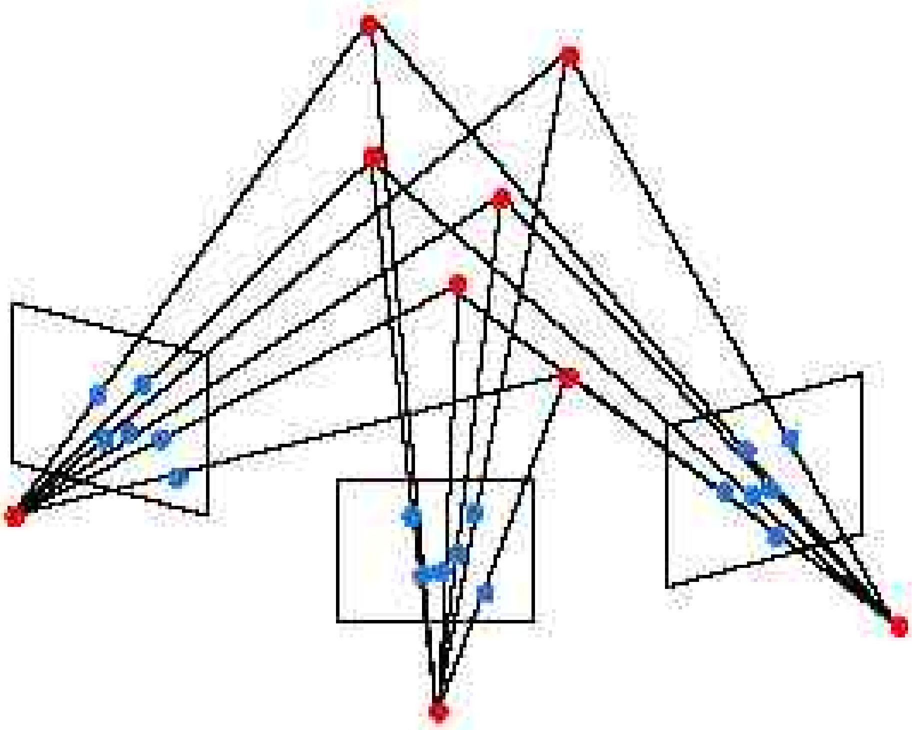
$$\min_{\hat{P}_k,\hat{M}_i} \sum_{k=1}^{m} \sum_{i=1}^{n} D\left(m_{ki}, \hat{P}_k \hat{M}_i\right)^2$$

  - Maximum Likelyhood Estimation
    (if error zero-mean Gaussian noise)
  - Huge problem but can be solved efficiently
    (Bundle adjustment)

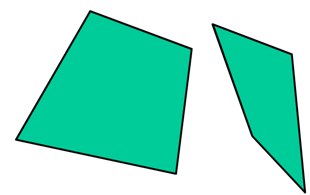
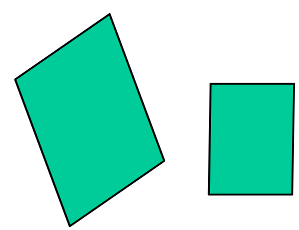
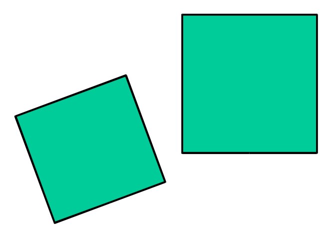
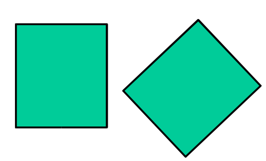
Bundle adjustment

- Developed in photogrammetry in 50´s

(Slama´80;Triggs et al´00;
Hartley and Zisserman'00)



Hierarchy transformations

		transformed squares	invariants
Projective 8dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, order of contact (intersection, tangency, inflection, etc.), cross ratio
Affine 6dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallellism, ratio of areas, ratio of lengths on parallel lines (e.g midpoints), linear combinations of vectors (centroids). **The line at infinity l_∞**
Similarity 4dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratios of lengths, angles. **The circular points I,J**
Euclidean 3dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		lengths, areas.

Five points define a conic

For each point the conic passes through

$$a x_i^2 + b x_i y_i + c y_i^2 + d x_i + e y_i + f = 0$$

or

$$\left(x_i^2, x_i y_i, y_i^2, x_i, y_i, 1\right).\mathbf{c} = 0 \qquad \mathbf{c} = \left(a, b, c, d, e, f\right)^\mathsf{T}$$

stacking constraints yields

$$\begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2 y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & x_3 y_3 & y_3^2 & x_3 & y_3 & 1 \\ x_4^2 & x_4 y_4 & y_4^2 & x_4 & y_4 & 1 \\ x_5^2 & x_5 y_5 & y_5^2 & x_5 & y_5 & 1 \end{bmatrix} \mathbf{c} = 0$$

Quadrics and dual quadrics

$X^T Q X = 0$ (Q : 4x4 symmetric matrix)

$$Q = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet \\ \circ & \circ & \bullet & \bullet \\ \circ & \circ & \circ & \bullet \end{bmatrix}$$

- 9 d.o.f.
- in general 9 points define quadric
- det Q=0 ↔ degenerate quadric
- tangent plane $\pi = QX$

$\pi^T Q^* \pi = 0$

- relation to quadric $Q^* = Q^{-1}$ (non-degenerate)

The line at infinity

$$\mathbf{l}'_\infty = \mathbf{H}_A^{-\mathsf{T}} \mathbf{l}_\infty = \begin{bmatrix} \mathbf{A}^{-\mathsf{T}} & 0 \\ -\mathbf{A}\,\mathbf{t} & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \mathbf{l}_\infty$$

The line at infinity l_∞ is a fixed line under a projective transformation H if and only if H is an affinity

Note: not fixed pointwise

The plane at infinity

$$\pi'_\infty = \mathbf{H}_A^{-\mathsf{T}} \pi_\infty = \begin{bmatrix} \mathbf{A}^{-\mathsf{T}} & 0 \\ -\mathbf{A}\,\mathrm{t} & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \pi_\infty$$

The plane at infinity π_∞ is a fixed plane under a projective transformation H iff H is an affinity

1. canical position $\pi_\infty = (0,0,0,1)^\mathsf{T}$
2. contains directions $\mathrm{D} = (X_1, X_2, X_3, 0)^\mathsf{T}$
3. two planes are parallel ⇔ line of intersection in π_∞
4. line // line (or plane) ⇔ point of intersection in π_∞

The absolute conic

The absolute conic Ω_∞ is a (point) conic on π_∞.

In a metric frame:

$$\left.\begin{array}{c} X_1^2 + X_2^2 + X_3^2 \\ X_4 \end{array}\right\} = 0$$

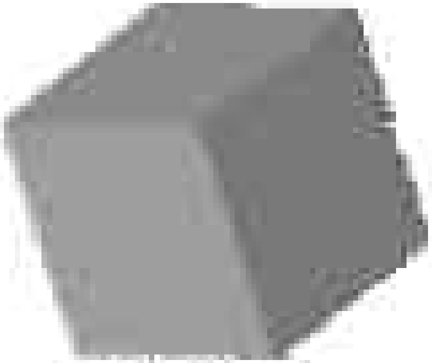
or conic for directions:
(with no real points)

$$\left(X_1, X_2, X_3\right)\mathrm{I}\left(X_1, X_2, X_3\right)^{\mathsf{T}}$$

The absolute conic Ω_∞ is a fixed conic under the projective transformation $\mathbf{H}$ iff $\mathbf{H}$ is a similarity

1. Ω_∞ is only fixed as a set
2. Circle intersect Ω_∞ in two circular points
3. Spheres intersect π_∞ in Ω_∞

Stratification of geometry

Projective	Affine	Metric
		
15 DOF	12 DOF	7 DOF
	plane at infinity	*absolute conic*
colinearity, cross-ratio	parallelism	angles, rel.dist.

More general ←

More structure →

Constraints ?

Scene constraints

- Parallellism, vanishing points, horizon, ...
- Distances, positions, angles, ...

Unknown scene $\rightarrow$ no constraints

Camera extrinsics constraints

- Pose, orientation, ...

Unknown camera motion $\rightarrow$ no constraints

Camera intrinsics constraints

- Focal length, principal point, aspect ratio & skew

Perspective camera model too general
$\rightarrow$ some constraints

Self-calibration

- Upgrade from *projective* structure to *metric* structure using *constraints on intrinsic* camera parameters
  - Constant intrinsics (Faugeras et al.´92; Hartley´93; Pollefeys and Van Gool´96;Triggs´97, ...)
  - Some known intrinsics, other varying (Pollefeys et al. ´98,...)
  - Constraints on intrincs and restricted motion (e.g. pure translation, pure rotation, planar motion)

    (Moons et al.´94, Hartley´94, Armstrong et al.´96,...)

Euclidean projection matrix

Factorization of Euclidean projection matrix

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R}^{\mathsf{T}} \quad -\mathbf{R}^{\mathsf{T}}\mathbf{t}\right]$$

Intrinsics: $\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix}$ (camera geometry)

Extrinsics: $(\mathbf{R}, \mathrm{t})$ (camera motion)

Note: every projection matrix can be factorized,
but only meaningful for euclidean projection matrices

Constraints on intrinsic parameters

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix}$$

- Constant

  e.g. fixed camera: $\mathbf{K}_1 = \mathbf{K}_2 = \cdots$

- Known

  e.g. rectangular pixels: $s = 0$ (sufficient in general, Pollefeys et al.'98)

  square pixels: $s = 0, f_x = f_y$

  principal point known: $(c_x, c_y) = \left(\frac{w}{2}, \frac{h}{2}\right)$

The Absolute Quadric

Eliminate extrinsics from equation

$$K\left[R^T \quad -R^T t\right] \longrightarrow K R^T R K^T \longrightarrow KK^T$$

Equivalent to projection of quadric

$$P \Omega P^T = KK^T \qquad \Omega^* = \mathrm{diag}(1110) \quad \text{Absolute Quadric}$$

Absolute Quadric also exists in projective world

$$KK^T = P\Omega^* P^T = (PT^{-1})(T\Omega^* T^T)(T^{-T} P^T)$$

$$= P'\Omega'^* P'^T$$

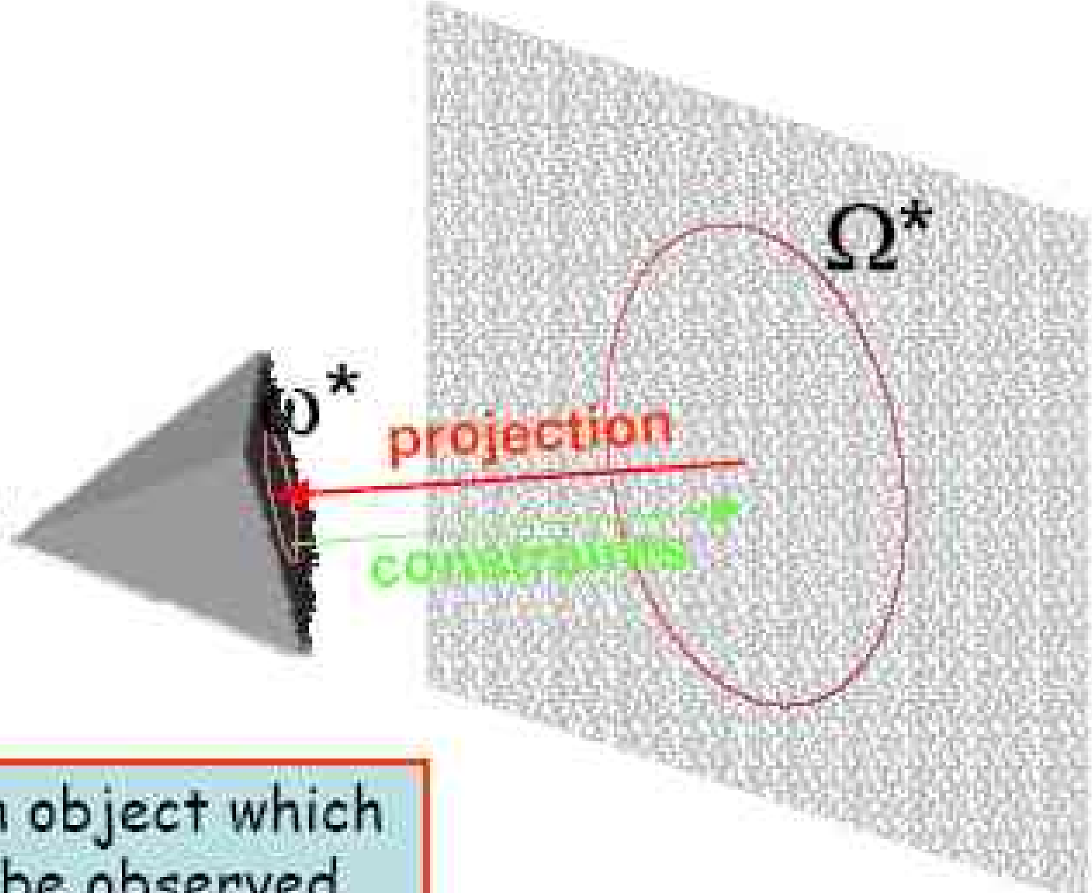
Transforming world so that $\Omega'^* \rightarrow \Omega^*$ reduces ambiguity to metric

Absolute quadric and self-calibration

Projection equation:

$$\omega_i^* = \mathbf{P}_i \Omega^* \mathbf{P}_i^\top = \mathbf{K}_i \mathbf{K}_i^\top$$

- Translate constraints on K
- through projection equation
- to constraints on Ω^*

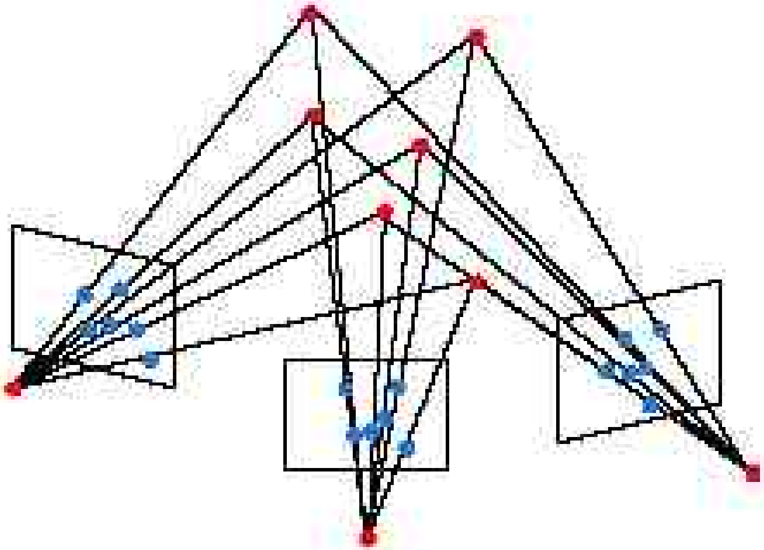


> Absolute Quadric = calibration object which is always present but can only be observed through constraints on the intrinsics

Refine Metric Structure and Motion

- Use metric bundle adjustment
  - Use Euclidean parameterization for projection matrices
  - Same sparseness advantages, also use radial distortion

$$\arg\min_{P_k, M_i} \sum_{k=1}^{m} \sum_{i=1}^{n} D\left(m_{ki}, P_k(M_i)\right)^2$$



Practical linear self-calibration

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix}$$

(Pollefeys et al., ECCV'02)

Don't treat all constraints equal

$$\mathbf{KK} = \mathbf{P}\Omega^*\mathbf{P}^\top \approx \begin{bmatrix} \hat{f}^2 & 0 & 0 \\ 0 & \hat{f}^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

after normalization!

(relatively accurate for most cameras)

$$\hat{f} \approx 1$$

(only rough aproximation, but still usefull to avoid degenerate configurations)

$$\frac{1}{0.2}\left(\left(\mathbf{P}\Omega^*\mathbf{P}^\top\right)_{11} - \left(\mathbf{P}\Omega^*\mathbf{P}^\top\right)_{22}\right) = 0$$

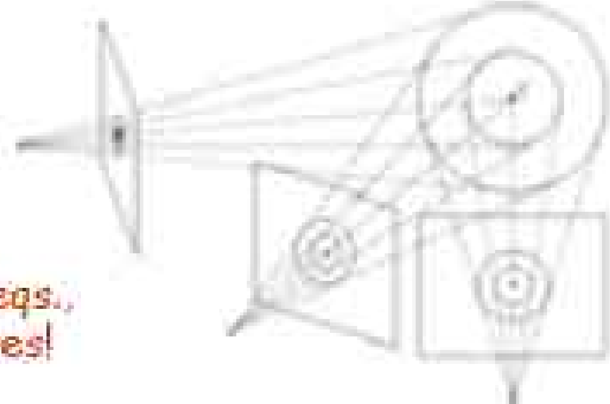
$$\frac{1}{0.01}\left(\mathbf{P}\Omega^*\mathbf{P}^\top\right)_{12} = 0$$

$$\frac{1}{0.1}\left(\mathbf{P}\Omega^*\mathbf{P}^\top\right)_{13} = 0$$

$$\frac{1}{0.1}\left(\mathbf{P}\Omega^*\mathbf{P}^\top\right)_{23} = 0$$

$$\frac{1}{9}\left(\left(\mathbf{P}\Omega^*\mathbf{P}^\top\right)_{11} - \left(\mathbf{P}\Omega^*\mathbf{P}^\top\right)_{33}\right) = 0$$

$$\frac{1}{9}\left(\left(\mathbf{P}\Omega^*\mathbf{P}^\top\right)_{22} - \left(\mathbf{P}\Omega^*\mathbf{P}^\top\right)_{33}\right) = 0$$



when fixating point at image-center not only absolute quadric diag(1,1,1,0) satisfies ICCV98 eqs., but also diag(1,1,1,a), i.e. real or imaginary spheres!

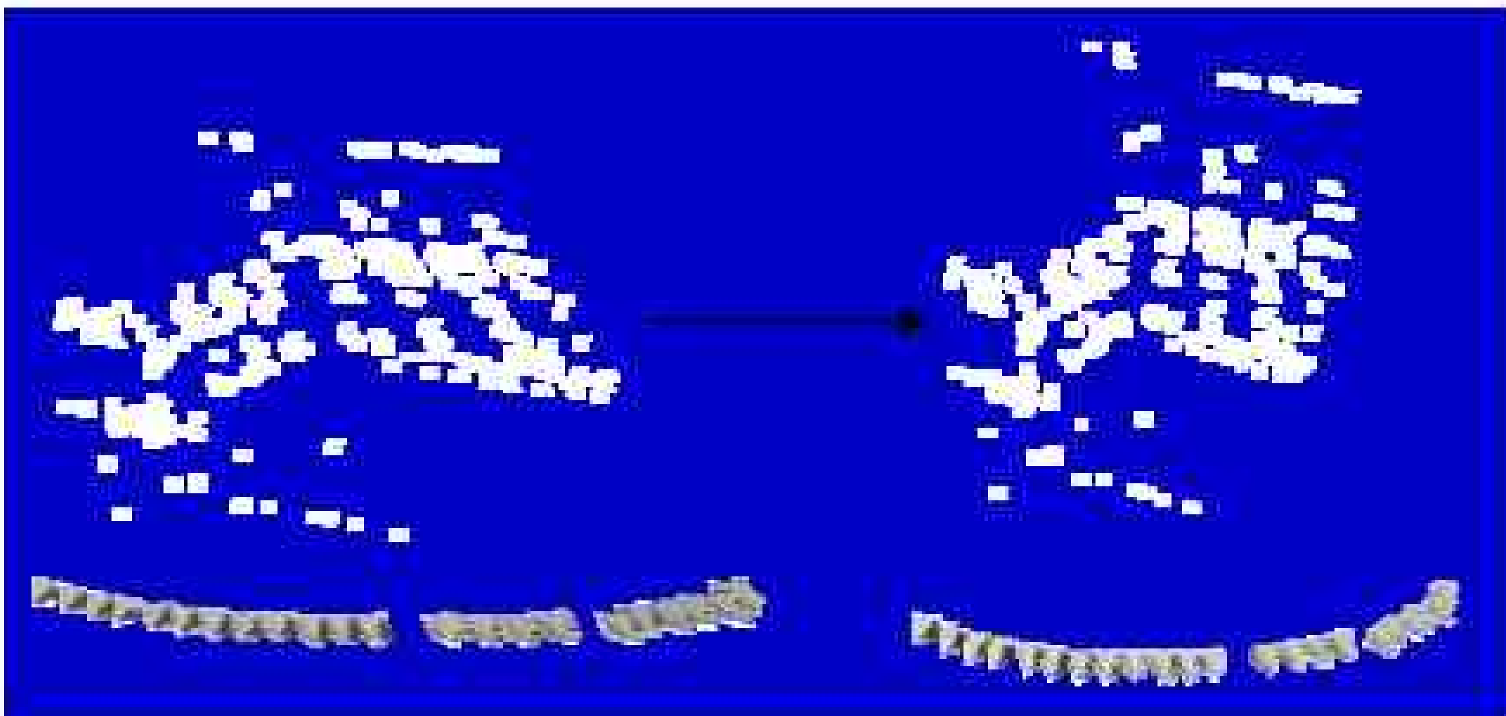
Upgrade from projective to metric

Locate Ω^* in projective reconstruction (self-calibration)

Problem: not always unique (Critical Motion Sequences)

Transform projective reconstruction

to bring Ω^* in canonical position



Dense model reconstruction