6. 3D Reconstruction

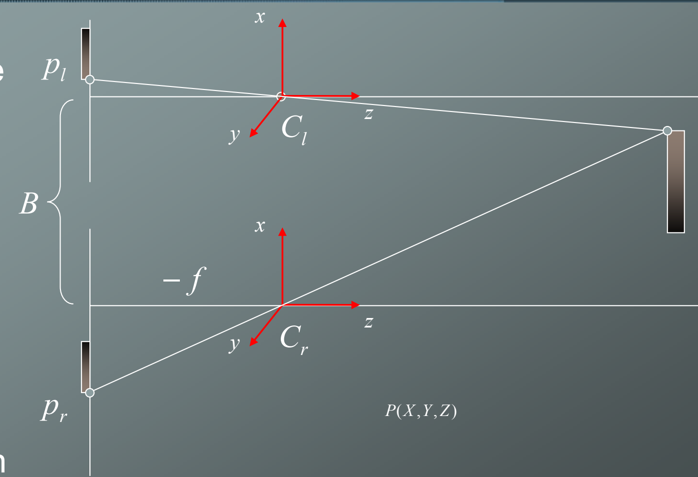
Computer Vision

Zoltan Kato

http://www.inf.u-szeged.hu/~kato/

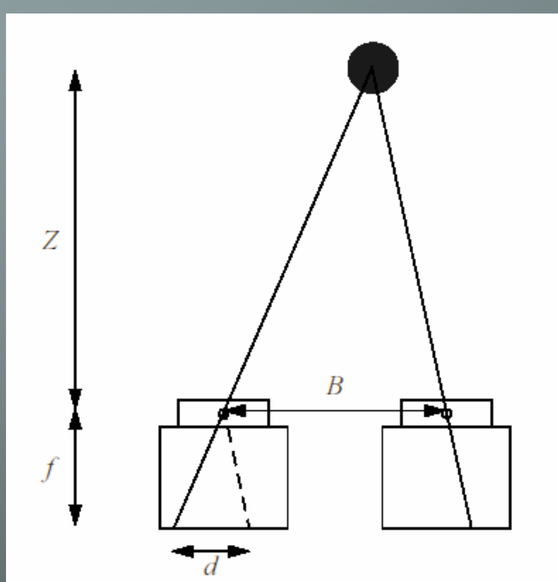
Standard stereo setup

- Image planes of cameras are parallel.
- Focal points are at same height.
- Focal lengths same.
- → epipolar lines are horizontal scan lines.



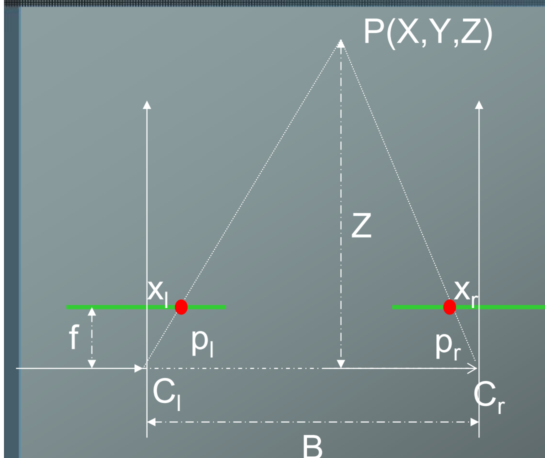
Derive expression for Z as a function of x_l, x_r, f, B

Disparity and depth



adapted from D. Young

Depth reconstruction



$$\frac{B + x_r - x_l}{Z - f} = \frac{B}{Z}$$

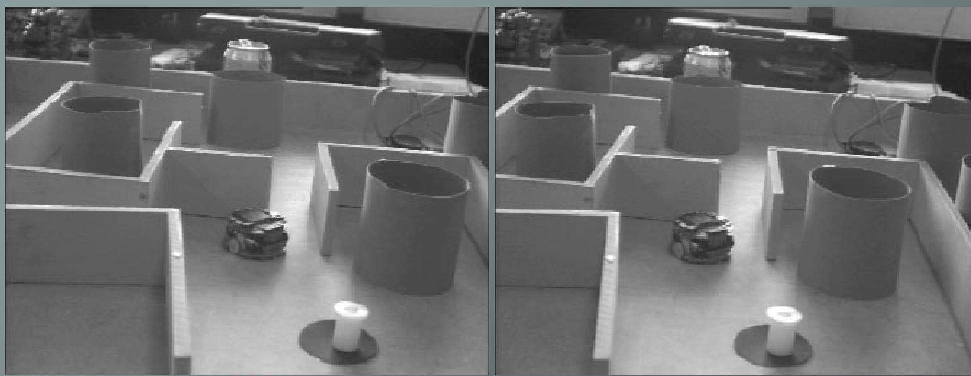
$$Z = f\frac{B}{x_l - x_r}$$

Disparity: $d = x_l - x_r$

$$Z = f\frac{B}{d}$$

Then given **Z**, we can compute **X** and **Y**.
B is the stereo baseline
d measures the difference in retinal position between corresponding points

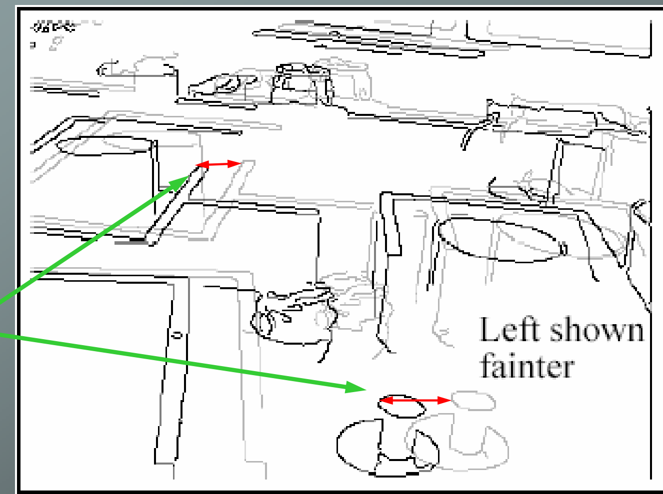
Stereo disparity example



Cameras have parallel optical axes, separated by horizontal translation (approximately)

Stereo disparity example

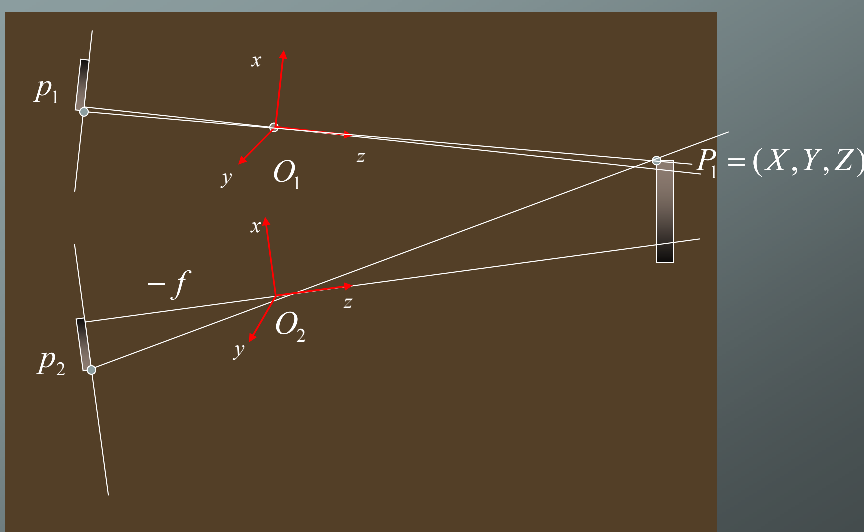
Disparity gets smaller with increasing depth



courtesy of D. Young

Left and right edge images, superimposed

What If…?

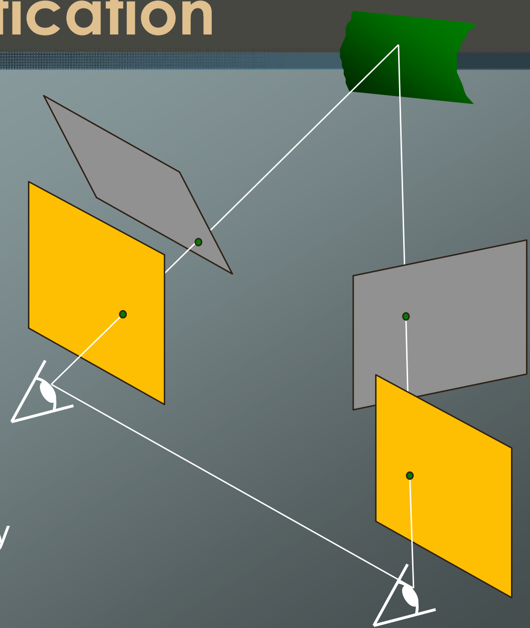


Two view (stereo) geometry

Geometric relations between two views are fully described by recovered 3X3 matrix **F**

Planar rectification

- Brings two views into standard stereo setup
  - reproject image planes onto common plane parallel to line between optical centers
- Notice, only focal point of camera really matters

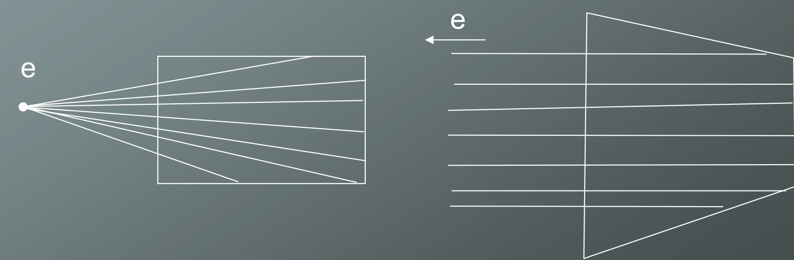


(Seitz)

Image pair rectification

- simplify stereo matching by warping the images
- Apply projective transformation so that epipolar lines correspond to horizontal scanlines
  - map epipole e to (1,0,0)
  - try to minimize image distortion
  - problem when epipole in (or close to) the image

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



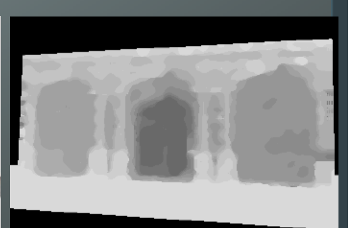
Extracting Structure

- The key aspect of epipolar geometry is its linear constraint on where a point in one image can be in the other
- By matching pixels (or features) along epipolar lines and measuring the **disparity** between them, we can construct a **depth map** (scene point depth is inversely proportional to disparity)

courtesy of P. Debevec

View 1

View 2



Computed depth map

Stereo matching

- What should be matched?
  - Pixels?
  - Collections of pixels?
  - Edges?
  - Objects?
- Correlation-based algorithms
  - Produce a **DENSE** set of correspondences
- Feature-based algorithms
  - Produce a **SPARSE** set of correspondences

Stereo matching: constraints

- Photometric constraint
- Epipolar constraint (through rectification)
- Ordering constraint
- Uniqueness constraint
- Disparity limit
- Disparity continuity constraint

Photometric constraint

- Photometric constraint
  - **Assumption**: Same world point has same intensity in both images
  - **Issues**: noise, specularity,…
  - ➔ matching standalone pixels won't work
- ➔ match windows centered around individual pixels.

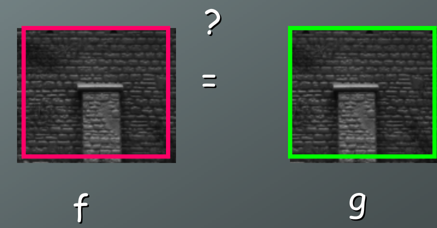


Image Normalization

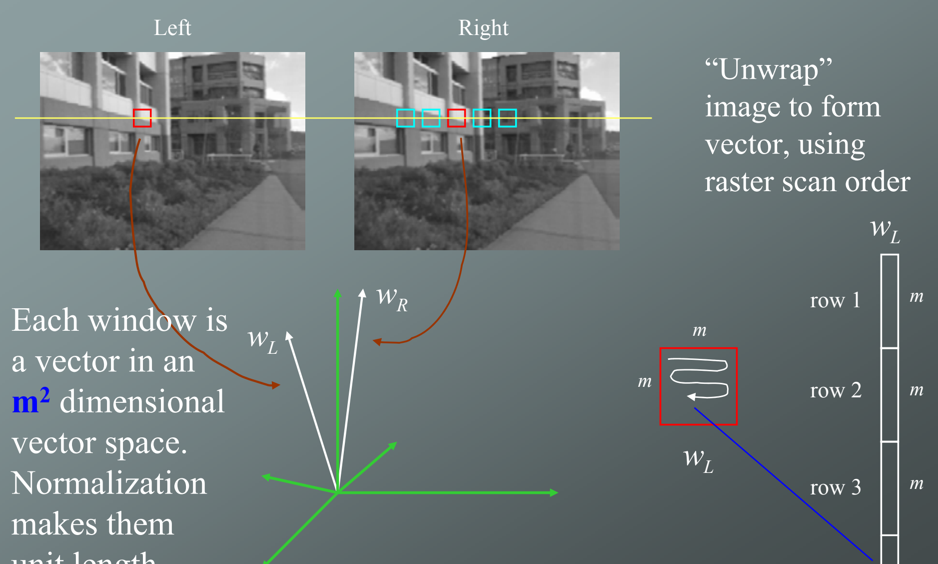
- Even when the cameras are identical models, there can be differences in gain and sensitivity.
- The cameras do not see exactly the same surfaces, so their overall light levels can differ.
- For these reasons and more, it is a good idea to normalize the pixels in each window:

$$\bar{I} = \frac{1}{|W_m(x,y)|} \sum_{(u,v)\in W_m(x,y)} I(u,v)$$ Average pixel

$$\|I\|_{W_m(x,y)} = \sqrt{\sum_{(u,v)\in W_m(x,y)} [I(u,v)]^2}$$ Window magnitude

$$\hat{I}(x,y) = \frac{I(x,y)-\bar{I}}{\|I-\bar{I}\|_{W_m(x,y)}}$$ Normalized pixel

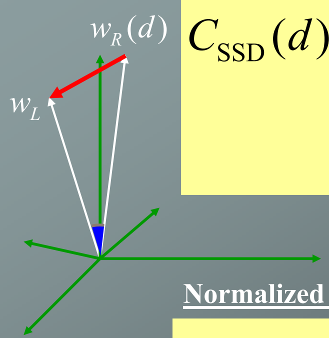
Images as Vectors



"Unwrap" image to form vector, using raster scan order

Each window is a vector in an m^2 dimensional vector space. Normalization makes them unit length.

Image Metrics



(Normalized) Sum of Squared Differences (SSD)

$$C_{\mathrm{SSD}}(d) = \sum_{(u,v)\in W_m(x,y)} [\hat{I}_L(u,v) - \hat{I}_R(u-d,v)]^2$$

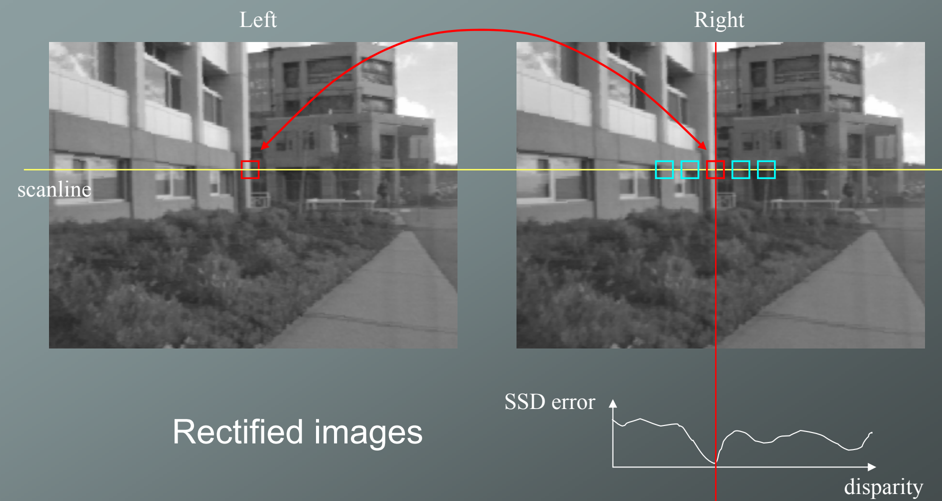
$$= \| w_L - w_R(d) \|^2$$

Normalized Correlation (NC)

$$C_{\mathrm{NC}}(d) = \sum_{(u,v)\in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v)$$

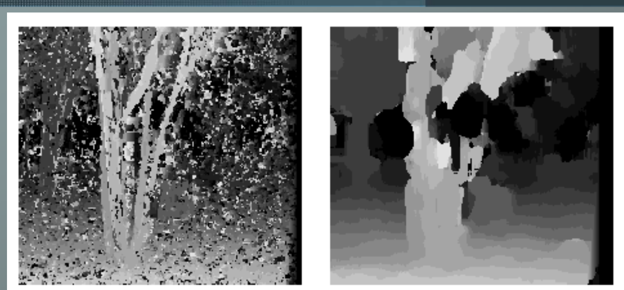
$$= w_L \cdot w_R(d) = \cos\theta$$

Correspondence via correlation



$$d^* = \arg\min_d \| w_L - w_R(d) \|^2 = \arg\max_d w_L \cdot w_R(d)$$

Window size



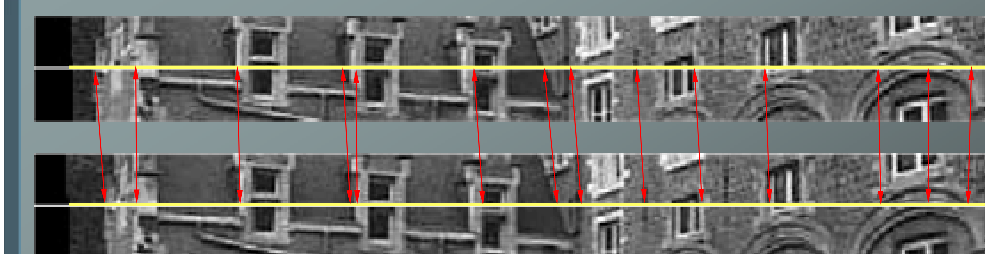
W = 3 W = 20

Better results with *adaptive window*

- T. Kanade and M. Okutomi, A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment,, Proc. International Conference on Robotics and Automation, 1991.
- D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. International Journal of Computer Vision, 28(2):155-174, July 1998

(Seitz)

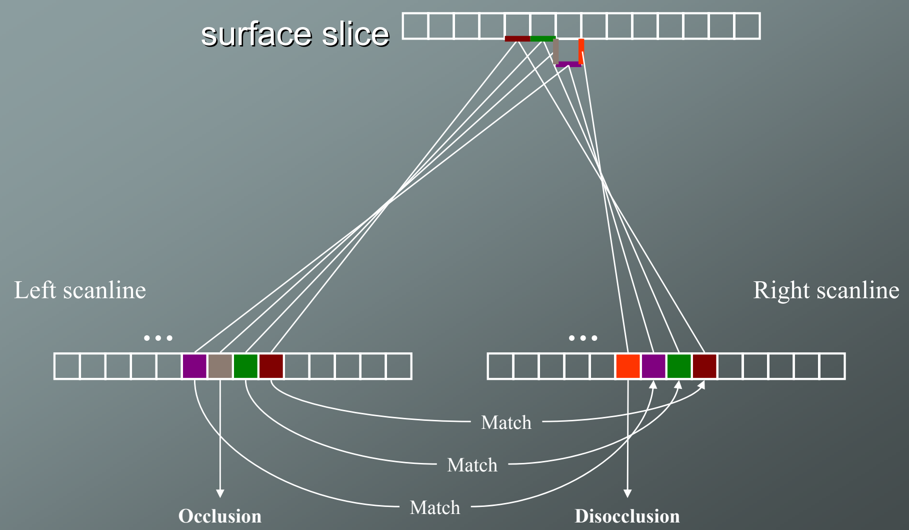
Epipolar constraint



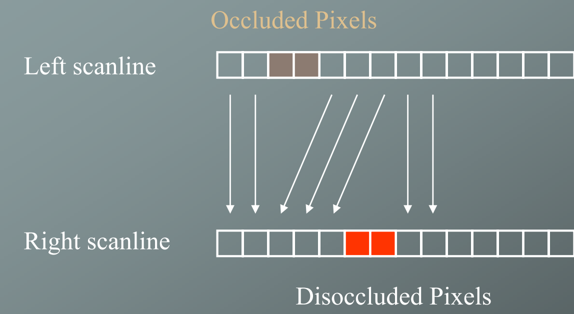
- Match pixels along corresponding epipolar lines (1D search)

Ordering constraint

- order of points in two images is usually the same



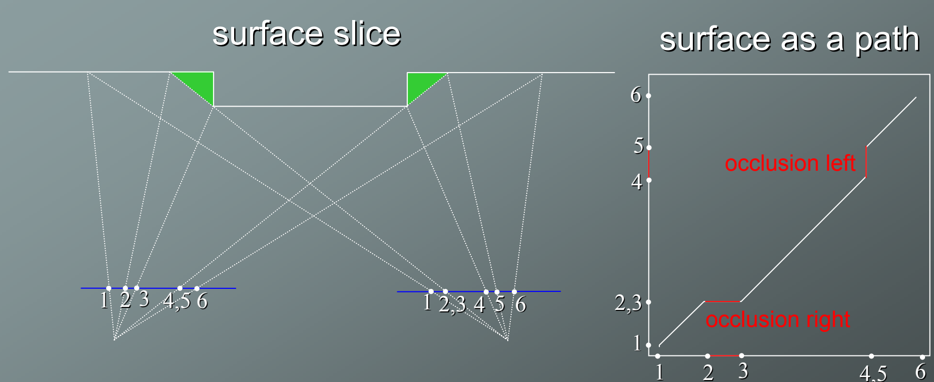
Ordering constraint



Three cases:

- Sequential – cost of match
- Occluded – cost of no match
- Disoccluded – cost of no match

Surface as a path

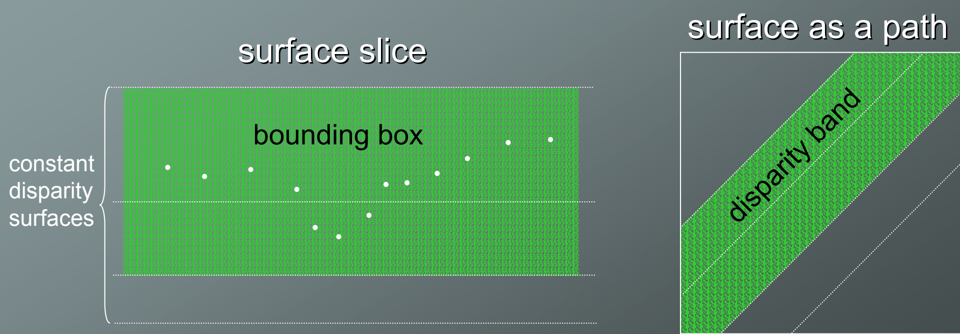


Uniqueness constraint

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

Disparity constraint

- Range of expected scene depths guides maximum possible disparity.
  - Search only a **segment** of epipolar line



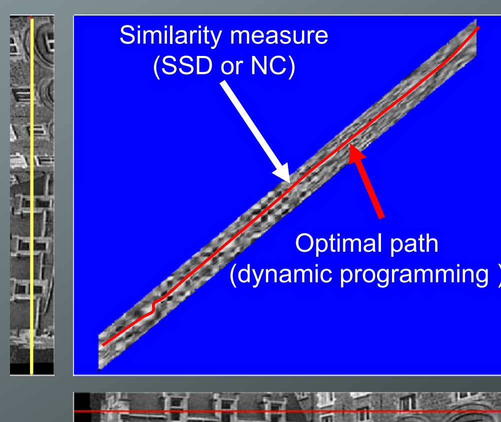
use reconsructed features to determine bounding box

Disparity continuity constraint

- Assume piecewise continuous surface
- ➔ piecewise continuous disparity
  - In general disparity changes continuously
  - discontinuities at occluding boundaries
  - Unfortunately, this makes the problem 2D again.
  - Solved with a host of graph algorithms, Markov Random Fields, Belief Propagation, ….

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)

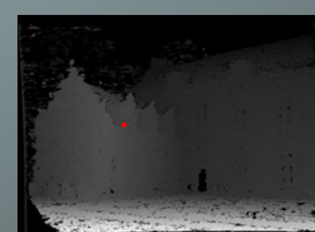


image I´(x´,y´)

(x´,y´)=(x+D(x,y),y)

Feature-based Methods

- Conceptually very similar to Correlation-based methods, but:
  - They only search for correspondences of a sparse set of image features.
  - Correspondences are given by the most similar feature pairs.
  - Similarity measure must be adapted to the type of feature used.

Features commonly used

- Corners
  - Similarity measured in terms of
    - surrounding gray values (SSD, NC)
    - location
- Edges, Lines
  - Similarity measured in terms of:
    - orientation
    - contrast
    - coordinates of edge or line's midpoint
    - length of line

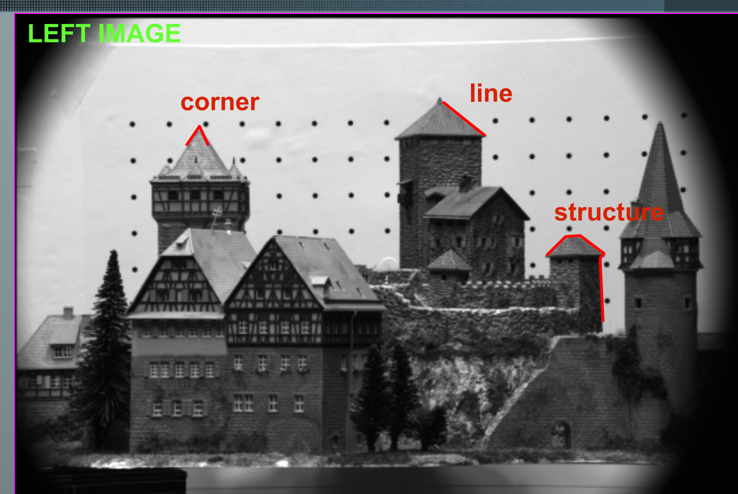
Example: Comparing lines

- l_l and l_r: line lengths
- θ_l and θ_r: line orientations
- (x_l,y_l) and (x_r,y_r): midpoints
- c_l and c_r: average contrast along lines
- $\omega_l\ \omega_\theta\ \omega_m\ \omega_c$: weights controlling influence

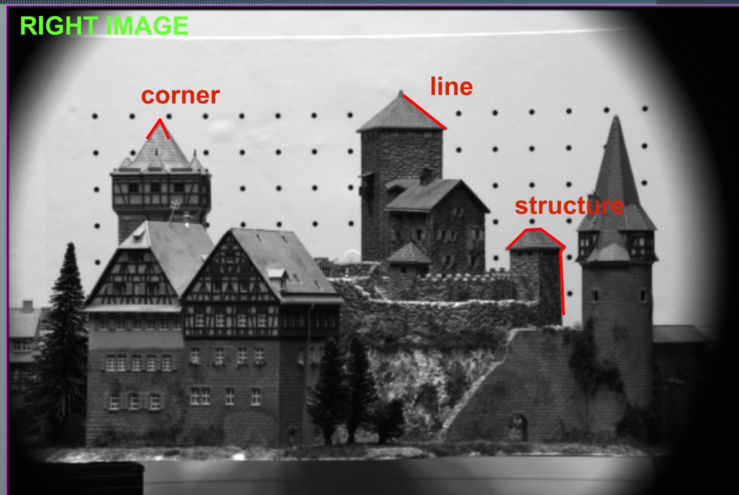
$$S = \frac{1}{\omega_l(l_l - l_r)^2 + \omega_\theta(\theta_l - \theta_r)^2 + \omega_m[(x_l - x_r)^2 + (y_l - y_r)^2] + \omega_c(c_l - c_r)^2}$$

The more similar the lines, the larger S is!

Correspondence By Features



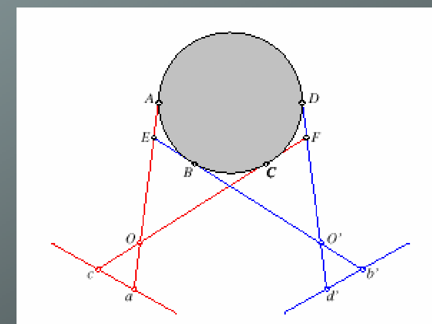
Correspondence By Features



- Search in the right image… the disparity **(dx, dy)** is the displacement when the similarity measure is maximum

Computing Correspondence

- Which method is better?
  - Edges tend to fail in dense texture (outdoors)
  - Correlation tends to fail in smooth featureless areas
- Both methods fail for smooth surfaces



Three geometric questions

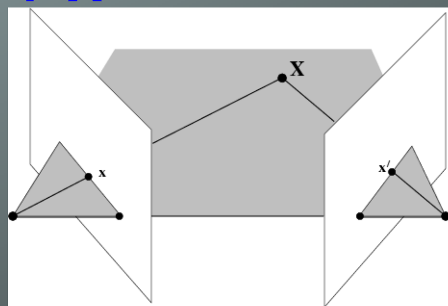
3. **Scene geometry (structure):** Given corresponding image points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ and cameras **P**, **P'**, what is the position of (their pre-image) **X** in space?

Terminology

- Point correspondences: $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$
- Original scene (pre-image): $\mathbf{X}_i$
- Projective, affine, similarity reconstruction =
  - reconstruction that is identical to original up to projective, affine, similarity transformation
  - *Literature:* Metric and Euclidean reconstruction = similarity reconstruction

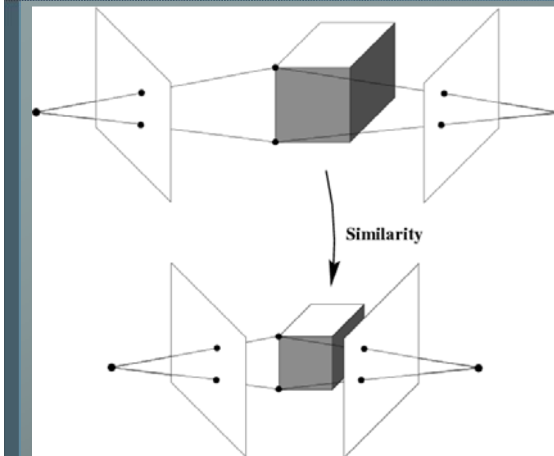
Computing Structure

- Recall that canonical camera matrices **P**, **P'** can be computed from fundamental matrix **F**
  - E.g. $\mathbf{P}=[\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}'=[[\mathbf{e}']_{\times}\mathbf{F}|\mathbf{e}']$,
- **Triangulation**: Back-projection of rays from image points **x**, **x'** to 3-D point of intersection **X** such that $\mathbf{x}=\mathbf{PX}$ and $\mathbf{x}'=\mathbf{P}'\mathbf{X}$



from Hartley & Zisserman

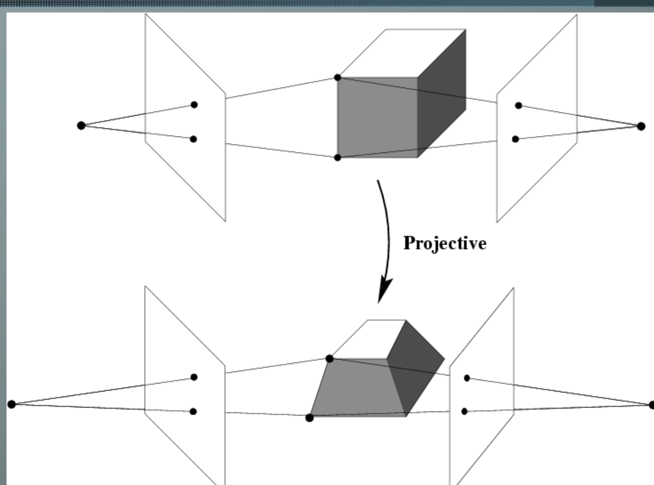
Reconstruction ambiguity: similarity



$$x_i = PX_i = \left(PH_S^{-1}\right)\left(H_S X_i\right)$$

$$PH_S^{-1} = K[R\,|\,t]\begin{bmatrix} R'^T & -R'^T t' \\ 0 & \lambda \end{bmatrix} = K[RR'^T\,|\,-RR'^T t' + \lambda t]$$

Reconstruction ambiguity: projective



$$x_i = PX_i = \left(PH_P^{-1}\right)\left(H_P X_i\right)$$

The projective reconstruction theorem

If a <u>set of point correspondences</u> in two views <u>determine the fundamental matrix uniquely</u>, then the <u>scene and cameras</u> may be reconstructed from these correspondences alone, and any two such reconstructions from these correspondences are **<u>projectively equivalent</u>**

$$\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \qquad \left(\mathbf{P}_1, \mathbf{P}_1', \{\mathbf{X}_{1i}\}\right) \quad \left(\mathbf{P}_2, \mathbf{P}_2', \{\mathbf{X}_{2i}\}\right)$$

$$\mathbf{P}_2 = \mathbf{P}_1\mathbf{H}^{-1} \qquad \mathbf{P}_2' = \mathbf{P}_1'\mathbf{H}^{-1} \qquad \mathbf{X}_2 = \mathbf{H}\mathbf{X}_1 \qquad \left(\text{except}: \mathbf{F}\mathbf{x}_i = \mathbf{x}_i'\mathbf{F} = \mathbf{0}\right)$$

$$\mathbf{P}_2\left(\mathbf{H}\mathbf{X}_{1i}\right) = \mathbf{P}_1\mathbf{H}^{-1}\mathbf{H}\mathbf{X}_{1i} = \mathbf{P}_1\mathbf{X}_{1i} = \mathbf{x}_i = \mathbf{P}_2\mathbf{X}_{2i}$$

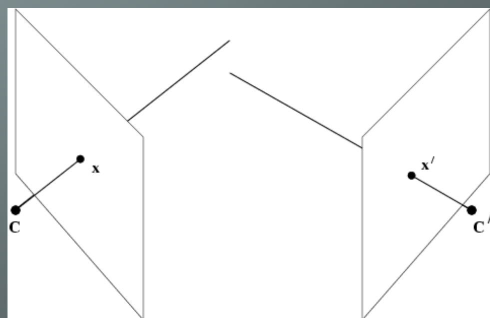
$\Rightarrow$ along same ray of $\mathbf{P}_2$, idem for $\mathbf{P}'_2$

two possibilities: $\mathbf{X}_{2i}=\mathbf{H}\mathbf{X}_{1i}$, or points along baseline

<u>key result:</u> allows reconstruction from pair of uncalibrated images

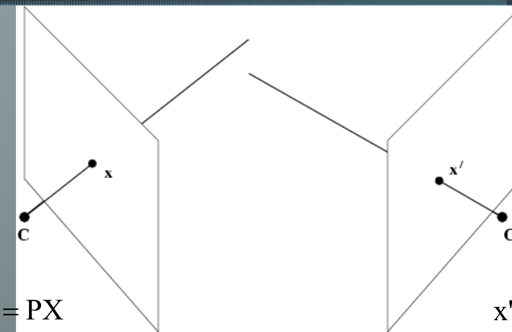
Triangulation: Issues

- Errors in
  - points **x**, **x'** & **F** such that $\mathbf{x'^T F x = 0}$ or
  - **X** such that **x=PX** and **x'=P'X**
- This means that rays are *skew* — they don't intersect

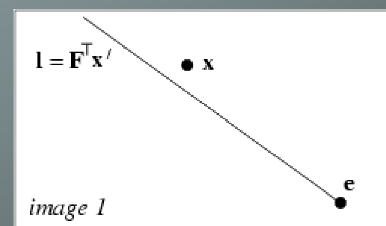


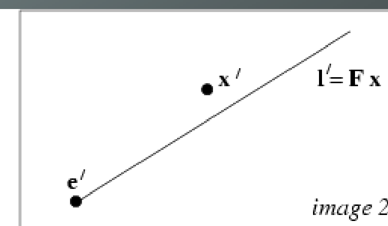
from Hartley & Zisserman

Point reconstruction



$x = PX$ $\qquad$ $x' = P'X$





Linear triangulation

$$x = PX \qquad x' = P'X$$

$$x \times P'X = 0$$

$$x(p^{3T}X) - (p^{1T}X) = 0$$
$$y(p^{3T}X) - (p^{2T}X) = 0$$
$$x(p^{2T}X) - y(p^{1T}X) = 0$$

$$AX = 0$$

$$A = \begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p'^{3T} - p'^{1T} \\ y'p'^{3T} - p'^{2T} \end{bmatrix}$$

homogeneous $\|X\| = 1$

inhomogeneous $(X, Y, Z, 1)$

invariance? $(AH^{-1})(HX) = e$

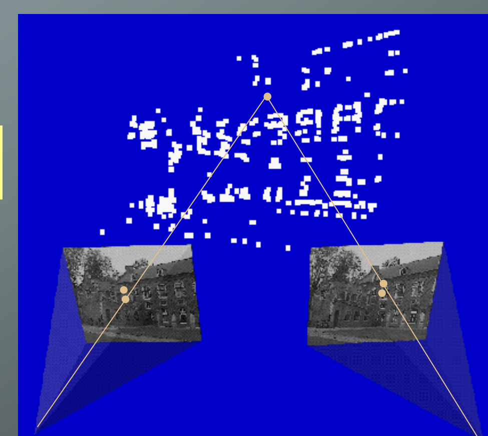
algebraic error yes, constraint no (except for affine)

Geometric error

- Reconstruct matches in projective frame by minimizing the reprojection error

$$d(x_1, P_1X)^2 + d(x_2, P_2X)^2$$

- Non-iterative optimal solution (see Hartley&Sturm,CVIU´97)

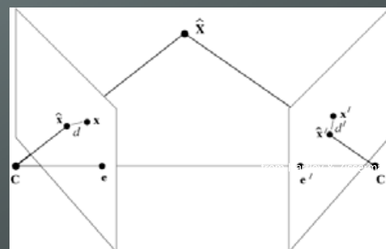


Optimal Projective-Invariant Triangulation: Reprojection Error

- Pick $\hat{\mathbf{X}}$ that exactly satisfies camera geometry so that $\hat{\mathbf{x}} = \mathbf{P}\hat{\mathbf{X}}$ and $\hat{\mathbf{x}}' = \mathbf{P}'\hat{\mathbf{X}}$, and which minimizes

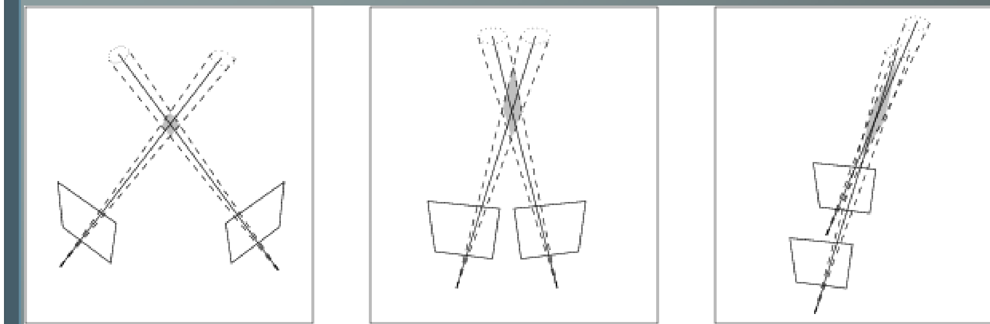
$$d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$$

- Can use as error function for nonlinear minimization on two views
  - Polynomial solution exists



Covariance of Structure Recovery

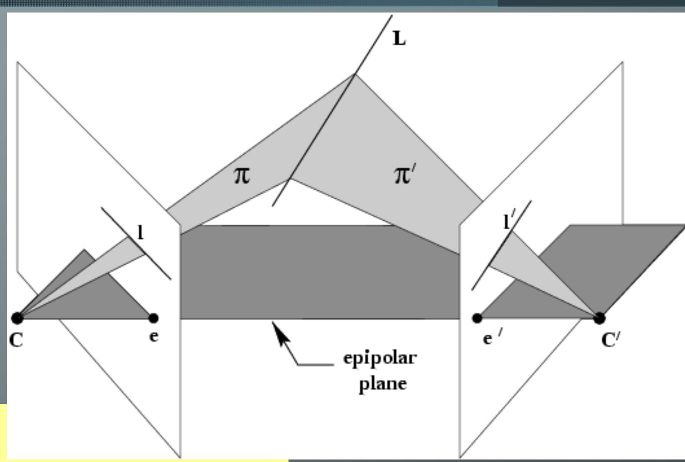
- Bigger angle between rays → Less uncertainty



from Hartley & Zisserman

- Can't triangulate points on baseline (epipoles) because rays intersect along entire length

Line reconstruction



$$\mathbf{L} = \begin{bmatrix} \mathbf{l}^T \mathbf{P} \\ \mathbf{l}'^T \mathbf{P}' \end{bmatrix}$$

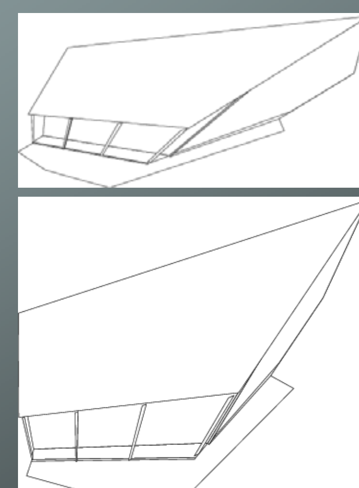
$\mathbf{L}\mathbf{X} = \mathbf{0}$ for a point $\mathbf{X}$ on line $\mathbf{L}$

doesn't work for epipolar plane

Projective Reconstruction Ambiguity

from Hartley & Zisserman

Two views from which **F** and hence **P**, **P'** are computed



Reconstructions related by a 4 x 4 projection **H**

Hierarchy of transformations

Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths).
Affine 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at infinity, $\mathbf{l}_\infty$.
Metric/ Similarity 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratio of lengths, angle. The circular points, **I**, **J** (see section 1.7.3).
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

Less ambiguity

from Hartley & Zisserman

Properties of transformations (2-D)

Stratified Reconstruction

- Idea: Try to **upgrade** reconstruction to differ from the truth by a less ambiguous transformation
- Use additional constraints imposed by:
  - Scene:
    - known 3-D points (no 4 coplanar) ➔ Euclidean reconstruction
    - Identify parallel, orthogonal lines in scene
  - Camera calibration: Known **K**, **K'** ➔ Metric/similarity reconstruction
  - Camera motion: Known **R**, **t**

Projective ➔ Affine Upgrade

- Identify plane at infinity π_∞ (in the "true" coordinate frame, $\pi_\infty = (0, 0, 0, 1)^T$)
  - E.g., intersection points of three sets of parallel lines define a plane
  - E.g., if one camera is known to be affine

Projective ➔ Affine Upgrade

- Then apply 4 x 4 transformation:

$$(\mathrm{P}, \mathrm{P}', \{\mathrm{X}_i\})$$

$$\pi_\infty = (A, B, C, D)^T \mapsto (0,0,0,1)^T$$

$$\mathrm{H}^{-T}\pi_\infty = (0,0,0,1)^T$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} \mid \mathbf{0} \\ \boldsymbol{\pi}_\infty^T \end{bmatrix}$$

- This is the 3-D analog of affine image rectification via the line at infinity $\mathbf{l}_\infty$
- Things that can be computed/constructed with only affine ambiguity:
  - Midpoint of two points
  - Centroid of group of points
  - Lines parallel to other lines, planes

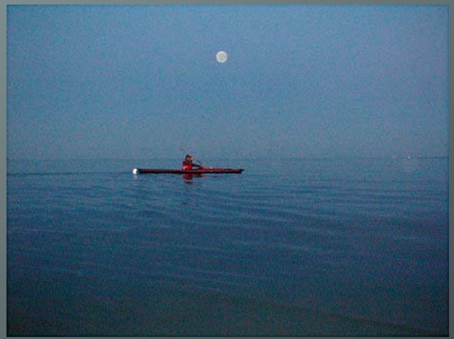
Translational motion

- points at infinity are fixed for a pure translation
  - reconstruction of $\mathbf{x}_i \leftrightarrow \mathbf{x'}_i$ is on π_∞

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

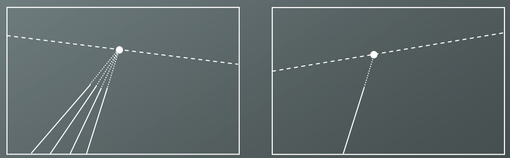
$$P = [I \mid 0]$$

$$P' = [I \mid e']$$

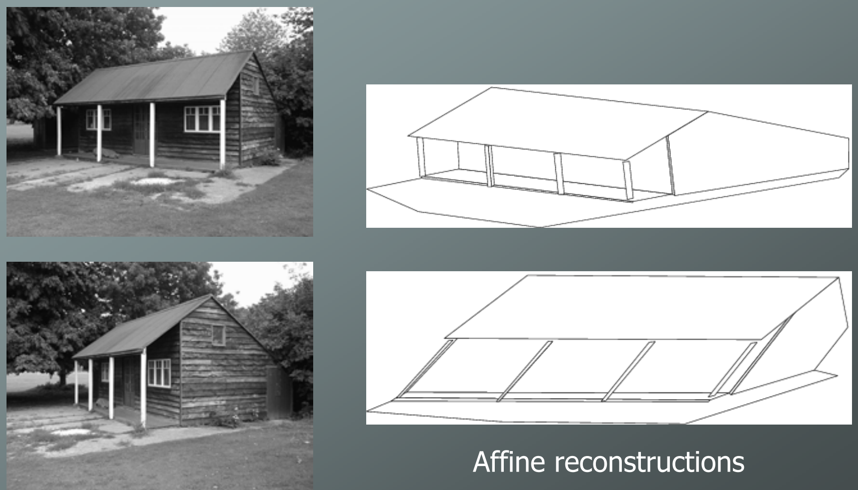


Scene constraints

- **Parallel lines**
  - parallel lines intersect at infinity
  - reconstruction of corresponding vanishing point yields point on plane at infinity
  - 3 sets of parallel lines allow to uniquely determine π_∞
- **Remarks**:
  - in presence of noise determining the intersection of parallel lines is a delicate problem
  - obtaining vanishing point in one image can be sufficient



Affine Reconstruction Ambiguity



Affine reconstructions

Affine → Metric Upgrade

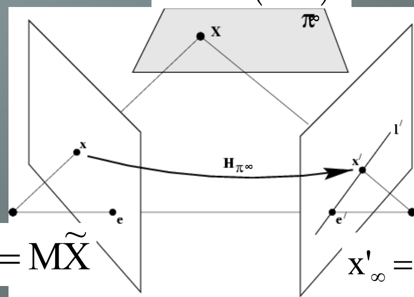
- Identify **absolute conic** Ω_∞ on π_∞ via image of absolute conic (IAC) ω
  - From scene
    - E.g., orthogonal lines
  - From known camera calibration
    - Completely constrained: $\omega = K^{-T} K^{-1}$
    - Partially constrained:
      - Zero skew
      - Square pixels
  - Same camera took all images (e.g. moving camera)

The infinity homography

$$P=[M\,|\,m] \quad P'=[M'\,|\,m']$$

$$H_\infty = M'M^{-1}$$

$$X_\infty = (\tilde{X},0)^T$$



$$x_\infty = M\tilde{X} \qquad x'_\infty = M'\tilde{X}$$

Unchanged under affine transformations

$$P=[M\,|\,m]\begin{bmatrix} A & a \\ 0 & 1 \end{bmatrix}=[MA\,|\,Ma+m]$$

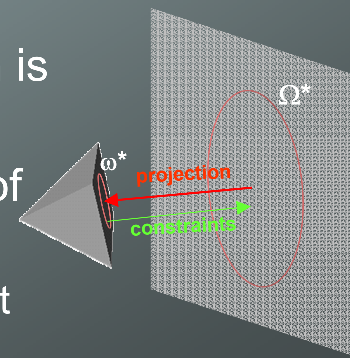
$$H_\infty = M'AA^{-1}M^{-1}$$

affine reconstruction:

$$P=[I\,|\,0] \quad P=[H_\infty\,|\,e]$$

Affine to metric

- Identify absolute conic Ω_∞
- transform so that $\Omega_\infty : X^2+Y^2+Z^2=0, \text{on } \pi_\infty$
- then projective transformation relating original and reconstruction is a similarity transformation
- in practice, find ω (image of absolute conic)
  - ω back-projects to cone that intersects π_∞ in Ω_∞



Affine to metric

- Given

$$P=[M\,|\,m] \qquad \omega$$

- possible transformation from affine to metric is

$$H=\begin{bmatrix} A^{-1} & 0 \\ 0 & 1 \end{bmatrix} \qquad AA^T=\left(M^T\omega M\right)^{-1}$$

(cholesky factorisation)

Same camera for all images

- Same intrinsics ➔ same image of the absolute conic
  - For example: moving camera
- Given sufficient images there is in general only one conic that projects to the same image in all images, i.e. the absolute conic
  - This approach is called *self-calibration*
- Transfer of IAC:

$$\omega' = H_\infty^{-T}\,\omega\, H_\infty^{-1}$$

Direct metric reconstruction using ω

- Approach 1
  - calibrated reconstruction

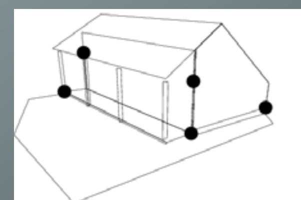
$$\omega = K^{-T}K^{-1} \Rightarrow K$$

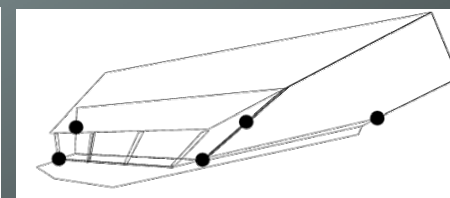
- Approach 2
  - compute projective reconstruction
  - back-project ω from both images
  - intersection defines Ω_∞ and its support plane π_∞

Direct metric reconstruction using ground truth

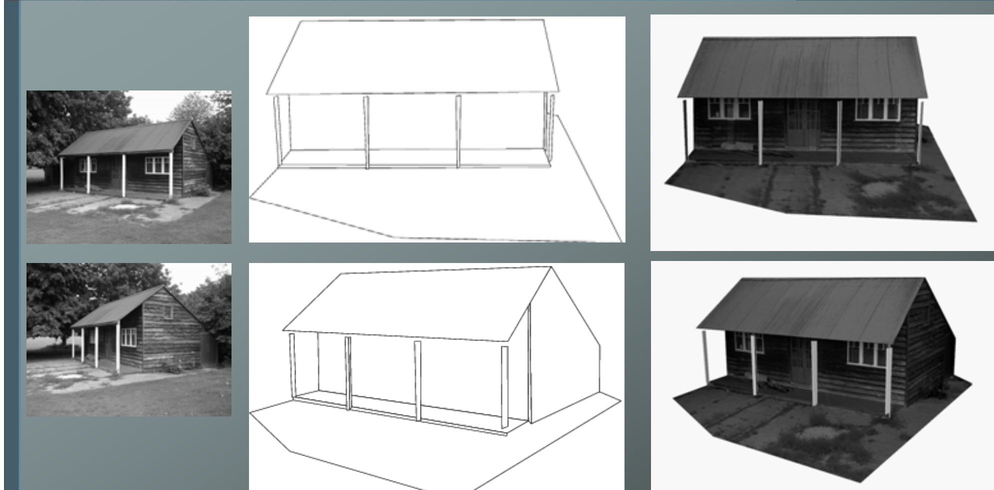
- Use control points X_{Ei} with know coordinates to go directly from projective to metric
  - Need 5 points (no 4 coplanar)
  - 2 linear eq. in H^{-1} per view, 3 for two views)

$$X_{Ei} = HX_i \qquad x_i = PH^{-1}X_{Ei}$$





Metric Reconstruction Example



from Hartley & Zisserman

Original views Synthesized views of reconstruction

Only overall scale ambiguity remains—i.e., what are units of length?

Metric Reconstruction

- **Objective:** Given two uncalibrated images compute **(PM,P‘M,{XM$_i$})** (i.e. within similarity of original scene and cameras)
- **Algorithm**

1. Compute projective reconstruction **(P,P‘,{X$_i$})**
   1. Compute **F** from $x_i \leftrightarrow x'_i$
   2. Compute **P**, **P‘** from **F**
   3. Triangulate X_i from $x_i \leftrightarrow x'_i$
2. Rectify reconstruction from projective to metric
   1. Direct method: compute **H** from control points

      $$X_{Ei} = HX_i \qquad P_M = PH^{-1} \qquad P'_M = P'H^{-1} \qquad X_{Mi} = HX_i$$

   2. Stratified method:
      1. Affine reconstruction: compute π_∞

         $$H = \begin{bmatrix} I \mid 0 \\ \pi_\infty \end{bmatrix}$$

      2. Metric reconstruction: compute IAC ω

         $$H = \begin{bmatrix} A^{-1} & 0 \\ 0 & 1 \end{bmatrix} \qquad AA^T = \left(M^T \omega M\right)^{-1}$$

Reconstruction summary

Image information provided	View relations and projective objects	3-space objects	reconstruction ambiguity
point correspondences	F		projective
point correspondences including vanishing points	F, H_∞	π_∞	affine
Points correspondences and internal camera calibration	F, H_∞ ω, ω'	π_∞ Ω_∞	metric