3D Recognition from stereo vision : correspondence & triangulation

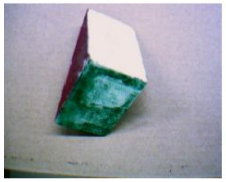
Advanced Vision – Lecture 9

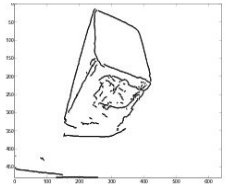
Toby Breckon
toby.breckon@ed.ac.uk

Computer Vision Lab.
Institute for Perception, Action & Behaviour
School of Informatics

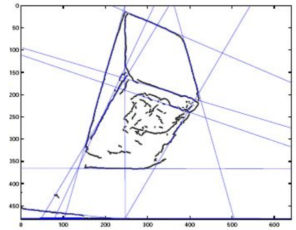
From Feature identification ....

- **Canny Edge Detection**
  - **Result:** edge fragments

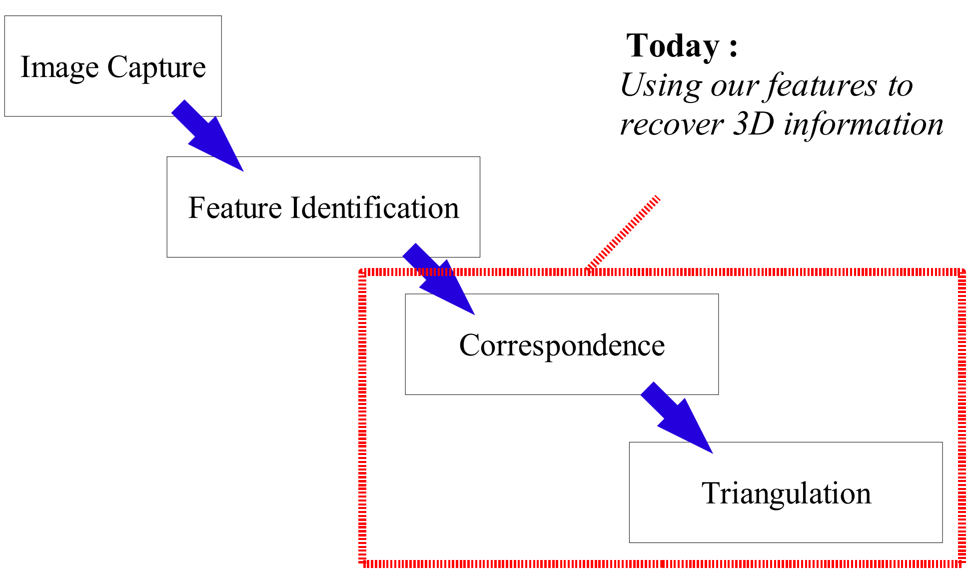




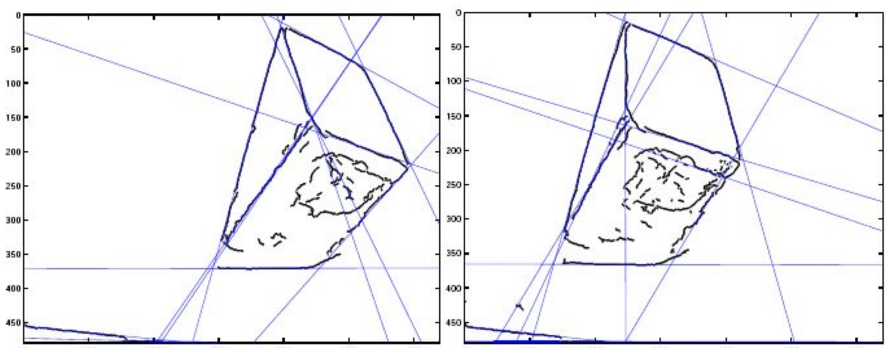
- **RANSAC algorithm**
  - parametric line descriptions that correspond to feature in image
  - N.B. infinite line model (i.e. vector direction)



Stereo Vision - Overview



RANSAC : line descriptions

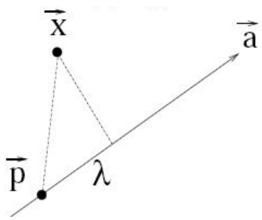


- RANSAC : Infinite unbounded lines – geometric description
  - point $\vec{p}$ and direction $\vec{a}$ - **position** (in 2D)
- For correspondence matching finite description useful
  - bounds on length - **shape**

Preprocessing : line segments

- **Requirement** : approx. start and end point of line segment.
- **Input** : geometric descriptions & set of points { $\vec{x}_i$ } for each segment identified by RANSAC.

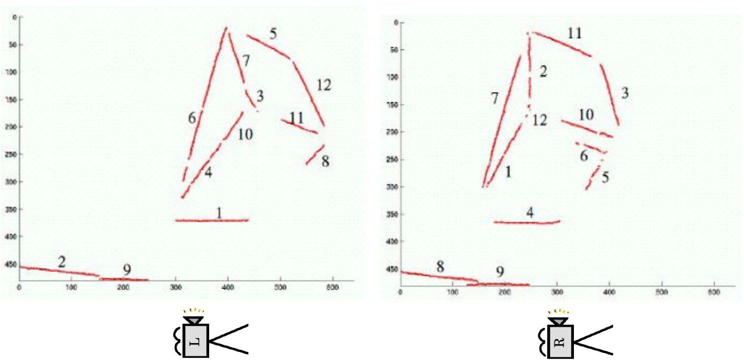
1. Project points { $\vec{x}_i$ } onto ideal line through point $\vec{p}$ with direction $\vec{a}$: $\lambda_i = (\vec{x}_i - \vec{p}) \cdot \vec{a}$. Projected point is $\vec{p} + \lambda_i \vec{a}$



2. Remove projected points not having N neighbour points within D pixels distance - **outlier removal** (set N and D to be similar; e.g. 43 and 45)
3. Endpoints of line segment are now given by smallest / largest λ_i

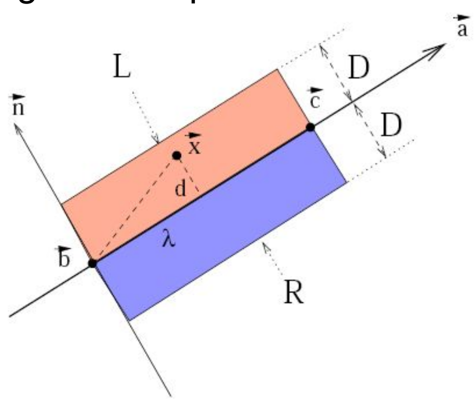
Preprocessing : line segments

- **Identified line segments**
  - geometric description (from RANSAC)
  - length description (from preprocessing)



Preprocessing : line description 1

- Given line segment endpoints $\vec{b}$ and $\vec{c}$ with direction $\vec{a}$ = (u,v)



- Compute perpendicular : $\vec{n}$ = (-v, u) **(orientation)**
- Estimate midpoint : $\vec{m}$ = ($\vec{b}$ + $\vec{c}$) / 2 **(position)**
- Estimate length : || $\vec{b} - \vec{c}$ || **(shape)**

Preprocessing : line description 2

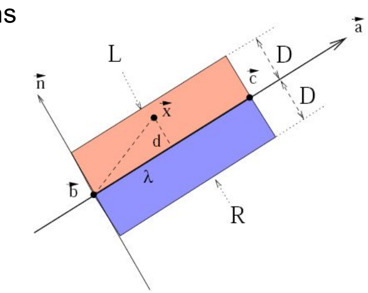
- Estimate edge contrast of line
  - consider area within distance D of line in regions L & R
  - Divide original images pixels into L/R regions

For any pixel $\vec{x}$

If $0 \leq (\vec{x} - \vec{b}) \cdot \vec{a} \leq \| \vec{b} - \vec{c} \|$

If $0 < (\vec{x} - \vec{b}) \cdot \vec{n} \leq D$, put $\vec{x}$ in L

If $-D \leq (\vec{x} - \vec{b}) \cdot \vec{n} < 0$, put $\vec{x}$ in R



  - Estimate contrast: g = ((average brightness of pixels in L) - (average brightness of pixels in R))
  - **Overall final line description:**
    - **direction** $\vec{a}$, **midpoint** $\vec{m}$, **length** & **contrast** g

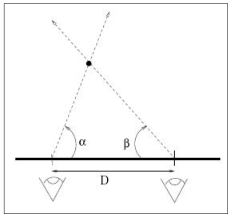
Corresponding Lines : Ground Truth

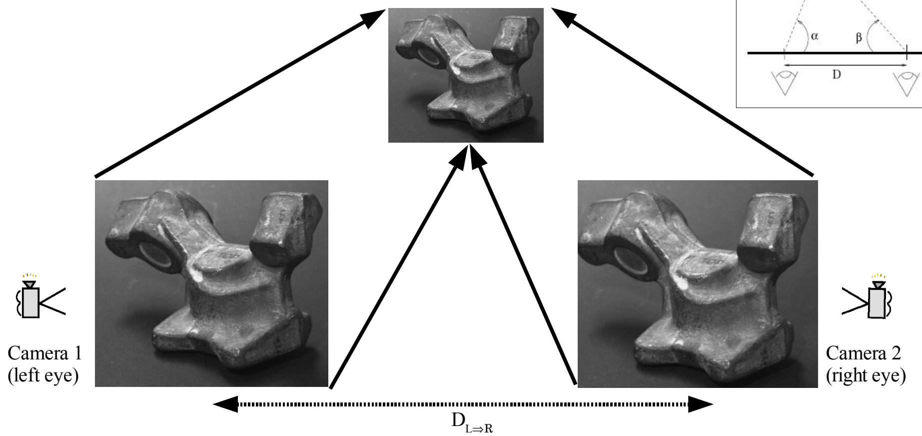
Edges	L dir	R dir	L len	R len
L1:R4	(0.00,-1.00)	(0.00,-1.00)	137	125
L5:R11	(0.45,0.89)	(0.38,0.93)	90	119
L6:R7	(0.96,-0.28)	(0.96,-0.29)	291	251
L7:R2	(0.95,0.32)	(1.00,0.00)	93	140
L8:R5	(0.70,-0.71)	(0.86,-0.50)	49	61
L10:R12	(0.81,-0.59)	(0.86,-0.52)	87	82
L11:R10	(-0.33,-0.94)	(-0.29,-0.96)	72	101
L12:R3	(0.89,0.45)	(0.96,0.28)	129	115

Edges	L mid	R mid	L con	R con
L1:R4	(370,369)	(364,242)	-91	-37
L5:R11	(55,477)	(40,309)	28	18
L6:R7	(158,355)	(181,194)	44	55
L7:R2	(74,416)	(90,246)	-117	-44
L8:R5	(250,567)	(278,371)	-66	-31
L10:R12	(208,402)	(205,221)	38	55
L11:R10	(199,536)	(195,358)	141	113
L12:R3	(140,554)	(132,400)	47	23

- Comparison of properties between lines that correspond in reality – **Ground Truth**
- But we need to find a correspondence automatically
  - we require **stereo correspondence**

Binocular Stereo

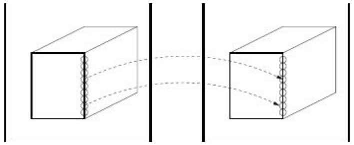




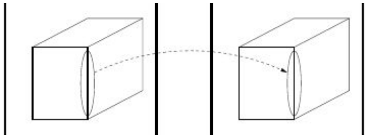
Key principle : **triangulation** between **corresponding** image **features**

Image Features - edges

- Correspondence = matching : but ... what to match ?
  - **Edge Fragments** from edge detector
    - numerous, noisy, limited description



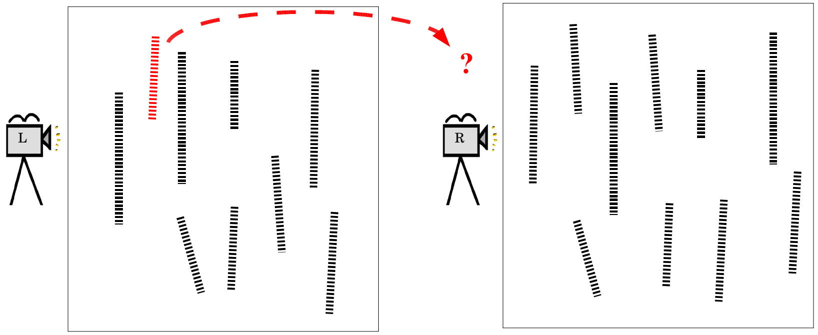
  - **Edge Lines** from RANSAC (post-processed edge fragments)
    - larger structures, less noise, rich description



- Which ?
  - Lines : few and larger (easier) but harder to obtain and less dependable (noise, occlusion) – **used here**
  - Fragments : thought to be used by human visual system

Stereo Correspondence Problem

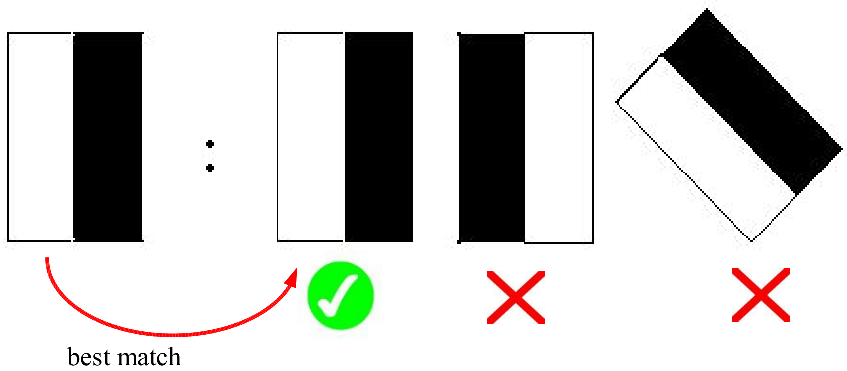
Q: For a given feature in the left, what is the correct correspondence?



- Different pairing result in different 3D results
  - inconsistent correspondence = inconsistent 3D (!)
  - Key problem in all stereo vision approaches

Correspondence : Matching Constraints (local)

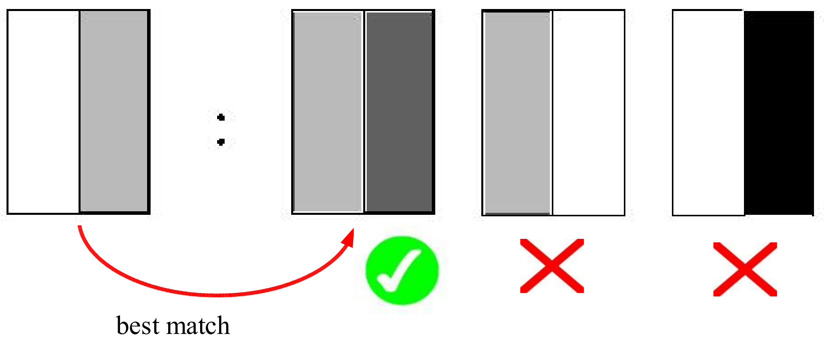
- **Orientation**
  - *edge direction* $\vec{a}$



Correspondence : Matching Constraints (local)

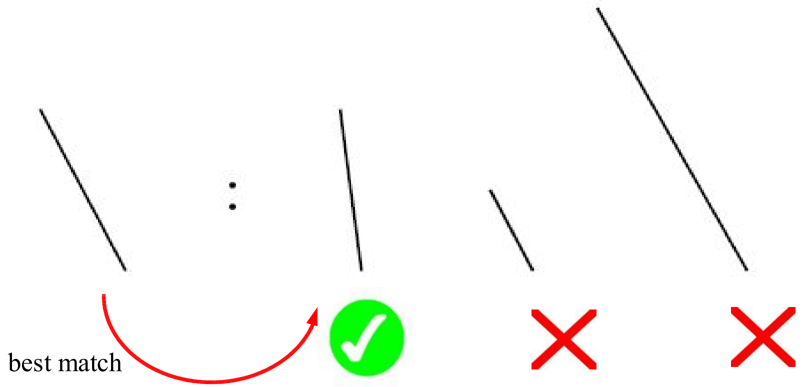
- **Colour**
  - *edge contrast g*

(nearly the same across the edge, find closest match)

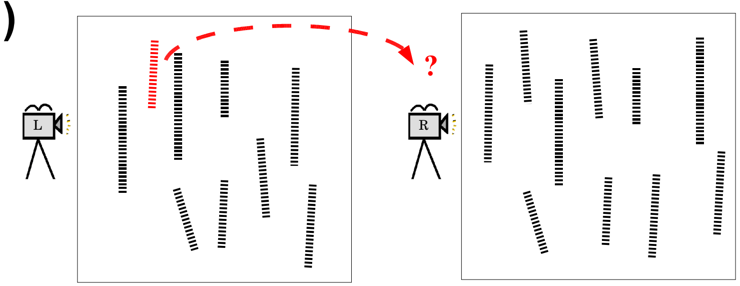


Correspondence : Matching Constraints (local)

- **Shape**
  - *edge length*



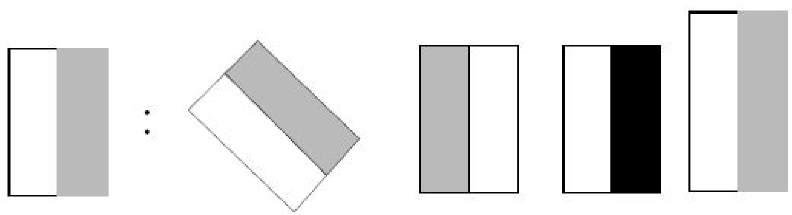
Correspondence : Matching Constraints (global)



- **Smoothness**
  - matched features have consistent depth to neighbours
- **Uniqueness**
  - one to one, left to right, correspondence
    - 0 : no match (due to occlusion in one image)
    - 1 : normal 1-to-1 matching
    - 2+ : difficult cases (wires, vines, etc. from coincidental)

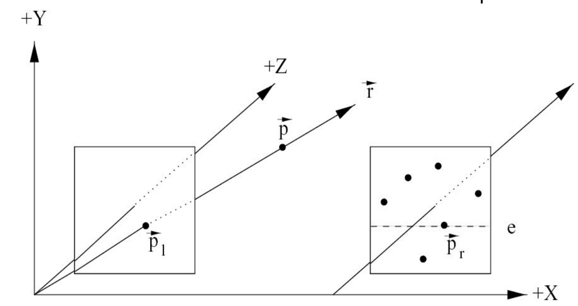
Lecture Problem

- Which stereo correspondence constraint would you use to reject these matches ?



Correspondence : Epipolar Geometry

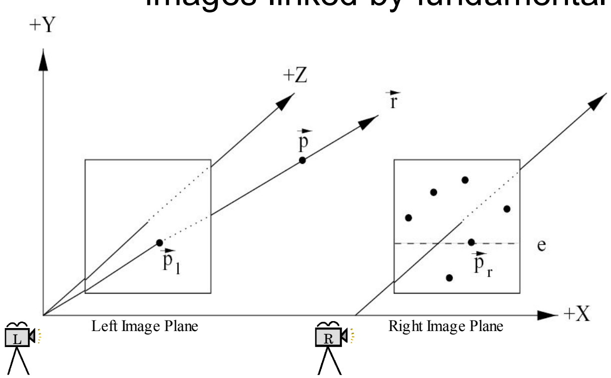
- Feature $\vec{p}_l$ in the left image lies on a ray $\vec{r}$ in space
- $\vec{r}$ projects to an epipolar line *e* in the right image
  - along which the matching feature $\vec{p}_r$ must lie



  - If the images are "rectified", then epipolar line is the image row
    - i.e. cameras (or images via post-processing) both perfectly axis aligned

Correspondence : Epipolar Geometry

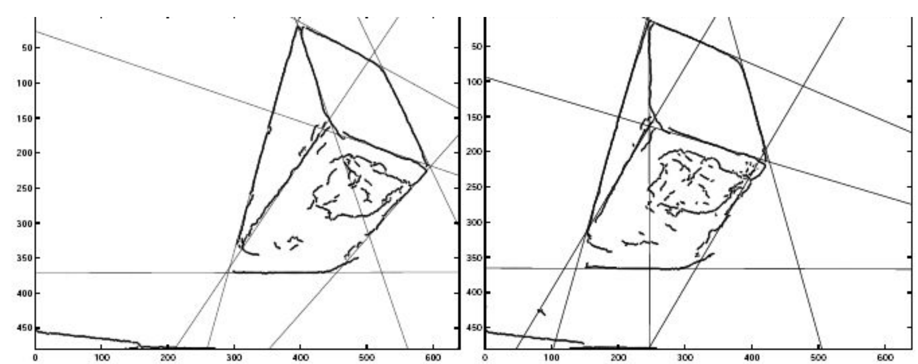
- Constraining L->R Correspondence
  - reduces 2D search to 1D
  - images linked by fundamental matrix, *F.*



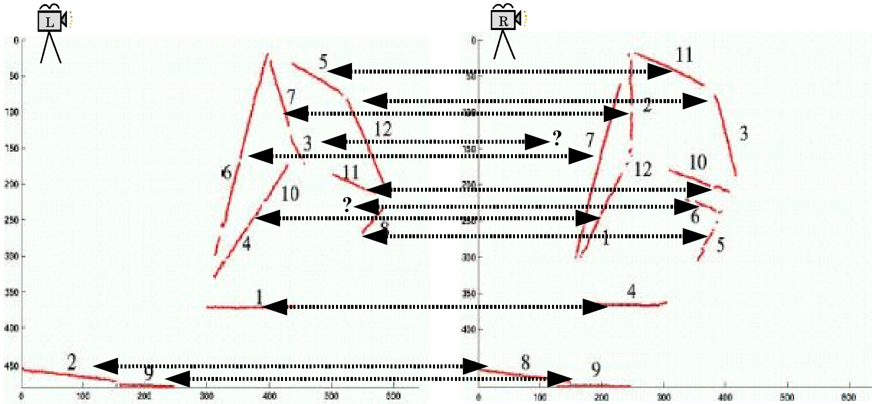
- For matched points $\vec{p}_l F \vec{p}_r = 0$.
- *F* generally derived from prior calibration routine (with pre-known target).
- Points are homogeneous
- *F* is 3x3

  - Match for point $\vec{p}_l$ on ray $\vec{r}$ (left) must lie on epipolar line *e* (right).

Stereo Matching Results



Stereo Matching Results

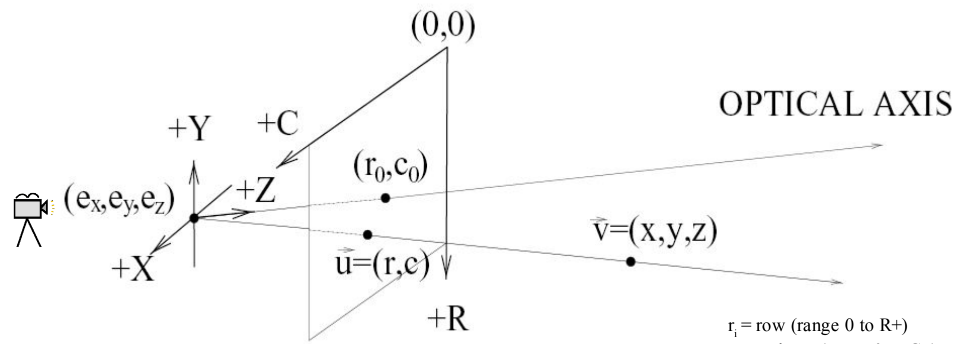


- Maximally consistent set of correspondences
- Based on local, global and eipolar constraint

  (interpretation tree or similar, wildcard matches)

Triangulation : Image Projection

- Projection of object point $\vec{v} = (x, y, z, 1)'$ onto image plane, image point $\vec{u}_i = (r_i, c_i, 1)'$. (3D world to 2D image)
- $\lambda \vec{u}_i = P_i \vec{v}$ (λ = homogeneous scaling factor)
  - where $\mathbf{P_i}$ **= camera projection matrix** for i ∈ {L,R}



r_i = row (range 0 to R+)
c_i = column (range 0 to C+)

Triangulation: Projection Matrix

- $\mathbf{P_i = K_i R_i [I \mid -\vec{e}_i]}$, i ε {L,R}
  - $\mathbf{R_i}$: camera orientation
  - $\vec{\mathbf{e}}_i = (e_{xi}, e_{yi}, e_{zi})$: camera centre (in world)
  - $\mathbf{K_i} = \begin{bmatrix} f_i m_{ri} & s_i & r_{0i} \\ 0 & f_i m_{ci} & c_{0i} \\ 0 & 0 & 1 \end{bmatrix}$
  - $\mathbf{f_i}$ = camera focal length (mm)
  - $\mathbf{m_{ri}, m_{ci}}$ = row/column pixel/mm scale
  - $\mathbf{r_{0i}, c_{0i}}$ = optical axis / image plane intersection
  - $\mathbf{s_i}$ = skew factor

Lab Cameras (example)

$R_L = R_R = I$

$\vec{e}_L = (0,0,0)$ $\vec{e}_R = (D_{L\Rightarrow R},0,0)$

(camera separation of d)

$K_L = K_R$

(L and R = identical camera(s))

$f_i m_{ci} = f_i m_{ri} = 832.5$

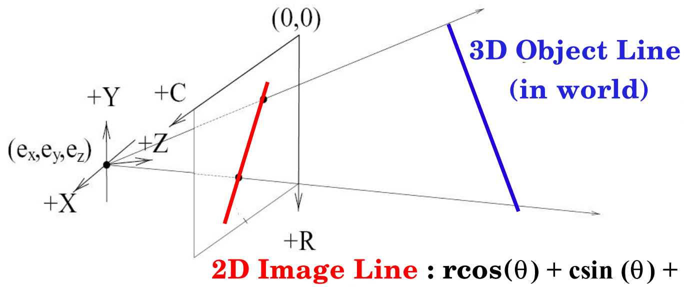
r_{0i}, c_{0i} = (height/2, width/2)

$s_i = 0$ (typical for CCD)

Triangulation : 3D line calculation

- **Aim :** recovery of 3D line position
- **Assume :** line successfully matched in L & R images
- Two key stages
  1. **Compute 3D plane** that goes through 2D line in image and camera origin (for both L and R cameras)
  2. **Compute intersection of these two 3D planes** from two cameras
     - **intersection of two 3D planes = 3D line**

Triangulation – Step 1: Find 3D planes



$$P=\begin{bmatrix} 0 & -fm_r & r_0 \\ fm_c & 0 & c_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Slightly different due t axis labelling and image axis directions

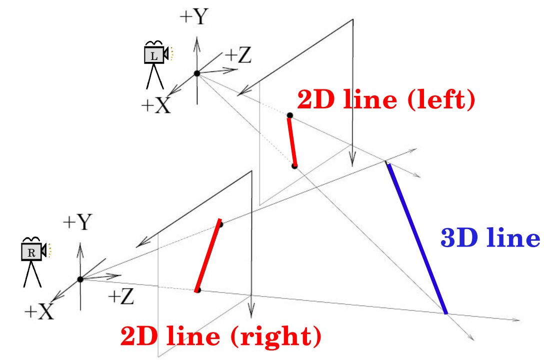
- **Find 3D plane for each image L and R : $\vec{n}\,.\,(x, y, z)' + d = 0$**
  - $(\lambda r, \lambda c, \lambda)' = P(x, y, z, 1)'$ from projection relation. *Solving we get:*

$$r = -fm_r\frac{y-e_y}{z-e_z} + r_0 \qquad c = fm_c\frac{x-e_x}{z-e_z} + c_0$$

  - ***3D plane :*** $\vec{n} = (fm_c\sin(\theta),\ -fm_r\cos(\theta),\ r_0\cos(\theta) + c_0\sin(\theta) + b)'$

$$d = -\vec{n}\,'\,.\,(e_x, e_y, e_z)'$$

Triangulation – Step 2: Planar Intersection



Two 3D Planes :

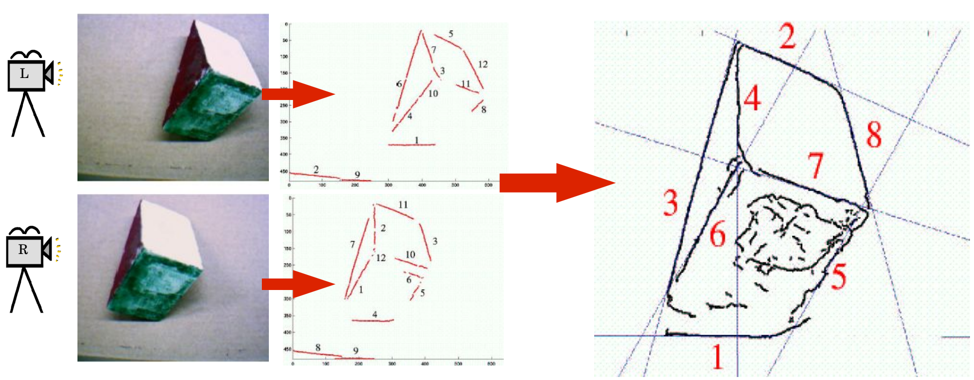
- LEFT : $\vec{n}_L\,.\,(x, y, z)' + d_L = 0$
- RIGHT: $\vec{n}_R\,.\,(x, y, z)' + d_R = 0$

Intersection of two 3D planes gives a 3D line

- **3D line defined as : direction *v*, point *a***
  - $\vec{v} = \vec{n}_L \times \vec{n}_R$ *(cross product of planar normals)*

    let : $M = [\,\vec{v},\ \vec{n}_L,\ \vec{n}_R]'$ *(3 x 3 matrix)* $\vec{b} = [0, -d_L, -d_R]'$ *(3D point)*
  - $\vec{a} = M^{-1}\vec{b}$ *(point on line closest to origin)*

Projected Stereo Line Positions



- 3D awareness of our lines as : vector $\vec{v}$ & point $\vec{a}$
  - defined in $\mathbb{R}^3$

Correspondence Between Lines

Lines in 3D space

Number	Pairs	direction	point
1	L1:R4	(-0.82, 0.08,-0.56)	(9.1,2.0,-13.0)
2	L5:R11	(0.61,-0.06, 0.78)	(-125.3,98.6,107.1)
3	L6:R7	(-0.28,-0.95,-0.03)	(0.9,-10.6,294.4)
4	L7:R2	(0.07,-0.62,-0.77)	(48.3,-97.0,82.9)
5	L8:R5	(-0.18,-0.45, 0.87)	(114.8,91.8,72.1)
6	L10:R12	(-0.50,-0.73, 0.44)	(71.5,77.0,208.8)
7	L11:R10	(0.79,-0.20, 0.57)	(-98.4,57.2,154.6)
8	L12:R3	(0.11,-0.69,-0.70)	(110.4,-123.6,140.1)

- each line represented by point / vector

Summary

- **Feature Identification**
  - Last lecture : Canny Edge Detection & RANSAC
- Feature pre-processing
  - extract extended position/shape/colour description
- **Stereo Correspondence**
  - local and global **matching constraints**
  - **epipolar geometry**
- **Triangulation**
  - intersection of 3D planes → 3D line

Next lecture : recognition