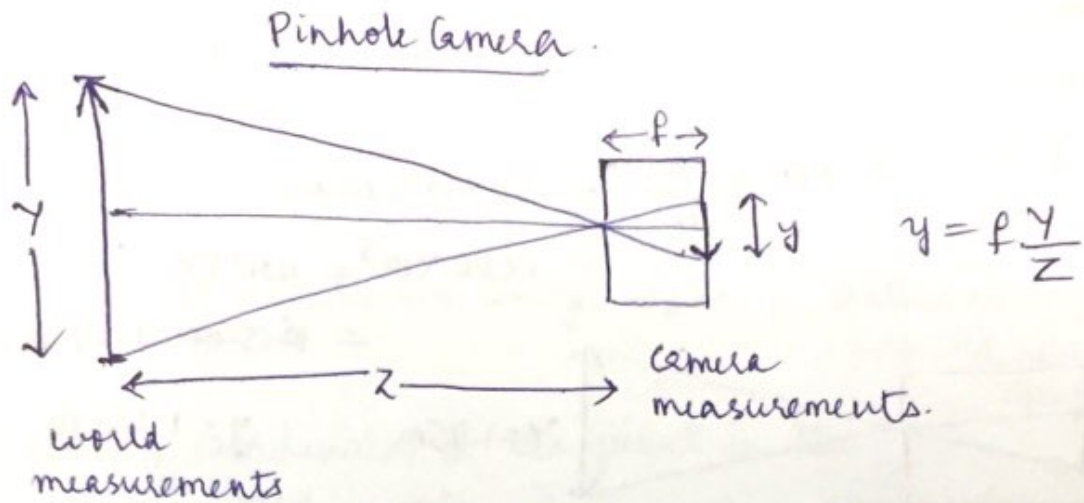


COMPUTER VISION

→ Geometry - Imaging and Camera model -

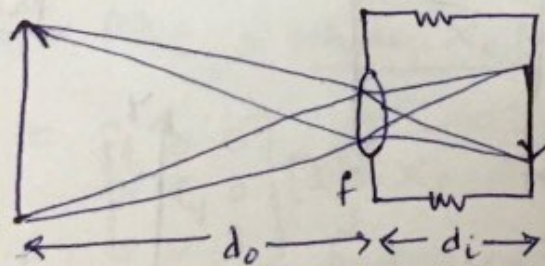
Camera - light tight box with a small hole.



Size of the box → zooming measure.
can be changed.

Camera with lens

Then lens equation $\equiv \frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i}$



for a fixed d_i ,
for only one
value of d_o ↓
you get a sharp
'image', the other
distances are blurred

Lens + Aperture → sharp and focus. — the depth of the world.
(for small apertures) ↑

Depth of field $\propto \frac{1}{\text{Aperture.}}$

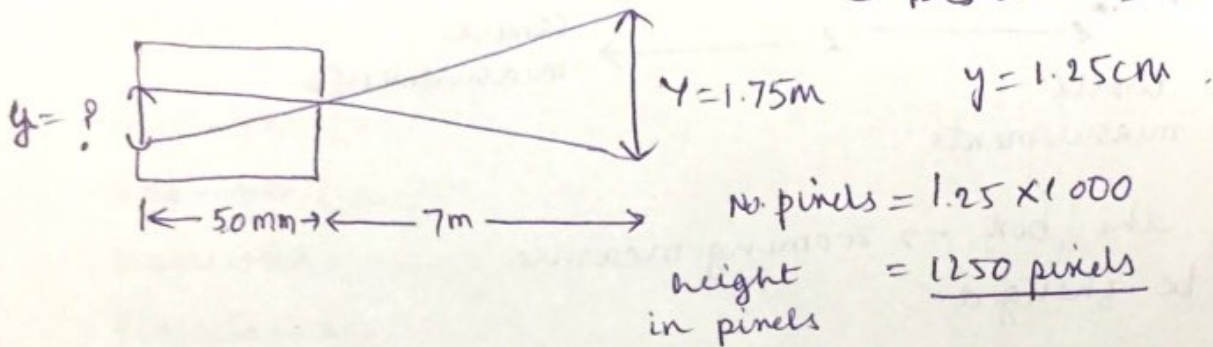
Focal Ratio $= \frac{f}{d} \rightarrow$ aperture size.

Resolution - Number of samples in an image (no. of sensor elements)

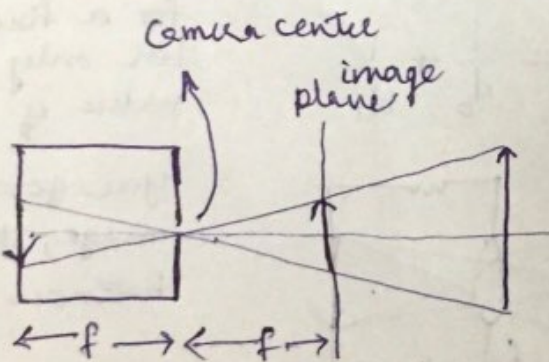
Camera - maps 3D to 2D.

$Y = 1.75\text{m}$, sensor = 3cm tall, with a resolution of 4000×3000 .
 $Z = 7\text{m}$.
 $f = 50\text{mm}$.
 3cm - 3000 pixels
 1cm - 1000 pixels

$$y = f \frac{Y}{Z} = 50 \times 10^{-3} \times \frac{1.75}{7} = 50 \times 10^{-3} \times 25 \times 10^{-2} \\ = 1250 \times 10^{-5} = 0.0125\text{m} \\ = 1.25\text{cm}$$



Camera is raised; the image comes down. $\Delta y = \frac{f}{Z} \Delta Y$

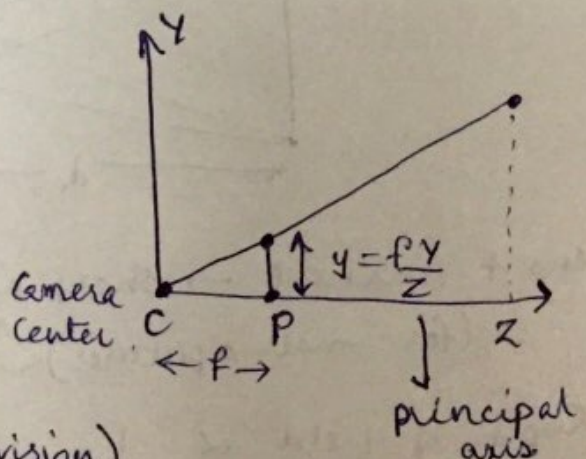


world coordinates = x, y, z
 image coordinates \equiv

$$x = \frac{fx}{z}, y = \frac{fy}{z}$$

Non linear (division)

Perspective Projection



* Non linear \rightarrow converted to linear \equiv (Homogeneous coordinates)

3dimensional \rightarrow Homogeneous

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \leftarrow \frac{x}{w} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{matrix} \text{Camera} \\ P \end{matrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{3 \times 4} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{\bar{X}_w} \quad \begin{matrix} x = PX \\ , w = z \end{matrix}$$

real world coordinates \rightarrow image plane \equiv matrix multiplication \rightarrow "linear operation".

\bar{X}_w - vector, coordinates of the point in the world $\Rightarrow \bar{X}_c$ - camera coordinate system.

origin - camera center

x-axis - principal axis.

Camera - matrix, entity that transforms world into image.
 $\hookrightarrow P$

If camera changes, world coordinates change.

Image point, $x = PX_c$, where $x_c \equiv$ 3D point in camera coordinates.

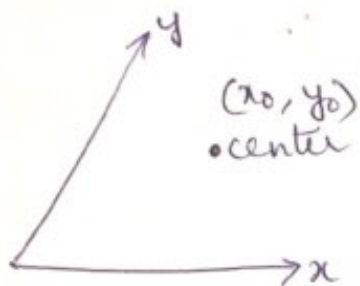
$$= \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3 \times 3} \begin{matrix} [I|0] \\ \downarrow \\ \text{appending} \end{matrix} X_c$$

$$= K [I|0] X_c$$

K - "internal" camera calibration matrix.

Focal length - pixel units

\downarrow
 x focal length and y focal length might be different.



Non orthogonal axes with skew s
 add x_0, y_0
 skew - component of y added to x .

upper triangular matrix
 with 5 degrees of freedom
 (f_x, f_y, s, x_0, y_0)

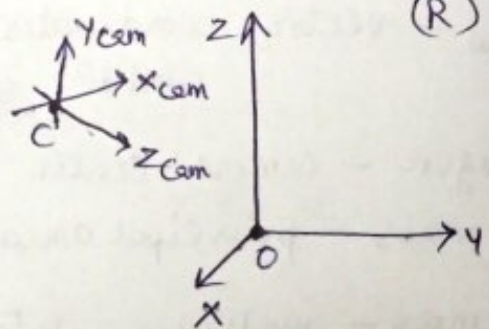
$$K = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x = \frac{f_x X + x_0 Z}{Z}$$

$$= \frac{f_x X}{Z} + x_0$$

Camera moved to $C \equiv$
 points in the world moved
 to $-C$

Camera is at a point C
 Camera axes - rotated by R
 (R)



Camera and world relation -

$$X_C = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} X_W, \text{ translation by } -RC$$

2D projection x of a 3D point $X_W \equiv$

$$x = K [I | 0] X_C = K [R | -RC] X_W.$$

$$x = P X_W, \text{ camera matrix, } P = [KR | -KRC]$$

$$= [M | p_4]$$

$$\begin{bmatrix} p_1 & p_2 & p_3 & p_4 \end{bmatrix}, \begin{bmatrix} p^1 \\ p^2 \\ p^3 \end{bmatrix}$$

3 vectors
 4 vectors.

R, C - external parameters

common $K \rightarrow \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$

general $K \rightarrow \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$

Feature Matching - detect, describe and match the interesting points.

↳ invariance - image transformations, illumination

Feature detection -

* Harris corner detection - interest points (corners)
 ↑
 junctions of contours.
 ↓
 areas with intensity variations along both the axes.

$$A(x, y) = \sum_{w(x, y)} (I(x, y) - I(x + \Delta x, y + \Delta y))^2$$

$$= \sum_w (I_x \Delta x)^2 + (I_y \Delta y)^2 + 2 I_x I_y \Delta x \Delta y$$

$$= [\Delta x \ \Delta y] \left(\sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$A(x, y) = [\Delta x \ \Delta y] M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \rightarrow \text{ellipse equation.}$$

↓
 Covariance matrix

→ eigen values decide the interest point

$$R^T \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix}$$

General Camera Equation -

3x4 - P matrix \rightarrow projects the world-C \rightarrow Image-C
 \rightarrow Left 3x3 submatrix \equiv non singular
 \rightarrow is singular \rightarrow Orthographic projection (flat, parallel projection)

3x4 camera matrix \rightarrow 11 degrees of freedom + (one is fixed to 1)

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 \end{bmatrix} = \begin{bmatrix} P_1 & P_2 & P_3 \end{bmatrix}^T$$

\downarrow column vectors of P \downarrow row vectors of P

$\begin{bmatrix} 3 \\ 4 \\ 5 \\ w \end{bmatrix}$ $w=1$, point in the world-C $\equiv (3, 4, 5)$
 $w=2$, point in the world-C [towards origin] $= (\frac{3}{2}, \frac{4}{2}, \frac{5}{2})$
 $w=0.5$, away from the origin.

$\begin{bmatrix} 3 \\ 4 \\ 5 \\ 0 \end{bmatrix} \rightarrow$ point at ∞
 α
vanishing point \rightarrow ray shot out of the world from the world origin which vanishes.
 \uparrow passing through the world-C

$\begin{bmatrix} 3 \\ 4 \\ 5 \\ 0 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 6 \\ 0 \end{bmatrix}$ these are two different (vanishing)
rays from the origin in different directions

$$P = [P_1 \ P_2 \ P_3 \ P_4]$$

* Columns of P, P_1, P_2, P_3, P_4 are the images of vanishing points of the world X, Y and Z directions.

$$\text{Vanishing point in X direction} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow P \cdot V_x = P_1$$

$$\text{||ly } v_y = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad P \cdot v_y = R_2$$

what is P_4 ? $\rightarrow P_4 = P \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ \nearrow origin

So, camera matrix's columns \rightarrow come from 4 points.

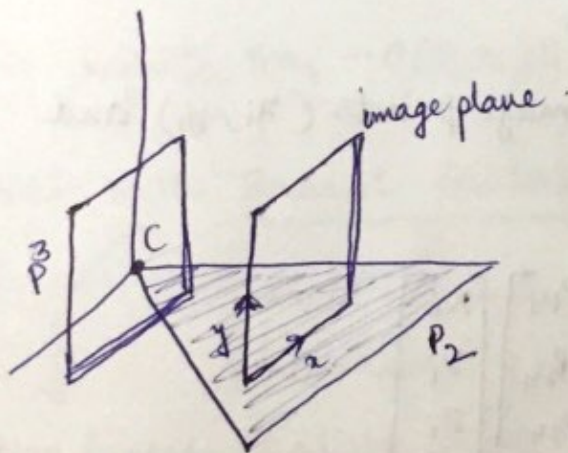
Row vector, $P^3 \rightarrow$ physical meaning?

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

if $P^3 [x \ y \ z \ w]^T = 0$
 $\rightarrow w = 0.$

Physical image point \rightarrow at infinity

\hookrightarrow when the vector is parallel to the image plane
 \hookrightarrow lies on the x - y plane of camera plane



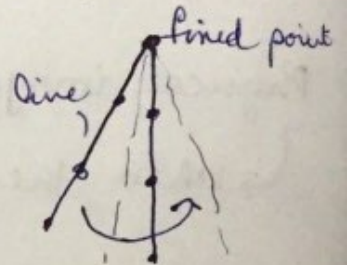
$P_2 \rightarrow \begin{bmatrix} u \\ 0 \\ w \end{bmatrix}$
 \downarrow
 x -plane.

\rightarrow Principal point is given by $x_0 = Mm_3$, $m_3 \equiv$ third row vector of matrix M .

$\rightarrow \det(M)m_3 \rightarrow$ principal axis as a vector from the camera centre through the principal point to the front of the camera.

Camera Calibration -

- Camera is a combination of intrinsic (K) and extrinsic (RC) parameters.
- To solve for camera matrix P, we need to know world and camera coordinate pairs. [3D Reference object based]
- Calibration using a plane with unknown motion precise.
- Calibration from a set of collinear points that moves such that the lines passing through a fixed point
- Self Calibration (world is static and point correspondence across images).
~ Structure from motion



3D Reference object based →

- world points (X_i, Y_i, Z_i) , image points (x_i, y_i) and solve for equations, P_{mn} .

$$\begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ w_i \end{bmatrix}$$

we don't know w_i , $(x_i/w_i, y_i/w_i) \equiv (u_i, v_i)$

$$u_i = \frac{x_i}{w_i} = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}}$$

measured in pixels. linear equation

$$x = f \frac{X}{Z}$$

12 variables \rightarrow multiple equations \rightarrow multiple points
 1 pair \rightarrow 2 equations
 6 points = 12 rows
 [world-C + image-C]
 pairs

$$\left. \begin{array}{l} \text{Equations, } G \\ \text{Parameters, } P \end{array} \right\} \equiv Gp = 0$$

\rightarrow Decompose P into K, R and t

$$P = [M \ P_4] \quad , \quad M = KR \quad \text{and} \quad P_4 = -Kt \quad , \quad K = \begin{bmatrix} \alpha & \delta & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow (KR)(KR)^T = \underbrace{KR R^T K^T}_{\substack{\downarrow \\ I}} = \underline{KK^T}$$



$$R = K^{-1}M$$

$$t = K^{-1}P_4 = -R^{-1}K^{-1}P_4$$

\rightarrow Refine P ; \rightarrow tweak $P \equiv$ real points start coinciding with the image points

$$\min_P \sum_i \|x_i - \phi(P, x_i)\|^2$$

\rightarrow Deal with Radial Distortion = shift: $\boxed{\delta} = 1 + (K_1 r_c^2 + K_2 r_c^4)$
 $r_c^2 = x_c^2 + y_c^2$

\downarrow
 barrel - pixels move away from center 
 pincushion - pixels move towards the center 

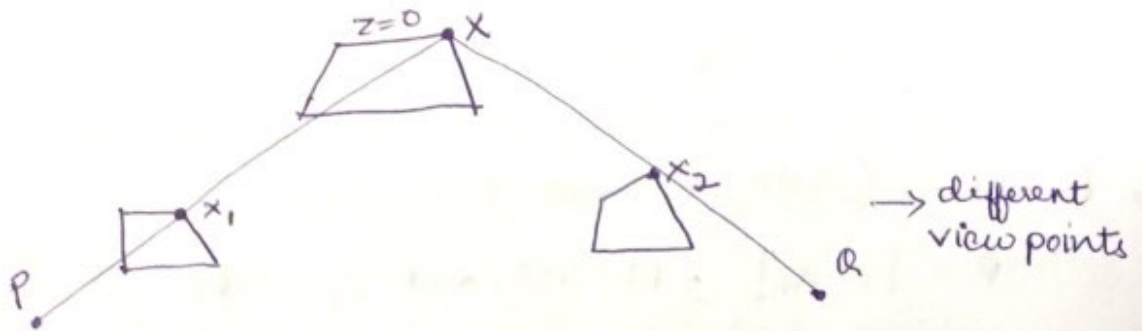
Non linear scaling, $\left. \begin{array}{l} \hat{x}_c = \delta x_c \\ \hat{y}_c = \delta y_c \end{array} \right\}$ modified coordinates

\rightarrow should be applied before linear operations.

\Rightarrow tweak K_1, K_2 (parameters) such that lines are lines

Geometry -

Case 1: Planar world ($z=0$) - $x-y$ plane



Relationship between x_1, x_2 ?

From P
view-point

$$\bar{x}_1 = K[R \ t] \bar{X}$$

$$= K \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} \rightarrow \text{because it is an } x-y \text{ plane.}$$

↓
can be rewritten as [without the third-column]

$$= K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \} \rightarrow (x, y) \text{ coordinates of the point.}$$

$\underbrace{\hspace{10em}}_H$

$$\bar{x}_1 = H_1 \bar{X}$$

\downarrow
 3×3

H_1 can be invertible.
world \rightarrow image

3d \rightarrow 2d (loss of information)

$$\bar{x} = P \bar{X}$$

\downarrow
 3×4

because P is not invertible
so, we cannot go back, hence
loss of information

From Q
view-point

|| e_y , K may or maynot be the same
(same/diff) camera

$R, t \rightarrow$ will be different.

$$\bar{x}_2 = H_2 \bar{X}$$

$$\Rightarrow \bar{x}_1 = H_1 H_2^{-1} \bar{x}_2$$

$$\bar{x}_1 = H_{12} \bar{x}_2$$

3x3
matrix.

$$\bar{x}_2 = H_2 H_1^{-1} \bar{x}_1$$

$$= H_{21} \bar{x}_1$$

(Since H_1 & H_2 are invertible)
plane to plane mapping

Homography

with no loss of
information.

(transformation across images)

⇒ Every point in Image-1 can be mapped to a point in Image-2 using a single 3x3 matrix in the case of a planar world.

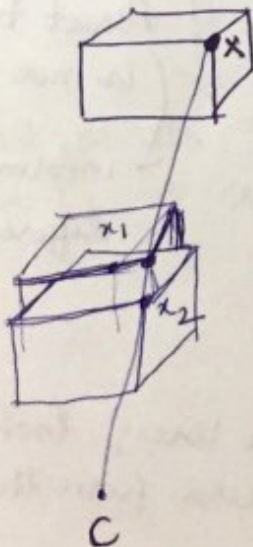
No loss of information doesn't happen if world^{that} is not planar (i.e. has depth).

⇒ There exists "Homography" between the 2-view images.

$H \Rightarrow$ singular \Rightarrow when camera center lies on the planar world. ↓

all the image points lie on line.

Case 2 :- Same Camera Center.



Arbitrary 3D world.

Relationship between x_1 and x_2 ?

→ World is 3D and there are ~~no~~ obstructions ∞ it is not completely captured in the image.

→ Since camera center is same, line (points on the line) don't get occluded.

$$\bar{x}_1 = K_1 R_1 [I - c] \bar{x}$$

→ cannot go back to the world as inversion is not trivial.

$$\bar{x}_2 = K_2 R_2 [I - c] \bar{x}$$

$$\bar{x}_2 = K_2 R_2 (K_1 R_1)^{-1} K_1 R_1 [I - c] \bar{x}$$

$$\bar{x}_2 = \underbrace{K_2 R_2 (K_1 R_1)^{-1}}_{3 \times 3 \text{ matrix}} \bar{x}_1$$

$$\bar{x}_2 = H_{21} \bar{x}_1$$

$$\begin{aligned} \bar{x}_1 &= (K_1 R_1) (K_2 R_2)^{-1} \bar{x}_2 \\ &= H_{12} \bar{x}_2 \end{aligned}$$

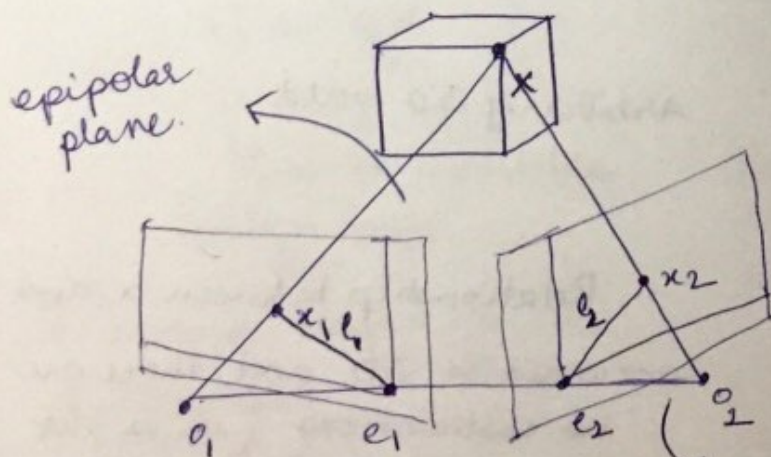
, t → Remains same, but the cameras are differently rotated.

Even if one of the images is zoomed in
↓
then also, both the images are related by homography.

Two images are related by homography,
 H_{12}, H_{21} - non singular.

Same center → with different rotations - Panorama photography
Computing homography ⇒ Panorama stitching.

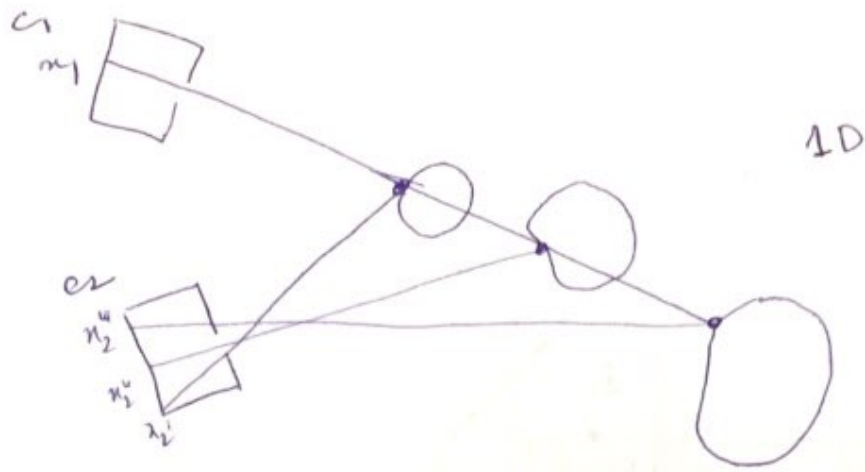
Case 3: generic world and cameras.



l_1, l_2 → epipolar line
 e_1, e_2 → epipoles.

Direct transformation
(is not possible)
→ information difference exists.

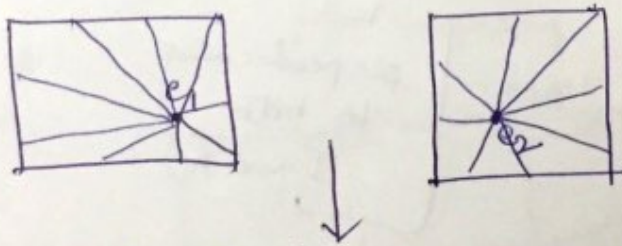
this line; looking at one camera from the other



$x_2 \rightarrow$ should be lying on a line.

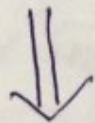
Any point of $l_1 \rightarrow$ will be lying on the line l_2
 For different $x_s \rightarrow$ different planes that move on the
 same O_1 and O_2 .

\downarrow
 different lines \rightarrow radially going out from e_1 and e_2



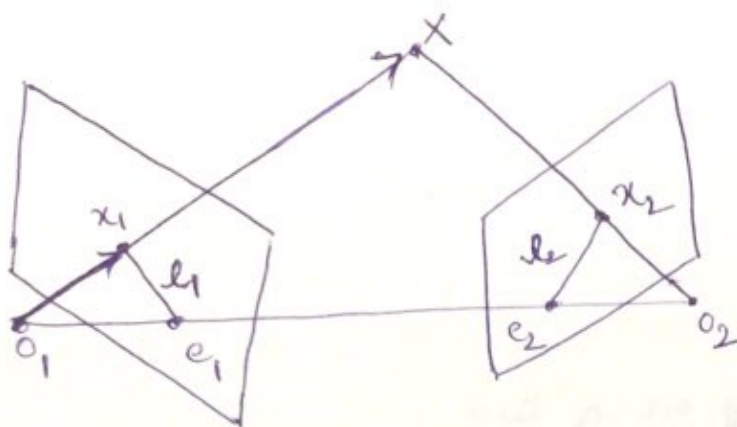
and there's a mapping between lines.

Points in one image \rightarrow are constrained to lie on one
 line in the other image.



\rightarrow All world points that map to x_1 in I_1 , map to a
 line l_2 in $I_2 \rightarrow$ called an epipolar line.

Epipolar geometry \Rightarrow



$$\begin{aligned} \vec{O_1 X} &\rightarrow \lambda_1 \vec{x}_1 = \vec{X} \\ \vec{O_2 X} &\rightarrow \lambda_2 \vec{x}_2 = R \vec{X} + T \end{aligned}$$

$$\lambda_2 \vec{x}_2 = R(\lambda_1 \vec{x}_1) + T$$

\hat{T} (Cross product matrix)

$$\begin{aligned} T \times T &= 0 \\ \hat{T} \hat{T} &= 0 \end{aligned}$$

$\hat{T} \vec{x}_2 \equiv$ cross product of \vec{T} and \vec{x}_2

$$T \times \vec{x}_2 = \hat{T} \vec{x}_2$$

($\lambda_1 \rightarrow$ scalar)

$$\hat{T} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ T_y & T_x & 0 \end{bmatrix}$$

$E \rightarrow$ Essential matrix.

* x_1 falls on the line $E \vec{x}_2$
epipolar line

$$\lambda_2 \hat{T} \vec{x}_2 = \lambda_1 \hat{T} R \vec{x}_1 + 0$$

$$\lambda_2 \vec{x}_2^T \hat{T} \vec{x}_2 = \lambda_1 \vec{x}_2^T \hat{T} R \vec{x}_1 + 0$$

perpendicular to both \hat{T} and \vec{x}_2

\downarrow
0.

$$0 = \lambda_1 \vec{x}_2^T \hat{T} R \vec{x}_1$$

$$\Rightarrow \vec{x}_2^T \hat{T} R \vec{x}_1 = 0$$

$$\vec{x}_2^T E \vec{x}_1 = 0$$

$$\vec{x}_1^T E \vec{x}_2 = 0$$

3×3

Strong calibration

$$x_1 = K_1 X$$

$$x_2 = K_2 X (RX + T)$$

} generic cameras $\rightarrow K_1, K_2$
camera matrices

$$\bar{x}_2^T K_2^{-T} T^T R K_1^{-1} \bar{x}_1 = 0$$

$$\bar{x}_1^T F \bar{x}_2 = 0$$

} F = Fundamental matrix

\rightarrow weakly calibrated case.

\downarrow
3x3

$E, F \rightarrow$ independent of world point.

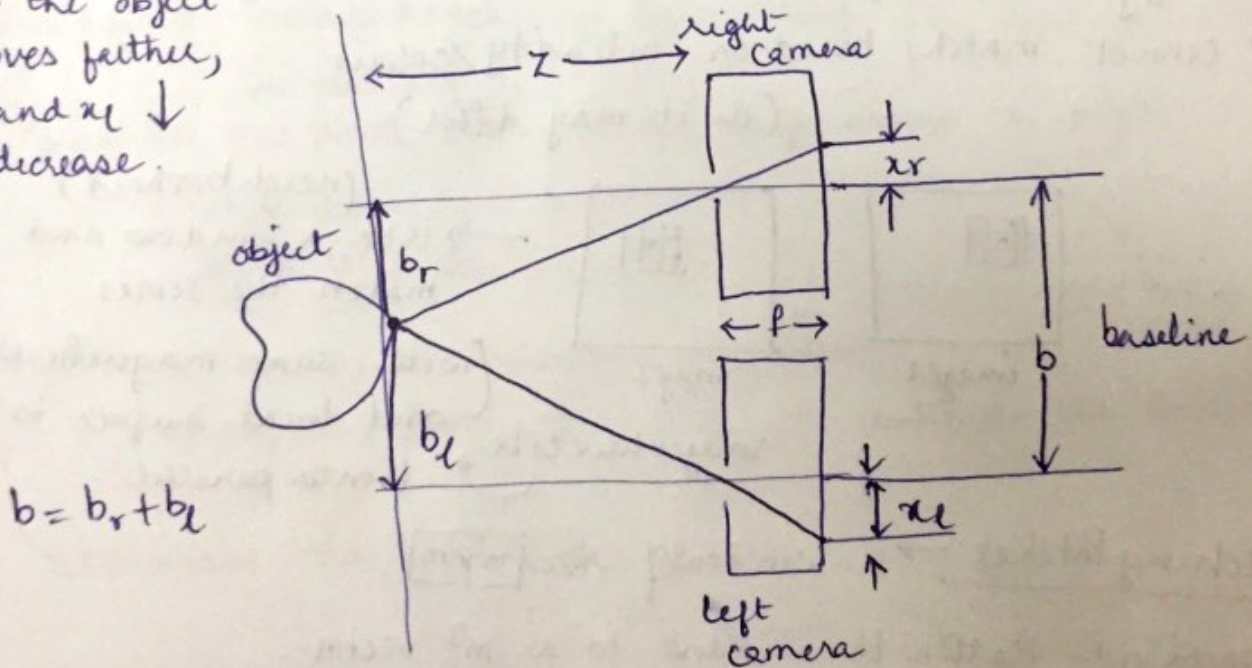
\downarrow
contains $K_1 \& K_2$

Stereo Matching \rightarrow

\downarrow left eye view \neq right eye view.

you always look at only one eye view and the other eye is (dominant eye) just supporting.

\rightarrow as the object moves farther, x_r and x_l decrease.



if $b=0$, no stereo perception

$b \downarrow$, stereo perception \downarrow

if $b \uparrow$, overlap between images comes down

$$\frac{b_r}{z} + \frac{b_l}{z} = \frac{x_r + x_l}{f}$$

$$\frac{b_r + b_l}{z} = \frac{x_r + x_l}{f} \quad (\text{absolute values of } x_l \text{ and } x_r)$$

$$\frac{b}{z} = \frac{d}{f}, \quad \text{Total disparity} = x_r + x_l = d$$

$$z = \frac{f \cdot b}{(x_r + x_l)}$$

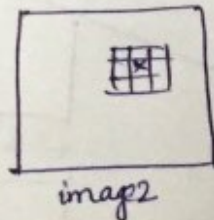
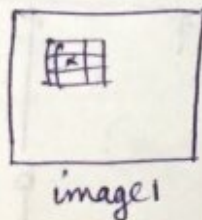
\uparrow object appearing in right camera
 \nwarrow obj in left camera

* large baseline \rightarrow more reliable estimates of depth.

Steps:

* Identify common points.

\rightarrow cannot match based on intensity/colour.
(as it may differ).



(neighborhood)
 \rightarrow take a window and match the scores
 (with same magnification)
 and local surface is
 fronto-parallel.

Matching Patches \rightarrow window of size $m \times m$

vectors v, v' - flatten the window to a m^2 vector.

Matching scores:-
 ① Absolute difference $\equiv \|v - v'\|_1$
 ② Squared difference $\equiv \|v - v'\|_2$

③ Correlation: - $\cos \theta$, angle between them

$$\frac{v^T v'}{\sqrt{v^T v} \sqrt{v'^T v'}}$$

④ Normalised correlation:

$$\frac{\bar{v}_1^T \bar{v}}{\sqrt{\bar{v}_1^T \bar{v}_1} \sqrt{\bar{v}^T \bar{v}}}$$

, \bar{v}, \bar{v}_1

vectors with mean subtracted.

Invariant to affine changes in intensity (colour).

Square root - takes time computationally. ↑

to make it easier →

⑤ Census transform → 0,1 based on whether the values in the window are going down or going up.

↓
bit vector.

to describe the patch

bit arithmetic is way easier.

→ speeds up the process - for real time.

⑥ Birchfield-Tomasi Match - for accuracy. [move the window by fractional units.]

in matching
even for one pixel error - causes huge errors in depth.

(when disparity is small, object is farther away).

$$Z = \frac{f \cdot b}{x_l + x_r}$$

suppose $x_l + x_r = 0$, at infinity, but when $x_l + x_r = 1$, jumps closer and it continues the same way.

→ increase the resolution of depth

→ Matching pixels → takes a lot of time in real time (like in self-driving cars)
for 1 pixel → search over the whole image

→ Time complexity

→ False matches (noise).

Epipolar → match lies on the epipolar line

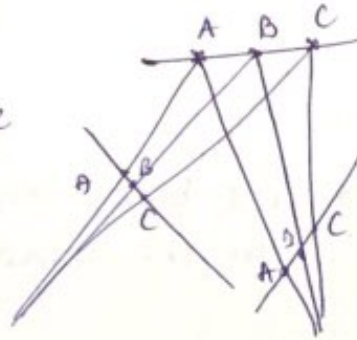
↳ need to search only the line.
→ reduces the search time.

(line and line)
- sequence matching
"DP"

* Colour constancy → generally doesn't change across views

* Ordering or Monotonicity →

order doesn't change
[searching in only one
direction] ↓
eliminate half
the matches

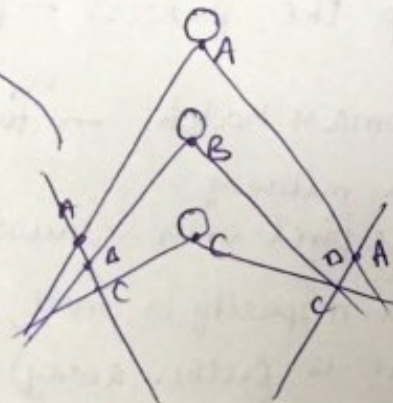


violated when great difference in depth

* Uniqueness

ordering
is violated

one point on left image
↕ mapped to
only one point on right image



* sparse correspondence

good points to match.

(Harris corner
detection)

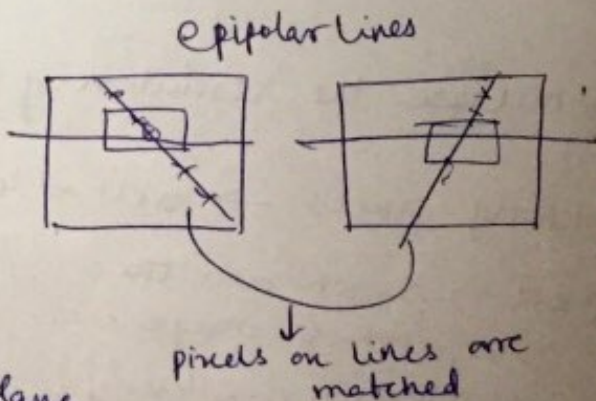
Epipolar → "search is
limited to line"

Rectification →

convert such epipolar lines
to parallel lines

cameras have same image plane
with only a translation
No rotation (only translation)

epipoles are at infinity



→ In such case, Epipolar constraints changes to $y = y$

To convert the epipolar lines [image planes parallel]
parallel

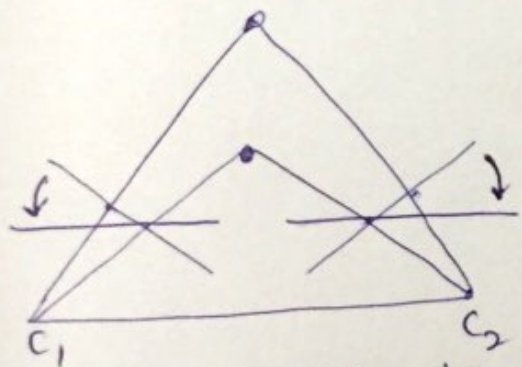
plane transformation \sim homography \Rightarrow you get parallel images
(by rotation) H

and also
may be
change in
scale / image
center.

Rectification

[you get
stereo -
rectified
images]

might get parallel
images but may
not be in the
same plane



Homography to
both the images.

image planes are
made parallel