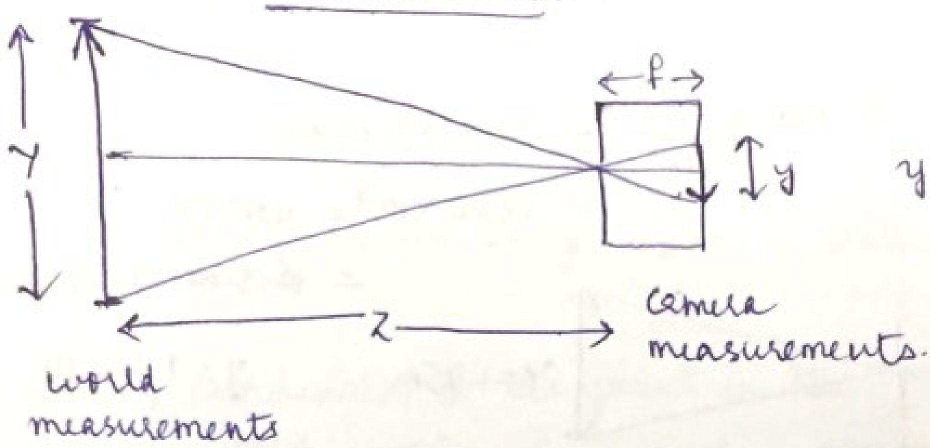
COMPUTER VISION

→ Geometry – Imaging and Camera model –

Camera – light tight box with a small hole.

Pinhole Camera.



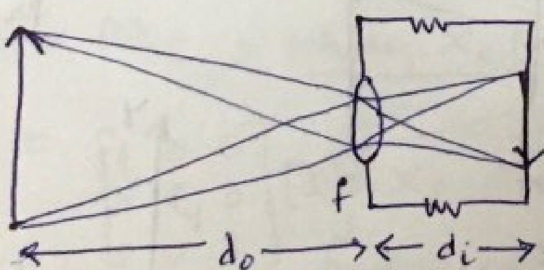
$$y = f\frac{Y}{Z}$$

size of the box → zooming measure
can be changed.

Camera with lens

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

for a fixed di,
for only one
value of do ⟶
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 = $\frac{f}{d}$ → aperture size.

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

<u>Camera</u> – maps 3D to 2D.

$Y = 1.75m$. , sensor = 3cm tall, with a resolution of 4000×3000.

$Z = 7m$.

$f = 50mm$.

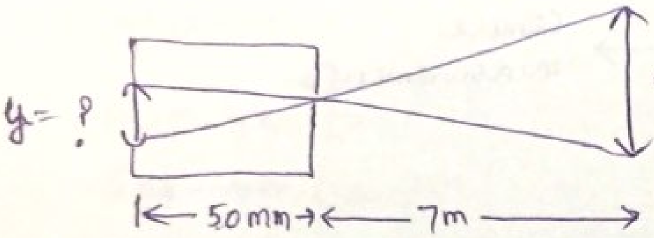
3cm – 3000 pixels

1cm – <u>1000 pixels</u>

$$y = f\frac{Y}{Z} = 50\times10^{-3}\times\frac{1.75}{7} = 50\times10^{-3}\times25\times10^{-2}$$

$$= 1250\times10^{-5} = 0.0125m$$

$$= 1.25 \text{ cm}.$$

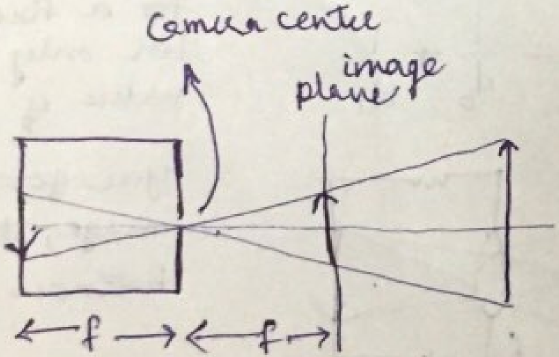


$y = 1.25cm$.

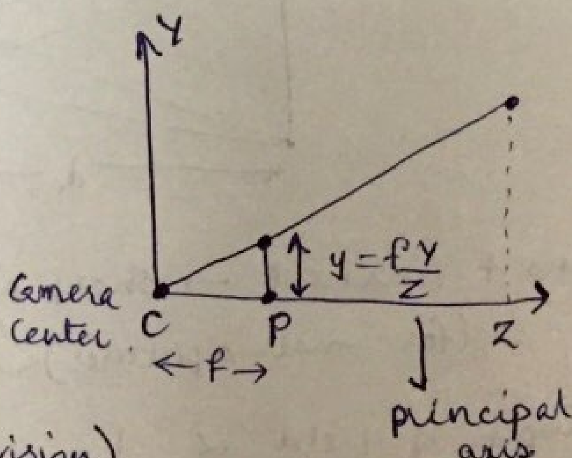
No. pixels = 1.25×1000

height in pixels = <u>1250 pixels</u>

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



Perspective Projection



world coordinates = X, Y, Z

image coordinates ≡

$$x = \frac{fX}{Z}, y = \frac{fY}{Z}$$

Non linear (division)

* Non linear → converted to linear. ≡ (Homogeneous coordinates)

3dimensional $\rightarrow$ Homogeneous

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \xleftarrow{\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W}} \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

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

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.

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

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

K- "<u>internal</u>" camera calibration matrix.

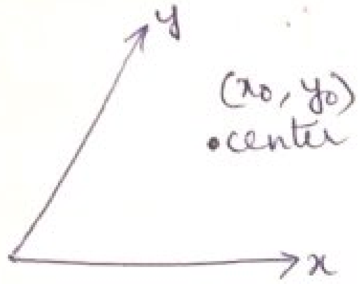
Focal length - pixel units

$\downarrow$

x focal length and y focal length might be different.

Scanned with CamScanner

Non orthogonal axes with skew s



add x_0, y_0

skew – component of y added to x

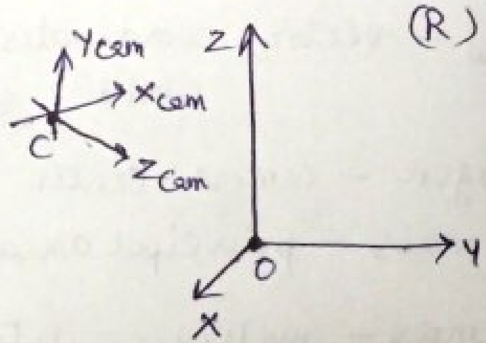
$$K = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

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

Camera is at a point C (t)

Camera axes – rotated by R (R)



Camera moved to C ≡

points in the world moved to −C

<u>Camera and world relation —</u>

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

2D projection x of a 3D point X_W ≡

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

$x = PX_W$, camera matrix, $P = [KR|-KRC]$

$= [M|p_4]$

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

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 -

junctions of contours.

* Harris corner detection — interest points (corners)

areas with intensity variations along both the axes.

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

$$= \sum_{W} (I_x \Delta x)^2 + (I_y \Delta y)^2 + 2 I_x I_y \Delta x \Delta y$$

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

$$A(x,y) = [\Delta x \ \Delta y] \underbrace{M}_{\text{covariance matrix}} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$ ⤳ ellipse equation.

→ eigen values decide the interest point

R T

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

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

Scanned with CamScanner

General Camera Equation –

- 3x4 – P matrix → projects the world-C → Image-C
  - → Left 3x3 submatrix ≡ non singular
  - → is singular → Orthographic projection (flat, parallel projection)

3x4 camera matrix → 11 degrees of freedom + (one is fixed ~~at~~ 1)

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & \cdot & P_{24} \\ P_{31} & P_{32} & \cdot & P_{34} \end{bmatrix} = \underbrace{[P_1 \; P_2 \; P_3 \; P_4]}_{\text{column vectors of P}} = \underbrace{[P_1 \; P_2 \; P_3]^T}_{\text{row vectors of P.}}$$

$\begin{bmatrix} 3 \\ 4 \\ 5 \\ w \end{bmatrix}$ $w=1$, 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

$\begin{bmatrix} 3 \\ 4 \\ 5 \\ 0 \end{bmatrix}$ → point at ∞ or vanishing point → ray shot out of the world from the world origin which vanishes

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.

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

$\|^{ly}$ $V_y = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, $P.V_y = P_2$

what is P_4? $\rightarrow$ $P_4 = P.\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\text{origin}}$

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

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

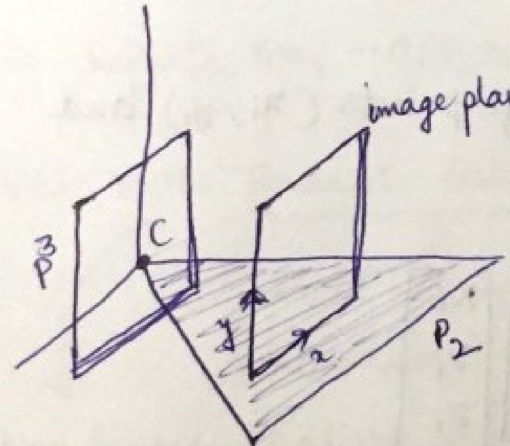
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ W \end{bmatrix}$$

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

Physical image point $\rightarrow$ at infinity

↳ when the vector is parallel to the image plane

↳ lies on the x-y plane of camera plane



$P_2 \rightarrow \begin{bmatrix} u \\ 0 \\ w \end{bmatrix}$

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 (R, C) parameters.

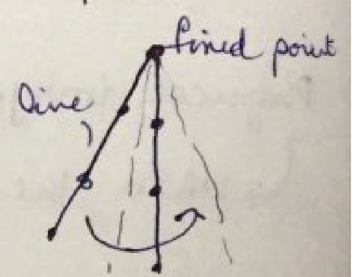
→ To solve for camera matrix P, we need to know world and camera coordinate pairs. [3D Reference object based]

→ Calibration using a precise plane with unknown motion

→ 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}.

$$\begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

we dont 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 → multiple equations → multiple points [world-C + image-C] pairs

1 pair → 2 equations

6 points = 12 rows

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

→ Decompose P into K, R and t

$P = [M \; p_4]$, $M = KR$ and $P_4 = -KRt$, $K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$

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

$R = K^{-1}M$

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

→ Refine P; → tweak P ≡ real points start coinciding with the image points

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

→ Deal with Radial Distortion = Shift : $\boxed{\gamma} = 1 + k_1 r_c^2 + k_2 r_c^4$

$r_c^2 = x_c^2 + y_c^2$

barrel - pixels move away from center

pincushion - pixels move towards the center

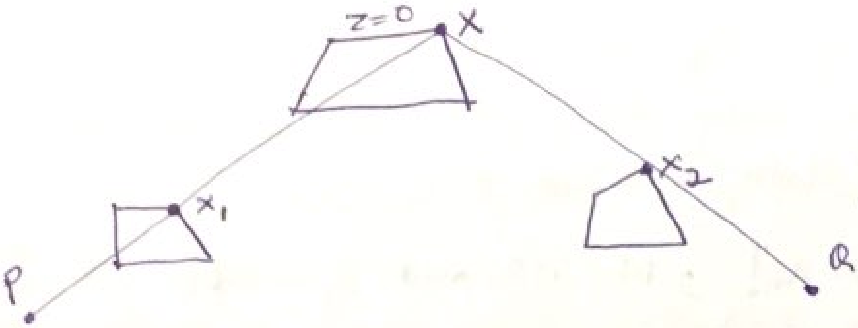
Non linear scaling, $\left.\begin{array}{l}\hat{x}_c = \gamma x_c \\ \hat{y}_c = \gamma y_c\end{array}\right\}$ modified Coordinates

→ should be applied before linear operations.

⇒ tweak k_1, k_2 (parameters) such that lines are lines

Geometry -

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



→ different view points

Relationship between x_1, x_2?

From P view-point: $\bar{x}_1 = K[R\ t]\bar{X}$

$$= K[r_1\ r_2\ r_3\ t]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$ → because it is an x-y plane.

↓

can be rewritten as [without the third-column]

$$= \underbrace{K[r_1\ r_2\ t]}_{H}\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$ } → (x,y) coordinates of the point.

$\bar{x}_1 = \underset{3\times3}{H_1}\bar{X}'$

↓

H_1 can be invertible.

world → image

3d → 2d (loss of information)

$\bar{x} = \underset{3\times4}{P}\bar{X}$

↓

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

From Q view point: IIly, K may or maynot be the same (same/diff camera)

R, t → will be different.

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

$\Rightarrow \bar{x}_1 = H_1 H_2^{-1} \bar{x}_2$ (Since H_1 & H_2 are invertible)

$\bar{x}_1 = \underbrace{H_{12}}_{\substack{3\times3 \\ \text{matrix}}} \bar{x}_2$ plane to plane mapping

Homography

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

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

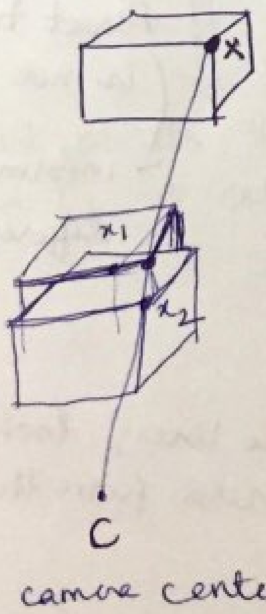
No loss of information doesnt happen ~~if~~ world that is not planar (i.e has depth).

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

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

all the image points lie on line.

<u>Case 2</u> :- Same Camera Center.



Arbitrary 3D world.

Relationship between x_1 and x_2?

→ World is 3D and there are obstructions, so it is not completely captured in the image.

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

$\bar{x}_1 = K R_1 [I \; -c] \bar{X}$

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

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

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

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

$\bar{x}_2 = \underbrace{K_2 R_2 (K_1 R_1)^{-1}}_{3\times3 \text{ matrix}} \bar{x}_1$

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

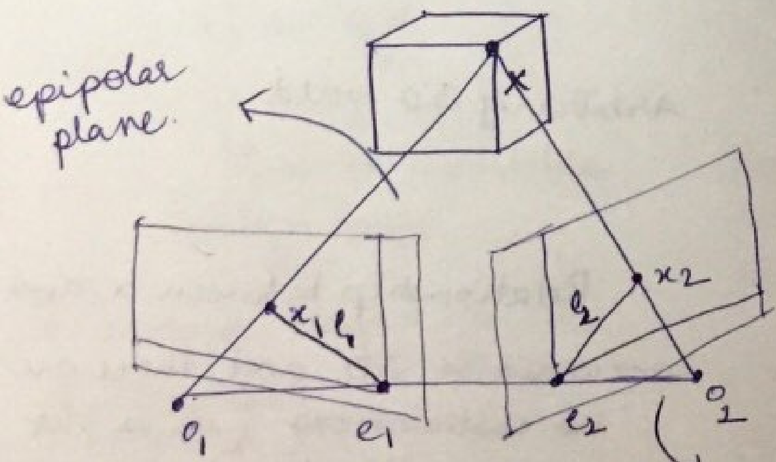
$$\left.\begin{aligned} \bar{x}_2 &= H_{21} \bar{x}_1 \\ \bar{x}_1 &= (K_1 R_1)(K_2 R_2)^{-1} \bar{x}_2 \\ &= H_{12} \bar{x}_2 \end{aligned}\right\}$$

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

Same center → with diffrent rotations – Panaroma photography

Computing homography ⇒ Panaroma stitching.

<u>Case3</u>: Generic world and cameras.

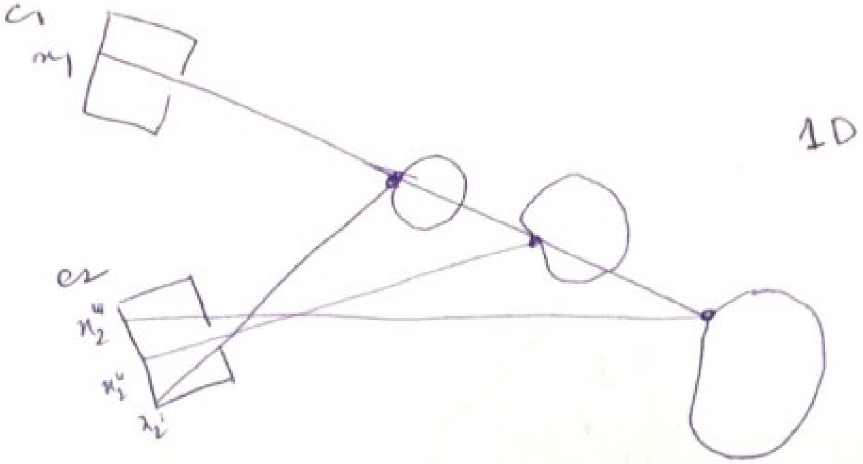


Direct transformation is not possible
→ information difference exists.

this line; looking at one camera from the other

l_1, l_2 → epipolar line

e_1, e_2 → epipoles.

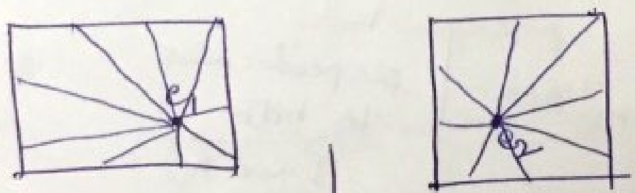


$x_2 \rightarrow$ should be lying on a line.

Any point of $e_1 \rightarrow$ will be lying on the line l_2

For different xs $\rightarrow$ different planes that move on the same O_1 and O_2.

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



and there's a mapping between lines.

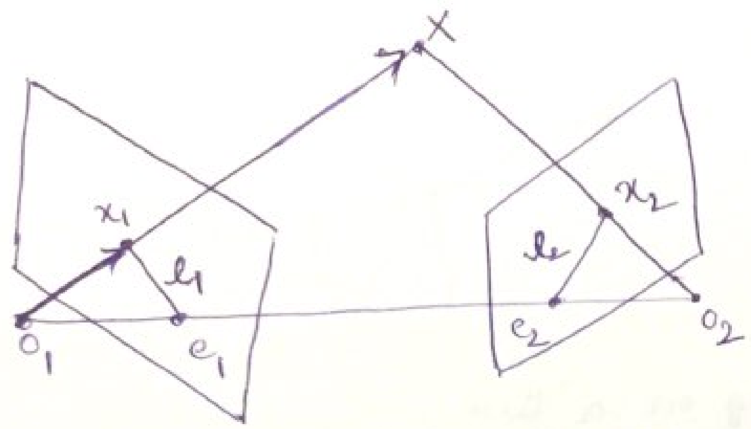
Points in one image $\rightarrow$ are constrained to lie on one line in the other image.

$\Downarrow$

$\rightarrow$ 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.

Scanned with CamScanner

Epipolar geometry ⇒



$\overline{o_1X}$ ↷ $\lambda_1 \bar{x}_1 = \bar{X}$

$\overline{o_2X}$ ↷ $\lambda_2 \bar{x}_2 = R\bar{X} + T$

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

$\hat{T}$ (cross product matrix)

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

$\boxed{T \times T = 0. \quad \hat{T}\hat{T} = 0}$

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

perpendicular to both $\hat{T}$ and $\bar{x}_2$ → 0.

$\hat{T}\bar{x}_2 \equiv$ cross product of $T, \bar{x}_2$

$\boxed{T \times \bar{x}_2 = \hat{T} x_2}$

($\lambda_1 \to$ scalar)

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

$$\Rightarrow \bar{x}_2^T \underbrace{\hat{T} R}_{} \bar{x}_1 = 0$$

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

$$\underbrace{\bar{x}_2^T E}_{l_2} \underbrace{E \bar{x}_1}_{l_1} = 0$$

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

E → Essential matrix. → 3X3

strong calibration

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

epipolarline

$x_1 = K_1 X$

$x_2 = K_2 X \; (\rightarrow RX + T)$

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

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

$\bar{x}_1^T F x_2 = 0$; F = Fundamental matrix $\rightarrow$ weakly calibrated case

$\downarrow$ 3x3

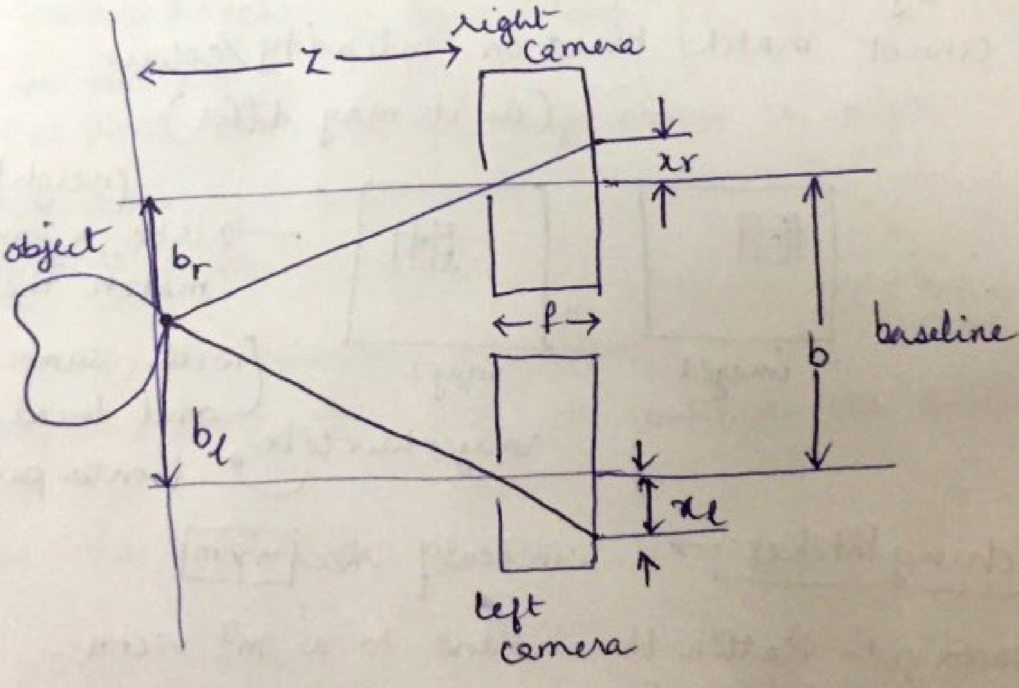
E, F $\rightarrow$ independent of world point.

$\downarrow$ contains K_1 & K_2

Stereo Matching $\rightarrow$ left eye view $\neq$ right eye view.

$\downarrow$ you always look at only one eye view and the other eye is just supporting. (dominent eye)

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



$b = b_r + b_l$

If b=0, no stereo perception

b $\downarrow$, stereo perception $\downarrow$

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

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

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

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

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

object appearing in right camera

$\|^{ly}$ in left camera.

* large baseline → more reliable estimates of depth.

Steps:

* Identify common points.

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

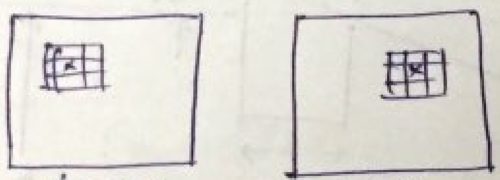


image1   image2

→ take a window (neighborhood) and match the scores

images have to be (with same magnification and local surface is fronto-parallel.

Matching Patches → window of size $\boxed{m \times m}$

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

Matching scores:– (1) Absolute difference $\equiv \|v - v'\|_1$

(2) Squared difference $= \|v - v'\|_2$

(3) Correlation :– $\cos\theta$, angle between them

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

(c) Normalised correlation : $\dfrac{\bar{v}'^T \bar{v}}{\sqrt{\bar{v}_0^T \bar{v}}\ \sqrt{\bar{v}'^T \bar{v}'}}$, $\bar{v}, \bar{v}_e'$

vectors with mean subtracted.

invariant to affine changes in intensity / colour.

Square root – takes time computationally. ↑

to make it easier →

(a) Census thsansform → 0,1 based on whether the valees 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.

(b) Birchfield-Tomasi Match – for accuracy in matching [move the window by fractional units.]

even for one pixel error – causes huge errors in depth.

(when disparity is small, object is farther away)

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

suppose $x_e + x_r = 0$, at infinity, but when $x_e + x_r = 1$, 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. (line and line)

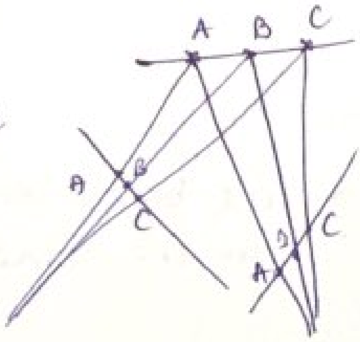
→ reduces the search time. – sequence matching "DP"

* Colour constancy → generally doesnt change across views

* Ordering or Monotonicity →

order doesn't change

[searching in only one direction] ↓ eliminate half the matches



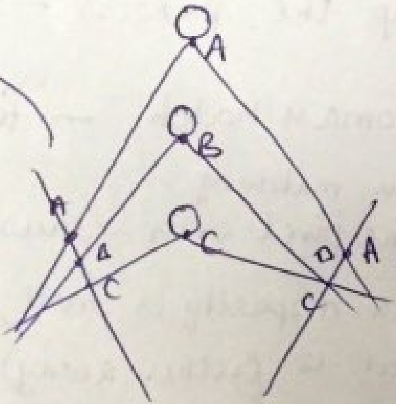
violated when great difference in depth

* Uniqueness

↓

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

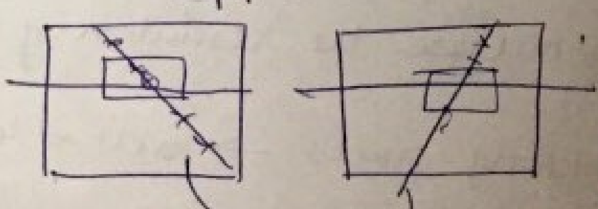
ordering is violated



* Sparse correspondence ↓ good points to match (Harris corner detection).

Epipolar → "search is limited to line"

epipolar Lines



↓ pixels on lines are matched

Rectification → Convert such epipolar lines to parallel lines

cameras have same image plane with only x translation

↓ No rotation (only translation) ↘ epipoles are at infinity

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

To convert the epipolar lines parallel [image planes parallel]

plane transformation ~ homography ⟶ you get parallel images
(by rotation) H

Rectification

and also may be change in scale / image center.

[you get stereo-rectified images]

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

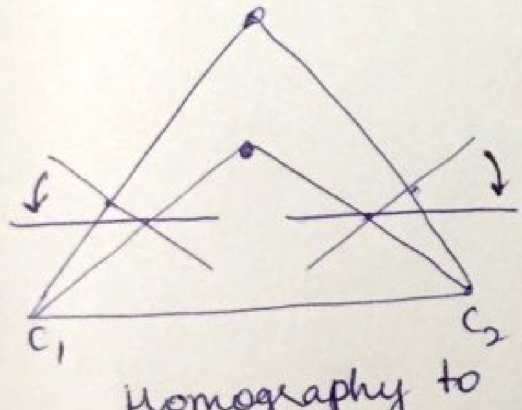


image planes are made parallel

Homography to both the images.

Interest point detection and description →

Challenges ↗ scale
→ viewpoint
↘ lighting.
} recognition

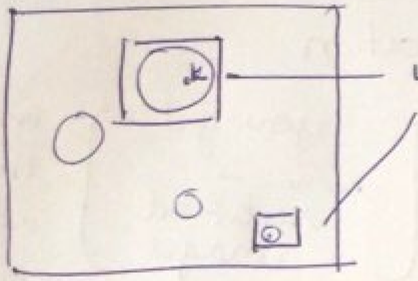
Feature detection

→ Harris feature detection [corners] — not scale invariant
— limitation of scale
↓
to overcome, this → SIFT detector → pyramid of images
"scale space"

Arrows in SIFT point detector → orientation of the feature.

MOPS — MultiScale Oriented Patches. — Descriptor

- keypoints from the detectors → give positions of strong features.
  ↓
  with support regions acc to scale.



capture a support region — defined as a window

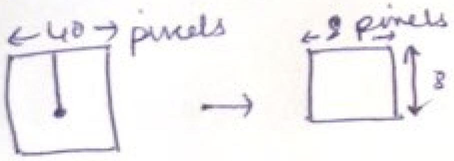
MOPS : 1. 8X8 oriented patch (around the point) ≃ 64 value descriptor

Sampled at 5xscale — orient it wrt the gradients

Sampling a large window

2. Normalisation → for absolute scaling [illumination invariance]

whitening transform ⇒ $I' = (I-\mu)/\sigma$



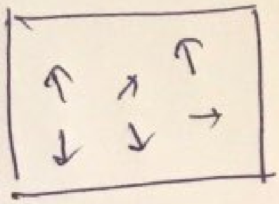
3. Haar wavelet transform applied.

Locality — handles occlusion.

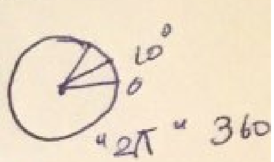
* Illumination invariance → normalisation.
* Scale invariance → Pyramids — Scale Space
  "DOG" — Difference of Gaussians
* Rotation invariance → Histogram of gradients
  ⇓
  Rotate to the most dominant

gradient → first order derivative.

- Histogram of gradients ↓ reduces space



rotation



↓ $360 = 36 \times 10$

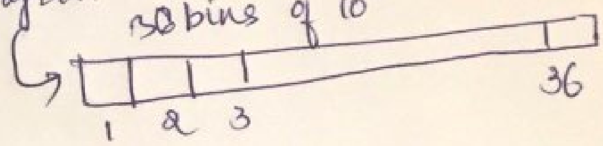
SIFT descriptors -

$16 * 8 = 128$ dimensional descriptor

cells orientations

Histogram

36 bins of 10°



→ Throw out weak edges [noise in the image]

Feature Matching →

Exhaustive search → for each feature, find the best match

↓

can be inconsistent [global structure has to be maintained]

Outliers? — RANSAC — estimating homography.

→ feature pairs (4 points at random)

↓

compute homography estimate

↓

reproject [find the correspondence of the features]

↓

done to maximize the number of inliers

[SSD [patch1, patch2] < threshold

↓

patch based matching

→ RANSAC loop -

- Select four feature points (at random)
- Compute homography, H (exact)
- Compute inliers where $SSD(P_1', H, P_1) < \varepsilon$
- keep "largest set of inliers"