INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY

HYDERABAD

CSE578:
Computer Vision

Professor : Anoop Namboodiri
Notes By : Sai Manaswini Reddy I

2021

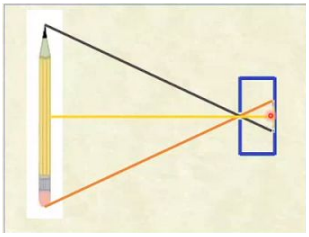
GEOMETRY

Geometry : Imaging and Camera Model :

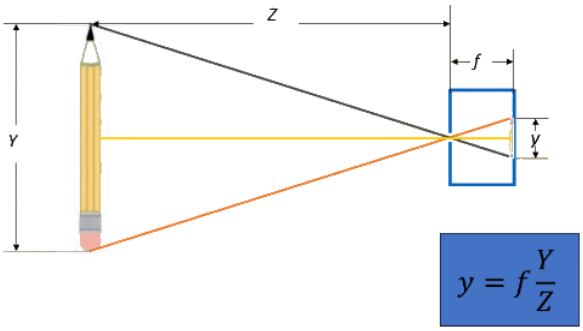
Imaging :

Pinhole camera :

It is a box, with a small hole in it. Light rays will enter into the box through this hole (pinhole) and fall on the backside of the box inside it. This has a photographic plate in the back.



Diagrammatic representation :



Blue box is the camera. y is the length of the screen, part illuminated by yellow ray is lit as yellow. A ray of pink light travels to the other side. This point in the film will only be seeing one point in the world. Only one light ray from a point in the scene will be falling on the screen i.e., *a point on the image screen/sensor is illuminated by a single point in the world*. So, this results in a dim image. Picture captured is inverted.

We can trace back the object from the image point through the pinhole out into the world.

This also gives the geometry of the image. This gives a very simple geometric model of the camera. Triangles formed by any two particular rays are similar.

Y is the size of the pencil, y is image size and z is distance from camera to the pencil.

If we increase the size of the pinhole, one point in the screen will be illuminated by multiple points in the world. This results in a brighter and blur image. We insert a lens to take care of this. This helps us in attaining a brighter and sharper image provided the thin lens equation holds. If we keep on reducing the size of the pinhole, it gets blurrier again due to diffraction

Example :

Question :

You have a person who is 1.75m tall standing at a distance of 7m from a camera. The pinhole camera has a focal length of 20mm. The sensor is 1cm tall and has a resolution of 4000x3000.

1. Find the height of the person in pixels in the image.
2. If the camera is raised by 1m, how much does the person move in the sensor (in pixels)?
3. How much does the Sun move in the above case

Note: Sun is 150 million kms away (in pixels)?

Solution (1):

$$Y = 1.75 \text{ m}$$

$$f = 20 \text{ mm} = 2 \text{ cm} = 0.02 \text{ m}$$

$$z = 7 \text{ m}$$

$$y = f^*(y/z)$$

$$= 0.02^*(1.75/7) = 5 \text{ mm}$$

Sensor height is 10 mm and there are 3000 pixels for a length of 10mm. Therefore, 1500 pixels.

Solution (2):

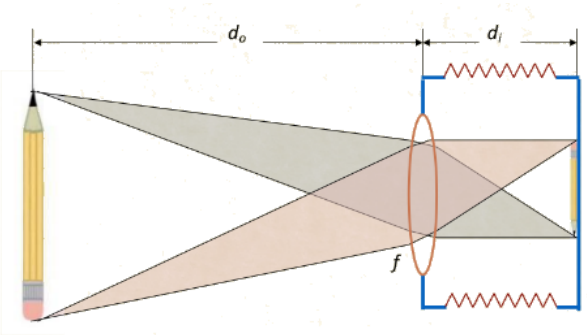
Solution (3):

Does not affect much.

Example : looking at the sun/moon while traveling.

Lens :

Focus and DOF :



If we have a fixed distance from the camera screen to the lens then only one plane in the world can satisfy this equation. DOF is Depth of Field, range of acceptable focus.

Thin lens equation : $\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i}$ and $d_i = f\,\frac{d_o}{(d_o - f)}$

This tells us that if the sum of inverses of d_o and d_i is equal to 1/f then the image is said to be in sharp focus, anything other than this will be blurred. A cone of light coming from a point in the real world, passes through the lens and this all comes back to a single point at a distance d_i from the lens. This results in sharp and bright point. Issue is there is only one d_o for which this equation holds.

We can change the Plane of Focus by changing d_i :

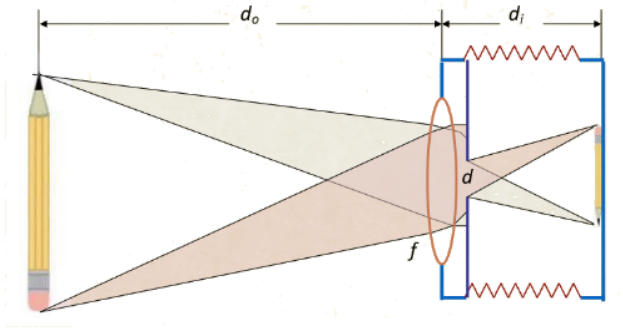
- By the act of focusing the camera.
- Moving the camera further or far away from the object.

Aperture :

We can also include aperture in here. This is a diaphragm with the hole in the middle, size of the hole can be varied. If the hole is very small this becomes a pinhole camera, this does not have an issue of focus. If we increase the size of the hole, we notice the blurriness. If we decrease the size of the we see everything in focus.

We can change DOF by changing the aperture of the lens.

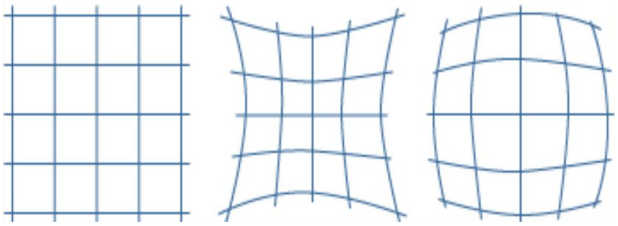
Focal ratio = f/d



Aspects of the camera to be considered :

Geometric Distortions :

Centre line does not have any distortion. Distortion type depends on focal length of the lens. In high focal length/telephoto lenses pincushion distortion is observed and barrel distortion is seen in wide angle lenses.



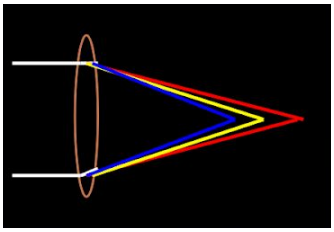
Original Pincushion Barrel

Lens Flare :

If there is a bright object in the image then a secondary image of the bright object is formed. This is due to bouncing back of light in the lens.

Chromatic Aberration :

The amount of Refraction is proportional to its wavelength. If we have a ray of whitelight into the lens then, not all wavelengths will focus at the same point (red will focus farther away



Example :

If we look at the part of the image where there is a sharp change from a bright region to a dark region, we notice a purple colour ; this is known as purple fringing. This is due to blue light being focused first. The effect is seen on the darker side of the image.

Image sampling :

Resolution of sensor : Number of samples in an image/Number of photosensitive elements in the sensors. We will have a grid of photo sensitive receptors in a camera.

A photosensitive receptor computes the average intensity of light received by that particular sensor. More number of sensors results in finer details.

- The number of samples in an image (number of sensor elements) is referred to as its resolution

- The resolution is typically represented as the product of the number of samples in the horizontal and the vertical directions in the image. e.g.: 32x32, 256x256, 640x480

Common Resolutions:

NTSC:	648 x 486
Typical Webcam:	1280 x 720
High-end SLR:	11468 x 8736
Hubble's Telescope:	1600 x 1600

What does a Camera do?

- Form an image on the 2D image plane of the 3D world visible to it.
- Image is behind the lens; the scene is in front.
- 3D world is projected down to a 2D plane.
- Significant loss of information as one dimension is dropped.
- Mathematical depiction of this projection

Camera Model :

Camera Model: Objectives

- Mathematically model what a camera does
  - Also understand what the model means
- Getting the model for a real-world camera
  - Estimation from real world measurements
- Special imaging configurations with simpler properties
  - Simpler relationships
- General theory on fitting linear models under noisy observations
  - Techniques that work across problems.

Perspective Projection :

Camera coordinates are anchored at the pinhole (c is the camera centre of the pinhole). Image plane is usually behind the camera, shifted to the front (f units behind to f units front, equations still hold). This is for the sake of keeping the picture upright.

Cartesian Coordinates : Looking along the YZ plane, we derive Y = f*(y/z) ; similarly for XZ plane we get X = f*(x/z)

For convenience we represent this as a matrix multiplication :

For this we convert the coordinate system to a homogeneous coordinate system.

$$x = \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = PX$$

In Homogeneous coord system if we have a point in the world [x y z] we add a 1 to it to make it a homogeneous coordinate system. So, we add a third coordinate w in the output. To convert back to real coordinates we have to divide by w.

P is used to represent the matrix we created i.e. projection matrix.

Basic Camera Equation :

A pinhole camera projects a 3D point in X_c in camera coords to an image point x via the 3x4 camera matrix p as :

$$x = PX_C = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} [I|0] = K[I|0]X_C$$

Where K (3x3 matrix ; [I|0] makes it 3x4) is an internal camera calibration matrix (all the internal properties of the camera are in this).

- The camera is at the origin
- Z is the camera or Optical axis
- Principal Point : Center of the image
- Focal length in Pixel units
- Orthogonal image axes with uniform scale.

A general Camera :

We initially assumed that the camera centre is the origin and the image centre is along the z-axis, this may not be convenient for all the cases. When we look at the image we usually tend to place origin in the corner.

Here we give the camera centre as (x_c, y_c). angle between 2 camera axes need not be 90^0 and the scale also might differ.

Image center at (x_0, y_0), Non-orthogonal axes with skew *s*, and different scales for axes with focal lengths, α_x and α_y.

New K is :

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

We observe all the required changes, K is an upper diagonal matrix with 5 degrees of freedom.

Moving Camera from Origin :

- General Setting : Camera is not at origin and Z is not the optical axis.
- Camera is at point C in world coordinates. The camera axes are also roasted by a matrix R.

We can incorporate all these into the camera matrix. We convert all world coordinates into camera coordinates.

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

- As we move the camera back, the world moves forward. RC is translation amount and R is rotation amount.

2D projection of x of a 3D point x_w given by :

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

K 3x3 R 3x3 RC 3x1 everything related to camera into K. everything related to motion of the camera into X_C ([R|-RC] is an External Calibration matrix)

$$x = PX_w \text{ ; camera matrix } P = [KR|-KRC] = [M|p_4]$$

Projective Geometry :

Projective geometry in a plane (points and lines in P^2) :

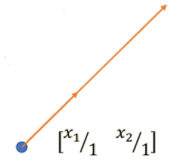
- Points are **x** = [x y 1]
- Consider the line equation : ax+by+c = 0
- Writing this as matrix matrix multiplication (lines and points are 3 d vectors, for a 2D plane thing) :

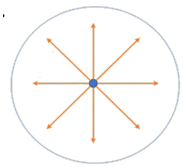
$$[a\ b\ c][x\ y\ 1]^T = l^T.x = 0 \text{ ; } l = [a\ b\ c]^T$$

- Lines are represented by 3- vectors.
    - Note : Overall scale is unimportant.

- $l^T.x = 0$:
    - Therefore, it can be told as (l has to pass through x) or (x is a point on l)
        - x is fixed and l is variable (set of all lines)
        - l is fixed and x is variable (set of all points)

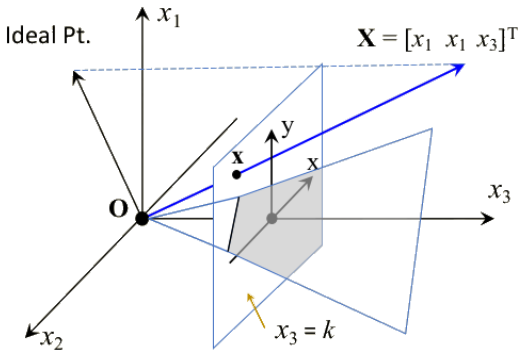
Points/lines at infinity :





- $x = [x_1 \ x_2 \ x_3]^T$ represents $[x_1/x_3 \quad x_2/x_3]$
- x_3 -> 0 :
    - Becomes point at infinity, or vanishing point or ideal point in the direction (x 1 , x 2).
    - Points at infinity can be handled like any other point in projective geometry
    - [x y 0] are all points at infinity on the plane.
    - Collection of all these points at infinity form a circle with infinite radius.
    - But we call this line, because a circle radius infinity is a line.
- $l_\infty = [0 \ 0 \ 1]^T$

Visualization :



- $x = [x_1 \ x_2 \ x_3]^T$ represents rays from the origin in 3-space.
- The plane can be any cross section $\perp$ to x_3 ($x_3 = k$).
- Ideal points are rays on the $x_3 = 0$ plane.
    - Any vector on the x_1,x_2 plane, x_3=k is parallel to it.
    - Plane under consideration for a vector from origin is parallel to it.
    - Real point is point of intersection between ray, line and plane that are parallel to each other.
    - They intersect at infinity (l_∞)
- Lines are planes passing through the origin.
    - To represent the line (line of intersection of 2 planes in the image. We choose a plane and intersect it with x_3=k. The plane through this line that passes through origin gives the equation of the line.
    - Vector that represents the line is perpendicular to the plane.
    - We know that all the points on the plane will form 0 dot product with the vector, so are the points on the line.
- Line at infinity, l_∞ , corresponds to $x_3 = 1$, x_1,x_2=0.

Line joining 2 points :

Representing Points and lines in 2-D plane using projective geometry helps us represent lines and points at infinity without infinity and also helps in simplification. Point at infinity is [x y 0] and lines at infinity is [0 0 1].

- Let p and q be points (x_1,y_1) and (x_2,y_2). We have : $l^Tp = l^Tq = 0$.
- Equation of l :

$$y_1 + \frac{(y_2 - y_1)}{(x_2 - x_1)}(x - x_1) = 0$$

$$(y_2 - y_1)x - (x_2 - x_1)y + (x_2y_1 - x_1y_2) = 0$$

$$l = [(y_2 - y_1) \quad -(x_2 - x_1) \quad (x_2y_1 - x_1y_2)]$$

- Considering them as vectors in 3-space, we want to find the vector l orthogonal to both p and q .
- The cross-product x × y is a solution. Thus, l = p × q

- $p \times q = [(y_2 - y_1) \quad -(x_2 - x_1) \quad (x_2y_1 - x_1y_2)]$

Example :

Line through (5,2) and (3,2)

$$\begin{bmatrix} i & j & k \\ 5 & 2 & 1 \\ 3 & 2 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ -2 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 2 \end{bmatrix}}_{\text{Overall scale is unimportant}}$$

Therefore, the line is y = 2.

Ideal point of the line $[0\ 1\ -2]^T$ is $[1\ 0\ 0]^T$. This is the same for $[0\ 1\ k]^T$ for any k. Line joining $[3\ 4\ 0]^T$ and $[2\ 3\ 0]^T$ is $[0\ 0\ 1]^T$ or l_∞.

Point of intersection of two lines :

- Lines l,m intersect at a point x with $l^Tx = m^Tx = 0$.
- $x = l \times m$
- $l : a_1x + b_1y + c_1 = 0$; and $m: a_2x + b_2y + c_2 = 0$.
- $x = (b_2c_1 - b_1c_2)/(a_2b_1 - a_1b_2)$
- $y = (a_1c_2 - a_2c_1)/(a_2b_1 - a_1b_2)$
- $x = [\ (b_2c_1 - b_1c_2) \quad (a_1c_2 - a_2c_1) \quad (a_2b_1 - a_1b_2)\ ]^T = l \times m$. (3-vectors)
- Duality at work : points and lines are interchangeable.
    - [a b c] line and [a b c] point exist, to get the [a b c] line, there exists a plane perpendicular to [a b c] vector, intersection of this plane and projective plane is [a b c] line.
    - Cross products two 3-vectors points gives us the line and vice versa.

Example :

- Intersection of x=1 and y=2 :

$$\begin{bmatrix} i & j & k \\ 1 & 0 & -1 \\ 0 & 1 & -2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

  Same as (1,2).
- Intersection of x=1 and x=2 :
- x = 1 and x = 2 are parallel to each other. So, their point of intersection is infinity.

$$\begin{bmatrix} i & j & k \\ 1 & 0 & -1 \\ 1 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

- [0 1 0] is a point at infinity.
- Ideal point(vanishing point) of the line $l = [a\ b\ c]^T$ is $[b\ -a\ 0]^T$. This is $l \times l_\infty$, the intersection of l with l_∞ .

Conics - 2nd order entities :

Second order entities can also be represented on the plane (ellipse, parabola, circle,..).

General quadric entity :

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

Introducing homogeneous coordinates ([x y 1] -> [x y w]):

$$ax^2 + bxy + cy^2 + dxw + eyw + fw^2 = 0$$

$$[x \quad y \quad w] \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} = 0$$

A symmetric C represents a conic : $x^TCx = 0$. (circle, ellipse, hyperbola, parabola, ...)

Degenerate Conics include a line (a = b = c = 0) and two lines when $C = lm^T + ml^T$.

General Camera Equation using Projective Geometry :

General projection equation in world coordinates :

$$x = K [R|-RC] X_W = [KR|-KRC] X_C = [M|p_4] X_W$$

3 × 4 matrix P maps/projects World-C to Image-C :

- Left 3 × 3 submatrix is non-singular for finite cameras (KR part , if it is singular then the camera would become an infinite camera-to capture an object infinite camera requires infinite plane)
- Orthographic projection: left submatrix is singular

Any 3 × 4 matrix P with a non-singular left submatrix represents a camera! It can be decomposed as:

- A non-singular upper diagonal matrix K
- An orthonormal matrix R and a vector C with the usual meanings!!
- Therefore, given a 3 × 4 matrix we can recompute K, R and C.

Camera matrix anatomy :

$$P = [p_1 \ p_2 \ p_3 \ p_4] = [p^1 \ p^2 \ p^3]^T$$

- 4-vector C with $PC = 0$ is the camera center.
    - Camera center is the only point with no projection or projects to the vector 0, which is undefined in P^2.
    - In the pinhole camera model, if the point is at the camera centre, then the projected point is undefined.
- Columns p_1, p_2, p_3 are the images of vanishing points of the world X, Y and Z directions.
    - $p_1 = P [1\ 0\ 0\ 0]^T$, the image of the ideal point in X direction and similarly the rest.
    - X-axis and ideal point in direction's projection on projective plane and it's intersection of x vector with image plane to get p_1.
    - First column of the camera matrix is the image of the vanishing point in x-direction. Similarly, for the second and the third column.
- p_4 is the image of world origin: $p_4 = P [0\ 0\ 0\ 1]^T$.

Prove the following :

a. Row vector p^3 is the principal plane :

Principal plane is a plane parallel to the projective plane passing through the origin.

b. Row vectors p^1 and p^2 are axis planes for image Y and X axis respectively :

c. The principal point (or image center) is given by $x_0 = M^*m_3$, with m_3, the third row vector of matrix M.

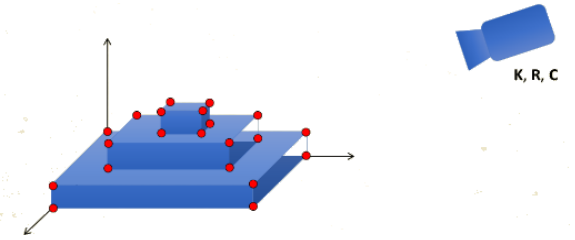
M is the left 3 × 4 submatrix.

Camera calibration :

Given a real camera, we want to compute the camera matrix, R, C, K, focal length, axis scale,..

3D Reference based Methods :

1. Choose a 3D world where we know information about a set of points, points that are easily detectable like corners (can be detected using harris corner detector)



2. Given a set of world points: (X_i, Y_i, Z_i) and their corresponding image coordinates: (x_i, y_i), we can write a set of linear equations in p_{mn}, the entries of the camera matrix.

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

3. We do not know (x_i, y_i). we only know $(x_i/w_i, y_i/w_i)$. Let it be (u_i, v_i)

$$u_i = \frac{x_i}{w_i} = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}}$$
$$v_i = \frac{y_i}{w_i} = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}}$$

on rearranging and solving the linear equation in p, we get :

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i & -v_i \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ . \\ . \\ p_{34} \end{bmatrix} = [0]$$

4. Stack equations for all points to get Gp=0.
5. We have 12 unknown, if the number of points are more than 6, then this is overdetermined.Solving this overdetermined linear system of equations, we can recover the camera matrix. This is camera calibration.
6. The matrix P can then be decomposed into the external and internal parameters: K, R and t.

$$p = K[R|t] \qquad K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

   a. Let $P=[M|p_4]$; where $M = KR$ and $p_4 = Kt$.
   b. $MM^T = KRR^TK^T = KK^T$. We can solve for elements of K .
7. Solving for k :

$$KK^T = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha & 0 & 0 \\ \gamma & \beta & 0 \\ u_0 & v_0 & 1 \end{bmatrix} = MM^T$$

we know M

$$\begin{bmatrix} \alpha^2 + \gamma^2 + u_0^2 & \beta\gamma + u_0v_0 & u_0 \\ \beta\gamma + u_0v_0 & \beta^2 + u_0^2 & v_0 \\ u_0 & v_0 & 1 \end{bmatrix} = \begin{bmatrix} mm_{11} & mm_{12} & mm_{13} \\ mm_{21} & mm_{22} & mm_{23} \\ mm_{31} & mm_{32} & mm_{33} \end{bmatrix}$$

we get u_0 and v_0 directly
compute β from mm_{22}, γ from mm_{12} and α from mm_{11}

8. $R = K^{-1} M$
9. $t = K^{-1} p^4$.
10. Now we have all the intrinsic parameters in K and all the extrinsic parameters, R and t .

Refining P: Non-linear Optimization

P values that we compute maynot be accurate, there may be errors. So we try to do non-linear optimization to minimize the difference between the corresponding product close to 0. In the real world we want the projection matrix of a world point is as close as possible to the observed points i.e., mathematical and physical projection should match.

- The distance metric used in the linear solution is not geometrically meaningful.
- We would like to minimize the distance between the points project by P and the observed points. I.e.,

$$\min\left(\sum_i \left|\left|x_i - \varphi(P, X_i)\right|\right|\right)\Big|_p$$

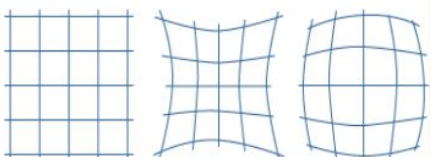
- Can be solved by the Levenberg-Marquardt algorithm.
- Use the linear solution as a starting point.

Dealing with Radial Distortion :

Radial distortion will be there since we capture using the lens. We calculate these radial distortion parameter, so that radial distortion parameters disappear.

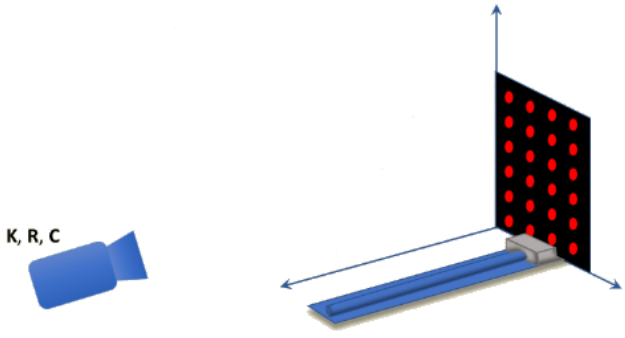
- Each pixel moves radially away from (barrel) or towards (pincushion) the image center (c).
- As a function of squared distance from c : $r_c^2 = x_c^2 + y_c^2$
- The shift g can be modelled as: $\gamma = 1 + k_1 r_c^2 + k_2 r_c^4$, where k_1 and k_2 are radial distortion parameters.
- The modified coordinates are:

$$\hat{x} = \gamma x_c$$
$$\hat{y} = \gamma y_c$$



- This is applied before the focal-length multiplier and center shift are applied.

Calibration from a precisely moving plane :

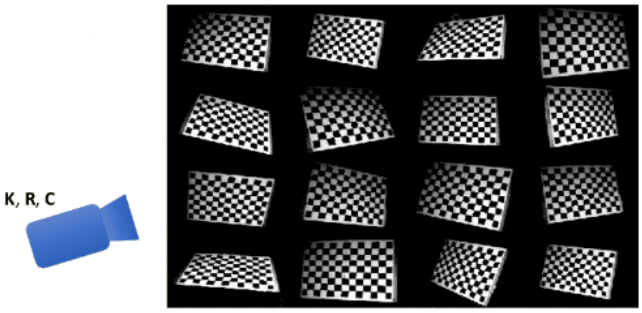


- We have a plane with a set of points (easily detectable, and precise)
- Put it on a linear actuator, move it very precisely.
- Measure these movements to capture the 3D movements.
- Here we have a bunch of precise 3D points to compute camera parameters.

Note :

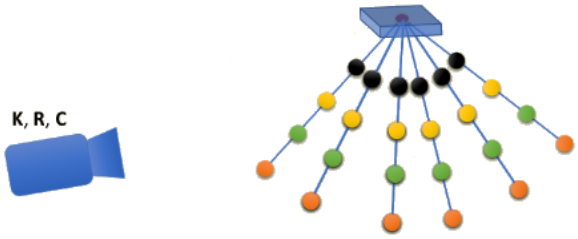
This is a very robust method, equations will not go into numerical degeneracy. This results in an accurate and precise camera calibration matrix. Precision in the plane is really important.

Calibration using a Plane with Unknown motion :



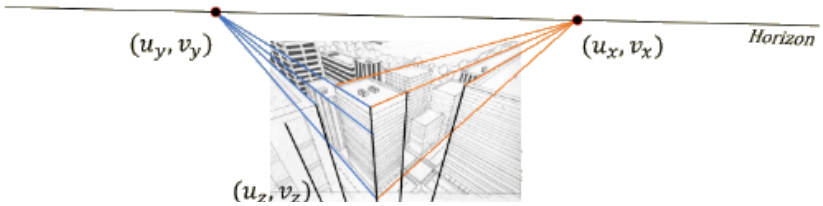
We create a plane and we move the plane at random angles. We compute the camera parameters. All these points are coplanar. OpenCV can compute this.

Calibration from a set of collinear points that moves such that the lines passing through a fixed point :



A line and a set of collinear points on the line, we move the line such that all the points pass through a single point. All these images taken help us compute the camera calibration matrix.

Calibration from Vanishing points in orthogonal directions :



We can use vanishing points, corresponding to the columns of the matrix. We know that the world origin image is the fourth column. 2 vanishing points in horizontal direction parallel to the ground plane. Line passing through these two points is the horizon line (line at infinity)

Self Calibration (rigid static world and point correspondences across images are available) :

If we assume that the world is rigid. We can recover camera parameters from a set of world images. This does not assume about how the images are taken. This has a very little restriction about the world.

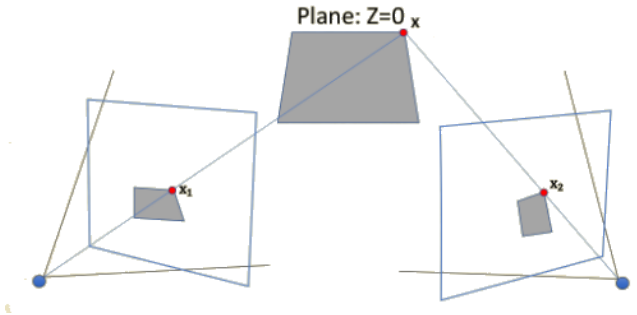
Multiple View geometry :

Two View geometry :

Geometry involved when we look at the world from two different points of view.

Case 1 - Planar World :

Consider a specific point x in the world z=0. Projecting x on to the camera centre from 2 different views result in x_1 and x_2.



Relation between x_1 and x_2 :

Projection equation of points on a plane :

(point is on the z=0 plane, so r_3 does not play any role).

$$x = K[R \quad t]X = K[r_1 \quad r_2 \quad r_3 \quad r_4]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K[r_1 \quad r_2 \quad t]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = HX$$

(w.k.t all the rows in R are independent of each other, they are orthonormal). H is a 3 × 3 non-singular matrix. Since this planar we are able to represent 3 × 3 camera matrix as 3 × 4.

Two different views of the same world point :

$$x_1 = H_1X \qquad x_2 = H_2X$$

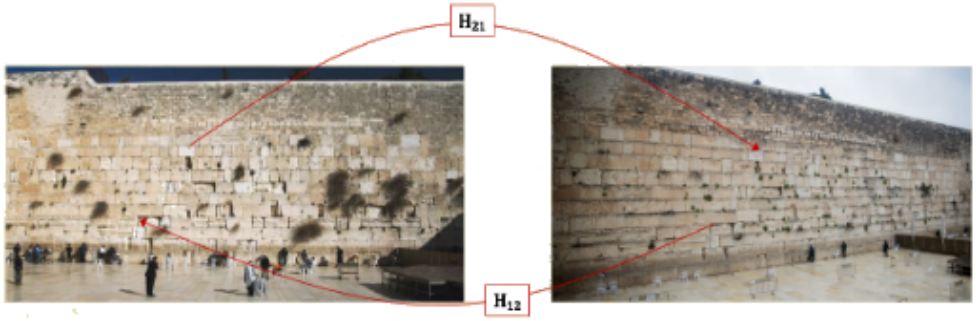
Combining these both,

$$x_2 = H_2H_1^{-1}x_1 = H_{21}x_1$$

This tells us that every pixel in image 1 has a corresponding pixel in image 2. We can transform points from image 1 and image 2 and vice versa.

$$x_1 = H_{12}x_2 \quad \text{and} \quad x_2 = H_{21}x_1$$
$$H_{12}H_{21} = I$$

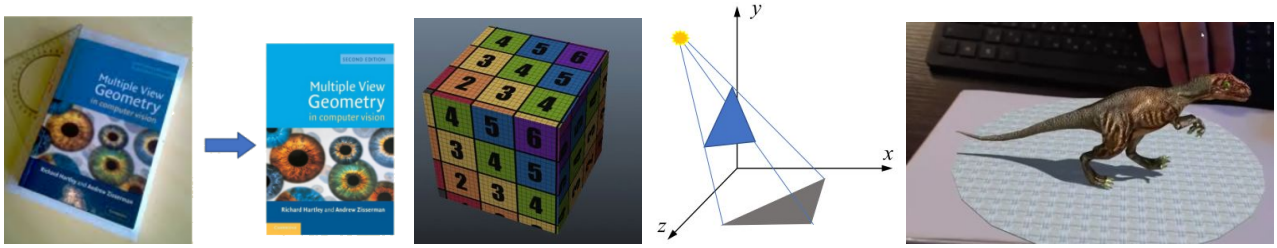
H_1, H_2, H_{12} and H_{21} are homographies (image to world or image to image)



We just need to know 4-5 points in both images to compute homographies.

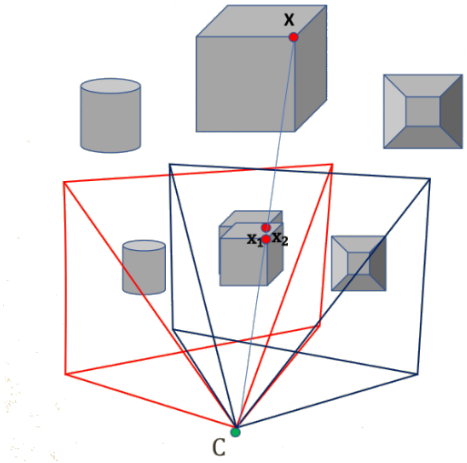
Planar Homography - Applications :

- Removing perspective distortion.
- Rendering planar textures
- Rendering planar shadows.
- Estimating camera pose ; AR
    - We use a plane (we know the transformation between the world plane and camera - we know rotation and translation of the object from that plane), so we know extrinsic parameters or pose of the camera.
    - AR games



Case 2 - Same Camera Centre :

Here, we deal with the 3D world. We assume that the camera centre for two different viewpoints remain the same (we can rotate and zoom the camera, but cannot translate).



Projection equation for two cameras with same C :

Let image points be x_1, x_2 and rotations be R_1 and R_2.

$$\begin{aligned} x_1 &= K_1R_1[I \quad -C] \\ x_2 &= K_2R_2[I \quad -C] \\ &= K_2R_2(K_1R_1)^{-1}K_1R_1[I \quad -C] \\ &= K_2R_2(K_1R_1)^{-1}x_1 \\ &= H_{21}x_1 \end{aligned}$$

R,K and H_{21} are 3×3 matrices. H is a 3×3 non-singular matrix

$$x_1 = H_{12}x_2 \quad \text{and} \quad x_2 = H_{21}x_1$$
$$H_{12}H_{21} = I$$

Homography exists between the images if :

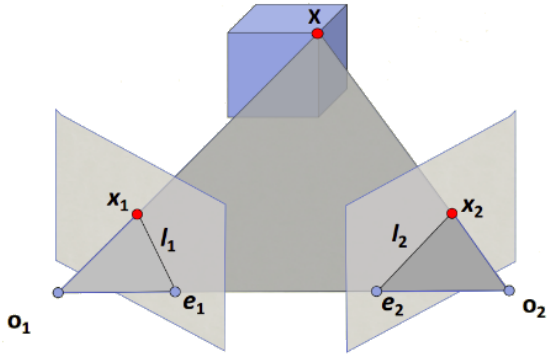
- World is planar (in this case homography between world and image also exists)
- Camera centre is the same

Applications :

- Image stitching/mosaicing
- Detecting the camera translation
- Multi-frame Super-resolution

Epipolar Geometry :

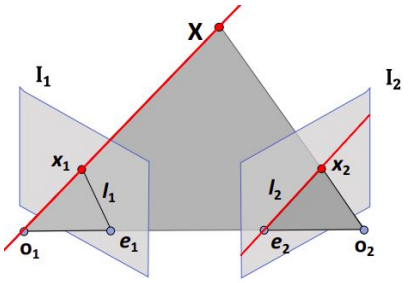
o_1 and o_2 are camera centres, x_1 and x_2 are image projections corresponding to point x. This does not have homography, a generic case. Points of intersection of line joining the camera centres with the imaging planes are e_1 and e_2. o_1o_2x form a triangle (so all these three points are planar). The plane in which triangle lies intersects image planes at lines l_1 and l_2. So, line l_1 and l_2 pass through epipoles e_1 and e_2.



The geometry that relates these two views is Epipolar Geometry.

We know that line joining o_1 and x_1 will pass through x.

- All the points that map to x_1 (pre-image of x_1) in I_1 will map to l_2 in image 2 and vice versa. This line is an epipolar line. Epipolar line depends on the image point.



- Image of o_1 in I_2 (e_2) is an epipole. So is e_1.
- Plane containing these is the epipolar plane.
- This results in a set of constraints, which are referred to as the epipolar constraints and the resulting geometry is called the epipolar geometry.

Epipolar constraint :

- Consider X in camera 1's coordinates assuming origin is at o_1:
  - $\lambda_1 x_1 = X$
- Viewing it in camera 2's coordinates, but we summed that origin is at o_2 :
  - $\lambda_2 x_2 = RX + T$
  - $\lambda_2 x_2 = R(\lambda_1 x_1) + T$
- Premultiplying by $\widehat{T}$, and then by x_2^T, we get :

$$\hat{T}\lambda_2 x_2 = \hat{T}R(\lambda_1 x_1) + \hat{T}T$$
$$\text{we know that } \hat{T}T = 0,$$
$$\hat{T}\lambda_2 x_2 = \hat{T}R(\lambda_1 x_1)$$

$$\text{on multiplying with } x_2^T, \text{ we get :}$$
$$\lambda_2 x_2^T \hat{T} x_2 \;=\; \lambda_1 x_2^T \hat{T} R(x_1) + 0$$
$$\hat{T}x_2 \text{ is } \perp \text{ to both T and } x_2, \text{ so } \lambda_2 x_2^T \hat{T} x_2 \;= 0$$
$$\lambda_1 x_2^T \hat{T} R(x_1) \;=\; 0$$
$$x_2^T E x_1 \;=\; 0 \text{ or } \; x_1^T E^T x_2 \;=\; 0$$

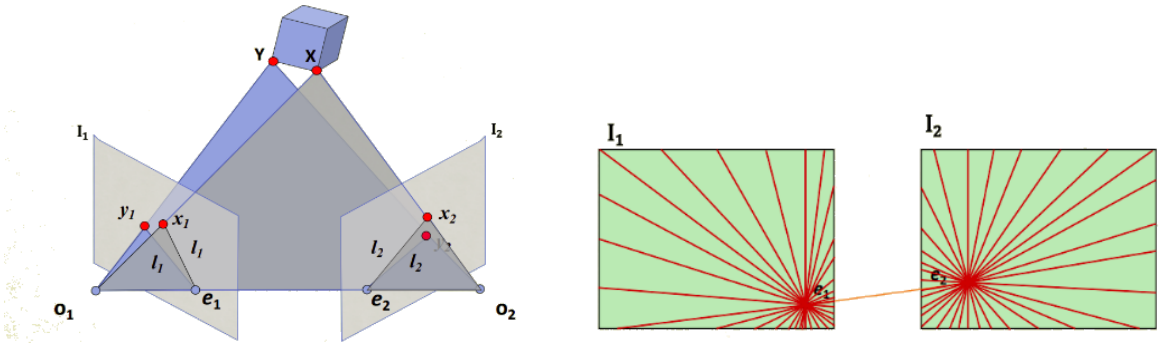
- Here, we assumed that world origin is at o_1. We ignored focal lengths, inter-pixel distance,... and other standard simplification assumptions.
- If we have intrinsic parametric matrix, let them be K_1 and K_2 :

$$x_1 = K_1 X$$
$$x_2 = K_2 X$$
$$(x_2^T K_2^{-T}) \hat{T} R (x_1 K_1^{-1}) \;=\; 0$$
$$x_2^T F x_1 \;=\; 0 \text{ or } \; x_1^T F^T x_2 \;=\; 0$$

This is a weak calibration case, we considered focal lengths,...

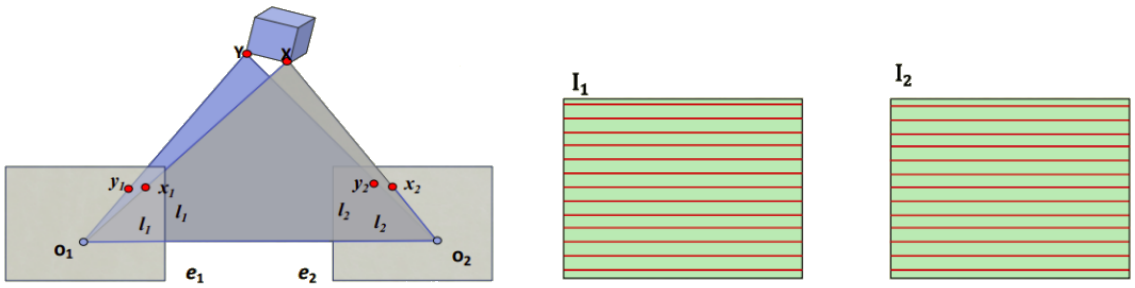
- Both essential and fundamental matrices are 3 × 3 and are independent of the world point.

Epipolar Lines :



When we move the world point from X to Y epipolar lines change. If we keep moving the world point then all the epipolar lines will move up if the world point is moved up and vice versa, but all of them pass through the same epipole and the equation of this line is $x_1^T F x_2 = 0$.

If two cameras are coplanar (principal axis are parallel to each other) :



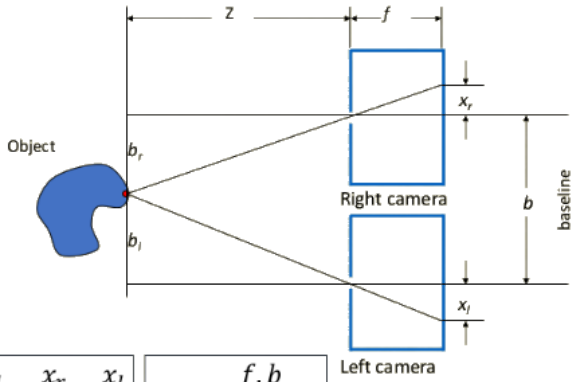
If two imaging planes are coplanar, then corresponding o_1 and o_2 are parallel to imaging planes. So, they intersect at infinity. Epipoles are at infinity. All the epipolar lines are parallel.

Depth from Stereo :

Perceiving depth using two images of the same scene with different baseline.

Stereo :

1. We see slightly different images of the world through the two eyes. This difference in images gives us depth/stereo perception.
2. This shift in image is proportional to the distance to the object. (object speed:image speed) depends on the image depth. Closer object's image shifts by a larger amount compared to the image of the farther object.



$$\frac{b_r}{Z} + \frac{b_l}{Z} = \frac{x_r}{f} + \frac{x_l}{f} \qquad Z = \frac{f.b}{(x_r + x_l)}$$

b is base line, f is focal length. We can compute the depth (Z) using the difference in location of the images of the point in the two cameras. Difference in image location in this case is x_l+x_r . When disparity increases (search becomes difficult) , x_l+x_r also increases. Therefore, Z decreases (closer object) and vice versa. For the object at infinity, disparity is 0. Larger baselines give reliable depth estimates, but the matching can become harder.

Stereo Geometry (Calculating corresponding points in pair of images) :

- Find a world point in 2 or more views
- Appearance is the only clue to identify them
- Individual pixel colors are similar very often. However, the match is too noisy.
- Match a (small) neighborhood of colors (patch match of the features extracted) from one image to a similar neighborhood in others.
- Will work if the local surface is fronto-parallel and images have similar magnification (for a surface with high oblique angle, a small shift can cause a big change in appearance, making the image matching of a point difficult).
- Foreshortening can happen when viewing an oblique surface.
- Here we assume that camera parameters of both the cameras are approximately the same.
- Many ambiguities. We need a lot of help!

Difficult to match :

Convenient to match :

Matching Patches :

- Compare m×m patches from two views; Form vectors v and v′ of length m^2 using concatenation.
- Matching scores between patches:
  - Sum of Absolute Difference (SAD) : $||v - v'||_1$
  - Sum of Squared Difference (SSD) : $||v - v'||_2$
  - Normalized correlation:

$$\frac{(\bar{v}')^{\mathrm{T}}\bar{v}}{\sqrt{\bar{v}^{\mathrm{T}}\bar{v}}\sqrt{\bar{v}'^{\mathrm{T}}\bar{v}'}} \quad : \text{Range} : [-1,1]$$

where $\bar{v}$ and $\bar{v}'$ are vectors with respective patch-mean colour subtracted. Invariant to affine changes in intensity/colour.

- Census Transform (represent a pixel in terms of its neighborhood, i.e., larger or smaller compared to its neighbor)

$$\begin{array}{|c|c|c|} \hline 124 & 74 & 32 \\ \hline 124 & 64 & 18 \\ \hline 157 & 116 & 84 \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|} \hline 1 & 1 & 0 \\ \hline 1 & x & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \longrightarrow 11010111$$

- Birchfield-Tomasi Match (fractional pixel match by interpolation).

Constraints on Matching :

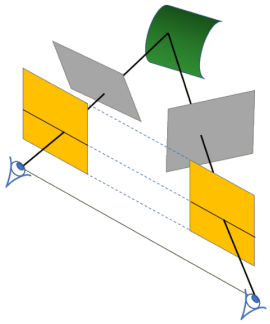
- Epipolar : Match lies on the epipolar line of the pixel
  - When we are creating a stereo rig where we have two cameras to capture, we mount cameras such that the image planes are parallel to each other.
  - If the image planes are not parallelly aligned then we do image rectification.
- Colour Constancy : The appearance/colour does not change from one view to another
- Uniqueness : A point on left image can match with only one point on the right and vice versa
- Ordering or Monotonicity: If point A is to the left of B in view 1, it will to the left of B in view 2 also. (Violated if great difference in depth exists)
- Continuity : Disparity values vary smoothly (violated at occlusion boundaries)
- Sparse correspondence : only for good feature points
- Dense correspondence : a match for every pixel

Epipolar : Reduced Search and Rectification :

Search :

- Search is limited to a line if fundamental matrix is known (weakly calibrated)
- Simplest if left and right cameras have the same image plane and pure X-translation between them.
  - Fundamental matrix has a simple form and the Epipolar constraint reduces to y′ = y.
  - Matches constrained to lie in the same scan line.
  - Y-translation also has the lines parallel but at an angle.

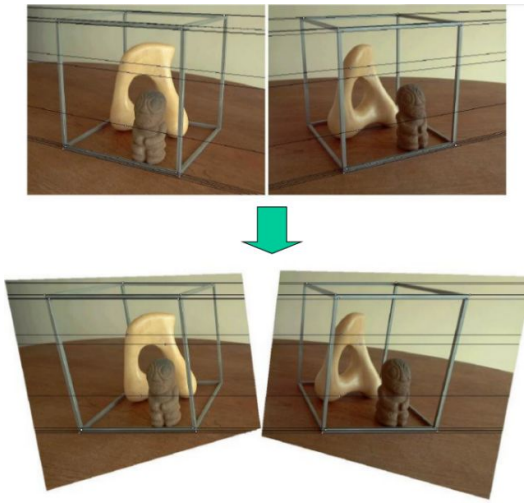
Rectification :



A rotation of the camera (to make image planes parallel) and a change in K matrix (focal length, image center). Converts the images as if they were taken with the cameras with pure x translation between the images.

- Can be represented using a homography H to align one image plane to the other.
- Or, homographies H_1 , H_2 to align them to a third plane.
- Reproject images onto a common plane parallel to the line between the optical centers.
- Pixel motion is horizontal after this transformation.
- This becomes a string matching problem.
    - It can also be done using dynamic programming.
- Correctness improves

Example :



- We apply homography by first estimating how much we should translate and rotate using stereo calibration.
- Images now are no longer rectangular, we notice the epipolar lines are parallel to each other.
- Now, (i,j) pixels in both pictures are matched for different columns.
- This becomes a string matching problem.
    - It can also be done using dynamic programming.
    - Ordering constraint : if the j^{th} column matches with j'^{th} then $j+1^{th}$ has to match with $j'+1^{th}$.
- We need to define a nice metric for comparison.

Stereo vs. Optical Flow :

Optical flow is also used for image matching/pixel matching. Optical flow is usually used if there is some movement in the world or camera is moving in some unknown direction i.e., there is no calibration between the images. But epipolar constraint is still valid here.

- Calibrated vs. Uncalibrated cameras
- 1-D (because of epipolar constraints) vs. 2-D Matching
- Occlusions are handled (marked) explicitly in optical flow.
- Disparities are quantized, allowing special optimization techniques (DP, Graph Cuts) in stereo.
- Special constraints on matching in stereo.
- Simultaneous vs. sequential image capture.