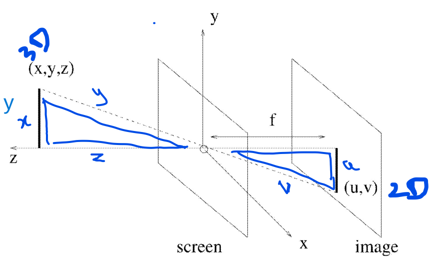
Pinhole camera model

its okay to assume x and y alternatively, doesnt matter



We try to establish a relationship between 2D and 3D, there are 2 triangles of same ratio

The farther (z is larger) the object, the smaller the image will be (u and v are inversely proportional to z). If the object is bigger (x,y larger), then the image will also be bigger(u and v are directly proportional to (x,y). The focal length of the lens is the distance f here between the image and camera. Camera with high focal length magnifies the object

The relationship here is non-linear (division by z).

A linear relationship is one where one value can be expressed as the weighted sum of the other (like linear combo - coeff times of the other variables)

What is the benefit of linearity?
If there are multiple linear combos , we can combine all the transformations as one

The below equations helps to convert 3d point (x,y,z) to 2d (u,v)

$$\frac{x}{z} = -\frac{u}{f} \qquad u = - f\ (x/z)$$

$$\frac{y}{z} = -\frac{v}{f} \qquad v = - f\ (y/z)$$

-ve sign to show that image is inverted

Pinhole camera model

In matrix form:

$$\underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\text{image}} = \frac{1}{z}\begin{bmatrix} -f & 0 \\ 0 & -f \end{bmatrix}\underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\text{world}}$$

Without inversion:

Based on triangle similarity:

Assuming that somebody rotated the image and no need of -ve sign

$$\frac{x}{z} = \frac{u}{f}$$

$$\frac{y}{z} = \frac{v}{f}$$

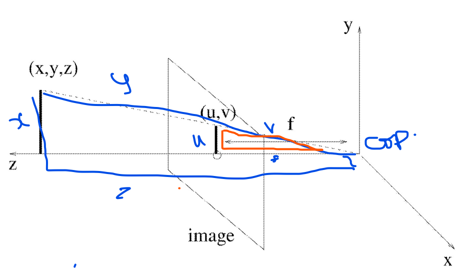
converted 3d to 2d $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \to \begin{bmatrix} u \\ v \end{bmatrix}$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{z}\begin{bmatrix} +f & 0 \\ 0 & +f \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

Alternative pinhole camera model

$\frac{x}{z} = \frac{u}{f}$

$\frac{y}{z} = \frac{v}{f}$



Here, there is no screen . but the image is created by projecting the points (x,y,z) to the Center of projection(COP) which represents the camera.

The picture is not inverted here, because we see that there is no inverted triangle and projector doesnt invert it

In matrix form:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{z}\begin{bmatrix} f & 0 \\ 0 & f \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

To convert non linear to linear equation

Homogeneous coordinates

Moving from Cartesian to homogeneous coordinates:

2D → 2DH

$$(x, y) \leftrightarrow (x, y, 1)$$

3D 3DH

$(x, y, z) \Leftrightarrow (x, y, z, 1)$

Equivalence class:

$$(x, y, 1) \leftrightarrow (kx, ky, k)$$

Every 2d point has multiple 2DH points. This doesnt change the 2d point

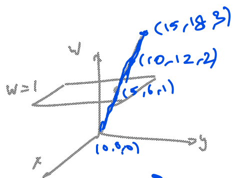
Moving from homogeneous to Cartesian coordinates:

2DH → 2DH → 2D

$$(x, y, w) \leftrightarrow (x/w, y/w, 1)$$

$(x/w, y/w)$

similarly, to move back to 2d, we can divide by the third coordinate to get 2d point and it does not change the position of the 2d point

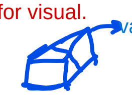


$(x, y, 0)$ = point at infinity represents direction

Any 2dH point has an equivalent 2d point but not (x,y,0) because when we try to apply the algorithm (/ by 3rd coordinate), x/0,y/0, will result in infinity. Thus , this point in 2dh is called point at infinity and it represents a direction rather than a point.

This infinity point can be seen in images and artistic 3d drawings as vanishing points. Any 2 parallel lines will be shown as meeting somewhere to create the perspective of 3d.

Refer to the word doc (notes 1) picture for visual.



vanishing point in cube

A point in 2d represents a ray in 2dh
why the ray starts at (0,0,0) for any 2d point? --->because we can multiply by any k to get 2dh and k=0 here
so, for (5,6,1) , we can interpret as the 2d point (5,6) in the plane w=1
for (10,12,2), the 2d point (10,12) will be in the plan w=2

There can be infinite number of vanishing point in an image. Every vanishing point is a projection of the direction.

Projection in homogenous coordinates

Perspective projection in homogeneous coordinates:

$(\alpha u, \alpha v, \alpha)$

$$\underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{2D} = \frac{1}{z} \begin{bmatrix} f & 0 \\ 0 & f \end{bmatrix} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{2D}$$

$$\underbrace{\alpha \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{2dH} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{3d}$$

$$u = \frac{\alpha u}{\alpha} = \frac{fx}{z}$$

$$v = \frac{\alpha v}{\alpha} = \frac{fy}{z}$$

2dh (the point can be multiplied by any constant)
the 2d point is (u,v)....2dh is $(\alpha u, \alpha v, \alpha)$. we have actually achieved the linearity by postponing the division of z to later. The division is anyways present, just that it is not there at the moment of matrix calculations allowing to combine the transformations

$(\alpha u, \alpha v, \alpha) = (fx, fy, z)$

get point → $(u, v) = \left(\frac{fx}{z}, \frac{fy}{z}\right)$

Balanced pinhole model:
make it all homogenous

$$\underbrace{\alpha \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{2DH} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{Q,\ 3 \times 4} \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{P,\ 3DH}$$

Projection in homogenous coordinates

* Matrix form:

$$p = QP$$

Block notation

$$Q = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [\underbrace{K}_{3*3} | \underbrace{0}_{3*1}] = K[I|0]$$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$p = K[I|0]P$

means 0 translation of camera

internal camera parameter that doesnt change

means no rotation

Pls pls pls refer second image in Notes document for clarity:

For Now, we used only 1 coordinate system (that is the camera coordinate system) for looking at the world axis, image axis and the camera coordinate system
But, it would be hard if we just keep it that way, because whenever we move the camera, it means we change the coordinate system as well.
So, we need separate coordinate system for world, image and camera. We ideally wanted to relate the world points to the image points/pixels.
To be able to do this, we should find relation based on what all can be done with the camera to eliminate its coordinate system.
We can change the camera position(translation), rotate the camera(rotation) and change mm to pixels in the image (Scale)

Transformations in homogenous coordinates

cv2. warp affine function takes a input vector multiples with 4x4 matrix and gives an output

Affine transformations in homogeneous coordinates: 3d affine matrix - 4x4 matrix

Affine matrix has 0001 in last row to ensure we convert 2dh to only 3dh.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{17}$$

The matrix H is a combination of rotation, translation, and scale transformations combined by multiplication.

$$p' = Hp \tag{18}$$

What is affine transformation matrix -4x4 - it is based on the multiplications of the rotational, translational and scaled transformation matrices. So, any changes to the camera are matrices which are multiplied and then we reach the affine transformation matrix. This captures all the changes in the camera and helps to find the appropriate changed 3dh point from the original 3dh point.

This affine transformation matrix is not the projection matrix which we used before K [I|0]

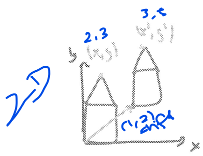
3D Translation:

4x4 3D

$$T(t_x, t_y, t_z) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} I & T \\ 0^\top & 1 \end{bmatrix} \tag{19}$$

3x3 3x1

1x3 1x1



$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases} \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \iff \underbrace{\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}}_{2dh} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{2dh}$$

2D

For easy transformation matrix, we want the matrix addition be changed to multiplication with the help of the homogenous coordinates

Affine transformation -->works on linear matrices

Transformations in homogenous coordinates

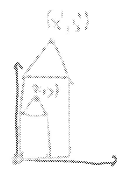
3D Scaling about the origin:

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S & 0 \\ 0^\top & 1 \end{bmatrix}$$

Eg - a 3d point - (x,y,z) = (1,2,3). scale it by 2.

The scale matrix ,S= $\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

2D



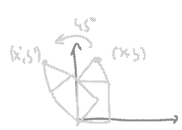
$$\begin{cases} x' = s_x \cdot x \\ y' = s_y \cdot y \end{cases} \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Now we do S times the 3d point , we have to use the 3dh point for multiplication

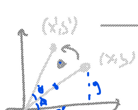
$$S \underbrace{\begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}}_{\text{3dh point}} = \underbrace{\begin{bmatrix} 2 \\ 4 \\ 6 \\ 1 \end{bmatrix}}_{\text{scaled 3dh point}}$$

Rotation matrix - positive rotation is counterclockwise

Transformations in homogenous coordinates



Given (x,y), how to find x',y'



$(x', y') \longrightarrow (r, \phi + \theta)$

$(x, y) \longrightarrow (r, \phi)$

usage of polar coordinates. the point are rotated by the origin

$$\begin{cases} x = r\cos\phi \\ y = r\sin\phi \end{cases} \qquad \begin{cases} x' = r\cos(\phi + \theta) \\ y' = r\sin(\phi + \theta) \end{cases}$$

cos(a + b) = cos(a)cos(b) – sin(a)sin(b)
sin (a + b) = sin a cos b + sin b cos a

$$x' = \overbrace{r\cos\phi}^{x}\cos\theta - \overbrace{r\sin\phi}^{y}\sin\theta = x\cos\theta - y\sin\theta$$

$$y' = \underbrace{r\cos\phi}_{x}\sin\theta + \underbrace{r\sin\phi}_{y}\cos\theta = x\sin\theta + y\cos\theta$$

Write these equations in matrix form

Transformations in homogenous coordinates

2D

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2DH

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Given a point(x,y) and the rotation angle
Cv2. rotationmatrix function - >calculatoes the R(theta) matrix
cv2.warp affine function -->calculates the matrix transformation by applying the R matrix

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos 60^\circ & -\sin 60^\circ & 0 \\ \sin 60^\circ & \cos 60^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 1 \end{bmatrix}$$

example - rotate (4,5) by 60 degrees

Transformations in homogenous coordinates

3D Rotation about the z axis:

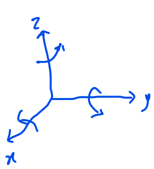
$$R_z(\theta) = \left[\begin{array}{ccc|c} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} R_z & 0 \\ 0^T & 1 \end{bmatrix}$$

3D Rotation about the x axis:

$$R_x(\theta) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} R_x & 0 \\ 0^T & 1 \end{bmatrix}$$

3D Rotation about the y axis:

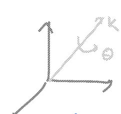
$$R_y(\theta) = \left[\begin{array}{ccc|c} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} R_y & 0 \\ 0^T & 1 \end{bmatrix}$$



Transformations in homogenous coordinates

Rotating the camera need not be neccessarily with respect to x,yz. It can be any new axis k.
k - axis

$$R_k(\theta) = R_x R_y R_z$$



k vector with k axis coordinates

$$k = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$$

Rodrigues formula:

$$\boxed{\underset{3\times3}{\mathbf{R}} = \mathbf{I} + (\sin\theta)\underset{3\times3}{\mathbf{K}} + (1-\cos\theta)\underset{3\times3}{\mathbf{K}^2}}$$

In above topics, we saw the rotation matrix for 3dh.
Here, we define the 3d matrix using Rodrigues formula

3D

$$\mathbf{K} = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}$$

$$R_k(t) = \left[\begin{array}{c|c} R & 0 \\ \hline 0^T & 1 \end{array}\right]$$

3DH

cv.rotationmatrix - Given an axis of rotation (k) and an angle, we get the rotation matrix in 3d

The axis of rotation should be a unit vector

This K is a matrix with
k vector elements

skew symmetric matrix---> $A^T = -A$