Computer Vision and Machine Learning
(Visual system and Camera)

Bhabatosh Chanda
chanda@isical.ac.in

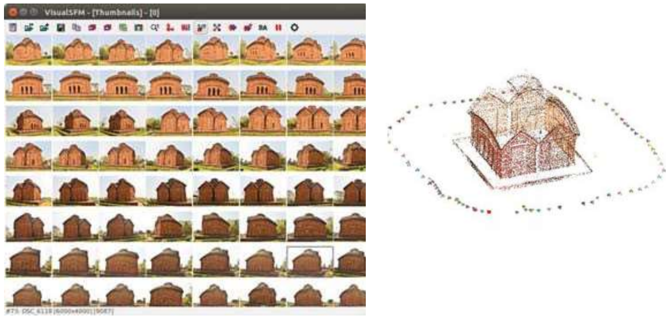
Image formation

- Basic components
  - Light (Wave length, intensity)
  - Camera (intrinsic and extrinsic parameters)
  - Scene (reflectance, orientation, curvature)

Shape from X

- Reconstructing 3D object from 2D images
  - Stereo
  - Motion
  - Shading
  - Texture
  - Contour

Geometry: 3D reconstruction (cond.)



Geometry: 3D reconstruction

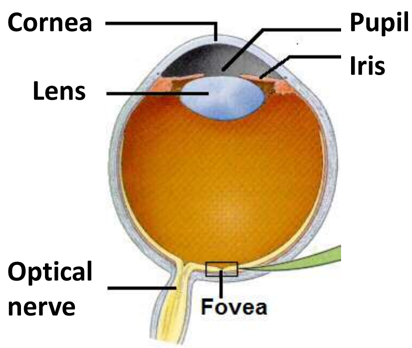
Motion detection and tracking

Human Visual system and Camera

Human vision

- Most important and powerful sensor in human body.
- The image formation process is well understood.
- The image understanding is the one that remains mysterious.

Human eye structure



Pathways to the brain

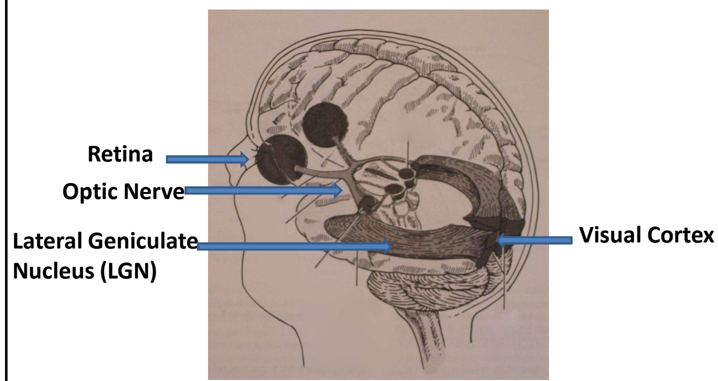
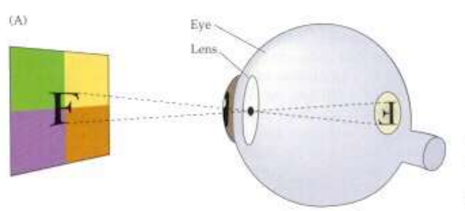
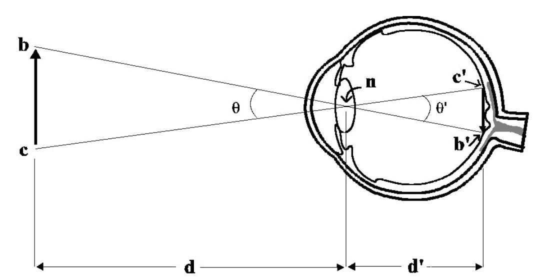


IMAGE FORMATION



Camera model for eye

PERSPECTIVE PROJECTION

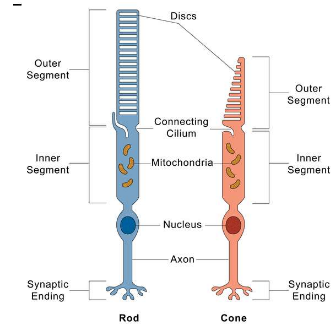


$bc/d = b'c'/d'$ **θ is the visual angle**

Light sensors in the eye

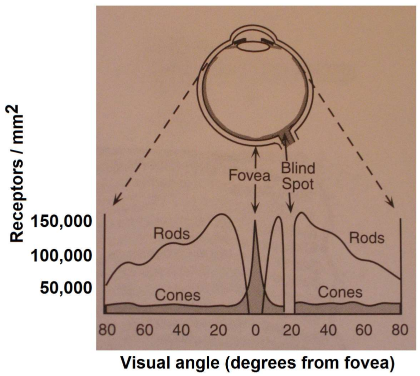
- Image formed on the retina is sensed by special type of cells called ***photoreceptors***.
- Photoreceptors convert physical light signal into biological signal that passes through optical nerves.
- There are two types of photoreceptors:
  - Cones
  - Rods

Rod and cone cells in eye



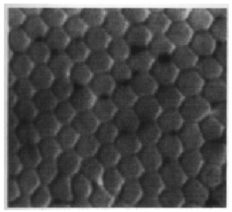
- ***Cones:*** Color-vision with higher intensity light.
- ***Rods:*** Low-intensity light vision, e.g. night vision.
- Cones are larger in size and are more densely packed in fovea region.
- Rods are smaller in size, much higher number in periphery.

Receptor cell distribution

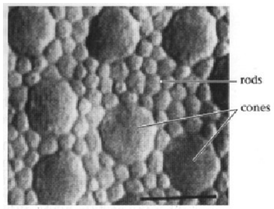


Cell packing at fovea and periphery

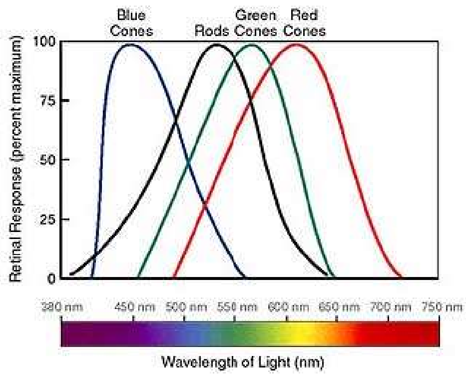
Receptor cells at fovea



Receptor cells at periphery



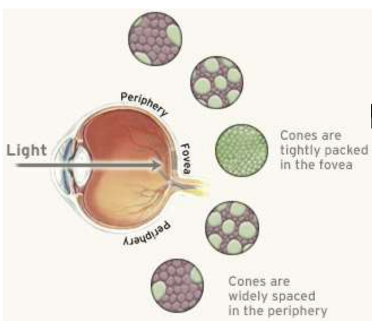
Color sensitivity of receptors

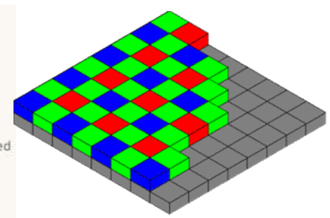


Colour sensitivity

- Wavelength of visible spectrum ranges from 0.38 μm to 0.76 μm (approx.)
- Response characteristics $h_B(v)$ for blue attains maximum at about 6.8 x 10^4 Hz or 0.44 μm
- Response characteristics $h_G(v)$ for green attains maximum at about 5.8 x 10^4 Hz or 0.52 μm
- Response characteristics $h_R(v)$ for red attains maximum at about 4.3 x 10^4 Hz or 0.70 μm

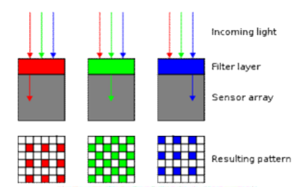
Retina and camera sensor

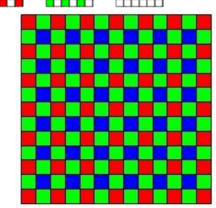




Bayer filter

- Bayer filter is a color filter array used with sensor of digital camera.
  - Arranged in square grid with 50% G, 25% R and 25% B.
  - Each pixel gets one particular color that the filter allows to reach the sensor.
  - Other color components are interpolated (averaged) from the neighborhood.





Bayer filter

Human eye: Summary

Frontal part
- Cornea
- Pupil and Iris
- Lens

Retina:
- Rods (low-intensity light, night vision)
- Cones (color-vision)
- Synapses and ganglions
- Optic nerve fibers

Does sensing and low-level processing.

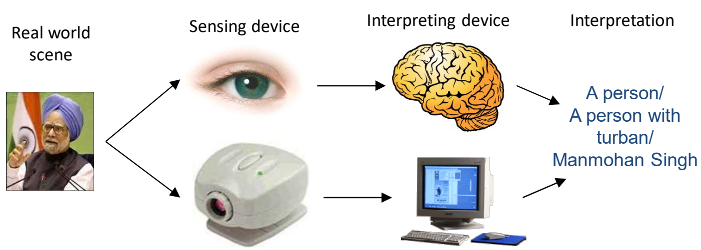
Human vision *vs.* Computer Vision

The camera replaces the eye:
- Eye lens → Camera Optics
- Cones and Rods → CCD array
- Ganglion cells → Filter banks

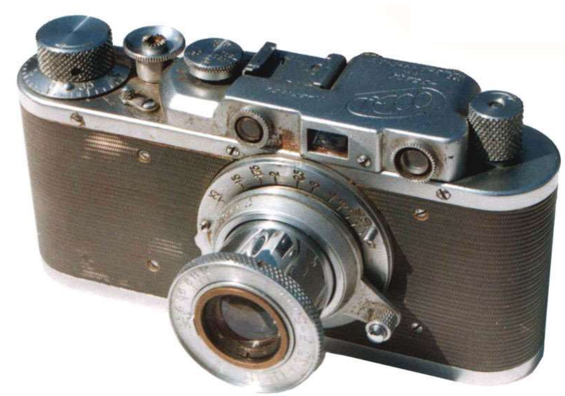
The computer replaces the brain:
- But how?

The Objective

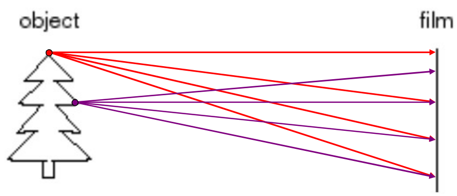
- Want to make a computer understand images
- We know it is possible – we do it effortlessly!



Camera model



How does a camera see the world?

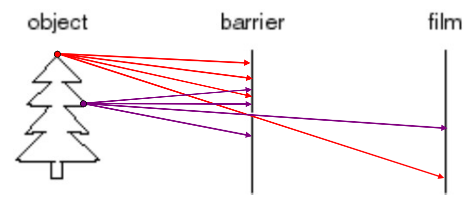


How a camera works:

- Put a piece of film in front of an object
- Light radiates from object in all possible directions.
- Do we get a reasonable image?

Slide inspired by by Steve Seitz

Pinhole camera

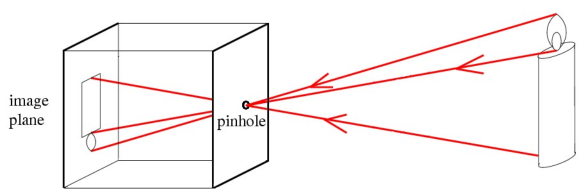


Add a barrier with a small hole to block off most of the rays

- This reduces blurring
- The opening known as the **aperture**

Slide inspired by by Steve Seitz

Pinhole camera model

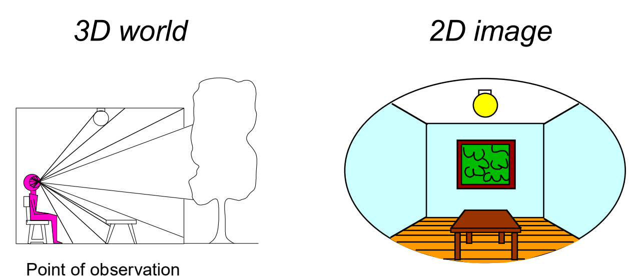


Pinhole model:

- Captures **pencil of rays** – all rays through a single point
- The point is called **Center of Projection**
- The image is formed on the **Image Plane**

Slide by Steve Seitz

Dimensionality Reduction Machine (3D to 2D)



What have we lost?

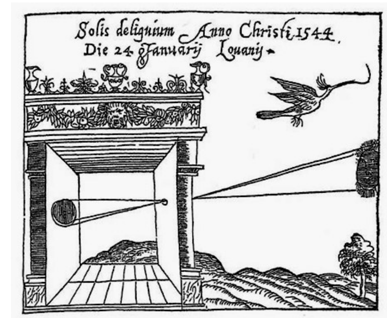
- Distances (lengths)
- Angles

Slide by A. Efros

Projection properties

- Many-to-one: any point along same *visual ray* map to same point in image
- Point → point
- Line → line (collinearity is preserved)
  - But line through focal point (visual ray) projects to a point
- Plane → plane (or half-planes)
  - But plane through focal point projects to line

Camera Obscura

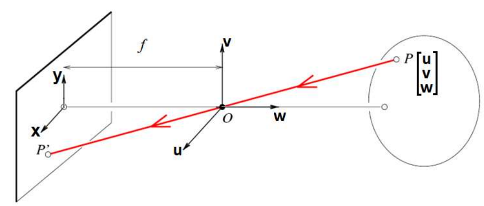


- Basic principle known to Mozi (470-390 BCE), Aristotle (384-322 BCE)

Camera Obscura

Illustration of the *camera obscura* principle from James Ayscough's *"A short account of the eye and nature of vision"* (1755 fourth edition)

Modeling the projection

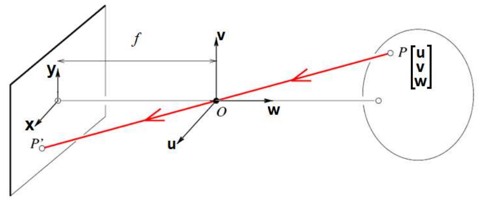


The coordinate system

- Optical center or center of projection ***O*** is at the origin
- Optical axis is in w direction
- The image plane (xy-plane) is parallel to uv-plane (perpendicular to w axis)

Source: J. Ponce, S. Seitz

Modeling projection



Projection equations

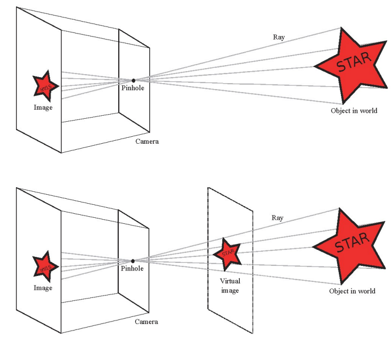
- Ray from ***P(u,v,w)*** through ***O*** intersects image plane at P'
- Derived using similar triangles

$$(u, v, w) \rightarrow (-f\frac{u}{w}, -f\frac{v}{w}) = (-x, -y)$$ **(Perspective projection)**

3/6/2023 33

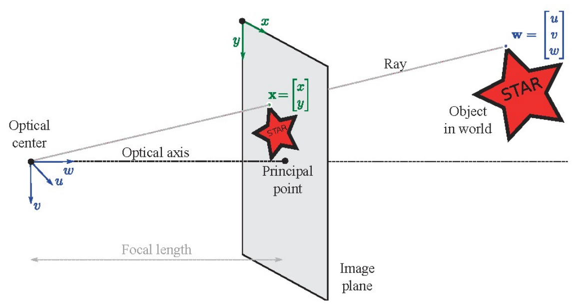
Source: J. Ponce, S. Seitz

Negative to positive



3/6/2023 34

Image and scene coordinate

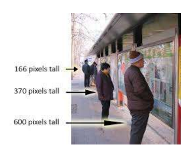


$$(u, v, w) \rightarrow (f\frac{u}{w}, f\frac{v}{w}) = (x, y)$$ **(Perspective projection)**

3/6/2023 35

Properties of Perspective projection

1. A straight line in 3D maps to a straight line in 2D.
2. Distant objects appear smaller.
3. A set of parallel lines in 3D (not perpendicular to optical axis) maps to a set of concurrent lines in 2D.
   – The common point in 2D is called ***vanishing point***.



3/6/2023 36

Vanishing points

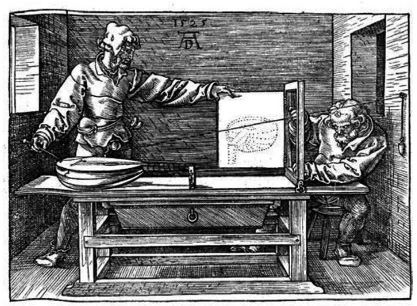
- Each direction in 3D has its own vanishing point.
  - All lines going in that direction converge at that point
  - Exception: directions parallel to the image plane
- All directions in the same plane have vanishing points on the same line.

Perspective distortion

- Problem for architectural photography: converging verticals

Source: F. Durand

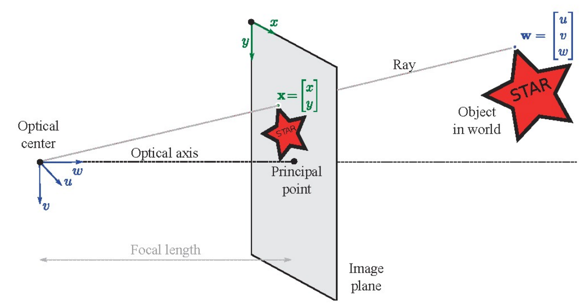
Perspective projection for artist



- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)

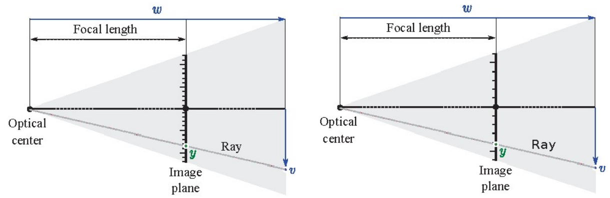
(https://blogs.scientificamerican.com/roots-of-unity/the-slowest-way-to-draw-a-lute/)

Image and scene coordinate



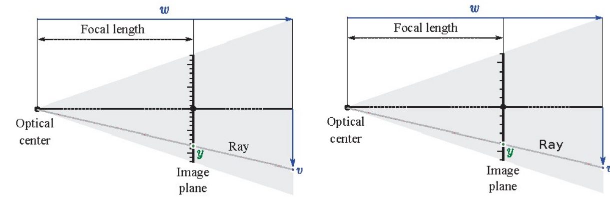
$$(u, v, w) \rightarrow (f\frac{u}{w}, f\frac{v}{w}) = (x, y)$$ **(Perspective projection)**

Photoreceptor density



- We measure x and y (image coordinates) in terms of number of pixels, so should be for depth (w) also.
- With same image size, depth ***value*** varies with photoreceptor density.

Photoreceptor density



- Hence, photoreceptor density should be combined with focal length.
- x unit of length $\equiv$ $x\varphi_x$ number of pixels
- y unit of length $\equiv$ $y\varphi_y$ number of pixels

Shift of origin

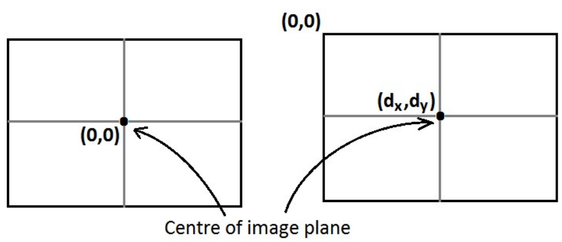


Image size in mm or inch: $(x, y) = \left(f\frac{u}{w}, f\frac{v}{w}\right)$

With shift of origin: $(x, y) = \left(f\frac{u}{w} + S_x, f\frac{v}{w} + S_y\right)$

where $d_x = \varphi_x S_x,\ d_y = \varphi_y S_y$

Photoreceptor density and shift

$$(u, v, w) \rightarrow (f\frac{u}{w}, f\frac{v}{w}) = (x, y)$$

turns to $(u, v, w) \rightarrow (f\frac{\varphi_x u}{w}, f\frac{\varphi_y v}{w}) = (x, y)$

Instead of origin (0,0) at the centre of image it usually at top-left corner, i.e., principal point is at (d_x, d_y). Thus

$$\mathrm{x} = f\frac{\varphi_x u}{w} + d_x$$
$$\mathrm{y} = f\frac{\varphi_y v}{w} + d_y$$

$(f, \varphi_x, \varphi_y, d_x, d_y)$ are ***intrinsic parameters*** of the camera.

Perspective Projection Matrix

$$(u, v, w) \to (f\frac{u}{w}, f\frac{v}{w})$$

Is this a linear transformation?

- no—division by w is non-linear

Slide by Steve Seitz

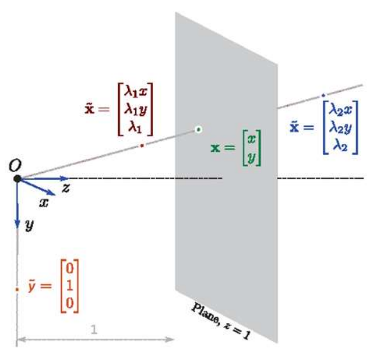
Homogeneous coordinates

Cartesian coordinate Homogeneous coordinate

$$(x, y) \to (\kappa x, \kappa y, \kappa)$$
$$(x/w, y/w) \leftarrow (x, y, w)$$
$$(x, y, z) \to (\kappa x, \kappa y, \kappa z, \kappa)$$
$$(x/w, y/w, z/w) \leftarrow (x, y, z, w)$$

Special case: $(x, y, z) \to (x, y, z, 1)$

Geometrical interpretation of homogeneous coordinate system



Perspective Projection Matrix

- $(u, v, w) \to (f\frac{u}{w}, f\frac{v}{w})$ is a non-linear transformation.
- To convert it to a linear transformation we adopt homogeneous coordinate system.
- Perspective projection is a matrix multiplication using homogeneous coordinates:

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1/f & 0\end{bmatrix}\begin{bmatrix}u\\ v\\ w\\ 1\end{bmatrix} = \begin{bmatrix}u\\ v\\ w/f\end{bmatrix}$$

$\Rightarrow (f\frac{u}{w}, f\frac{v}{w})$ (Divide by 3rd coordinate)

Projection matrix for camera

Projection is a matrix multiplication using homogeneous coordinates:

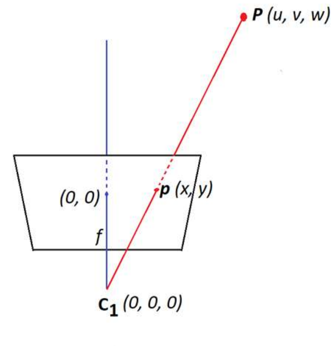
$$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1/f & 0\end{bmatrix}\begin{bmatrix}u\\ v\\ w\\ 1\end{bmatrix}=\begin{bmatrix}u\\ v\\ w/f\end{bmatrix} \Rightarrow (f\frac{u}{w}, f\frac{v}{w})$$

Considering photoreceptor density and shift:

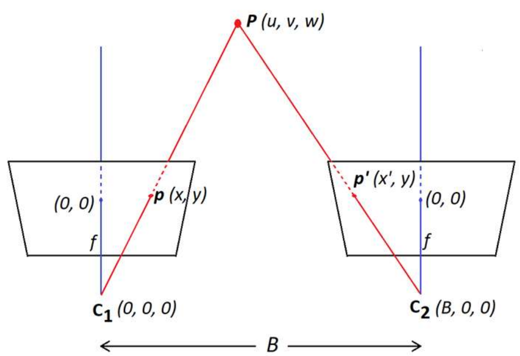
$$\begin{bmatrix}\varphi_x & 0 & d_x\\ 0 & \varphi_y & d_y\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1/f & 0\end{bmatrix}\begin{bmatrix}u\\ v\\ w\\ 1\end{bmatrix}=\begin{bmatrix}u\varphi_x + wd_x/f\\ v\varphi_y + wd_y/f\\ w/f\end{bmatrix}$$

$$\Rightarrow \left(f\frac{\varphi_x u}{w} + d_x, f\frac{\varphi_y v}{w} + d_y\right)$$

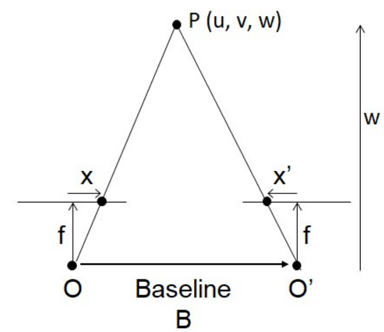
Depth from disparity



Depth from disparity



Depth from disparity



Depth from disparity

- Camera-2 is shifted from camera-1 only along horizontal direction and no rotation.
- Camera-1:
  - Optical centre O_1: $(0,0,0)$
  - World point P: (u, v, w)
  - Image coordinate: $x = f\frac{u}{w}$ $\Rightarrow$ $\frac{x}{f} = \frac{u}{w}$ (1)
- Camera-2:
  - Optical centre O_1: $(B, 0,0)$
  - World point P: $(u - B, v, w)$
  - Image coordinate: $x' = f\frac{u-B}{w}$ $\Rightarrow$ $\frac{x\prime}{f} = \frac{u-B}{w}$ (2)

Depth from disparity (contd.)

- Subtracting equation (2) from equation (1):

$$\frac{x - x'}{f} = \frac{B}{w}$$

- This implies

$$w = \frac{B \cdot f}{x - x'}$$

- Once we know w, u can be easily determined as

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

Similarly y.

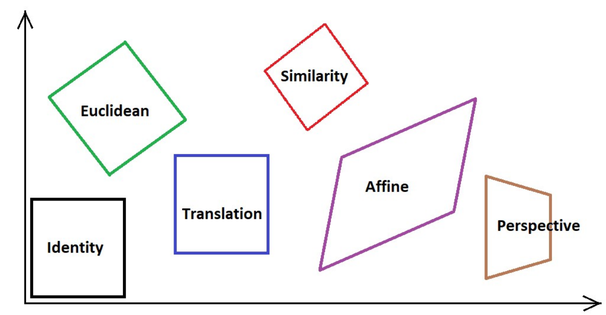
Position and orientation of camera

- Camera may be placed anywhere in a scene.
- Same object may appear differently while captured by same camera at different position and orientation.
- It would be easier to handle if the relation between their geometry is known.

Coordinate transformation

- World coordinate system
  - World (object) centric scene coordinates
- Camera coordinate system
  - Camera centric scene coordinates
- Transforming world (scene) coordinate to camera coordinate
  - Rotation, translation and (optional) scaling
  - Linear transformation is advantageous.

2D transforms



2D transforms

Rotation: $\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \equiv p' = \mathrm{R}\boldsymbol{p}$

(Coordinate axis is rotated anti-clockwise by an angle θ.)

Scaling: $\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} S_u & 0 \\ 0 & S_v \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \equiv p' = \mathrm{S}\boldsymbol{p}$

(Coordinate axis is squeezed / expanded $1/S_u$ and $1/S_v$.)

Translation: $\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \mathrm{t_u} \\ \mathrm{t_v} \end{bmatrix} \quad p' = \boldsymbol{p} + \mathrm{t}$

(Coordinate axis is translated by $-\mathrm{t_u}$ and $-\mathrm{t_v}$.)

3D transforms

Rotation: $R_u = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix}$

$$R_v = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \quad R_w = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = R_u R_v R_w \equiv p' = Rp, \text{ where } R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

3D transforms

Scaling: $\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} s_u & 0 & 0 \\ 0 & s_v & 0 \\ 0 & 0 & s_w \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \equiv p' = \mathrm{S}\boldsymbol{p}$

Translation: $\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} t_u \\ t_v \\ t_w \end{bmatrix} \equiv p' = \boldsymbol{p} + \boldsymbol{t}$

3D translation

Translation:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} u + t_u \\ v + t_v \\ w + t_w \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} t_u \\ t_v \\ t_w \end{bmatrix} \quad \equiv \quad \boldsymbol{p}' = \boldsymbol{p} + \boldsymbol{t}$$

In homogeneous coordinate system

$$\begin{bmatrix} u' \\ v' \\ w' \\ 1 \end{bmatrix} = \begin{bmatrix} u + t_u \\ v + t_v \\ w + t_w \\ 1 + 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_u \\ 0 & 1 & 0 & t_v \\ 0 & 0 & 1 & t_w \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} \quad \equiv \quad p_h' = \mathrm{T} p_h$$

Projection matrix for camera

- World to camera co-ordinate system using homogeneous coordinates:

$$\begin{bmatrix} u^c \\ v^c \\ w^c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_u \\ r_{32} & r_{22} & r_{23} & t_v \\ r_{31} & r_{32} & r_{33} & t_w \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$

$$\boldsymbol{p}^c = T\boldsymbol{p} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \boldsymbol{p}$$

- The elements of the matrix T, *i.e.*, $r_{i,j}$ and t_k are ***extrinsic parameters***.

Projection matrix for image formation

In practice: lots of coordinate transformations…

$$\begin{pmatrix} \text{2D} \\ \text{point} \\ \text{(3x1)} \end{pmatrix} = \begin{pmatrix} \text{Camera to} \\ \text{pixel coord.} \\ \text{trans. matrix} \\ \text{(3x3)} \end{pmatrix} \begin{pmatrix} \text{Perspective} \\ \text{projection matrix} \\ \text{(3x4)} \end{pmatrix} \begin{pmatrix} \text{World to} \\ \text{camera coord.} \\ \text{trans. matrix} \\ \text{(4x4)} \end{pmatrix} \begin{pmatrix} \text{3D} \\ \text{point} \\ \text{(4x1)} \end{pmatrix}$$

$$\begin{bmatrix} u^c\varphi_x + w^c d_x/f \\ v^c\varphi_y + w^c d_y/f \\ w^c/f \end{bmatrix} = \begin{bmatrix} \varphi_x & 0 & d_x \\ 0 & \varphi_y & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_u \\ r_{32} & r_{22} & r_{23} & t_v \\ r_{31} & r_{32} & r_{33} & t_w \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u^c\varphi_x + w^c d_x/f \\ v^c\varphi_y + w^c d_y/f \\ w^c/f \end{bmatrix} = \begin{bmatrix} \varphi_x & 0 & d_x/f & 0 \\ 0 & \varphi_y & d_y/f & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_u \\ r_{32} & r_{22} & r_{23} & t_v \\ r_{31} & r_{32} & r_{33} & t_w \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$

Thank you!

Any question?