Camera Model + Geometric Transformation

Jenn-Jier James Lien (連 震 杰)
Professor

Computer Science and Information Engineering
National Cheng Kung University

(O) (06) 2757575 ext. 62540
jjlien@csie.ncku.edu.tw
http://robotics.csie.ncku.edu.tw

Major Issues

1. Camera Model:

1) Intrinsic + Extrinsic Parameters
2) Homogenous Matrix

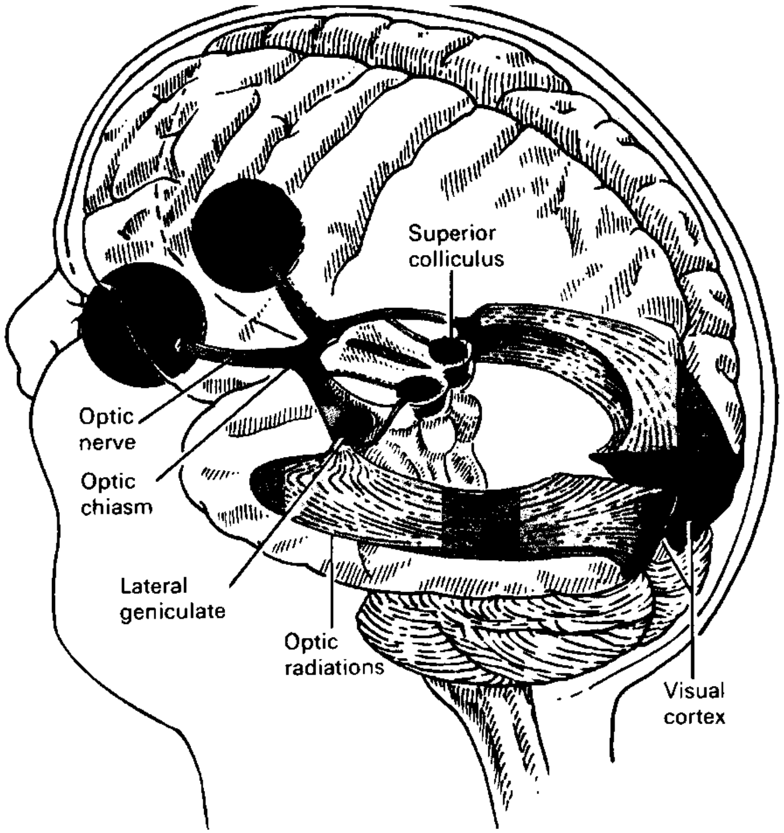
2. Geometric Transformation:

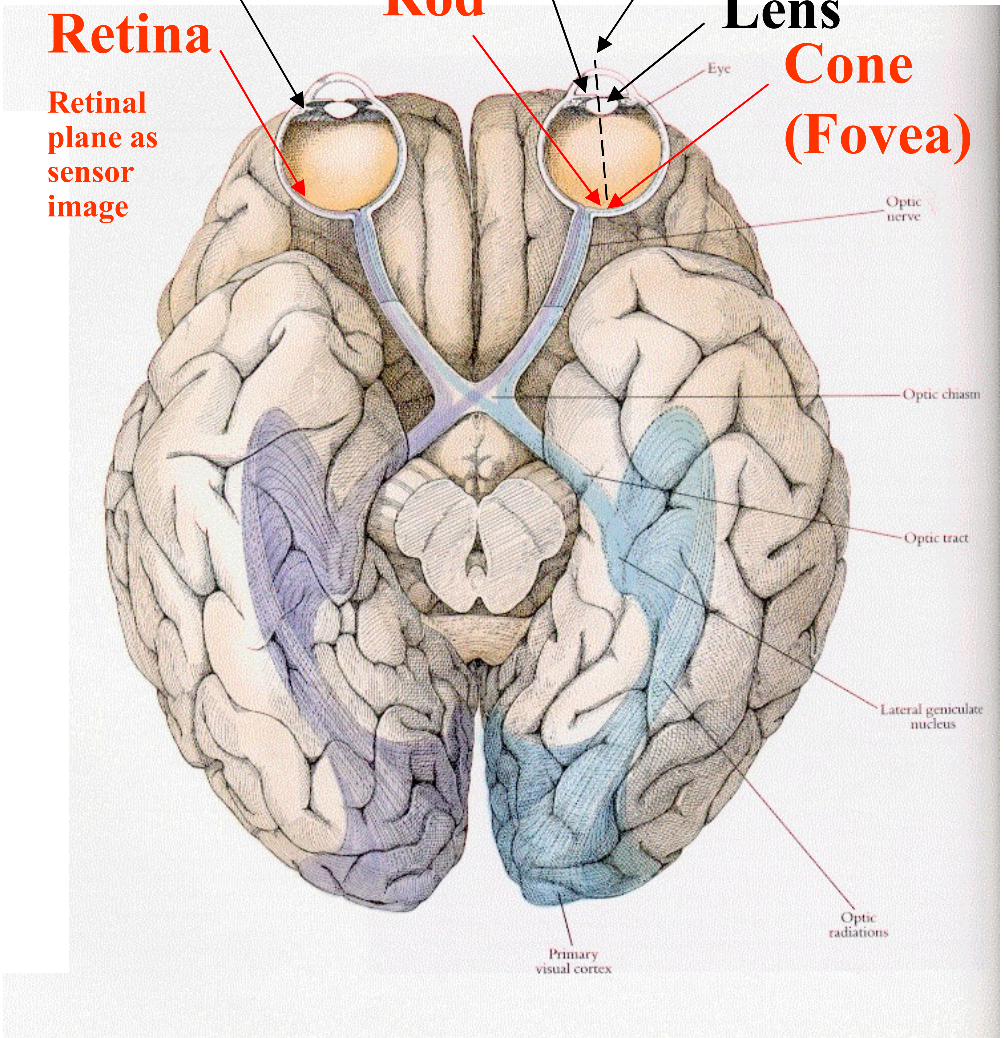
1) Affine Transformation
2) Perspective Transformation

Human Eyes

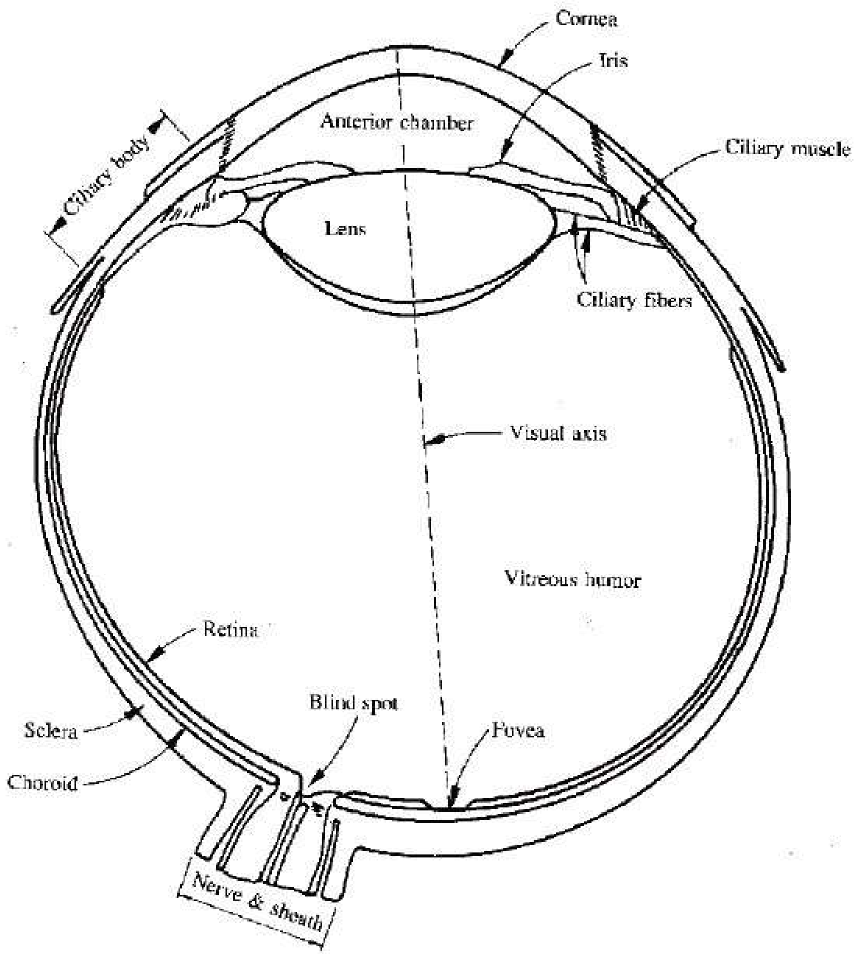
- **Components of the human eyes**
  - Pupil 瞳孔
  - Lens
  - Ciliary muscle
  - Retina
  - Fovea
  - Blind spot

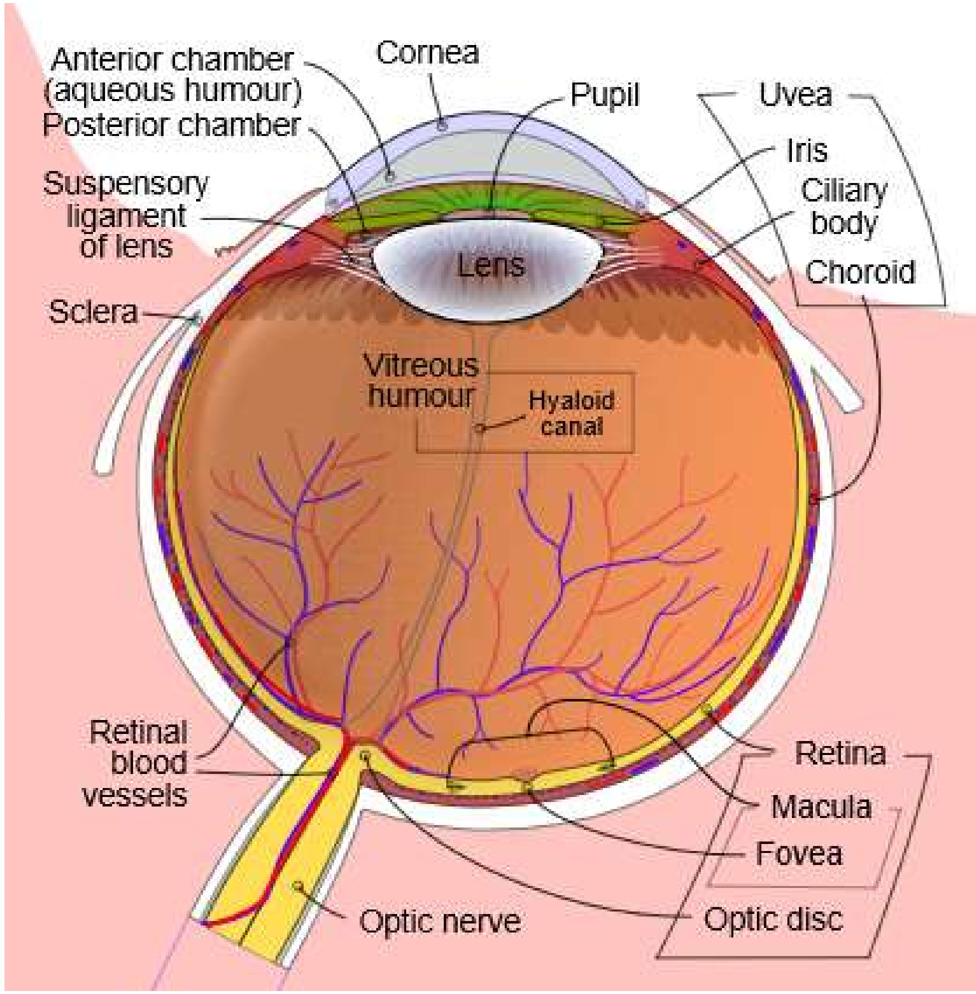
Visual Cortex





Cross Section of Human Eye

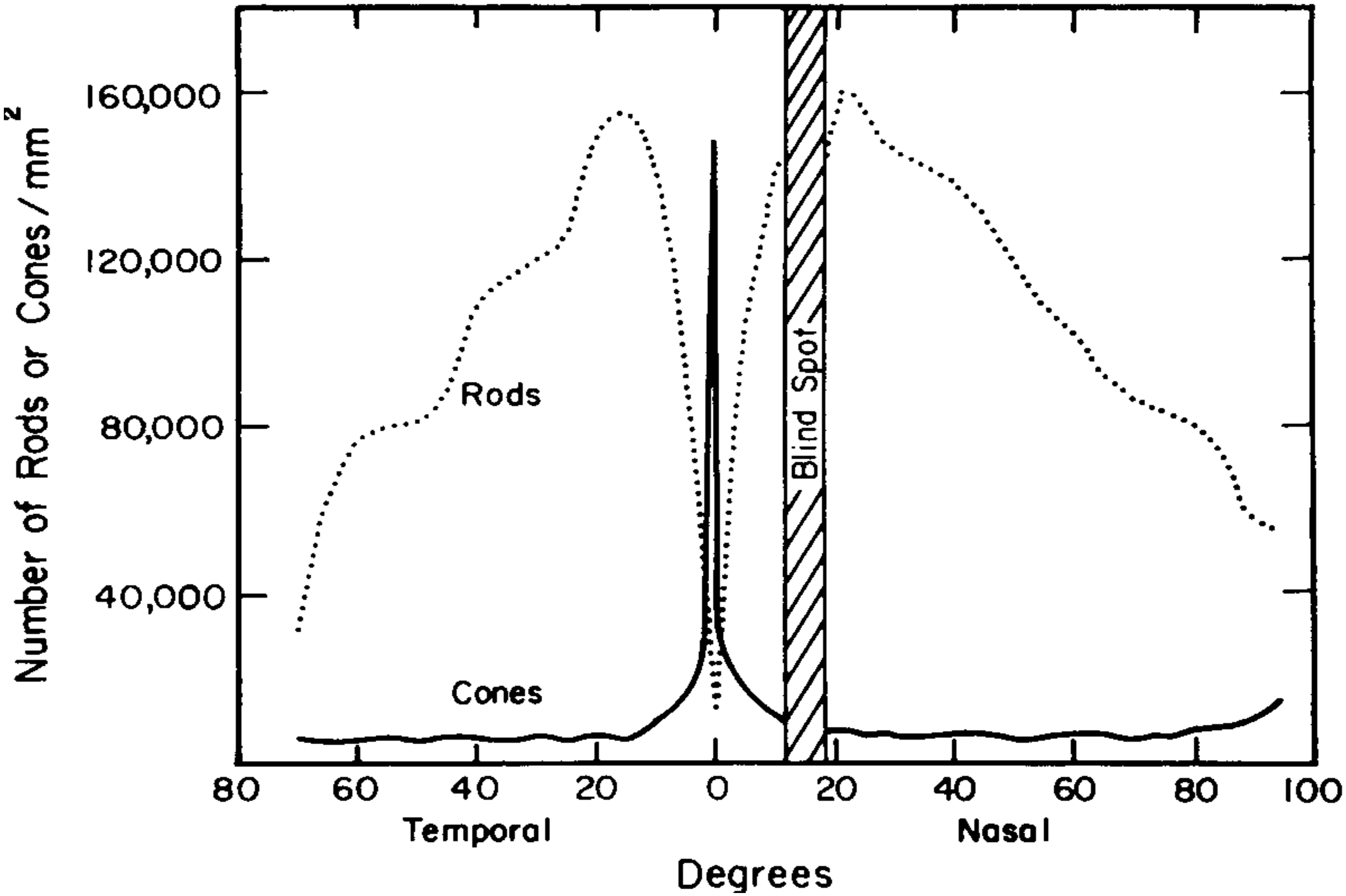




The Retina

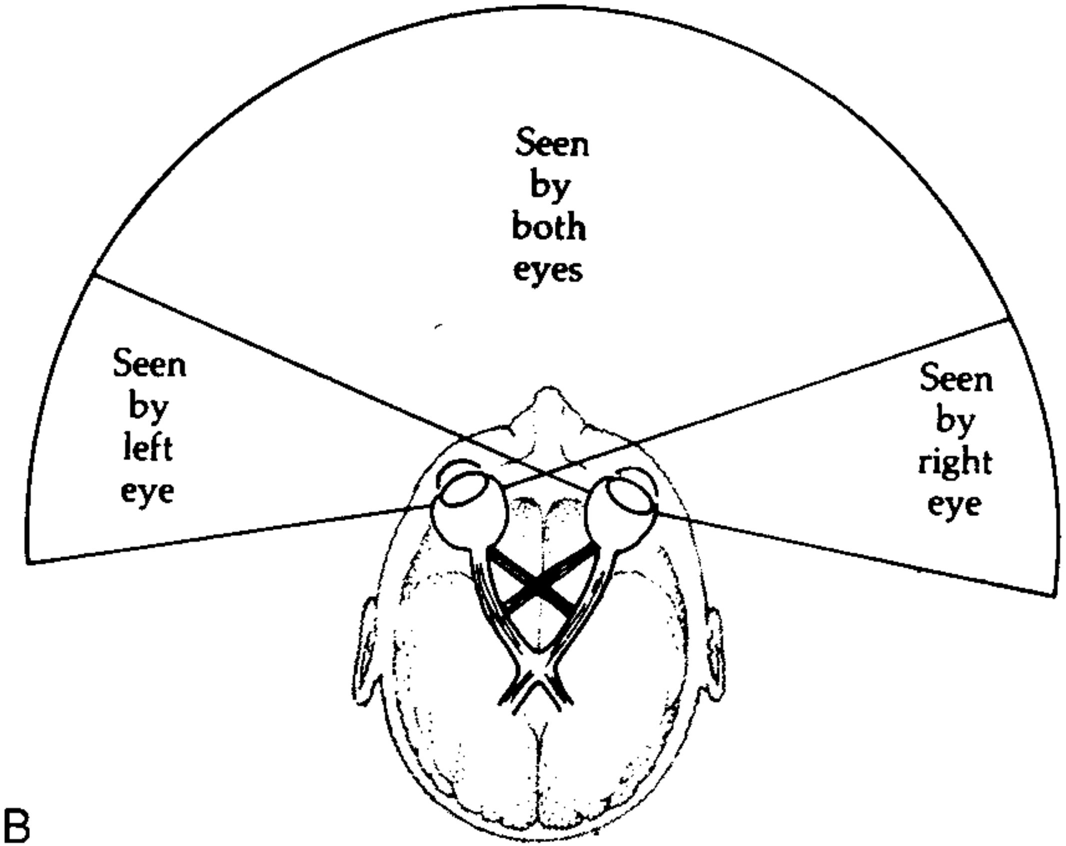
- **Retina: There are two types of photosensitive cells in the retina, rods and cones**
  - **Cones: come in three flavors which exhibit different sensitivities to different wavelengths of light, red green and blue.**
  - **Rods: are not sensitive to variations in wavelength but they are more sensitive than cones and can pick up much dimmer light**
- **Fovea: is populated entirely by cones.**

Receptor Density Vs. Eccentricity



Human Visual Field

Monocular Visual Field: 160 deg (w) X 175 deg (h)
Binocular Visual Field: 200 deg (w) X 175 ? deg (h)



Camera: A Brief History

- **From the latin *Camera Obscura* - Dark Chamber**
- **The cameras of the 11th century were literally dark rooms with a small opening on one side**

The Invention of the Lens

- **The first major improvement occurred in the 16th century when Giovanni Battista della Porta added a lens to the design**
- **The lens improved the light gathering power of the device so brighter images were possible**
- **The problem with being ahead of your time...**
  - Unfortunately Signor della Porta's contribution was not universally appreciated at first. He was called before the papal court on charges of sorcery

Forming Images

- **The first cameras were used by painters who would literally sit in the box and trace the outline of the image on their canvases**
- **Film**
  - The first film was developed by Niepce (1822) who took the worlds first photograph of a farmyard in central France – it required an exposure time of eight hours.
  - He joined forces with Louis Jacques Mande Daguerre who improved the process. His exposures only required half an hour and were quite sharp.

The Worlds' First Known Photo?

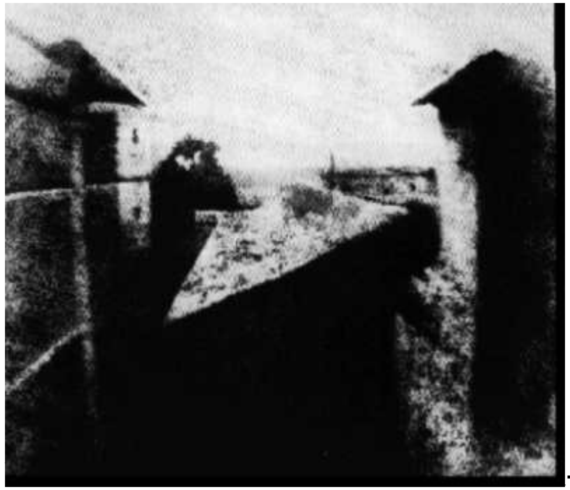


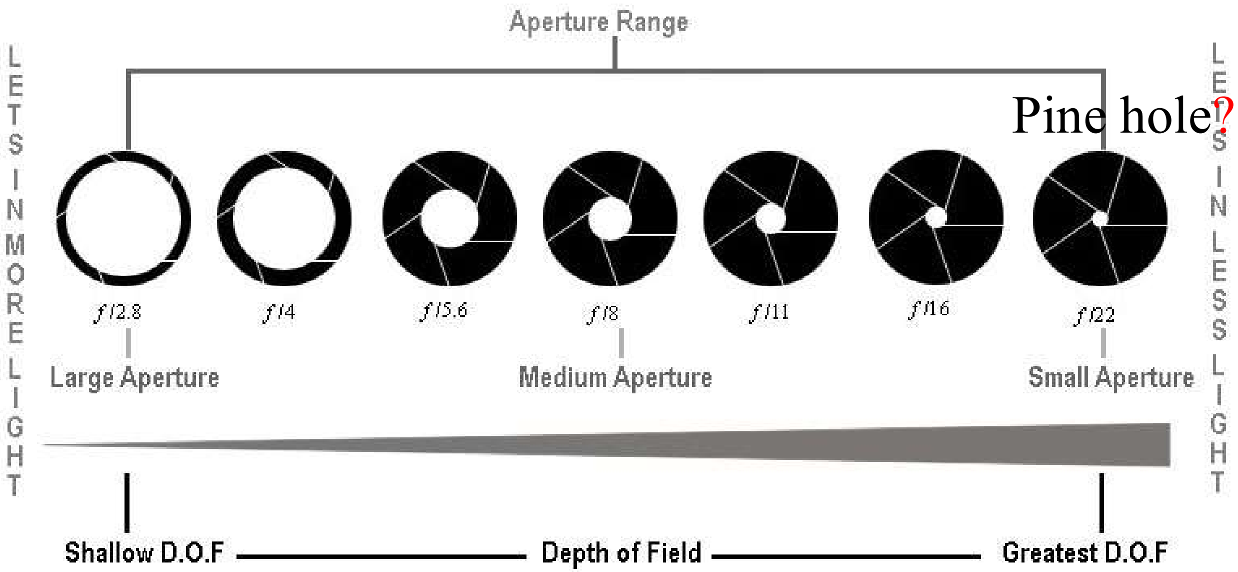
Image Formation Models

- **Optics**
  - Reflection, lens, aperture…
- **Sensor**
  - Pixel aspect ratio, color technology, dynamic range…
- **Projections**
  - Perspective projections
  - Parallel projections (e.g., orthogonal projection)

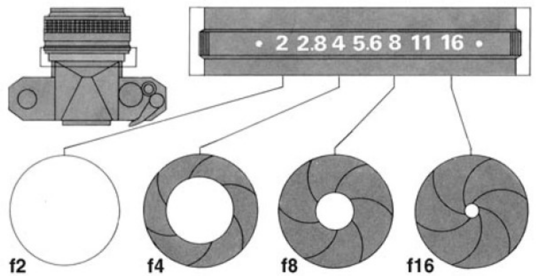
Optics

The Anatomy of a Modern Camera ??

- **Lens**
  - Focusing Control Vs. Focal Length *f*
  - Shutter Speed 快門速度
  - Aperture/Iris 光圈大小
- **Film**



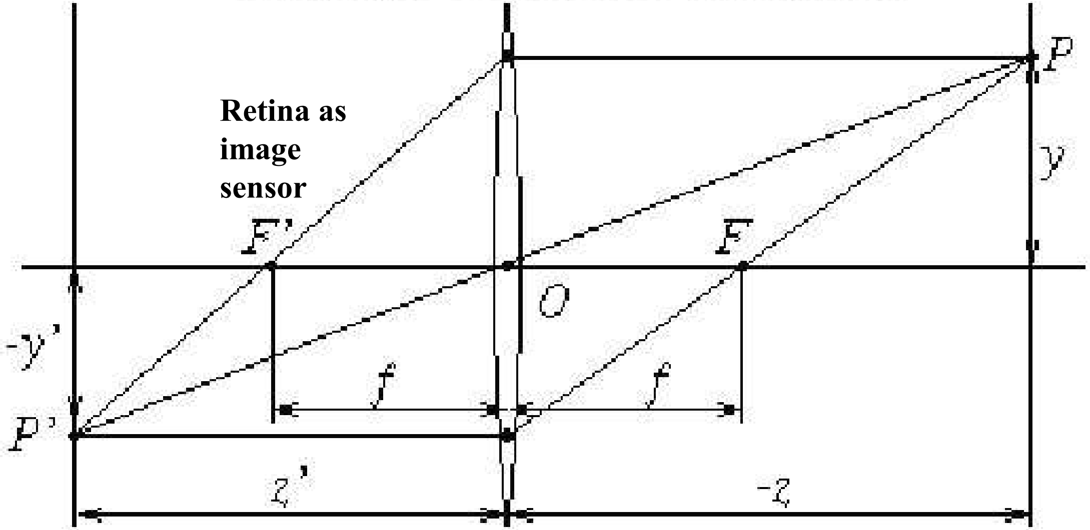
Pine hole??



Issues with Real Lenses

- **Depth of Field (DOF) Vs. Field of View (FOV)**
  - **(or Depth of resolution Vs. image resolution)**
- **Focus**
- **Lens aberrations**
  - Chromatic aberration
  - Spherical aberration

Ideal Thin Lenses



All rays emanating from ***P*** converge to a single point ***P'***

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

Points at infinity are focused on plane $z' = f$

Ideal because: infinite aperture

infinite field of view

infinitely small distance between surfaces

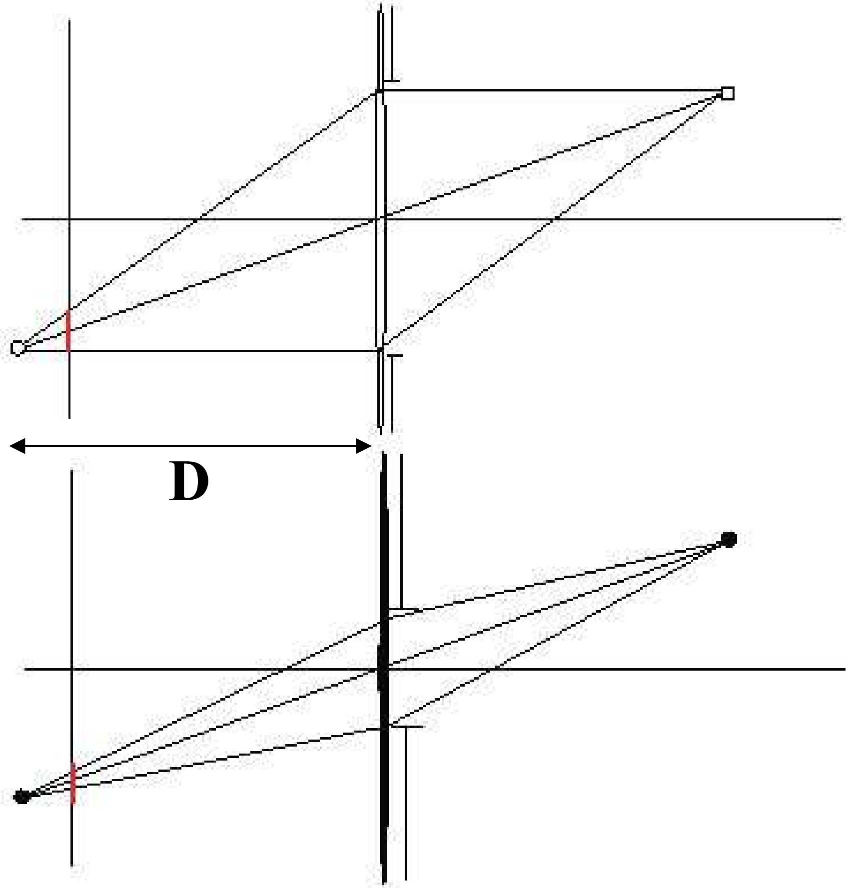
- **Lenses are used to increase the light gathering power of the instrument**

Finite Aperture

Ideal case: Only the points on one plane are in perfect focus

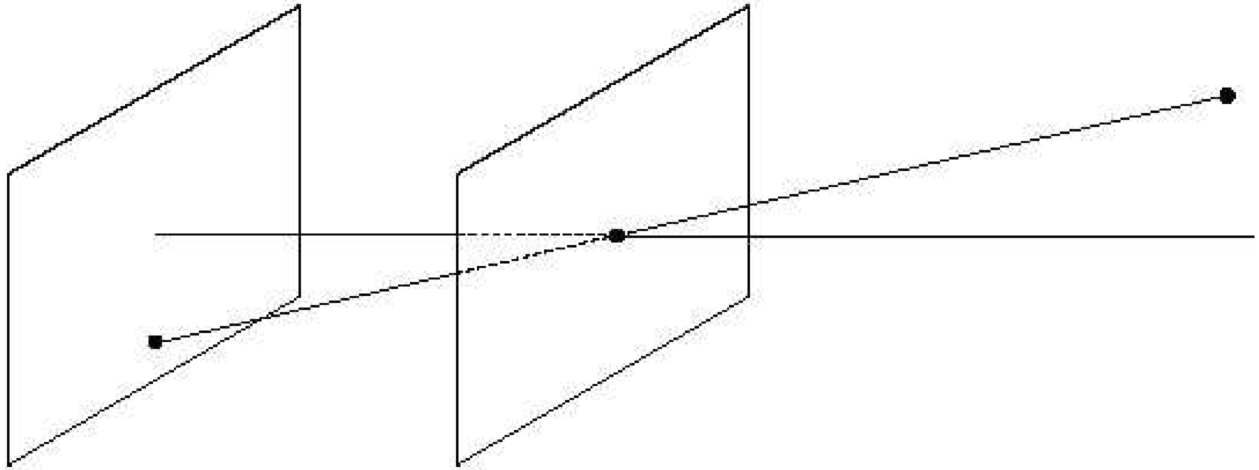
Finite aperture: points within a region of depth D (depth of field) are in focus.

For a given f, the larger the aperture, the smaller D



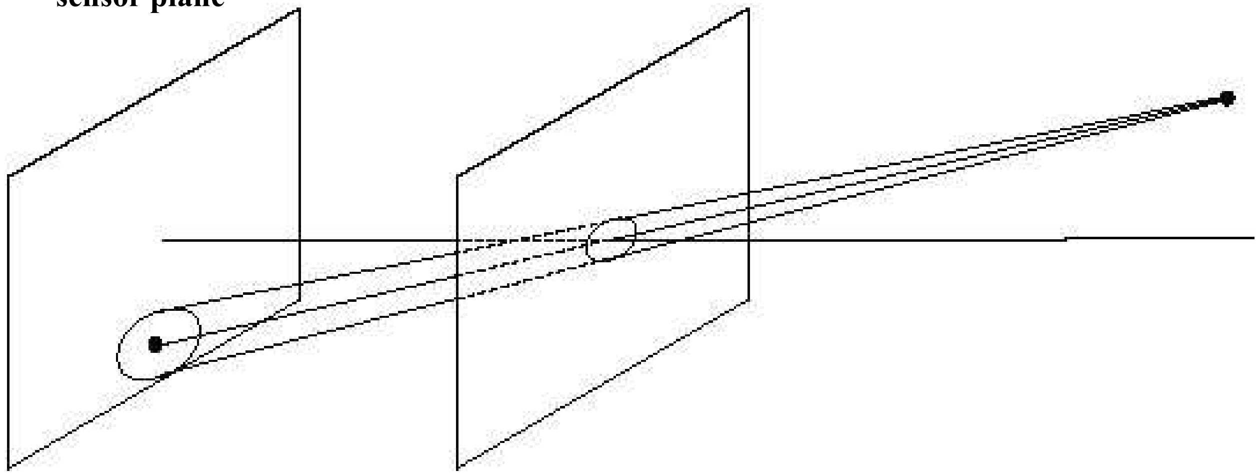
Limitations of the Pinhole Model

Ideal pinhole:
Single scene point generates single image
but:
Diffraction
Low light level

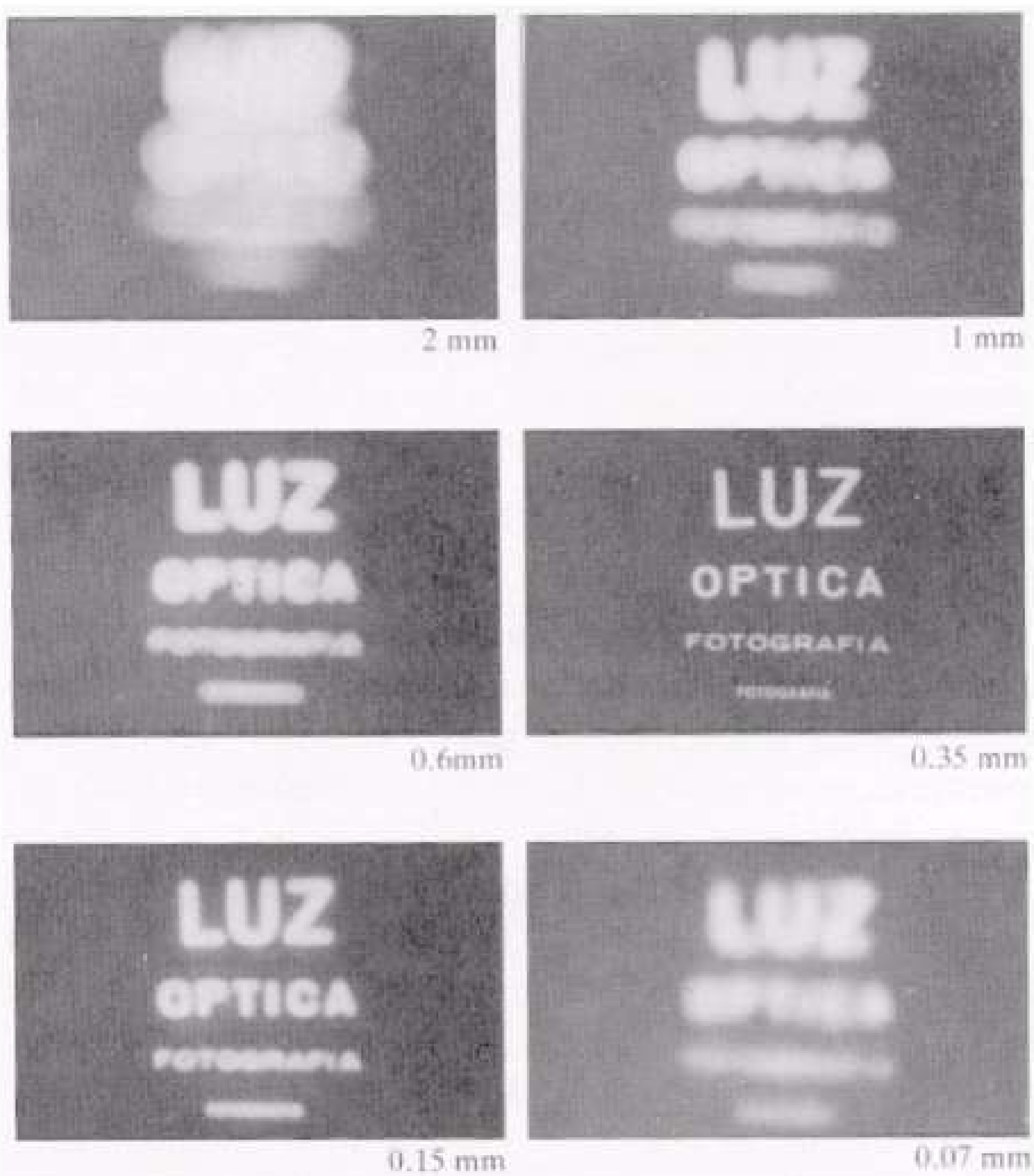


Retinal plane as image sensor plane

Finite-size pinhole:
Single scene point generates extended image.
Resulting image is blurry



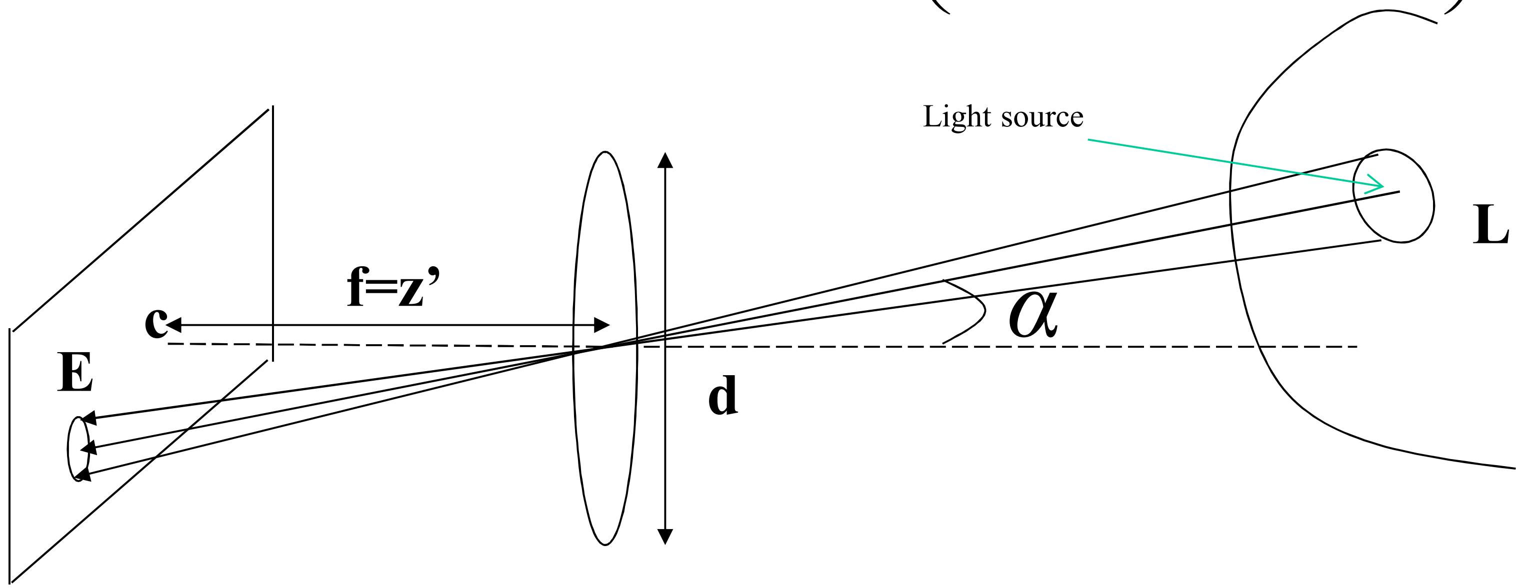
Focus and Focal Length



A Physical Model: Fundamental Equation

- The fundamental equation relates **scene radiance L** to **image irradiance E**
  - *E* linearly related to *L*

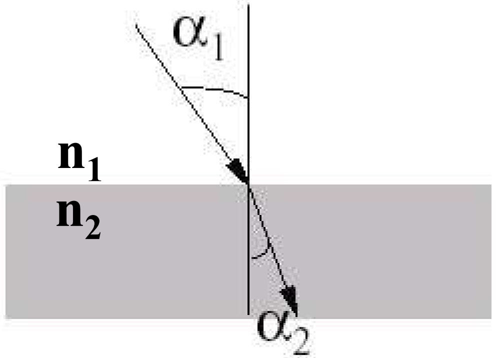
$$E = \left(\left(\frac{\pi}{4}\right)\left(\frac{d}{z'}\right)^2 \cos^4 \alpha \right) L$$



Aberrations

Snell's law of refraction:

$$n_1 \sin(\alpha_1) = n_2 \sin(\alpha_2)$$



First-order optics: appropriate for ideal model of thin lens

$$n_1\alpha_1 \approx n_2\alpha_2$$

Higher order optics: necessary for real lenses

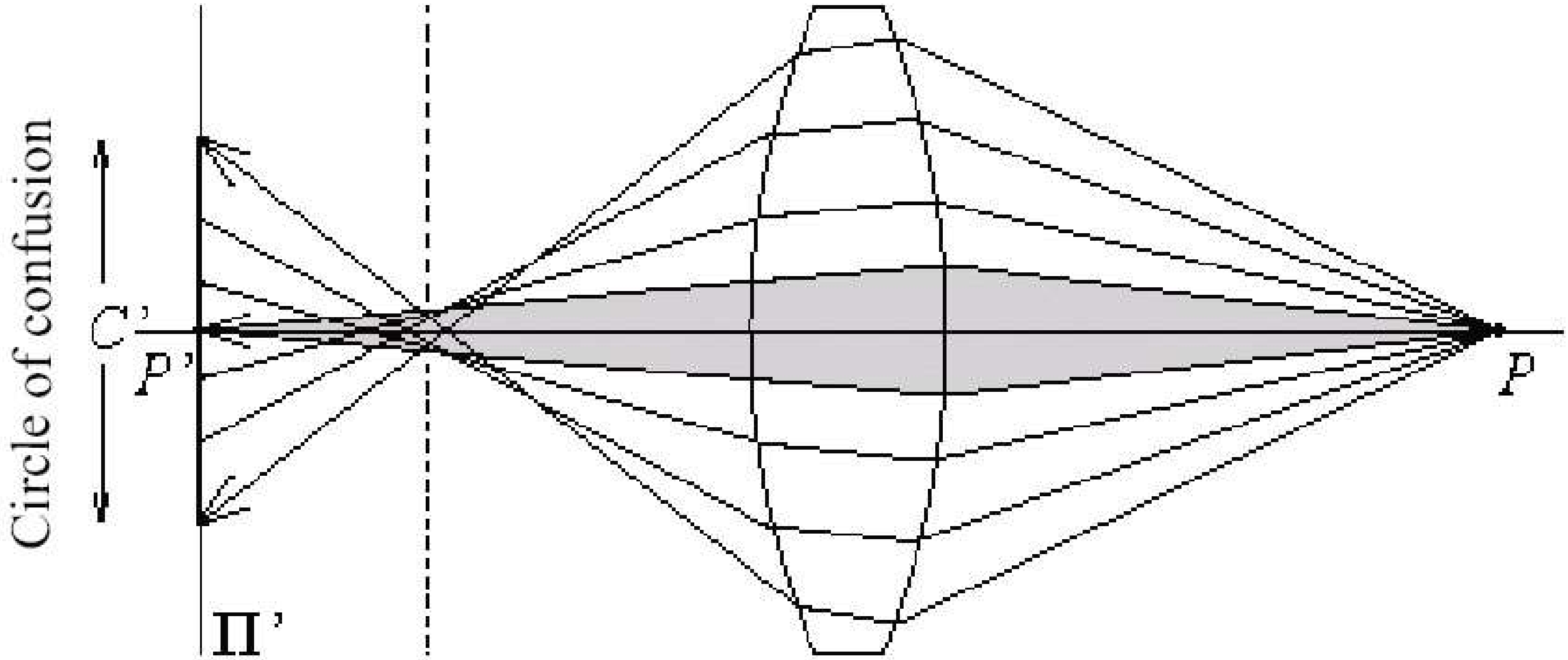
$$\sin\alpha \approx \alpha - \frac{\alpha^3}{6} + \cdots$$

Aberrations:

Blurring: e.g., spherical aberrations

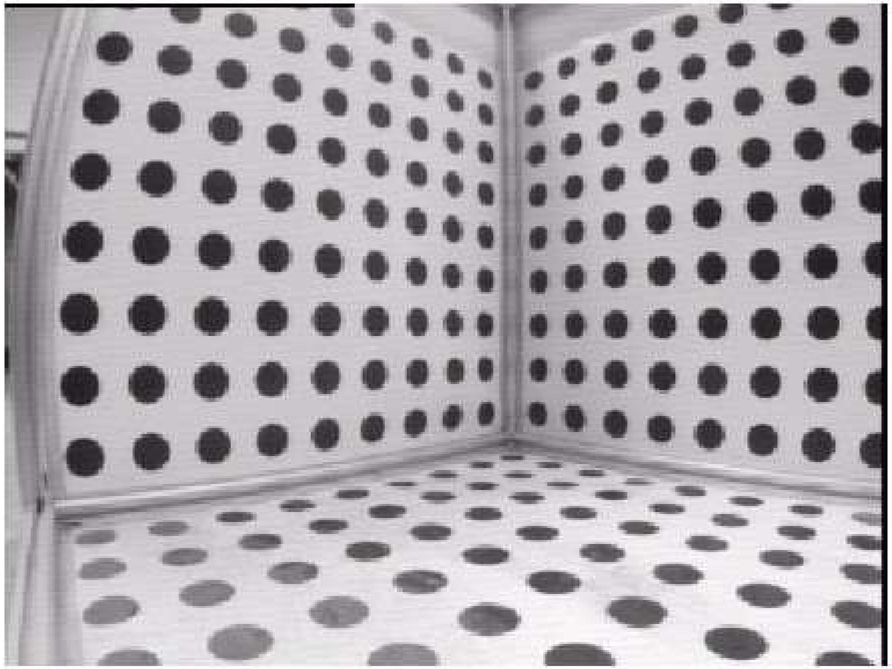
Geometric distortion

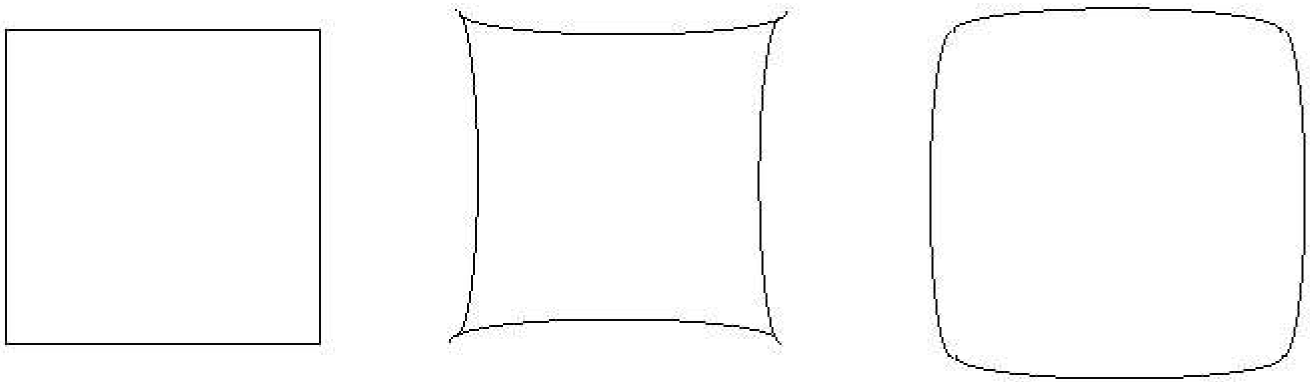
Spherical Aberration



Rays further from the optical axis are focused closer to the lens

Geometric Distortion

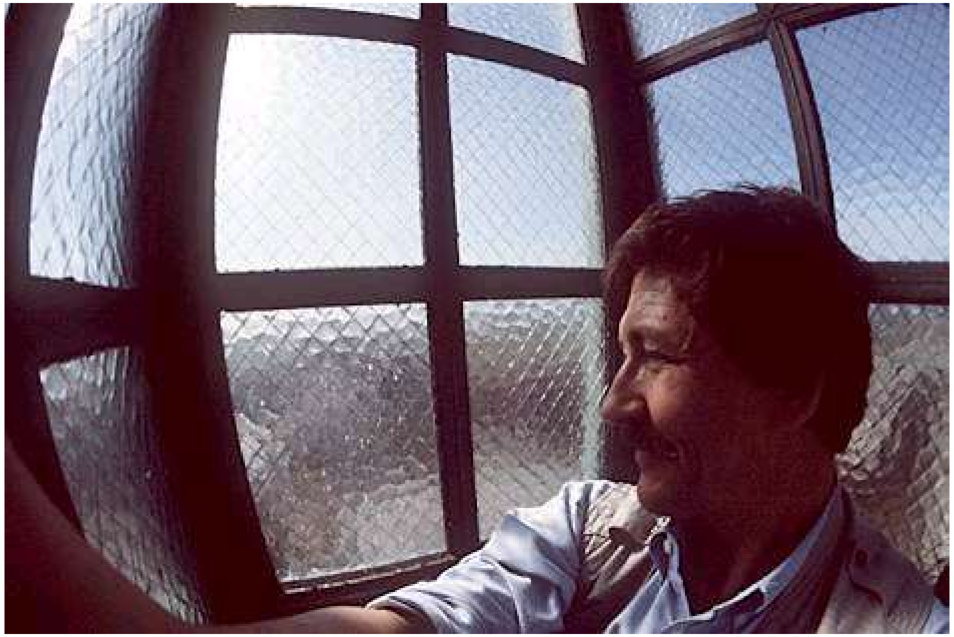




pincushion barrel

Radial Distortion

- Note how straight lines are distorted into curves by radial distortion in this image



- Corresponds to a dilation of the image
- It is most pronounced in optical systems with a wide field of view

$$x_u = x_d(1 + k_1 r^2 + k_2 r^4)$$

$$y_u = y_d(1 + k_1 r^2 + k_2 r^4)$$

$$where$$

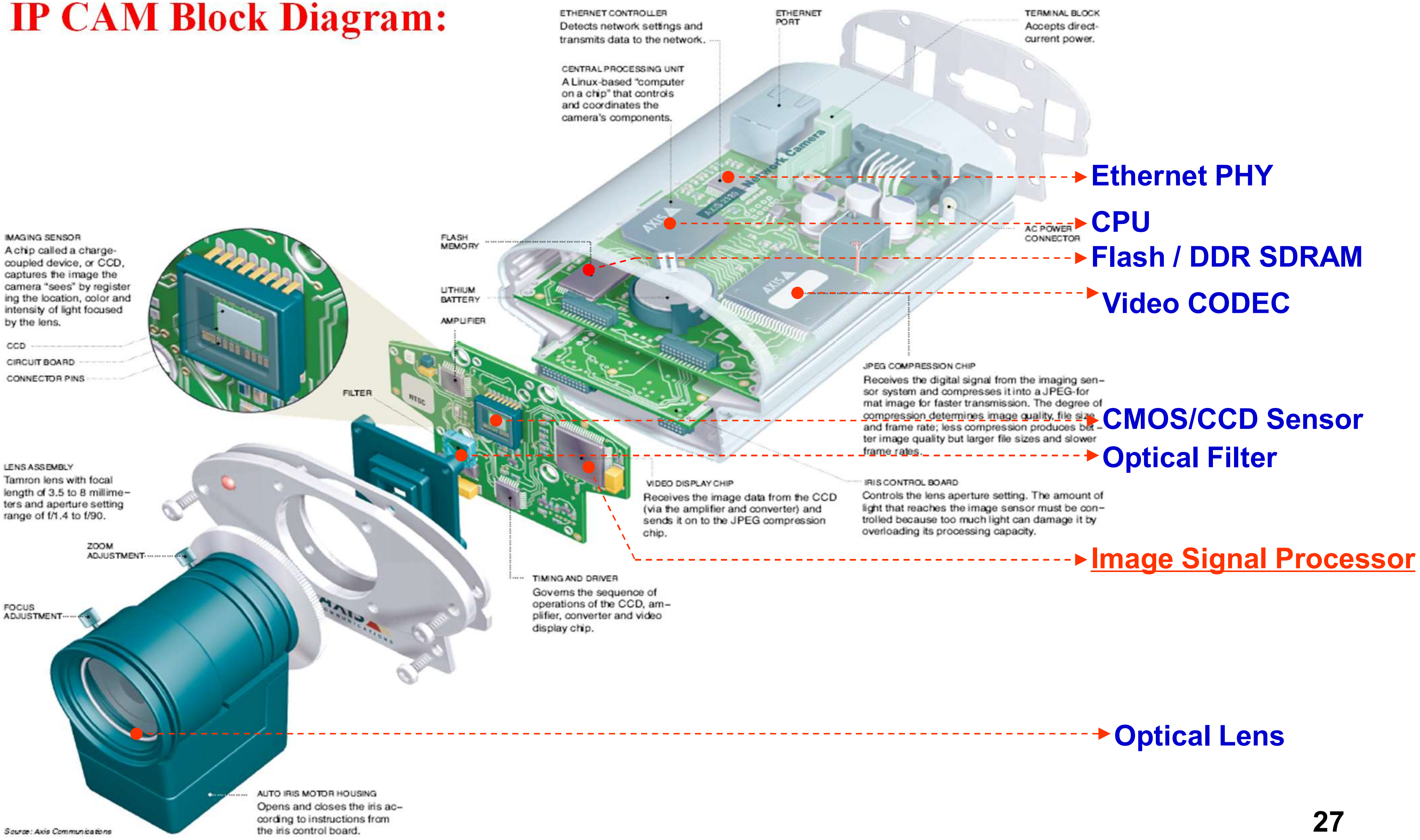
$$r^2 = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$$

Image Sensors

- **Images are formed by the interaction of the incident image irradiance with light sensitive elements on the image plane**
- **Light sensitive elements**
  - Analog: Charge Coupled Device (CCD)
    - Film
    - Good for capturing high speed moving targets
  - Digital: CMOS Imaging Element
    - IP Camera +
    - ISP (Image Signal Processing) +
    - IVA (Intelligent Video Analytics)

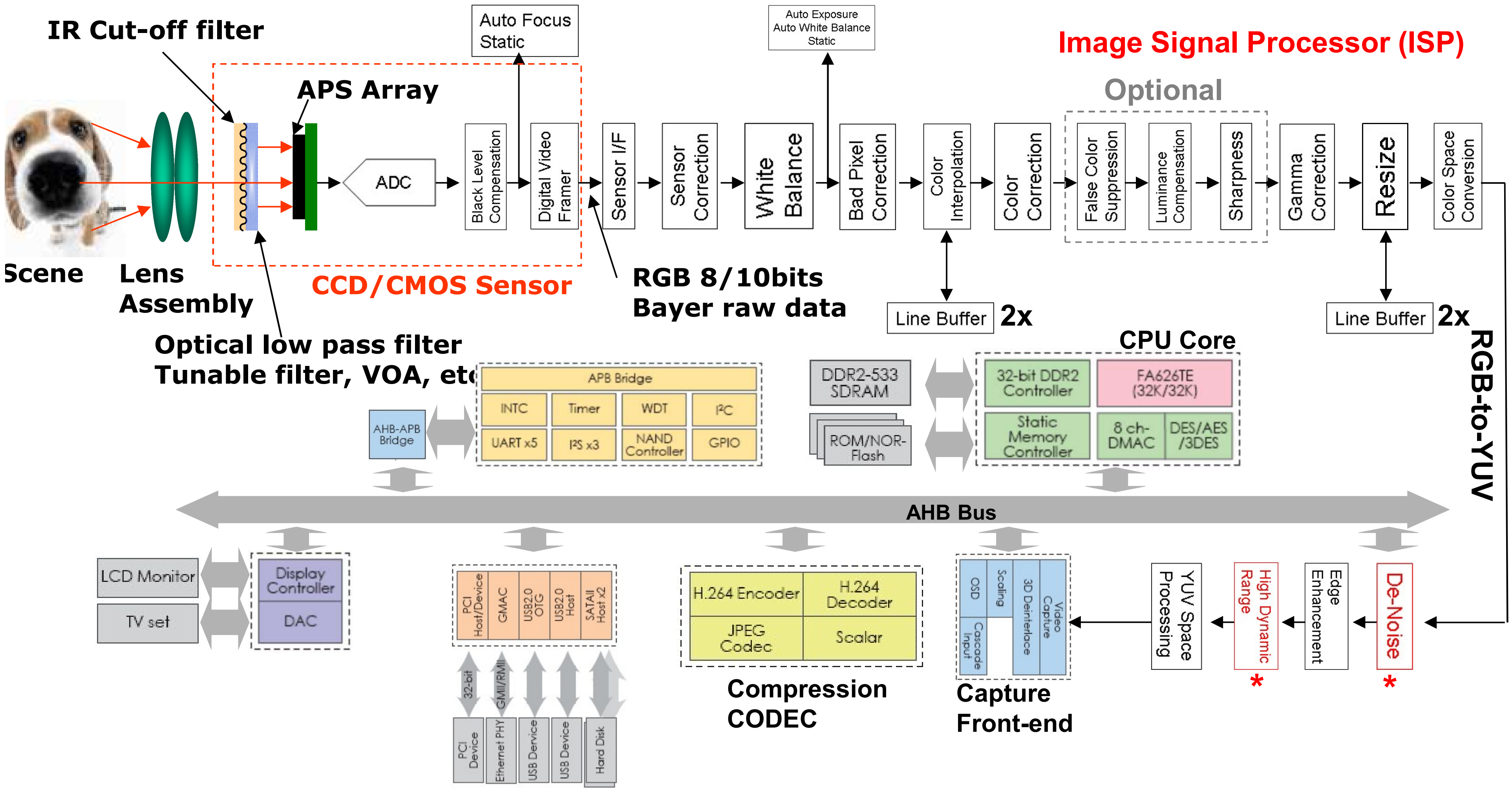
1.0 Image and Signal Processing (ISP) (1/2)

■ IP CAM Block Diagram:



1.0 Image and Signal Processing (ISP) (2/2)

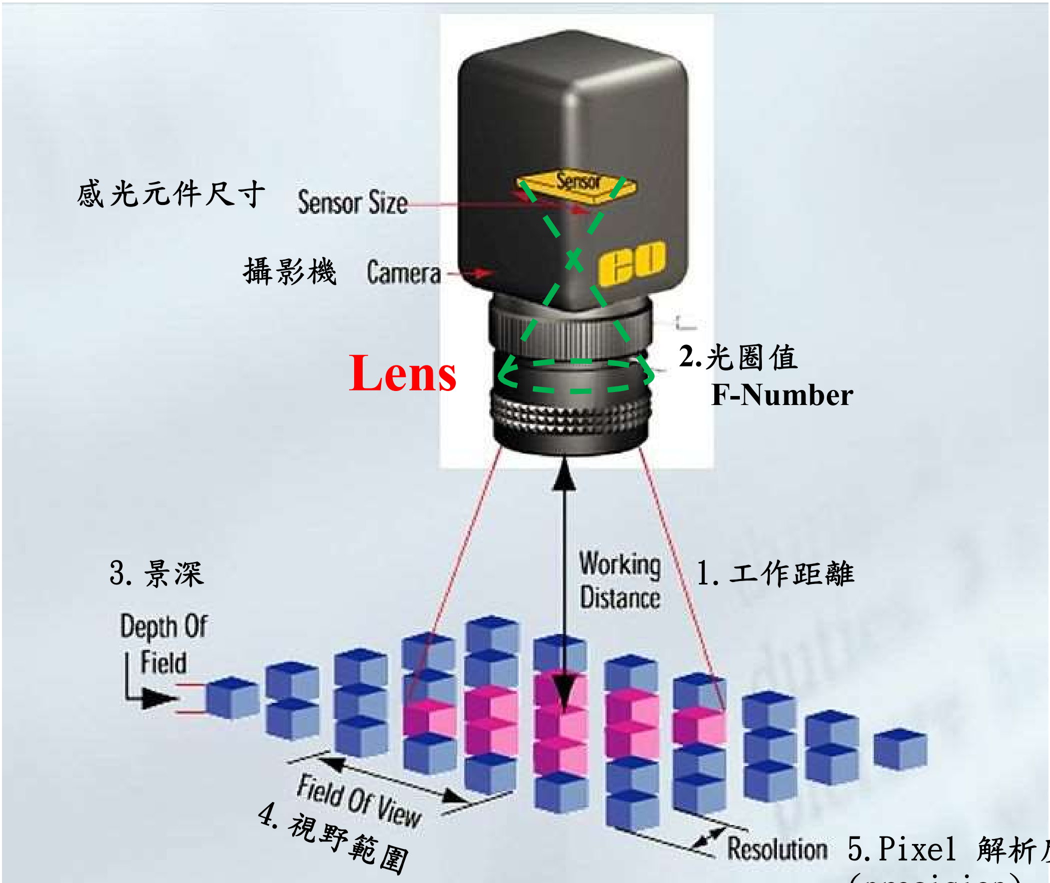
- **IP CAM Function Block Diagram:**



鏡頭參數

- 攝影機與鏡頭組合的基本名詞解釋：

  鏡頭參數有5個：**1.**工作距離 **2.**光圈值 **3.**景深 **4.**視野範圍 **5.**圖形分辨率

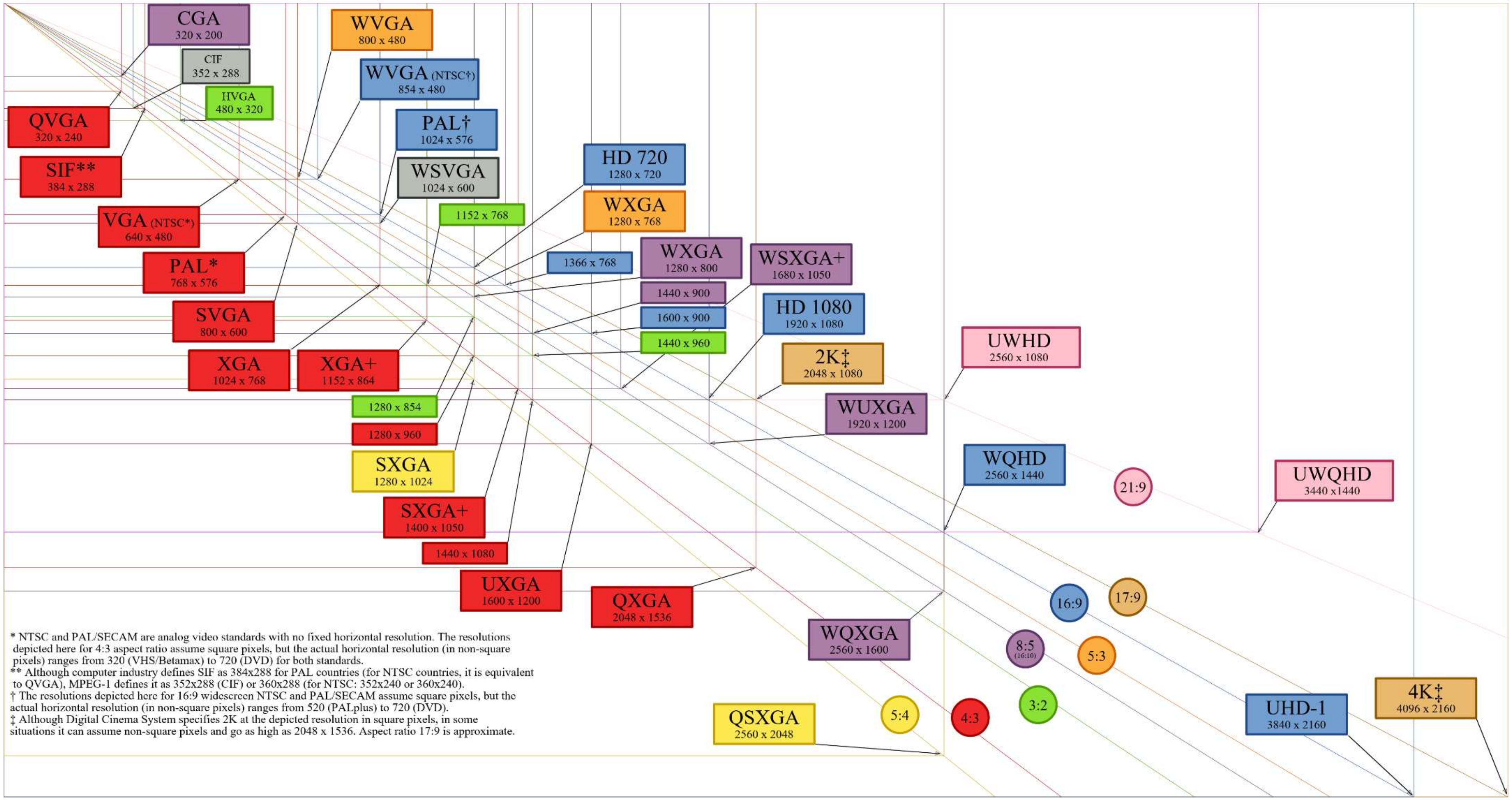


5. Pixel 解析度 (precision)

PPI: Pixels per inch

Image Resolution: 640 x 480 pixels

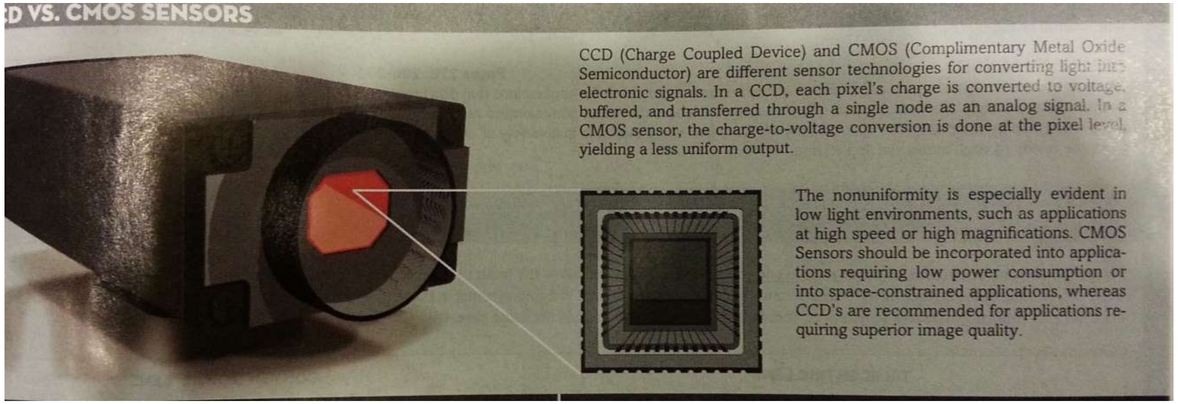
Display Resolution

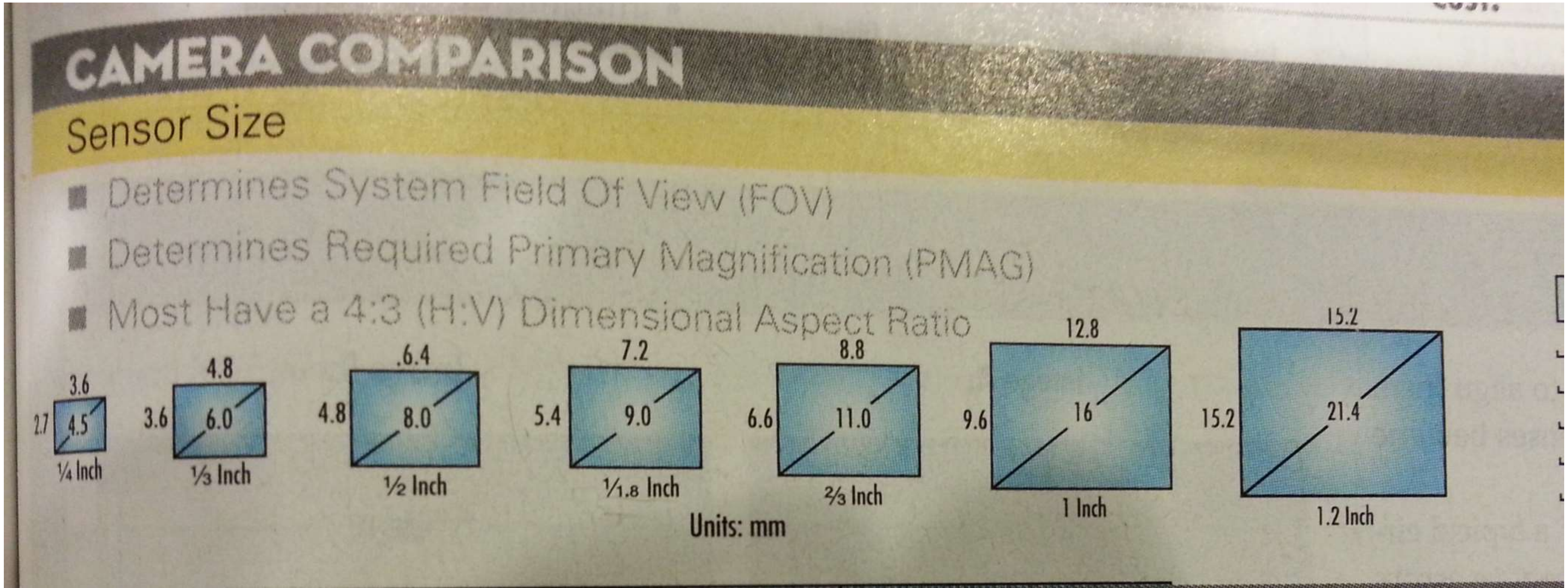


Televisions are of the following resolutions:

- **Standard-definition television (SDTV):**
  - 480i (NTSC-compatible digital standard employing two interlaced fields of 243 lines each)
  - 576i (PAL-compatible digital standard employing two interlaced fields of 288 lines each)
- **Enhanced-definition television (EDTV):**
  - 480p (720 × 480 progressive scan)
  - 576p (720 × 576 progressive scan)
- **High-definition television (HDTV):**
  - HD (1280 × 720 progressive scan)
  - Full HDi (1920 × 1080 split into two interlaced fields of 540 lines)
  - Full HD (1920 × 1080 progressive scan)
- **Ultra-high-definition television (UHDTV):**
  - 4K UHD (3840 × 2160 progressive scan)
  - True 4K (4096 × 2160)
  - 8K UHD (7680 × 4320 progressive scan)
  - True 8K (8192 × 4320)

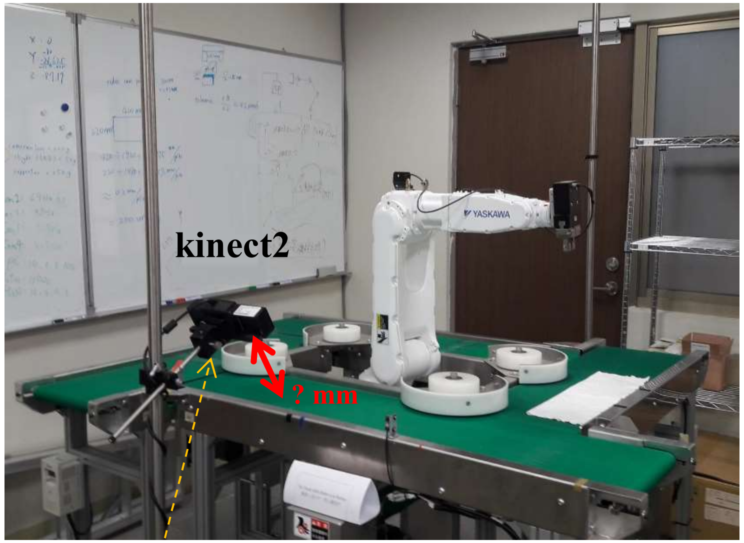
Sensor Size


D VS. CMOS SENSORS
CCD (Charge Coupled Device) and CMOS (Complimentary Metal Oxide Semiconductor) are different sensor technologies for converting light into electronic signals. In a CCD, each pixel's charge is converted to voltage, buffered, and transferred through a single node as an analog signal. In a CMOS sensor, the charge-to-voltage conversion is done at the pixel level, yielding a less uniform output.
The nonuniformity is especially evident in low light environments, such as applications at high speed or high magnifications. CMOS Sensors should be incorporated into applications requiring low power consumption or into space-constrained applications, whereas CCD's are recommended for applications requiring superior image quality.


CAMERA COMPARISON
Sensor Size
Determines System Field Of View (FOV)
Determines Required Primary Magnification (PMAG)
Most Have a 4:3 (H:V) Dimensional Aspect Ratio
3.6 2.7 4.5 ¼ Inch
4.8 3.6 6.0 ⅓ Inch
6.4 4.8 8.0 ½ Inch
7.2 5.4 9.0 1/1.8 Inch
8.8 6.6 11.0 ⅔ Inch
12.8 9.6 16 1 Inch
15.2 15.2 21.4 1.2 Inch
Units: mm

System Setup: Specification

- **Decide precision and camera position:**



Move kinect2 outside of the conveyor to avoid occlusion by arm.

Opened Gripper (Max gripper width)	18.2mm
Target (nail width)	16.5mm

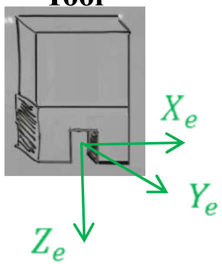
Kinect2 Camera Spec.	w	H
Image Resolution	1920 pixels	1080 pixels
Focal Length[1]	3.29mm	
Sensor Size[1]	6.00 mm	3.38 mm
Pixel Size Precision[1]	3.13 µm/pixel	3.13 µm/pixel
Field of View (FOV)	400 mm	225 mm
Pixel size (FOV) ?	$208.3\mu m/pixel$	$208.3(\mu m/pixel)$

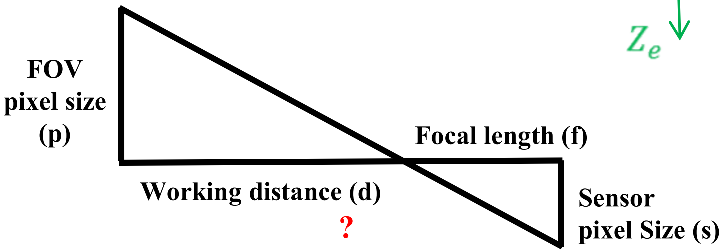
Image I (1920*1080 pixels) :

h:225mm

w: 400mm

Tool





$$\frac{p\ (\mu\mathbf{m}/pixel)}{?\ d(\mathrm{m}\mathbf{m})} = \frac{s\ (\mu\mathbf{m}/pixel)}{f(\mathrm{m}\mathbf{m})}$$

$$width{:}\,\mathrm{P} = \frac{\textbf{FOV}}{\textbf{Resolution}} = \frac{400{,}000}{1920} = 208.3(\mu\mathbf{m}/pixel)$$

$$\frac{225{,}000}{1080} = 208.3(\mu\mathbf{m}/pixel)$$

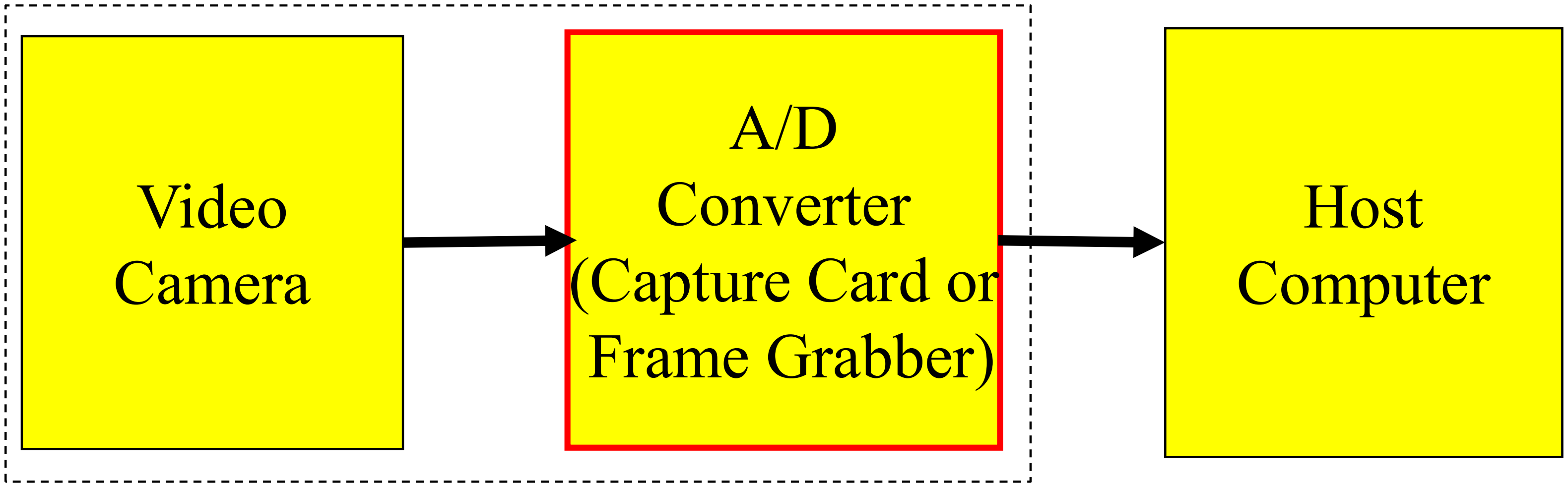
$$\frac{208.3}{d} = \frac{3.13}{3.291} \Rightarrow d = 219(mm) = 21.9(cm)$$

View-based gripper one side tolerance:

$$\frac{\textbf{(18.2mm–16.5mm)/2}}{0.2083\ mm} = \frac{0.85\mathrm{mm}}{0.2083\ mm} = 4.1\ pixels$$

Digital Imaging Systems

- **CCD or CMOS imaging array**
  - When light falls on the cells in these arrays, a charge accumulates which is proportional to the incident light energy
- **Analog to digital conversion unit**
- **Host Computer**



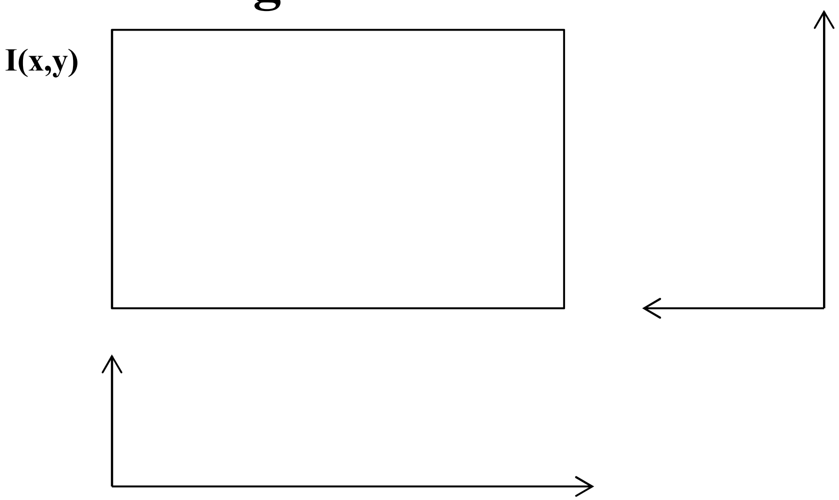
Digital Snapshots

Pixel ? Resolution ?

- **A digital image is an array of numbers indicating the image irradiance at various points on the image plane**
- **Image intensities are spatially sampled**
  - The image irradiance function across the retinal plane is sampled to obtain the digital image
  - The spacing of the image elements limits the resolution of the image
  - The frequency content of the irradiance function is limited by the effective aperture of the camera
- **Intensity values (grayvalue) are quantized (8-bits, 10-bits, 12-bits)**
  - 2D image plane to 1D+1D histogram distribution => Probability
- **Video Imagery**
  - For a video camera, images are taken sequentially by opening and closing the shutter 30 times per second

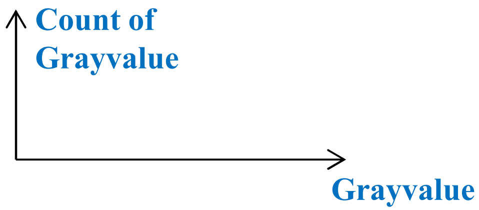
2D Image Plane to 1D+1D Histogram Distribution => Probability

1.1 Projection Histogram



1.2 Histogram (lose position information)

Histogram



2. Probability

Representing Color

- RGB
- HSV⇔RGB, (check powerpoint character color map)
- YCrCb
- CIE
- CMYK
- CIE LAB
- …

- H.264 ⇔YUV⇔RGB
- H.265: 3D

Sensing Color

- **In a 3-CCD video camera, the light path is split into three components which are passed through colored light filters and then imaged**
  - As a result - a color image contains three channels of information; red, green and blue image intensities
- **In a 1-CCD color camera, color information is obtained by covering the individual elements with a spatially varying pattern of filters, RGB**

Pinhole Camera Model

Pinhole Camera Model

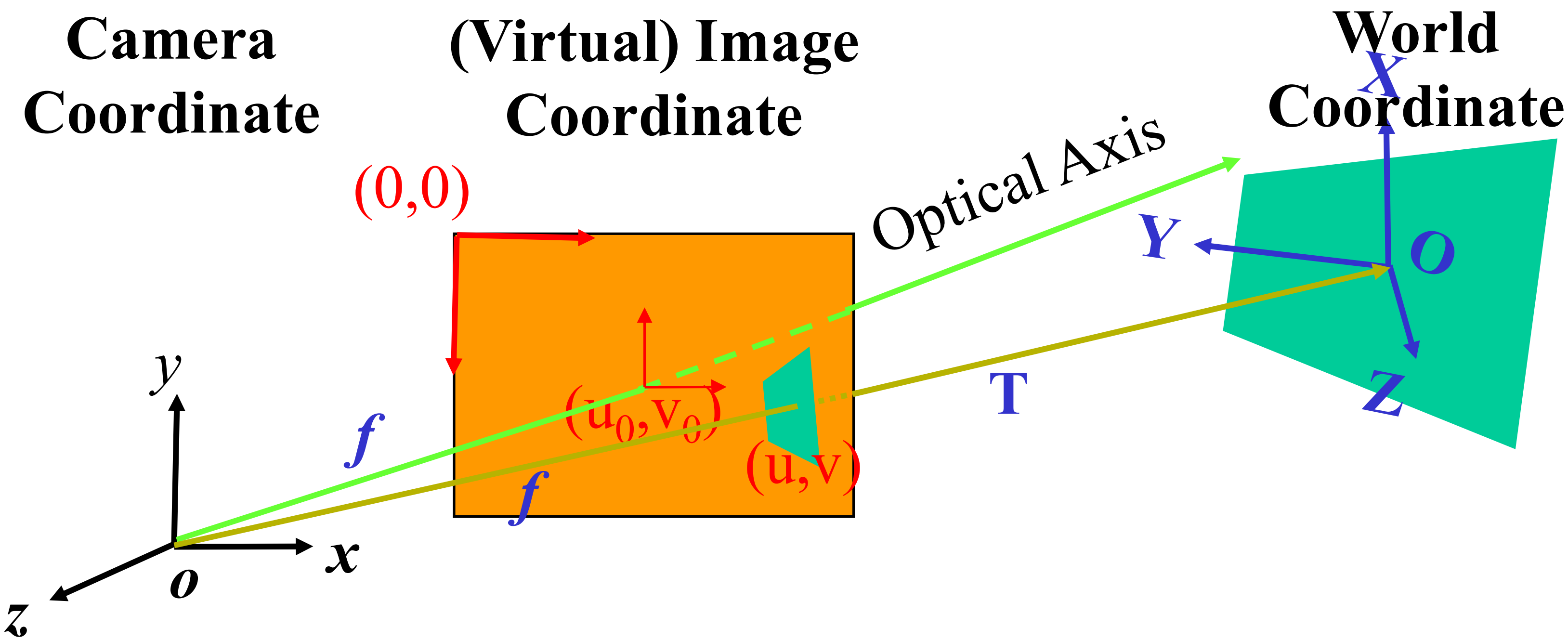
Camera Parameters

- **Intrinsic parameters (Camera Coordinate ⇔ Image Coordinate)**
  - Do not depend on the camera location
    - Focal length, CCD dimensions, lens distortion
- **Extrinsic parameters (World Coordinate ⇔ Camera Coordinate)**
  - Depend on the camera location
    - Translation, and Rotation parameters

❑ **Affine Transformation**

World to Camera to Image Coordinates (1/3)

3D (World) **3D** (Camera) **2D** (Image)



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{pmatrix} R_{11} & R_{21} & R_{31} \\ R_{12} & R_{22} & R_{32} \\ R_{13} & R_{23} & R_{33} \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

Extrinsic Parameters: 6 DOF

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Intrinsic Parameters
= Projection matrix

Camera Parameters: Intrinsic + Extrinsic Parameters (2/3)

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_{11} & R_{21} & R_{31} & T_1 \\ R_{12} & R_{22} & R_{32} & T_2 \\ R_{13} & R_{23} & R_{33} & T_3 \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Scale Factor: λ

Normalize to unit vector

$u = -fk_u(x/z) + \gamma\,(y/z) + u_0$

$v = -fk_v(y/z) + v_0$

- **Intrinsic Parameters:**
  - **Scale Factor**: $\alpha = -f\,k_u$, f: focal length
  - **Scale Factor**: $\beta = -f\,k_v$

    Aspect Ratio$= \alpha/\beta = k_u/k_v$
  - **Skew Factor:** γ
  - **Principal Point:** (u_0, v_0)
- **Extrinsic Parameters:**
  - **Rotation: R**
  - **Translation: T**

Pinhole Camera Model:
World ⇔ Camera ⇔ Image Coordinates

Framework: distortion vs undistortion??

Chessboard:棋盤格影像
Chessboard pattern:投影棋盤格

3D camera: 1 chessbd images 15

3D DLP: 15 images 1 pattern at

$$\min \sum_{n=1}^{N} \left(p_n^u - p''^u_n\right)^2 \qquad s\begin{bmatrix} u_n \\ v_n \\ 1 \end{bmatrix} = \begin{bmatrix} h11 & h12 & h13 & h14 \\ & & & \\ h41 & h42 & h43 & 1 \end{bmatrix}\begin{bmatrix} X_n^w \\ Y_n^w \\ Z_n^w \\ 1 \end{bmatrix}$$

$Z_n^w = 0, N_{rc}^c = 48$ $\quad P_n^w, Z_n^w = 0$ $\quad H_{t,3\times3}$

2D reprojection 2D

?? Distortion or undistortion?

1) From 2D image (distorted image) coordinate (u_n,v_n) to 2D pixel (undistorted) coordinate $P_n^u = [u_n^u \quad v_n^u]$

(1) Radial (barrel) distortion:

$$u_n^u = u_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$v_n^u = v_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

(2) Tangential distortion:

$$u_n^u = u_n + [2p_1 v_n + p_2(r^2 + 2u_n{}^2)]$$
$$v_n^u = v_n + [p_1(r^2 + 2v_n{}^2) + 2p_2 u_n] \qquad \textit{where } r^2 = u_n{}^2 + v_n{}^2$$

2) Perspective Projection: From 3D camera coordinate projects to 2D image coordinate

?? If this is correct, then D doesnot calculate included.??

$$s\begin{bmatrix} u_n \\ v_n \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma = 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_n^c \\ Y_n^c \\ Z_n^c = 0 \\ 1 \end{bmatrix}$$

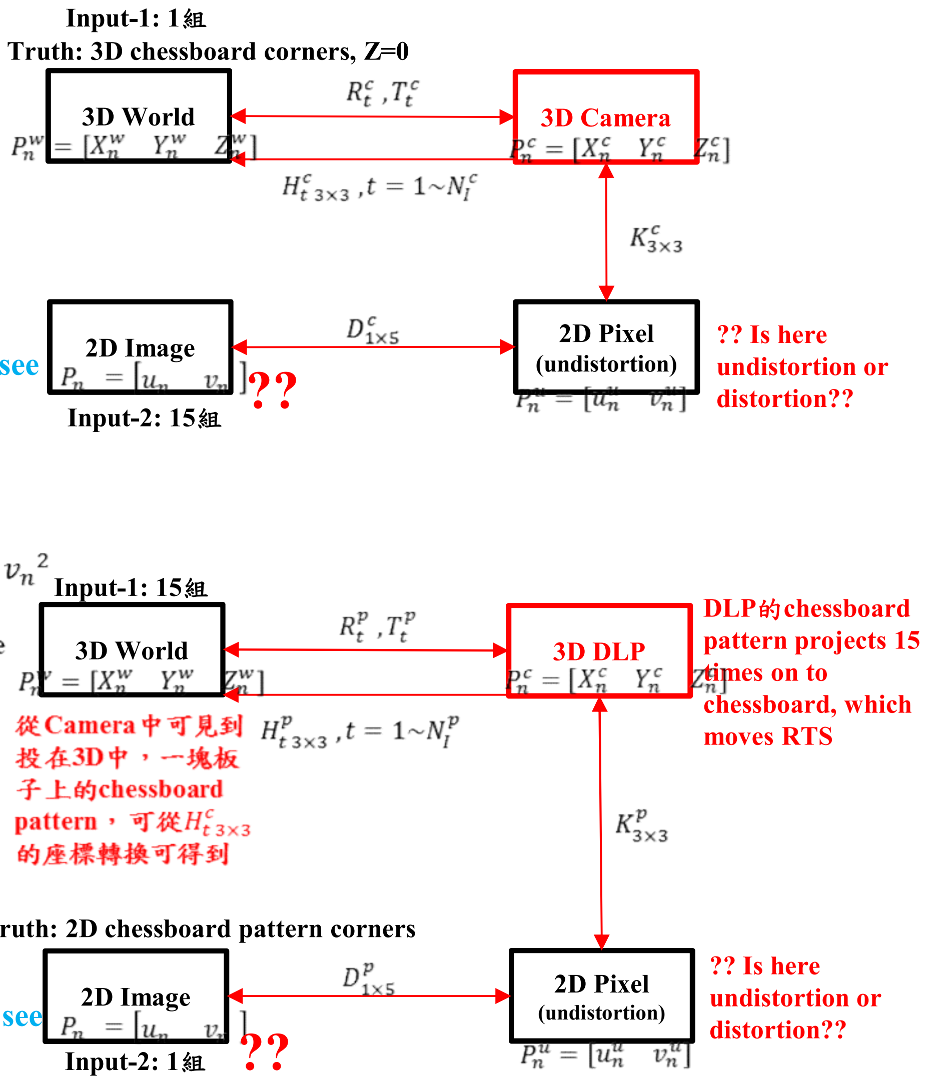
3) Affine: From 3D world coordinate to 3D camera coordinate

$$s\begin{bmatrix} X_n^c \\ Y_n^c \\ Z_n^c = 0 \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_n^{w'} \\ Y_n^{w'} \\ Z_n^{w'} = 0 \\ 1 \end{bmatrix}$$

Burn to DLP 裏的chessboard pattern，以此來投射

4) From 3D world coordinate to 2D coordinate

$$s\begin{bmatrix} u_n \\ v_n \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma = 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_n^{w'} \\ Y_n^{w'} \\ Z_n^{w'} = 0 \\ 1 \end{bmatrix}$$



Camera Parameters: Intrinsic + Extrinsic Parameters (3/3)

- **Intrinsic parameters: Projection matrix**
  - Is calculated via camera calibration
  - Most important intrinsic parameter is focal length *f*.
    - 3D-to-2D projection: From 3D world/camera coordinate projects to 2D image coordinate.
    - 2D-to-3D back-projection/reconstruction: From 2D image coordinate back-projects to 3D world coordinate.

- **Extrinsic parameters: Affine transformation**
  - Geometric transformation
    - 3D world/camera coordinate Vs. 3D world/camera coordinate
    - 2D image coordinate Vs. 2D image coordinate

- Corresponds to a dilation of the image
- It is most pronounced in optical systems with a wide field of view

$$x_u = x_d(1 + k_1 r^2 + k_2 r^4)$$

$$y_u = y_d(1 + k_1 r^2 + k_2 r^4)$$

$$where$$

$$r^2 = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$$

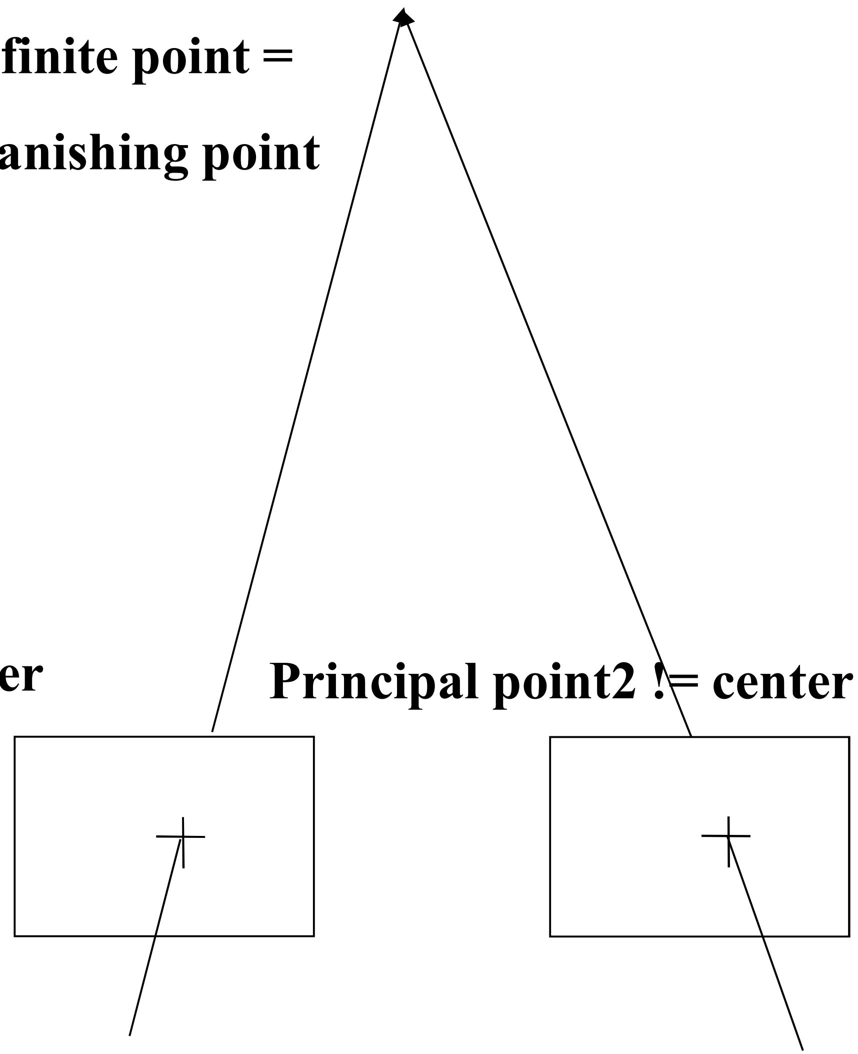
Vanishing Point

Infinite point =

Vanishing point

Principal point1 != center

Principal point2 != center



Principal point1 position at image 1 !=

Principal point2 position at image 2

Rotation Matrix Representation: Euler angles

Assume rotation matrix R results from successive Euler rotations of the camera frame around its X axis by ω, its once rotated Y axis by ϕ, and its twice rotated Z axis by κ, then

$$R(\omega, \phi, \kappa) = R_X(\omega) R_Y(\phi) R_Z(\kappa)$$

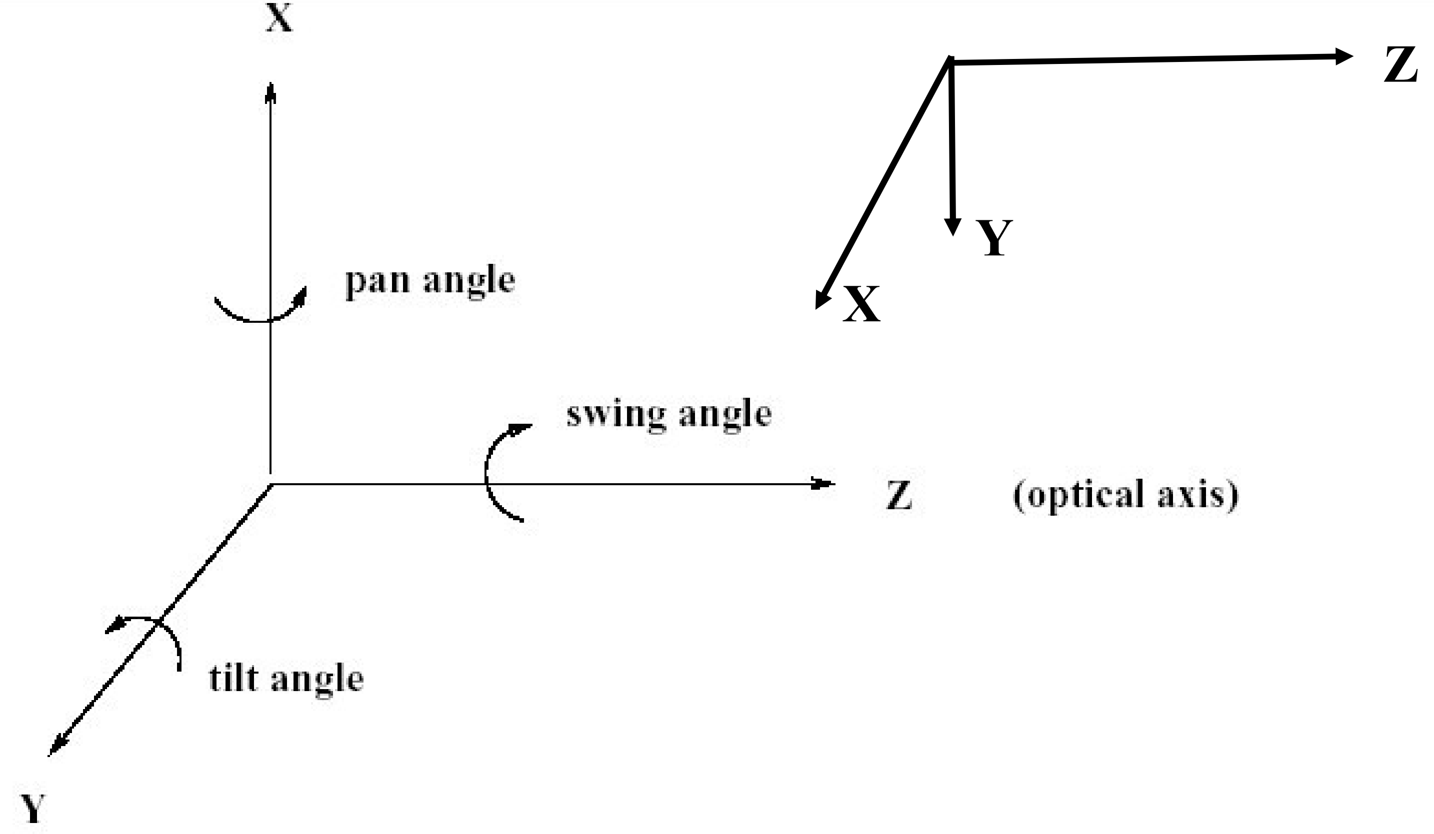
Different orders have different results

where ω, ϕ, and κ are often referred to as tilt, pan, and swing angles respectively.

roll

pitch **yaw**

Rotation Matrix Representation: Euler angles



- **In-Plane Rotation/Motion: Roll (or Swing)**
- **Out-of-Plane Rotation Motion: Pitch (or Tilt) and Yaw (or Pan)**

❑**DIP book**

$$R_x(\omega) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{pmatrix}$$

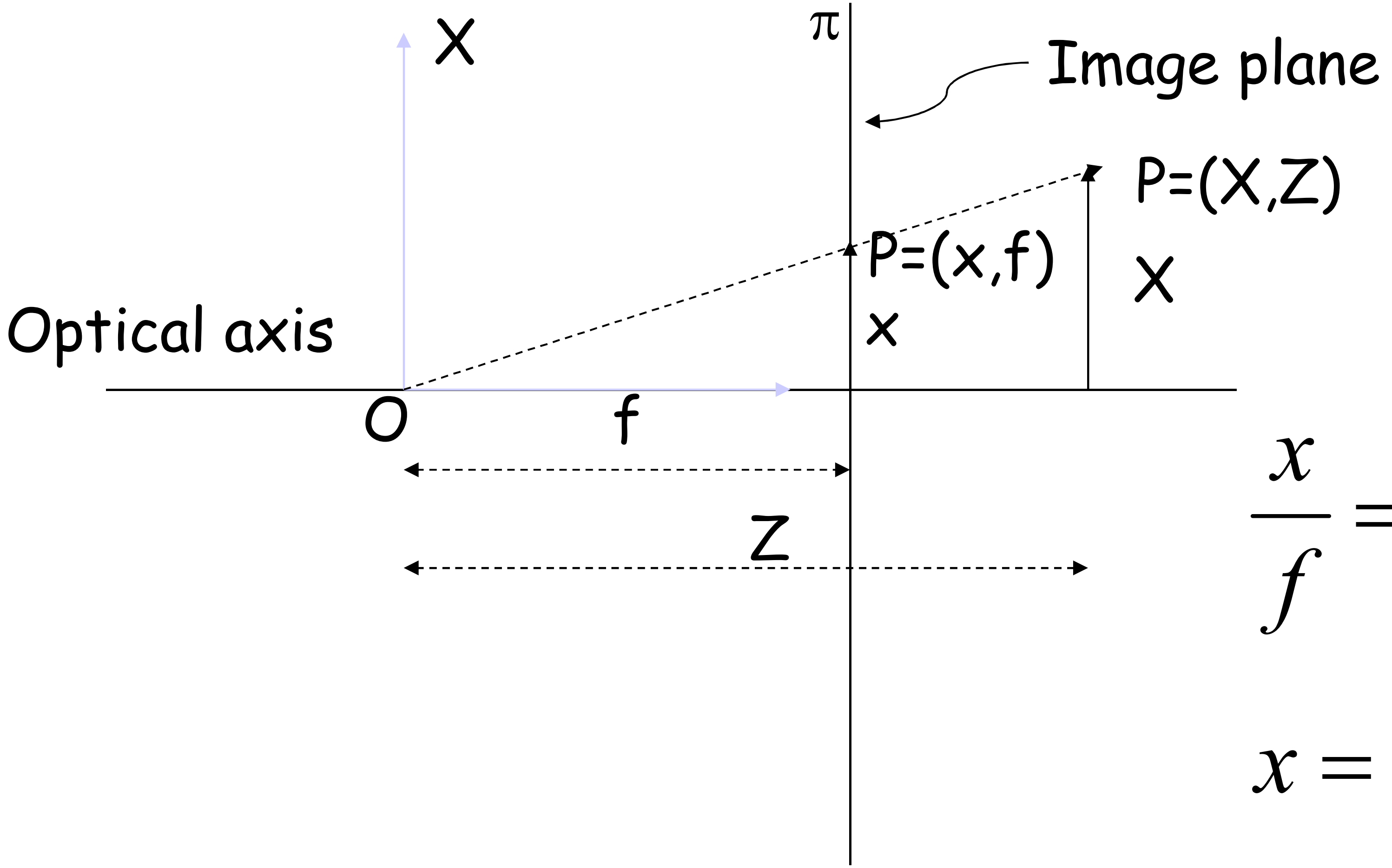
$$R_y(\phi) = \begin{pmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{pmatrix}$$

$$R_z(\kappa) = \begin{pmatrix} \cos\kappa & \sin\kappa & 0 \\ -\sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Pinhole Camera Properties

- Non-linear equations
- Lines project into lines
- Does not preserve angles
- Circles project into ellipses
- Farther objects appear smaller

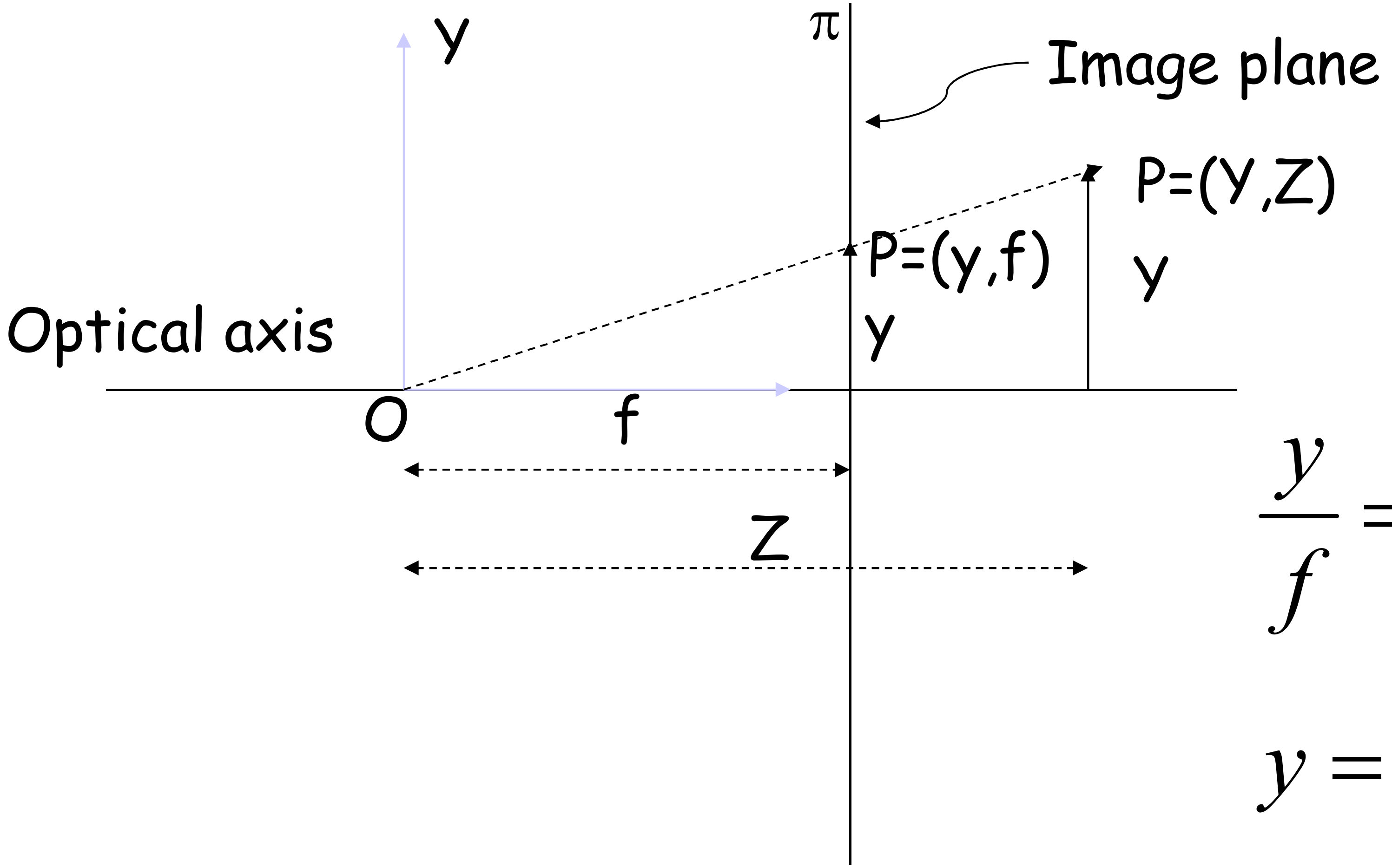
Camera to Image Coordinate: X-Z Plane



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

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

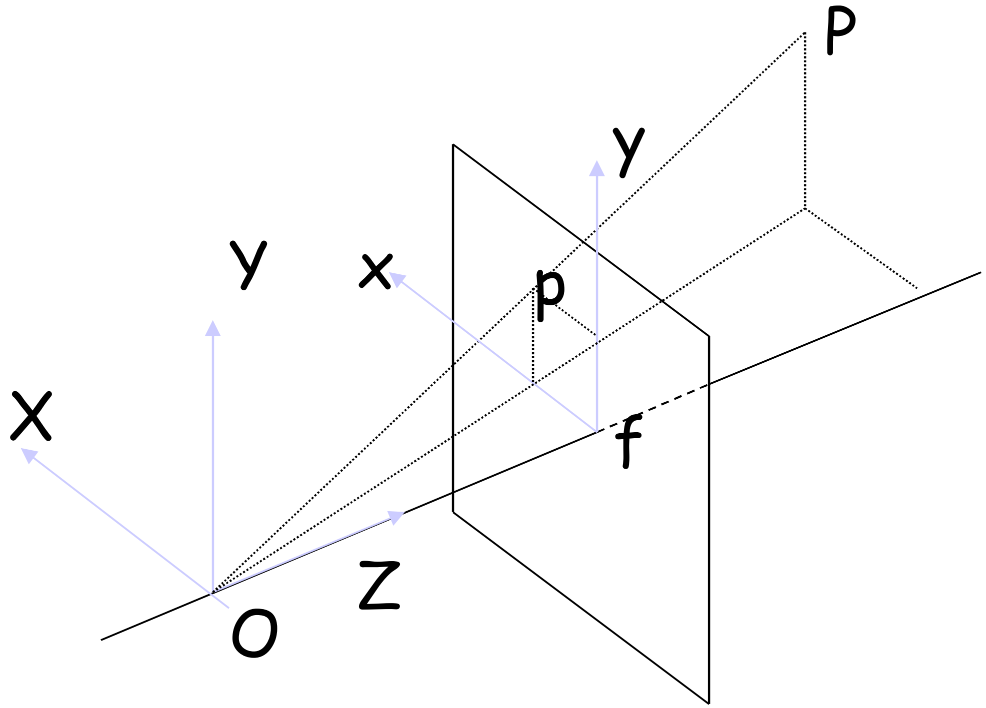
Camera to Image Coordinate: Y-Z Plane



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

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

Camera to Image Coordinate: X-Y-Z Camera Coordinate



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

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

- Non-linear equations
- Any point on the ray OP has image p !!

Camera to Image Coordinate

Non-linear

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

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

Linear

Using homogeneous coordinates:

Intrinsic parameter ?

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Scaling factor

$$x = \frac{x'}{w'} \qquad y = \frac{y'}{w'}$$

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$p = M_{\text{int}} \cdot P_c$$

Intrinsic parameters

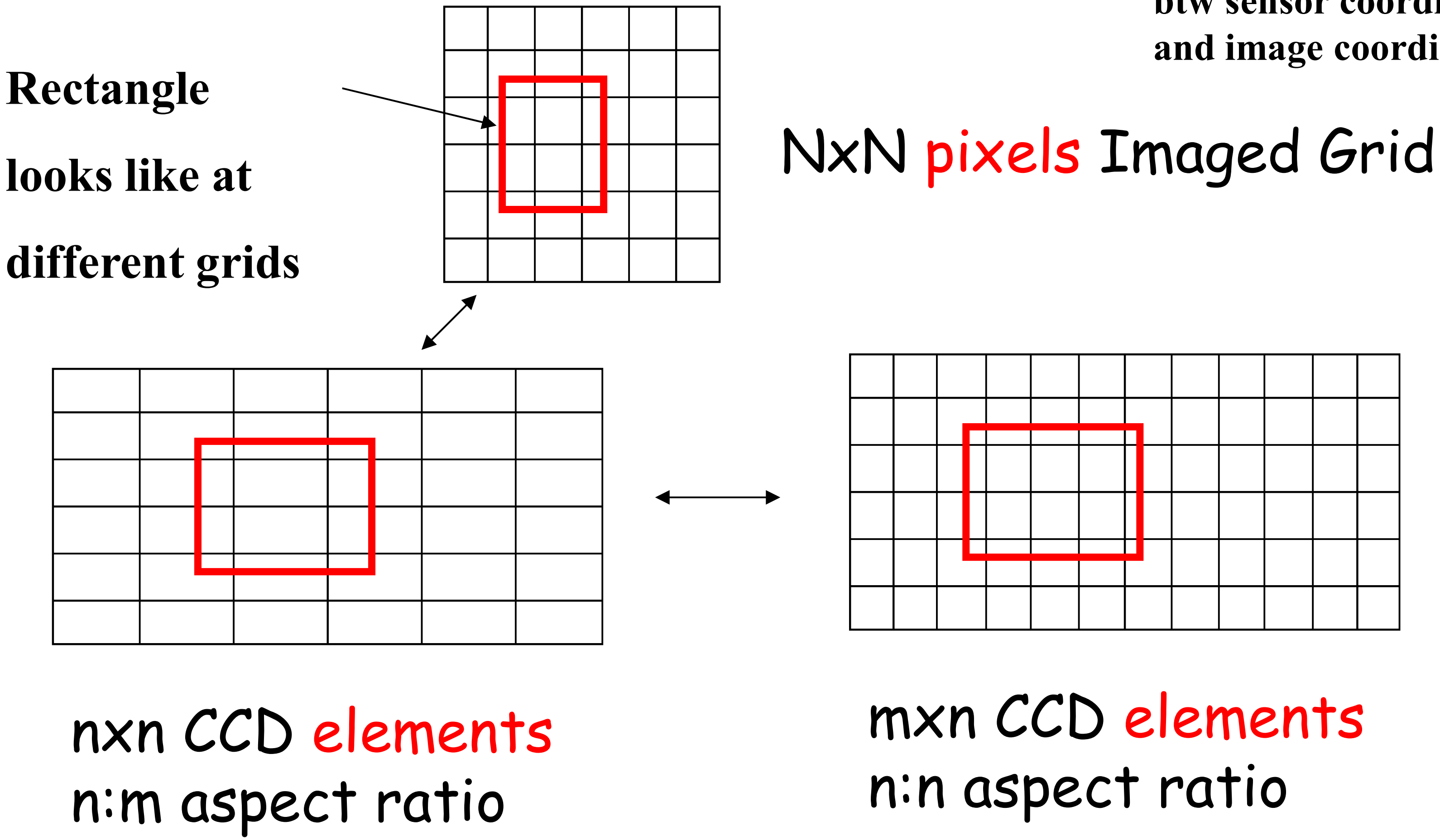
More Intrinsic Parameters

- **The CCD sensor is made of a rectangular grid nxm of photo sensors.**
- **Each photo sensor generates an analog signal that is digitized by a frame grabber into an array of NxM pixels.**

  - Scale Factor: $\alpha = -f\, k_{u,}$ f: focal length
  - Scale Factor: $\beta = -f\, k_v$

    Aspect Ratio= $\alpha/\beta = k_u/k_v$

Affine Transformation btw sensor coordinate and image coordinate



- **Scale Factor**: $\alpha = -f\,k_u$,
- **Scale Factor**: $\beta = -f\,k_v$

Aspect Ratio= $\alpha/\beta = k_u/k_v$

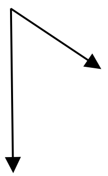
Effective Sizes: S_x and S_y

- There is an affine (s,o) relationship between image plane coordinates (x_{im}, y_{im}) and CCD sensor coordinates (pixel coordinate) (x,y)=(u,v)

In practice, we will assume that there is a 1-1 correspondence between CCD elements and pixels.

Ray projects to CCD

(u,v)



Ray projects to CCD pixel Coordinate | Image Coordinate

$$u = x = f\frac{X}{Z} = (x_{im} - (-o_x))s_x$$

$$v = y = f\frac{Y}{Z} = (y_{im} - (-o_y))s_y$$

s_x=1

s_y=1

$(o_x,o_y)=(0,0)$

Where o_x and o_y are the coordinates of the image center

- **Scale Factor**: α = -f $k_{u,}$
- **Scale Factor**: β = -f k_v

Aspect Ratio= $\alpha/\beta = k_u/k_v$

A More Complete M_{int}

$$
\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} f/s_x & 0 & -o_x & 0 \\ 0 & f/s_y & -o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
$$

$$
p = M_{\text{int}} \cdot P
$$

$$
x_{im} = \frac{x'}{w'} = \frac{f}{s_x}\frac{X}{Z} - o_x \qquad y_{im} = \frac{y'}{w'} = \frac{f}{s_y}\frac{Y}{Z} - o_y
$$

Note:

Sometimes, the image and the camera coordinate systems have opposite orientations:

$$x = f\frac{X}{Z} = -(x_{im} - (-o_x))s_x$$

$$y = f\frac{Y}{Z} = -(y_{im} - (-o_y))s_y$$

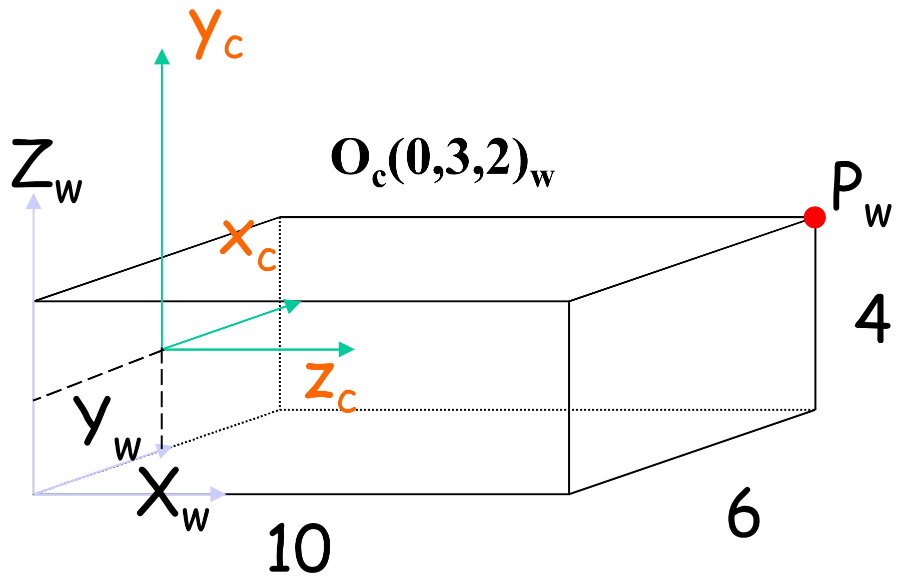
$$p = M_{\text{int}} \cdot P_c$$

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} -f/s_x & 0 & +o_x & 0 \\ 0 & -f/s_y & +o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x_{im} = \frac{x'}{w'} = \frac{f}{s_x}\frac{X}{Z} - o_x$$

$$y_{im} = \frac{y'}{w'} = \frac{f}{s_y}\frac{Y}{Z} - o_y$$

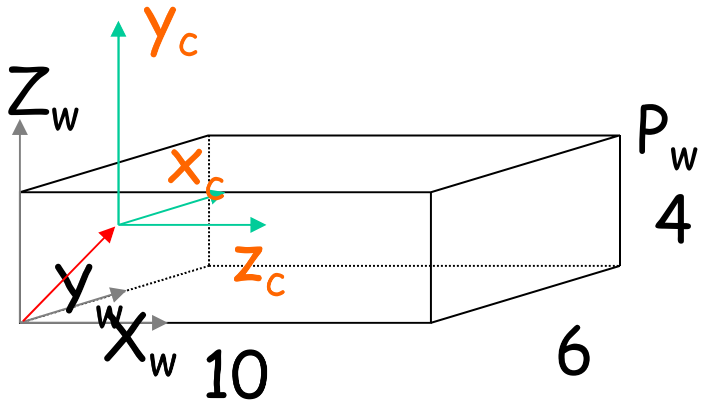
Example: World Coordinate to Camera Coordinate

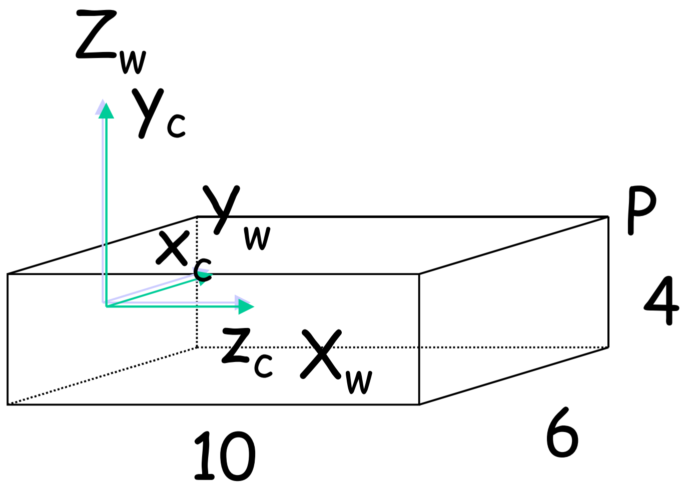


❑ Rigid Transformation (Affine Transformation)

- All rigid transformations can be written as follows
  - » $P_A = g_{AB}(P_B) = R^* P_B + T$
  - » Where R is an element of SO(3) and $T = g_{AB}(0)$ is in R^3
- If points and vectors are represented with homogenous coordinates then rigid transformations can be represented by matrices of the form

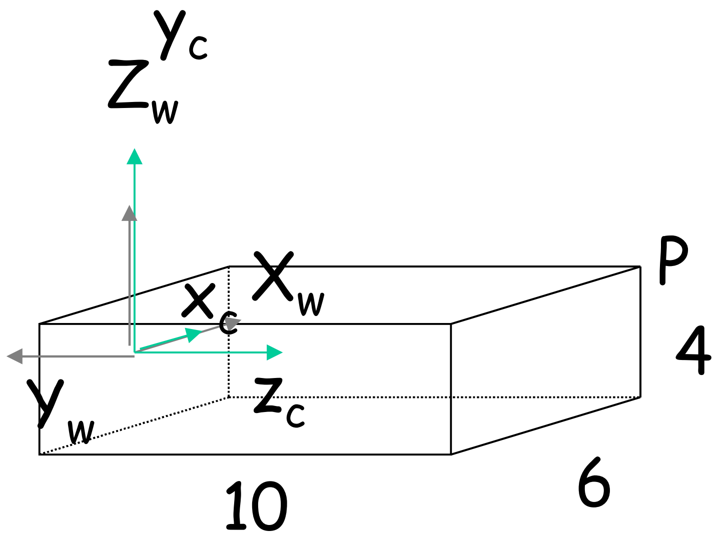
$$g = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \in R^{4\times 4}, R \in SO(3), T \in R^3$$





- Translate W to C:

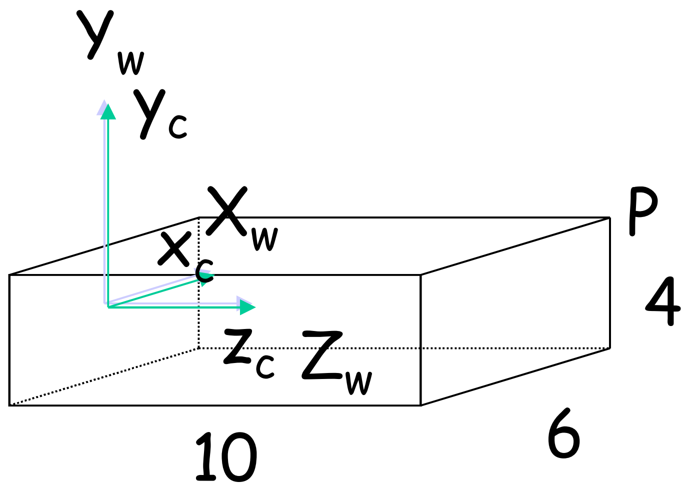
$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Align X_w to X_c

- Rotate W 90 dg. Ccw around Z

$$R_{z,90} = \begin{bmatrix} \cos 90 & \sin 90 & 0 & 0 \\ -\sin 90 & \cos 90 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



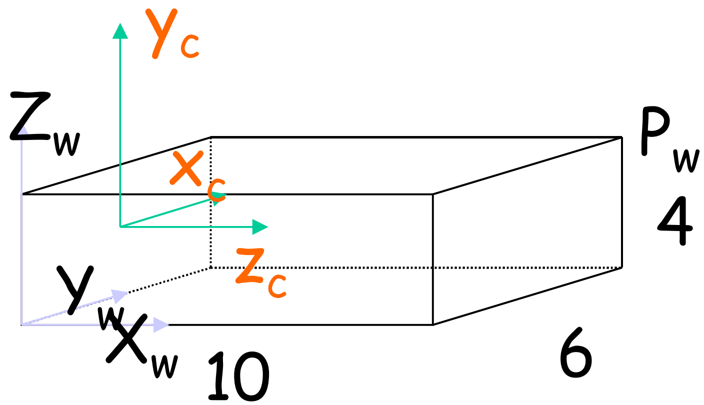
Align Y_w to Y_c and Z_w to Z_c

- Rotate W 90 dg. ccw around X (viewed from x direction)

$$R_{x,90} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90 & \sin 90 & 0 \\ 0 & -\sin 90 & \cos 90 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Why homogenous matrix?

$O_c(0,3,2)_w$



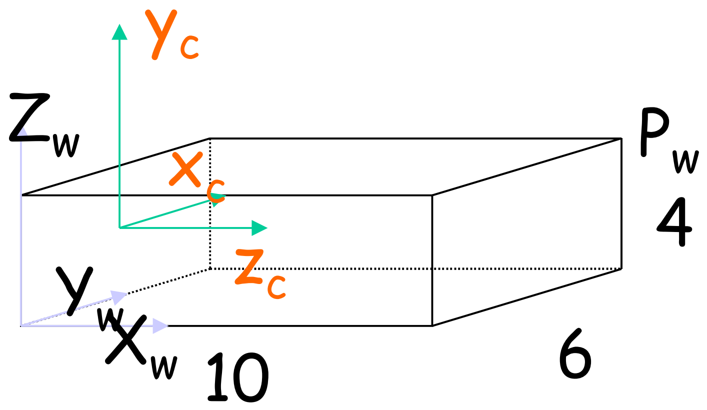
?

$$\begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix} = M_{\text{ext}} \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} =$$

$$= \left[\begin{array}{ccc|c} 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -2 \\ 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix}$$

$$P_c = R \cdot T \cdot P_w = M_{\text{ext}} \cdot P_w$$

Another Way by Using Unit Vector



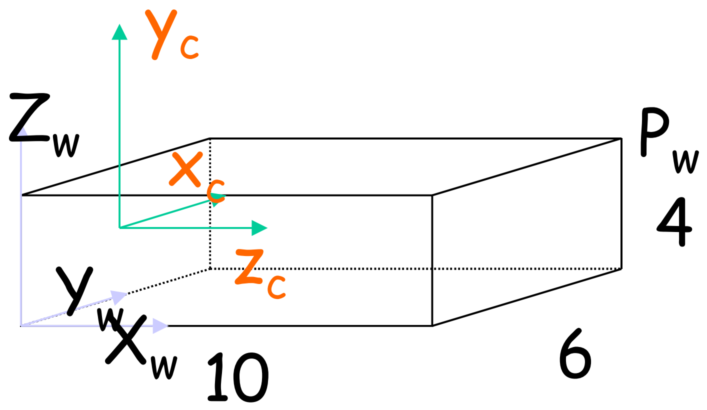
- Find R in one shot:

If W (1, 0, 0), then C (0,0,1)

$$\underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Normalization => Unit Vector}} \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{W} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{C}$$

$$\begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

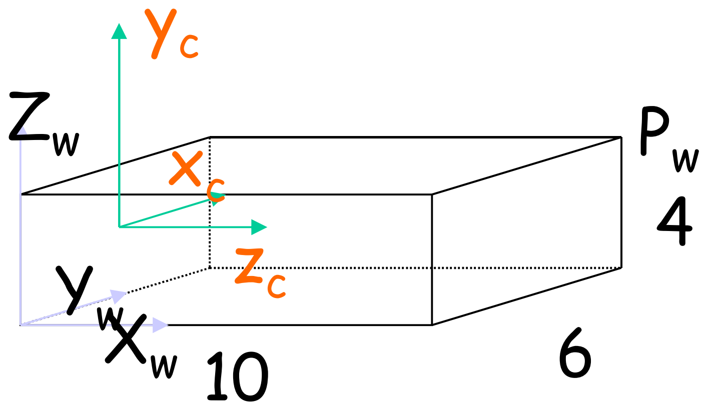
Normalization => Unit Vector



If W (0, 1, 0), then C (1,0,0)

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{W} = \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{C}$$

$$\begin{bmatrix} r_{12} \\ r_{22} \\ r_{32} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



If W (0, 0, 1), then C (0,1,0)

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

C

W

$$\begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Xw => Zc

Yw => Xc

Zw => Yc

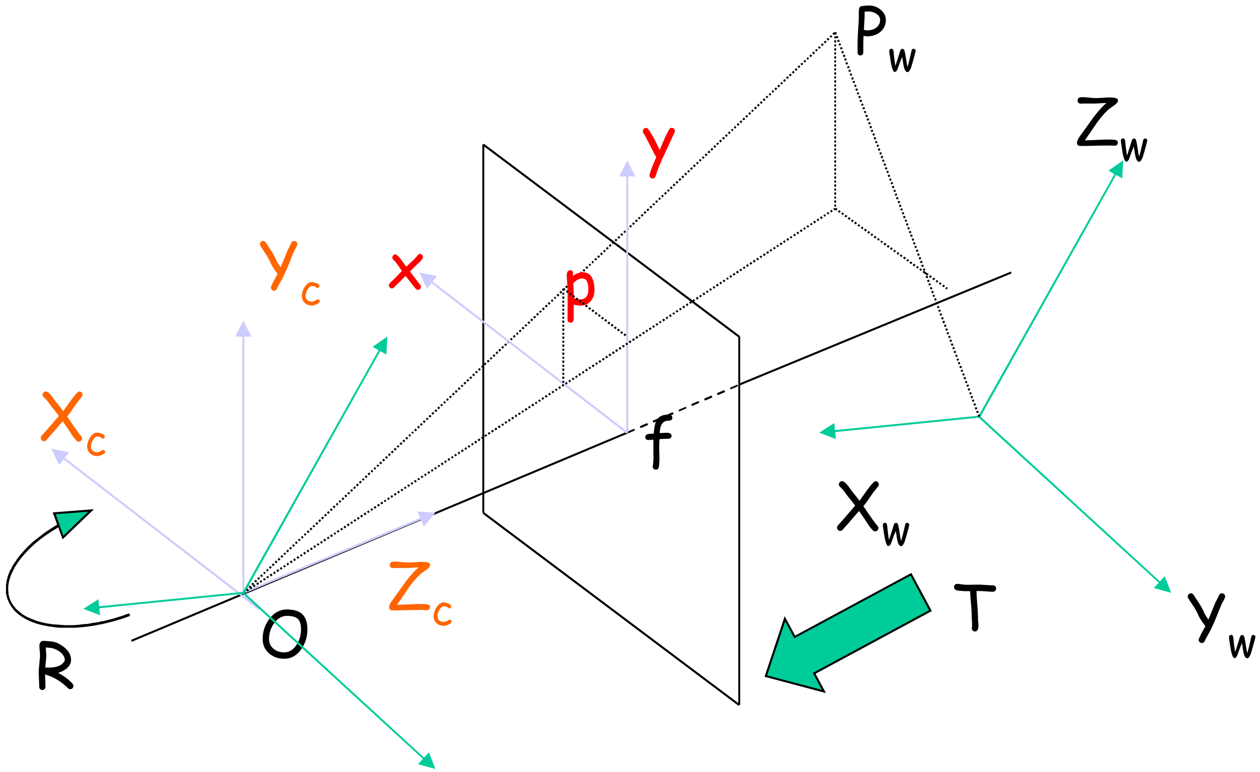
$$R = \begin{array}{c} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \begin{array}{cccc} \text{Xw} & \text{Yw} & \text{Zw} & \end{array} \end{array}$$

World Coordinate to Camera/Image Coordinate

- **PROBLEM: In general, the camera coordinate system is not aligned with the world coordinate system!**

- **SOLUTION: Find a transformation between coordinate systems.**

World Coordinate to Image Coordinate



W to C: $$P_c = R \cdot T \cdot P_w = M_{\text{ext}} \cdot P_w$$

C to I: $$p = M_{\text{int}} P_c = M_{\text{int}} M_{\text{ext}} \cdot P_w$$

- Extrinsic parameters (R, T):

W to C: $$P_c = R \cdot T \cdot P_w = M_{\text{ext}} \cdot P_w$$

- Intrinsic parameter (f):

C to I: $$p = M_{\text{int}} P_c = M_{\text{int}} M_{\text{ext}} \cdot P_w$$

W to I:

R_x, R_y, R_z

T_x, T_y, T_z

$$p = M \cdot P_w$$

M is 3x4
M has 6 dof (3 R + 3 T)
(assuming *f* is known)

W to I:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_{11} & R_{21} & R_{31} & T_1 \\ R_{12} & R_{22} & R_{32} & T_2 \\ R_{13} & R_{23} & R_{33} & T_3 \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Scale Factor: λ

Normalize to unit vector

$$p(x, y) = M \cdot P_w(X_w, Y_w, Z_w)$$

M is 3x4
M has 6 dof (3 R + 3 T)
(assuming f is known)

- **Each image point *p*(x, y) must satisfy:**

$$\begin{bmatrix} x' \\ y' \\ s' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{31} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Homogenous Matrix M

$$x = \frac{x'}{s'} = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + \cancelto{1}{m_{34}}}$$

$$y = \frac{y'}{s'} = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + \cancelto{1}{m_{34}}}$$

Non-linear equations

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}Xx - m_{32}Yx - m_{33}Zx - m_{34}x$$

$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}Xy - m_{32}Yy - m_{33}Zy - m_{34}y$$

Coordinate Transformation: Homogenous Matrix M

- **M has 12 (11) entries/unknown, but only 6 dof (3 R + 3 T).**
- **Each image point provides 2 equations, so it needs at least 6 image points (2D (x,y)). (Overdetermined matrix)**
- **Solve a system of linear equations.**
  - Non-linear equation to linear equation by Homogenous Matrix M.

2D I to 2D I: 6 unknown needs 3 points => Affine

$$\begin{bmatrix} x' \\ y' \\ \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

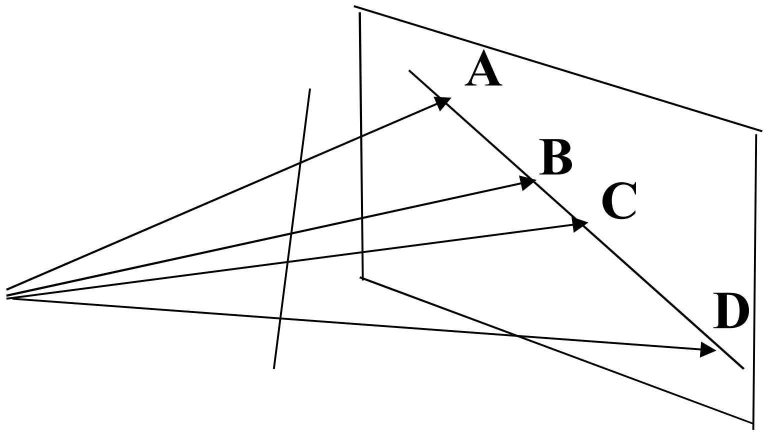
In Reality: Image => Camera => World Coordinate

- Inverse transformation matrix

The Cross Ratio

- Given 4 points in projective plane with homogenous coordinates p1, p2, p3, and p4, the cross ratio of this point set can be computed

$$[ABCD] = \frac{\dfrac{AC}{BC}}{\dfrac{AD}{BD}}$$

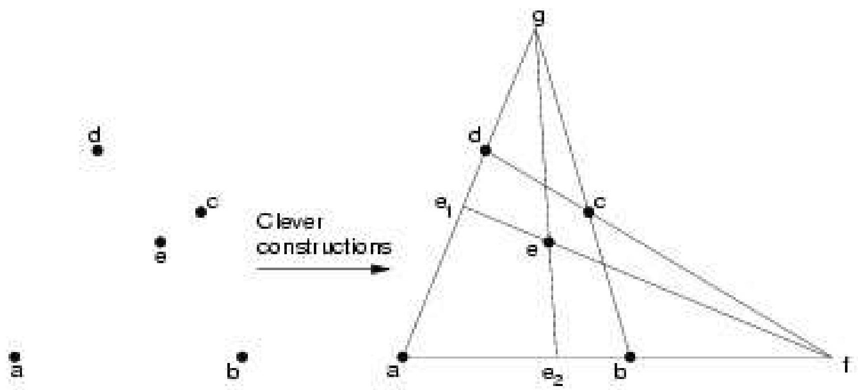


[ABCD]=[BADC]=[CDAB]=[DCBA], [ABCD]=1/[ABDC]

- Projective transformations of projective plane preserve cross ratios

Five point invariants on the plane

Even though we have developed the cross-ratio for four points *on a line*, we can also use it in planar imaging situations. We need 5 distinguished points to form invariants on the plane.



Given the image of the 5 points a . . . e, we can use the invariant property of intersection to find 4 more distinguished points: f, the intersection of the extrapolated lines a-b and d-c; g, similarly; e_1, the intersection of the line joining f and e with the side a-d, and e_2 similarly. We can now form two cross-ratios:

$$\begin{aligned} \delta_1 &= \text{cross-ratio of } \{a, e_2, b, f\} \\ \delta_2 &= \text{cross-ratio of } \{a, e_1, d, g\} \end{aligned}$$

These will be the same measured in any view of the 5 points.

Perspective projection:

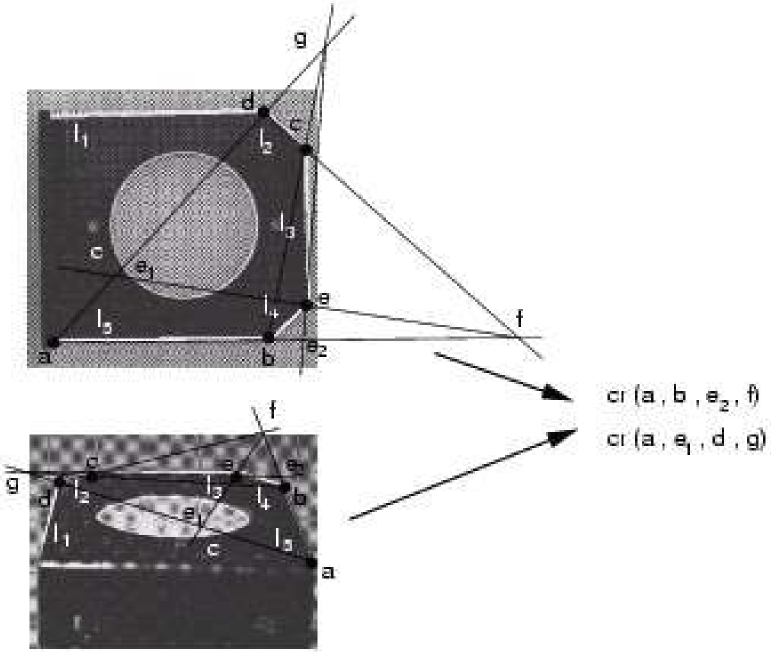
sometimes l_{ad} // l_{bc} and l_{ab} // l_{dc} ?

=> Vanishing points: g and f

Jenn-Jier James Lien

Five point

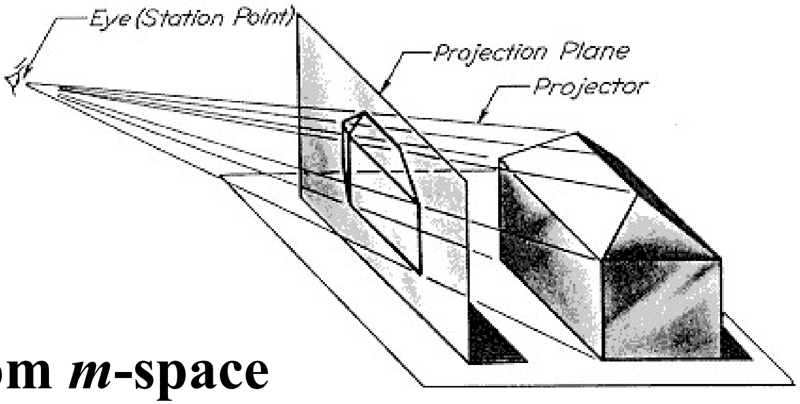
Here's an example of how we could use the five point invariants for object recognition.



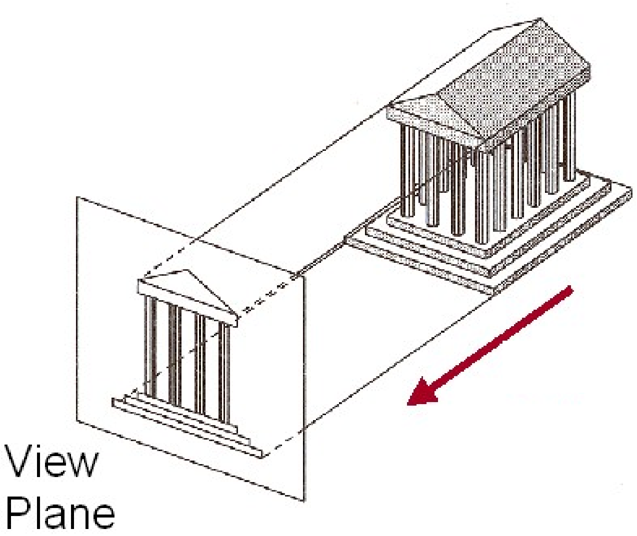
We identify five distinguished points a . . . e at the corners of the bracket and construct intersections to find four more distinguished points f, g, e_1 and e_2. We now have two sets of four collinear points, $\{a, b, e_2, f\}$ and $\{a, e_1, d, g\}$, for which we can calculate cross-ratios. These will be the same in any view, and can be used to identify the bracket. Other configurations of five planar points will yield different cross-ratios.

How to find the intersection point, e_1 or e_2, of lines ?

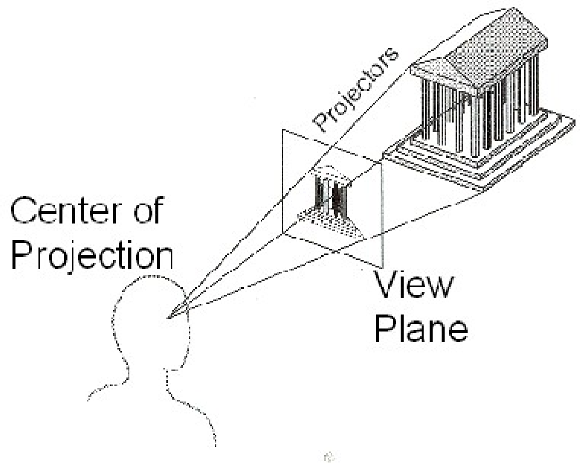
Projective Geometry



- **Projection – a transformation from *m*-space to *n*-space (*m*>*n*)**
  - For vision and graphics, it's 3D to 2D

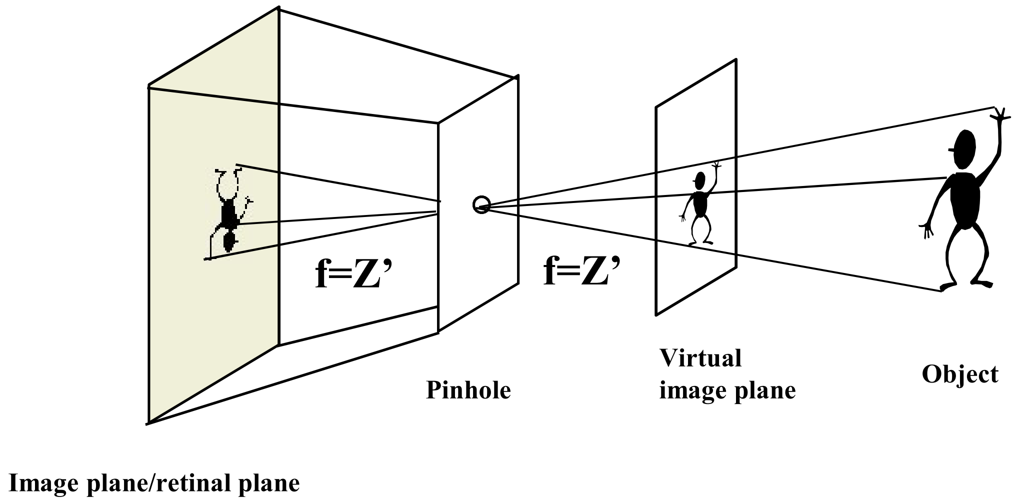


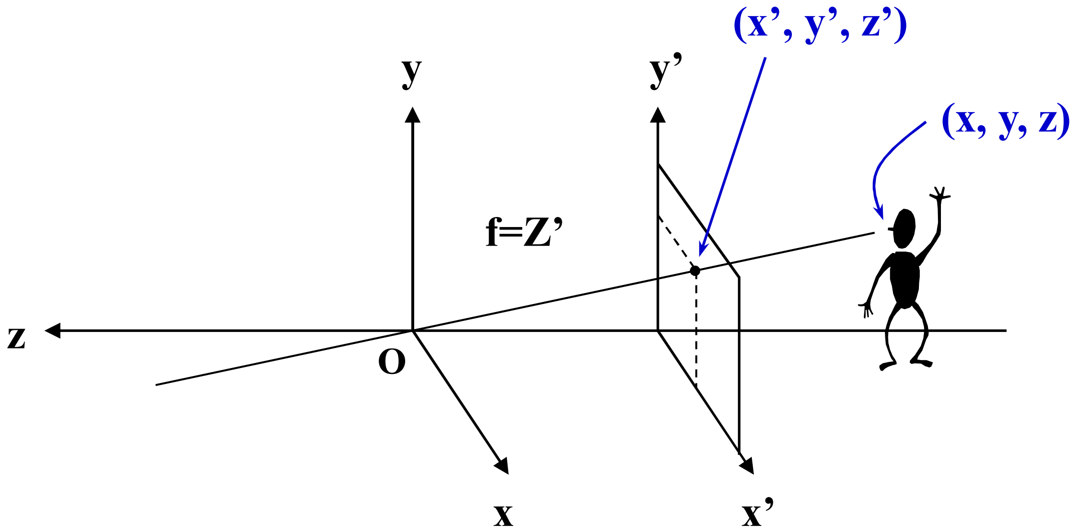
Orthogonal planar



Perspective planar

Perspective Projection





Perspective Projection Model

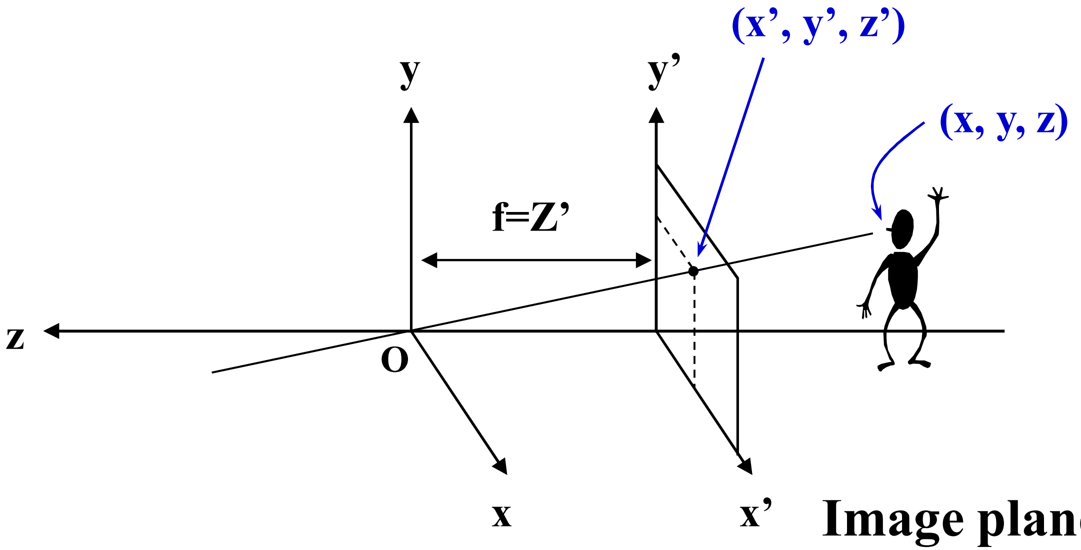


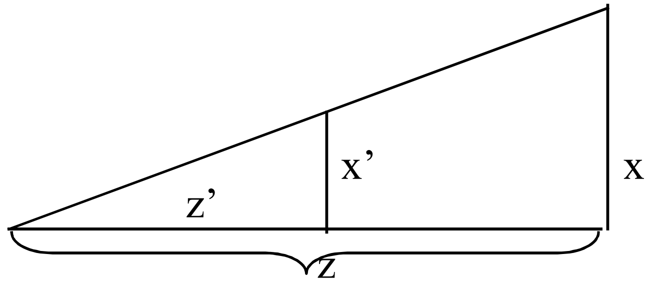
Image plane

Projection from (x,y,z) to (x',y',z'):

$$x' = z' \frac{x}{z}$$

$$y' = z' \frac{y}{z}$$

$$z' = z' \;= \mathbf{f}$$



$$x' = z'\frac{x}{z}, \quad y' = z'\frac{y}{z}, \quad z' = z'$$

- **Can we compute this in a 3x3 matrix?**
  - No, it's not linear => exists denominator
- **Instead, use *homogeneous coordinates* for image plane coordinate**

Find ?

Ax=b

$$\begin{bmatrix} x'' \\ y'' \\ z'' \\ w'' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/z' & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x''/w'' \\ y''/w'' \\ z''/w'' \end{bmatrix}$$

Point or sample in high dimension:
20x20 pixels face image is a
point/sample/vector in 400 dimensions

Homogenous Coordinates ??

- **A simplistic view**
  - Homogenous coordinates (matrix) are a mechanism that allows us to associate points and vectors in space with vectors in R^4 Real
- **?? Consider the vector $(p, q, r, s)^t$**
  - If s is not equal to zero then this vector denotes the point with coordinates $(p/s, q/s, r/s)^t$. $(p/s, q/s, r/s, 1)^t$ is as unit vector or direction $\frac{u}{|u|}$?
  - If s is equal to zero then this vector denotes the vector in the direction $(p, q, r)^t$ => Unit Vector=? $\frac{u}{|u|}$
  - N.B. the vector $(0, 0, 0, 0)^t$ is explicitly disallowed
- **Note that under this association $(p, q, r, s)^t$ and $a*(p, q, r, s)^t$ where a is an arbitrary non-zero scalar denote the same entity**
- **This means that we cannot make distinctions between vectors that point along the same ray but have different magnitudes or signs**
- **Location/Point u = Unit vector * Amplitude = $\frac{u}{||u||}$* ||u||**
  => Vector [p/s q/s r/s 1] s

Unit vector = direction

- **SVD: $A= UWV^T$; $AV=UW$**

where U: Orthonormal Matrix, W : Scale Matrix, V: Rotation (Orthonormal) Matrix

Advantage of Homogenous Matrix

- **Homogenous coordinates allow us to**
  - Express rigid transformations in terms of matrix multiplication
    - » (Rigid motion Vs. Non-rigid motion
    - » Affine Transformation)
  - Treat points and vectors in a unified framework
  - Easy to do optimization process: (vector and matrix => Matlab)

    1) Ax=0, closed-form solution…, A uses

       (1) SVD (eigenvector => eigenvalue closer to 0)

       (2) …

    2) Ax=b,

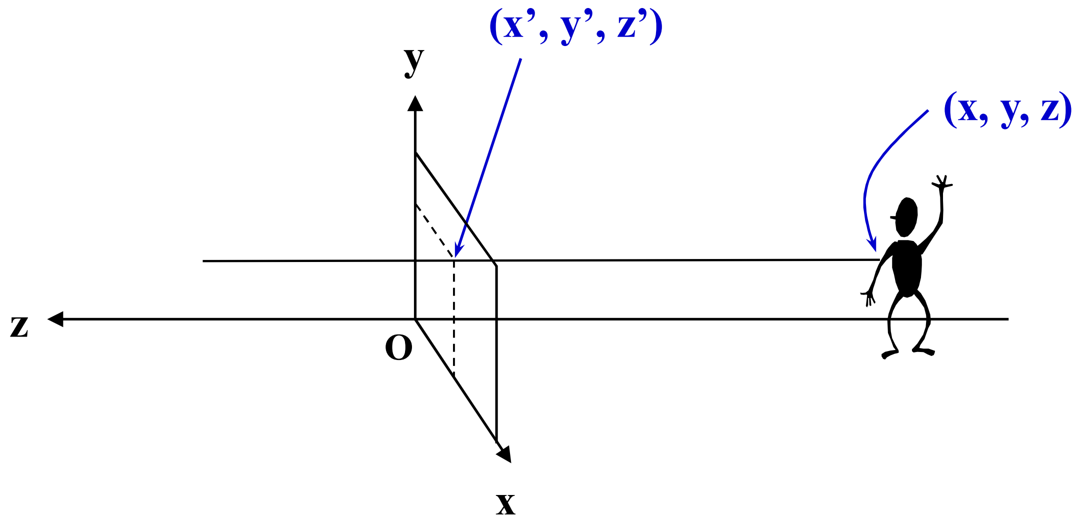
       (1) Pseudo inverse,

       (2) SSD, (LSE: Least Squares Error, MMSE: Minimum Mean Square Error, Mahalanobis distance) $\| b - b' \|^2$

       (3) Lagrange (with constraint)

       (4) …

Orthogonal/Orthographic Projection Model



Projection from (x,y,z) to (x',y',z'):

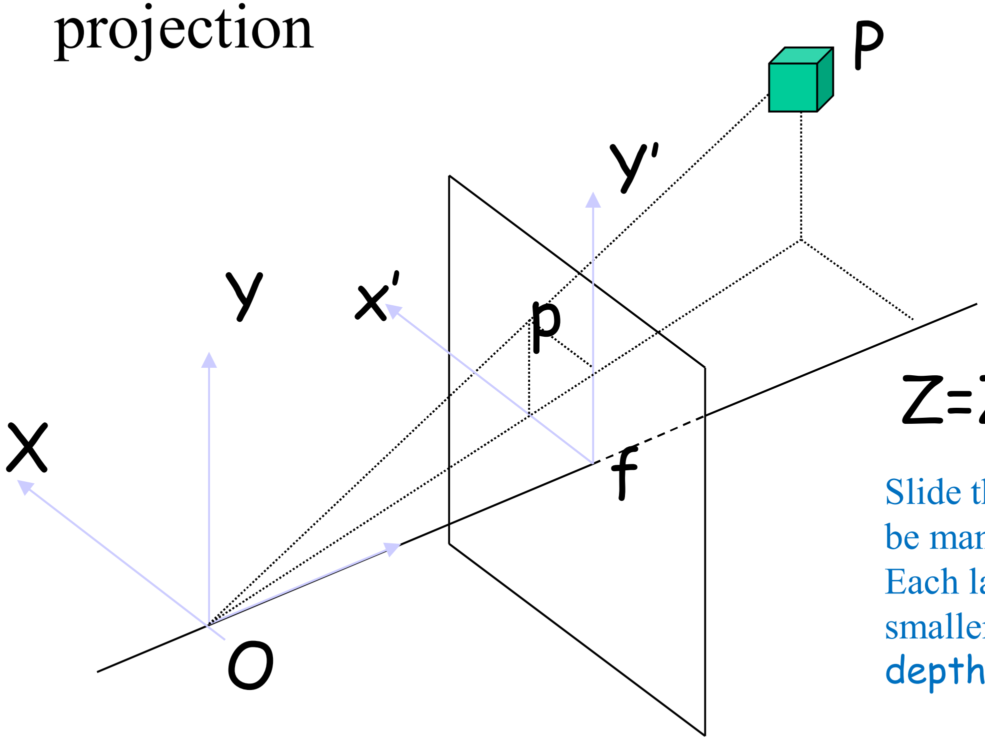
$$x' = x \qquad\qquad x' = kx$$
$$y' = y \qquad \text{or} \qquad y' = ky$$
$$z' = const \qquad\qquad z' = const$$

k is constant, not variable

Divide an object to
different small layers

Weak Perspective Model

❑Appropriate when the depth of field of the relevant objects is small compared to the distance of those objects from the center of projection



Z=Zo + δ

Slide the depth D to be many layers. Each layer has smaller (thinner) depth δ

$$x' \approx f \frac{X}{Z_o}$$

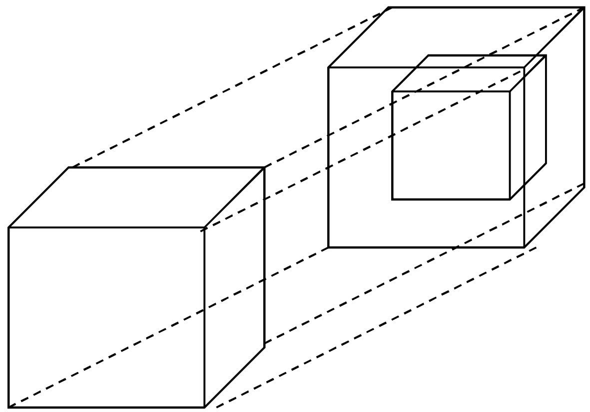
$$y' \approx f \frac{Y}{Z_o}$$

Not Z

- Object depth δ << Camera distance Zo
- Linear equations !!

$$x \approx f\,\frac{X}{Z_o}$$

$$y \approx f\,\frac{Y}{Z_o}$$



Weak perspective = Orthographic projection +
Isotropic Scaling

Projection

- **When Δz is small (little depth variation in scene), then orthographic is a good approximation to perspective**

  i.e., Big Δz divides into several small Δz

$$x' = z' \frac{x}{z} \approx kx \qquad k \text{ is constant, not variable}$$

$$y' = z' \frac{y}{z} \approx ky$$

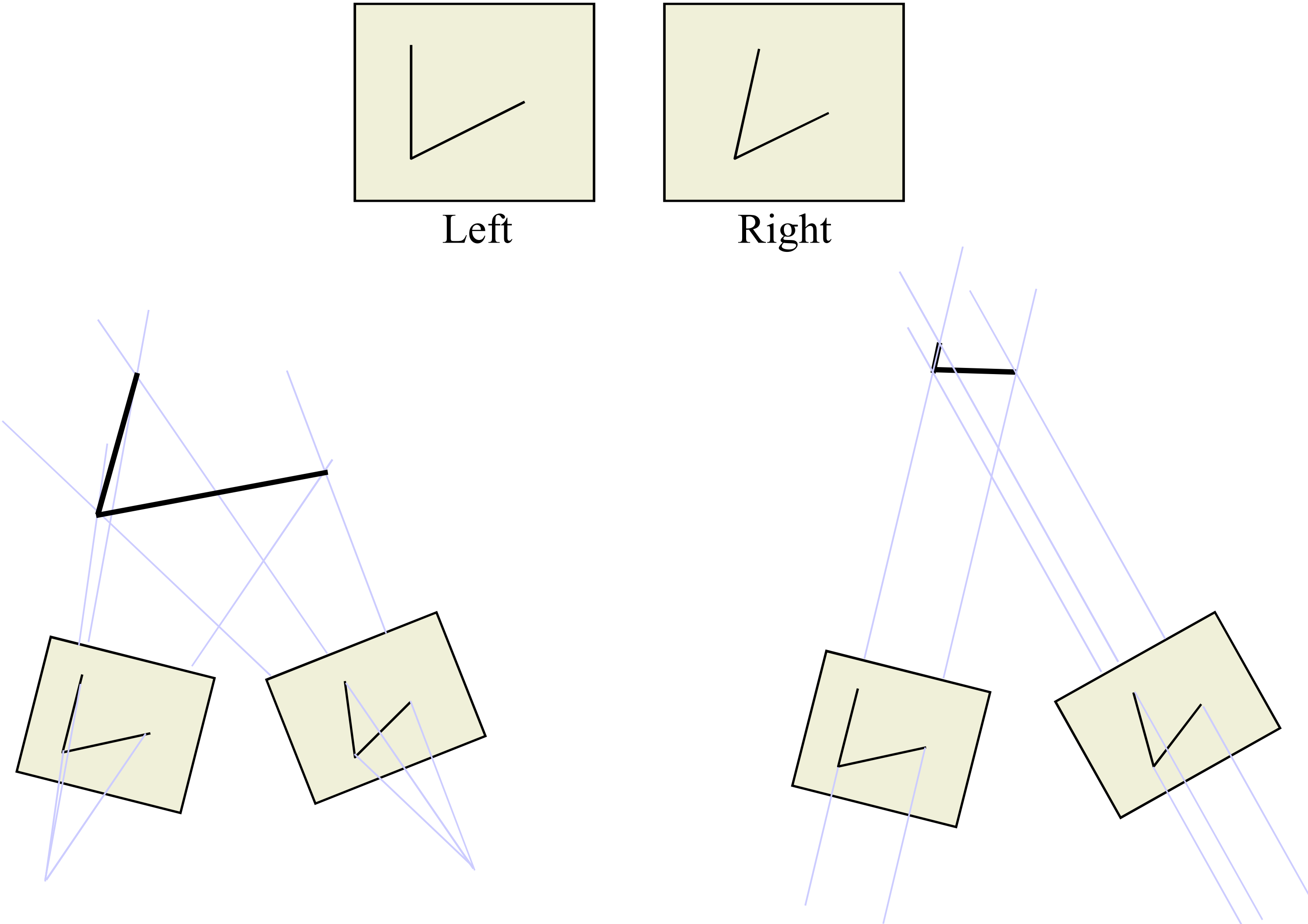
- Questions:
  - Points and lines project to what in the image?
  - Parallel lines? Circles?
  - The camera performs a perspective projection – what does it mean to "use" another projection, like orthogonal?
  - Why do we care?

Perspective Vs. Orthogonal (Parallel)

- **Perspective (pinhole camera model)**
  - Size varies inversely with distance – looks realistic
  - Distance and angles are not preserved (in general)
  - Parallel lines do not remain parallel (in general)

- **Orthogonal**
  - Good for exact measurements
  - Parallel lines remain parallel
  - Angles are not preserved (in general)
  - Less realistic looking

Perspective Vs. Orthogonal: Example - Shape Reconstruction



2D (Planar) Motions (Transformation) (1/6) jj

◆ 2D Planar Motion Transformations:

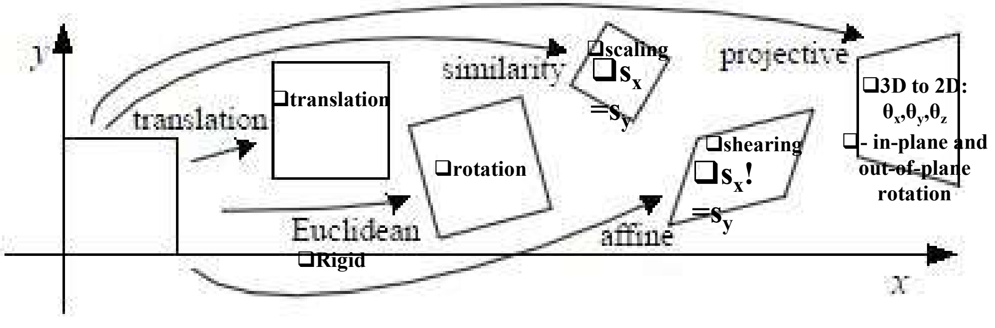


Figure 2: *Basic set of 2D planar transformations*

I : Identity matrix (2x2).

$\tilde{x}$ (x, y , 1) : Homogeneous or projective 2D coordina

$\tilde{x}'$ (x', y') : Transformed coordinate

t (t_x, t_y) : Translation.

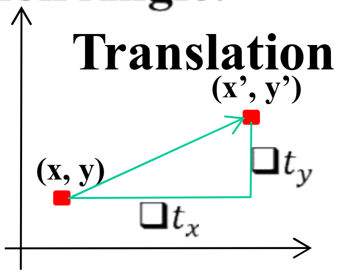
R : Rotation Matrix.

θ : Rotation Angle.

1) Translation Transformation: Translation

$$\tilde{x}' = \tilde{x} + t \quad or \quad \tilde{x}'_{2x1} = [I_{2x2}\ t_{2x1}]\tilde{x}_{3x1}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

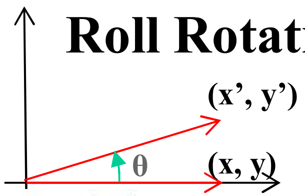


2) Euclidean Transformation (Rigid, 2D rigid body): Rotation + Translation

$$\tilde{x}' = R\tilde{x} + t \quad \textbf{or} \quad \tilde{x}' = [R \quad t]\tilde{x}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

is an orthonormal rotation matrix with $RR^T = I$ and $|R| = 1$.



$a = \cos\ \theta$

$b = \sin\ \theta$

$a^2+b^2=1$

2D (Planar) Motions (Transformation) (2/6) jj

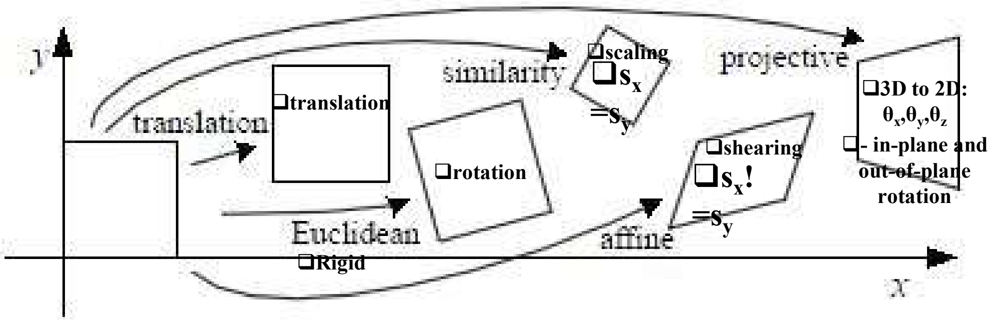


Figure 2: *Basic set of 2D planar transformations*

$\tilde{x}$ (x, y , 1) : Homogeneous coordinates

$\tilde{x}'$ (x', y') : Transformed coordinates.

t (t_x, t_y) : Translation.

R : Rotation matrix.

s : Scale factor.

θ : Rotation Angle.

3) Similarity Transformation (Scaled rotation): Scaling (s_x=s_y) + Rotation + Translation

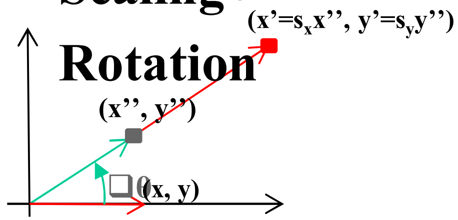
$$\tilde{x}' = sR\tilde{x} + t \quad \textbf{or} \quad \tilde{x}' = [sR \quad t]\tilde{x}$$

$$a : s\cos\theta$$

$$b : s\sin\theta$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad sR = \begin{bmatrix} s\cos\theta & -s\sin\theta \\ s\sin\theta & s\cos\theta \end{bmatrix}$$

Scaling+ Rotation



where we no longer require that $a^2 + b^2 = 1$.

2D (Planar) Motions (Transformation) (3/6) jj

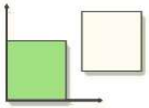
4) Affine Transformation: Scaling (s_x!=s_y) + Shearing + Rotation + Translation

any transformation that preserves parallel lines

$$\tilde{x}' = A\tilde{x} \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

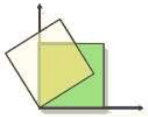
Translation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



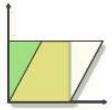
Rotation:

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



Shear:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & k_x \\ k_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Scaling:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ y_1 \\ w_1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} = 0 & h_{21} = 0 & h_{22} = 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Linear Combination:

$$x' = \frac{x_1}{w_1} = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} = \frac{h'_{00}x + h'_{01}y + h'_{02}}{h'_{20}x + h'_{21}y + 1}$$

$$y' = \frac{y_1}{w_1} = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}} = \frac{h'_{10}x + h'_{11}y + h'_{12}}{h'_{20}x + h'_{21}y + 1}$$

1. Adding Translation ($\tilde{x}' = \tilde{x} + t$):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

2. Adding Rotation ($\tilde{x}' = R.\tilde{x} + t$):

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

3.1. Adding Scaling ($\tilde{x}' = S.R.\tilde{x} + t$):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x\cos\theta & -S_x\sin\theta \\ S_y\sin\theta & S_y\cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

3.2. Adding Shear ($\tilde{x}' = S.R.\tilde{x} + t$):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & k_x \\ k_y & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta + k_x\sin\theta & -\sin\theta + k_x\cos\theta \\ k_y\cos\theta + \sin\theta & -k_y\sin\theta + \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

4. All ($\tilde{x}' = S.S.R.\tilde{x} + t$):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} 1 & k_x \\ k_y & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x(\cos\theta + k_x\sin\theta) & S_x(-\sin\theta + k_x\cos\theta) \\ S_y(k_y\cos\theta + \sin\theta) & S_y(-k_y\sin\theta + \cos\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Matrix form(2x3):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x(\cos\theta + k_x\sin\theta) & S_x(-\sin\theta + k_x\cos\theta) & t_x \\ S_y(k_y\cos\theta + \sin\theta) & S_y(-k_y\sin\theta + \cos\theta & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\tilde{x}$ (x, y , 1) : Homogeneous coordinates.

$\tilde{x}'$ (x', y') : Transformed coordinates.

t (t_x, t_y) : Translation.

R : Rotation matrix.

$\tilde{x}' = A\tilde{x}$

S_x: Scale factor to the x-axis.

S_y: Scale factor to the y-axis.

A : Affine Matrix.

θ : Rotation Angle.

k_x: Shear factor parallel to the x-axis.

k_y: Shear factor parallel to the y-axis.

H : Homogeneous matrix.

$$\begin{bmatrix} 1 & 1 \\ ky/kx & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \gamma & 1 \end{bmatrix}$$

?? Intrinsic skew factor??

Scale factor =?

Aspect Ratio= α/β = k_u/k_v

2D (Planar) Motions (Transformation) (4/6) jj

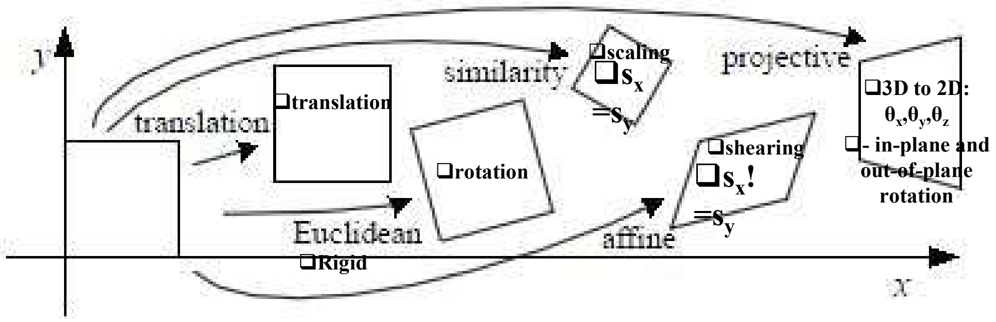


Figure 2: *Basic set of 2D planar transformations*

$\tilde{x}$ (x, y , 1) : Homogeneous coordinates.

$\tilde{x}'$ (x', y') : Transformed coordinates.

H : Homogeneous matrix.

Rotation:

In-Plane rotation + Out-of-plane rotation: $\theta_x, \theta_y, \theta_z$

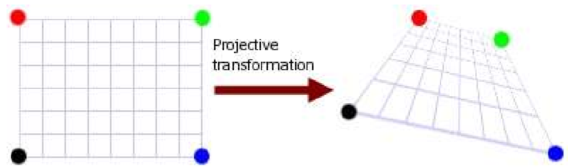
Including 3x3 intrinsic (or projection) parameters + affine transform with in-plane and out-of-plane rotations

5) Projective Transformation (Perspective Transform): 3D project to 2D

Projective transformations preserve straight lines.

$$\tilde{x}' \approx \tilde{H}\tilde{x}$$

$$\begin{bmatrix} x_1 \\ y_1 \\ w_1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \tilde{H} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix}$$



$$x' = \frac{x_1}{w_1} = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} = \frac{h'_{00}x + h'_{01}y + h'_{02}}{h'_{20}x + h'_{21}y + 1}$$

$$y' = \frac{y_1}{w_1} = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}} = \frac{h'_{10}x + h'_{11}y + h'_{12}}{h'_{20}x + h'_{21}y + 1}$$

where $h'_{ij} = \dfrac{h_{ij}}{h_{22}}$

Non-Linear combination

2D (Planar) Motions (Transformation) (5/6-1/11)

6) 8-Parameter planar transformation Vs. perspective projection transformation

Motion flow

$$u(x,y) = a_0 + a_1 x + a_2 y + p_0 x^2 + p_1 xy$$
$$v(x,y) = a_3 + a_4 x + a_5 y + p_0 xy + p_1 y^2$$

Motion flow

$$u(x,y) = a_0 + a_1 x + a_2 y \quad (8)$$
$$v(x,y) = a_3 + a_4 x + a_5 y + cx^2 \quad (9)$$

$$\text{divergence} = a_1 + a_5 = (u_x + v_y), \quad (3)$$
$$\text{curl} = -a_2 + a_4 = -(u_y - v_x), \quad (4)$$
$$\text{deformation} = a_1 - a_5 = (u_x - v_y) \quad (5)$$

Yaw = p_0,

Pitch = p_1 .

a_i: Constants.

$u(x) = [u(x,y), v(x,y)]^T$: Horizontal and vertical components of the flow at the image point $x = (x,y)$.

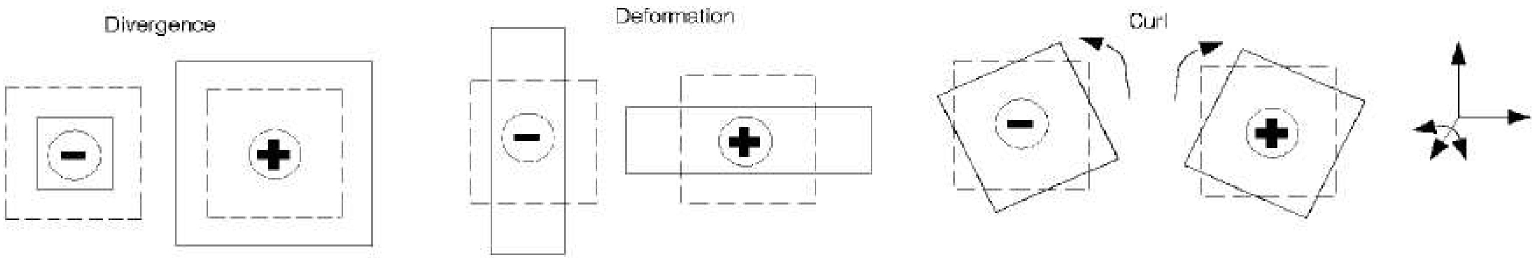


Figure 3. The figure illustrates the motion captured by the various parameters used to represent the motion of the regions. The solid lines indicate the deformed image region and the "−" and "+" indicate the sign of the quantity.

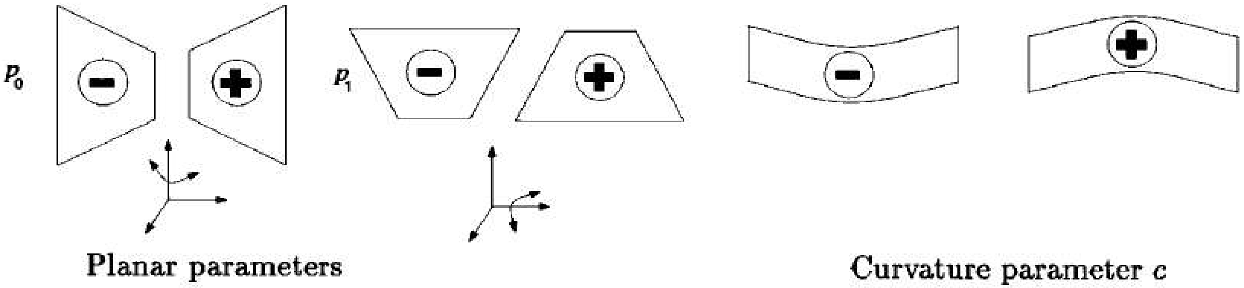


Figure 4. Additional parameters for planar motion and curvature.

M. Black and Y. Yaccob, "Recognizing Facial Expressions in Image Sequences Using Local Parameterized Models of Image Motion", IJCV1997.

2D (Planar) Motions (Transformation) (5/6-2/11)

$$\mathbf{X}(\mathbf{x}) = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy & 0 \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 & x^2 \end{bmatrix} \quad (10)$$

$$\mathbf{A} = [a_0 \;\; a_1 \;\; a_2 \;\; a_3 \;\; a_4 \;\; a_5 \;\; 0 \;\; 0 \;\; 0]^T \quad (11)$$

$$\mathbf{P} = [a_0 \;\; a_1 \;\; a_2 \;\; a_3 \;\; a_4 \;\; a_5 \;\; p_0 \;\; p_1 \;\; 0]^T \quad (12)$$

$$\mathbf{C} = [a_0 \;\; a_1 \;\; a_2 \;\; a_3 \;\; a_4 \;\; a_5 \;\; 0 \;\; 0 \;\; c]^T \quad (13)$$

such that $\mathbf{u}(\mathbf{x};\mathbf{A}) = \mathbf{X}(\mathbf{x})\mathbf{A}$, $\mathbf{u}(\mathbf{x};\mathbf{P}) = \mathbf{X}(\mathbf{x})\mathbf{P}$, and $\mathbf{u}(\mathbf{x};\mathbf{C}) = \mathbf{X}(\mathbf{x})\mathbf{C}$ represent, respectively, the affine, planar, and affine + curvature flow models described above.

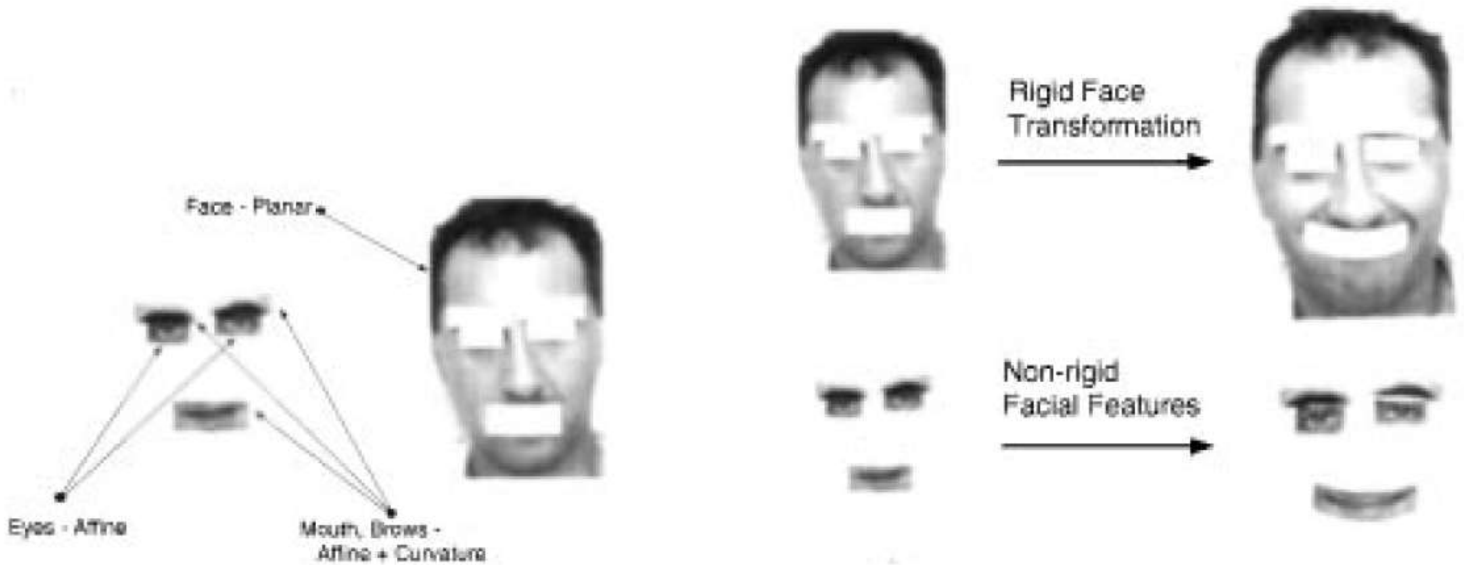


Figure 1. Illustration showing the parametric motion models employed and an example of a face undergoing a looming motion while smiling.

(5/6-3/11)

Table 1. The mid-level predicates derived from deformation and motion parameter estimates.

Parameter	Threshold	Derived predicates (mouth)
a_0	>0.25	Rightward
	<-0.25	Leftward
a_3	<-0.1	Upward
	>0.1	Downward
Div	>0.02	Expansion
	<-0.02	Contraction
Def	>0.005	Horizontal deformation
	<-0.005	Vertical deformation
Curl	>0.005	Clockwise rotation
	<-0.005	Counter clockwise rotation
c	<-0.0001	Curving upward ('U' like)
	>0.0001	Curving downward

Table 2. The mid-level predicates derived from deformation and motion parameter estimates as applied to head motion.

Parameter	Threshold	Derived predicates (head)
a_0	>0.5	Rightward
	<-0.5	Leftward
a_3	<-0.5	Upward
	>0.5	Downward
Div	>0.01	Expansion
	<-0.01	Contraction
Def	>0.01	Horizontal deformation
	<-0.01	Vertical deformation
Curl	>0.005	Clockwise rotation
	<-0.005	Counter clockwise rotation
p_0	<-0.00005	Rotate right about neck
	>0.00005	Rotate left about neck
p_1	<-0.00005	Rotate forward
	>0.00005	Rotate backward

Table 3. The rules for classifying facial expressions (B = beginning, E = ending).

Expr.	B/E	Satisfactory actions
Anger	B	Inward lowering of brows and mouth contraction
Anger	E	Outward raising of brows and mouth expansion
Disgust	B	Mouth horizontal expansion and lowering of brows
Disgust	E	Mouth contraction and raising of brows
Happiness	B	Upward curving of mouth and expansion or horizontal deformation
Happiness	E	Downward curving of mouth and contraction or horizontal deformation
Surprise	B	Raising brows and vertical expansion of mouth
Surprise	E	Lowering brows and vertical contraction of mouth
Sadness	B	Downward curving of mouth and upward-inward motion in inner parts of brows
Sadness	E	Upward curving of mouth and downward-outward motion in inner parts of brows
Fear	B	Expansion of mouth and raising-inwards inner parts of brows
Fear	E	Contraction of mouth and lowering inner parts of brows

2D (Planar) Motions (Transformation) (5/6-4/11)

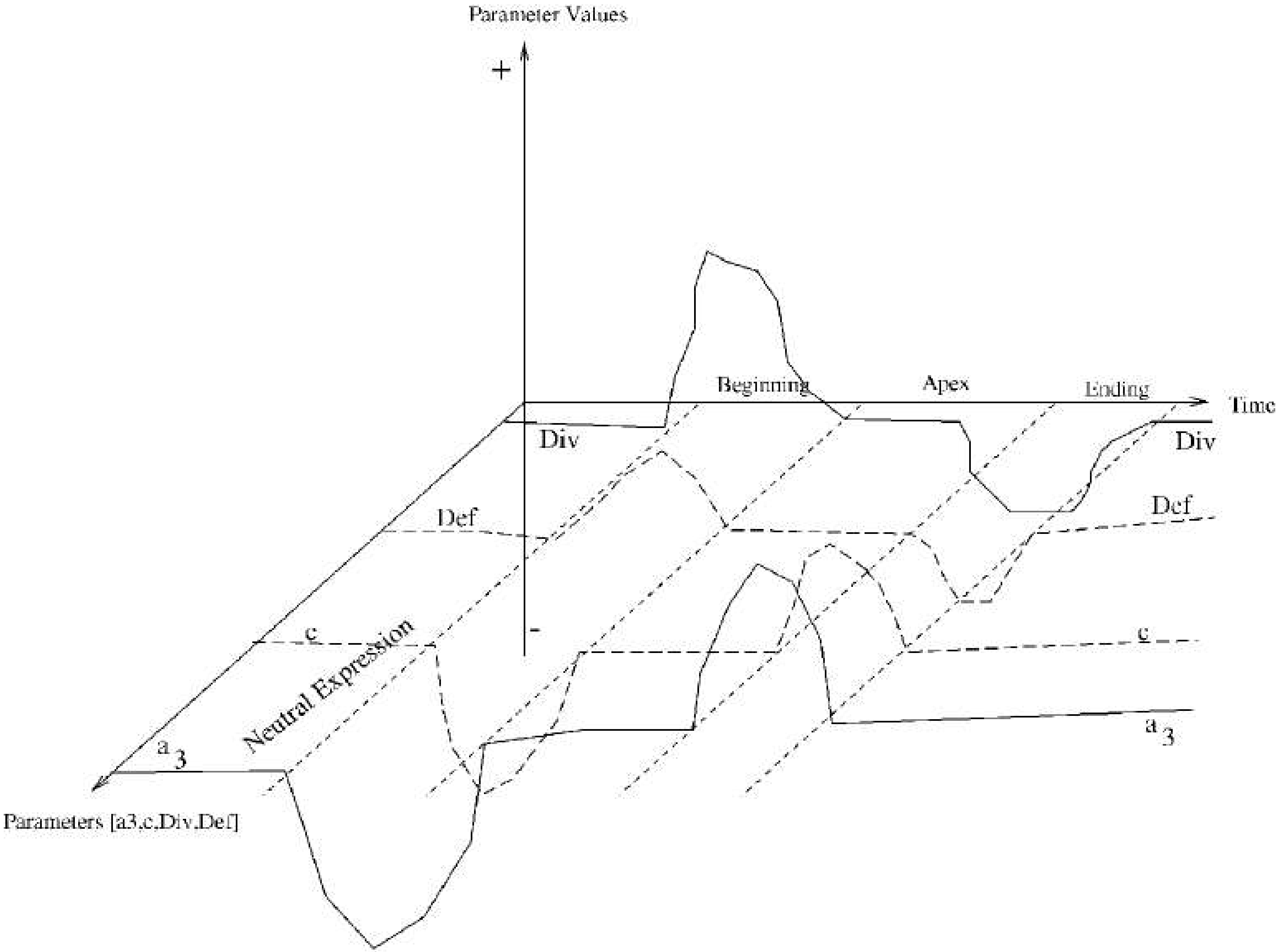


Figure 5. The temporal model of the "smile" expression.

(5/6-5/11)

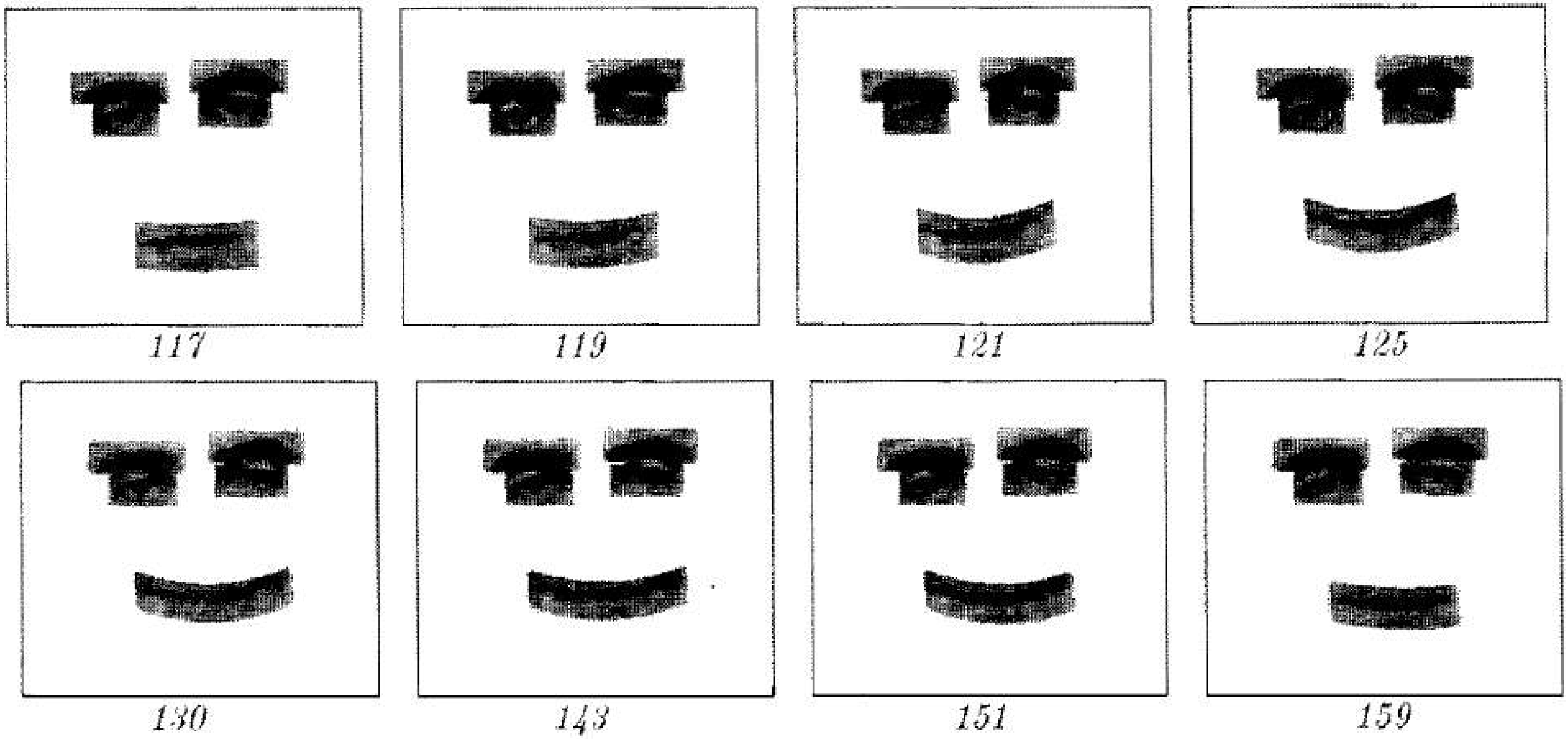


Figure 6. Smile experiment: facial expression tracking.

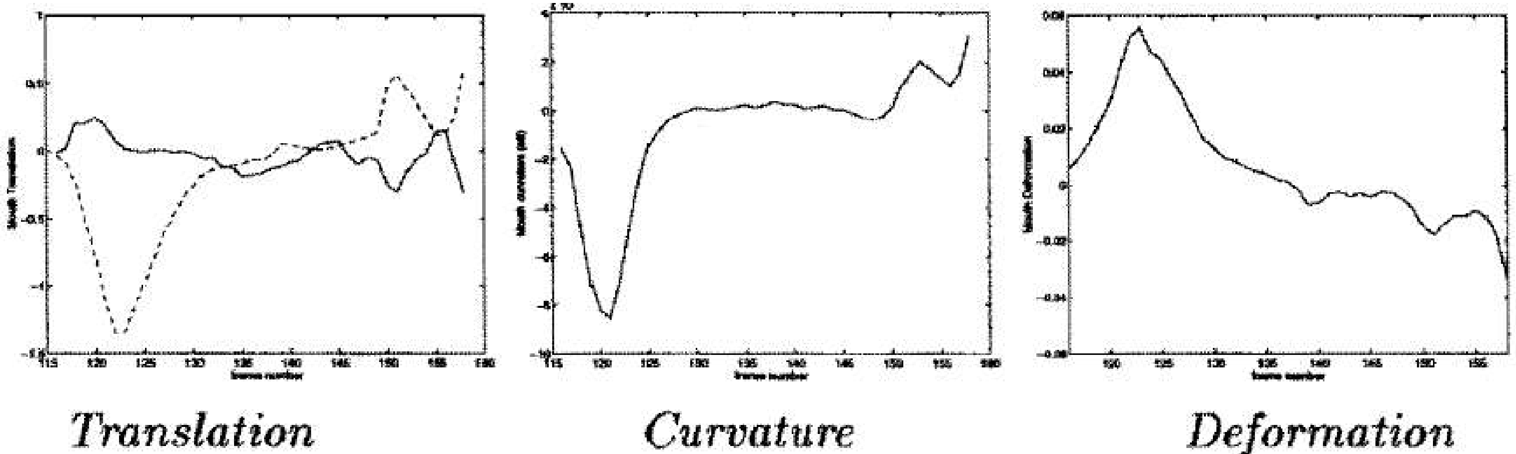


Figure 7. The smile mouth parameters. For translation, solid and dashed lines indicate horizontal and vertical motion respectively.

<u>2D (Planar) Motions (Transformation)</u> (5/6-6/11)

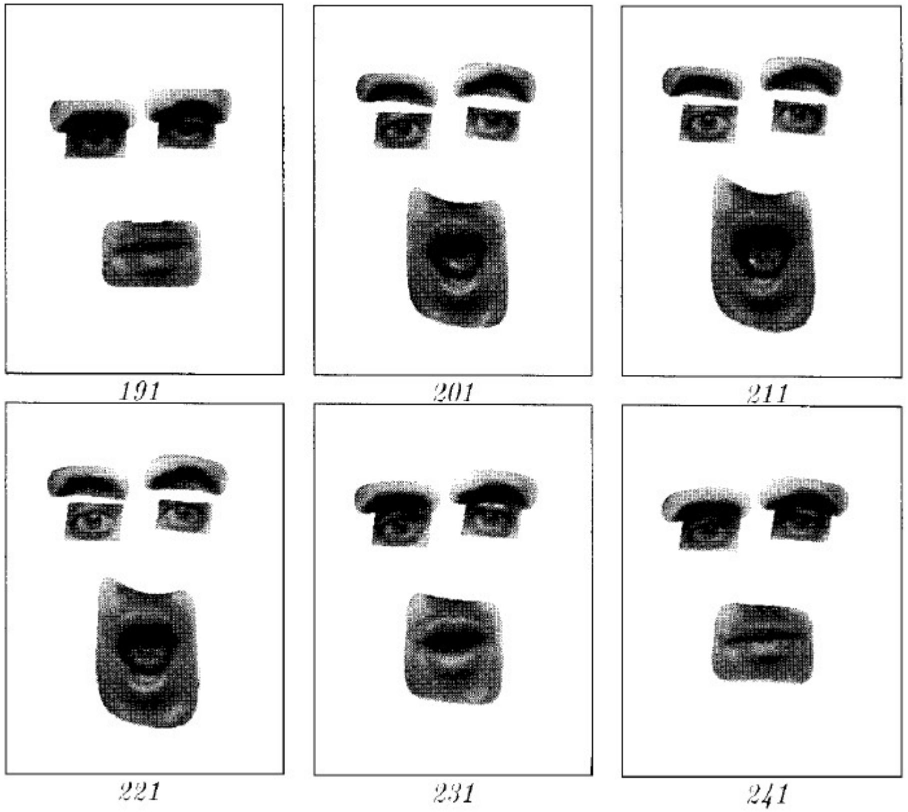


Figure 10. Surprise experiment: facial expression tracking. Features every 10 frames.

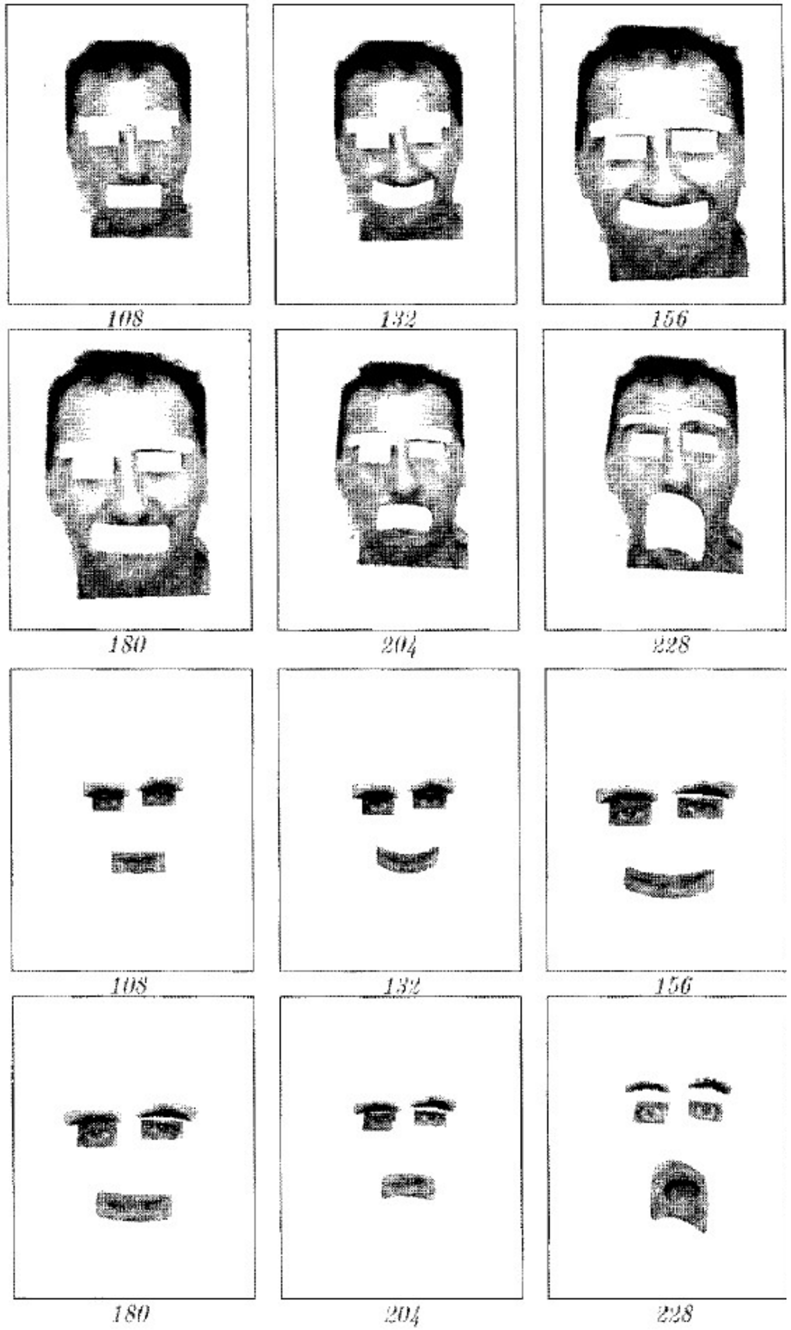


Figure 14. Looming experiment. Facial expression tracking with rigid head motion (every 24 frames).

2D (Planar) Motions (Transformation) (5/6-7/11)

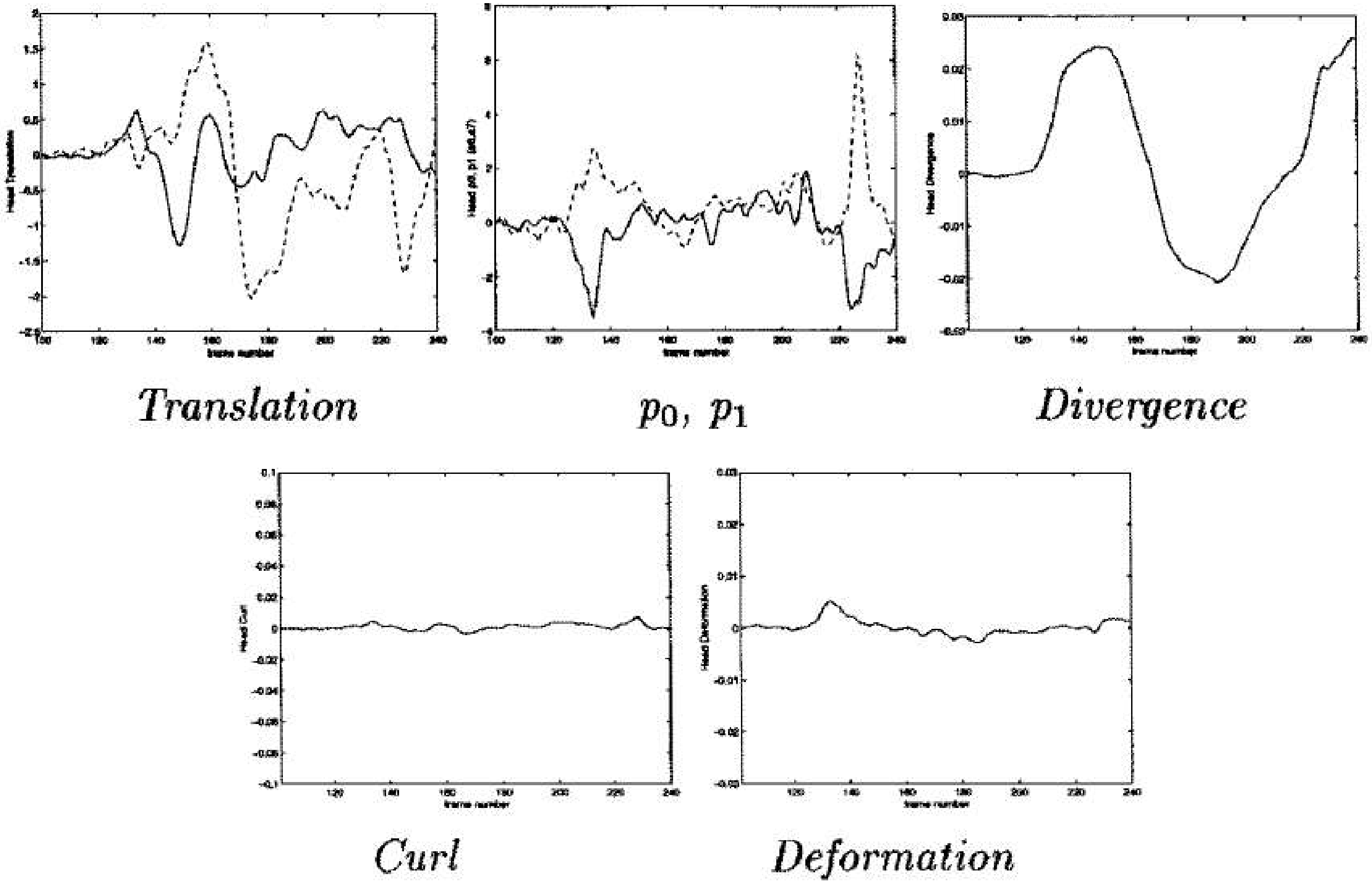


Figure 15. The looming face motion parameters. Translation: solid = horizontal, dashed = vertical. Quadratic terms: solid = p_0, dashed = p_1.

2D (Planar) Motions (Transformation) (5/6-8/11)

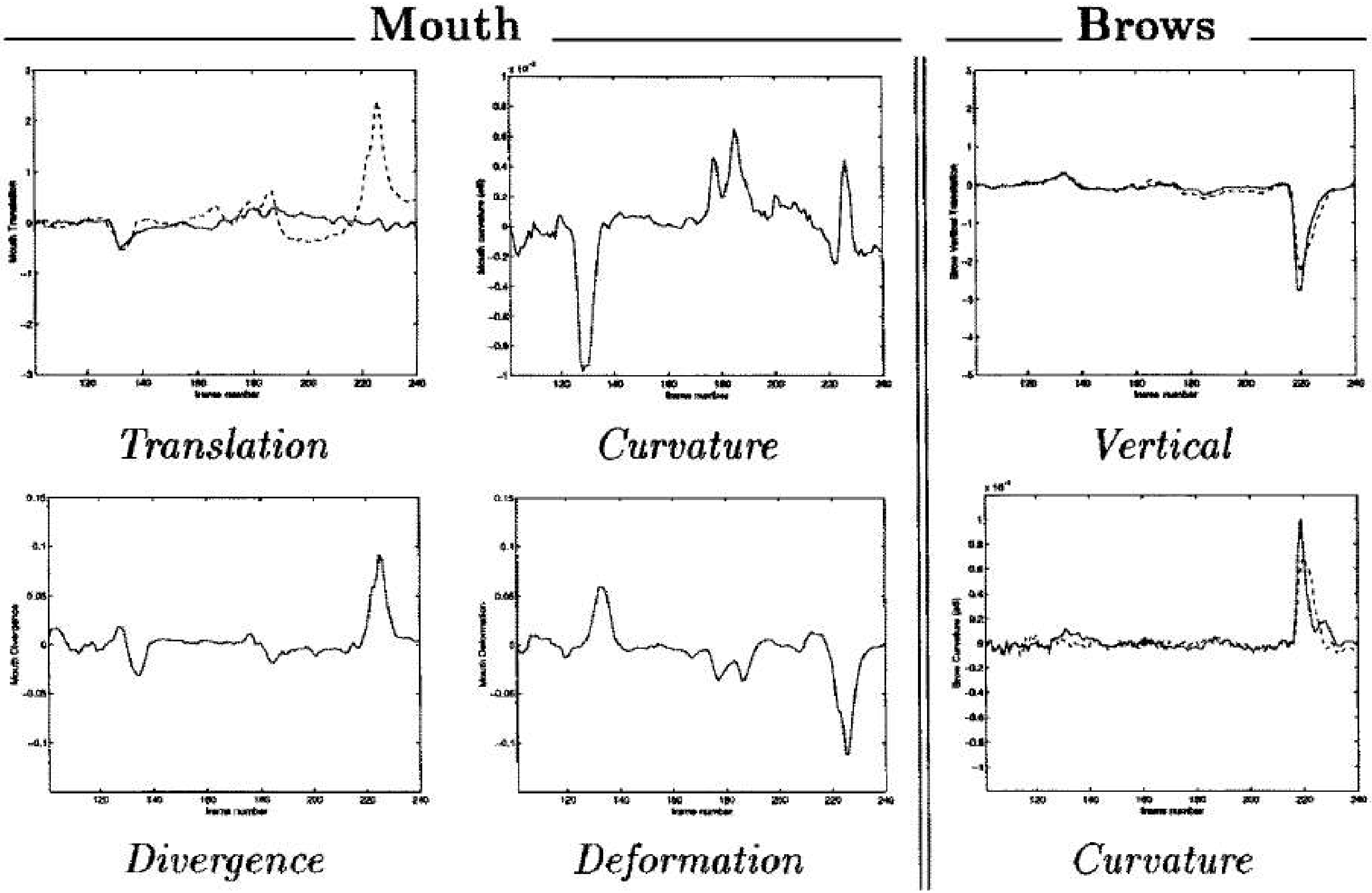


Figure 16. The looming sequence. Mouth translation: solid and dashed lines indicate horizontal and vertical motion respectively. For the brows, the solid and dashed lines indicate left and right brows respectively.

2D (Planar) Motions (Transformation) (5/6-9/11)

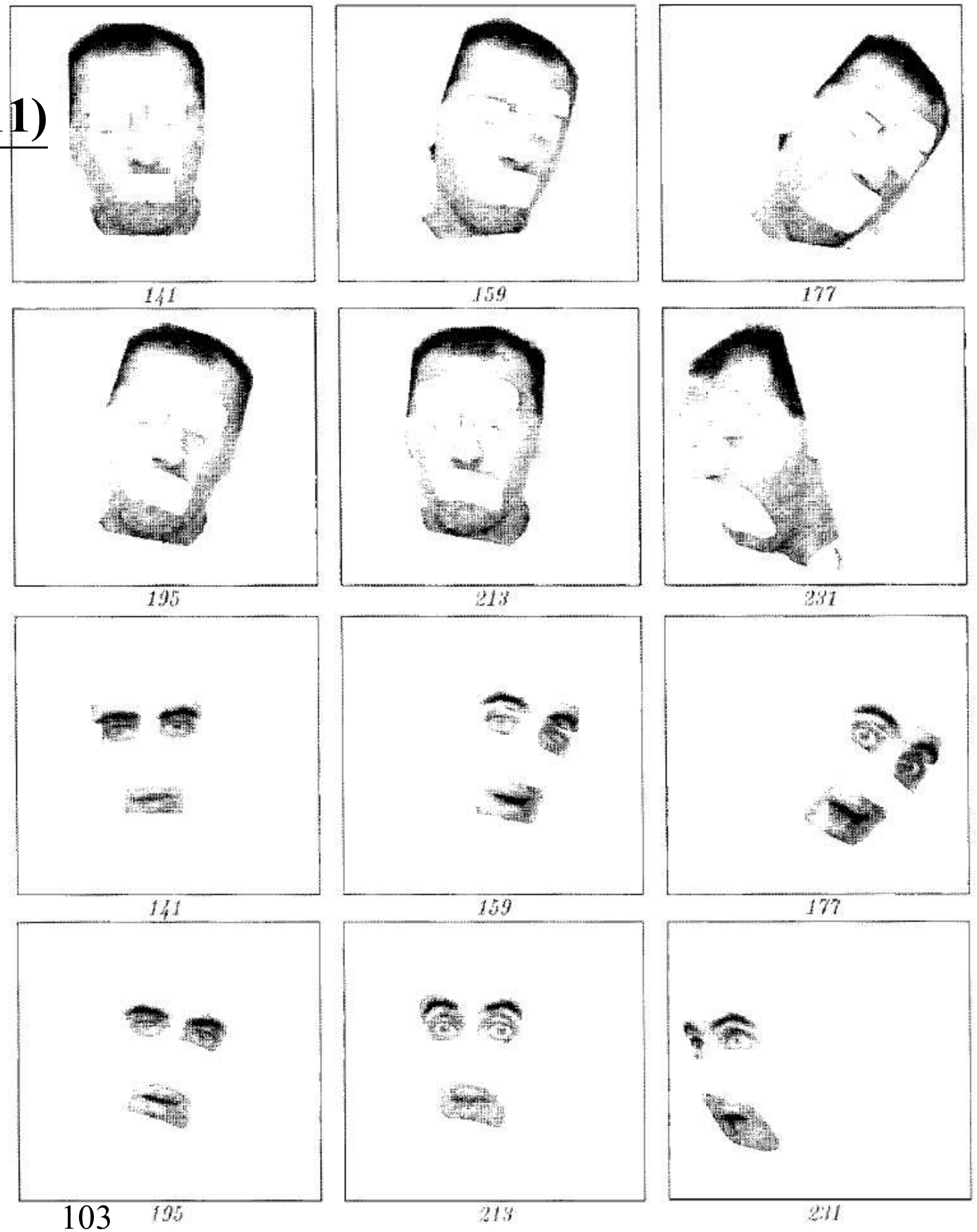
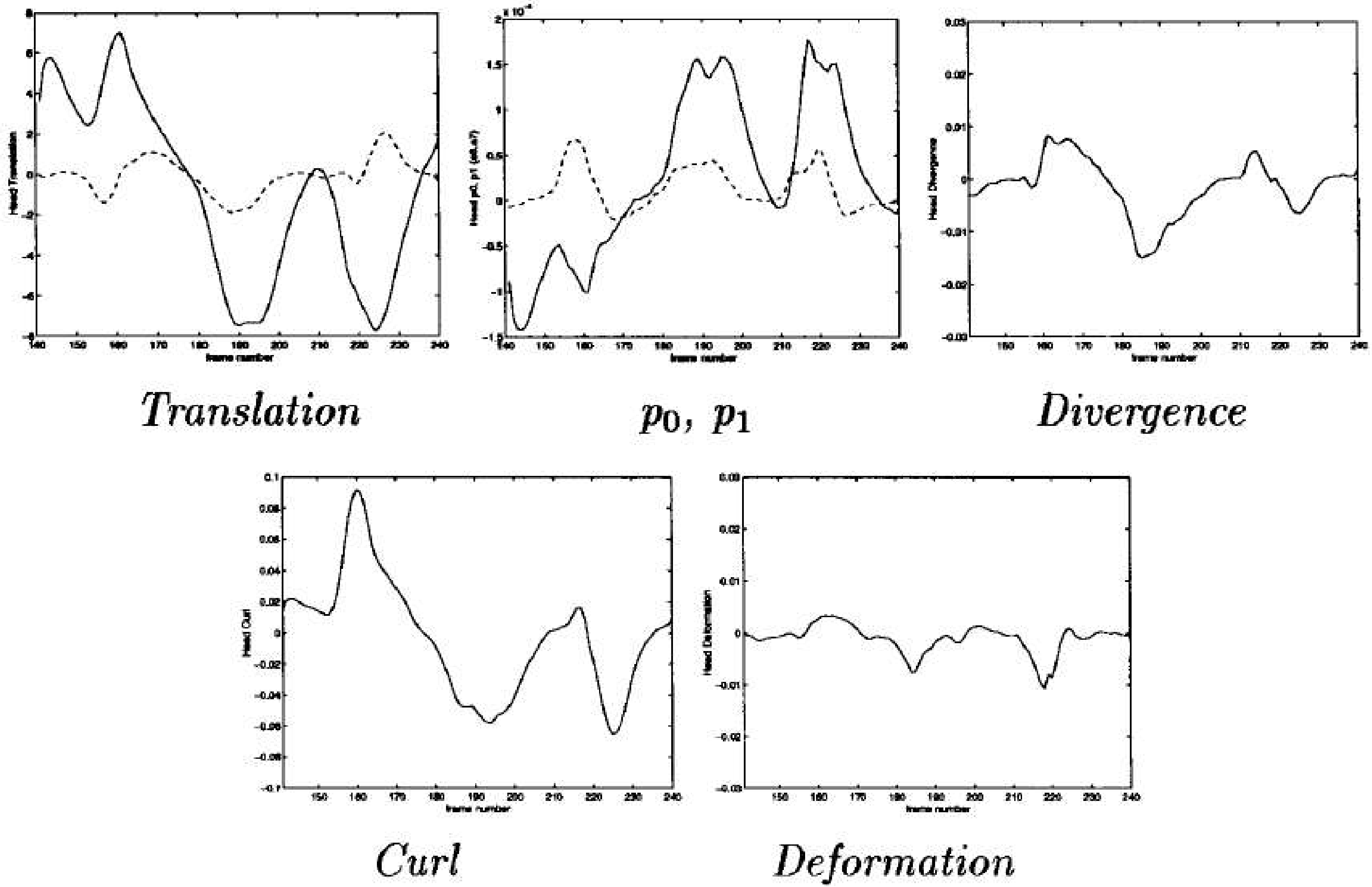


Figure 17. Rotation experiment. Rigid head tracking, every 18th frame.

2D (Planar) Motions (Transformation) (5/6-10/11)



Translation $p_0,\ p_1$ *Divergence*

Curl   *Deformation*

Figure 18. The rotate sequence face motion parameters. Translation: solid = horizontal, dashed = vertical. Quadratic terms: solid = p_0, dashed = p_1.

2D (Planar) Motions (Transformation) (5/6-11/11)

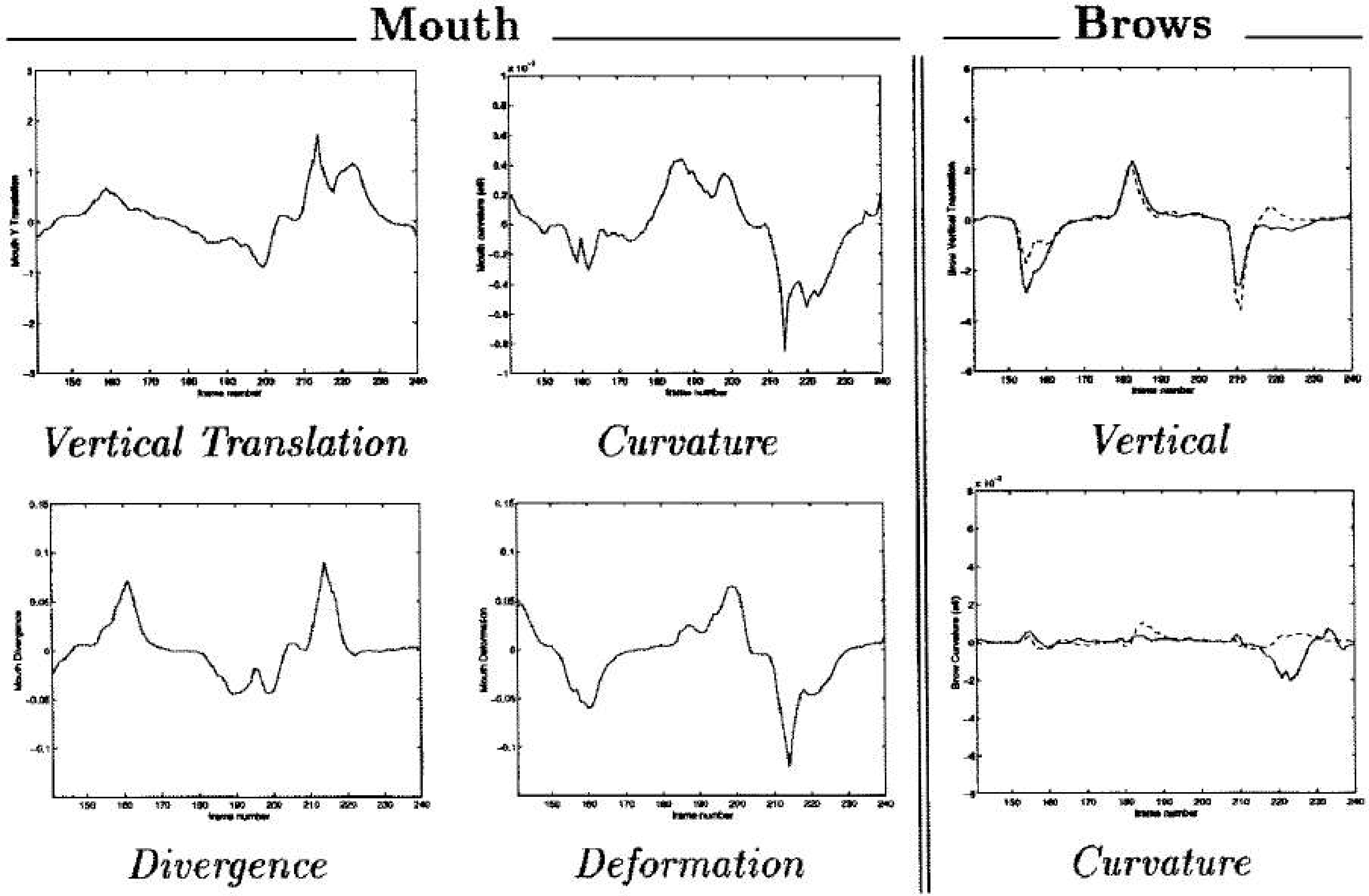


Figure 19. The rotate sequence. For the brows, the solid and dashed lines indicate left and right brows respectively.

2D (Planar) Motions (Transformation) (6/6) jj

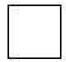
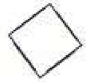
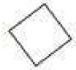
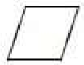
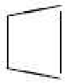
Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\left[\ I \mid t\ \right]_{2\times 3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\left[\ R \mid t\ \right]_{2\times 3}$	3	lengths $+\cdots$	
similarity **S_x=S_y**	$\left[\ sR \mid t\ \right]_{2\times 3}$	4	angles $+\cdots$	
affine **S_x!= S_y**	$\left[\ A\ \right]_{2\times 3}$	6	parallelism $+\cdots$	
projective	$\left[\ \tilde{H}\ \right]_{3\times 3}$	8	straight lines	

Table 1: *Hierarchy of 2D coordinate transformations. The 2×3 matrices are extended with a third $[0^T\ 1]$ row to form a full 3×3 matrix for homogeneous coordinate transformations.*

#D.O.F : Degrees Of Freedom (i.e. Unknown parameters)

1) **Translation :** t_x, t_y
2) **Euclidean :** t_x, t_y, θ
3) **Similarity :** t_x, t_y, θ, s
4) **Affine :** $a_{00}, a_{01}, a_{02}, a_{10}, a_{11}, a_{12}$
5) **Projective :** $h'_{00}, h'_{01}, h'_{02}, h'_{10}, h'_{11}, h'_{12}, h'_{20}, h'_{21}$
6) **8-Parameter planar transformation: a0, a1, a2, a3, a4, a5, p0 and p1**

References

1. G. Bradski and A. Kaebler, *Learning OpenCV, Computer Vision with the OpenCV Library*, O'Reilly, 2008. ISBN-10: 0596516134 or ISBN-13: 978-0596516130.
2. Olivier Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint,* Third printing, 1999, The MIT Press.
3. R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 3nd edition, 2007.
4. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
5. B.K.P. Horn, *Robot Vision*, 1989, The MIT Press.
6. R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010. ISBN-10: 1848829345 or ISBN-13: 978-1848829343.

- **Keywords: CMOS Camera Vs. CCD Camera:**
  - http://www.vvl.co.uk/whycmos/whitepaper.htm
  - http://www.extremetech.com/article2/0,3973,75160,00.asp
  - http://www.kodak.com/US/en/corp/researchDevelopment/technologyFeatures/cmos.shtml
  - http://www.siliconimaging.com/