Digital Image Processing

Geometrical Modification

Ming-Sui (Amy) Lee

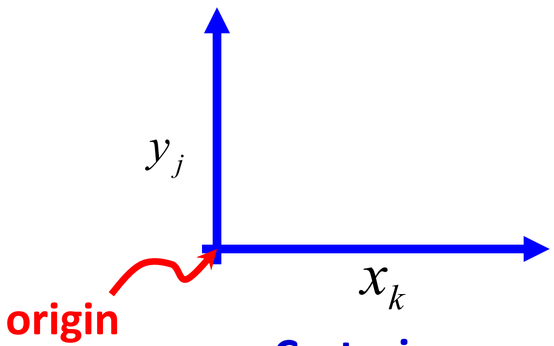
Lecture 03

Geometrical Modification

- Goal
  - Translate, scale, rotate, reflect or nonlinear warp an image
- Applications
  - Zoom-in/zoom-out
  - Image registration
  - Image mosaicking
    - https://www.youtube.com/watch?v=wzEfkQ6zHZA
    - http://www.vision.huji.ac.il/dynmos/
  - 2D to 3D
    - https://manual.reallusion.com/iClone_6/CHT/Pro_6/09_3D_Vision/The_Concepts_of_Stereo_Vision.htm
    - https://www.youtube.com/watch?v=ksfFQcwio2s
    - https://www.youtube.com/watch?v=EHxMwbZzzX8
    - https://www.youtube.com/watch?v=0lLnHe0xbZE

Geometrical Modification

- Coordinates
  - Geometrical transformations
    - Cartesian coordinates
  - Discrete image
    - Cartesian coordinates v.s. Image coordinates



Cartesian Coordinates

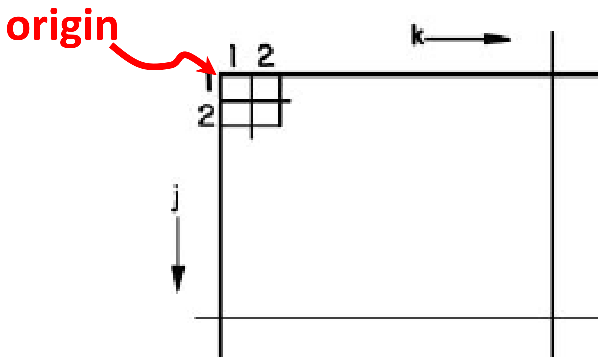


Image Coordinates

Geometrical Modification

- **Linear/Affine coordinates transformation**

$$F(p,q) \quad P\times Q \quad 1\le p\le P \quad 1\le q\le Q$$

$$\text{input} \rightarrow \boxed{\text{Geometrical Modification}} \rightarrow \text{output}$$

$$G(j,k) \quad J\times K \quad 1\le j\le J \quad 1\le k\le K$$

$$\begin{cases} u_q &= q-\dfrac{1}{2} \\ v_p &= P+\dfrac{1}{2}-p \end{cases}$$

$$\begin{cases} x_k &= k-\dfrac{1}{2} \\ y_j &= J+\dfrac{1}{2}-j \end{cases}$$

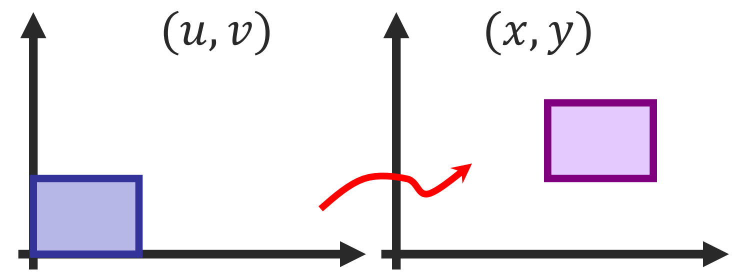
<< Cartesian Coordinates >>

Geometrical Modification

$$F(p,q) \rightarrow G(j,k)$$

- **Translation (Shift)**

$$\begin{cases} x_k &= u_q + t_x \\ y_j &= v_p + t_y \end{cases}$$



substitute $\begin{cases} u_q &= q - \dfrac{1}{2} \\ v_p &= P + \dfrac{1}{2} - p \end{cases}$ **and** $\begin{cases} x_k &= k - \dfrac{1}{2} \\ y_j &= J + \dfrac{1}{2} - j \end{cases}$

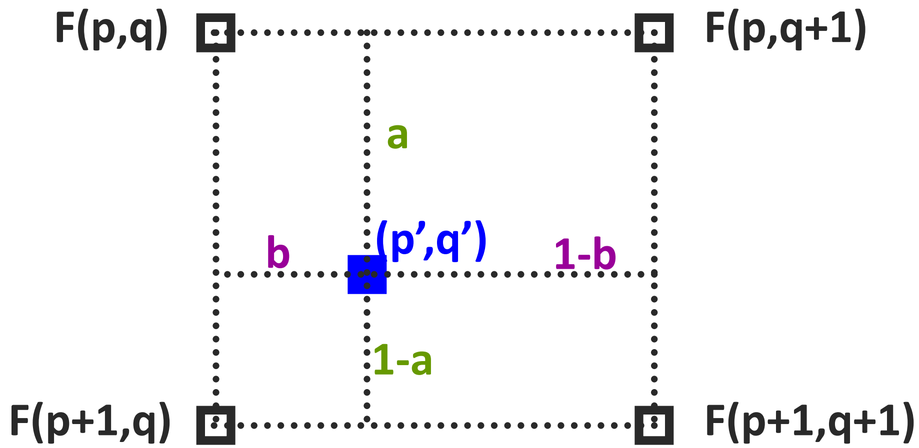
$\Rightarrow \begin{cases} k' &= q + t_x \\ j' &= p - (P - J) - t_y \end{cases}$ $\quad \begin{cases} k &= q' + t_x \\ j &= p' - (P - J) - t_y \end{cases}$

Forward treatment **Backward treatment**

Better

Geometrical Modification

- Translation (Shift)
  - Non-integer pixel positions
  - i.e. How to compute p' and q'?
    - Bilinear interpolation

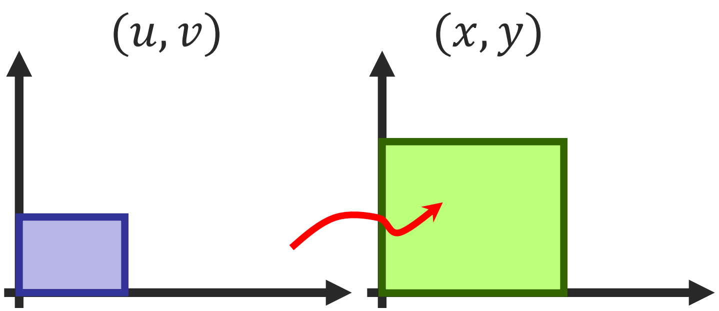


$$
\begin{aligned}
F(p',q') &= (1-a)(1-b)F(p,q) \\
&+ (1-a)bF(p,q+1) \\
&+ a(1-b)F(p+1,q) \\
&+ abF(p+1,q+1)
\end{aligned}
$$

Geometrical Modification

- **Scaling**

$$\begin{cases} x_k & = & s_x u_q \\ y_j & = & s_y v_p \end{cases}$$



where s_x & s_y are scaling parameters, and $s_x \,\&\, s_y > 0$

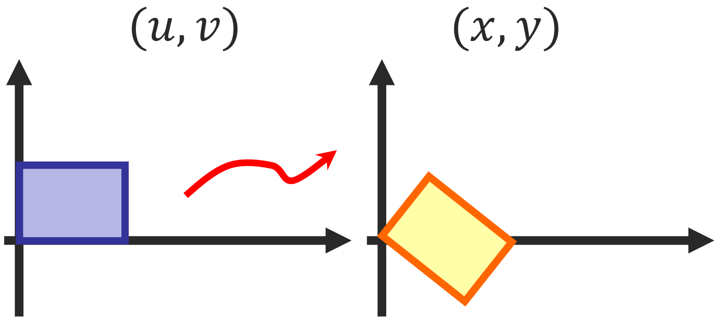
$$\begin{cases} s_x \,\&\, s_y > 1: & \textbf{magnification} \\ s_x \,\&\, s_y < 1: & \textbf{minification} \end{cases}$$

$$\Rightarrow \begin{cases} p' & = & \frac{1}{s_y}(j - J - \frac{1}{2}) + P + \frac{1}{2} \\ q' & = & \frac{1}{s_x}(k - \frac{1}{2}) + \frac{1}{2} \end{cases}$$

Geometrical Modification

- ## Rotation

$$\begin{cases} x_k = u_q \cos\theta - v_p \sin\theta \\ y_j = u_q \sin\theta + v_p \cos\theta \end{cases}$$



Rotate by an angle with respect to the origin of the Cartesian coordinates

What if the reference point is not the origin of the Cartesian coordinate?

Geometrical Modification

- **Generalized Linear Geometrical Transformations**
  - **translation** $$\begin{bmatrix} x_k \\ y_j \end{bmatrix} = \begin{bmatrix} u_q \\ v_p \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$
  - **scaling** $$\begin{bmatrix} x_k \\ y_j \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} u_q \\ v_p \end{bmatrix}$$
  - **rotation** $$\begin{bmatrix} x_k \\ y_j \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} u_q \\ v_p \end{bmatrix}$$

Geometrical Modification

- **Generalized Linear Geometrical Transformations**
  - **Compound operator**

$$\begin{bmatrix} u_q \\ v_p \end{bmatrix} \rightarrow \text{translation} \rightarrow \text{scaling} \rightarrow \text{rotation} \rightarrow \begin{bmatrix} x_k \\ y_j \end{bmatrix}$$

$$\begin{bmatrix} x_k \\ y_j \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \left[\begin{pmatrix} u_q \\ v_p \end{pmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right]$$

How to convert the above affine system to a linear one?

Geometrical Modification

- **Generalized Linear Geometrical Transformations**
  - **Expand the system from 2D to 3D**

$$T(t_x,t_y)=\begin{bmatrix}1 & 0 & t_x\\ 0 & 1 & t_y\\ 0 & 0 & 1\end{bmatrix}\quad S(s_x,s_y)=\begin{bmatrix}s_x & 0 & 0\\ 0 & s_y & 0\\ 0 & 0 & 1\end{bmatrix}\quad R(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$\begin{bmatrix}x_k\\ y_j\\ 1\end{bmatrix}=R(\theta)S(s_x,s_y)T(t_x,t_y)\begin{bmatrix}u_q\\ v_p\\ 1\end{bmatrix}\Rightarrow\begin{bmatrix}u_q\\ v_p\\ 1\end{bmatrix}=T^{-1}(t_x,t_y)S^{-1}(s_x,s_y)R^{-1}(\theta)\begin{bmatrix}x_k\\ y_j\\ 1\end{bmatrix}$$

Geometrical Modification

- Exercise
  - Write down a linear system which represents the following operation:
    - Rotate an image by an angle of θ w.r.t. a pivot point (x_c, y_c)

$$H = \begin{bmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix}$$

Geometrical Modification Part II

Geometrical Modification

- **Non-linear Coordinates Transformation and Spatial Warping**
  - **Non-linear address mapping**
    - **Forward** $\begin{cases} x = X\{u,v\} \\ y = Y\{u,v\} \end{cases}$
    - **Backward (reverse)** $\begin{cases} u = U\{x,y\} \\ v = V\{x,y\} \end{cases}$

$$\begin{cases} u_q = q - \dfrac{1}{2} \\ v_p = P + \dfrac{1}{2} - p \end{cases} \quad \boxed{\text{input}} \Rightarrow \boxed{\text{output}} \quad \begin{cases} x_k = k - \dfrac{1}{2} \\ y_j = J + \dfrac{1}{2} - j \end{cases}$$

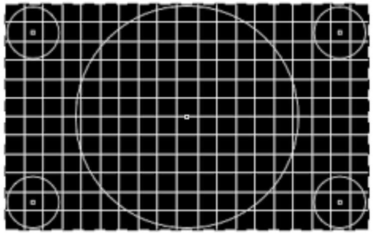
Geometrical Modification

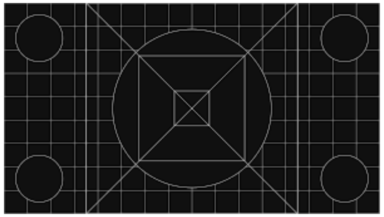
- **Polynomial Warping (2nd-order)**

$$(u, v) \rightarrow (x, y) \quad \begin{cases} u = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 \\ v = b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 xy + b_5 y^2 \end{cases}$$

$$\Rightarrow \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix}$$

Geometrical Modification

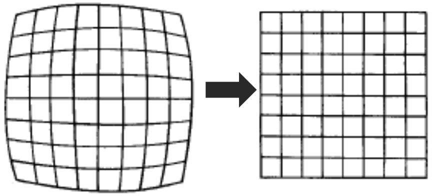




- **Polynomial Warping**

  - Rubber-sheet stretching
  - **Identify spatial distortion**
    - **Calibration → test patterns**
    - **Two steps:**
      - **Based on 'known' input and output pairs (control points), compute the coefficients 'a' and 'b' (either exact or least squares solution),**

        i.e. the spatial warping matrix
      - **Use the spatial warping matrix to compute all the output (input) points from their corresponding input (output) points**

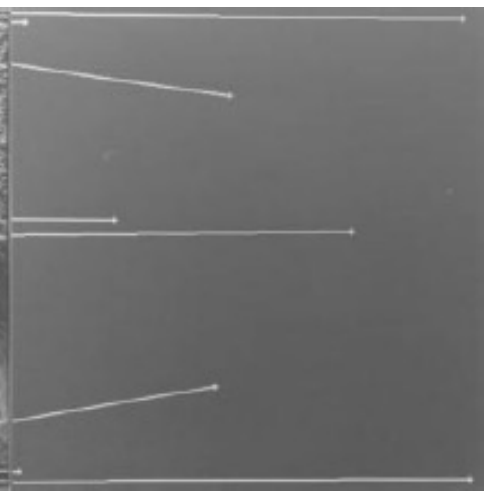


→ proper interpolation is necessary

Geometrical Modification

- **Polynomial Warping**
  - **Example**

input

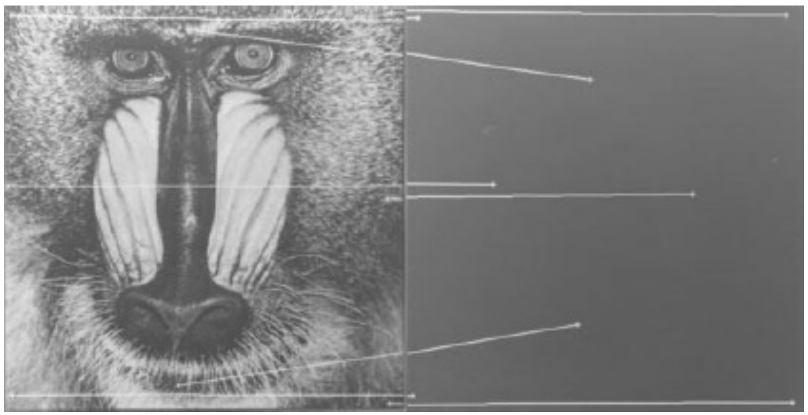


control points

desired output

Geometrical Modification

- Example

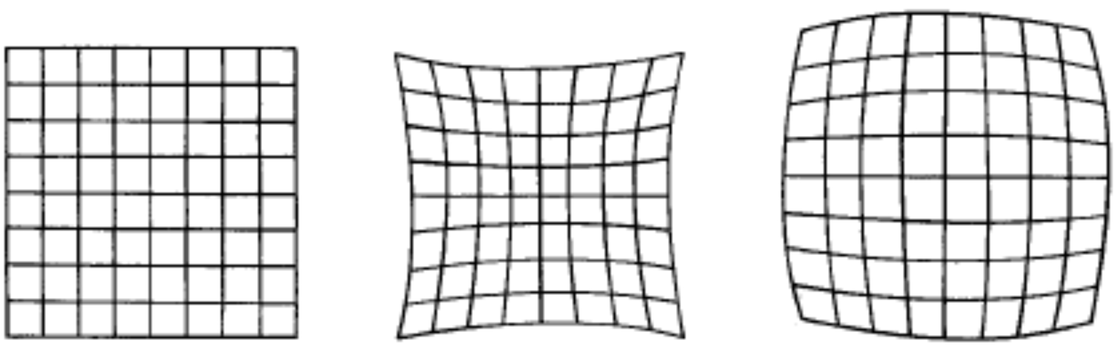


$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix}$$

$$\begin{bmatrix} u_1 & u_2 & \cdots & u_k \\ v_1 & v_2 & \cdots & v_k \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_k \\ y_1 & y_2 & \cdots & y_k \\ x_1^2 & x_2^2 & \cdots & x_k^2 \\ x_1 y_1 & x_2 y_2 & \cdots & x_k y_k \\ y_1^2 & y_2^2 & \cdots & y_k^2 \end{bmatrix}$$

Geometrical Modification

- Polynomial Warping
  - Useful to compensate the spatial distortion caused by the limitation of a physical imaging system



Original | Pincusion distortion | Barrel distortion

Geometrical Modification

- **Examples**

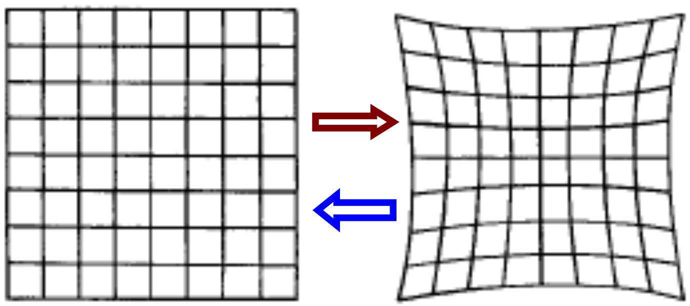
Original

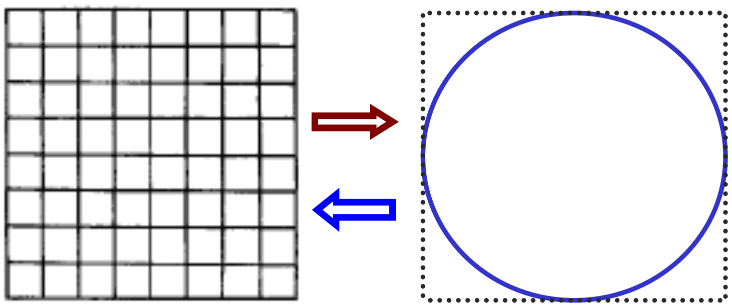
Barrel distortion

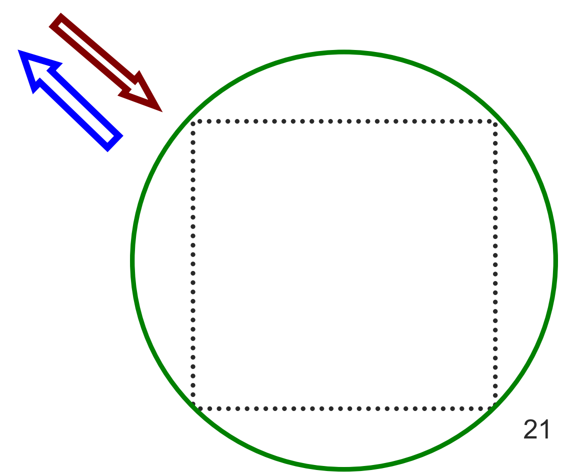
Pincusion distortion

Geometrical Modification

- **Example**



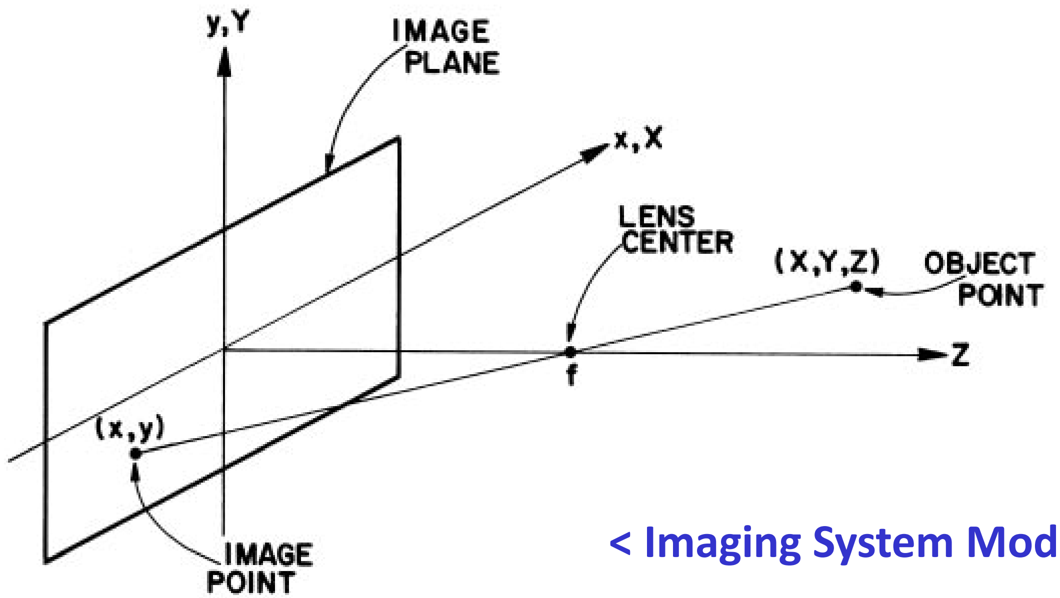




Can we warp back to the original image?

Geometrical Modification

- Perspective Transformation
  - Imaging in the 3D space
    - Fundamentals of computer graphics



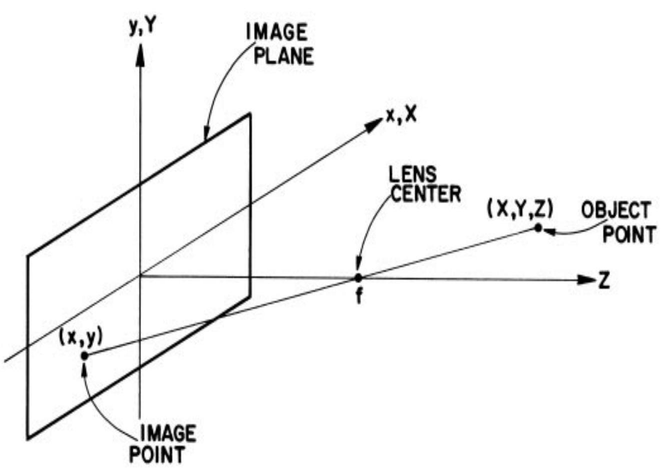
< Imaging System Model >

Geometrical Modification

- **Perspective Transformation**
  - **Cartesian to image coordinates**
    - **Similar triangle property**

$$\frac{X}{-x} = \frac{Z-f}{f} \quad \Rightarrow \quad x = \frac{fX}{f-Z};$$

$$y = \frac{fY}{f-Z}$$



→ Many-to-one mapping

Geometrical Modification

- Perspective Transformation
  - Image to Cartesian coordinates
    - Need another degree of freedom

$$X = \frac{fx_i}{f + z_i}; \quad Y = \frac{fy_i}{f + z_i}; \quad Z = \frac{fz_i}{f + z_i}$$

z_i is a free variable

    - Given Z, we may compute z_i and then X & Y via

$$X = \frac{x_i}{f}(f - Z) \qquad Y = \frac{y_i}{f}(f - Z)$$

Geometrical Modification

- **Perspective Transformation**

P **is a perspective transformation matrix,**

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix}$$

$$\tilde{v} = s\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous vector 3D object
s: scaling factor

$$\tilde{w} = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

homogeneous image position vector

$$\tilde{w} = P\tilde{v} = \begin{bmatrix} sX \\ sY \\ sZ \\ s - sz/f \end{bmatrix} \qquad \Rightarrow s = \frac{f}{f - z}$$

Geometrical Modification

- Camera Imaging Model
  - Camera is supported by a gimbal (X_G, Y_G, Z_G)
  - Gimbal can do 3D movements
    - panning (θ) /tilting (ϕ)
  - Offset between the gimbal support and the image plane center is (X_0, Y_0, Z_0)
  - The complete camera imaging model can be derived by sequentially operating on the homogeneous vector

$$\widetilde{w} = PT_c R T_G \widetilde{v}$$