Image Geometrical Manipulation and Warping

C.-C. Jay Kuo

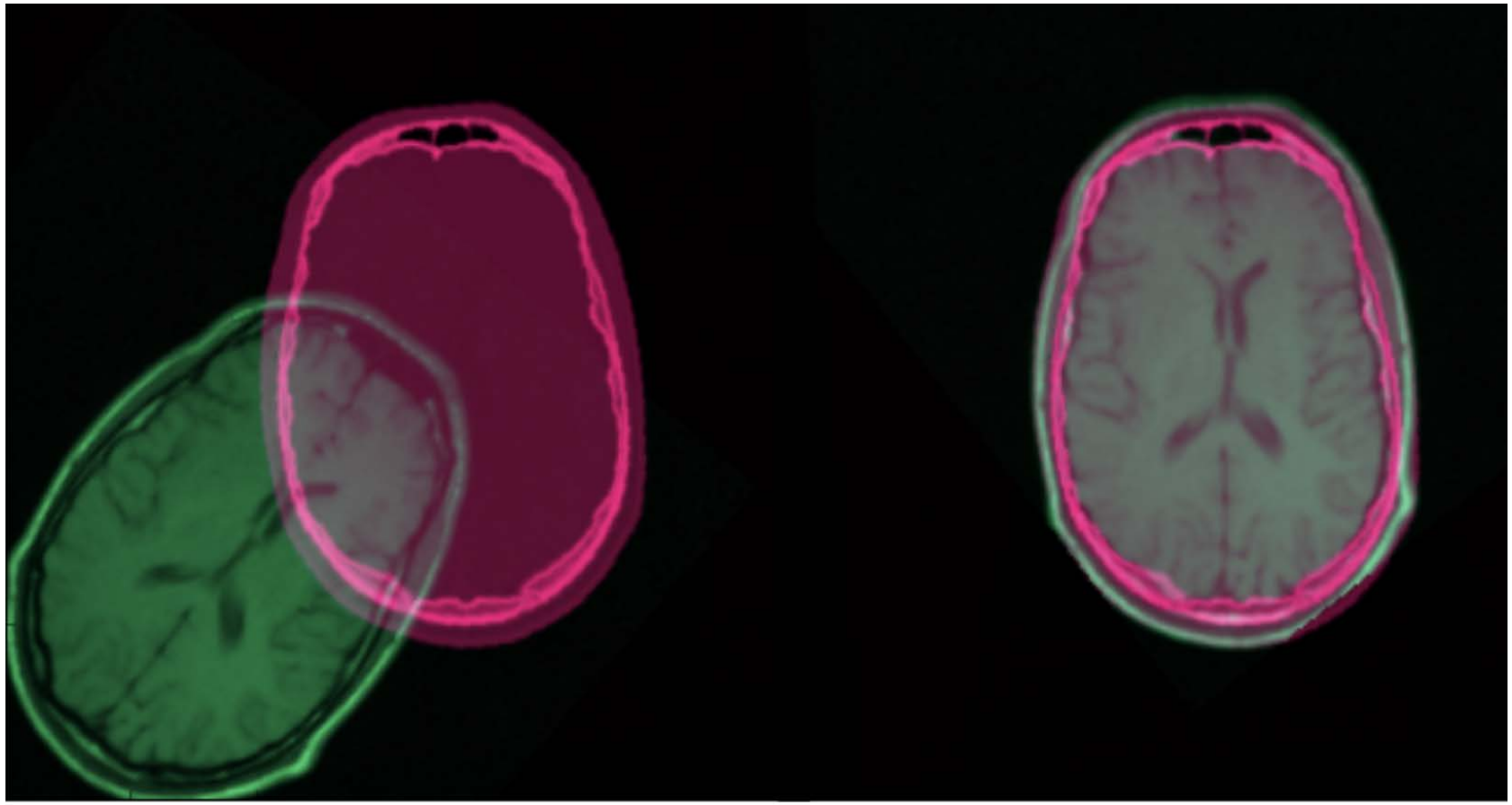
University of Southern California

Introduction

- Geometrical Manipulation
  - Translation
  - Scaling (zoom-in and zoom-out)
  - Rotation
  - Affine transformation
- Advanced Manipulation
  - Image warping
  - 3D object warping
  - Computer graphic rendering (from 3D world coordinates to 2D image coordinates)

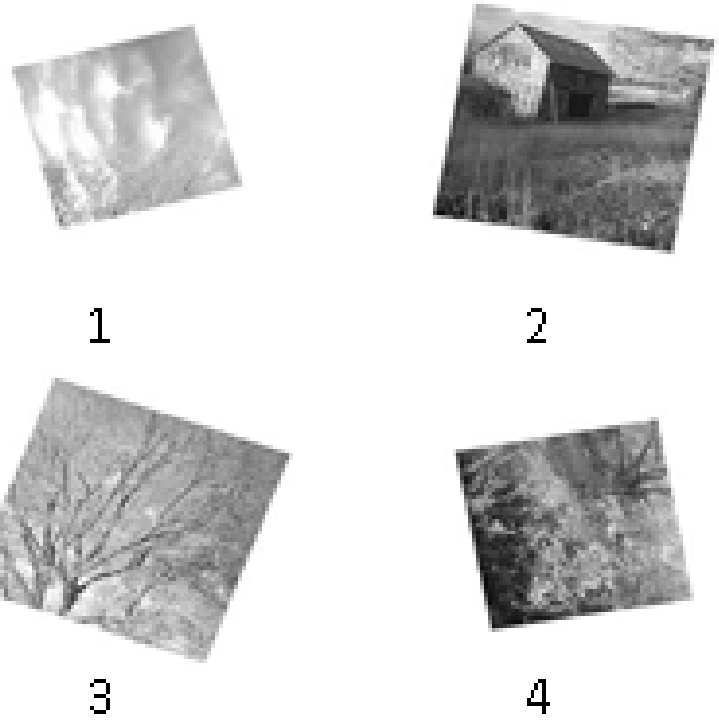
Applications

- Image registration



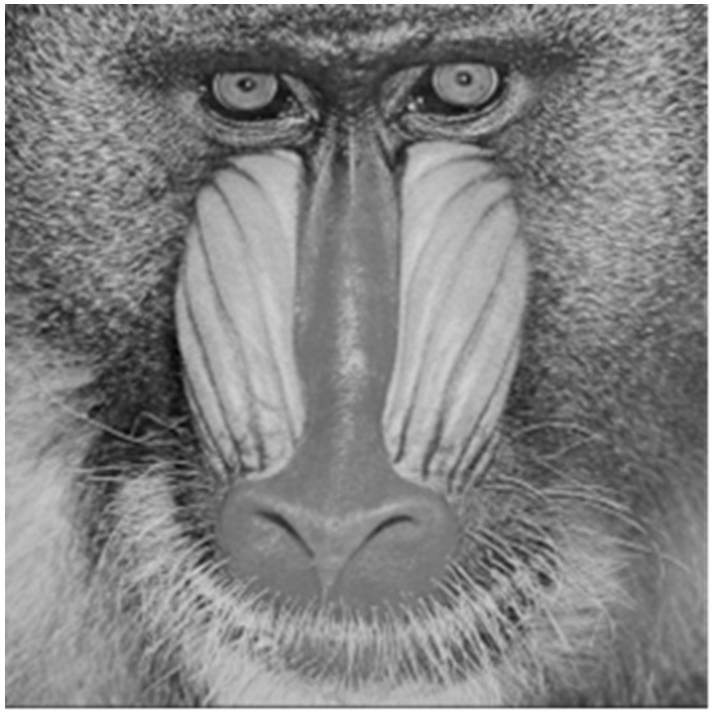
Applications

- Image registration



Applications

- Image warping



Applications

- Image warping

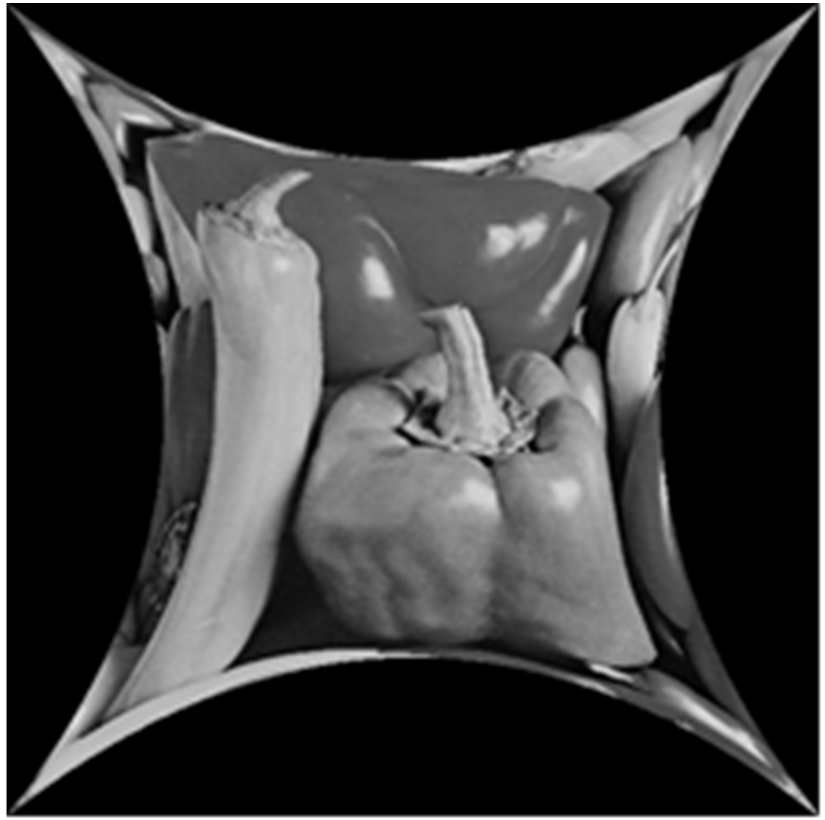
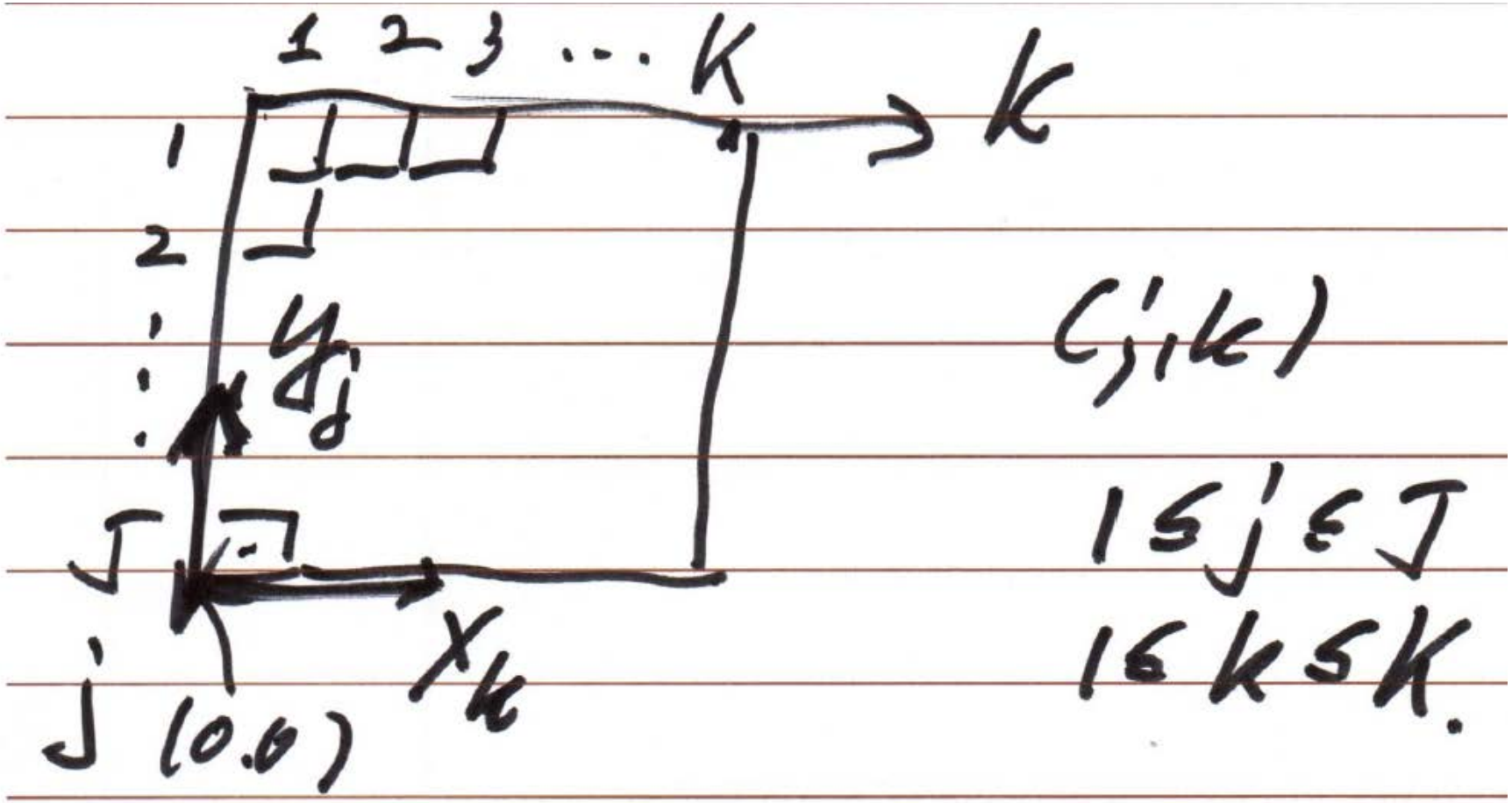
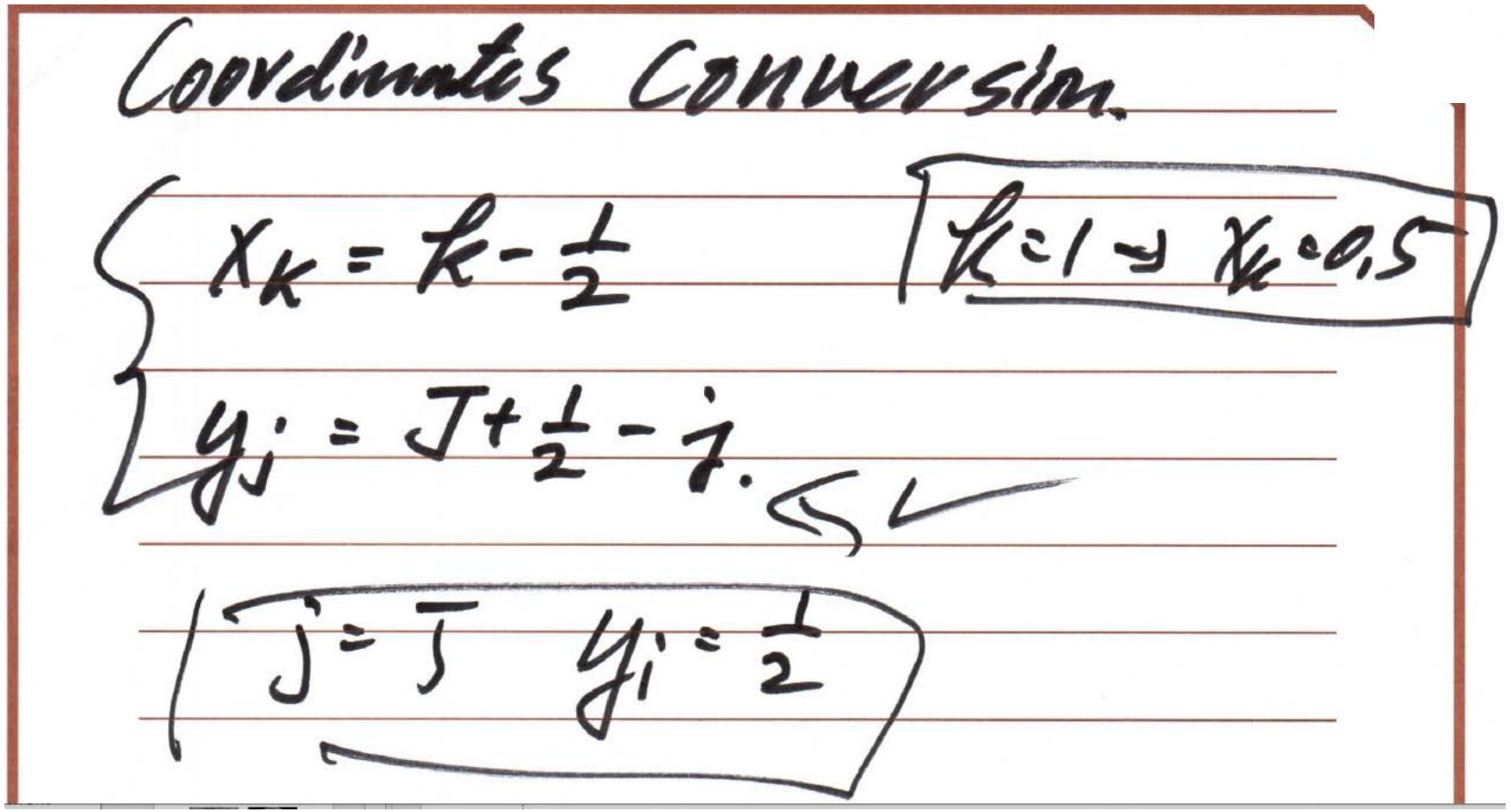


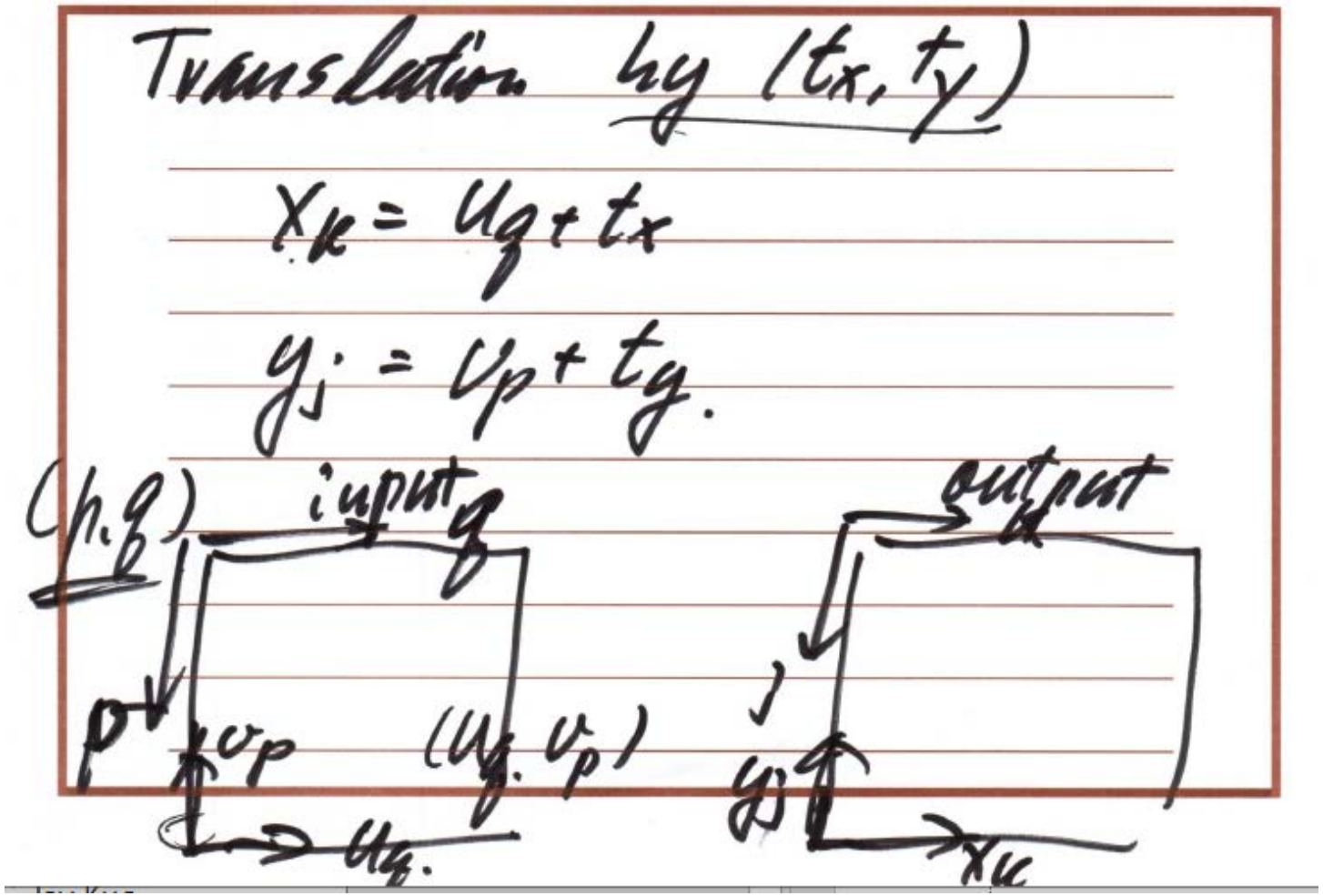
Image Coordinates versus Cartesian Coordinates



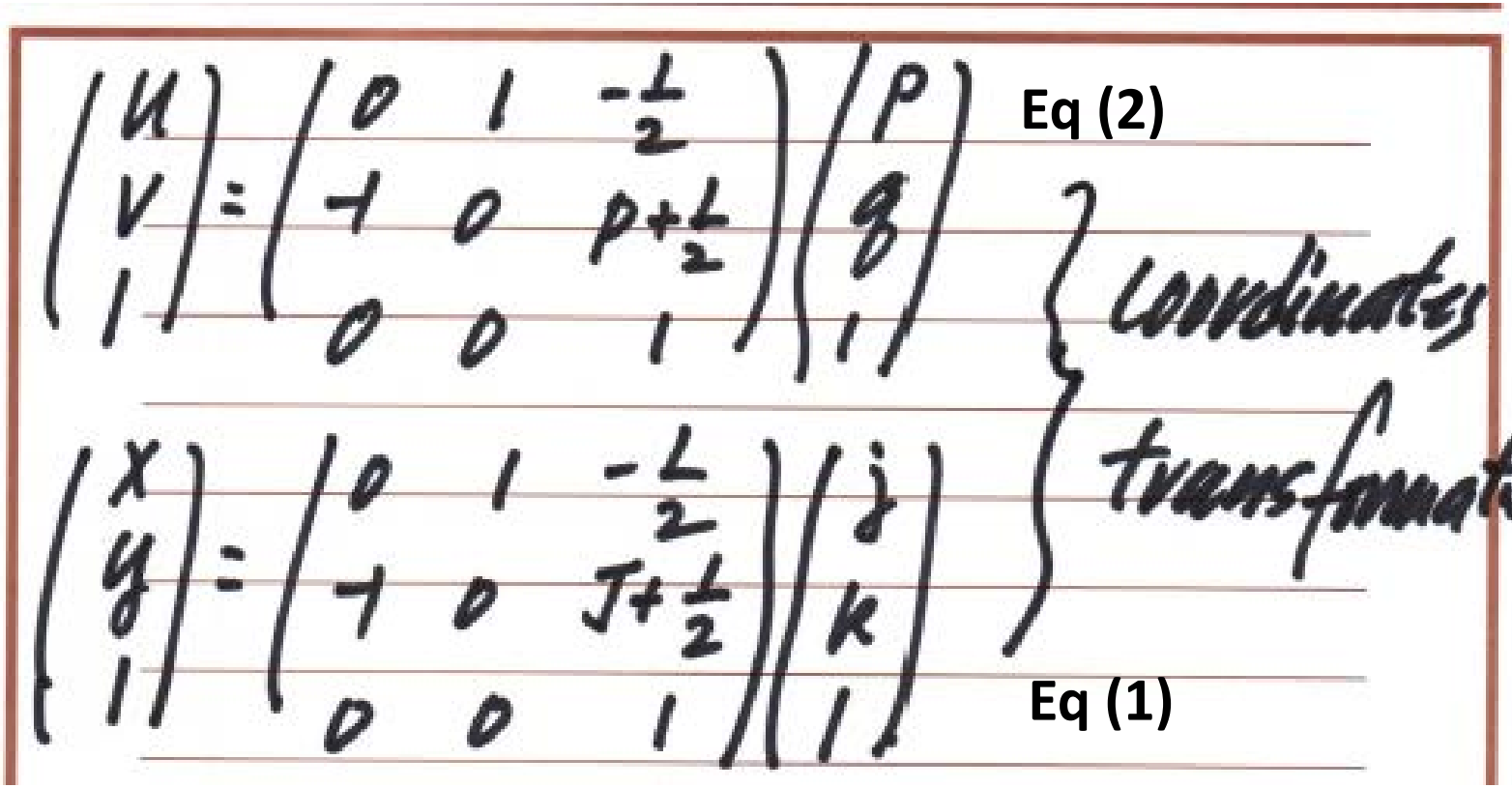
Transformation between Image and Cartesian Coordinates



Translation

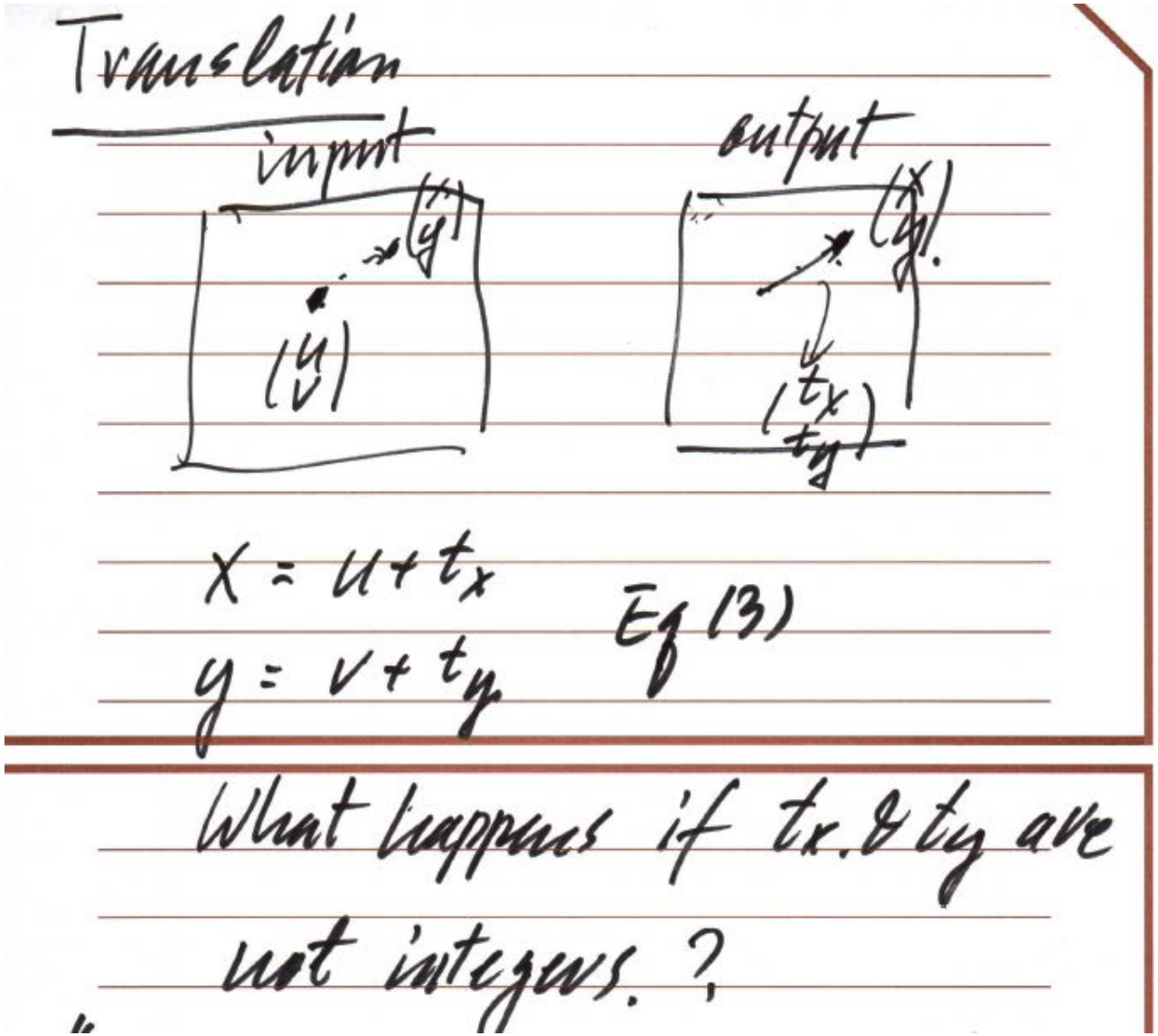


Translation



	Input	Output
Cartesian	$\begin{pmatrix} u \\ v \end{pmatrix}$	$\begin{pmatrix} x \\ y \end{pmatrix}$
image	$\begin{pmatrix} p \\ q \end{pmatrix}$	$\begin{pmatrix} j \\ k \end{pmatrix}$

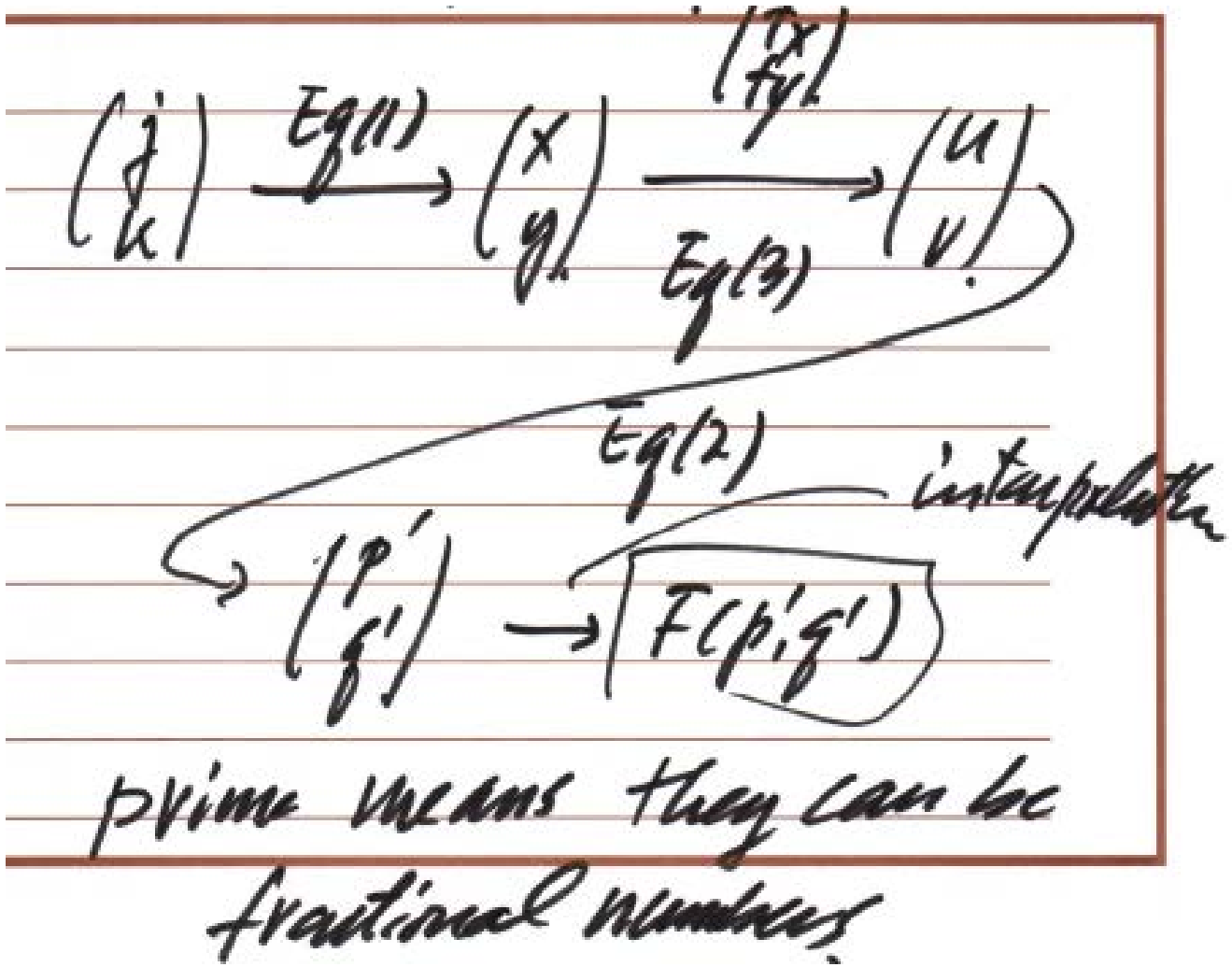
Non-Integer Translation



Inverse Address Mapping

- Render the output image (in integer pixel locations) by tracing back to the corresponding input pixel locations
- The input pixel locations can be fractional numbers
  - Use the interpolation technique to generate the corresponding values

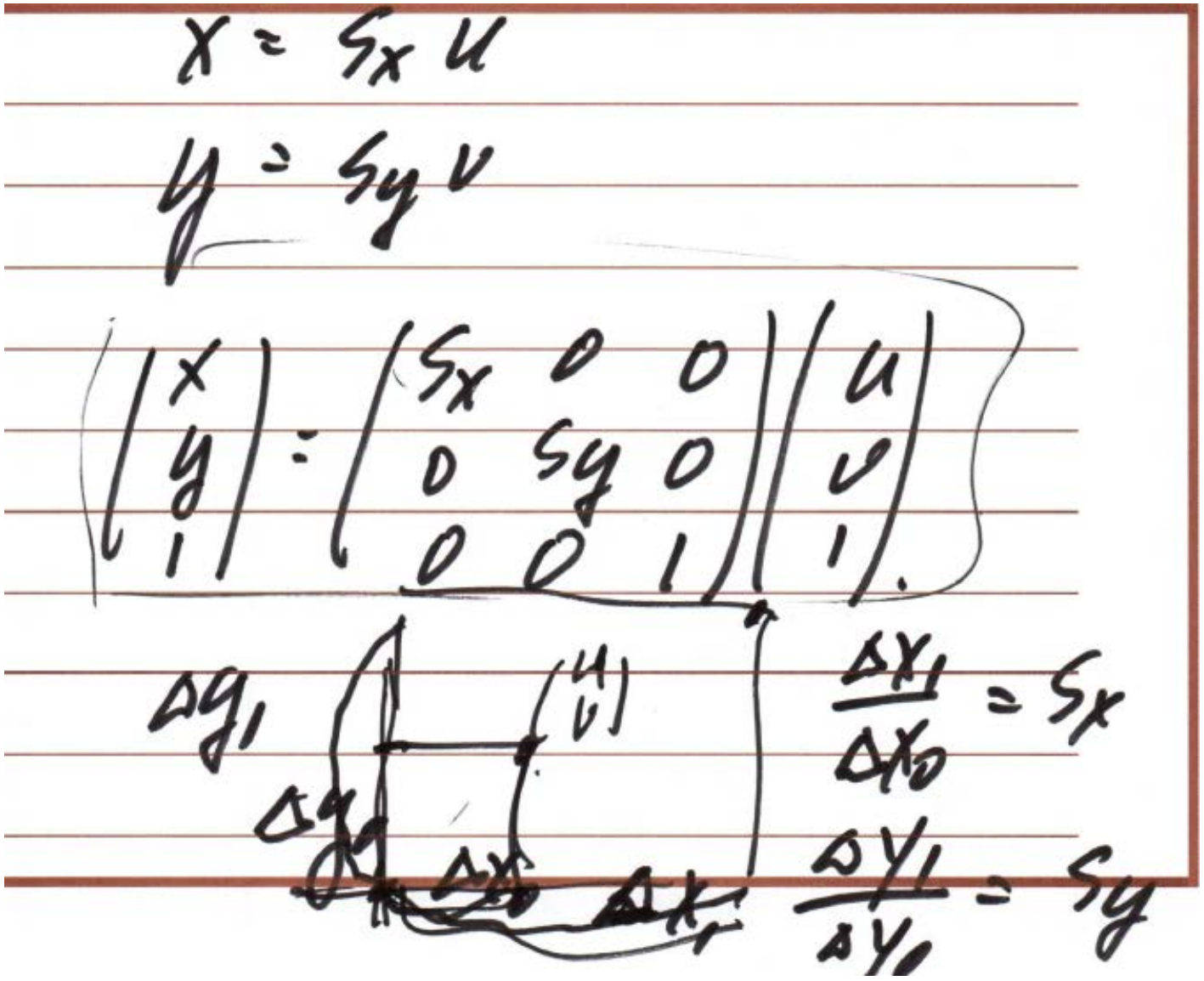
Inverse Address Mapping



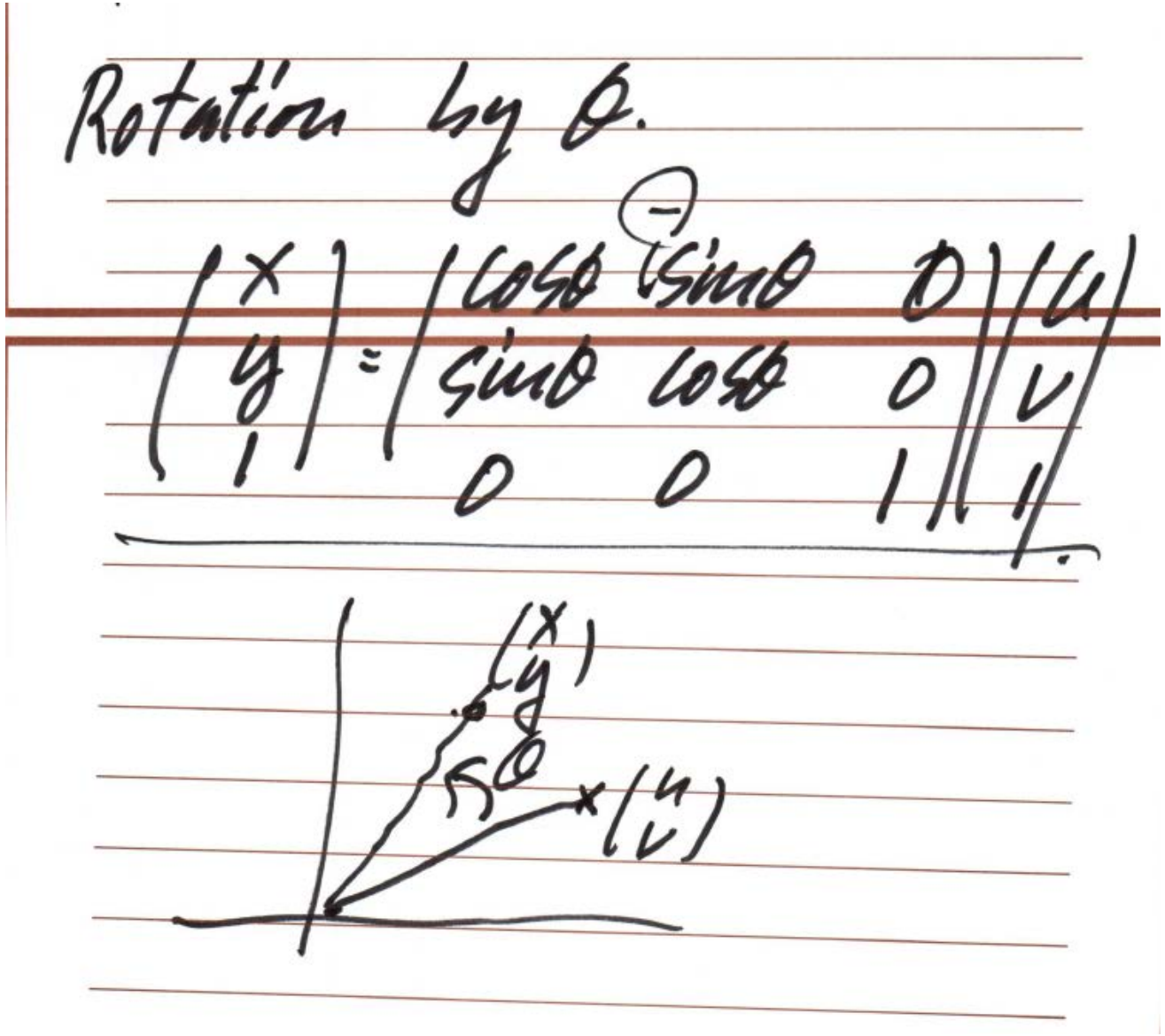
Rewrite Eq. (3): Affine to Linear System Conversion

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad \text{Eq. (3)}$$

Scaling


x = Sx u
y = Sy v
x
y
1
=
Sx 0 0
0 Sy 0
0 0 1
u
v
1
Δy1
Δy0
Δx0
Δx1
(u)
(v)
Δx1 / Δx0 = Sx
Δy1 / Δy0 = Sy

Rotation

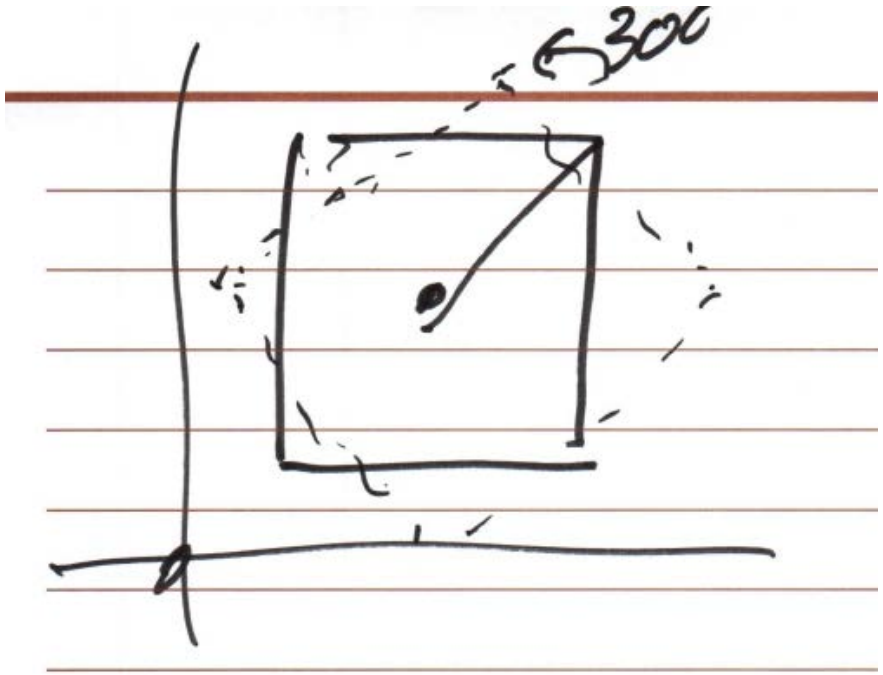

Rotation by θ.
$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{vmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$
(x, y)
θ
(u, v)

Compound Operation

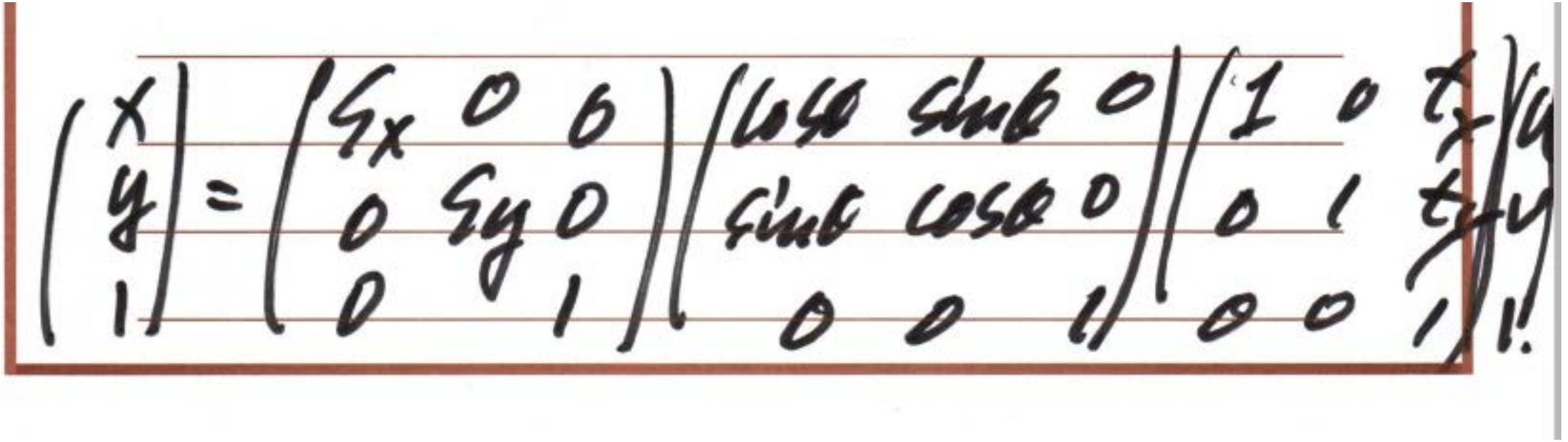
1) translate by $\begin{pmatrix} t_x \\ t_y \end{pmatrix}$.

2) rotate by θ.

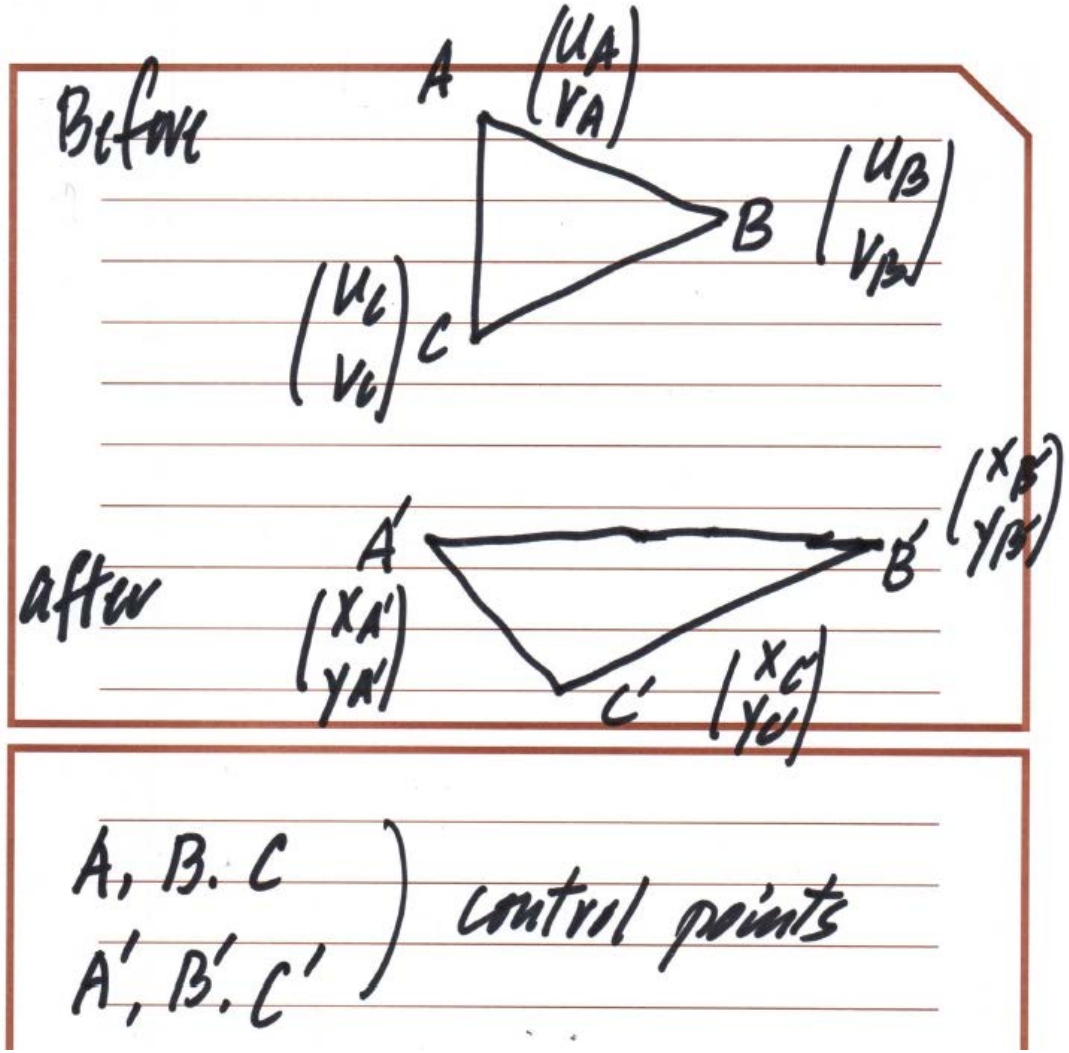
3) Scale by $\begin{pmatrix} s_x \\ s_y \end{pmatrix}$



Compound Operation



Manipulation of A Triangle



Manipulation of A Triangle

$$\bar{X}_{out} = T\,\bar{X}_{in}$$

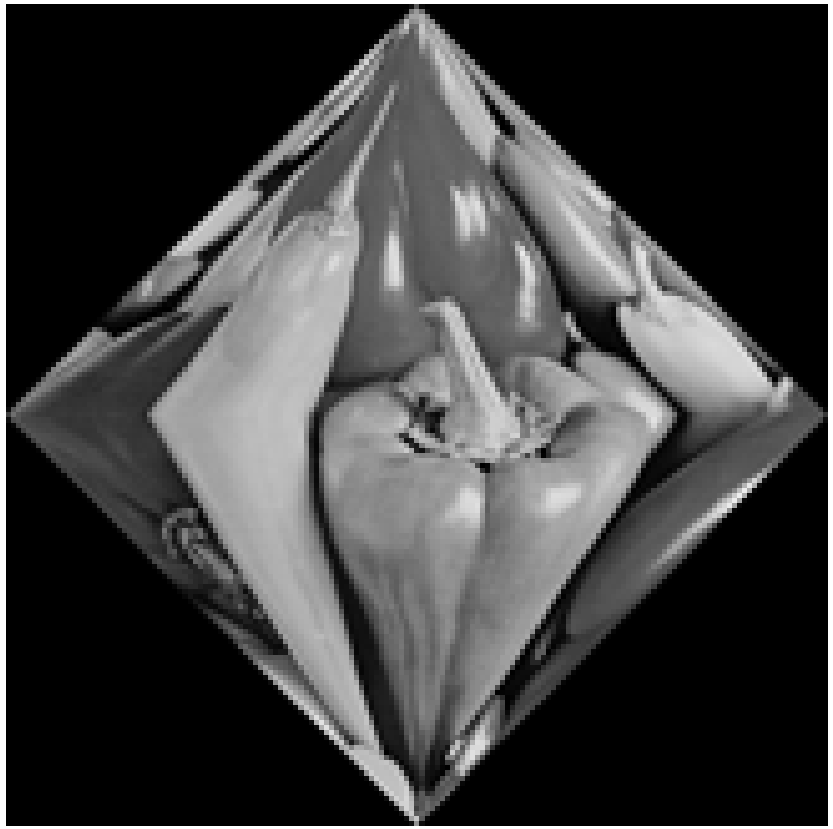
$$\begin{pmatrix} x_{A'} & x_{B'} & x_{C'} \\ y_{A'} & y_{B'} & y_{C'} \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix} \begin{pmatrix} u_A & u_B & u_C \\ v_A & v_B & v_C \\ 1 & 1 & 1 \end{pmatrix}$$

↑ A model

a system of 9 equations of 9 parameters

$$T = \bar{X}_{out}\,\bar{X}_{in}^{-1}$$

Polygon Manipulation Example (1)

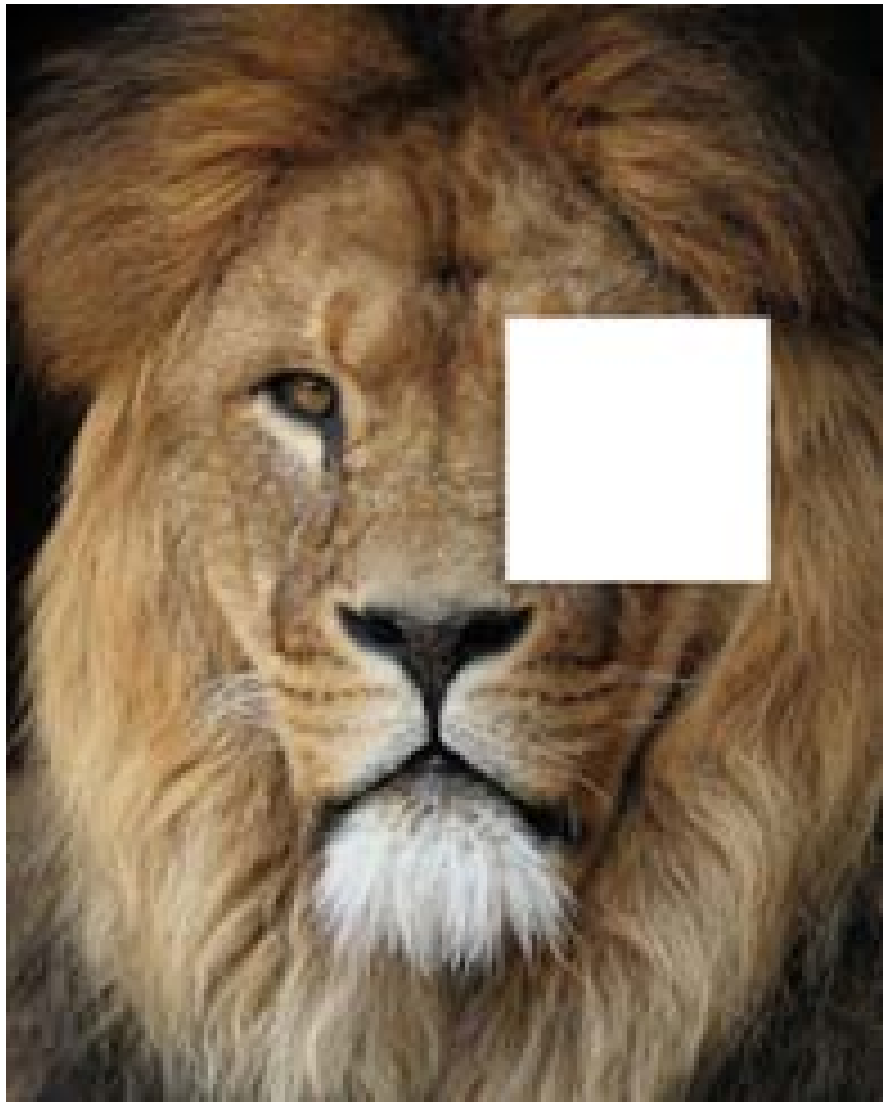


Polygon Manipulation Example (2)

Polygon Manipulation Example (3)

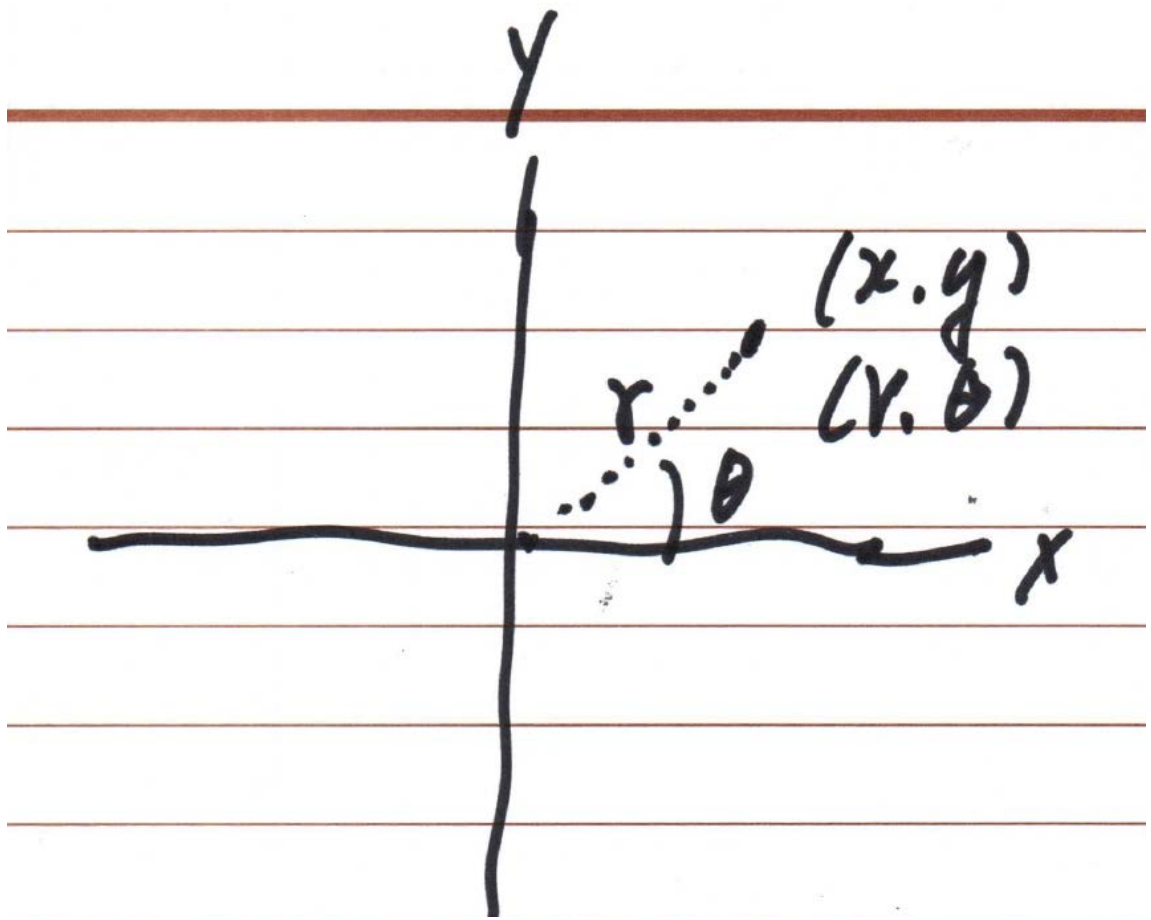
Polygon Manipulation Example (4)

Polygon Manipulation Example (5)

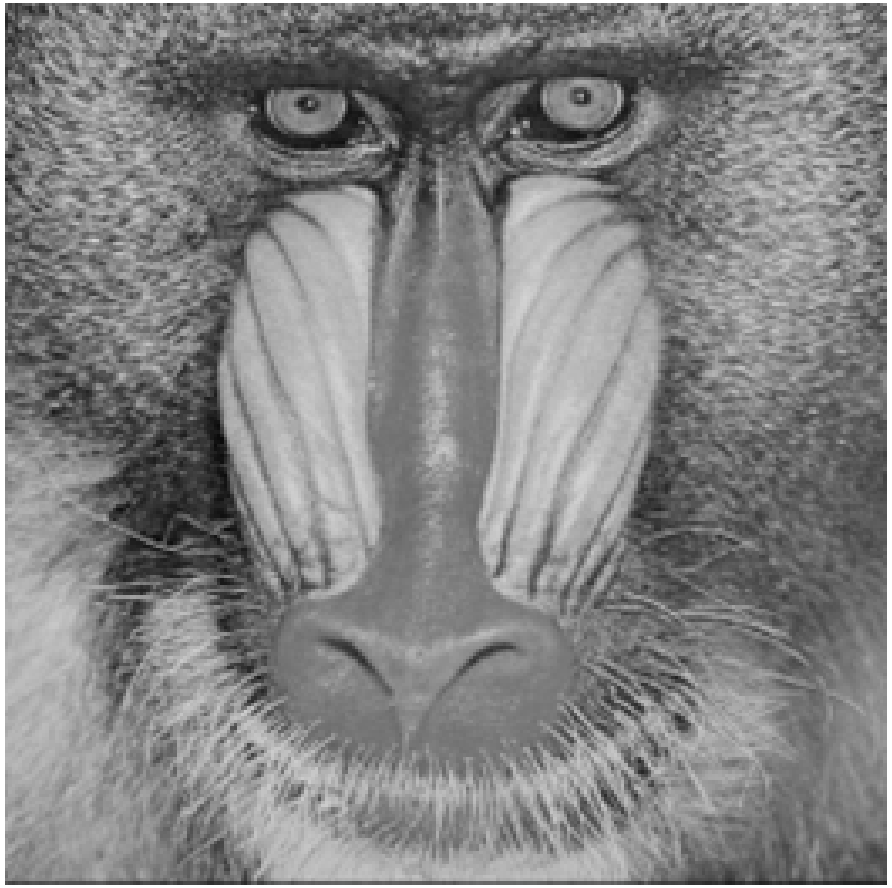


Cartesian/Polar Coordinates Transformation

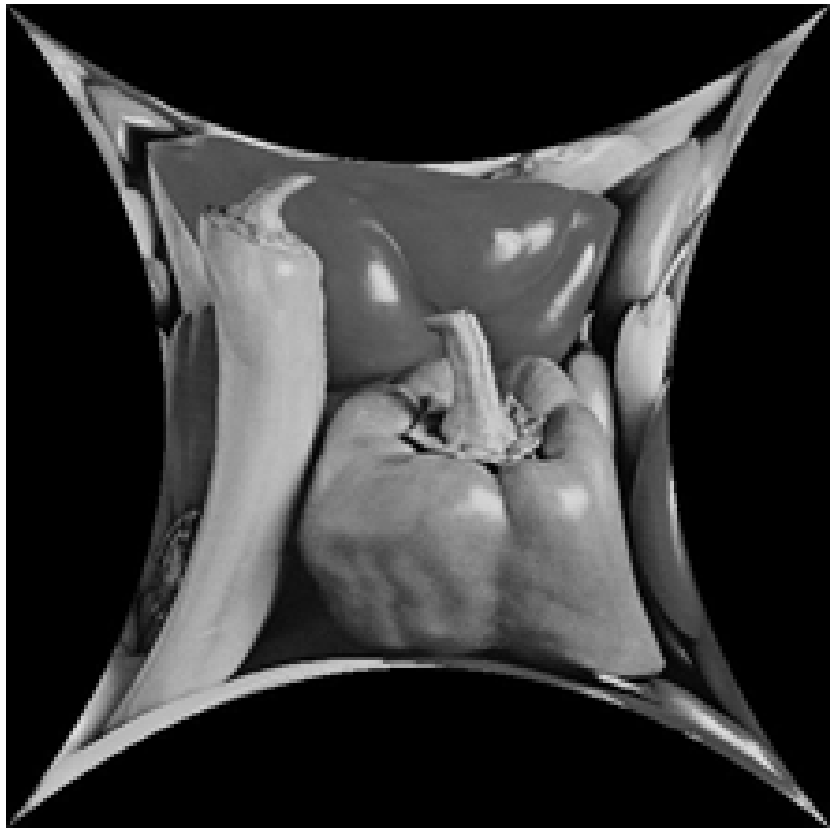
Some transformation is easier to do in the polar coordinates



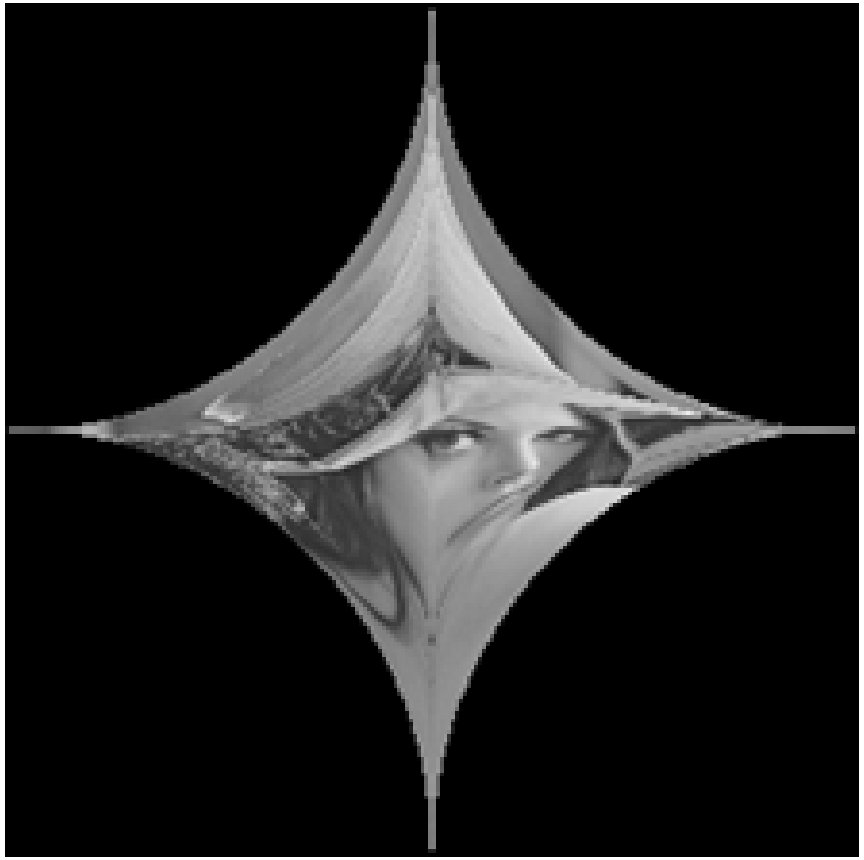
Example of Cartesian/Polar Coordinates Transformation (1)



Example of Cartesian/Polar Coordinates Transformation (2)



Example of Cartesian/Polar Coordinates Transformation (3)



Example of Cartesian/Polar Coordinates Transformation (4)

Example of Cartesian/Polar Coordinates Transformation (5)

Address Mapping + Intensity Mapping

- We have considered pixel address mapping without modifying its intensity (or color) value
- Can we do pixel address/intensity/color value mapping at the same time?
  - It is more challenging but feasible

Example: Facial Morphing

Trump

Biden

Face Image Morphing Video Clips

- Two sample projects on facial image morphing conducted by Prof. Kuo's graduate students in class EE569 (Introduction to Digital Image Processing) can be viewed from the following two websites:
- The best 14 results in the 2012 Fall class:
- The best 12 results in the 2011 Fall class:

  https://vimeo.com/user6200043
- Each video clip was conducted by a single student.