GENERAL MORPHING

Joe Manke, Adam Meaney, & Jonathan Tomes

THE BASICS

- Modify the spatial relationship between pixels in an image
- Consists of 2 basic operations
  - Spatial transformation of coordinates
  - Intensity interpolation

MULTIPLE APPROACHES

- Affine
- Bilinear Transformation
- Perspective

AFFINE TRANSFORMATION

AFFINE TRANSFORMATIONS

- Transform an image using the affine transformation matrix
- $[x \quad y \quad 1] = [v \quad w \quad 1]T = [v \quad w \quad 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$
- Scale, rotate, translate, or shear depending on the values in *T*
- Can chain multiple transformations by multiplying their matrices

Transformation	Affine Matrix *T*	Coordinate Equations
Scaling	$\begin{bmatrix} C_x & 0 & 0 \\ 0 & C_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = C_x v$ $y = C_y w$
Rotation	$\begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = vcos\theta - wsin\theta$ $y = vsin\theta + wcos\theta$
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$	$x = v + T_x$ $y = w + T_y$
Vertical Shear	$\begin{bmatrix} 1 & 0 & 0 \\ S_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + S_v w$ $y = w$
Horizontal Shear	$\begin{bmatrix} 1 & S_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = S_h v + w$

AFFINE MATRICES AND COORDINATE EQUATIONS

FORWARD AND INVERSE MAPPING

- Forward mapping – Map each pixel (u,v) from original image to (x,y) of output image
- Inverse mapping – Map each pixel (x,y) of output image to a pixel (u,v) of input image
  - $(u, v) = T^{-1}(x, y)$
- Interpolate intensity value from nearest input pixel(s)
- Inverse mappings are more efficient

- Scaling
  - $v = x/C_x$
  - $w = y/C_y$
- Rotation
  - $v = (x - x_0)cos\theta - (y - y_0)sin\theta + x_0$
  - $w = (y - y_0)cos\theta + (x - x_0)sin\theta + y_0$
  - (x_0, y_0) = center of image

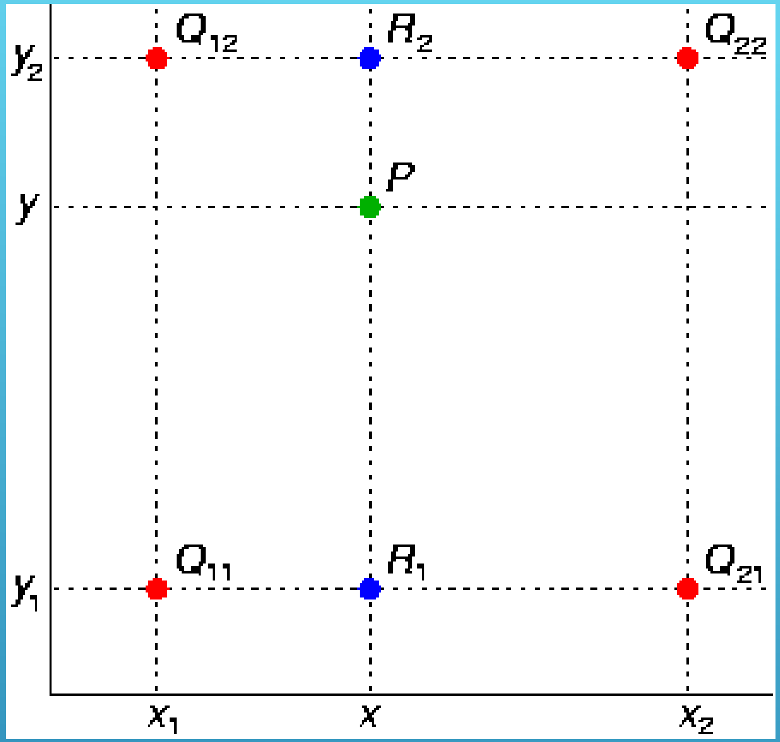
INVERSE EQUATIONS

BILINEAR INTERPOLATION

- R1 = ((x2 – x)/(x2 – x1))*Q11 +

  ((x – x1)/(x2 – x1))*Q21
- R2 = ((x2 – x)/(x2 – x1))*Q12 +

  ((x – x1)/(x2 – x1))*Q22
- P = ((y2 – y)/(y2 – y1))*R1 +
  - ((y – y1)/(y2 – y1))*R2



Original

Bilinear

Nearest Neighbor

COMPARISON

Original

Bilinear

Nearest Neighbor

COMPARISON

NOT THE BEST SOLUTION

- Works well for mapping triangles
- Done piecewise on other shapes
- The transformation is continuous, but there is discontinuity in the first derivative

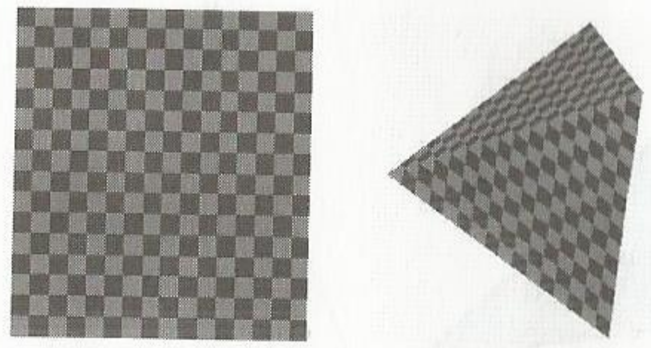


Figure 3.13 Piecewise affine interpolation.

BILINEAR AND PERSPECTIVE

BILINEAR

- Pros
  - Non-planar – useful for mapping none 2-D things
  - Very General
- Cons
  - Non-planar – none vertical or horizontal lines map to quadratic curves
  - Quadratic in the Inverse

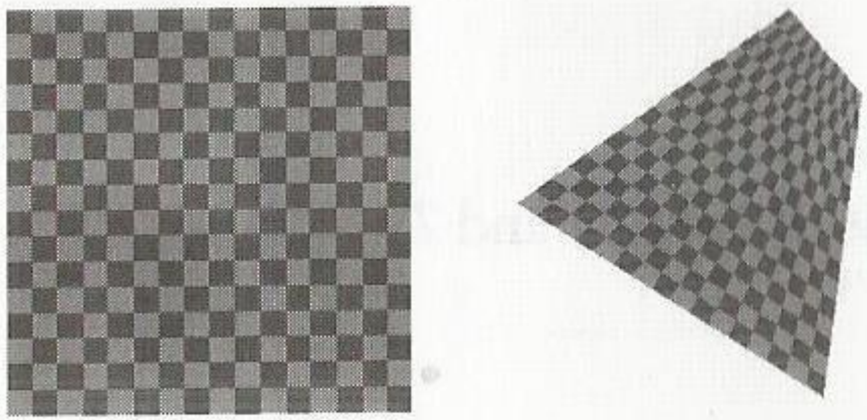


Figure 3.15 Bilinear interpolation.

- Pros
  - uses matrix operations much like the affine Transformations
  - Planar like the Affine Transformations – does Quadrilateral transforms well
  - Lines remain lines - though they foreshorten distant lines to a vanishing point, renders a nice realistic image
- Cons
  - Not as general as bilinear

PERSPECTIVE TRANSFORM

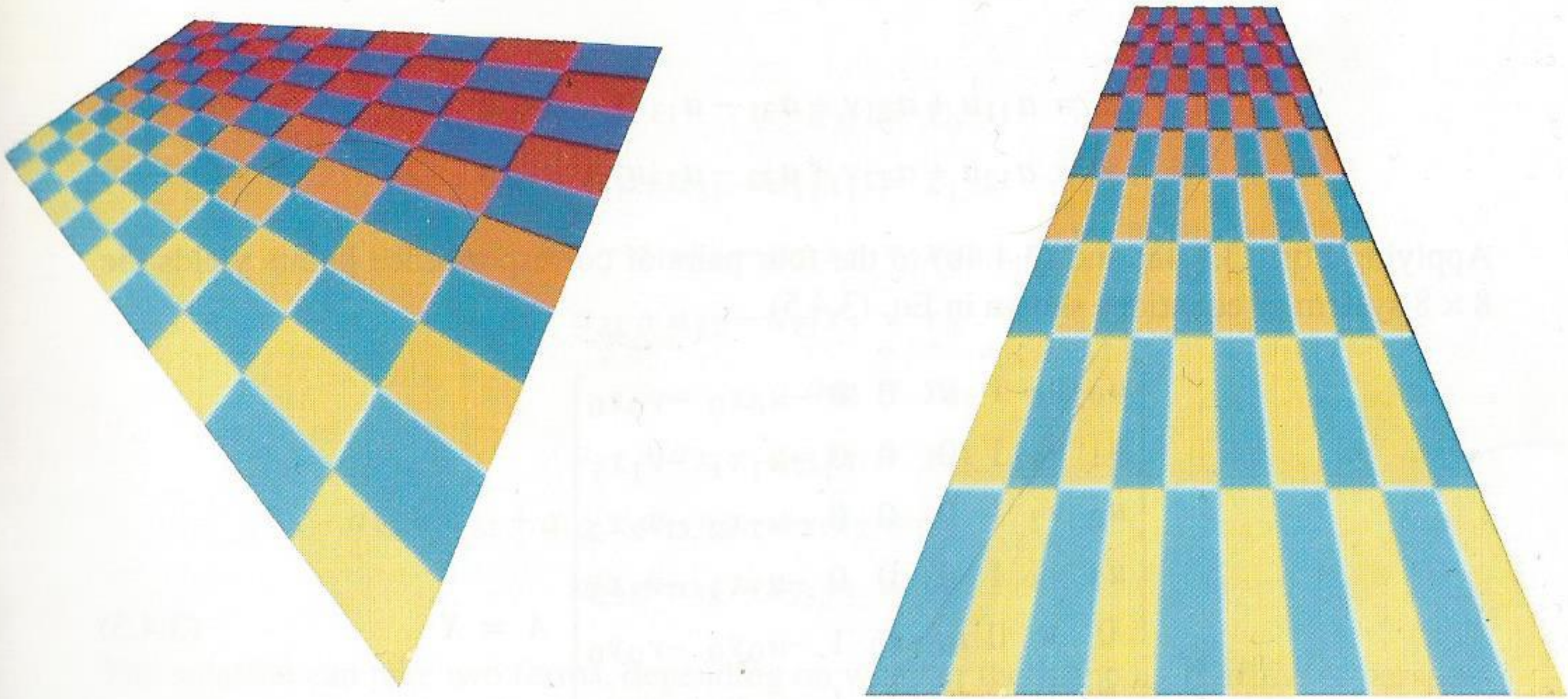


Figure 3.6: Perspective warps.

DOING THE PERSPECTIVE TRANSFORM

- $[x', y', w'] = [u, v, w]\, A$
- $x = \frac{x\prime}{w\prime} = \frac{a_{11}u + a_{21}v + a_{31}}{a_{13}u + a_{23}v + a_{33}}$
- $y = \frac{y\prime}{w\prime} = \frac{a_{12}u + a_{22}v + a_{32}}{a_{13}u + a_{23}v + a_{33}}$

THE FORWARD EQUATIONS

THE INVERSE EQUATIONS

- The Inverse is:
  - $[u, v, w] = [x', y', w']A^{-1}$
- But the inverse matrix is equal to the adj(A), so we can just use the Adjoint (also called adjugate or adjunct) matrix of A

- $a_{33} = 1$
- $x = a_{11}u + a_{21}v + a_{31} - a_{13}ux - a_{23}vx$
- $y = a_{12}u + a_{22}v + a_{32} - a_{13}uy - a_{23}vy$

INFERRING PERSPECTIVE TRANSFORMATIONS

- First some Definitions:
  - $\Delta x_1 = x_1 - x_2$, $\Delta x_2 = x_3 - x_2$, $\Delta x_3 = x_0 - x_1 + x_2 - x_3$
  - $\Delta y_1 = y_1 - y_2$, $\Delta y_2 = y_3 - y_2$, $\Delta y_3 = y_0 - y_1 + y_2 - y_3$

CASE 1: SQUARE TO QUADRILATERAL

- $a_{11} = x_1 - x_0 + a_{13}x_1$
- $a_{21} = x_2 - x_1 + a_{23}x_3$
- $a_{31} = x_0$
- $a_{12} = y_1 - y_0 + a_{13}y_1$
- $a_{22} = y_2 - y_1 + a_{23}y_3$
- $a_{32} = y_0$
- $a_{13} = 0 \text{ OR } \dfrac{\begin{vmatrix} \Delta x_3 & \Delta x_2 \\ \Delta y_3 & \Delta y_2 \end{vmatrix}}{\begin{vmatrix} \Delta x_1 & \Delta x_2 \\ \Delta y_1 & \Delta y_2 \end{vmatrix}}\ if\ \Delta x_3$!= 0 and Δy_3 != 0
- $a_{23} = 0$ OR $\dfrac{\begin{vmatrix} \Delta x_1 & \Delta x_3 \\ \Delta y_1 & \Delta y_3 \end{vmatrix}}{\begin{vmatrix} \Delta x_1 & \Delta x_2 \\ \Delta y_1 & \Delta y_2 \end{vmatrix}} if\ \Delta x_3$!= 0 and Δy_3 != 0

- Do Case 1 then find the inverse matrix (the Adj of the A you get)

CASE 2: QUADRILATERAL TO SQUARE

- Do case 2 then Case 1

CASE 3: QUADRILATERAL TO QUADRILATERAL

References:
Warping and Morphing of Graphical Objects
by Jonas Gomes, Lucia Darsa, Bruno Costa, Luiz Velho
Digital Image Warping
by George Wolberg
Digital Image Processing
by Rafael C. Gonzalez , Richard E. Woods