UBA – Facultad de Ciencias Exactas y Naturales – Departamento de Computación

Álgebra Lineal Computacional

Transformaciones Lineales

1. Introducción

Sean (w, z) y (x, y) dos sistemas coordenados, denominados el espacio de entrada y espacio de salida respectivamente. Una transformación geométrica de coordenadas define al mapeo desde el espacio de entrada al de salida de la forma:

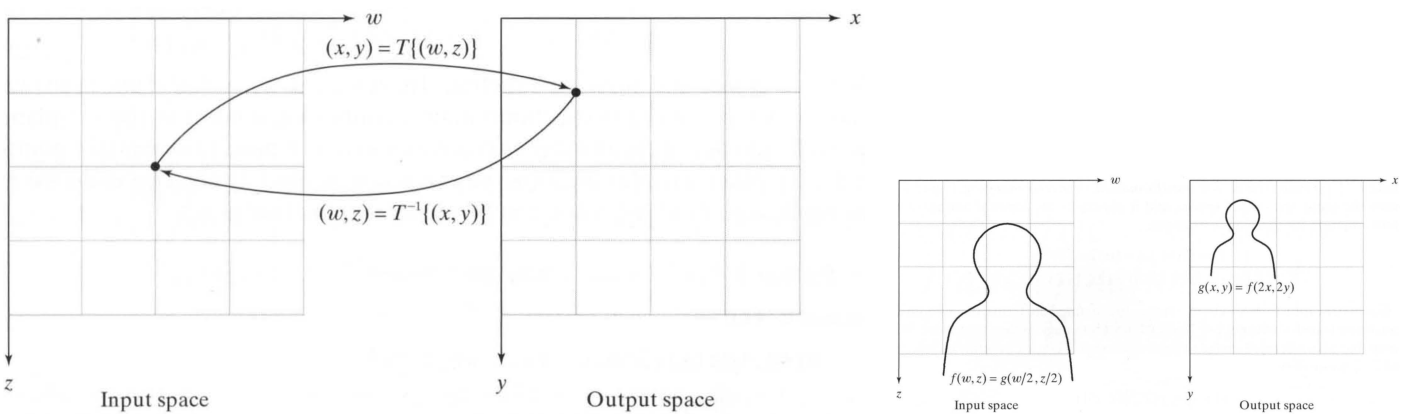
$$(x, y) = T\{(w, z)\}$$

donde $T\{\cdot\}$ se llama transformación o mapeo directo. Si $T\{\cdot\}$ posee una inversa, el mapeo inverso es aquel que traslada los puntos desde el espacio de salida al de entrada:

$$(x, y) = T^{-1}\{(w, z)\}$$

En la fig. 1 se ilustra el mapeo directo e inverso para el ejemplo:

$$\begin{aligned} (x, y) &= T\{(w, z)\} = (w/2, z/2) \\ (w, z) &= T^{-1}\{(x, y)\} = (2x, 2y) \end{aligned}$$



(a) (b)

Figura 1: Transformación lineal

Las transformaciones geométricas de las imágenes se definen en términos de la transformación de coordenadas.

La fig. 1 (b) muestra el efecto de aplicar la transformación lineal definida por $(x, y) = T\{(w, z)\} = (w/2, z/2)$ que escala la figura origina encogiéndola a la mitad.

De forma matricial, la proyección (y su inversa) se puede formarlizar como:

$$\begin{pmatrix} x \\ y \end{pmatrix} = T \begin{pmatrix} w \\ z \end{pmatrix} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} w \\ z \end{pmatrix}$$

$$\begin{pmatrix} w \\ z \end{pmatrix} = T^{-1} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Ejercicio 1. *Deformación*

En este caso se trasladan las coordenadas horizontales un factor que depende de las verticales, y las coordenadas verticales quedan sin modificación, provocando una deformación de la salida:

$$(x, y) = T\{(w, z)\} = (w + 0{,}4z, z)$$

Se pide:

(a) Encontrar la expresión de la matriz T y T^{-1}.

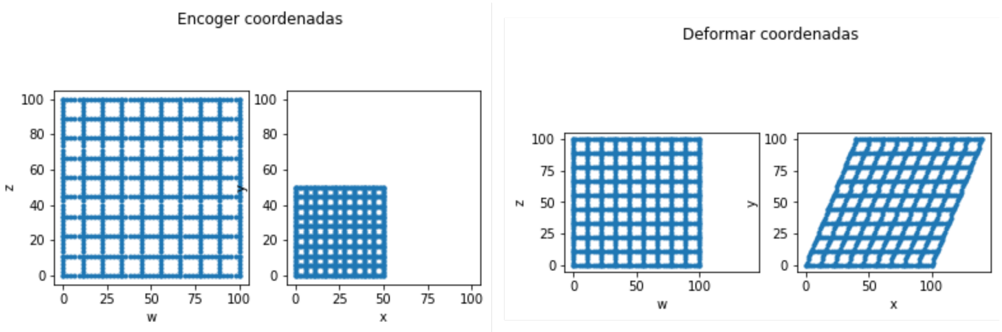


Figura 2: Salida del script de Python

(b) Desarrollar una función en Python `proyectarPunto(ptos, T)`, donde ptos es un array de $2 \times n$ dimensiones, donde n es la cantidad de puntos a proyectar, y T tiene 2×2 dimensiones. La función devuelve la proyección de los puntos a partir de la transformación lineal definida por T. Utilizar el template de Python dado por la cátedra.

Ejercicio 2. *Rotación*
Sea $R_\theta : \Re^2 \to \Re^2$ denota la rotación de un ángulo θ alrededor del origen, como muestra la figura 3.

$$\begin{pmatrix} X_\theta \\ Y_\theta \end{pmatrix} = R \begin{pmatrix} X \\ Y \end{pmatrix}$$

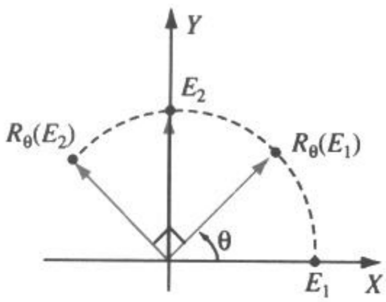


Figura 3: Transformación de rotación

Se pide:

(a) Encontrar la expresión matemática de la matriz R, para cualquier valor de θ.

(b) Validar R utilizando el script Python.

2. Transformación proyectiva

En los ejercicios precedentes se estudió el caso de matrices de transformación que encogían, dilataban o rotaban la posición de puntos iniciales, en un nuevo espacio a través del álgebra lineal.

Estas transformaciones son un caso particular de la transformación proyectiva y denominada *afina*. En general se puede formalizar las operaciones a través de:

$$\begin{pmatrix} x \\ y \end{pmatrix} = T \begin{pmatrix} w \\ z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

al cual hemos agregado la posibilidad de hacer una traslación lineal de las coordenadas origen en los dos ejes utilizando los valores consantes (b_1, b_2). A través de estas transformaciones se puede: escalar, rotar, trasladar o deformar. Como una conveniencia matemática y computacional, todas estas tranformaciones pueden ser efectuadas en las llamadas **coordenadas homogéneas** y utilizando una matriz de 3×3:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w \\ z \\ 1 \end{pmatrix}$$

agregarle el 1 a las coordenadas, las convierte en homogéneas.

Ejercicio 3. Se pide:

(a) Modificar la función proyectarPts para que sea capaz de realizar la operación de transformación afina, o sea con matrices de 3×3.

(b) Encontrar la expresión para hacer cada una de estas operaciones con las coordenadas **{escalar, trasladar, rotar, deformar}**, usando la matriz afina. Chequear con el script de Python.