Laboratorio 4
TRANSFORMACIONES GEOMÉTRICAS UTILIZANDO EL CONCEPTO DE MAPEO BILINEAL

Abiel Porras Garro - 2020209597[1], Elias Castro Montero - 2020098930 [2]
[1]Ingeniería en Computación, Instituto Tecnológico de Costa Rica, San José, Costa Rica
[2]Ingeniería en Computación, Instituto Tecnológico de Costa Rica, San José, Costa Rica

I. DESARROLLO

Para los ejemplos se utilizará la siguiente imagen como base:

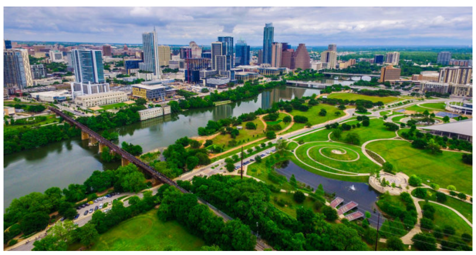

Fig. 1: Imagen de un paisaje

A. Punto 1.1

Para deducir la fórmula de mapeo inverso de acuerdo con el concepto general de mapeo bilineal, primero iniciar con la fórmula de mapeo bilineal:

$$w = \frac{az+b}{cz+d}$$

donde:

$$\{a, b, c, d\} \in \mathbb{C}$$

De acuerdo con [1] cuando el término $bc - ad$, llamado *determinante del mapeo*, es cero, el mapeo degenera en $w = \frac{a}{c}$, por lo que no tiene mapeo inverso, ya que dados los valores de a, c, sin importar cual sea el valor de z siempre se obtendrá el mismo valor w, por lo que cualquier valor de z mapeará al mismo punto en w.

Por otro lado, si el determinante del mapeo es distinto de cero, entonces el mapeo inverso existe, y está dado por el siguiente mapeo inverso, que también es invertible [1]:

$$z = \frac{-dw+b}{cw-a}$$

Con base en lo anteriormente descrito, es posible identificar que solo basta con definir un caso en el que la función de mapeo $w = f(z)$ sí tiene mapeo, este es, cuando la determinante del mapeo es distinto de cero. Por lo que se deduce la siguiente condición que cumplen los $w = f(z)$ que tengan mapeo inverso:

$$bc - ad \neq 0 \Leftrightarrow z = f(w) = \frac{-dw+b}{cw-a}, \text{ con } \{a, b, c, d\} \in \mathbb{C}$$

Esto se puede describir con la siguiente fórmula,

$$z = \frac{-dw+b}{cw-a}, bc - ad \neq 0$$

donde:

$$\{a, b, c, d\} \in \mathbb{C}$$

B. Punto 1.2

La siguiente función recibe como entradas las constantes complejas a, b, c, d y retorna `true` si estas generan una función de variable compleja que tenga mapeo inverso o `false` si su mapeo inverso no existe; para lo cual se utiliza la condición/caso deducido en el punto anterior, esto es que su determinante del mapeo, $bc - ad$ sea diferente de cero.

```
def tiene_mapeo_inverso(a, b, c, d):
  return b*c-a*d != 0
```

C. Punto 1.3

La función `newImage()` recibe la imagen y las constantes complejas como entradas para generar una nueva representación en el plano w. Primeramente mediante la función `new_size()` que realiza el mapeo bilineal de los bordes de la imagen para obtener el tamaño de la nueva imagen que se va a generar.

Posteriormente se realiza un bucle donde se pasa por todos los nuevos píxeles y mediante mapeo inverso se obtiene el valor del píxel en la imagen original. Se debe verificar que el píxel que se está evaluando este dentro de los píxeles de la imagen original en el plano z. Finalmente se retorna la nueva representación creada.

```
def newImage(img, a, b, c, d):
  new_width, new_height = new_size(img, a, b,
      c, d)
  mapeo = crear_mapeo_inverso(a, b, c, d)
  new_img = np.zeros((new_width, new_height),
      np.uint8)
```

```python
for x in range(new_width):
  for y in range(new_height):
    z = mapeo(complex(x, y))
    new_x = int(z.real)
    new_y = int(z.imag)
    if 0 <= new_x < img.shape[0] and 0 <=
        new_y < img.shape[1]:
      new_img[x, y] = img[new_x, new_y]
return new_img
```

Para el siguiente ejemplo se utilizaron los valores en las constantes complejas: $a = 3 + 2.1j; b = 0; c = 0.004; d = 4 + 6j$.

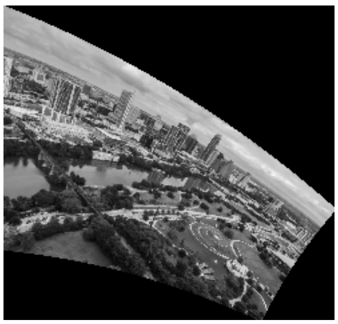

Fig. 2: Nueva representación obtenida

D. Punto 1.4

Para realizar los siguientes ejemplos se realizó una función que solo recibe como entrada la imagen y las constantes complejas a y b. Las constantes c y d tienen como valor 0 y 1 respectivamente con el fin de solo afectar el escalado, rotación y desplazamiento de la nueva representación.

```python
def newImageAB(img, a, b):
  new_img = newImage(img, a, b, 0, 1)
  return new_img
```

1) Punto 1.4.1

Para este ejemplo se debía utilizar un valor real en la constante a y el valor 0 en la constante b, por lo que se utilizaron los valores $a = 3; b = 0$. Como resultado la cantidad de píxeles se incrementó de 800x416p a 2397x1245p.

Fig. 3: Ejemplo para el punto 1.4.1

2) Punto 1.4.2

Para este ejemplo se debía utilizar un valor complejo en la constante a y el valor 0 en la constante b, por lo tanto se utilizaron los valores $3+1j; b = 0$. Como resultado la cantidad de píxeles se incrementó de 800x416p a 2812x1245p, y se giró la imagen hacia la izquierda.

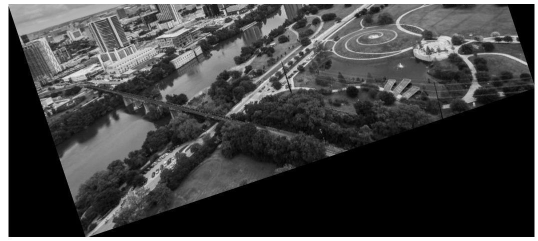

Fig. 4: Ejemplo para el punto 1.4.2

3) Punto 1.4.3

En este ejemplo el valor de a debe ser 1 y el valor de b debe ser distinto de 0 para comprobar el desplazamiento de la imagen. Para esta representación se utilizaron los valores $a = 1; b = 600 + 450j$. La imagen se desplazó hacia abajo y a la derecha.

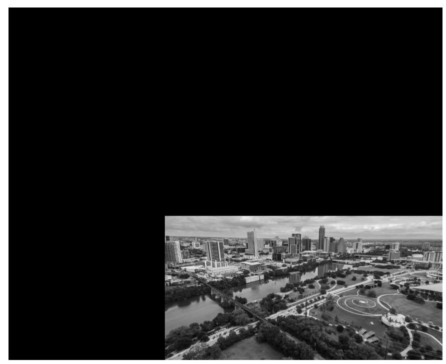

Fig. 5: Ejemplo para el punto 1.4.3

4) Punto 1.4.4

Finalmente, para este ejemplo los valores de a y b deben ser distintos de 0, por eso se les asignó los valores $a = 3+1j; b = 400 + 300j$. La cantidad de píxeles cambio a 3112x1645p, la imagen giro hacia la izquierda, y se desplazó había abajo y a la derecha.

Fig. 6: Ejemplo para el punto 1.4.4

REFERENCES

[1] P. Moya, *Señales y Sistemas. Fundamentos Matemáticos*, 1st ed. Costa Rica: Instituto Tecnológico de Costa Rica, Centro de Desarrollo de Material Bibliográfico, 2008.