Tarea de Investigación 2

1st Ericka Céspedes Moya
Tecnológico de Costa Rica
San José, Costa Rica
ericka.cespedes@gmail.com

2nd Esteban Alonso González Matamoros
Tecnológico de Costa Rica
Heredia, Costa Rica
esteb.gonza29@gmail.com

***Resumen*—El presente documento comprende la primera y segunda parte de la tarea de investigación 2 para el curso Introducción al Reconocimiento de Patrones. Para la primera parte se usa el concepto general de mapeo bilineal para programar varios ejercicios de procesamiento de imágenes en Python. Luego, para la segunda parte, se aplican varios métodos de filtrado de imágenes.**

***Index Terms*—procesamiento, imágenes, mapeo, bilineal, Python**

I. Primera parte

1. Deducir $z = f(w)$ (mapeo inverso), de acuerdo a la literatura recomendada definir de forma genera los casos de la función de mapeo $w = f(z)$ que sí tienen mapeo inverso.

Se le denomina polinomio bilineal de z y w a la expresión de la forma:

$$a_1 zw + a_2 z + a_3 w + a_4 = 0$$

Si se definen $a = -a_2, b = -a_4, c = a_1$ y $d = a_3$ se obtiene la forma más usual para un mapeo bilineal:

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

Este caso es un caso especial del mapeo bilineal con $c = 0$ y $d = 1$ mientras que el caso del mapeo de inversión es con $a = d = 0$ y $b = c = 1$.
El mapeo anterior se puede transformar en el de inversión. Multiplíquese para ello el término az por c/c y súmese $ad/c-ad/c$:

$$w = \frac{az+b}{cz+d} = \frac{\frac{a}{c}(cz+d)+b-\frac{ad}{c}}{cz+d} = \frac{a}{c} + \frac{bc-ad}{c(cz+d)}$$

donde la variable z aparece ahora una sola vez en el denominador del segundo término.
Si el término $(bc-ad)$ (denominado determinante del mapeo) es diferente de cero, entonces su mapeo inverso existe. [1]

2. Desarrolle una función que reciba como entradas las constantes complejas a, b, c, d y determine si las mismas genera una función de variable compleja cuyo mapeo inverso sí existe.

La siguiente función recibe como entrada las variables a, bc y d y retorna true o false según tenga mapeo inverso o no de acuerdo con la fórmula del determinante del mapeo.

```
def has_inverse_map(a, b, c, d):
  if (b*c-a*d) == 0:
    return False
  else:
    return True
```

Figura 1. Código del ejercicio 2

3. Desarrolle una función que reciba como entradas una imagen y las constantes complejas a, b, c, d y, tomando la imagen de entrada como el Plano z genere la representación de dicho plano en el Plano w.

```
def w_plane_representation(img, a, b, c, d):
  height = img.shape[0]
  width = img.shape[1]
  w_plane = np.zeros((height,width,3), np.uint8)
  for i in range(height):
    for j in range(width):
      z = complex(j, i)
      w = (a*z + b) / (c*z + d)
      if w.imag < height and w.real < width:
        w_plane[int(w.imag), int(w.real)] = img[i, j]


  cv2_imshow(w_plane)
```

Figura 2. Código del ejercicio 3

Figura 3. Imagen original

4. Desarrolle una función que reciba como entradas una imagen y las constantes complejas a, b y asuma que $c = 0 \wedge d = 1$ y genere el mapeo lineal y demuestre que:

```
def lineal_mapping_demonstration(img, a, b):
  c = 0
  d = 1
  w_plane_representation(img, a, b, c, d)
```

Figura 5. Código del ejercicio 4

a. El mapeo genera una magnificación cuando $b = 0$ para todos los casos $a \neq 0 \wedge a \in R$
Sí.

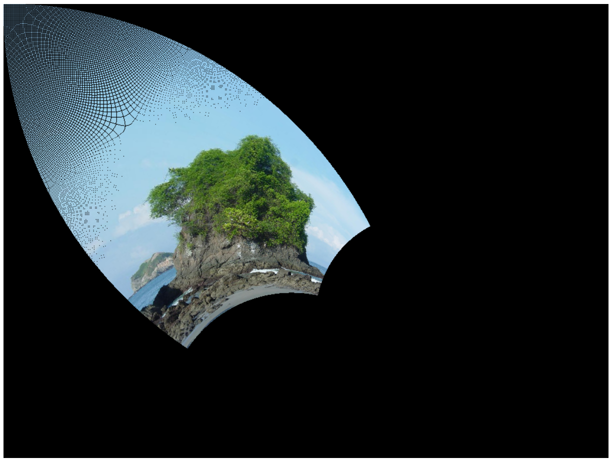

Figura 4. Resultado

Figura 6. Resultado del ejercicio a

b. El mapeo genera una magnificación y una rotación cuando b = 0 para todos los casos donde $a \neq 0 \wedge a \in C \wedge a \notin R$
Sí genera una rotación.

Figura 7. Resultado del ejercicio b

c. El mapeo genera únicamente un desplazamiento de todo el Plano z cuando $b \neq 0 \wedge a = 1 \wedge b \in C$
Sí se desplaza en el plano z.

Figura 8. Resultado del ejercicio c

d. Para el caso donde $a \neq 0 \wedge b \neq 0$, que el mapeo genera la combinación de una magnificación, una rotación y un desplazamiento de la imagen del Plano z en el Plano w.
Se magnifica, rota y se desplaza en el plano z.

Figura 9. Resultado del ejercicio d

II. Estudio comparativo de imágenes resultantes

II-A. Imágenes obtenidas

Para empezar el estudio comparativo, es necesario comprender la imagen original que fue utilizada durante el desarrollo de esta tarea:

Figura 10. Imagen original.

Luego, en el punto 2 de la tarea se le aplico un mapeo con unas constantes a, b, c, d que sí tuvieran un mapeo inverso. En nuestro caso elegimos $a = 2, b = 0, c = 0, d = 1$ con el propósito de que fueran constantes relativamente simples pero que demostraran los filtros aplicados en los próximos puntos de la tarea, esto resultó en que se le diera una magnificación a la imagen original.

Figura 11. Imagen obtenida luego del mapeo en el punto 2.

En el punto 3 se ejecuta el mapeo inverso de la imagen obtenida en el punto anterior (Punto 2), como resultado obtenemos la parte izquierda superior de la imagen original, esto nos deja como observación que no se podría reconstruir la imagen original a menos que la transformación dada por el mapeo bilineal logre trasladar correctamente todos los píxeles de la imagen original al espacio de la nueva imagen.

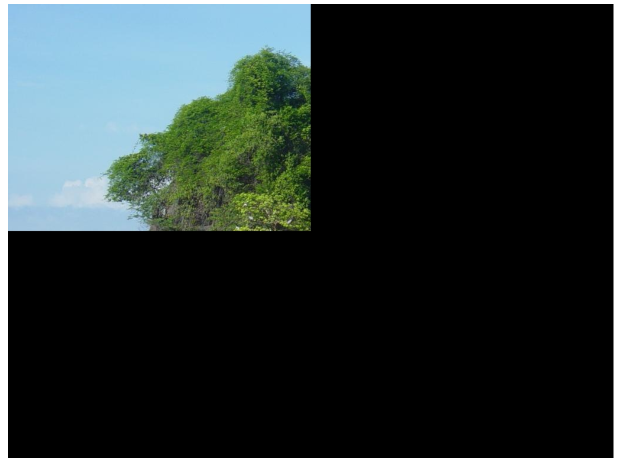

Figura 12. Imagen obtenida luego del mapeo en el punto 3.

En el punto 4 se le aplica filtro de colindancia con interpolación de $n = 4$, donde se aplica un proceso de mapeo inverso para cada píxel donde su RGB es igual a 0 (De este modo identificamos píxeles .ªpagados"), lo que permite averiguar su posible posición en la imagen original, luego, se saca el promedio del RGB de los píxeles superior, inferior, izquierdo y derecho, y el resultado se aplica al píxel de la imagen ingresada.

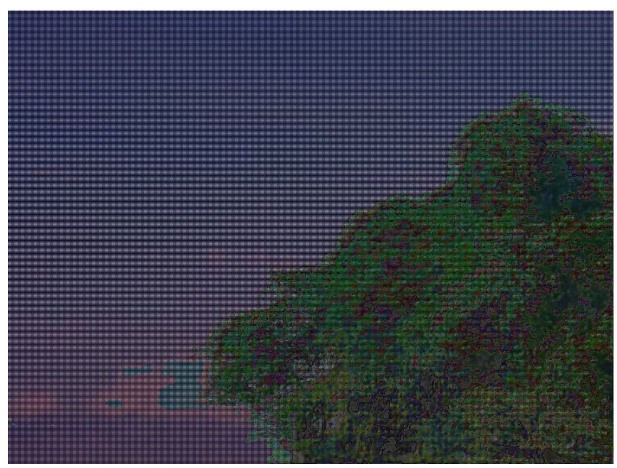

Figura 13. Imagen obtenida luego del filtro en el punto 4 a la imagen de la Figura 11.

En el punto 5 se le aplica filtro de colindancia con interpolación de $n = 8$, dentro de este punto se realiza un proceso muy parecido al anterior, con la única diferencia que el promedio que se obtiene es entre los 8 píxeles que rodean al píxel del cual queremos obtener el posible valor real de su RGB.

Figura 14. Imagen obtenida luego del filtro en el punto 5 a la imagen de la Figura 11.

En el punto 6 se utiliza un filtro Gausseano con una máscara de 5x5, para suavizar la *Figura 11*. El filtro en este punto se obtiene desenfocando o suavizando una imagen utilizando una función Gausseana para reducir el nivel de ruido. Según la fórmula, σ es la desviación estándar de la distribución mientras x y y son los índices de ubicación. El valor de σ controla la varianza alrededor de un valor medio de la distribución Gausseana, que determina la extensión del efecto de desenfoque alrededor de un píxel. Distintos valores de la σ y del tamaño del kernel cambian la forma de la Gausseana. Distintos valores de σ tienen distintos efectos en el filtrado del ruido. Mayores σ eliminan más ruido, pero difuminan más la imagen.

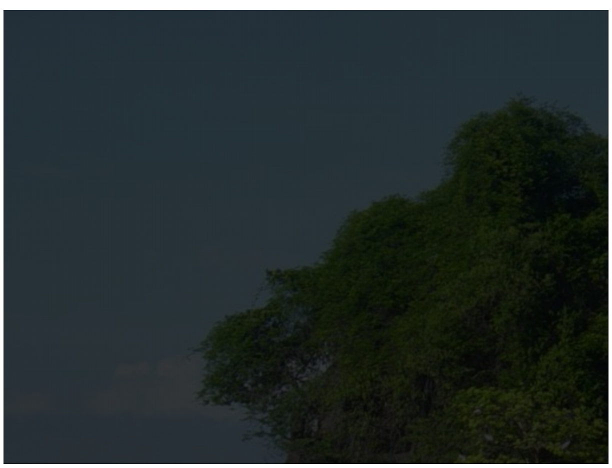

Figura 15. Imagen obtenida luego del filtro en el punto 6 a la imagen de la Figura 11.

En el punto 6 se utiliza un filtro Gausseano con una máscara de 5x5, para suavizar la *Figura 12*.

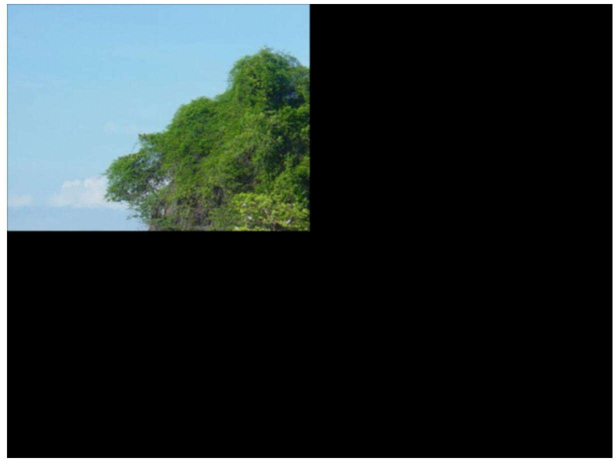

Figura 16. Imagen obtenida luego del filtro en el punto 6 a la imagen de la Figura 12.

II-B. Comparaciones

En base a la figura 13. En este caso, la imagen obtenida parece tener mucho contraste lo cual se refleja en que los colores del árbol resulten verse más fuertes al de la imagen sin filtro, además al fondo parece que se le agregaran más nubes y su color se aclarara un poco.

En base a la figura 14. En este ejemplo logramos observar que la imagen final es mucho más nítida y limpia con respecto a la anterior, las formas de las hojas y el árbol se notan de mejor manera, su contraste se ve mucho mejor que cuando se utilizó $n = 4$, del mismo modo, podemos apreciar que el cielo posee un color más estable con pequeñas variaciones en las zonas donde más se acerca al árbol.

III. Resumen de conclusiones más importantes

En este trabajo de investigación trabajamos con dos tipos principales de procesamiento de imágenes: filtrado de imágenes y deformación de imágenes. El filtrado de imágenes cambia el rango (es decir, los valores de píxeles) de una imagen, por lo que los colores de la imagen se alteran sin cambiar las posiciones de los píxeles, mientras que la deformación de la imagen cambia el dominio (es decir, las posiciones de los píxeles) de una imagen, donde los puntos se asignan a otros puntos sin cambiar los colores. El objetivo de utilizar filtros es modificar o mejorar las propiedades de la imagen y/o extraer información valiosa de las imágenes, como bordes, esquinas y manchas.

Con el filtro de interlineado con colindancia, se observa un aumento del contraste en la imagen, donde se acentúan más los distintos colores y se genera un tipo de sombra dependiendo de la zona de la imagen. También, se concluye que la mejor opción para aplicar un filtro de este tipo sería utilizar el de $n = 8$, esto debido a que genera una imagen más realista y con mejores colores, donde se logran observar mejor las distintas formas y detalles dentro de la imagen obtenida.

Con el filtro Gausseano, los píxeles más cercanos al píxel actual tienen más peso que los exteriores. Los pesos de los píxeles se calculan con una campana de Gauss y su varianza indica el nivel de suavizado. Los filtros Gausseanos tienen como ventaja el hecho de ser separables, es decir, se pueden realizar con la convolución de dos vectores unidimensionales en lugar de una máscara bidimensional. También mejoran la capacidad de suavizado, introduciendo un parámetro (la varianza) que es independiente del tamaño de la máscara.

REFERENCIAS

[1] P. Alvarado-Moya, *Señales y Sistemas. Fundamentos Matemáticos*. Cartago, Costa Rica: Centro de Desarrollo de Material Bibliográfico (CDMB), 2020, ch. 2: Variable compleja.