



DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE
EL7008-1 PROCESAMIENTO AVANZADO DE IMÁGENES

Tarea 1

Piramides de *Gauss* y *Laplace*

Diego Irarrázaval I.

Profesor:

Javier Ruiz del Solar.

Auxiliar:

Patricio Loncomilla Z.

Fecha:

24 de septiembre de 2020

Índice

1. Introducción	1
2. Marco Teórico	2
2.1. Cnvolución	2
3. Desarrollo	3
3.1. Pirámide de <i>Gauss</i> :	3
3.2. Pirámide de <i>Laplace</i> :	4
3.3. Reconstrucción imagen:	4
4. Conclusión	4
Bibliografía	5
5. Anexos	5

Índice de figuras

Índice de tablas

Índice de Códigos

1. Implementación de convolución en Cython. 3

1. Introducción

En esta tarea, se implementará el cálculo de pirámides de *Gauss* y *Laplace* de una imagen y, luego, se reconstruirán a partir de dichas pirámides. Para lograr esto, se deberá implementar también la convolución en dos dimensiones.

Los principales objetivos corresponden en primer lugar a introducir algunas formas de representaciones multi-resolución calculadas a partir de una imagen e implementar operaciones desde cero (por ejemplo la convolución) que usualmente se cargan con librerías.

El informe comienza con el marco teórico donde se expone brevemente sobre la convolución, la pirámide de Gauss, la pirámide de Laplace y la reconstrucción de la imagen.

A continuación, la sección desarrollo se divide en tres sub-secciones:

1. Pirámide de Gauss.
2. Pirámide de Laplace.
3. Reconstrucción imagen.

En las secciones enumeradas anteriormente, se incluye tanto el código implementado como análisis teórico de lo desarrollado.

Finalmente, se presentan las conclusiones del desarrollo de la tarea 1.

2. Marco Teórico

2.1. Cnvolución

En matemáticas, la convolución es una operación que recibe dos funciones (f y g) y entrega una tercera función ($f * g$) que describe como la forma de una es influida por la otra [1].

En procesamiento de señales, se puede entender como afecta se ve afectada señal pasar por un filtro. Para entender, es útil la siguiente ilustración:

- Describir operación de convolución

- Describir brevemente cálculo de la pirámide de Gauss

- Describir brevemente cálculo de la pirámide de Laplace

- Describir brevemente reconstrucción de la imagen original

3. Desarrollo

3.1. Pirámide de *Gauss*:

Convolución:

A continuación, se presenta el código de la implementación de la convolución:

```
1  cpdef float[:, :] convolution_cython(float[:, :] input, float[:, :] mask
   ):
2  cdef int a, b, r, c, rows, cols, row_init, col_init, i, j,
3  cdef float sum
4  # Imagen de salida
5  cdef np.ndarray output=np.zeros([input.shape[0], input.shape[1]], dtype =
   np.float32)
6
7  # Posicion a partir de la cual se puede realizar convolucion:
8  # Ejemplo 1: Para un kernel de 3x3, es (1,1).
9  # Ejemplo 2: Para un kernel de "a" x "b" es ("r"//2, "c"//2)
10 a = mask.shape[0]
11 b = mask.shape[1]
12
13 row_init = a // 2
14 col_init = b // 2
15
16 # tamaño de la imagen
17 rows = input.shape[0]
18 cols = input.shape[1]
19
20 sum = 0
21
22 # Recorremos la imagen input:
23 for r in range(row_init, rows - row_init):
24     for c in range(col_init, cols - col_init):
25         # Se recorre la mascara o kernel:
26         for i in range(a):
27             for j in range(b):
28                 sum += mask[i,j] * input[r-i,c-j]
29         # Guardamos el resultado de la suma correspondiente en el arreglo
   output:
30         output[r, c] = sum
31         sum = 0
32 return output
```

Código 1: Implementación de convolución en Cython.

La implementación en 1, corresponde a la convolución en dos dimensiones con padding.

- Describir implementación de convolución, incluyendo código - Describir implementación de cálculo de máscaras, incluyendo código - Describir implementación de suavizado de imágenes, incluyendo código - Describir implementación de submuestreo, incluyendo código - Describir implementación de pirámide de Gauss, incluyendo código - Describir implementación: graficar pirámide

de Gauss, incluyendo código - Prueba del sistema de cálculo de pirámide de Gauss sobre 4 imágenes entregadas, incluir las imágenes de las pirámides resultantes en el informe - Análisis del desempeño del cálculo de la pirámide de Gauss, analizando las imágenes resultantes

3.2. Pirámide de *Laplace*:

- Describir implementación de resta de imágenes, incluyendo código - Describir implementación de pirámide de Laplace, incluyendo código - Describir implementación de valor absoluto y escalamiento, incluyendo código - Describir implementación: graficar pirámide de Laplace, incluyendo código - Prueba del sistema de cálculo de pirámide de Laplace sobre 4 imágenes entregadas, incluir las imágenes de las pirámides resultantes en el informe - Análisis del desempeño del cálculo de la pirámide de Laplace, analizando las imágenes resultantes

```
1  from google.colab import files
2  uploaded = files.upload() #elegimos el archivo en el directorio
   correcto
3
4  !unzip EMG_data_for_gestures-master.zip #descomprimimos
5
```

3.3. Reconstrucción imagen:

- Describir implementación de suma de imágenes, incluyendo código - Describir implementación de duplicación de tamaño de imágenes con interpolación, incluyendo código - Describir implementación de reconstrucción de imagen original, incluyendo código - Prueba del sistema de reconstrucción de la imagen original usando las pirámides de las cuatro imágenes entregadas, incluir las imágenes reconstruidas en el informe - Análisis del resultado de la reconstrucción respecto a las imágenes originales

4. Conclusión

Bibliografía

[1] Wikipedia: Convolution.

https://en.wikipedia.org/wiki/Convolution#Visual_explanation

5. Anexos