



Objetivos

Esta experiencia tiene como objetivo aplicar herramientas de procesamiento de señales a imágenes. Específicamente se aplicarán convoluciones en dos dimensiones para implementar filtros en imágenes.

Instrucciones

1. El trabajo es en las mismas parejas de las experiencias pasadas.
2. Fecha de entrega: **Viernes 25 de Octubre de 2019 hasta las 23:55.**
3. La entrega consta de un informe de laboratorio (en PDF) y el código fuente con que se hicieron las pruebas.
4. La entrega tanto del informe (en PDF) como del programa debe ser en un archivo comprimido a través del link publicado en el curso de la plataforma Moodle www.udesantiagoovirtual.cl
5. Cualquier copia detectada entre los trabajos será calificada con nota mínima y será causal de reprobación del laboratorio.

Herramientas

Se utilizará el lenguaje de programación [Python 3](#) y algunos módulos de utilidad como: [Numpy](#), [Scipy](#), [Matplotlib](#).

Se recomienda utilizar algún IDE adecuado para el desarrollo de su trabajo por ejemplo: [PyCharm](#) o [Spyder](#), entre otros.

Si bien usar ejemplos y tutoriales encontrados en la web es una buena aproximación inicial, se recomienda fuertemente revisar o utilizar la documentación oficial del lenguaje y librerías. Debe citar los sitios web que usó como referencias.

Introducción

Las imágenes son señales de dos dimensiones, por lo tanto las herramientas para procesar señales en una dimensión tienen su extensión natural en el campo de las imágenes. En cada posición de la imagen o píxel se encuentra un valor que indica el brillo de ese píxel. Las imágenes pueden ser representadas como matrices con valores entre 0 y 255 o bien entre 0 y 1 según la aplicación.

Un filtro es un sistema lineal, como el de la Figura 1, que puede ser representado por su función de transferencia (respuesta al impulso) h . De este modo para una entrada f la salida g queda dada por la operación de convolución.

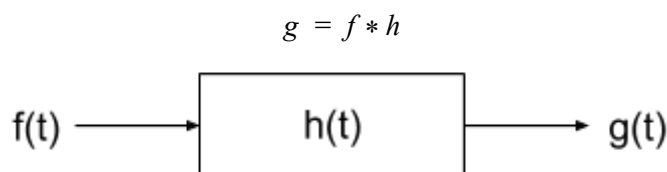




Figura 1. Sistema lineal

Es conocida la expresión unidimensional de la convolución

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(\tau - x)d\tau \quad \forall x$$

Por lo tanto la extensión a dos dimensiones queda dada por

$$g(x,y) = f(x,y) * h(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(i,j)h(i-x,j-y)di dj \quad \forall x,y$$

Cómo las imágenes son señales discretas y de valores finitos, la versión discreta de la convolución con un filtro h de dimensiones $n \times m$ se resume en:

$$g[k,l] = f[k,l] * h[k,l] = \sum_{i=0}^n \sum_{j=0}^m f[i,j] \cdot h[i-k,j-l] \quad \forall k,l$$

Esto es, para cada píxel (k,l) de la imagen se debe tomar los $n \times m$ píxeles adyacentes y calcular la suma ponderada por los $n \times m$ términos del filtro (o *kernel*), como se detalla en la Figura 2. Note que como la convolución en un punto usa los píxeles adyacentes, puede presentar dificultades en los bordes, lo que deben ser solucionado.

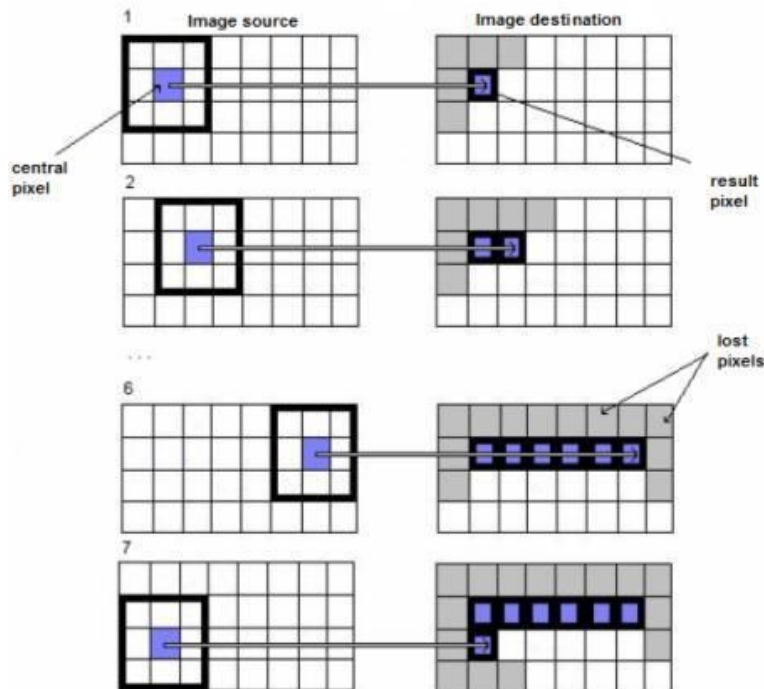


Figura 2: Se superpone H (cuadrado 3×3) sobre F (image source) para calcular cada valor de la matriz de salida G (image destination). El filtro H recorre toda la imagen F . Note que la salida G tiene dimensiones menores que la entrada F .



Desarrollo

1. Implemente la convolución en 2D entre una imagen de $N \times M$ y un *kernel* de $n \times m$ píxeles. La función debe retornar una matriz de $N \times M$ con el resultado de la convolución. No puede utilizar funciones similares existentes en otras librerías.
2. Utilice el siguiente kernel para implementar un filtro de suavizado gaussiano:

$$h = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

3. Implementar un filtro detector de bordes, utilizando el siguiente kernel:

$$h = \begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 2 & 0 & -2 & -1 \end{bmatrix}$$

4. Aplique los filtros implementados sobre la imagen de prueba "leena512.bmp" que se encuentran en la plataforma virtual y compare los resultados.
5. Calcule la transformada de Fourier en 2D para la imagen original y las imágenes filtradas. Puede utilizar funciones existentes en otras librerías.
6. En términos de frecuencias, explique a qué tipo de filtros corresponde cada uno de los kernels usados.

Informe

Se debe enviar un informe de laboratorio en formato PDF con todo el trabajo realizado y que incluya al menos las siguientes secciones:

- **Introducción** (0.5 - 1.0 página): Contexto, objetivos e información bibliográfica de relevancia (no es necesario repetir información que existe en la bibliografía, pero sí citar y/o sintetizar). *¿Qué se hará y por qué?*
- **Marco Teórico**: Explicaciones básicas sobre todos los temas y tópicos tratados en la actividad.
- **Desarrollo**: Códigos y resultados de cada etapa. *¿Qué se hizo y qué se obtuvo?*
- **Análisis de resultados**: Análisis de cada resultado, está correcto?, por qué salió ese resultado, relacionar resultados con los contenidos del curso. *¿Tienen sentido mis resultados, por qué obtuve estos resultados?*
- **Conclusiones** (0.5 - 1.0 página): Síntesis de los principales resultados encontrados y su relación con los contenidos. Problemas encontrados y cómo fueron solucionados. Conclusiones personales. *¿Qué aprendí con este trabajo?*



- **Bibliografía:** Listado de referencias usadas en el trabajo. Todas!. Libros (indicando capítulos), publicaciones, sitios web y videos (enlace y fecha de última visita), material de clases, etc. Formato APA. *¿Qué fuentes utilicé en este trabajo?*

Código

Se debe adjuntar el código del programa realizado, el cual debe cumplir con los principios de buenas prácticas de programación y documentación. Se evaluará:

- Completitud y correctitud: el código resuelve todo el laboratorio y funciona sin errores.
- Orden y documentación: el código está ordenado, es auto explicativo, presenta comentarios para explicar qué se resuelve en cada paso. (se valora/recomienda programar -funciones, variables- y comentar en inglés)
- Técnicas de programación: adecuado uso de paradigmas de programación (funcional, orientado a objetos, paralelismo, etc), estructura del código (correcto uso de funciones, clases, tipos de datos, estructuras de datos), testeo, documentación.
- Instrucciones de uso del código. Incluya instrucciones en el informe y/o en un archivo README.

Notas

1. Es muy probable que necesite normalizar la imagen a valores entre 0 y 1.
2. Implemente funciones que permitan leer, guardar y visualizar las imágenes generadas.
3. Note el problema de los bordes que presenta la convolución. Existen varias soluciones para este problema entre las que se incluyen: agregar un borde, recortar, rellenar con ceros, reflejar, etc.

Dudas y consultas por correo a los ayudantes y profesor de cátedra.