

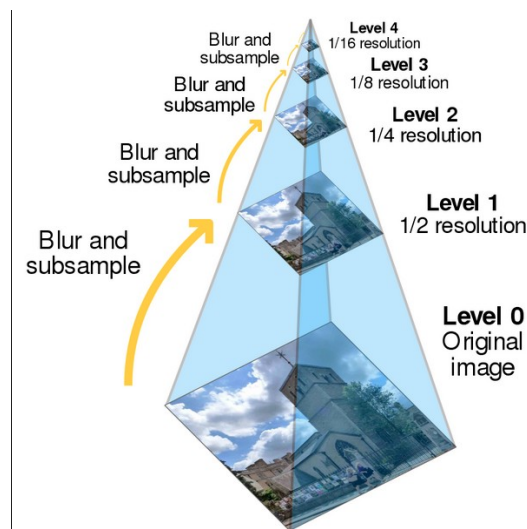


Objetivo

Utilizar MATLAB para el filtrado de información visual en el dominio espacial. En particular, se busca que el estudiante se familiarice con las posibilidades que brinda la herramienta para el filtrado de imágenes en diferentes contextos.

Pirámides Gaussianas y Laplacianas

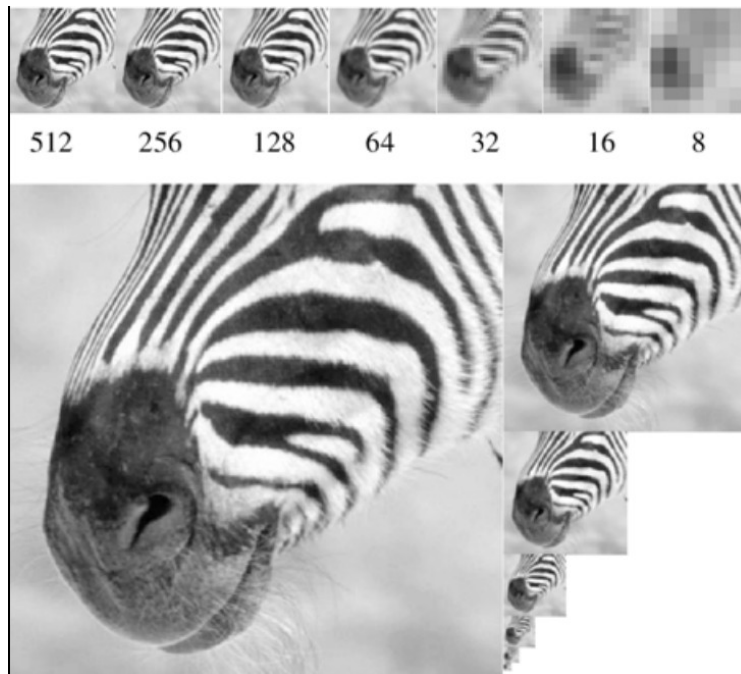
Dentro de las diferentes formas que existen para representar la información de una imagen, se encuentra el uso de representaciones piramidales, o pirámides. Es un tipo de representación multi-escala, en donde una señal o una imagen es iterativamente suavizada y submuestreada. De manera gráfica, la representación multi-escala (o jerárquica) puede verse como una pirámide, con la imagen original en la base y cada imagen resultado apilada encima de la anterior.



[https://en.wikipedia.org/wiki/Pyramid_\(image_processing\)](https://en.wikipedia.org/wiki/Pyramid_(image_processing))

Existen diferentes tipos de representaciones piramidales, las más comunes son:

- Pirámide Gaussiana: para el suavizado, se utiliza un kernel Gaussiano, y luego se aplica el submuestreo. De esta forma, se generan versiones de menor resolución de la imagen original, donde cada pixel contiene un promedio local de un vecindario en un nivel más bajo de la pirámide.



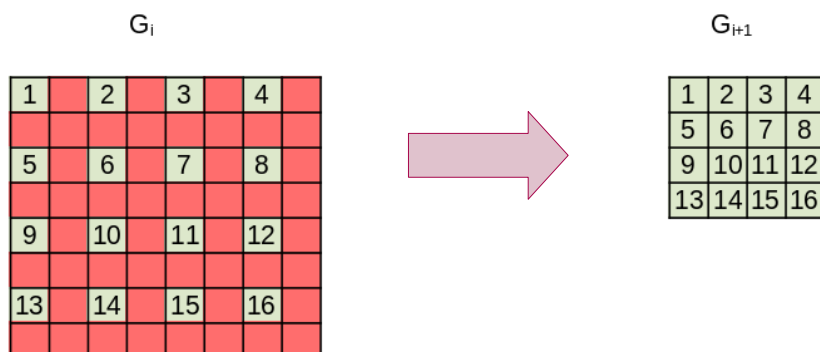
http://cs.haifa.ac.il/hagit/courses/ip/Lectures/Ip11_MultiscaleRepx4.pdf

Para generar una versión de menor resolución G_{i+1} de una imagen G_i , en una pirámide Gaussiana, se deben realizar los siguientes 2 pasos:

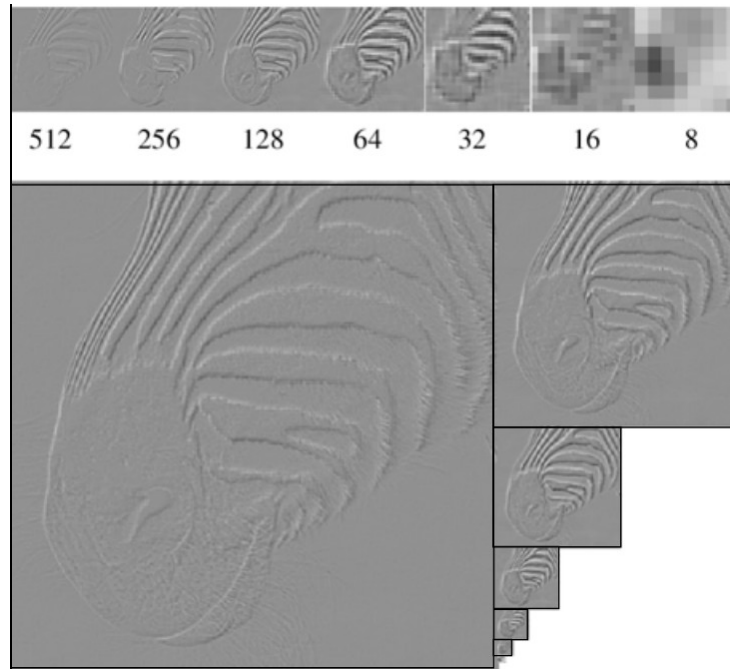
1. Filtrar la imagen G_i con un kernel Gaussiano como el siguiente:

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

2. Eliminar todas las filas y columnas pares de la imagen:



- Pirámide Laplaciana: similar a la pirámide Gaussiana, pero se almacena la diferencia entre las versiones suavizadas en niveles sucesivos. El único nivel que no cumple esta condición es el más pequeño, garantizando la reconstrucción completa de la imagen al ir incorporando las diferencias a medida que se avanza en la pirámide. Es muy útil como método de compresión de imágenes.



http://cs.haifa.ac.il/hagit/courses/ip/Lectures/Ip11_MultiscaleRep4.pdf

Desarrollo del taller

MATLAB provee en particular una función (`impyramid`) para cambiar la resolución de una imagen utilizando una pirámide Gaussiana. Sin embargo, el objetivo del ejercicio va a ser realizar el mismo proceso pero sin utilizar esta función, permitiendo generar las pirámides anteriormente descritas y utilizarlas como proceso de reconstrucción (o deconstrucción) de imágenes. Para desarrollar el taller, es necesario generar en MATLAB el código necesario que permita realizar los siguientes pasos:

1. Generar una pirámide Gaussiana a partir de la imagen original, calculando 4 niveles más pequeños de acuerdo a la operación descrita anteriormente. Las 5 imágenes de la pirámide (original y 4 niveles más pequeños) deben almacenarse en disco con numeración sucesiva para facilitar la visualización. Para un mejor (y más sencillo) resultado, se sugiere utilizar imágenes de tamaño cuadrado en potencias de 2.

Ejemplo: asumiendo una imagen de entrada de tamaño 512x512, los 5 niveles de la pirámide Gaussiana deberían ser:

- Nivel 4: imagen 32x32

- Nivel 3: imagen 64x64
 - Nivel 2: imagen 128x128
 - Nivel 1: imagen 256x256
 - Nivel 0: imagen original 512x512
2. Adicionalmente, generar una pirámide Laplaciana, a partir de la pirámide Gaussiana anterior, nuevamente en 5 niveles, partiendo de la imagen en su versión más pequeña (nivel 4 de la pirámide Gaussiana) y almacenando en cada nivel superior la diferencia con el siguiente nivel en la pirámide Gaussiana. Las 5 imágenes de la pirámide Laplaciana también deben almacenarse en disco con numeración sucesiva (y que pueda diferenciarse de la pirámide Gaussiana) para facilitar la visualización.

Ejemplo: siguiendo con la imagen de entrada de tamaño 512x512 para la cual ya se calculó la pirámide Gaussiana anteriormente, los 5 niveles de la pirámide Laplaciana serían:

- Nivel 4: imagen del nivel 4 Gaussiano, 32x32
 - Nivel 3: imagen diferencia entre nivel 3 y nivel 4, 64x64
 - Nivel 2: imagen diferencia entre nivel 2 y nivel 3, 128x128
 - Nivel 1: imagen diferencia entre nivel 1 y nivel 2, 256x256
 - Nivel 0: imagen diferencia entre nivel 0 y nivel 1, 512x512
3. Finalmente, utilizar la información almacenada en la pirámide Laplaciana para reconstruir la imagen original. Esto se logra al sumar la información de todos los niveles de la pirámide en una sola imagen (del tamaño de la imagen original). La imagen reconstruida debe almacenarse en disco para facilitar la visualización posterior. Para identificar la cantidad de información perdida en el proceso, se puede comparar esa reconstrucción con la imagen original.

Recomendaciones:

- El kernel Gaussiano para filtrar la imagen debe montarse manualmente y aplicarse con la función `conv2` o `imfilter`.
- Para cambiar los tamaños de las imágenes entre los diferentes niveles, puede utilizarse la función `imresize`, teniendo cuidado en seleccionar el método de interpolación más adecuado entre las opciones dadas por la función.

Entrega del taller

La entrega del taller consistirá en un único archivo comprimido (único formato aceptado: .zip), nombrado con los apellidos de los integrantes del grupo, el cual contendrá el código fuente del ejercicio propuesto (en MATLAB), las imágenes de ejemplo utilizadas (mínimo 3 imágenes médicas de modalidades diferentes, solo 2D) y los resultados obtenidos sobre esas imágenes. Este archivo deberá enviarse a través de la correspondiente asignación en BrightSpace antes de finalizar la clase del martes 31 de octubre de 2023. El envío del archivo comprimido en otro formato diferente al especificado resultará en una calificación de (0.0/5.0) para el taller.

La escala de evaluación es la siguiente:

- **Excelente (5.0/5.0):** El estudiante propone un ejercicio que realiza todos los pasos del procesamiento para ambas pirámides, y genera resultados adecuados.
- **Bueno (3.5/5.0):** El estudiante propone un ejercicio que realiza todos los pasos del procesamiento sólo para una de las pirámides, y genera resultados adecuados.
- **Acceptable (2.5/5.0):** El estudiante propone un ejercicio que realiza sólo algunos de los pasos del procesamiento, y genera algunos resultados, no todos correctos.
- **Malo (1.0/5.0):** El ejercicio propuesto por el estudiante no es correcto en su sintaxis (no puede interpretarse en el entorno de ejecución adecuadamente).
- **No entregó (0.0/5.0):** El estudiante no entrega el ejercicio solicitado.