

Aseguramiento de la Calidad del Software: Proyecto Semestral II

M. Sc. Saúl Calderón Ramírez
Instituto Tecnológico de Costa Rica,
Escuela de Ingeniería en Computación,
PAttern Recognition and MACHine Learning Group (PARMA-Group)

19 de febrero de 2017

El presente proyecto pretende ser desarrollado en grupos de tres personas, a lo largo del curso de aseguramiento de la calidad del software. El proyecto será insumo para aplicar distintos estándares de calidad en sus varias etapas, por lo que a lo largo del curso se realizarán tareas y agregados específicos al proyecto. El objetivo es desarrollar un sistema que mejore el contraste y atenúe el ruido presente en las imágenes

Fecha de entrega: 31 de mayo.

1. Introducción y motivación

Los dispositivos de captura y procesamiento de imágenes biomédicas, como resonancias magnéticas, radiografías y tomografías computarizadas, cobran cada vez mayor protagonismo los procesos de toma de decisiones para los profesionales de salud. Más recientemente, la industria biomédica ha desarrollado interesantes sistemas de software que realizan diagnósticos automáticos a partir de imágenes y señales biomédicas.

Un ejemplo de ello es el Laboratorio de Quimiosensibilidad tumoral, de la Universidad de Costa Rica, donde los doctores Steve Quirós y Rodrigo Mora desarrollan proyectos de investigación relacionados con la terapia personalizada para pacientes con cáncer. Su premisa consiste en desarrollar tratamientos quimioterapéuticos personalizados a cada paciente, de forma que los químicos utilizados sean optimizados. Para ello es necesario realizar experimentos *in vivo* para estudiar la reacción de a nivel celular en el tejido cancerígeno a un tratamiento específico. El laboratorio cuenta con microscopios electrónicos de fluorescencia, los cuales permiten capturar imágenes digitales de la actividad celular en un tejido, posibilitando la visualización de las distintas reacciones de un conjunto de células a un tratamiento específico.

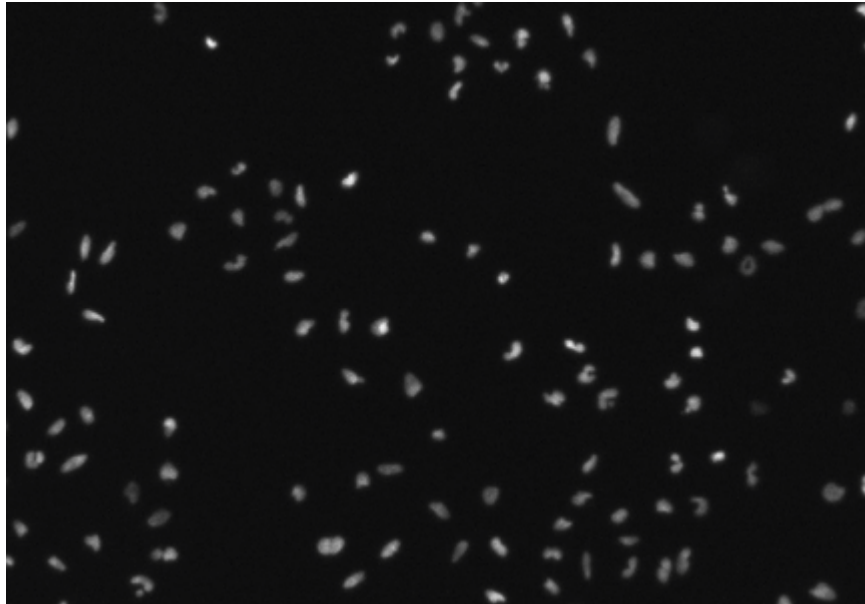


Figura 1: Células en tejido de glioblastoma, en las primeras horas del experimento.

Los experimentos típicos tardan de 72 a 96 horas, y se toman muestras cada 10 minutos, por lo que es usual la generación de miles de imágenes para cada experimento. El análisis manual de esas miles de imágenes para determinar la cantidad de células en un momento específico, o incluso determinar la descendencia de una o más células, son tareas difíciles (e incluso muchas veces imposibles) para el experto. Un sistema que automatice la detección, rastreo y determinación de la línea hereditaria de cada célula tiene el potencial de ahorrar una cantidad importante de tiempo a los expertos, además de la posibilidad de analizar muchos más experimentos, facilitándoles el enfocarse en aspectos de mayor relevancia.

2. Preprocesamiento de las imágenes

Las imágenes generadas por un microscopio electrónico basado en fluorescencia usualmente presentan ruido. El ruido se refiere a cambios aleatorios en la señal debido a fuentes externas, como por ejemplo variaciones térmicas, electromagnéticas, etc. Existen distintos tipos de ruido, el más frecuente es el ruido Gaussiano, el cual se puede observar en la Figura 2.

Para atenuar el ruido en las señales, y en el caso específico de las imágenes, se realiza una transformación de la señal de entrada conocida como *filtrado*. El filtrado se refiere a la eliminación de ciertas características de la señal. El

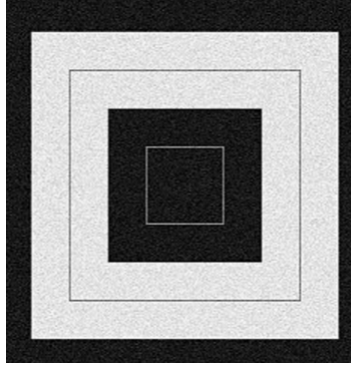


Figura 2: Ruido Gaussiano.

filtrado para una imagen $U \in \mathbb{R}^{N \times M}$, usando un funcional conocido como *filtro* $F \in \mathbb{R}^{K \times L}$, con $K < N$ y $L < M$, se realiza mediante la operación de convolución, la cual se define como:

$$(U * F)[x, y] = \sum_{u=x-\frac{(K-1)}{2}}^{x+\frac{(K-1)}{2}} \sum_{v=y-\frac{(L-1)}{2}}^{y+\frac{(L-1)}{2}} U[u, v] F[x-u, y-v]. \quad (1)$$

La convolución consiste en *deslizar* la función F sobre la función o matriz U , como lo ilustra la Figura 3, donde se calcula la convolución para el punto $[x=2, y=2]$, $(U * F)[2, 2]$, y para este caso $N = M = 7$ y $K = L = 3$.

Observe que dependiendo de la posición $[x, y]$ a evaluar y las dimensiones del filtro F , puede que la convolución no esté definida, por lo que se evita su evaluación, o alternatively se agregan filas y columnas de ceros alrededor de la matriz antes de realizar la convolución.

Para atenuar el ruido, se utilizan filtros como el Gaussiano **con origen** en $\frac{K-1}{2} + 1$ y $\frac{L-1}{2} + 1$ y con desviación estándar σ y sin normalizar:

$$F[x, x] = \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right),$$

La Figura 4 ilustra al filtro Gaussiano.

Para evitar el cambio de las magnitudes en la imagen de salida, es usual normalizar el resultado de la convolución, con la magnitud acumulada de todos los coeficientes del filtro, de modo que:

$$(U * F)[x, y] = \frac{1}{\alpha} \sum_{u=x-\frac{(K-1)}{2}}^{x+\frac{(K-1)}{2}} \sum_{v=y-\frac{(L-1)}{2}}^{y+\frac{(L-1)}{2}} U[u, v] F[x-u, y-v]. \quad (2)$$

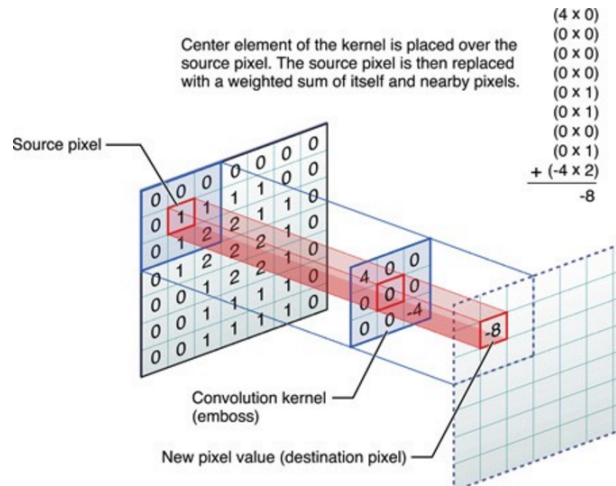


Figura 3: Ilustración de la convolución.

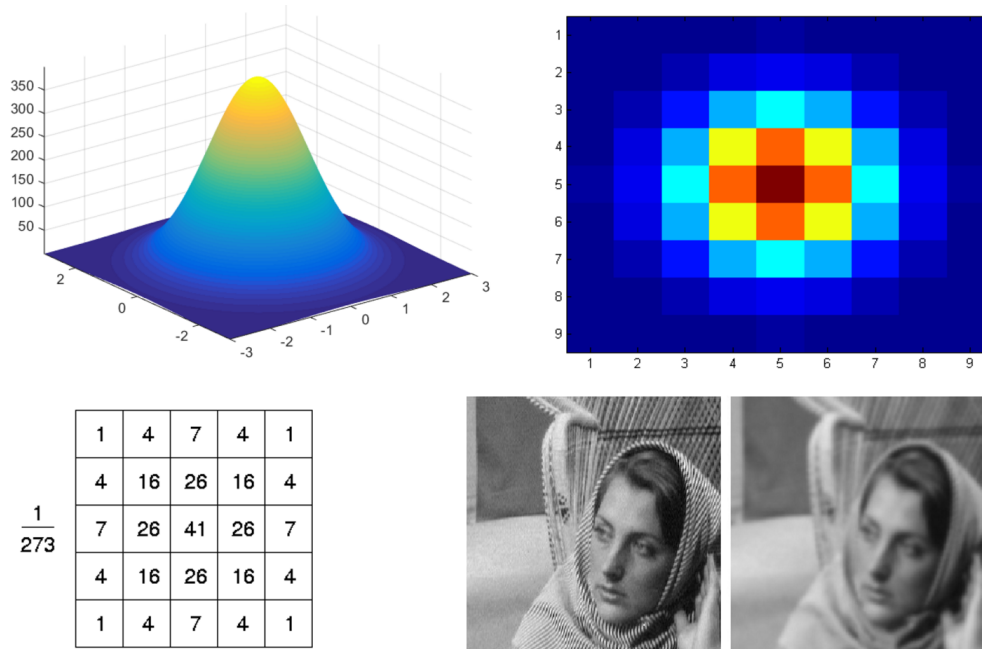


Figura 4: El filtro Gaussiano en distintas visualizaciones, y el resultado de la convolución de una imagen con tal filtro.

con

$$\alpha = \sum_{u=0}^K \sum_{v=0}^L F[u, v].$$

El filtro Gaussiano presenta un efecto indeseado; la degradación de los bordes, lo cual se ilustra en la Figura 4. Para evitar la degradación de los bordes, se utilizan filtros no lineales como por ejemplo el filtro bilateral. Tal filtro, además de pesar el dominio espacial con una función Gaussiana, pesa el dominio de la intensidad con otra función Gaussiana, de modo que los pixeles muy distintos al pixel en el que está centrado la ventana deslizante, obtienen un peso más bajo, evitando la degradación de los bordes.

El filtro bilateral [1] implementa una función dependiente de la entrada U :

$$F[x, y] = \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right) \exp\left(-\frac{(U(x, y) - U(x, y))^2}{2\sigma_r^2}\right).$$

Como se puede observar, el filtro bilateral define dos parámetros, σ_r y σ_s , los cuales controlan el ancho de la campana Gaussiana en el dominio de la intensidad y del espacio, respectivamente. La Figura 5 muestra el efecto de cada parámetro.

Otro problema recurrente en el procesamiento de imágenes biomédicas es el contraste pobre. Para mejorarlo, se usan técnicas como la ecualización de histogramas [2]. La Figura 6 muestra el resultado de ecualizar el histograma de la imagen, usando el algoritmo de ecualización de histograma con límite (CLAHE).

El resumen de los pasos a implementar en el presente proyecto son los siguientes:

1. Mejorar el contraste usando ecualización de histogramas. Puede utilizar la función correspondiente en OpenCV <http://docs.opencv.org/java/2.4.9/org/opencv/imgproc/CLAHE.html>.
2. Implementar el filtro Gaussiano y posibilitar su uso con distintos parámetros.
3. Implementar el filtro bilateral y posibilitar su uso con distintos parámetros.

2.1. Métricas para la eliminación del ruido: Relación señal a ruido

Existen distintas métricas para medir la cantidad de ruido presente en una señal. La relación pico señal-ruido (*peak signal to noise ratio*) es muy utilizada en Ingeniería Eléctrica, y la medición de las ondas sonoras. Se basa en el logaritmo de base 10 para representar la relación entre la potencia eléctrica y la potencia de la señal (para bajas potencias eléctricas, un cambio tiene mayor efecto en la

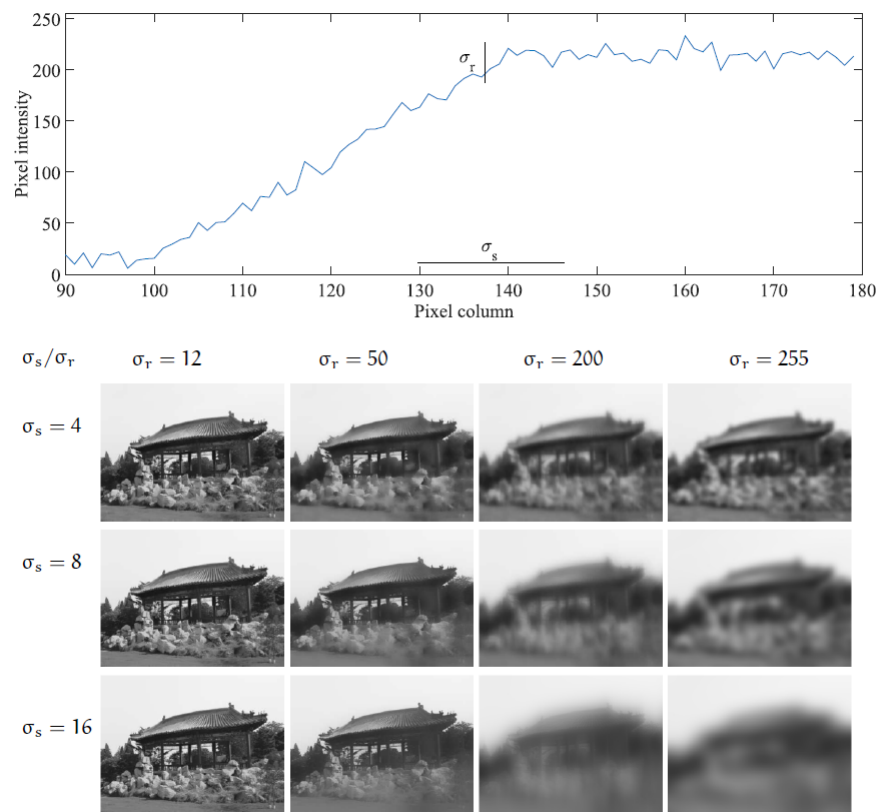


Figura 5: Efecto de los parámetros del filtro bilateral.

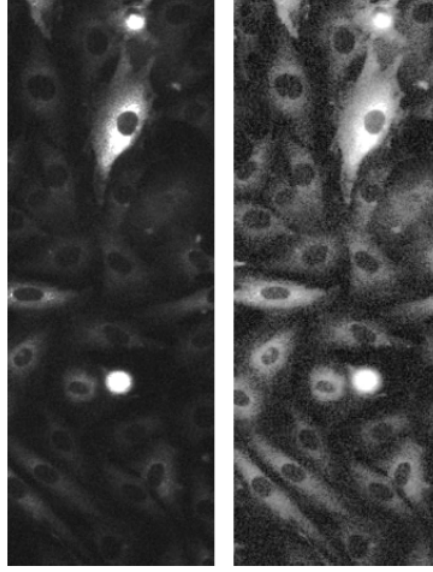


Figura 6: Ecualización de histograma.

potencia de la señal que en altas potencias eléctricas). Su fórmula viene dada por:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) = 20 \log_{10} \left(\frac{\text{MAX}}{\sqrt{\text{MSE}}} \right) = 20 \log_{10} (\text{MAX}) - 10 \log_{10} (\text{MSE}),$$

donde MAX se refiere al máximo valor posible de la señal, por ejemplo, para imágenes a escala de grises $\text{MAX} = 255$ y el error medio cuadrático (*mean squared error*) de la diferencia entre la señal original, sin contaminar por ruido $U \in \mathbb{R}^{m \times n}$ (la imagen de entrada), y la imagen Y contaminada por ruido:

$$\text{MSE} = \frac{1}{m n} \sum_{i=1}^m \sum_{j=1}^n (U(i, j) - Y(i, j))^2.$$

La relación $\frac{\text{MAX}^2}{\text{MSE}}$ asegura que el argumento del logaritmo sea siempre mayor que 1, pues el máximo error promedio puede llegar a ser MAX^2 . El logaritmo multiplicado por 10 (de allí el prefijo *deci*) se comporta como lo mostrado en la Figura 7, donde se observa que para bajos valores, un cambio supone una mayor ganancia, y en altos valores, un cambio supone una ganancia menor.

Observe que en caso de existir un alto error medio cuadrático, relación señal a ruido será más baja, de lo contrario, en presencia de poco ruido, la relación señal a ruido es bastante más alta, como lo ilustra la Figura FIGURA.

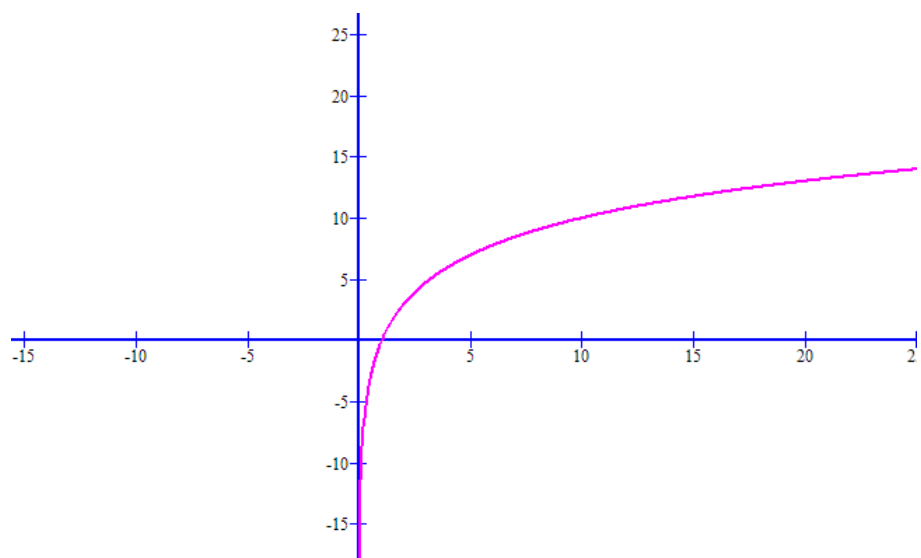


Figura 7: Graficación de la función \log_{10} .

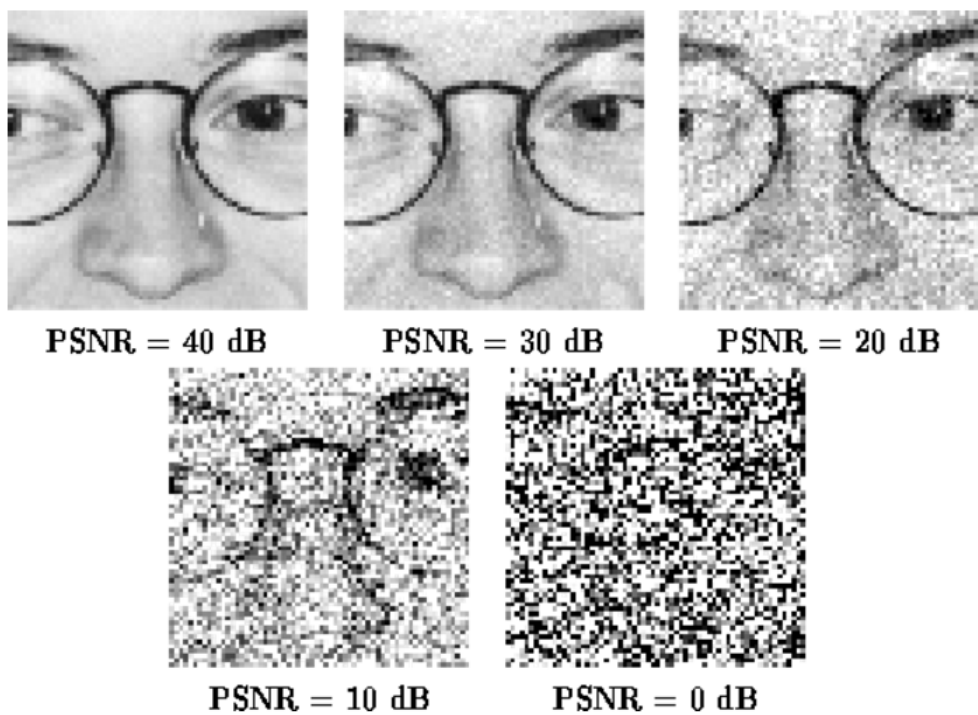


Figura 8: PSNR de distintos valores.

3. Requerimientos de la aplicación

La aplicación debe proveer una interfaz gráfica de usuario amigable, que permita (por orden de prioridad):

1. Cargar una secuencia de imágenes almacenada en la dirección provista por el usuario.
2. Ecualizar el histograma de la imagen para mejorar el contraste.
3. Filtrar las imágenes usando el filtro Gaussiano y el filtro Bilateral, y mostrar y descargar las imágenes resultantes.
 - a) Se debe permitir la personalización de los parámetros de los filtros, como el tamaño de ventana y las desviaciones estándar.
4. Permitir la navegación entre imágenes y resultados.
5. Visualizar el tiempo en ejecutar los lotes de imágenes.
6. **Métrica de exactitud en los resultados:** Calcular la relación pico señal a ruido (*Peak signal to noise ratio*) usando una imagen sin contaminar con ruido, y otra contaminada con ruido artificial, para medir la mejora de la calidad de la imagen.
 - a) Opcionalmente implemente una métrica cualitativa, el *method to noise*, como al diferencia absoluta de la imagen original sin contaminar, y la filtrada.

4. Implementación

Para el diseño e implementación del proyecto, es necesario usar la metodología de desarrollo Scrum, usando alguna herramienta para gestión de proyectos en línea como *zoho projects*. Como herramientas de diseño, se recomienda usar *StarUML*, *Altova*, *Visio*, etc. Para promover el atributo de mantenibilidad del proyecto, al menos un patrón de diseño debe ser implementado.

Respecto a la codificación del proyecto, es muy recomendado el uso del lenguaje Java, usando Eclipse como ambiente de desarrollo, pues existen herramientas y *plugins* relacionados con el cálculo de métricas y gestión de la calidad del proyecto, lo cual será un rubro importante a evaluar. **Los distintos estándares y métricas se aplicarán a lo largo del curso.** Es conocido el soporte de Java con OpenCV además, librería de algoritmos de visión por computador y aprendizaje automático.

La documentación final del proyecto debe incluir todos los estándares implementados a lo largo del curso, incluyendo los realizados en tareas, de manera consistente. Es el obligatorio el uso de LaTeX o algún editor basado en esa tecnología como LyX para la documentación del proyecto. Para facilitar la edición colaborativa, se recomienda el uso de la herramienta *Overleaf*.

Referencias

- [1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 839–846.
- [2] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*. Academic Press Professional, Inc., 1994, pp. 474–485.