
	Computación	Docente: Vladimir Robles Bykbaev
	VISIÓN POR COMPUTADOR	Período Lectivo: Marzo – Agosto 2024

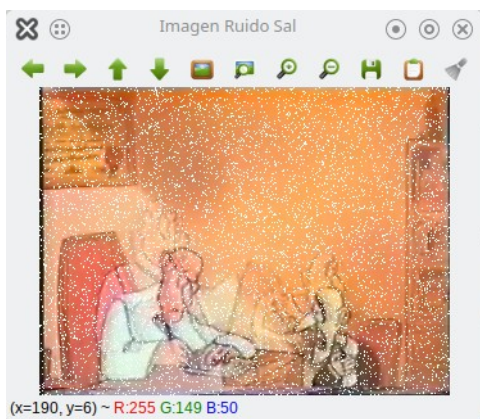
			FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES		
CARRERA: COMPUTACIÓN			ASIGNATURA: VISIÓN POR COMPUTADOR		
NRO. PRÁCTICA:	2-1	TÍTULO PRÁCTICA: Reconocimiento de formas en base a detección de bordes, binarización por umbral, aplicación de filtros y operaciones morfológicas.			
OBJETIVO: Reforzar los conocimientos adquiridos en clase sobre la aplicación de filtros para reducción de ruido y kernels para la detección de bordes, operaciones morfológicas y normalización del histograma.					
INSTRUCCIONES:		<ol style="list-style-type: none">1. Revisar el contenido teórico del tema2. Profundizar los conocimientos revisando los libros guías, los enlaces contenidos en los objetos de aprendizaje y la documentación disponible en fuentes académicas en línea3. Deberá desarrollar un conjunto de <i>scripts</i> y programas que permitan realizar el procesamiento digital de imágenes, ecualización de histograma, detección de bordes y llevar a cabo mejora de imágenes médicas empleando operaciones morfológicas y aplicación de filtros.4. La práctica se subdividirá en tareas específicas que guardarán relación con la representación de varios conceptos como: reducción de ruido, ecualización de histograma, operaciones morfológicas (dilatación y erosión) y detección de bordes.			
ACTIVIDADES POR DESARROLLAR					

<ul style="list-style-type: none"> • Parte 1. Desarrollar una aplicación de escritorio que permita realizar la detección de movimiento usando la técnica de substracción adaptativa de fondo (Adaptive Background Subtraction). Para ello, debe tomar en consideración lo siguiente: <ol style="list-style-type: none"> 1. Programar un <i>método</i> que permita detectar el movimiento que se detecta en vídeos capturados de flujos de stream (<i>cam live</i>, cámaras remotas en vivo) a través de una aplicación de escritorio usando OpenCV/C++, de forma similar a cómo se hizo en clase. Algunos ejemplos de estas cámaras son los siguientes (Ud deberá buscar su propia <i>cam live</i>): <ul style="list-style-type: none"> • Péndulo de Foucault del Instituto Kirchoff para la Física: <ul style="list-style-type: none"> ◦ Stream HTTP: http://pendelcam.kip.uni-heidelberg.de/mjpg/video.mjpg ◦ Página web: http://pendelcam.kip.uni-heidelberg.de/ • Cámara en vivo de Kuroshio Honjin <ul style="list-style-type: none"> ◦ Stream HTTP: http://honjin1.miemasu.net/nphMotionJpeg?Resolution=640x480&Quality=Standard ◦ Página web: http://honjin1.miemasu.net/live/index.html?Language=11&ViewMode=pull 2. Programar una función que permita detectar movimiento y mostrar los FPS en el vídeo original y los filtros u otras operaciones que aplique en los demás espacios disponibles, como se puede ver en la Ilustración 1. Para realizar esta tarea, deberá abrir un flujo de vídeo de Internet.



Ilustración 1. Ejemplo donde se visualiza un vídeo capturado del péndulo de Foucault y la detección de movimiento (resta movimiento).

3. Aplique distintos filtros que permitan mejorar los problemas de iluminación vistos en clase. Para ello debe probar la ecualización de histograma, el método CLAHE y **uno más que Usted investigue**.
4. Explicar cómo funciona la técnica que ha investigado en comparación con la ecualización de histograma y el método CLAHE.
5. Puede usar como referencia el tutorial “**Motion Detection Techniques (With Code on OpenCV)**” que explica 4 formas de detectar el movimiento:
 - Tutorial:
 - <https://medium.com/@abbessafa1998/motion-detection-techniques-with-code-on-opencv-18ed2c1acfaf>
 - GitHub:
 - <https://github.com/safaabbes/Motion-Detection-Techniques-using-OpenCV>
- **Parte 2.** Desarrollar **un programa de escritorio** que permita generar ruido de sal y pimienta y aplicar filtros para reducir dicho ruido. Para ello, deberá llevar a cabo:
 1. Programar un *método* que genere un porcentaje de ruido de sal o pimienta en un **video a color**, considerando las dimensiones del mismo. Se deberá poder ingresar un porcentaje de ruido a través de dos *trackbars* (uno para sal y otro para pimienta).




(a)



(b)

Ilustración 1. Ejemplo de una imagen en la que se ha generado ruido de sal (a) y pimienta (b)

	Computación	Docente: Vladimir Robles Bykbaev
	VISIÓN POR COMPUTADOR	Período Lectivo: Marzo – Agosto 2024

- Programar una función para aplicar los siguientes filtros: mediana, *blur*, Gaussiano, probando con diferente tamaño de máscara.
- Compare los resultados obtenidos por cada filtro y **muestre el resultado en una sola ventana (usando el método *copyTo()* de OpenCV)**, y reflexione cuál ha obtenido mejores resultados.
- Aplicar al menos 2 algoritmos de detección de bordes y comparar los resultados de usar o no filtros de suavizado.

• **Parte 3.** Desarrollar **un programa de escritorio** que permita aplicar operaciones morfológicas para mejorar la calidad de imágenes médicas, para ello deberá realizar las siguientes tareas:

- Seleccionar 3 imágenes médicas a las que se les aplicarán las operaciones morfológicas. Las imágenes deben estar en escala de grises y deben corresponder a radiografías, angiografías, TACs, etc.
- Aplicar las siguientes operaciones sobre las imágenes, probando al menos 3 tamaños de máscaras (de tamaño aproximado de 37x37, como se sugiere en el artículo “*Using morphological transforms to enhance the contrast of medical images*”¹):
 - Erosión
 - Dilatación
 - Top Hat*
 - Black Hat*
 - Imagen Original + (*Top Hat* – *Black Hat*)
- Compare los resultados de la imagen original con respecto a las que tienen operaciones morfológicas y reflexione sobre su nitidez y si mejora la posibilidad de observar mejor los objetos, como se aprecia en las Ilustraciones 2 y 3 (las imágenes a las que se han aplicado estas operaciones se han obtenido de Hamid Hassanpour, Najmeh Samadiani, S.M. Mahdi Salehi, 2015):

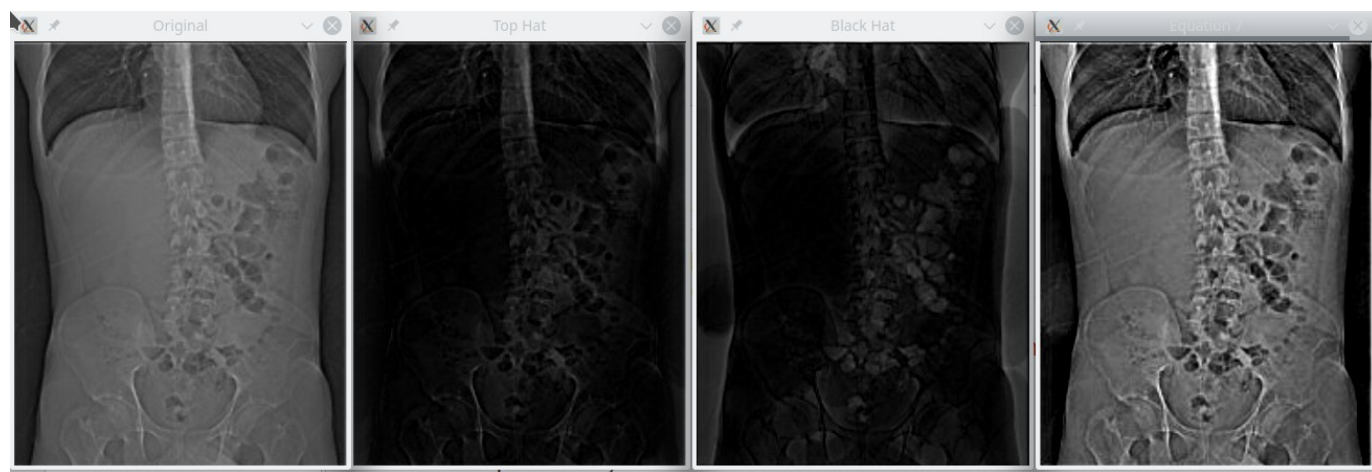


Ilustración 2. Ejemplo una imagen de un *CT Scan* de tórax al que se han aplicado las operaciones c), d) y e) indicadas previamente.

¹<https://www.sciencedirect.com/science/article/pii/S0378603X15000054#e0005>

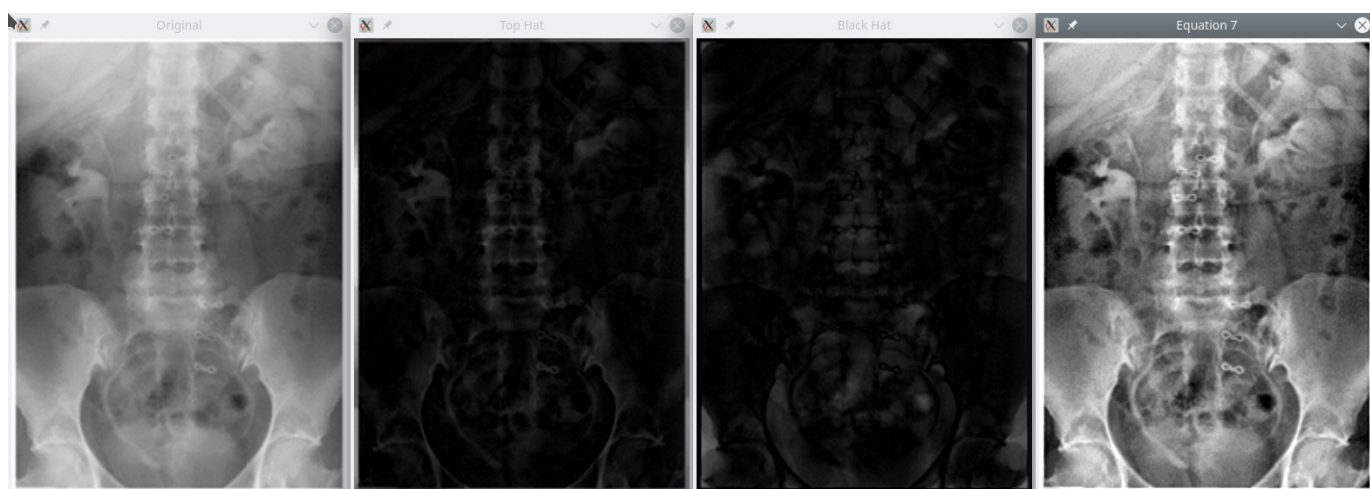


Ilustración 3. Ejemplo una imagen de un *Nephrogram* al que se han aplicado las operaciones c), d) y e) indicadas previamente.

RESULTADO(S) OBTENIDO(S):

Obtiene imágenes a las que se ha realizado procesamiento morfológico, ecualización de histograma, suavizado y detección de bordes, tanto estáticas como en vídeo.

CONCLUSIONES:

Los estudiantes comprenden los principales aspectos relacionados con las operaciones morfológicas (apertura, cierre, erosión, dilatación, *Top-Hat*, *Black-Hat*), la ecualización de histograma y los elementos matemáticos usados en los filtros/*kernel*s tanto para suavizado como para detección de bordes.

RECOMENDACIONES:

- Revisar la información proporcionada por el docente previo a la práctica.
- Haber asistido a las sesiones de clase.
- Consultar con el docente las dudas que puedan surgir al momento de realizar la práctica.

Docente / Técnico Docente: Ing. Vladimir Robles, Bykbaev

Firma: _____