

# PROCESAMIENTO DIGITAL DE IMÁGENES

## Laboratorio 2: Segmentación por histograma

Prof. Sebastián Godoy ([segodoy@udec.cl](mailto:segodoy@udec.cl))  
Departamento de Ingeniería Eléctrica  
Universidad de Concepción

11 de abril de 2016

<b>Fecha publicación:</b> 11 de abril de 2016 <b>Fecha de entrega:</b> 25 de abril de 2016
--

**Tareas a cumplir:** Para este laboratorio Ud. debe completar las siguientes tareas:

1. Capturar imágenes de un texto escrito a mano bajo distintas condiciones
2. Mostrar el histograma de cada una de estas imágenes
3. Identificar “manualmente” las zonas de interés en los histogramas que diferencian texto del resto de la imagen capturada
4. Implementar algoritmo  $k$ -means visto en clases
5. Segmentar por histograma el texto del fondo de forma manual y mediante el algoritmo  $k$ -means
6. Generar una imagen binaria que identifique solamente el texto (texto en blanco y fondo en negro), comparando resultados
7. Sus codigos deben estar disponibles en GitHub para la revisión de sus resultados.

**Evaluación:** Conforme a lo estipulado en el syllabus Ud. debe presentar sus resultados en clases así como también un informe ejecutivo para la evaluación del presente laboratorio. [Para este laboratorio se pedirá una demostración en vivo de los resultados más la explicación de la metodología utilizada.](#)

**Formato:** Su informe ejecutivo debe tener el nombre `Apellido1_Apellido2.pdf` para facilitar el manejo de los informes. Enviar al correo del profesor ([segodoy@udec.cl](mailto:segodoy@udec.cl)) y CC. al ayudante ([priquelmej@udec.cl](mailto:priquelmej@udec.cl))

### 1. Captura de imágenes

Evidentemente, para esta parte debe capturar la imagen con SimpleCV y utilizando su cámara web asignada. La idea es que el análisis sea hecho sobre imágenes representativas. Así, en la demostración en clases de sus resultados, Ud. asegurará su éxito.

Para estudiar manualmente el histograma, se recomienda capturar varias imágenes de texto, por ejemplo sobre papel blanco, sobre papel cuadriculado y sobre papel con líneas. Podría testear con papel

de distintos colores. Para todos los casos, guarde la imagen correspondiente en su tarjeta para analizarla manualmente después.

## 2. Histogramas e identificación de zonas de interés

Convierta sus imágenes en escala de grises para su análisis. Para ello, si su imagen es llamada `img`, entonces su versión en escala de grises se obtiene mediante el comando `imgGray = img.greyscale()`. Para sus imágenes en escala de grises Ud. puede obtener su histograma mediante `hist = imgGray.histogram()` y visualizarlo mediante `plot(histogram)`. Más detalles de como usar la función la puede encontrar en la página de referencia de SimpleCV: [Documentación SimpleCV](#).

Puede que bajo algunas condiciones de iluminación, algún canal (rojo, verde o azul) de la imagen le permita una mejor segmentación que al hacerlo sobre la escala de grises. Para ello, separe la imagen capturada en sus canales mediante `(red, green, blue) = img.splitChannels(False)`. Luego visualice el histograma individualmente para cada caso. Por ejemplo, `red_histogram = red.histogram(255)` y luego `plot(red_histogram)`. Para mayores referencias consulte el [tutorial de SimpleCV](#).

Tome una decisión si hará su segmentación sobre la escala de grises o sobre uno de los canales de la imagen a color. Justifique esta decisión en su informe ejecutivo.

Ahora, una vez identificadas las zonas de interés en el histograma (independiente de la imagen con la que Ud. las obtuvo), Ud. puede generar la imagen segmentada mediante el comando `binarize`. Conforme a la página de la documentación, se debe especificar el umbral. Si el umbral no se da, entonces se utiliza por defecto el algoritmo de Otsu visto en clases. Probablemente este algoritmo le puede servir para obtener alguna de las imágenes solicitadas.

Genere las máscaras para identificar el texto respecto al fondo de la imagen. Genere máscaras para identificar solamente el fondo y máscaras para identificar las líneas de la cuadrícula, si aplica. Muestre estas imágenes en su resumen ejecutivo.

## 3. Clustering mediante $k$ -means y segmentación

El algoritmo  $k$ -means ya fue implementado para Ud. en el paquete `sklearn` de Python. Para instalarla, siga los pasos en el siguiente tutorial (busque “Machine Learning Environment Setup on Raspberry Pi 2” si el link no funciona): [Tutorial SKLearn](#). Otra forma es mediante la instalación del paquete “Anaconda” (disponible en [este link](#)) para luego instalar `sklearn` mediante `conda install scikit-learn`. Cualquiera sea la forma en que lo instale, documentarlo en el resumen ejecutivo. Una vez instalado el programa, siga el ejemplo dado en la página [de PyImageSearch](#) para realizar la segmentación por histograma usando el algoritmo  $k$ -means.

## 4. Código final e informe ejecutivo

Para el código final a entregar, se espera que Ud. logre implementar un código en python que capture una imagen de la webcam, utilizando los valores pre-calculados segmente el histograma de forma “manual” y mediante el algoritmo  $k$ -means para mostrar ambos resultados como imagenes binarias que identifiquen el texto sobre el fondo de la imagen. Su código debe estar disponible en su repositorio GitHub para ser revisado. Asegúrese de subir cada modificacion del código para poder evaluar parcialmente sus logros al momento de entregar (esto resulta beneficioso si Ud. no logra todos los objetivos del laboratorio). Para su informe ejecutivo, Ud. incluya las imágenes obtenidas más representativas y las secciones correspondientes: Problemas encontrados, soluciones implementadas y resultados. Si se pasa de 2 páginas está bien para este laboratorio. Tiene 14 días para resolver este laboratorio. Si mantiene su GitHub actualizado con sus avances parciales, se podría solicitar mayor plazo para la entrega previa evaluación del avance por parte del profesor.