

# CC5508: Procesamiento y Análisis de Imágenes

## Tarea 2: Mejoramiento del Contraste de Imágenes usando *Level Sets*

Profs. José M. Saavedra / Mauricio Cerda  
Ayudante: Camila Álvarez

03 abril 2017

### 1. Objetivo

El objetivo de esta tarea es implementar y analizar un método local basado en *level-sets* para mejorar el contraste de imágenes en escala de grises.

### 2. Descripción

Mejorar el contraste de imágenes consiste en modificar los valores de los píxeles de modo que haya mayor variación entre las tonalidades de los objetos presentes y así podamos distinguirlos con mayor claridad. Justamente, el objetivo de este proceso es poder distinguir fácilmente los objetos contenidos en la imagen. Uno de los métodos clásicos es la *ecualización del histograma*, que trata de obtener una imagen con distribución homogénea de los valores de gris presentes en la imagen de entrada. Idealmente, en la imagen ecualizada cada valor de gris tiene la misma probabilidad de ocurrir. En clase se demostró que podremos ecualizar una imagen usando la siguiente expresión:

$$G(x) = accum(I(x)) * (q_{max} - q_{min}) + q_{min} \quad (1)$$

donde,  $I$  es la imagen de entrada,  $G$  es la imagen resultante,  $x$  es la posición de un píxel  $x = (x, y)$ ,  $accum(\cdot)$  es la función de acumulación normalizada, y  $(q_{max}, q_{min})$  representa el máximo y mínimo valor del espectro de tonos de gris en el que queremos que varíe la imagen resultante.

Una variación del método clásico consiste en aplicar el proceso de ecualización en forma local, es decir por sub-regiones de la imagen, tratando de resaltar los objetos presentes en la región. Una forma de definir tal localidad es a través de *level sets*. Un *level set* define un conjunto de pixels cuyo valor de gris varía entre dos tonos  $(\lambda_1, \lambda_2)$ . Sea  $L_{i,j}$  un *level set* definido de la siguiente manera:

$$L_{i,j} = \{x : \lambda_{i,j} \leq I(x) < \lambda_{i,j+1}\}, \quad (2)$$

donde,  $i = 0, \dots, T$  indica una determinada iteración,  $j = 0 \dots 2^i - 1$  indica un determinado *level set* para la iteración  $i$ . Además, sea  $\lambda_{i,j}$  el límite de un *level set* para un  $i$  y  $j$  determinado, el cual se define como:

$$\lambda_{i,j} = 256 * \frac{j}{2^i}. \quad (3)$$

Para facilitar la explicación diremos que  $L_{i,j} = \Lambda(\lambda_{i,j}, \lambda_{i,j+1})$ . Así, la iteración  $i = 0$  contiene un solo *level set* definido por  $\Lambda(0, 256)$ , mientras que la iteración  $i = 1$  tiene 2 *level sets*,  $\Lambda(0, 128)$  y  $\Lambda(128, 256)$

Además, sea  $O_{i,j}$  el conjunto de los diferentes componentes conectados de  $L_{i,j}$ , en donde  $O_{i,j}^r$  es el componente  $r$ -ésimo de  $L_{i,j}$ .

Usando las definiciones anteriores, el algoritmo de ecualización local basado en *level sets* consiste en iterar desde  $i = 0$  hasta  $T$  y para cada  $i$  obtener sus diversos *level sets*  $L_{i,j}$  con  $j = 0 \dots 2^i - 1$ . Para cada *level set*, se obtienen los componentes conectados  $O_{i,j}$ , cada componente  $O_{i,j}^r$  se ecualiza por separado. La ecualización se realiza siguiendo la ecuación 1, en donde  $q_{max}$  y  $q_{min}$  dependerán del *level set* que se esté procesando. La unión de todos los componentes ecualizados formará la imagen resultante  $G_i$ , para la iteración  $i$ -ésima. Por lo tanto, en cada iteración el resultado  $G_i$  será una imagen con contraste mejorado. Es importante notar que  $G_0$  es la imagen ecualizada globalmente. Un ejemplo de este proceso se muestra en la Figura 1

En esta tarea se pide realizar un programa en *python* que implemente el algoritmo de ecualización por *level sets*. EL programa debe ser ejecutado por consola (entrada estándar). El programa debe ser nombrado como *tarea\_2.py* y debe recibir los siguientes argumentos:

- **-image**, indica la imagen de entrada. El procesamiento se realiza sobre una imagen en escala de grises. Por lo tanto, es importante realizar una transformación a grises si la imagen viene en otro formato. El resultado de la ecualización deberá ser guardado en un archivo nombrado igual al nombre de la imagen entrada pero con el sufijo *\_eqls\_<t>* y en formato png.

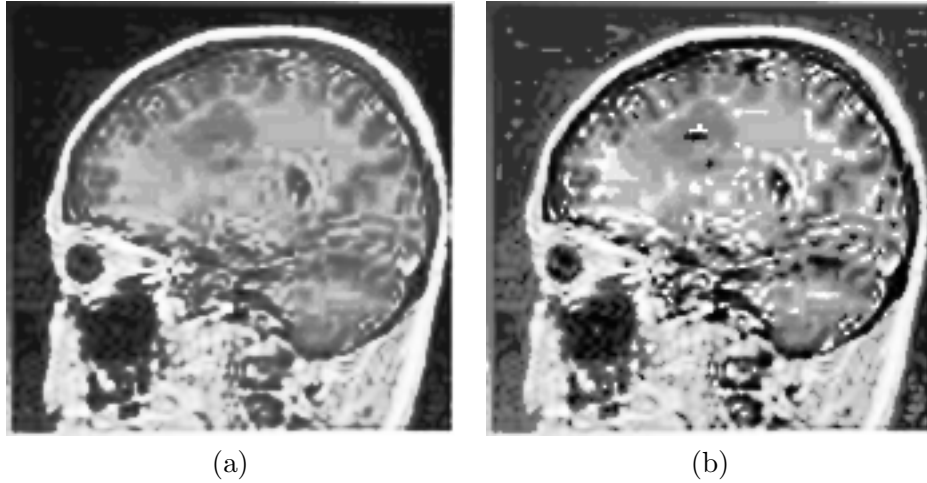


Figura 1: (a) Ecualización Global (b) Ecualización Local por Level Sets

- $-t$ , indica el número de iteraciones.

Ejemplo

```
$ python tarea_2.py --image imagen.png --t 3
```

Salida

```
imagen_eqls_3.png
```

Deberán entregar por u-cursos el código fuente junto con un informe (tipo paper) que describa el trabajo realizado junto con ejemplos de uso y un análisis de los resultados. Por ejemplo, podrían analizar el impacto de  $T$  en el proceso y explicar cuál sería el máximo valor para  $T$ .

### 3. Esquema de Informe

1. **Abstract o Resumen:** es el resumen del trabajo.
2. **Introducción:** se describe el problema y el contexto. Aquí deben definir el término *mejoramiento de contraste* y los métodos clásicos. (10 %)
3. **Desarrollo:** se describe el diseño e implementación del programa. (40 %)
4. **Resultados Experimentales y Discusión:** se deben presentar los resultados y hacer un análisis de los mismos. (40 %)
5. **Conclusiones** (10 %)

## 4. Restricciones y Condiciones

1. **Todo el código del algoritmo debe ser implementado por el mismo estudiante, incluyendo la obtención de los componentes conectados.**
2. NO se aceptan tareas sin informe.
3. NO hay atrasos.
4. La tarea es individual.
5. Poner mucho esfuerzo en la redacción del informe.
6. La implementación se realizará en Python.

## 5. Entrega

La entrega se realiza por u-cursos hasta el domingo 16 de abril, 2017, 23:50 hrs. Se debe incluir:

1. Código fuente (en Python)
2. Informe

## 6. Consultas

Escribir a la ayudante Camila Álvarez [camila.alvarez.i@ug.uchile.cl].