

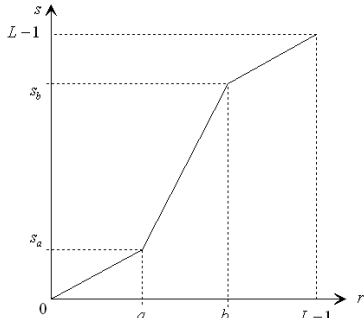
Bloque 1: Operadores puntuales

ENUNCIADO

Ejercicios Básicos

Ejercicio 2b: ajuste de contraste por tramos (continuación del ejercicio 2a).

Según se ha visto en las explicaciones teóricas, la expresión general de una operación de ajuste de contraste por tramos rectos viene dada por:

$$s = \begin{cases} \alpha \cdot r & 0 \leq r < a \\ \beta \cdot (r - a) + s_a & a \leq r < b \\ \gamma \cdot (r - b) + s_b & b \leq r < L \end{cases}$$


La figura presenta el aspecto de la transformación para unos valores concretos de sus parámetros, que modifica por tramos los valores de los píxeles de la imagen original, comprimiendo unos rangos $[0, a)$ y $[b, L-1]$ en este ejemplo) y expandiendo otros $[a, b)$. Los factores de expansión y compresión vienen definidos por los parámetros:

$$\alpha = s_a/a, \beta = (s_b - s_a)/(b - a), \gamma = (L - 1 - s_b)/(L - 1 - b)$$

La finalidad de este ejercicio es la de mostrar la aplicación principal de esta transformación: el ajuste o realce de una imagen poco contrastada. Para poder observar el efecto tanto visualmente como en su histograma, utilice la primera opción para la realización de operadores puntuales.

Cargue la imagen de prueba `edificio_bc_512.bmp`¹. Observe si se trata o no de una imagen indexada, si por lo tanto viene o no acompañada de una VLT y, en este caso, cuál es su número de niveles, L . Represente la imagen y su histograma e indique, observando éste, cuál es la característica que provoca el que una imagen tenga un bajo contraste:

Aplique la transformación de ajuste de contraste con parámetros:

$$a = 80, b = 160, s_a = 30, s_b = 220$$

A la hora de obtener los L valores $s_k = T(r_k)$, tenga en cuenta que éstos deben ser números enteros, ya que representan índices de la VLT (para forzarlo, haga $s = \text{round}(s)$ tras definir la transformación). Represente la imagen procesada, su histograma y la función de transformación.

¹ Descárguese esta imagen del Moodle de la asignatura.

Indique, observando todos ellos, cuál ha sido el efecto visual de la transformación sobre la imagen, cómo se traduce este efecto en su histograma, y cómo la función de transformación consigue el efecto deseado:

Ejercicio 3b: ecualización de imágenes en el espacio RGB

Descárguese del Moodle de la asignatura la imagen `7colores_uneq.bmp` aplique la técnica de ecualización de histograma al máximo rango disponible como se describe en el ejercicio 3a con dos aproximaciones distintas.

Opción 1: Ecualice independientemente cada canal RGB.

Opción 2: Transforme la imagen al espacio de color HSV (tono, saturación y valor), para ello haga uso de la función `rgb2hsv` y realice el proceso de ecualización sólo sobre el tercer canal de la imagen transformada (la componente V). Finalmente sustituya la componente V ecualizada en la imagen HSV y vuelva al espacio de color RGB mediante la función `hsv2rgb`. Durante este proceso, quizás tenga que tener en cuenta factores de normalización adicionales y conversiones `double()` a `uint8()`. Le recomendamos consultar la ayuda de Matlab sobre la función `hsv2rgb`:

`doc hsv2rgb.`

Compare los resultados obtenidos para las dos opciones en función de las imágenes procesadas obtenidas y de sus histogramas. Copie los resultados parciales del proceso en su memoria.

Operadores puntuales básicos

Negativo de una imagen en color

El negativo o inversión de una imagen es una operación puntual, es decir que afecta a cada pixel, independientemente de su vecindad. Para una imagen en escala de grises que puede tomar L posibles valores o niveles de intensidad, es decir con rango $r_k \in [0, L-1]$. El negativo de esa imagen viene definido por la transformación:

$$s_k = T(r_k) = (L-1) - r_k$$

, con $s_k \in [0, L-1]$. Es decir, puede entenderse como una inversión del eje de intensidades.

Si la imagen es en color, por ejemplo en el modelo RGB, el negativo de la imagen se calculará realizando el negativo de cada banda independientemente.

Si cada banda tiene un rango r_k^i :

$$\mathbf{r}_k = [r_k^R \quad r_k^G \quad r_k^B], \quad 0 \leq r_k^i \leq (L-1)$$

, el negativo se calculará como:

$$\mathbf{s}_k = [s_k^R \quad s_k^G \quad s_k^B], \quad 0 \leq s_k^i \leq (L-1)$$

, donde:

$$s_k^i = T(r_k^i) = (L-1) - r_k^i$$

Ejercicio 4a: obtención del negativo

Cargue y visualice la imagen `7_colores.jpg`² así como su histograma y el histograma de cada uno de sus tres canales RGB.

Calcule el negativo de cada canal independientemente según lo comentado en el apartado anterior, mezcle los tres negativos para crear una nueva imagen `ima_p` que contenga el negativo de la imagen original `ima`.

Obtenga y copie en su memoria las imágenes y los histogramas involucrados en el proceso. Observe y describa cuál es el efecto de esta transformación.

Ejercicio 4b: observación de la after-image

Obtenga la *after-image* negativo a partir del resultado obtenido en el apartado anterior. Para ello, puede copiar las siguientes líneas Matlab—**recuerde respetar la nomenclatura que haya utilizado en el ejercicio anterior**—:

```
%% after image
close all;
time = 30;
[r,c,~] = size(ima);
imgray = rgb2gray(ima);
hFig = figure('units','normalized','outerposition',[0 0 1 1]);
imshow(ima_p); hold on, plot(ceil(c/2),ceil(r/2),'kx','MarkerSize',15,'LineWidth',3);
set(hFig, 'outerposition',[0 0 1 1],'Color',[0.7 0.7 0.7])
title(sprintf('Mire fijamente la X durante %d sec',time));
pause(time);
hold on, imshow(imgray);
```

Para percibir mejor la ilusión tenga paciencia y observe la 'x' central de la pantalla procurando parpadear lo menos posible. Cuando la imagen cambie procure mantenerse sin parpadear y recorra muy despacio la imagen que aparece en pantalla. La ilusión óptica se aprecia especialmente en la parte superior izquierda de la imagen, correspondiente al cielo de la fotografía.

Explique el resultado observado, documéntese sobre el fenómeno. Pruebe a experimentar con diferentes imágenes.

Umbralización

Como han visto en teoría, la umbralización de imágenes es un caso particular del recorte, cuyo objetivo principal es la segregación, separación o segmentación de los dos niveles o clases de una imagen bimodal.

Para umbralizar una imagen podemos partir de conocimientos previos: por ejemplo, número y distribución de objetos o probabilidad de los objetos respecto de la del fondo, o bien, por el

² Descárguese esta imagen del Moodle de la asignatura

contrario, no disponer de ninguna información previa sobre la imagen ni sobre sus componentes.

En este caso, el proceso suele modelarse automáticamente partiendo de una estimación inicial del umbral y alterando el umbral en función de criterios de inter-separación entre clases o/e intra-homogeneidad en cada clase. Este tipo de procesos pueden ser iterativos, y convergen cuando la diferencia entre el umbral anterior y el estimado es menor que un umbral de convergencia prefijado.

En este ejercicio intentaremos umbralizar una imagen en escala de grises que en principio no es bimodal, pero en la cual puede apreciarse una clase diferenciada: **cuerpos celestes** y una clase definida por oposición: **resto**.

Para comprobar la bondad de los métodos descritos a continuación cargue y represente la imagen `cumulo_bala.jpg`³, transfórmela a escala de grises `rgb2gray()`.

Para cada método desarrollado copie y pegue en su memoria la imagen original y su histograma, la imagen umbralizada, su energía y su histograma y comente los resultados observados.

Ejercicio 5a: umbralización global por el método de Otsu

Genere una función MatLab que implemente un esquema de búsqueda exhaustiva del umbral que minimiza la varianza intra-clase, es decir, que fije automáticamente el umbral por el método de Otsu.

Le recomendamos sustituir en el esquema anterior el proceso iterativo por uno exhaustivo y cambiar el proceso de cálculo del umbral según lo visto en clase.

Esta función recibe una imagen en escala de grises (tipo `uint8`) y devuelve una máscara binaria (lógica), es decir, con los píxeles cuyo valor de gris esté por encima del umbral detectado a '1' y el resto a '0'. Ambas imágenes deberán tener un único canal.

```
function ima_th = UmbralizaGlobalOtsu(ima)
```

Nota: Puede comparar sus resultados con la implementación de Otsu de Matlab: `graythresh()`. Observe que la función de Matlab devuelve el umbral en el intervalo [0,1], por lo que tendrá que multiplicar el valor obtenido por 255 para poder comparar los resultados con la función implementada.

³ Descárguese esta imagen del Moodle de la asignatura