|  |  |  |  |
| --- | --- | --- | --- |
| Práctica 4: Operadores puntuales (I) | | Grupo |  |
| Puesto |  |
| Apellidos, nombre |  | Fecha | |
| Apellidos, nombre |  |  | |

El objetivo de esta práctica es presentar al alumno los fundamentos de los operadores puntuales y parte de sus aplicaciones.

Desarrolle cada ejercicio en un fichero de comandos ‘ejercicio\_X.m’ separado. Para conocer el funcionamiento preciso de los comandos que se introducen en este guión, utilice la ayuda de MATLAB. Para evitar posibles interferencias con otras variables o ventanas recuerde incluir siempre las instrucciones clear all y close all al principio de cada fichero de comandos.

Al finalizar la práctica, comprima el documento de observaciones y los ficheros ‘.m’ generados en un único fichero con el nombre ‘**FTDI\_P4\_I\_ApellidosNombre1\_ApellidosNombre2.zip**’, conéctese al sistema de entrega de prácticas de *Moodle* y entréguelo.

**NOTA IMPORTANTE**: En el desarrollo de esta práctica y posteriores se pide realizar operaciones repetidamente, por ejemplo, modificando individualmente píxeles concretos de una imagen. En estos casos se recomienda acudir al uso de estructuras de control (bucles, condiciones, etc.). Adicionalmente, debido a la creciente complejidad de los programas que se le va a pedir desarrollar, se recomienda encarecidamente el uso del depurador o *debugger* de MatLab, cuya funcionalidad está descrita en la ayuda del programa (*User’s Guide - Desktop Tools and Development Environment - Editing and Debugging MATLAB Code*).

# Aplicación de operadores puntuales

La aplicación de un operador sobre una imagen que denominaremos *original* (), da lugar a una nueva imagen que denominaremos *procesada* (). A efectos de esta práctica, asumiremos que un operador puntual queda definido por una transformación *T* que toma como entrada el valor de un píxel de la imagen original para generar un nuevo valor en el píxel homólogo de la imagen procesada:



, donde ×  es el tamaño de la imagen original,  son los valores que toman los píxeles de la imagen original, los que toman los píxeles de la imagen transformada y *L* el número de posibles valores o niveles de intensidad.



Este operador, por lo tanto, se limita a modificar el valor de cada píxel. En conclusión, la transformación llevada a cabo por el operador quedará totalmente definida por la relación entre los valores  de la imagen original y los valores  de la imagen procesada:



Mediante MatLab, podemos diseñar al menos tres estrategias para la realización de operaciones puntuales:

1. Modificación directa de los píxeles de la imagen.
2. Modificación del vector de valores de la imagen.
3. Modificación de la VLT (*Video Lookup Table*).

Comparemos las tres aproximaciones en función de los resultados obtenidos para cada una de ellas y en términos de eficiencia o coste computacional. Nótese que la tercera estrategia sólo podrá realizarse si la imagen de entrada es indexada, es decir tiene mapa asociado VLT.

Sea la transformación deseada:

, para distintos valores de .



Pueden describirse tres estrategias a seguir para realizar esta transformación

## Modificación directa de los píxeles de la imagen (opción 1)

Este es el modo más inmediato y a su vez el más ineficaz de implementar un operador puntual; sin embargo, en ocasiones es el único modo posible de hacerlo. Consiste en modificar uno por uno el valor de todos y cada uno de los píxeles de la imagen original, aplicándoles la transformación *T*.

Consiste en aplicar directamente la transformación sobre las matrices que definen la imagen original (**ima**) y la imagen procesada (**ima\_proc**). Es la opción más directa, pero puede resultar difícil de aplicar si la función de transformación es una función definida por tramos:

ima\_proc=min(ima+c,L-1);

## Modificación del vector de valores de la imagen (opción 2)

Consiste en definir primero el vector de valores y calcular el vector de valores transformados . A continuación, la imagen procesada se obtiene utilizando la original para referenciar los índices del vector de valores transformados.



Esta opción permite definir con relativa sencillez cualquier función de transformación:

r=[0:L-1];

s=min(r+c,L-1);

ima\_proc=s(ima+1); % Sumamos 1 porque el valor 0 corresponde a s(1)

## Modificación del vector de la VLT (opción 3)

Este es el modo más eficaz de implementar un operador puntual, pero sólo es válido en el caso de imágenes indexadas, es decir, imágenes cuyos píxeles referencian posiciones de una VLT.

Sea la VLT de la imagen original. Los valores de sus píxeles serán referencias o posiciones de esta VLT. Por lo tanto, una imagen representada de este modo es posible formalizarla según: .



Dado que la transformación que lleva a cabo un operador puntual sólo afecta a los valores de los píxeles, y éstos están definidos por la VLT, bastará con aplicar la transformación a la VLT.

En conclusión, **la imagen procesada será igual a la imagen original**, pero su VLT, *S*, se habrá modificado:

Para modificar una VLT debe tener en cuenta que en MatLab **cada valor** de una VLT es en realidad un vector fila de tres valores de tipo double, que indican respectivamente la cantidad de las componentes roja, verde y azul del color que representa:

En el caso de VLTs en escala de grises, las tres componentes tienen igual valor.

Para llevar a cabo la transformación de la VLT, deberá aplicarla sobre la terna de valores de cada matriz fila: .

Para obtener resultados equivalentes a las aproximaciones anteriores, escale previamente los valores de cada componente, e.g., , multiplicándolos por (ya que en la VLT toman valor entre 0 y 1, no entre 0 y L-1).

## Ejercicio 1: comparando los resultados de las tres aproximaciones para la realización de operadores puntuales

Cargue y visualice la imagen de prueba Skin\_gray\_bw\_560.tif**[[1]](#footnote-1)** e indique si se trata o no de una imagen indexada, y si por lo tanto viene o no acompañada de una VLT. En este caso, indique cuál es su número de niveles, .

Implemente la transformación definida por:

, para los distintos valores de .

Mediante las tres aproximaciones: modificación directa de los píxeles de la imagen, modificación de los valores de la imagen y modificación de la VLT.

Obtenga y copie en su memoria las doce imágenes procesadas (cuatro funciones variando la por cada una de las tres aproximaciones). Asimismo, obtenga y copie en su memoria las cuatro funciones de transformación (una para cada valor de ).

Observe y describa cuál es el efecto de esta transformación. ¿Existen diferencias en los resultados obtenidos para cada aproximación descrita?

## Ejercicio 2: comparando la eficiencia de las tres aproximaciones para la realización de operadores puntuales

Para estimar el tiempo que tarda MatLab en realizar cada una de las tres aproximaciones descritas, genere un nuevo fichero en el que, mediante un bucle for, se realice 1000 veces una misma transformación mediante una de las tres aproximaciones descritas. Para medir el tiempo introduzca antes del bucle la instrucción tic; (almacena en una variable interna el tiempo actual) y a continuación del end que cierra el bucle la instrucción toc;, que imprimirá por la línea de comandos el tiempo transcurrido en segundos desde que se ejecutó la instrucción tic. Repita la operación para cada una de las tres aproximaciones.

Repita de nuevo el proceso para la imagen ‘Skin\_gray\_bw\_1120.tif’.

Rellene en su memoria una tabla similar a la que se adjunta:

|  |  |
| --- | --- |
| Tiempo estimado con imagen de 560x560 (s) | Opción 1: |
| Opción 2: |
| Opción 3: |
| Tiempo estimado con imagen de 1120x1120 (s) | Opción 1: |
| Opción 2: |
| Opción 3: |

# Modelado de histograma

El histograma de una imagen permite observar cuantitativamente el efecto de los operadores puntuales; de ahí el interés de saber representarlo e interpretarlo. La función que ofrece MatLab para este fin es imhist, a la que hay que pasar como parámetros la imagen en cuestión y su VLT (si esta está disponible).

Puede encontrar una extensa descripción de la función **imhist**, incluyendo parámetros potenciales, requisitos de entrada y ejemplos invocando el comando:

**doc imhist**

, en la consola de Matlab.

Para una visualización completa de los histogramas, se recomienda utilizar el comando **axis tight** después de cada representación de un histograma.

## Estirado de histogramas

El **estirado o normalización** de histograma es un proceso por el cual se cambia el rango de intensidad de los píxeles de la imagen. Aunque la transformación puede no ser lineal, la forma más común de estirar el histograma es aplicando la transformación:

, donde y representan los límites del rango de la imagen original, es decir .

yrepresentan los límites del rango tras el procesado, es decir .

Nótese que, en general, y . Al estar por defecto fuera del rango , para visualizar la imagen escalada, deberá definir el rango de representación en el **imshow** o realizar un casting a uint8.

## Ecualizado o igualación de histogramas

Por otro lado, como han visto en la parte teórica de la asignatura, la **igualación o ecualización de histogram**a es un proceso distinto que, si bien también resulta en el estirado del histograma, tiene como efecto principal el producir una uniformización del histograma, es decir cambiar su forma de manera que el histograma resultante tienda a ser *plano o equi-distribuido*. Este proceso, en general, incrementa globalmente el contraste de la imagen.

Como han visto en teoría, la transformación a realizar para la ecualización de histograma viene definida por la expresión, tomando el histograma normalizado como estimador de la *f.d.p.*

, donde es la altura del histograma en el nivel y .

Para realizar los sumatorios, Matlab dispone de la función **cumsum** , una extensa descripción de esta función, incluyendo parámetros potenciales, requisitos de entrada y ejemplos invocando el comando:

**doc cumsum**

, en la consola de Matlab.

Nótese que el rango de será en general: , por lo tanto, se han de cuantificar los valores de al intervalo de números enteros .

Al estar por defecto fuera del rango , para visualizar las imágenes, deberá definir el rango de representación en el **imshow** o realizar un casting a uint8.

## Ejercicio 3: representando y observando histogramas

Repita el ejercicio 1 para las imágenes Skin\_gray\_bw\_560.tif**[[2]](#footnote-2)** y Skin\_gray\_bw\_1120.tif**[[3]](#footnote-3)**.

Represente y copie en su memoria los histogramas de las ocho imágenes procesadas: resultantes de aplicar sobre las dos imágenes las cuatro transformaciones definidas por los valores de , cualquiera de las dos primeras opciones expuestas en el ejercicio 1 podrá ser utilizada para este fin (modificación directa de los píxeles de la imagen o modificación de los valores de la imagen).

Indique finalmente, a la vista de los resultados de los ejercicios realizados hasta el momento, qué método utilizaría, y por qué, para aplicar un operador puntual.

## Ejercicio 4: estirando y ecualizando el histograma

Cargue y visualice la imagen Skin\_gray\_bc\_560.tif**[[4]](#footnote-4)** así como su histograma. Realice un estirado del histograma con  y . Incluya en su memoria capturas de la imagen original, la imagen estirada y de los histogramas de ambas imágenes.

¿Qué cambios observa entre las imágenes? ¿Y entre sus histogramas? Consigne en su memoria la respuesta a estas preguntas.

A continuación, realice una ecualización de histograma, para facilitar la visualización del efecto de la transformación, recuerde escalar el rango final de la imagen ecualizada al rango **.

Incluya en su memoria capturas de la imagen ecualizada y de su histograma.

¿Qué cambios observa ahora en la imagen y el histograma?

¿Y con respecto a la imagen estirada y a su histograma?

Consigne en su memoria la respuesta a estas preguntas.

## Ejercicio 5: ajuste de contraste por tramos

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

|  |  |
| --- | --- |
|  |  |

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 (y en este ejemplo) y expandiendo otros (). Los factores de expansión y compresión vienen definidos por los parámetros:

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 segunda aproximación (ejercicio 1) de realización de operadores puntuales.

Cargue la imagen de prueba Skin\_gray\_bc\_560.tif**[[5]](#footnote-5)**. 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, . Represente la imagen y su histograma e indique, observando este, cuál es la característica que provoca el que una imagen tenga un bajo contraste:

Aplique la transformación de ajuste de contraste con parámetros:

A la hora de obtener los ** valores , tenga en cuenta que éstos deben ser números enteros, ya que representan índices de la VLT (para forzarlo, haga s=round(s) ó s=uint8(s) si está segura de que el resultado estará entre , tras definir la transformación).

Represente la imagen procesada, su histograma y la función de transformación.

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:

Intente ahora conseguir el efecto contrario: tome como imagen original la imagen Skin\_gray\_bw\_560.tif y aplíquele una operación de ajuste de contraste con los parámetros necesarios para reducir su contraste (en vez de ampliarlo), es decir, para obtener una imagen similar a Skin\_gray\_bc\_560.tif.

Indique los valores que ha utilizado para los cuatro parámetros de la transformación y represente dicha función.

1. Descárguese esta imagen del Moodle de la asignatura [↑](#footnote-ref-1)
2. Descárguese esta imagen del Moodle de la asignatura [↑](#footnote-ref-2)
3. Descárguese esta imagen del Moodle de la asignatura [↑](#footnote-ref-3)
4. Descárguese esta imagen del Moodle de la asignatura [↑](#footnote-ref-4)
5. Descárguese esta imagen del Moodle de la asignatura. [↑](#footnote-ref-5)