Práctica 5: Operadores locales (II)		Grupo	
		Puesto	
Apellidos, nombre	**Sánchez Garzón, Laura**	**Fecha**	
Apellidos, nombre	**Remuiñán Cid, Sara**		

El objetivo de esta práctica es presentar al alumno los fundamentos de los operadores locales 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_P5_II_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, asignando 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*).

1 Aproximaciones a la realización de filtros morfológicos

La aplicación de filtros morfológicos es una aproximación no lineal, a diferencia de las vistas en las prácticas anteriores. Análogamente al caso de los operadores lineales, los operadores morfológicos están caracterizados por una máscara b, que aquí denominaremos *kernel*. En la mayoría de los casos, el *kernel* suele ser plano, es decir con un rango limitado a dos valores:$\{0, -\infty\}$. En la práctica estos valores se representan por la pareja $\{1,0\}$ respectivamente. La siguiente figura muestra ejemplos de *kernels* con diversas formas y simetrías (el origen de coordenadas del *kernel* se ha marcado con un punto negro). Observe que todos los elementos salvo el 6 son simétricos respecto del origen y que todos los elementos salvo el 5 son conexos.

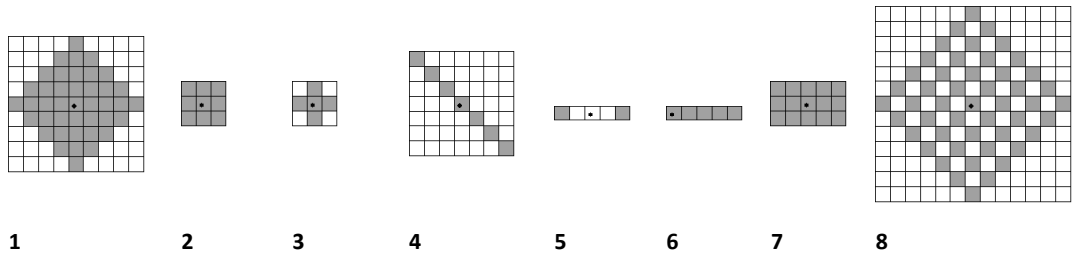

Figura 1: Ejemplos de elementos estructurantes (*kernels*) planos. Los valores oscuros representan ceros y los valores claros representan $-\infty$. El punto negro indica el origen de coordenadas.

Según las explicaciones de la parte teórica de la asignatura sabemos, por ejemplo, que la operación de dilatación con cualquiera de estos *kernels* aplicada sobre un píxel de una imagen consiste en situar el origen del *kernel* sobre el píxel considerado y hallar el máximo de los valores del entorno de dicho píxel que coinciden con ceros (valor oscuro o unidad en binario) del *kernel* utilizado.

Dilatación por correlación con el elemento estructurante

Estudie la función `imfilter_dilate` que se le suministra durante los ejercicios deberá obtener la función alternativa `imfilter_erode`. Tenga en cuenta las siguientes consideraciones:

- Se supone que el punto de aplicación del operador es su centro.
- Ambas funciones deberán tomar como parámetros de entrada la imagen sobre la que operan y el elemento estructurante que se va a aplicar: **(**`ima,mask`**)**. El elemento ha de ser una matriz con valores '1' y '0' y la imagen una imagen de tipo `uint8` en niveles de gris.
- Como se ha visto en teoría, para poder realizar la operación anterior es necesario que la máscara sea de tipo `uint8`, y en el rango `[0,255]`. Además, en el caso de la dilatación la máscara se suele invertir (por motivaciones históricas) al comienzo de cada función:

```
mask=uint8(255*fliplr(flipud(mask)));       % En la función 'imfilter_dilate'
```

- En el caso de la erosión, sin embargo, simplemente han de cambiarse los '1' por '0' y viceversa antes de pasarla a **`uint8`** y en el rango `[0,255]`. Es decir, no hace falta invertir la máscara, pero si deberemos 'negarla'.
- Para realizar la operación de erosión sobre cada píxel, vaya creando para cada píxel de la imagen original (`ima`) una subimagen (`simage`) de igual tamaño que la máscara y luego realice un OR entre ambas para quedarse con los elementos mínimos de la subimagen marcados por la máscara, para la dilatación esta operación está indicada debajo (en este caso se usa un AND y se seleccionan los máximos):

```
and_image    = bitand(simage,mask);       % En la función 'imfilter_dilate'
ima_res(f,c) = max(max(and_image));     % En la función 'imfilter_dilate'
```

- Para realizar una dilatación se han obtenido para cada píxel el máximo de los valores seleccionados con la máscara. Para realizar la erosión deberá por el contrario obtener el mínimo.
- La función deberá retornar el valor correcto de la operación sólo en los píxeles que sea posible, es decir, la función deberá devolver una imagen procesada donde el marco de la imagen (los píxeles cerca de los extremos, definidos en función del tamaño del elemento estructurante) sean iguales a los de la imagen original.
- **Como la aplicación de filtros morfológicos no modifica el rango de la imagen, en este caso no es necesario ajustarlo.**

Utilice la función suministrada para efectuar la dilatación morfológica sobre la imagen `bandas.bmp`[1].

Efectúe una dilatación de la citada imagen con el siguiente elemento estructurante:

```
mask=[1 1 1 1 1;1 1 1 1 1; 1 1 1 0 0;1 1 1 0 0; 1 1 1 0 0];
```

Para comprobar que el resultado es correcto, la imagen original incluye un máximo aislado (un píxel de valor 255) en la parte central izquierda de la banda superior. Como se ha visto en las explicaciones teóricas, al dilatar este máximo el resultado será precisamente el elemento estructurante utilizado.

Una vez haya comprobado que su función arroja el resultado adecuado, indique, visualizando la imagen original y la imagen dilatada, cuál es el efecto que esta operación ha producido sobre los elementos de la imagen, relacionándolo con las propiedades de la dilatación.

---

[1] Descárguese esta imagen del Moodle de la asignatura

1.1 Ejercicio 1: Erosión por correlación con el elemento estructurante

El objetivo de este ejercicio es aplicar filtros morfológicos siguiendo un esquema de filtrado no lineal mediante **correlación con el elemento estructurante**. Para ello, parta de la función `imfilter_dilate` que se le suministra y obtenga la función alternativa **`imfilter_erode`** de acuerdo con las indicaciones anteriores.

A continuación, efectúe una erosión de la imagen `bandas.bmp`[2] con el siguiente elemento estructurante:

```
mask=[1 1 1 1 1;1 1 1 1 1; 1 1 1 0 0;1 1 1 0 0; 1 1 1 0 0];
```

Observe que la imagen original incluye un mínimo aislado (un píxel con valor 0) en la parte central izquierda de la segunda banda superior. Como en el caso de la función **`imfilter_dilate`**, la erosión de este mínimo ha de resultar en una forma igual a la del elemento estructurante. Indique cuál es el efecto que esta operación ha producido sobre los elementos de la imagen original, relacionándolo con las propiedades de la erosión.

Dilatación y erosión por desplazamiento de la señal

Según se ha visto en las explicaciones teóricas, si un elemento estructurante es sencillo expresarlo como una superposición de funciones básicas (algo que siempre ocurre con los elementos planos que habitualmente se usan), es posible operar con las expresiones de la dilatación y de la erosión para obtener una nueva expresión con una interpretación operativa de gran utilidad.

Efectivamente, para el caso de señales unidimensionales, si se tiene un elemento:

$$b[n] = \bigvee\left(\delta_L[n-2], \delta_L[n-1], \delta_L[n], \delta_L[n+1]\right)$$

Note que este elemento corresponde a la máscara **`[1 1 1 1]`**, donde "_" indica la posición central.

, la respuesta (dilatación o erosión) a una señal de entrada es posible expresarla según:

$$y[n] = x[n] \oplus b[n] = \bigvee\left(x[n+1], x[n+2], x[n], x[n-1]\right)$$
$$y[n] = x[n] \ominus b[n] = \bigwedge\left(x[n+1], x[n], x[n-1], x[n-2]\right)$$

, es decir, como el máximo (supremo) o mínimo (ínfimo) respectivamente de versiones de la señal desplazadas siguiendo el mismo patrón de los ceros del elemento estructurante.

En esta práctica aprovecharemos la observación anterior para realizar una implementación alternativa, y mucho más eficiente, de las funciones del ejercicio 1 `imfilter_dilate` e `imfilter_erode`, a las que se denominará `imfilter_dilateD` **e** `imfilter_erodeD`.

Observe que, según lo explicado más arriba, la única diferencia entre estas dos últimas funciones está en el sentido del desplazamiento y en la operación (máximo o mínimo) que se realiza. En el siguiente ejercicio deberá implementar ambas funciones desde cero (es decir, no se inspire en `imfilter_dilate` e `imfilter_erode`) teniendo en cuenta para ello las siguientes consideraciones:

- Se supone que el punto de aplicación (origen) del operador es su centro.
- La función deberá ser genérica, permitiendo el uso de cualquier elemento estructurante, tanto en dimensión como en estructura interna (organización de ceros y unos).
- No es necesario crear simultáneamente tantas imágenes desplazadas como '1's tenga el elemento estructurante (lo que exigiría elevados recursos de memoria). Basta con ir creando imágenes desplazadas e ir calculando el máximo (o mínimo) con el anterior resultado parcial.

---

[2] Descárguese esta imagen del Moodle de la asignatura

- Para generar imágenes desplazadas MatLab dispone de la función **`circshift`** que desplaza circularmente la matriz de entrada (**`ima`**) dependiendo de los valores del vector de desplazamiento (**`shift_vector`**). Este vector está compuesto de dos valores que indican el número de desplazamientos a realizar en las filas y columnas:

```
ima_despl = circshift(ima, shift_vector);
```

- La función deberá retornar el valor correcto de la operación sólo en los píxeles que sea posible, es decir, la función deberá devolver una imagen procesada donde el marco de la imagen (los píxeles cerca de los extremos, definidos en función del tamaño del elemento estructurante) sean iguales a los de la imagen original.

1.2 Ejercicio 2: Dilatación y erosión por desplazamiento de la señal

Desarrolle las funciones **`imfilter_dilateD`** e **`imfilter_erodeD`** mediante **desplazamiento de la señal**. Utilice las funciones desarrolladas para efectuar las mismas operaciones morfológicas que propone el ejercicio anterior. Para comprobar que los resultados son correctos, calcule la energía de la diferencia entre las imágenes obtenidas en este ejercicio y las respectivas imágenes obtenidas en el ejercicio anterior; el resultado debe ser nulo.

Para comparar el tiempo que tarda MatLab en realizar una operación de dilatación con los dos métodos propuestos, incluya un bucle que realice 10 veces una misma dilatación con la función del ejercicio 1, precedido del comando **`tic`** y seguido del comando **`toc`**, y a continuación compare el tiempo obtenido con el que resulte de cambiar la función de dilatación por la desarrollada en este ejercicio. Rellene una tabla similar a la siguiente:

Tiempo estimado con	función basada en correlaciones:
	función basada en desplazamientos:

2 Combinación de operadores morfológicos básicos

Las aplicaciones de los operadores básicos (dilatación y erosión) son muy limitadas, con independencia del elemento estructurante plano que se aplique. Como se ha visto en la parte teórica de la asignatura, la variedad de filtros morfológicos se basa más bien en la combinación de sucesivas operaciones que en la variedad de elementos estructurantes. Este apartado explora las principales aplicaciones de las combinaciones de operadores morfológicos básicos.

En los siguientes ejercicios utilice las funciones de erosión y dilatación de las que dispone MatLab (**`imdilate`** e **`imerode`**). Ambas funciones reciben como parámetros de entrada la imagen a operar y un elemento estructurante definido por la función **`strel`**. A continuación, se muestra un ejemplo de uso de las funciones (consulte la ayuda de MatLab para más información sobre ambas funciones):

```
se = strel('square',3); % creación de un elemento estructurante 3x3 cuadrado
ima_d=imdilate(ima,se); % dilatacion de la imagen ima
ima_e=imerode (ima,se); % erosion de la imagen ima
```

2.1 Ejercicio 3: Gradiente morfológico.

Este apartado ilustra la aplicación de los operadores morfológicos básicos para detectar los contornos de una imagen. Según se ha visto en las explicaciones teóricas, dependiendo de la combinación de operadores que utilicemos podremos obtener distintos tipos de gradientes:

- Gradiente por dilatación: $(x \oplus b) - x$
- Gradiente por erosión: $x - (x \Theta b)$
- Gradiente morfológico: $(x \oplus b) - (x \Theta b)$

Para visualizar el comportamiento del gradiente aplique estas tres operaciones a la imagen `Tools.bmp`[3], con un elemento estructurante cuadrado de tamaño 3x3, y visualícelas. Para crear un elemento estructurante cuadrado de tamaño 3x3 puede utilizar la siguiente instrucción:

```
se = strel('square',3);
```

Observe las imágenes de contornos obtenidas y concluya las diferencias observadas entre los distintos contornos obtenidos con los tres tipos de gradientes morfológicos:

NOTA: debido a que la imagen inicial (`Tools.bmp`[4]) a procesar es binaria, la imagen resultante también debe serlo. Para ser consecuente con ello sustituya la operación aritmética '−' de las ecuaciones por la operación lógica XOR.

Apertura y cierre

Las aplicaciones de la apertura (erosión seguida de una dilatación, ambas con el mismo *kernel*) y de cierre (dilatación seguida de una erosión, ambas con el mismo *kernel*) pueden ser útiles para la eliminación de objetos (simplificación de imágenes) o para su detección.

La apertura y cierre están definidas por la siguiente secuencia de operaciones básicas:

• Apertura:

$$\gamma_b(x) = ((x \Theta b) \oplus b)$$

• Cierre:

$$\varphi_b(x) = ((x \oplus b) \Theta b)$$

2.2 Ejercicio 4: Restauración de un cuadro por apertura

Cargue la imagen **`MRI_gray_garab.jpg`**[5], realice operaciones de apertura sobre la imagen con distintos elementos estructurantes en forma y tamaño hasta conseguir eliminar el trazo blanco realizado sobre el cuadro. Comente la influencia del elemento estructurante en la operación e incluya el resultado final obtenido.

2.3 Ejercicio 5: Eliminación de valla por cierre.

Cargue la imagen **`verjanegra.jpg`**[6], realice operaciones de cierre sobre la imagen con distintos elementos estructurantes en forma y tamaño hasta conseguir eliminar el mallado negro entre las verjas de la imagen. Comente la influencia del elemento estructurante en la operación e incluya el resultado final obtenido.

---

[3] Descárguese esta imagen del Moodle de la asignatura

[4] Descárguese esta imagen del Moodle de la asignatura

[5] Descárguese esta imagen del Moodle de la asignatura

[6] Descárguese esta imagen del Moodle de la asignatura