Práctica 5: Operadores locales II		Grupo	
		Puesto	
Apellidos, nombre		**Fecha**	
Apellidos, nombre			

El objetivo de esta práctica es presentar al alumno las técnicas para realizar operadores morfológicos sobre imágenes binarias y en escala de grises.

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.

1.1 Aproximaciones a la realización de filtros morfológicos

La aplicación de filtros morfológicos puede en gran medida afrontarse mediante la aproximación de filtrado no lineal presentada en la práctica anterior. Análogamente al caso de los operadores lineales, los operadores morfológicos están caracterizados por un *kernel* b que, en la mayoría de los casos, 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.

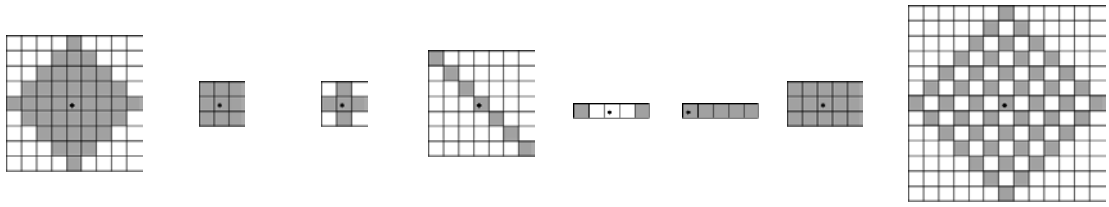

Figura 1: Ejemplos de elementos estructurantes (*kernels*) planos. Los valores oscuros representan ceros y los valores claros representan $-\infty$. El punto negro indica el origen de coordenadas.

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) del *kernel* utilizado.

1.1.1 Ejercicio 1: dilatación y erosión por correlación con el elemento estructurante

El objetivo de este ejercicio es aplicar filtros morfológicos siguiendo el mismo esquema de filtrado no lineal presentado en la práctica anterior. Para ello, partiendo de la función `imfilter_no_lineal_X` allí desarrollada, obtenga dos funciones `imfilter_dilate` e `imfilter_erode`. Para ello tenga en cuenta las siguientes consideraciones:

- Se supone que el punto de aplicación del operador es su centro.
- Para evitar los efectos del *padding* en los extremos de la imagen, la función aplicará la operación sólo en los píxeles que sea posible, es decir, devolverá una imagen procesada más pequeña que la original.

- Modifique las funciones de modo que tomen como parámetros 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 **`unit8`** en niveles de gris.
- Para realizar la operación de dilatación o 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 AND (o un OR) entre ambas para quedarse con los elementos máximos (o mínimos) de la subimagen marcados por la máscara:

```
>> and_image=bitand(simage,mask);          % En la función 'imfilter_dilate'
>> or_image=bitor(simage,mask);            % En la función 'imfilter_erode'
```

- Para poder realizar la operación anterior es necesario que la máscara sea de tipo **`unit8`**. Además, en el caso de la dilatación la máscara ha de invertirse previamente y en el caso de la erosión han de cambiarse los '1' por '0' y viceversa. Para ello, al comienzo de cada función modifique la máscara según:

```
>> mask=uint8(255*fliplr(flipud(mask)));   % En la función 'imfilter_dilate'
>> mask=uint8(255*(1-mask));               % En la función 'imfilter_erode'
```

- Recuerde que para realizar una dilatación ha de obtener para cada píxel el máximo de los valores seleccionados con la máscara y que para realizar una erosión debe obtener el mínimo.
- Como la aplicación de filtros morfológicos no modifica el rango de la imagen, en este caso no es necesario ajustarlo.

Utilice las funciones desarrolladas para efectuar operaciones morfológicas sobre la imagen **`bandas.bmp`**[1]. En primer lugar efectúe una dilatación de la citada imagen con el siguiente elemento estructurante:

```
>> mask=[1 1 1];
```

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 a tamaño real 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:

A continuación efectúe una erosión de la imagen original con el siguiente elemento estructurante:

[1] Descárguese esta imagen de la página *web* de la asignatura.

```
>> mask=[1 1 1]';
```

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 anterior, la erosión de este mínimo ha de resultar en una forma igual a la del elemento estructurante. Como en el caso anterior, indique cuál es el efecto que esta operación ha producido sobre los elementos de la imagen original, relacionándolo ahora con las propiedades de la erosión:

.

1.1.2 Ejercicio 2: 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:

$b[n] = \bigvee \left(\delta_L \left[n+1 \right], \delta_L \left[n \right], \delta_L \left[n-1 \right], \delta_L \left[n-2 \right] \right)$ (que corresponde a la máscara [1 1 1 1])

, 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 o mínimo (respectivamente) de versiones de la señal desplazadas siguiendo el mismo patrón de los ceros del elemento estructurante.

El objetivo de este ejercicio es aprovechar la observación anterior para realizar una implementación alternativa, y mucho más eficiente, de las funciones **`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. Implemente ambas funciones teniendo en cuanta las siguientes consideraciones:

- Se supone que el punto de aplicación (origen) del operador es su centro.
- Para evitar los efectos del *padding* en los extremos de la imagen, la función sólo calculará el máximo (o mínimo) de las imágenes desplazadas en la región en que se solapen todas ellas.
- 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 la anterior.

- 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 esta compuesto de dos valores que indican el número de desplazamientos a realizar en las filas y columnas:

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

- Para calcular el máximo (o mínimo) con la anterior recuerde que la imagen resultado y la desplazada tienen dimensiones distintas por lo tanto la operación se calculará en la región en que se solapen ambas imágenes.

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 la tabla adjunta:

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

1.2 Propiedades de la dilatación y de la erosión

La dilatación y la erosión son las dos operaciones básicas en que se basa el desarrollo de filtros morfológicos. El objetivo de este grupo de ejercicios es profundizar en su efecto sobre distinto tipo de imágenes y presentar sus propiedades.

1.2.1 Ejercicio 3: distribución

La propiedad de distribución de los dos operadores morfológicos básicos enuncia:

$$(x \vee y) \oplus b = (x \oplus b) \vee (y \oplus b)$$
$$(x \wedge y) \Theta b = (x \Theta b) \wedge (y \Theta b)$$

El objetivo de este ejercicio es comprobar la veracidad de este enunciado aplicándolo sobre las imágenes **`cuadros.bmp`** y **`edificio.bmp`**[2] y utilizando una máscara cuadrada de 3x3 ‘1’s.

Para comprobar la distribución de la dilatación respecto del supremo obtenga el máximo de ambas imágenes y a continuación dilátelas. Por otra parte obtenga el máximo de las dilataciones de cada imagen por separado. Para comprobar que ambos resultados son iguales obtenga la energía de su diferencia, energía que deberá ser nula.

[2] Descárguese ambas imágenes de la página *web* de la asignatura

Análogamente, para comprobar la distribución de la erosión respecto del ínfimo obtenga el mínimo de ambas imágenes y a continuación erosiónelas. Por otra parte obtenga el mínimo de las erosiones de cada imagen por separado. Para comprobar que ambos resultados son iguales obtenga nuevamente la energía de su diferencia, energía que deberá ser nula.

1.2.2 Ejercicio 4: composición

La propiedad de composición de los dos operadores morfológicos básicos enuncia:

$$x \oplus a \oplus b = x \oplus c, \quad c = a \oplus b$$
$$x \Theta a \Theta b = x \Theta c, \quad c = a \oplus b$$

El objetivo de este ejercicio es comprobar la veracidad de este enunciado (observe que la erosión no es asociativa) aplicándolo sobre la imagen **`bandas.bmp`**. En ambos casos, dilatación y erosión, el objetivo será diseñar una máscara final que elimine parte de las figuras, negras o blancas respectivamente, simplificando así la imagen.

Para el caso de la dilatación defina las máscaras:

$$a = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad b = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Dilate la imagen original con la máscara **`a`** y el resultado con la máscara **`b`**. Observe que el efecto combinado de ambas dilataciones elimina casi por completo las figuras negras. Obtenga ahora una la máscara resultante de dilatar **`a`** con **`b`** utilizando:

```
>> mask_c=imfilter_dilateD(uint8(mask_a), mask_b);
```

Dilate la imagen original con la máscara **`c`** y compruebe que el resultado es visualmente idéntico. Transcriba la máscara **`c`** resultante de combinar **`a`** y **`b`**:

Para el caso de la erosión defina las máscaras:

$$a=\begin{bmatrix}0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&1&1&1&0&0\\0&0&1&1&1&0&0\\0&0&1&1&1&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\end{bmatrix} \qquad b=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

Repita el mismo procedimiento que para el caso de la dilatación. Observe que ahora el efecto combinado de ambas erosiones elimina por completo las figuras blancas. Obtenga la máscara resultante de erosionar **a** con **b** y erosione con ella la imagen original.

```
>> mask_c=imfilter_erodeD(uint8(mask_a), mask_b);
```

Erosione la imagen original con la máscara **c** y compruebe que el resultado **no** es visualmente idéntico. En lugar de erosionar **a** con **b**, dilátela y aplique la máscara resultante a la imagen original. Compruebe que, ahora sí, el resultado es visualmente idéntico. Transcriba las máscaras resultantes de erosionar **a** con **b** y de dilatar **a** con **b**:

1.2.3 Ejercicio 5: extensividad

Según se ha visto en las explicaciones teóricas, si el elemento estructurante o *kernel* incluye el origen de coordenadas (algo que en las implementaciones de los ejercicios iniciales se da por hecho), la dilatación es extensiva y la erosión es anti-extensiva, es decir:

$$x \leq x \oplus b$$
$$x \ominus b \leq x$$

Para comprobarlo dilate la imagen **edificio.bmp** con el elemento estructurante de 3x3 que desee. A continuación reste la imagen dilatada de la imagen original. Para poder restarlas han de tener las mismas dimensiones; como las funciones de dilatación implementadas no hacen *padding*, la imagen dilatada es menor, por lo que tendrá que eliminar una orla de la imagen original antes de restarlas:

```
>> ima=ima(2:end-1,2:end-1);
>> ima_dif=ima_dilated-ima;
```

Como la imagen dilatada siempre es mayor o igual que la original (por ser la dilatación extensiva), el resultado siempre va a ser positivo, por lo que no es necesario hacer conversiones a **double** para realizar la operación. Pruebe con distintos elementos estructurantes para comprobar que así es.

Observe la imagen diferencia. Indique una posible aplicación directa de la propiedad de extensividad de la dilatación. Indique asimismo de qué manera afectaría la elección de uno u otro elemento estructurante (o de un tamaño u otro) para esta aplicación concreta y concluya cuál o cuáles son los *kernels* más adecuados:

La situación con la operación de erosión es dual a la anterior. En este caso la operación es anti-extensiva, por lo que el resultado de una erosión es siempre menor o igual que la imagen original. Para comprobarlo siga el mismo procedimiento que para el caso de la dilatación, pero ahora restando de la imagen original su erosión. Observe la imagen diferencia y concluya de nuevo una posible aplicación de la propiedad de anti-extensividad de la erosión:

1.3 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`**) que devuelven una imagen procesada con las mismas dimensiones que la imagen original. 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
```

1.3.1 Ejercicio 6: 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 comprobarlo 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

[3] Descárguese esta imagen de la página *web* de la asignatura.

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`**) a procesar es binaria, la imagen resultante también debe serlo. Para ello sustituya la operación aritmética '$-$' por la operación lógica XOR.

1.3.2 Ejercicio 7: Aperturas y cierres

Este apartado ilustra la aplicación de la apertura (erosión seguida de una dilatación, ambas con el mismo *kernel*) y del cierre (dilatación seguida de una erosión, ambas con el mismo *kernel*) para la eliminación de objetos (simplificación de imágenes) y 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: $\varphi_b(x) = ((x \oplus b) \Theta b)$

Para comprobarlo, primeramente aplique ambas operaciones a la imagen **`test_binary.bmp`**[4], con un elemento estructurante cuadrado de tamaño 3x3. Anote las diferencias observadas entre las imágenes resultantes de la aplicación de la apertura y el cierre sobre la imagen original:

A continuación aplique otra vez ambas operaciones sobre la imagen **`cuadro_bw_400_2.bmp`**, con un elemento estructurante cuadrado de tamaño 3x3 y sobre la imagen **`verja.bmp`** con un elemento estructurante no conexo del siguiente tipo

```
>> se=strel('arbitrary',[1 0 0 0 1]);
```

Anote las diferencias observadas entre la aplicación de una operación de apertura o cierre a una imagen de niveles de gris y el efecto de simplificación sobre máximos y mínimos que se produce en la imagen resultante:

[4] Descárguese esta imagen de la página *web* de la asignatura.

1.3.3 Ejercicio 8: Filtrado por reconstrucción

Según se ha visto en las explicaciones teóricas y en el apartado anterior, la apertura o cierre se utilizan para simplificar imágenes (e.g., eliminación de objetos). Aunque se preservan los contornos de los objetos restantes más que una dilatación o una erosión, aún generan deformaciones en los contornos de los objetos que no fueron eliminados. Este apartado ilustra la aplicación de una operación morfológica de la familia de los llamados *Filtros por Reconstrucción* que realiza la misma operación de simplificación en imágenes binarias pero preservando los contornos de los objetos presentes.

En general, una operación de *Filtrado por Reconstrucción* viene definida por:

- Una señal de entrada x
- Una señal marcador y

Un proceso de reconstrucción preserva las componentes conexas de x marcadas por y. Dependiendo del orden de las operaciones implicadas existen dos tipos de reconstrucciones:

- Apertura por reconstrucción $\gamma^{rec}(x;y)$
- Cierre por reconstrucción $\varphi^{rec}(x;y)$

En este apartado se propone la aplicación de la *Apertura por Reconstrucción* como método de simplificación de imágenes binarias preservando los contornos. Este método viene definido por los siguientes pasos:

- Cálculo de elemento estructurante con tamaño/forma adecuados a los objetos/detalles a simplificar
- Cálculo de imagen marcador que guía el proceso de reconstrucción
- Aplicación iterativa del siguiente proceso:
    - Cálculo de dilatación de imagen marcador
    - Cálculo del mínimo entre la imagen dilatada y la imagen original
    - Repetir hasta que no exista variación significativa entre iteraciones consecutivas

Para proceder a la realización del ejercicio, lea la imagen **`test_binary.bmp`** utilizada en el apartado anterior y calcule la *Apertura por Reconstrucción* sobre la dicha imagen. Tenga en cuenta las siguientes consideraciones:

- Como imagen marcador, una de las posibles opciones es utilizar una versión erosionada de la imagen original (en la que al erosionar no estén presentes los objetos a eliminar/simplificar, para ello utilice un elemento estructurante cuadrado de tamaño adecuado).
- El elemento estructurante a utilizar en la operación iterativa de dilatación ha de ser adecuado al crecimiento (o reconstrucción) de los objetos no simplificados (recomendación: utilice un elemento estructurante cuadrado de tamaño 3x3)
- En imágenes binarias, la operación *min* se puede sustituir por una operación lógica AND.
- Como condición de parada (fin de las iteraciones) se suele utilizar la diferencia en energía de la imagen reconstruida en dos iteraciones consecutivas (por ejemplo, establezca como condición de parada que la energía entre iteraciones consecutivas varíe menos de un 1%)

Visualice en una misma figura las distintas iteraciones de la *Apertura por Reconstrucción* y anote los resultados obtenidos