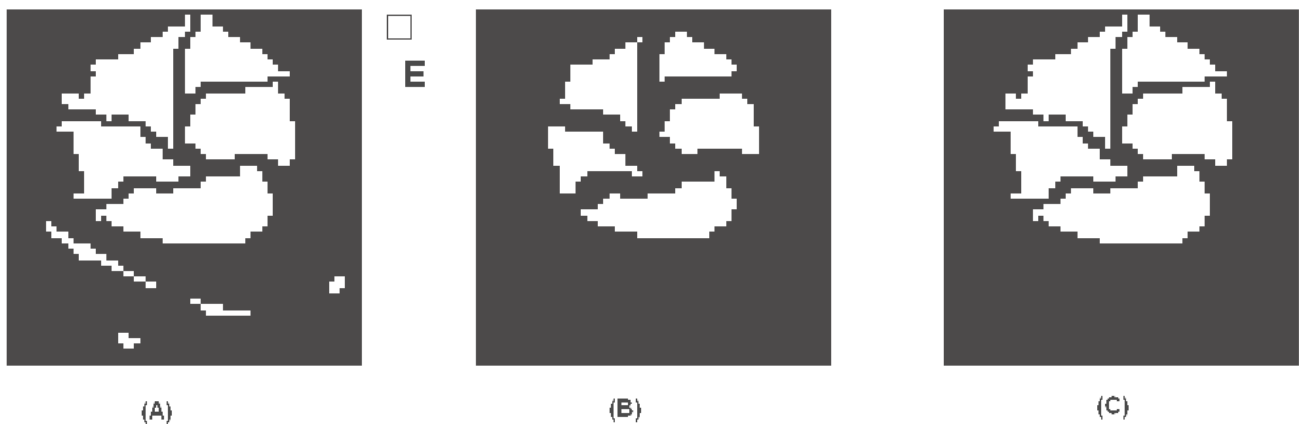


# PRUEBA N°2: PAUTA

Fecha: 23-07-2007

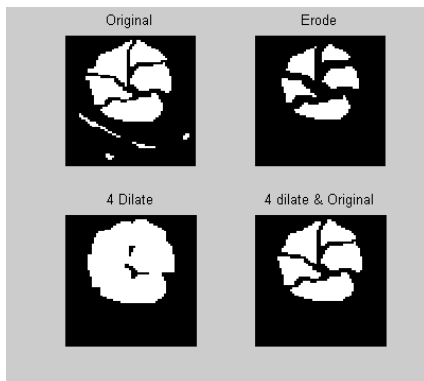
**1.- (10 Ptos.)** En procesamiento de imágenes es muy recurrente la necesidad de eliminar pequeños objetos o "artefactos" que rodean al objeto de interés, como los 4 pequeños objetos que se ubican en la parte inferior de la figura "A". Una solución simple para eliminarlos es aplicar una operación de "erosión" con el elemento estructural "E". El resultado de esta operación se muestra en la figura "B". El problema es que el objeto de interés que se observa en la figura "B" no es idéntico al original. Realizar una operación "dilatación" permitiría recuperar parte de la forma del objeto original, sin embargo no es posible recuperar todos los detalles. Proponga una secuencia de operaciones morfológicas (dilate, erode, open, close, or, xor, and y not) que permita eliminar los artefactos de la figura "A", pero que mantenga todos los detalles del objeto de interés, tal como se muestra en la figura "C".



La imagen B es "subconjunto" de la imagen A.

Para recuperar la imagen C a partir de la imagen B, es necesario volver a dilatar la imagen B. Por lo menos 2 o tres veces para garantizar que los objetos en imagen B dilatados sean más "grandes" que los originales. De esta forma es posible aplicar un AND entre la imagen original A y la imagen B dilatada N veces, obteniendo como resultado la imagen C.

Dado que la imagen B, no posee los "artefactos" de la parte inferior, aunque se dilate, no se recuperarán estos objetos, por lo tanto al realizar la operación AND con la imagen original, se eliminarán.

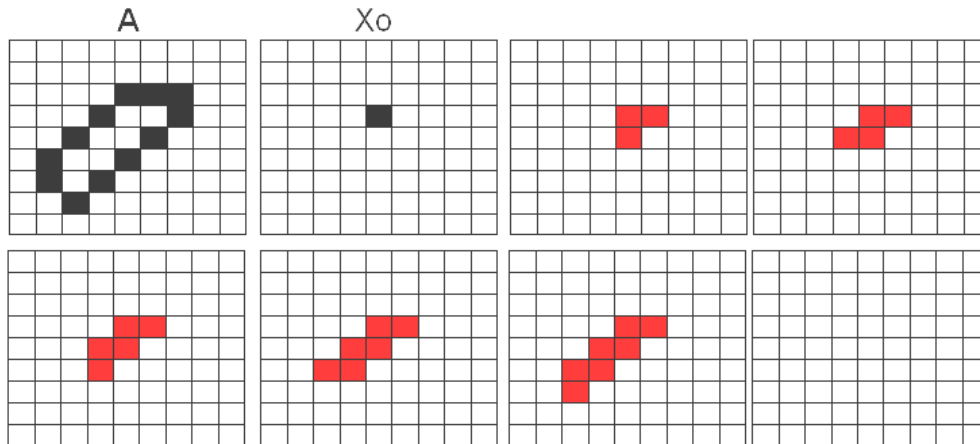


```
I=imread('img.bmp');
I=rgb2gray(I);
subplot(2,2,1);imshow(I,[]);title('Original');
E=ones(3);
B=erode(I,E);
subplot(2,2,2);imshow(B,[]);title('Erode');
B=dilate(B,E);
B=dilate(B,E);
subplot(2,2,3);imshow(B,[]);title('4 Dilate');
B=I&B;
subplot(2,2,4);imshow(B,[]);title('4 dilate & Original');
```

**2.- (10 Ptos.)** La ecuación presenta una operación morfológica recursiva que corresponde a una dilatación de la imagen  $X_{k-1}$  y una intersección con el complemento de la imagen A. a) Dibuje todos los pasos de esta secuencia hasta que los cambios entre  $X_{k-1}$  y  $X_k$  se detengan. b) ¿Qué utilidad tiene esta operación?

Suponga  $B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

$$X_k = (X_{k-1} \oplus B) \cap A^c$$



Esta operación morfológica recursiva permite rellenar un objeto con borde continuo, si se tiene como punto de partida un punto al interior del "objeto". La imagen resultante contiene sólo los puntos interiores (sin el borde).

**3.- (10 Ptos.)** Proponga un algoritmo para determinar el perímetro aproximado del objeto con etiqueta "1", utilizando alguna operación morfológica como parte principal del algoritmo. Comente además cuál es el elemento estructurante que utilizaría y por qué.

|  |   |   |   |   |   |  |
|--|---|---|---|---|---|--|
|  |   |   |   |   |   |  |
|  |   | 1 | 1 | 1 |   |  |
|  | 1 | 1 | 1 | 1 | 1 |  |
|  | 1 | 1 | 1 | 1 | 1 |  |
|  | 1 | 1 | 1 | 1 | 1 |  |
|  |   | 1 | 1 | 1 |   |  |
|  |   |   |   |   |   |  |

Una forma de determinar el perímetro de un objeto es determinar un borde de ancho 1, y contar todos los puntos que lo conforman.

Para determinar el borde de la imagen A, utilizando operaciones morfológicas se puede aplicar una operación  $B = \text{ERODE}(A)$ , y luego realizar una resta entre la imagen A y la imagen B.

El elemento estructurante para no generar distorsiones en la diagonales debe ser  $E = [0 \ 1 \ 0; \ 1 \ 1 \ 1; \ 0 \ 1 \ 0]$ ;

Algoritmo:

```
Sea A la imagen original
E=[0 1 0; 1 1 1; 0 1 0];
B=erode(A,E);
C=A-B;
Perimetro=sum(C)
```

**4.- (5 Ptos.)** Suponga una matriz  $F$  que corresponde a la transformada de Fourier de una imagen  $IM$ . Programe un filtro pasa banda ideal con frecuencias de corte  $fc1$  y  $fc2$ .

```
IM=imread('im.bmp');  
F=fft2(IM);  
F=fftshift(F);%Swap de cuadrantes  
fc1=5;  
fc2=12;
```

```
%Filtro Pasa Bandas  
%La frecuencia 0 está en el centro de la matriz  
[nf nc]=size(F)  
F2=zeros(nf,nc);  
cf=round(nf/2);%centro de la Matriz F  
cc=round(nc/2);%centro de la Matriz F  
for f=1:nf  
    for c=1:nc  
        D=sqrt((f-cf)^2+(c-cc)^2);  
        if D>=fc1 & D<=fc2  
            F2(f,c)=F(f,c);  
        end;  
    end;  
end;
```

```
F2=ifftshift(F2);%Swap inverso  
IM=real(ifft2(F2));  
imshow(IM,[]);
```

**5.- (10 Ptos.)** En restauración de imágenes es muy común utilizar el "filtro de mediana" para eliminar el ruido tipo "sal y pimienta". Escriba un pseudo código que permita implementar este filtro.



```

A=imread('foto.jpg');
A=double(rgb2gray(A));
subplot(1,2,1);imshow(A,[]);title('Original');

[nf nc]=size(A);
B=0*A;

for f=2:nf-1
    for c=2:nc-1

        vector=[];
        for ff=f-1:f+1
            for cc=c-1:c+1
                vector=[vector A(ff,cc)];
            end;
        end;

        for i=1:9-1
            for j=i+1:9
                if vector(j)<vector(i)
                    aux=vector(i);
                    vector(i)=vector(j);
                    vector(j)=aux;
                end;
            end;
        end;
        mediana=vector(5);

        B(f,c)=mediana;
    end;
end;

subplot(1,2,2);imshow(B,[]);title('Filtro Mediana');

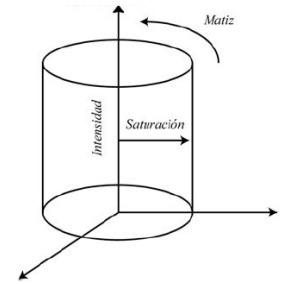
```

**6.- (5 Ptos.)** En procesamiento de imágenes en color es muy utilizada la transformación RGB a HSI. Explique qué representa cada una de las bandas H, S e I, y cómo se calculan (en palabras).

La banda S corresponde a la SATURACION. Representa el grado de predominancia de un color. Los colores puros como [255,0,0] están saturados. En cambio cuando un color no tiene una componente muy predominante, se dice que es poco saturado como por ejemplo [170, 175, 169]. Se calcula como la distancia radial del modelo cilíndrico HSI

La banda H corresponde al HUE, o matiz. Representa al color predominante. El matiz corresponde al ángulo del sistema de coordenadas polar.

La banda I corresponde a la INTENSIDAD. La intensidad es el eje perpendicular al plano de coordenadas polares y corresponde al promedio de las tres bandas RGB



Representación del espacio HSI

**7.- (10 Ptos.)** Dada una imagen en color, se desea filtrar todos aquellos píxeles que tienen un matiz azul. Escriba un pseudo código que permita realizar esta operación

```
delta=10;
[nf nc c]=size(A);
B=0*A;
for f=1:nf
    for c=1: nc
        r=A(f,c,1);
        g=A(f,c,2);
        b=A(f,c,3);
        if b>r+delta & b>g+delta
            B(f,c,:)=A(f,c,:);
        End;
    End;
End;
```