

Ejemplo práctico: borde y relleno de imágenes binarias utilizando operaciones morfológicas

Pablo Roncagliolo B.

Este documento presenta una aplicación simple que permite obtener el borde de una imagen binaria (binarizada) utilizando operaciones morfológicas. Esta forma de obtener bordes es una excelente alternativa al uso de filtros pasa altos. Sin embargo, como queda claro en el título, sólo se puede utilizar cuando se tiene imágenes binarias.

También se presenta un algoritmo clásico para rellenar imágenes binarias (Fill Hole) utilizando una secuencia de operaciones morfológicas. El relleno de imágenes es muy útil cuando se desea mejorar problemas en la binarización, que generan zonas sin relleno indeseadas.

ETAPA 1: BORDE

Para el ejemplo se utilizó como punto de partida la imagen binaria (1.a).

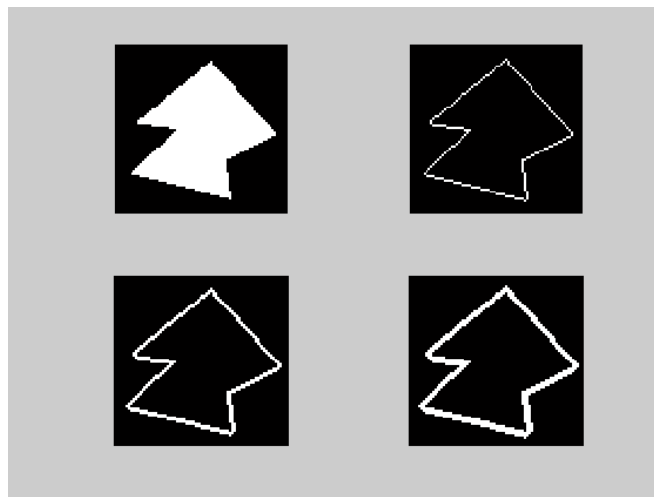


Figura 1

La técnica de obtención de borde utilizada consiste en utilizar un elemento estructurante para dilatar (o erosionar) la imagen original. Este proceso se puede repetir una o más veces. Posteriormente se procede a restar la imagen dilatada con original.

El elemento estructurante utilizado es:

0	1	0
1	1	1
0	1	0

En la figura 1.b, 1.c y 1.d se presenta el resultado de la operación para 1, 2 y 3 iteraciones de dilatación respectivamente.

La notación que representa esta operación es:

$$\beta(IM) = (IM \oplus B) - IM$$

Es decir la imagen de Borde corresponde a la dilatación menos la imagen original.

Es importante notar que el borde calculado con esta operación corresponde al borde "externo" de la imagen original. Es decir los puntos del borde no son parte de los puntos del objeto original.

Otra tal vez más apropiada para mantener las proporciones del objeto original es determinar el borde en base a la imagen original menos la imagen erosionada.

$$\beta(IM) = IM - (IM \ominus B)$$

El código del programa en Matlab es:

```
IM=imread('bw.bmp');  
subplot(2,2,1);imshow(IM);  
  
%Elemento Estructurante  
B=[0 1 0;1 1 1;0 1 0];  
  
%Dilatación  
IM2=dilate(IM,B);  
  
%Borde  
BORDE=double(IM2)-double(IM);  
subplot(2,2,2);imshow(BORDE);  
  
%Dilatación  
IM2=dilate(IM2,B);  
  
%Borde  
BORDE=double(IM2)-double(IM);  
subplot(2,2,3);imshow(BORDE);  
  
%Dilatación  
IM2=dilate(IM2,B);  
  
%Borde  
BORDE=double(IM2)-double(IM);  
subplot(2,2,4);imshow(BORDE);
```

ETAPA 2: RELLENO

La operación de relleno en base a operaciones morfológicas consiste es una secuencia de dilataciones, complementos e intersecciones.

Dada una imagen de borde original como la imagen A de la figura 2, se obtiene la imagen A^C , que corresponde al complemento de la imagen A. Se define también un elemento estructurante B.

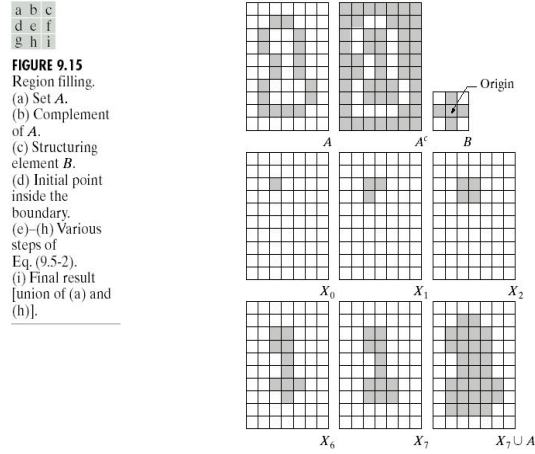


Figura 2

El proceso de relleno consiste en aplicar recursivamente la siguiente operación:

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

Se utiliza como punto de partida un punto al interior de la objeto que se desea rellenar como muestra la imagen X_0 y se realiza la operación dilatación con el elemento B. Esta dilatación expande la imagen en todas las direcciones. Sin embargo al realizar la intersección con la imagen A^C , se logra mantener los puntos dilatados siempre dentro de los bordes de la imagen original. Este proceso se repite "n-veces" hasta que la imagen X_k sea idéntica a la imagen X_{k-1} . Es decir, hasta que no se generen nuevos cambios y el proceso se detenga.

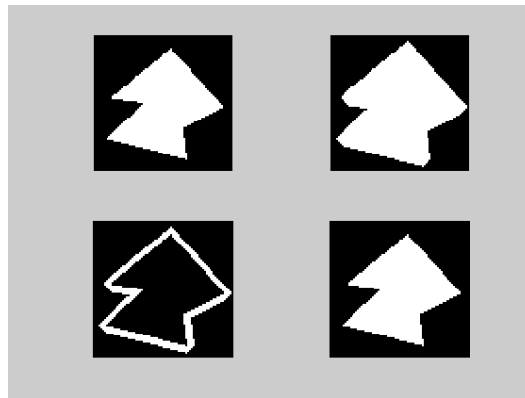


Figura 3

La fig. 3d presenta el resultado del relleno de la imagen 3.c utilizando el siguiente algoritmo en Matlab:

```
IM=imread('bw.bmp');
subplot(2,2,1);imshow(IM);

%Elemento Estructurante
B=[0 1 0;1 1 1;0 1 0];

%Dilatación 4 veces
IM2=dilate(IM,B);IM2=dilate(IM2,B);
IM2=dilate(IM2,B);IM2=dilate(IM2,B);
subplot(2,2,2);imshow(IM2);

%Borde
BORDE=double(IM2)-double(IM);
subplot(2,2,3);imshow(BORDE);

%FUNCION DE RELLENO
%Se Busca el primer punto al interior del borde
[nf nc]=size(IM);
f=1;
while f<nf
    c=1;
    while c<nc
        if IM(f,c)==1 % Si encuentra un punto termina los ciclos
            ff=f;cc=c;
            f=nf;c=nc;
        end;
        c=c+1;
    end;
    f=f+1;
end;

IMc=1-BORDE;%Obtiene imagen complemento
X=zeros(nf,nc);
X(ff,cc)=1;%Imagen con punto de partida
Xold=X;
while (1)
    X=double(dilate(X,B));
    X=X.*IMc;
    subplot(2,2,4);imshow(X);
    pause(0.05)
    if Xold==X, break;end;%Detecta si no hay cambios!
    Xold=X;
end;
```