

Ejemplo práctico: detección de componentes conectados y etiquetado utilizando operaciones morfológicas

Pablo Roncagliolo B.

Este documento presenta una aplicación simple que permite obtener el número de objetos independientes presentes en la imagen. Esto se logra detectando los componentes conectados y asignándoles un valor o etiqueta identificadora.

Para el ejemplo se utilizó como punto de partida la imagen binaria siguiente.

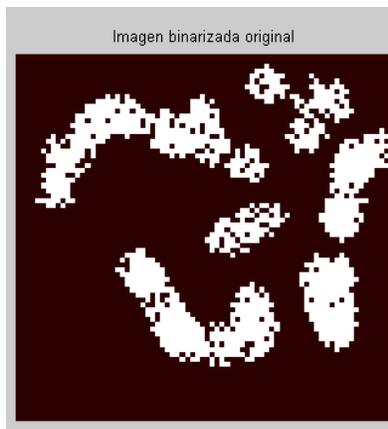


Figura 1

La técnica de obtención de componentes conectados (objetos independientes) consiste en utilizar un elemento estructurante para dilatar la imagen original a partir de 1 punto cualquiera del objeto y posteriormente aplicar la operación intersección respecto de la misma imagen original.

El elemento estructurante utilizado es:

1	1	1
1	1	1
1	1	1

La notación que representa esta operación es:

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

Es decir, en cada iteración se aplica la operación dilatación con el elemento B y luego se calcula la intersección respecto de la imagen original.

El algoritmo comienza buscando el primer punto (cualquiera) igual a "1" en la imagen. Este punto "p" representa la imagen X_0 sobre la cual se aplica la primera dilatación y posterior intersección con la imagen original. Las imágenes X_k se representan en la siguiente figura en base a píxeles con textura diagonal.

Esta operación se repite hasta que la imagen X_{k+1} sea igual a la imagen X_k . Es decir hasta que no existan más cambios al aplicar la operación.

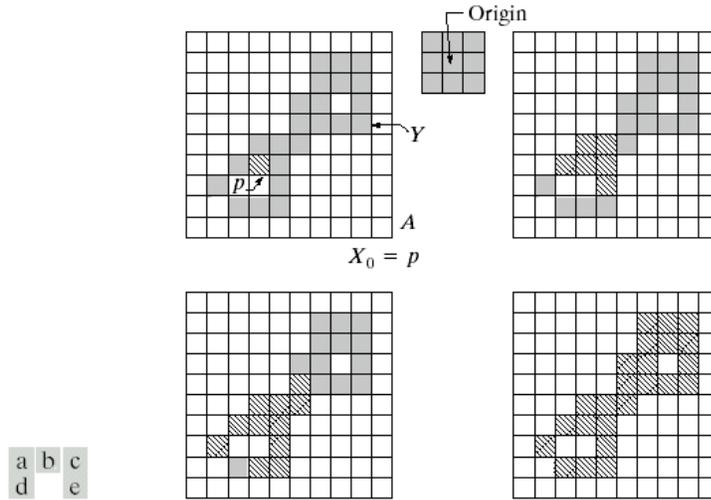


FIGURE 9.17 (a) Set A showing initial point p (all shaded points are valued 1, but are shown different from p to indicate that they have not yet been found by the algorithm). (b) Structuring element. (c) Result of first iterative step. (d) Result of second step. (e) Final result.

El resultado se muestra en la figura 2. El primer objeto "conectado" se logra después de aplicar varias veces la operación propuesta.

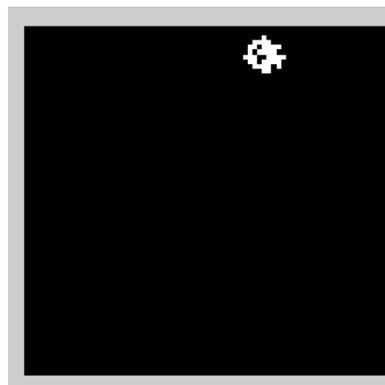


Figura 2

Este primer objeto encontrado se puede restar de la imagen original, para aplicar nuevamente el algoritmo en búsqueda del segundo objeto. También se puede almacenar el objeto encontrado en otra imagen y etiquetar con un identificador único igual en todos sus píxeles.

Al repetir el algoritmo hasta que todos los objetos sean etiquetados se puede visualizar en colores las diferentes etiquetas asignadas, tal como lo muestra la figura 3.

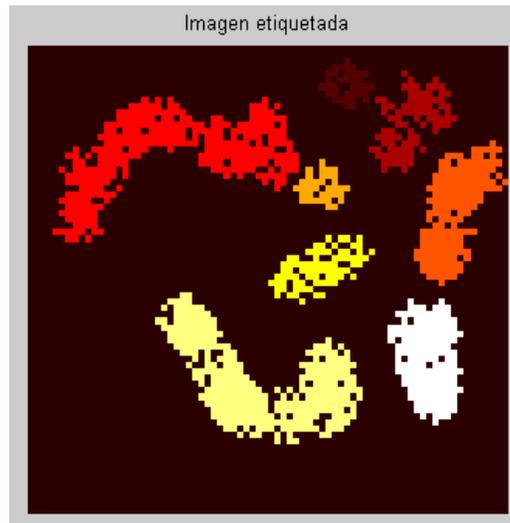


Figura 3

El código del programa en Matlab es:

```
IM=double(imread('bw2.bmp'));
subplot(2,2,1);imshow(IM);title('Imagen binarizada original');

%Elemento Estructurante
B=[1 1 1;1 1 1;1 1 1];
Y=zeros(nf,nc);
[nf nc]=size(IM);

n=0;
ok=0;
while ok==0
    %Se Busca el primer punto al interior del borde
    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;
    X=zeros(nf,nc);
    X(ff,cc)=1;%Imagen con punto de partida
    Xold=X;
    while (1)
        X=double(dilate(X,B));
        X=X.*IM;
        subplot(2,2,2);imshow(X);
        pause(0.001)
        if Xold==X, break;end;%Detecta si no hay cambios!
        Xold=X;
    end;
    if sum(sum(X))==0, %verifica si no hay nuevo objeto
        ok=1;
        n
    else
        n=n+1;
        IM=IM-X;%se resta el objeto de la imagen original
        X=(X==1)*n*2;%etiqueta
        Y=Y+X;%almacena objeto en imagen Y
    end;
end;

subplot(2,2,3);imshow(Y,hot(n*2));title('Imagen etiquetada');
```