

Visió per Computador

ENTREGA 6

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Raúl García Fuentes

Kleber Enrique Reyes Illescas

30 d'Octubre de 2018

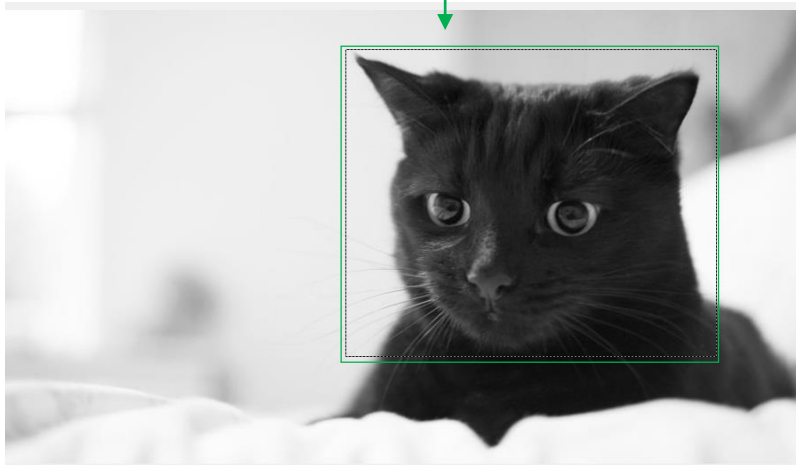
La pràctica 6 consisteix a binaritzar una determinada regió d'una imatge per tal de trobar el centroide de la figura principal de la regió.

Hem fet servir la imatge d'un gat com a exemple:

PROCEDIMENT:

- 1) Carregar la imatge, transformar-la a Blanc/Negre i seleccionar la regió que volem binaritzar:

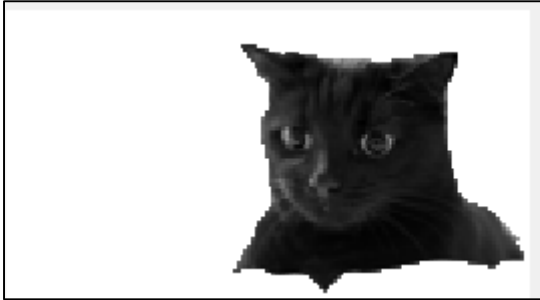
```
% Carreguem la imatge  
>> I = imread('tesla-cat.jpg');  
  
% Transformem la imatge a Blanc/Negre  
>> Indg = rgb2gray(I);  
  
% Seleccionem l'àrea que volem binaritzar  
>> [Selected, rect] = Select(Indg);
```



2) Dividim la imatge en “foreground” i “background” (Binarització)

```
% Obtenim els histogrames del nostre rectangle i del fons
>> hfore = imhist(Selected);
>> hback = imhist(Indg);

% Funció que ens binaritza la imatge a partir dels histogrames
>> Is = Fragmentacio(Indg, hback, hfore);
```



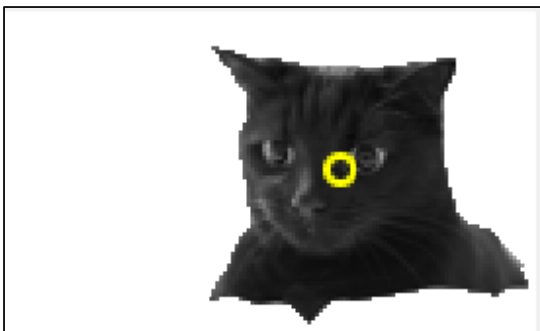
Hem hagut d'aplicar a la imatge un reescalat d'un 20% per millorar la eficiència de l'algorisme, per aquest motiu es veu una mica pixelada.

3) Busquem el centroide de la nostra figura (el gat)

```
% Fem servir la funció de Matlab “regionprops” per buscar els centroides
% de totes les figures que ha detectat automàticament
>> S = regionprops(Is, 'centroid');
>> centroids = cat(1, S.Centroid);

% Es possible que divideixi el gat en diverses figures i trobi molts
% centroides, per tant ens quedarem amb el “centroide central”, per mitjà
% de la funció “getCentroideCentral”
>> [centrX, centrY] = getCentroideCentral(centroids);

% Finalment representem els resultats obtinguts:
>> circleCentroid = [centrX centrY 3];
>> Final = insertShape(Is, 'circle', circleCentroid, 'LineWidth', 2);
>> figure; imshow(Final, []);
```



El cercle groc indica la posició del “centroide central” del gat.

CODI DE LES FUNCIONS EMPRADES:

```
function [Is] = Fragmentacio(Indg, Histo_background,Histo_foreground)
    [ndgF,ndgC] = size(Indg);
    Ir = imresize(Indg,[ndgF/6 ndgC/6]);

    [F,C] = size(Ir);
    WM = zeros(F*C,F*C); % creacio de la matriu de pesos (llogams entre pixels)

    Av_fore = Average_ndg(Histo_foreground);
    Av_back = Average_ndg(Histo_background);

    If = abs(double(Av_fore-Ir)); % lligam amb el foreground
    Ib = abs(double(Ir-Av_back)); % lligam amb el background

    % definim (arbitrariament) el node (1,1) de la matriu WM com a foreground
    % i el node WM(F*C,F*C) com a background

    % lligam entre el foreground (1,1) a tots els pixels
    WM(1,1:F*C) = If(:)';
    % la matriu ha de ser simetrica
    WM(1:F*C,1) = If(:);

    % lligam entre el background (1,1) a tots els pixels
    WM(F*C,1:F*C) = Ib(:)';
    WM(1:F*C,F*C) = Ib(:);

    % no hi ha d'haveri cap lligam entre el foreground i el background
    WM(1,F*C) = 0;
    WM(F*C,1) = 0;
    WM(F*C,F*C) = 0;
    WM(1,1) = 0;

    % lligam entre un pixel i els seus dos veïns
    for c=2:C-1
        for f=2:F-1
            %diferencia de ndg amb el veï de sota
            v = abs(Ir(f,c) - Ir(f+1,c)); % quant menys diferencia mes lligam
            v = max(0,min(1, 1 - v/32));
            WM(f+(c-1)*F,f+1+(c-1)*F) = v;
            WM(f+1+(c-1)*F,f+(c-1)*F) = v; % la matriu ha de ser simetrica
            %diferencia de ndg amb el veï del costat
            v = abs(Ir(f,c) - Ir(f,c+1)); % quant menys diferencia mes lligam
            v = max(0,min(1, 1 - v/32));
            WM(f+(c-1)*F,f+c*F) = v;
            WM(f+c*F,f+(c-1)*F) = v;
            %diferencia de ndg amb la diagonal
            v = abs(Ir(f,c) - Ir(f+1,c+1)); % quant menys diferencia mes lligam
            v = max(0,min(1, 1 - v/32));
            WM(f+(c-1)*F,f+1+c*F) = v;
            WM(f+1+c*F,f+(c-1)*F) = v;
        end
    end

    G = graph(WM);
    [mf,~,foreground,background] = maxflow(G,1,F*C);

    Is = Ir;
    if Av_fore < 128
        Is(background) = 255;
    elseif Av_fore >= 128
        Is(background) = 0;
    end
end
```

```
function [centrX, centrY] = getCentroideCentral(centroids)

    centroidsX = centroids(:,1);
    c1 = centroidsX(centroidsX>=0);

    centroidsY = centroids(:,2);
    c2 = centroidsY(centroidsY>=0);

    centrX = mean(c1);
    centrY = mean(c2);

end
```

```
function [Selected,rect] = Select(I)

    imshow(I);
    rect = getrect();
    %rect = [380.0 28.0 531.0 463.0];    %Valor obtingut amb imshow(Indg)+getrect()
    Selected = imcrop(I,rect);          %Retallem el rectangle obtingut

end
```

```
function [av] = Average_ndg(histogram)

    total = sum(histogram);
    av = 0;

    for i = 1:256
        prob = histogram(i)/total;
        av = av + prob*(i-1);
    end

end
```