

UNIVERSITAT POLITÈCNICA DE CATALUNYA

VISIÓ PER COMPUTADORS

## Informe Sessió 5b

*Carlota Catot Bragós*  
*Alejandro Domínguez Besserer*

Quadrimestre Tardor 2018-2019



Per resoldre el problema plantejat hem agafat la figura ?? per tal d'analitzar la imatge i determinar quins pixels pertanyen al Foreground i quins pertanyen al Background, hem seguit el següent procediment que es pot trobar al codi presentat més abaix.

En primer lloc, un cop redimensionada la imatge hem determinat manualment el rectangle on es troba el nostre gat. Un cop tenim determinada la subimatge, hem extret els histogrames de la imatge sense el gat (Ht) i de la imatge del gat (Hcat) per tal de trobar el nivell de gris representatiu pel Foreground i el Background. Per tal de fer aquest analisi s'ha realitzat la normalització dels histogrames, ja que no els dos histogrames tenen el mateix nombre de pixels. Un cop tenim els histogrames s'han de determinar per cada punt de la imatge si pertany al Foreground o al Background, hem utilitzat un bucle si el pixel té més representació al background o al foreground per tal de determinar a quin pertany. Per últim el que s'ha determinat que el Background es pintarà de blanc ja que hem escollit un gat negre per la prova.



Figura 1: Imatge a analitzar

```
[file,path] = uigetfile('*.'); % obre l'explorador d'arxius
I = imread(fullfile(path,file));
Irgb = rgb2gray(I);
Ir = imresize(Irgb,[100 120]); % ajust a una mida raonable
imshow(Ir);

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

Him = imhist(Ir);
rectIm = imcrop(Ir, rect);
Hcat = imhist(rectIm);
Ht = Him - Hcat;
Ht = Ht / sum(Ht);
Hcat = Hcat / sum(Hcat);

for i = 1 : numel(Ir)
    Ib(i) = Ht(Ir(i)+1) > Hcat(Ir(i)+1);
    If(i) = ~Ib(i);
end;

% definim (arbitrariament) el node (1,1) com a foreground
% i el node (F*C,F*C) com a background
```

```

WM(1,1:F*C) = If(:)'; % lligam entre el foreground (1,1) a tots els pixels
WM(1:F*C,1) = If(:); % la matriu ha de ser simetrica
WM(F*C,1:F*C) = Ib(:)'; % lligam entre el background (1,1) a tots els pixels
WM(1:F*C,F*C) = Ib(:);
WM(1,F*C) = 0; % no hi ha d'haveri cap lligam entre el foreground i el background
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);

Ir(background) = 255; % com utilitzem un gat negre posem el background a blanc.
imshow(Ir,[]);

```

