
Table of Contents

Practica 7	1
Parte 1 - Detección de cambios	1
Parte 2 - Diferencia acumulada	2
Parte 3 - Flujo óptico Lukas-Kanade	3
b) Imagenes propias	6

Practica 7

% Miguel Ascanio Gómez
% Carlos Ballesteros de Andrés

Parte 1 - Detección de cambios

```
clear all; close all;

Im1 = imread('Tema07-1.jpg', 'jpg');
Im2 = imread('Tema07-2.jpg', 'jpg');

Hsi1 = rgb2hsv(Im1);
Hsi2 = rgb2hsv(Im2);

Hsi1B = Hsi1(:,:,3);
Hsi2B = Hsi2(:,:,3);

Diff = abs(Hsi1B - Hsi2B);
D = mat2gray(Diff);

figure;
subplot(2,2,1); imshow(Im1); impixelinfo; title('Tema07-1');
subplot(2,2,2); imshow(Im2); impixelinfo; title('Tema07-2');
subplot(2,2,3); imshow(D); impixelinfo; title('Diferencia');
subplot(2,2,4); imshow(1-D); impixelinfo; title('Diferencia invertida');

% Se observa que en las imágenes de diferencia queda marcado donde las
% imágenes difieren, como por ejemplo el camión que aparece en la imagen 2
% pero no en la 1
```

Tema07-1



Tema07-2



Diferencia



Diferencia invertida



Pixel info: (X, Y) Pixel Value

Parte 2 - Diferencia acumulada

```
clear all; close all;

Ref = double(imread('Pica30.jpg'));

[M,N,o] = size(Ref);

Acu = zeros(M, N, o);

N = 17;

for i=31:57
    I = double(imread(strcat('Pica', num2str(i), '.jpg')));

    % Acumulador
    ak = (i-1)/(N-1);
    Acu = Acu + ak * abs(Ref - I);
end

Resultado = mat2gray(Acu(:,:,1));

figure;
subplot(1,2,1); imshow(Resultado); title('Diferencias');
```

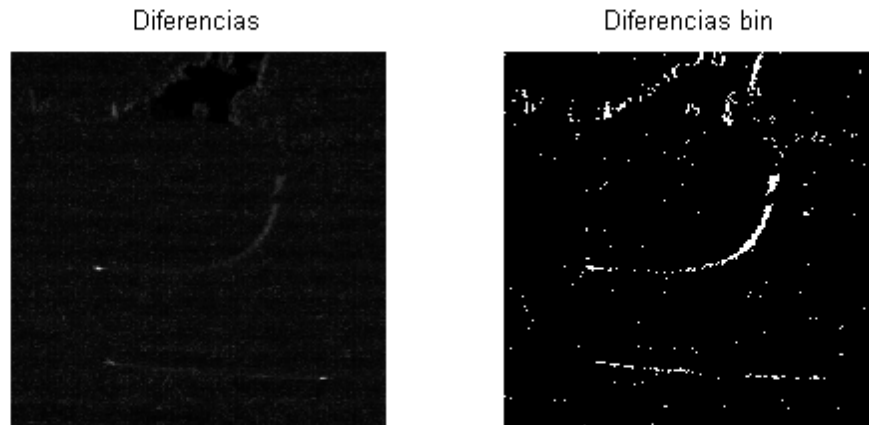
```
% b)

T = graythresh(Resultado);

Binaria = Resultado > T;
Binaria2 = imerode(Binaria, ones(3));

subplot(1,2,2); imshow(Binaria2); title('Diferencias bin');

% Aquí podemos observar que queda marcado el recorrido del avión, además de
% las nubes y los coches de la carretera.
% En la imagen primera se ve muy borroso, pero tras binarizarla los
% movimientos aparecen muchos más marcados, pero con bastante ruido.
```



Parte 3 - Flujo óptico Lukas-Kanade

```
clear all; close all;

% a)

Im1 = imread('Tema07-3.bmp', 'bmp');
Im2 = imread('Tema07-4.bmp', 'bmp');

Hsi1 = rgb2hsv(Im1);
Hsi2 = rgb2hsv(Im2);
```

```

Hsi1I = Hsi1(:,:,3);
Hsi2I = Hsi2(:,:,3);

ho = [-1 0 1;
      -2 0 2;
      -1 0 1];
hv = [ -1 -2 -1;
       0  0  0;
       1  2  1];
hu = ones(3);

fx = conv2(Hsi1I, ho, 'same') + conv2(Hsi2I, ho, 'same');
fy = conv2(Hsi1I, hv, 'same') + conv2(Hsi2I, hv, 'same');
ft = conv2(Hsi1I, hu, 'same') - conv2(Hsi2I, hu, 'same');

u = zeros(size(Hsi1I));
v = zeros(size(Hsi1I));

windowSize = 5;

halfWindow = floor(windowSize/2);
for i = halfWindow+1:size(fx,1)-halfWindow
    for j = halfWindow+1:size(fx,2)-halfWindow
        curFx = fx(i-halfWindow:i+halfWindow, j-halfWindow:j+halfWindow);
        curFy = fy(i-halfWindow:i+halfWindow, j-halfWindow:j+halfWindow);
        curFt = ft(i-halfWindow:i+halfWindow, j-halfWindow:j+halfWindow);

        curFx = curFx';
        curFy = curFy';
        curFt = curFt';

        curFx = curFx(:);
        curFy = curFy(:);
        curFt = -curFt(:);

        A = [curFx curFy];

        U = pinv(A'*A)*A'*curFt;

        u(i,j)=U(1);
        v(i,j)=U(2);
    end;
end;

% 1) Cambiamos las filas de orden
u = flipud(u); v = flipud(v);
%2) Aplicamos el filtro de la mediana en vecindades [5,5]
mu = medfilt2(u,[5 5]); mv = medfilt2(v,[5 5]);
%3) Aplicamos dos descomposiciones piramidales gaussianas para reducir la
%dimensión de las matrices u y v
ru = reduce(reduce(mu)); rv = reduce(reduce(mv));
escala = 0; %valor por defecto (escalado de las flechas de vectores)
figure;
subplot(2,1,1); imshow(Iml); title('Primera imagen');

```

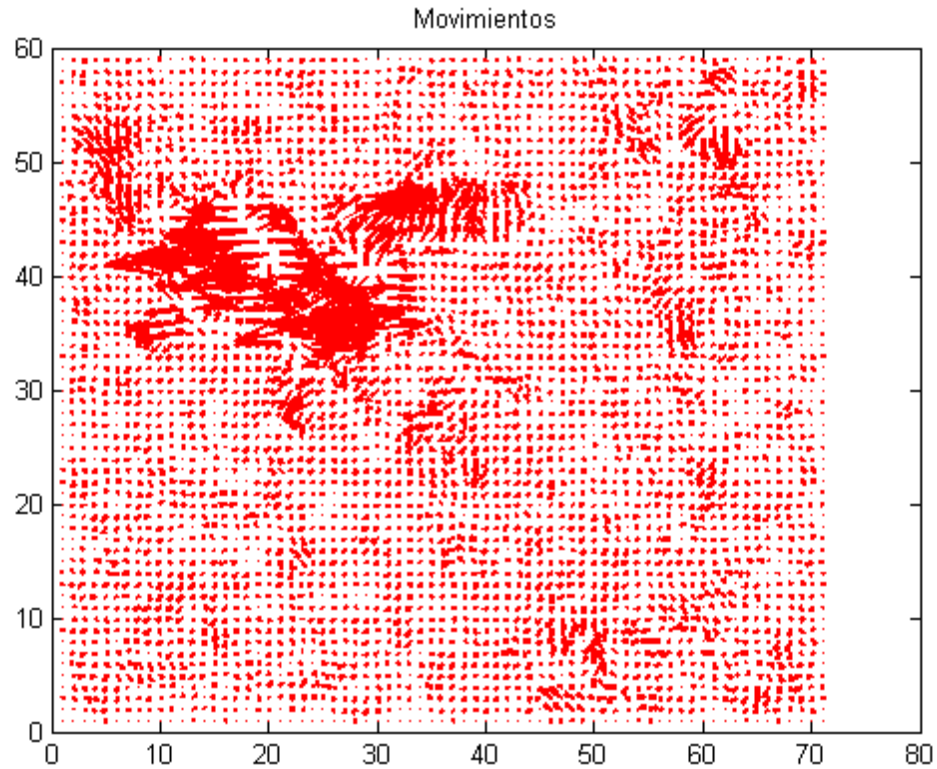
```
subplot(2,1,2); imshow(Im2); title('Segunda imagen');  
figure; quiver(ru, -rv, escala,'r','LineWidth',2); %axis equal  
title('Movimientos');
```

Primera imagen



Segunda imagen





b) Imagenes propias

```
clear all; close all;

Im1 = imread('a.jpg', 'jpg');
Im2 = imread('b.jpg', 'jpg');

Hsi1 = rgb2hsv(Im1(1:25:end, 1:25:end,:));
Hsi2 = rgb2hsv(Im2(1:25:end, 1:25:end,:));

Hsi1I = Hsi1(:,:,3);
Hsi2I = Hsi2(:,:,3);

ho = [-1 0 1;
      -2 0 2;
      -1 0 1];
hv = [-1 -2 -1;
      0 0 0;
      1 2 1];
hu = ones(3);

fx = conv2(Hsi1I, ho, 'same') + conv2(Hsi2I, ho, 'same');
fy = conv2(Hsi1I, hv, 'same') + conv2(Hsi2I, hv, 'same');
ft = conv2(Hsi1I, hu, 'same') - conv2(Hsi2I, hu, 'same');
```

```

u = zeros(size(HsilI));
v = zeros(size(HsilI));

windowSize = 5;

halfWindow = floor(windowSize/2);
for i = halfWindow+1:size(fx,1)-halfWindow
    for j = halfWindow+1:size(fx,2)-halfWindow
        curFx = fx(i-halfWindow:i+halfWindow, j-halfWindow:j+halfWindow);
        curFy = fy(i-halfWindow:i+halfWindow, j-halfWindow:j+halfWindow);
        curFt = ft(i-halfWindow:i+halfWindow, j-halfWindow:j+halfWindow);

        curFx = curFx';
        curFy = curFy';
        curFt = curFt';

        curFx = curFx(:);
        curFy = curFy(:);
        curFt = -curFt(:);

        A = [curFx curFy];

        U = pinv(A'*A)*A'*curFt;

        u(i,j)=U(1);
        v(i,j)=U(2);
    end;
end;

% 1) Cambiamos las filas de orden
u = flipud(u); v = flipud(v);
%2) Aplicamos el filtro de la mediana en vecindades [5,5]
mu = medfilt2(u,[5 5]); mv = medfilt2(v,[5 5]);
%3) Aplicamos dos descomposiciones piramidales gaussianas para reducir la
%dimensión de las matrices u y v
ru = reduce(reduce(mu)); rv = reduce(reduce(mv));
escala = 0; %valor por defecto (escalado de las flechas de vectores)
figure;
subplot(2,1,1); imshow(Im1); title('Primera imagen');
subplot(2,1,2); imshow(Im2); title('Segunda imagen');
figure; quiver(ru, -rv, escala,'r','LineWidth',2); %axis equal
title('Movimientos');

% Se observa que las flechas indican el sentido del movimiento (de los
% objetos) entre las dos imágenes. También se observa que, si se reescala
% la imagen para que sea más pequeña (en las imágenes propias se cambia el
% 25 por el 50), tienden a desaparecer las flechas que no apuntan
% exactamente a la dirección de movimiento, si bien es cierto que esto
% puede causar pérdida de información.

```

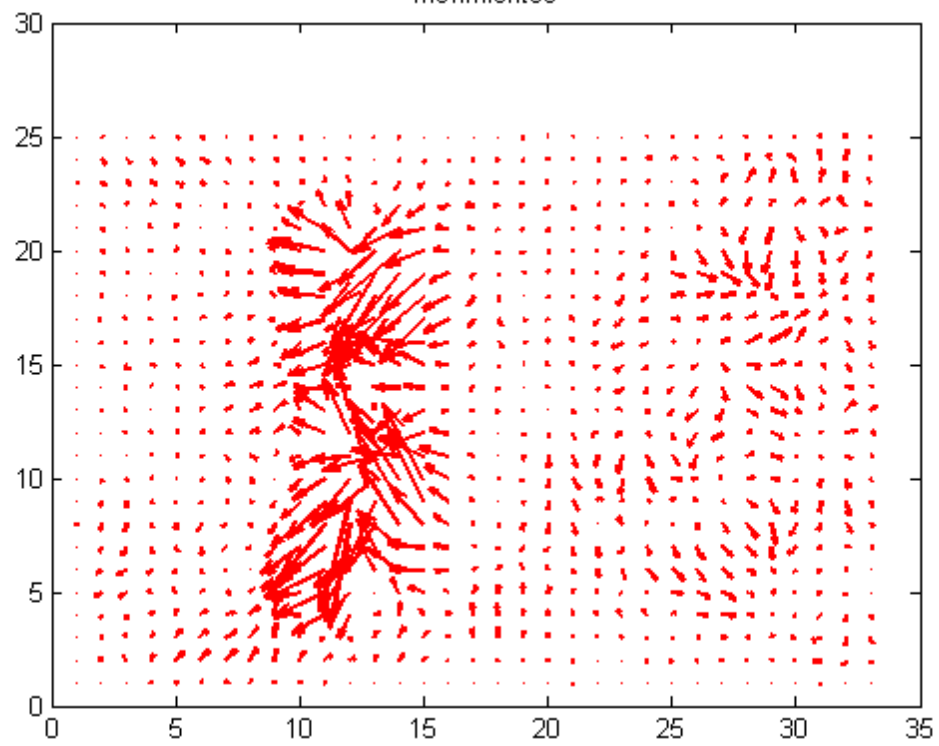
Primera imagen



Segunda imagen



Movimientos



Published with MATLAB® R2014a