

Enunciado:

El caso consiste en leer un fichero de vídeo recogido en una calle cualquiera (en este caso, de Algeciras) y, realizar una identificación del número de vehículos que pasan. Para mayor facilidad se usa un reconocimiento de imágenes que son capturadas cada cierto número de frames de vídeo (esto es un parámetro que puede modificarse).

En pantalla, van apareciendo los resultados de cada fase del análisis y al final el número de vehículos total detectados en la secuencia.

Implementación:

`ContadorVehiculos1.m`: script principal

`ccoches1b.mp4`: fichero que contiene una secuencia de vídeo con vehículos

Resultados - Identificador de vehículos en frames de vídeo

Lectura del vídeo	2
Algoritmo de reconocimiento	3
capturar frame y quitarle umbral.....	3
borrado de objetos menor que minpixels.....	4
eliminación de pequeños discos de radio.....	6
se rellenan huecos para poder usar regionprops para estimar	7
se calculan bordes.....	8
se recopila información de las áreas.....	12
resultado	13

Lectura del vídeo

```
clear all;  
close all;  
  
c = 0;  
trafficVid = VideoReader('c Cochés1b.mp4');  
get(trafficVid);  
imshow('c Cochés1b.mp4');
```

obj =

VideoReader with properties:

General Properties:

Name: 'c Cochés1b.mp4'

Path: 'C:\Users\Ignacio Turias\Documents\DOCTORADO2019\Innovacion'

Duration: 8.6186

CurrentTime: 0

Tag: ''

UserData: []

Video Properties:

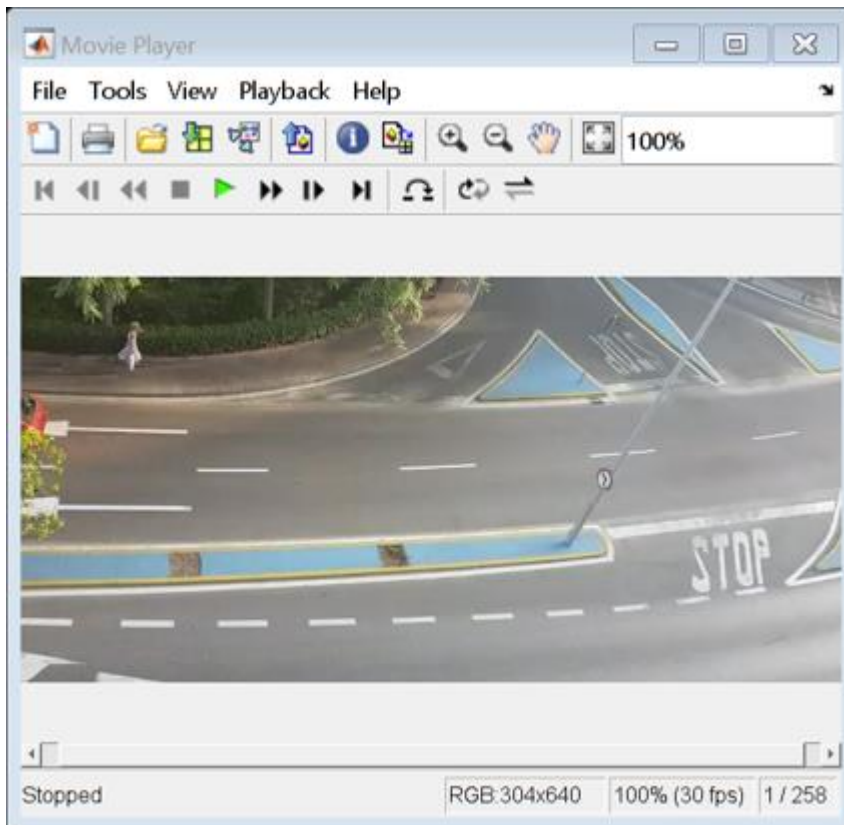
width: 640

Height: 304

FrameRate: 30

BitsPerPixel: 24

VideoFormat: 'RGB24'



Algoritmo de reconocimiento

```
% parámetros
umbral = 50;           % para separar el fondo
minpixels = 30;       % minimo de pixels para ser ruido
salto_frames = 60;    % se cuenta cada segundo (50 frames)
rdisco = 4;           % radio de pequeños elementos de ruido
minarea = 900;        % mínimo número de pixels para considerar coche
nframes = 260;        % frame final

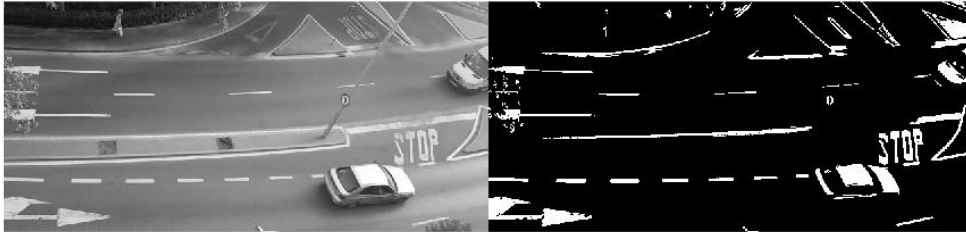
for i = 1:salto_frames:nframes

close all;
```

capturar frame y quitarle umbral

```
im_orig = rgb2gray(read(trafficVid,i));
im_tran = imextendedmax(im_orig, umbral);
figure, imshowpair(im_orig,im_tran,'montage');
```





borrado de objetos menor que minpixels

```
bwe = bwareaopen(im_tran,minpixels);  
figure, imshowpair(im_tran,bwe,'montage');
```



eliminado de pequeños discos de radio

```
se = strel('disk', rdisco);  
bwr = imopen(bwe, se);  
figure, imshowpair(bwe, bwr, 'montage');
```

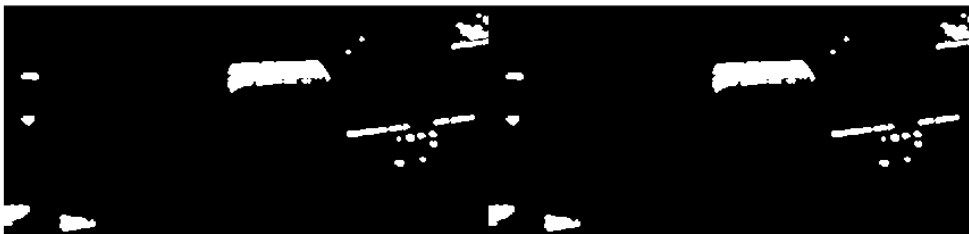
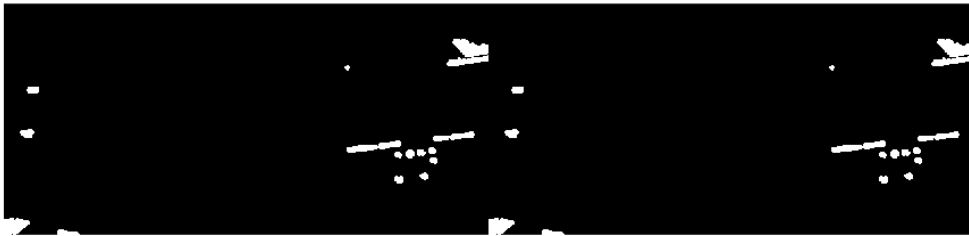


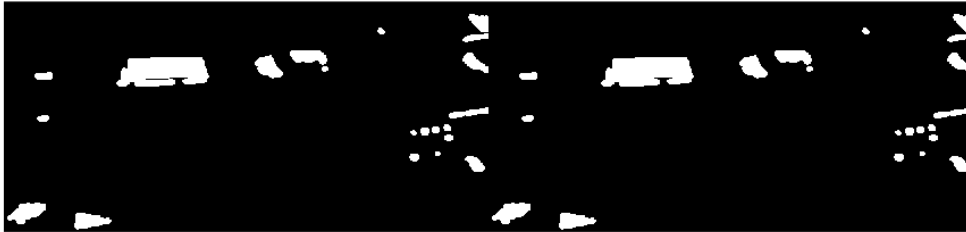


se rellenan huecos

el área encerrada por los bordes

```
bw_fill = imfill(bwr, 'holes');  
figure, imshowpair(bwr, bw_fill, 'montage');
```



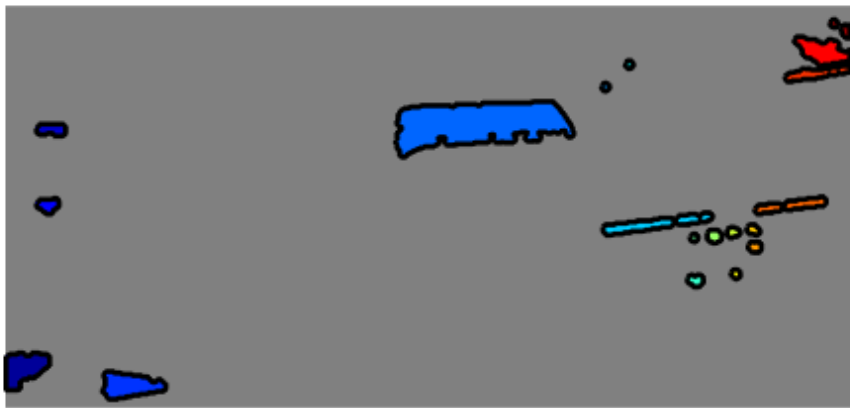


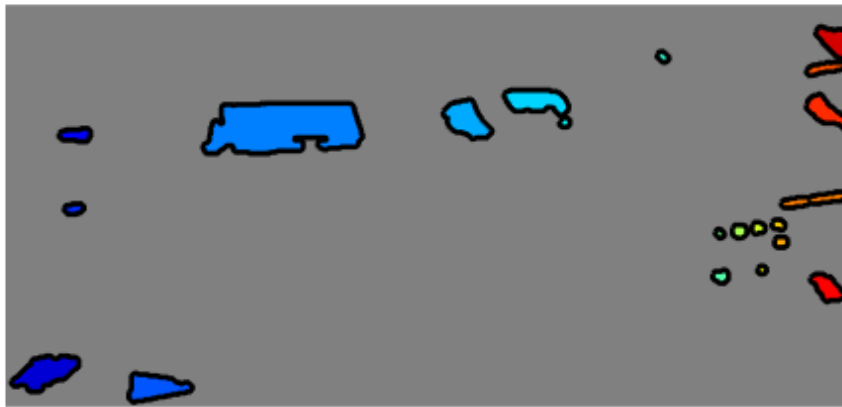
se calculan bordes

```
[B,L] = bwboundaries(bw_fill,'noholes');  
imshow(label2rgb(L,@jet,[.5 .5 .5]))  
hold on  
for k = 1:length(B)  
    boundary = B{k};  
    plot(boundary(:,2), boundary(:,1), 'w', 'Linewidth', 2)  
end  
bw_fillno = bwareaopen(bw_fill, 250);  
figure, imshow(bw_fillno)
```









se recopila información de las áreas

```
stats = regionprops(bw_fillno, {'Centroid','Area'});  
if ~isempty([stats.Area])  
    areaArray = [stats.Area];  
    for j = 1:length(areaArray)  
        if areaArray(j) >= minarea  
            c = c + 1;  
            hold on;  
            title(['coches = ' num2str(c)]);  
            drawnow;  
        end  
    end  
end  
end
```

coches = 1



coches = 2



coches = 4



coches = 5



```
end
```

resultado

```
disp('El numero de coches fue: ');  
disp(c);
```

```
El numero de coches fue:  
5
```

Published with MATLAB® R2018b