Proyecto IVC 2020/2021

En este proyecto se pedía el procesamiento de video usando el tracking de objetos o caras entre otras cosas, haciendo uso de Matlab y del “Image Proccessing ToolBox” y el "Computer Vision ToolBox”.

Tras investigar durante días a través de internet nos encontramos con bastantes códigos tanto buenos como malos y los estuvimos modificando para nuestro proyecto en concreto.

Tracking de objetos seleccionados

Hemos modificado un código de ejemplo de la página de matlab para el tracking de objetos añadiendo un imcrop al final para que haga el recorte en el objeto elegido, los valores del ancho y alto del recorte se pueden modificar a mano si el usuario lo requiere. El código del algoritmo consta de las siguientes partes:

1- Lectura de fichero de video y selección de la región a seguir.

```
videoReader = VideoReader('obj_1.mp4');
videoPlayer = vision.VideoPlayer('Position',[100,100,680,520]);
objectFrame = readFrame(videoReader);

figure; imshow(objectFrame);

objectRegion=round(getPosition(imrect));

```

2- Mostrar la caja del objeto y los puntos de interés del primer frame.

```
    bboxPoints = bbox2points(objectRegion(1, :));
    bboxPolygon = reshape(bboxPoints',1,[]);
objectImage = insertShape(objectFrame,'Polygon',bboxPolygon,'Color','red');
figure;
imshow(objectImage);
title('Caja del Objeto');
%roiRect = imrect;

points = detectMinEigenFeatures(im2gray(objectFrame),'ROI',objectRegion);
pointImage = insertMarker(objectFrame,points.Location,'+','Color','white');
figure;
imshow(pointImage);
title('Puntos de interes');
```

3- Inicializar el tracker y el video writer para generar el fichero de salida.

```
tracker = vision.PointTracker('MaxBidirectionalError', 50);
initialize(tracker,points.Location,objectFrame);


video = VideoWriter('yourvideo.mp4'); %create the video object
open(video); %open the file for writing
```

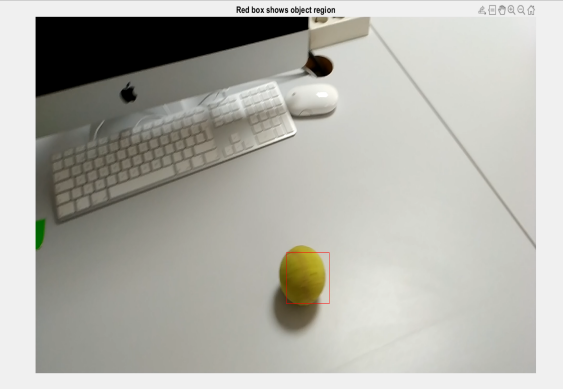
4- Hacemos un bucle frame a frame del video y detectamos los puntos de interés de cada frame, hacemos otro bucle para localizar los frames que están más abajo y a la izquierda y los valores máximos.

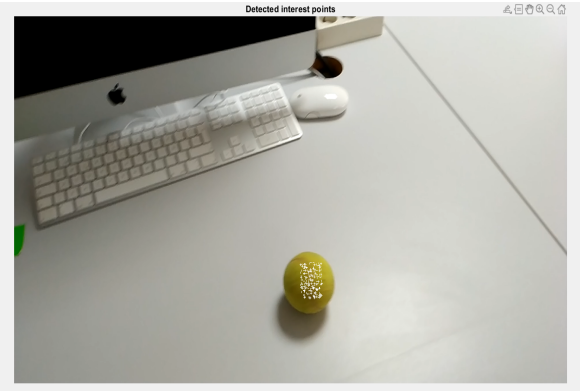
```
while hasFrame(videoReader)
        frame = readFrame(videoReader);
      [points,validity] = tracker(frame);

      pointSize= size(points);
      sizePoints= pointSize(1);
      %prueba= sizePoints/2;
      puntox=100000000;
      puntoy=100000000;
      ymax=-1;
      xmax=-1;
      recorte= size(frame);

      for i=1 :sizePoints
          if (puntox >points(i,1) && points(i,1)>recorte(2)/13)
              puntox=points(i,1);
          end
        if xmax < points(i,1)
            xmax=points(i,1);
        end
        if (puntoy > points(i,2))
            puntoy=points(i,2);
        end
        if ymax < points(i,2)
            ymax=points(i,2);
        end

      end

```

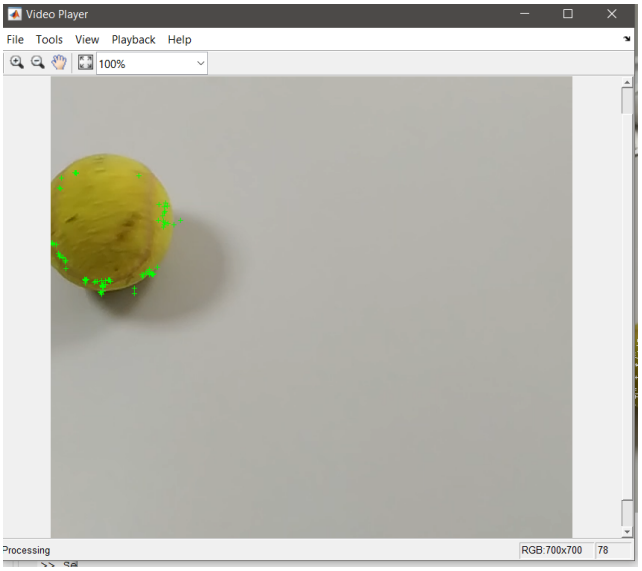
5- Insertamos marcas donde están los puntos de interés y hacemos el recorte desde donde puntox y puntoy, que son los valores mínimos de los puntos seleccionados con el ancho y alto que queramos (en este caso 700x700). A continuación se hará un resize de la imagen para evitar errores y se creará el video.

```
        out = insertMarker(frame,points(validity, :),'+');
        frame1 = imcrop(out, [puntox puntoy 700 700]);
        frameFinal= imresize(frame1,[700 700]);
        writeVideo(video,frameFinal);
      videoPlayer(frameFinal);
end

  release(videoPlayer);

    close(video);

```

Imágenes del algoritmo







Seguimiento automático de caras

1.Lectura de fichero de video, creación e inicialización de variables.

```
% Creamos el detector de caras
faceDetector = vision.CascadeObjectDetector();

% Creamos el tracker de puntos
pointTracker = vision.PointTracker('MaxBidirectionalError', 2);

% Leemos y alamcenamos el video
cam =  VideoReader('caras1.avi');

% Capturamos un fotograma para obtener su tamaño
videoFrame = readFrame(cam);
frameSize = size(videoFrame);

% Creamos el objeto videoPlayer
videoPlayer = vision.VideoPlayer('Position', [100 100 [frameSize(2), frameSize(1)]+30]);


% Inicializamos las variables
runLoop = true;
numPts = 0;
```

2.Bucle mientras la ventana esté abierta

```
while runLoop

    % Lectura del frame y binarización para detección de puntos
    videoFrame = readFrame(cam);
    videoFrameGray = rgb2gray(videoFrame);
```

3. Condición If dentro del bucle, donde entra inicialmente, ya que inicializamos el número de puntos a 0, y entrará cuando el número de puntos sea menor a 10.En el siguiente if chequea que la detección de la cara existe.

```
    if numPts < 10
        % Modo de detección
        bbox = faceDetector.step(videoFrameGray);

        if ~isempty(bbox)
            % Encontramos los puntos dentro de la región detectada
            points = detectMinEigenFeatures(videoFrameGray, 'ROI', bbox(1, :));

            % Volvemos a inicializar el tracker de puntos
            xyPoints = points.Location;
            numPts = size(xyPoints,1);
            release(pointTracker);
            initialize(pointTracker, xyPoints, videoFrameGray);

            % Guardamos una copia de los puntos anteriores
            oldPoints = xyPoints;

            % Convertimos el rectangulo representado por [ejeX, ejeY, ancho,
            % alto] en una matriz de N x 2.
            % Esto lo hacemos para poder transformar el cuadro de selección para la
            % orientación de la cara
            bboxPoints = bbox2points(bbox(1, :));

            %Convertimos las esquinas del cuadro en [x1 y1 x2 y2 x3 y3 x4 y4]
            % Esto es necesario para poder hacer el insertShape
            bboxPolygon = reshape(bboxPoints', 1, []);

            % Insertamos el cuadro alrededor de la cara detectada
            videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon, 'LineWidth', 3);

            % Mostramos los puntos detectados
            videoFrame = insertMarker(videoFrame, xyPoints, '+', 'Color', 'white');
        end
```

4. Condición else, donde entrará el algoritmo de detección de caras siempre que el número de puntos detectados sea mayor 10.

```
    else
        % Tracking
        [xyPoints, isFound] = step(pointTracker, videoFrameGray);
        visiblePoints = xyPoints(isFound, :);
        oldInliers = oldPoints(isFound, :);

        numPts = size(visiblePoints, 1);

        if numPts >= 10
            % Estimamos la tranformación geométrica entre los puntos
            % antiguos y los nuevos
            [xform, inlierIdx] = estimateGeometricTransform2D(...
                oldInliers, visiblePoints, 'similarity', 'MaxDistance', 4);
            oldInliers    = oldInliers(inlierIdx, :);
            visiblePoints = visiblePoints(inlierIdx, :);

            % Aplicamos la transformación al cuadro de selección
            bboxPoints = transformPointsForward(xform, bboxPoints);

            % Como hicimos anteriormente convertimos las esquinas del cuadro en [x1 y1 x2 y2 x3 y3 x4 y4]
            % Esto es necesario para poder hacer el insertShape
            bboxPolygon = reshape(bboxPoints', 1, []);

            % Insertamos el cuadro alrededor de la cara detectada
            videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon, 'LineWidth', 3);

            % Mostramos los puntos detectados
            videoFrame = insertMarker(videoFrame, visiblePoints, '+', 'Color', 'white');

            % Reseteamos los puntos
            oldPoints = visiblePoints;
            setPoints(pointTracker, oldPoints);
        end

    end
```

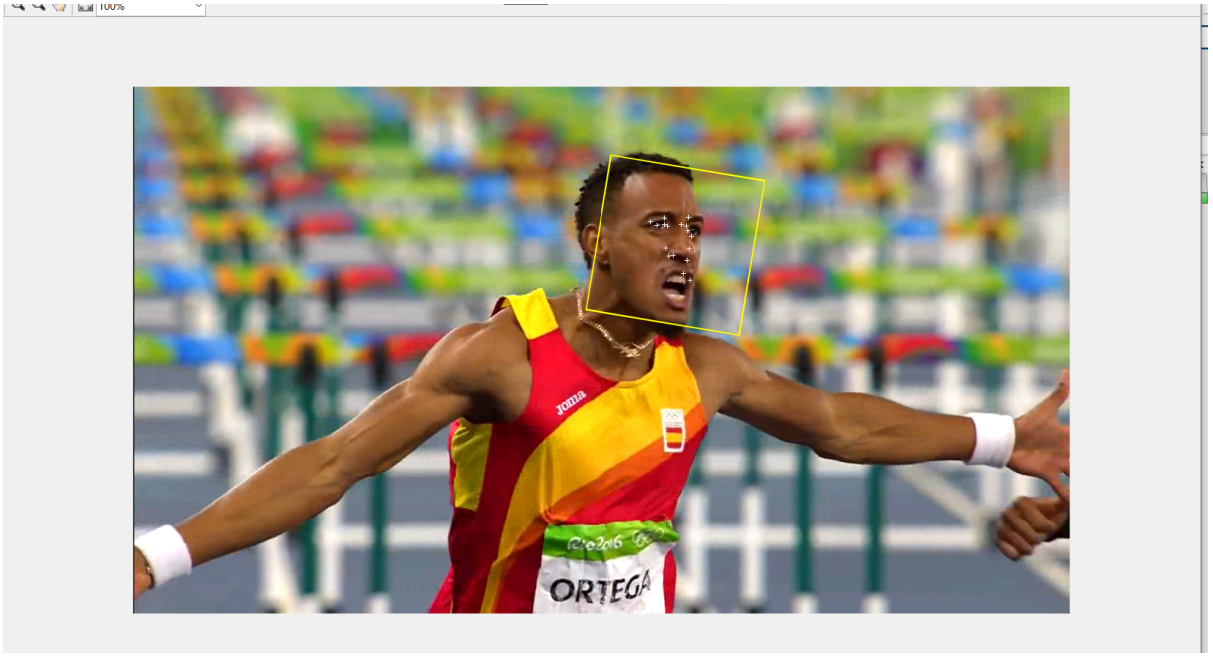
5. Mostramos el fotograma de vídeo correspondiente y comprobamos si se ha cerrado la ventana para continuar con el bucle de detección o no.

```
98          % Mostramos el fotograma de video correspondiente
99 -        step(videoPlayer, videoFrame);
100
101         % Comprobamos si de ha cerrado la ventana del reproductor de video
102 -        runLoop = isOpen(videoPlayer);
103 -    end
```

6. Una vez se salga del bucle, se limpian los objetos utilizados.

```
105     % Limpiamos los objetos
106 -   clear cam;
107 -   release(videoPlayer);
108 -   release(pointTracker);
109 -   release(faceDetector);
```

Imagen de ejecución:



Enlace videos: https://drive.google.com/drive/folders/1J5PSU-aAOfYgPlrn17R4v_j0r2-X01sG?usp=sharing

Eduardo Puentes Garay - 51254006R
Javier Martín Gómez - 74016116T
Alejandro Alcaraz Sánchez - 49224776S