

```

% Per a tots els vídeos
videoNames =dir("TinyTLP\");
nFrames = 100;
OverlappingRates = zeros([50, nFrames]);
for i=1:50
    name = videoNames(i).name;
    OverlappingRates(i,:) = ORatioCompHistrogrames(name,nFrames);
end

% Per a una selecció de vídeos
videoNames = {'Alladin','PolarBear1'};
nFrames = 100;
OverlappingRates = zeros([size(videoNames,2), nFrames]);
for i=1:size(videoNames,2)
    name = videoNames{i};
    OverlappingRates(i,:) = ORatioCompHistrogrames(name,nFrames);
end

function ORatio = ORatioCompHistrogrames(videoName, nFrames)
    close all
    hold off

    % Obtenir nom de les imatges i del fitxer groundtruth_rect
    path1 = strcat('./TinyTLP/', videoName);
    path11 = strcat(path1, '/groundtruth_rect.txt');
    path2 = strcat(path1, '/img/*.jpg');

    % Obtenir Bounding Boxes Òptimes
    BB = importdata(path11);
    Idir = dir(path2);

    figure
    hold on

    % Retall de l'objecte a rastrear
    previousBB = BB(1,2:5);
    filename = horzcat(Idir(1).folder, '/', Idir(1).name);
    I = imread(filename);
    previousCrop = imcrop(I,previousBB);

    % Vector de l'overlapping ratio de cada frame del vídeo
    ORatio = zeros(1,nFrames);

    for i=1:nFrames
        filename = horzcat(Idir(i).folder, '/', Idir(i).name);
        I = imread(filename);

        % Inicialitzem amb la posició actual
        bestBB = previousBB;
        bestCrop = imcrop(I, bestBB);
        min_diff = computeDiff(previousCrop,bestCrop);

        % Càlcul del nombre de píxels a comprovar
        i_step = floor(BB(i,4)/25);
        j_step = floor(BB(i,5)/25);

        % Iterar pels punts de la bounding box
        for i_cord = 1:i_step:previousBB(3)
            for j_cord = 1:j_step:previousBB(4)

                % Calcular distància respecte el retall inicial
                [tlx,tly] = topLeftCoordinate([previousBB(1)+i_cord,previousBB(2)+j_cord],previousBB(3),previousBB(4));
                possibleBB = [tlx,tly,previousBB(3),previousBB(4)];
                Crop = imcrop(I, possibleBB);
                diff = computeDiff(previousCrop,Crop);

                % Comprovar que la caixa sigui la de menys distància
                if diff < min_diff
                    min_diff = diff;
                    bestBB = possibleBB;
                end
            end
        end

        % Càlcul overlapping ratio de la millor Bounding Box
        ORatio(1,i) = bboxOverlapRatio([bestBB(1),bestBB(2),BB(i,4),BB(i,5)],BB(i,2:5));

        %Mostrar les imatges amb les caixes
        imshow(I)
        rectangle('Position',[bestBB(1),bestBB(2),BB(i,4),BB(i,5)], 'EdgeColor','red');
        rectangle('Position',BB(i,2:5), 'EdgeColor','yellow');
        drawnow

        previousBB = bestBB;
    end
end
end

```

```

% Retorna la distància entre les dues imatges
function diff = computeDiff(I1,I2)
    if size(I1) == size(I2)
        hist1 = calculateHistogram(I1);
        hist2 = calculateHistogram(I2);
        diff = compareHistograms(hist1,hist2);
    else
        diff = 10000000000;
    end
end

% Retrona les coordenades de la cantonada superior esquerra
function [minx, miny] = topLeftCoordinate(centralPoint, width, height)
    minx = centralPoint(1) - width/2;
    miny = centralPoint(2) - height/2;
end

% Retorna l'histograma de color normalitzat en dues dimensions
function [hist] = calculateHistogram(im)
    I = sum(rgb2gray(im),'all')/(size(im,1)*size(im,2))/255;
    im_norm = (double(im)/255)/(3*I);
    hist = zeros([16,16]);
    for i = 1:size(im, 1)
        for j = 1:size(im, 2)
            r = min(floor(15*im_norm(i,j,1)) + 1, 16);
            g = min(floor(15*im_norm(i,j,2)) + 1, 16);
            hist(r,g) = hist(r,g) + 1;
        end
    end
end

% Retorna la distància chi-quadrada entre els dos histogrames
function [s] = compareHistograms(hist1, hist2)
    s = 0;
    for i = 1:16
        for j = 1:16
            if (hist1(i,j) + hist2(i,j) > 0)
                s = s + (hist1(i,j) - hist2(i,j))*(hist1(i,j) - hist2(i,j))/(hist1(i,j) + hist2(i,j));
            end
        end
    end
end

```