Multi-Sensors Fusion and Tracking
Vibot Program 2018-2019

Ioannis Tsagatakis

1st Assignment: Background Subtraction

October 23, 2018

I haven't manage to implement all methods in the given amount of time and with my limited computational resources. The code and the result videos van be found in the following gtihub repository `https://github.com/jtsagata/image_bg_removal`. Please check there for the code and maybe imporoved versions of the code and the report.

1 Data Preperation

To speed up development the image files are read only once and saved as `'mat'` files. The code for the `'gtruth.mat'` file is given bellow. The code for the other image sequences is simmilar and it is on the github repository.

```
imPath = 'highway/groundtruth'; imExt = 'png';
SKIP_IMAGES=470;

% check if directory and files exist
if isdir(imPath) == 0
    error('USER ERROR : The image directory does not exist');
end

filearray = dir([imPath filesep '*.' imExt]);
NumImages = size(filearray,1);
if NumImages < 0
    error('No image in the directory');
end

disp('Loading image files from the video sequence, please be patient
   ...');
% Get image parameters
imgname = [imPath filesep filearray(1).name]; % get image name
I = imread(imgname); % read the 1st image and pick its size
VIDEO_WIDTH = size(I,2);
VIDEO_HEIGHT = size(I,1);

ImSeq = zeros(VIDEO_HEIGHT, VIDEO_WIDTH, NumImages-SKIP_IMAGES);
for i=1:NumImages
```

```
    imgname = [imPath filesep filearray(i).name]; % get image name
    if i >= SKIP_IMAGES
        ImSeq(:,:,i) = imread(imgname); % load image
    end
end
disp(' ... OK!');

gTruth = uint8(ImSeq);
save('gtruth.mat', 'gTruth','-v7.3');
disp(' Saving ... DONE!');
whos('-file','gtruth.mat')
```

2 Frame differencing

The following matlab function implements the *Frame differencing* method.

```
function video_noBg = bgsub_frame_diff(ImSeq,skip_frames, threshold)
%bgsub_frame_diff Remove background using frame differencing method.
%    ImSeq: the image sequence
%    skip_frames: The fisrt frames used for averaging
%    threshold: The threshold for image to beleong to background
%  RETURNS:
%    The image sequence without the background

    bgAvg = median(ImSeq,skip_frames);
    video_noBg = zeros(size(ImSeq),'like',ImSeq);

    idx = abs(ImSeq - bgAvg)> threshold;
    video_noBg(idx) = ImSeq(idx);

end
```

To demonstrate we need a fuction to annotate the video with data like frame number and location, and with a bounding box around the main moving object of the scene.

```
function outVideo = anotate_video_box(inVideo)
%anotate_video Add Frame number to a sequence of images
%   This will always return a color RGB image

    width = size(inVideo,1);
    height = size(inVideo,2);

    if size(size(inVideo),2) == 3
        monochrome=true;
        frames = size(inVideo,3);
    else
        monochrome=false;
        frames = size(inVideo,4);
    end

    outVideo = zeros(width,height,3,frames);
```

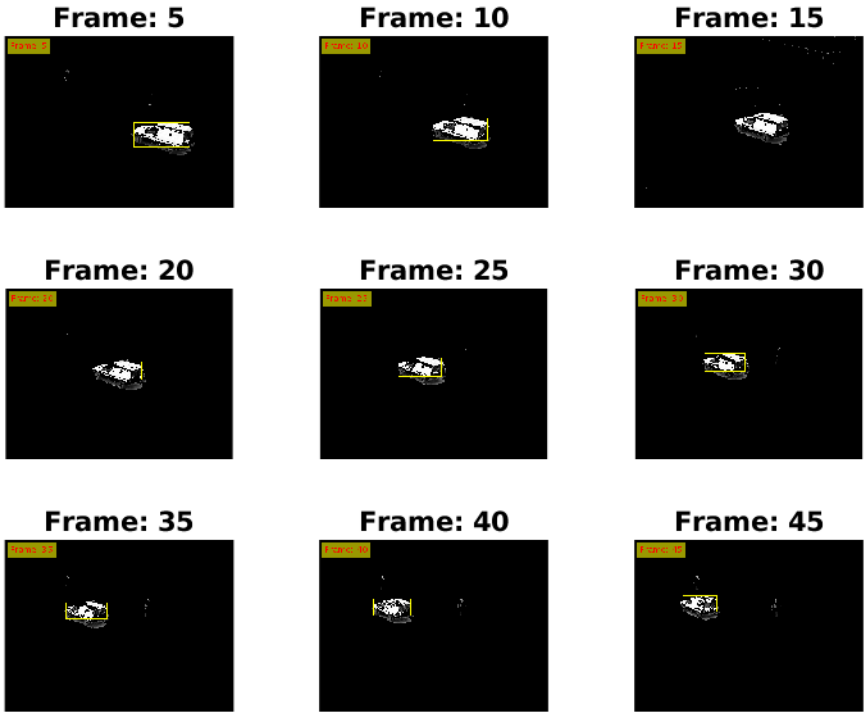

Figure 1: Frame differencing for the cars sequence

```
for k = 1:frames
    if monochrome
        grayImage = inVideo(:,:,k);
        rgbImage = cat(3, mat2gray(grayImage), mat2gray(
           grayImage), mat2gray(grayImage));
    else
        rgbImage = inVideo(:,:,k);
        grayImage = im2bw(rgbImage,0.5);
    end


    % Add the Box
    st = regionprops(grayImage, 'BoundingBox', 'Area' );
    areas=[st.Area];
    if sum(size(areas)) > 0
        [maxArea, indexOfMax] = max(areas);
        bb = st(indexOfMax).BoundingBox;
        rgbImage=insertShape(rgbImage,'rectangle',bb);
    end

    % Add a frame number
    text_str = ['Frame: ' num2str(k,'%4d') ];
```

```
        rgbImage = insertText(rgbImage, [5,25], text_str, '
           AnchorPoint','LeftBottom','TextColor','red');

        outVideo(:,:,:,k) = rgbImage;
    end

end
```

And a code to save the video sequence as a playable video on hard disk.

```
function  videoSave(file_name,video,frame_rate)
% videoSave Save an image sequence as video
%  Each image must be an RGB image

    v = VideoWriter(file_name);
    v.FrameRate = frame_rate;
    open(v);

    for k = 1:size(video,4)
        frame = video(:,:,:,k);
        writeVideo(v,frame);
    end

    close(v);
end
```

Finnaly a fuction is created to save some frames in png file format.

```
function video_to_img_seq(videoIn,fname)
%video_to_img_seq Save some frame from video in to an image file

frames = size(videoIn,4);
every_frame = floor(frames/9);

h = figure;
for k=1:9
    subplot(3,3,k);
    frame_no = k*every_frame;
    imshow( videoIn(:,:,:,frame_no));
    text_str = ['Frame: ' num2str(frame_no,'%4d') ];
    title(text_str);
end

saveas(gcf,fname)
close(h)
end
```

2.1 Cars Sequence

Let's apply the algorithm to the `'cars'` sequence.

```
clearvars;
close all
```

```
load('../Data/cars.mat');
video_noBg = bgsub_frame_diff(ImSeq, 3, 60);
video_annot = anotate_video_box(video_noBg);

videoSave('../Videos/bgsub_framediff_cars.avi',video_annot,10);
video_to_img_seq(video_annot,'../Videos/bgsub_framediff_cars.png');
```

Some frames is given in Figure 2 on page 3.

2.2 Highway Sequence

The highway sequence is in color. So we modify the algorithm. An Eucledian metric is used to estimate the threshold of a given pixel form the background. Also the confusion matrix is calculated. After many revisions the final code is given bellow

```
clearvars;
load('../Data/highway8.mat');
load('../Data/gtruth.mat');

close all

% Split into initial background estimation and work videos
skip_frames = 470;
first_frames = video(:,:,:,1:skip_frames-1);
video_frames = video(:,:,:,skip_frames:size(video,4));
gtruth_frames = gTruth(:,:,skip_frames:size(video,4));

% Storage for montage Video
[width,height,depth,frames]=size(video_frames);
video_montage = zeros([width*2,height*2,depth,frames],'uint8');

% Initial background
cur_bgr = median(double(first_frames),4);

% Parameters
threshold = 110;
alpha = 0.05;
im2bw_level = 0.1;
track_objects=6;
maxArea = 40;

max_frames = 100000;

s_TP=0; s_FP=0; s_FN=0;
for k=1:min(size(video_frames,4),max_frames)
    cur_img = double(video_frames(:,:,:,k));

    dist_mat = sqrt(sum((cur_img - cur_bgr).^2, 3));
    is_onBG= (dist_mat>threshold);
    % Improve results a bit
    is_onBG=imfill(is_onBG,'holes');
```

```
new_frame_noBG = cur_img .* is_onBG;

curr_bgr_upd = new_frame_noBG .* alpha + cur_bgr .* (1-alpha);
cur_bgr = cur_bgr .* not(is_onBG) + curr_bgr_upd .* is_onBG;

% Calculate Confusion matrix
curr_gtruth = imbinarize(gtruth_frames(:,:,k));
m1=uint8(curr_gtruth(:));
m2=uint8(is_onBG(:));

%cm = confusionmat(m1,m2);
TP = sum(bsxfun(@(a,b) a==1 & b==1, m1, m2));
FP = sum(bsxfun(@(a,b) a==1 & b==0, m1, m2));
FN = sum(bsxfun(@(a,b) a==0 & b==1, m1, m2));
s_TP= s_TP+TP; s_FP= s_FP+FP; s_FN= s_FN+FN;
PREC= TP/(TP+FP);
REC = TP/(TP+FN);


% get the bounding boxes
im_bw = imbinarize(rgb2gray(new_frame_noBG), im2bw_level);
st = regionprops(im_bw, 'BoundingBox', 'Area' );

color_frame = uint8(cur_img);
[maxAreas, indexOfMaxes] = maxk([st.Area],track_objects);
for ob=1:min(track_objects, size(st,1))
    if  st(indexOfMaxes(ob)).Area > maxArea
        bb = st(indexOfMaxes(ob)).BoundingBox;
        % Add the bounding box
        color_frame=insertShape(color_frame,'rectangle',bb, '
           LineWidth', 2, 'Color', 'red');
        new_frame_noBG=insertShape(uint8(new_frame_noBG),'
           rectangle',bb, 'LineWidth', 2, 'Color', 'red');
    end
end

% Add a frame number
text_str = ['Frame: ' num2str(k+skip_frames,'%4d') ];
cur_img = insertText(uint8(cur_img), [5,25], text_str, '
   AnchorPoint','LeftBottom','TextColor','red');
text_str_stat = ['Prec: ' num2str(PREC,3) ' Rec: ' num2str(REC,
   3) ];
cur_img = insertText(cur_img, [5,230], text_str_stat, '
   AnchorPoint','LeftBottom','TextColor','red');

% Montage
video_montage(:,:,:,k) = [cur_img, new_frame_noBG; cur_bgr,
   color_frame];
```

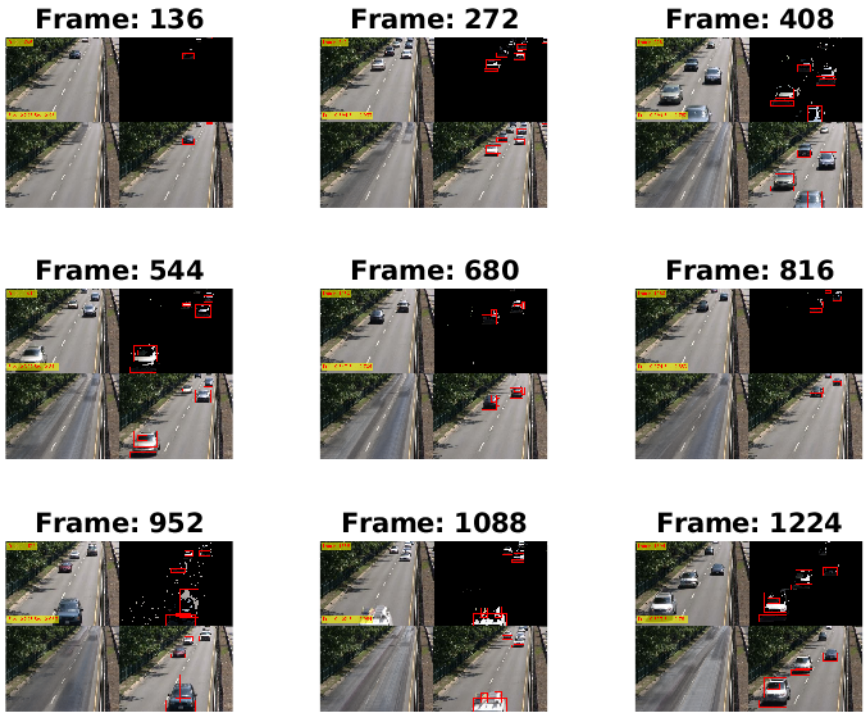

Figure 2: Frame differencing for the highway sequence

```
end

h=implay(video_montage);
h.Parent.Position = [100 100 700 550];

T_PREC= s_TP/(s_TP+s_FP);
T_REC = s_TP/(s_TP+s_FN);
T_FSCORE = 2 * T_PREC * T_REC / (T_PREC + T_REC);

fileID = fopen('../Videos/highway_frame_diff.txt','w');
fprintf(fileID,'Precision: %1.4f\n',T_PREC);
fprintf(fileID,'Recall:    %1.4f\n',T_REC);
fprintf(fileID,'F-Score:   %1.4f\n',T_FSCORE);
fclose(fileID);



videoSave('../Videos/highway_frame_diff.avi',video_montage,10);
video_to_img_seq(video_montage,'../Videos/highway_frame_diff.png');
```

Some frames is given in Figure 2.2 on page 7.

The precission, recall an F-score of the method is given bellow:

```
Precision: 0.3416
Recall:    0.7933
F-Score:   0.4775
```

3 The Running Average Gaussian Method

3.1 Cars Sequence

The code that implements the Running Average Gaussian Method on the cars sequence is given bellow

```
clearvars;
close all

% Load image sequence
load('../Data/cars.mat');
[height, width, frames]= size(ImSeq);

% CONTROL VARIABLES
threshold = 2.5;
alpha = 0.01;

frame_rate = 10;

% Lambda (acessible from cli also)
implay8 = @(V) implay(uint8(V), frame_rate);


%% START PROCESSING

% Estimate from first 'skip_frames' frames


% Calculate rest of frames
ImSeq_noBg = zeros([height, width, frames],'like',ImSeq);
ImSeq_meds = zeros([height, width, frames],'like',ImSeq);
ImSeq_vars = zeros([height, width, frames],'like',ImSeq);

% First Frame
var_mode = 1;
skip_frames = 3;
ImSeq_meds(:, :, 1) = ImSeq(:,:,1);
ImSeq_vars(:, :, 1) = var(ImSeq(:,:,1:skip_frames),var_mode,3);
% ImSeq_vars(:, :, 1) = 1;

for k=2:frames
    ImSeq_meds(:,:,k) = alpha .* ImSeq(:, :, k) + (1-alpha) .*
       ImSeq_meds(:,:,k-1);
    d_sq = abs( ImSeq(:, :, k) - ImSeq_meds(:,:,k)).^2 ;
```

```
    ImSeq_vars(:,:,k) = alpha .* d_sq  + (1-alpha) .* ImSeq_vars
       (:,:,k-1);

    crit = abs( (ImSeq(:,:,k) - ImSeq_meds(:,:,k)) ./ sqrt(
       ImSeq_vars(:,:,k)) );
    idx = crit > threshold;

    frame = zeros([height, width]);
    img = ImSeq(:, :, k);
    frame(idx)= img(idx);

    ImSeq_noBg(:,:,k) = frame;
end

%% RESULTS

% Show Video
implay8(ImSeq_noBg);
video_annot = anotate_video_box(uint8(ImSeq_noBg));

videoSave('../Videos/a3_gaussian.avi',video_annot,10);
video_to_img_seq(video_annot,'../Videos/a3_gaussian.png');
```

Some frames is given in Figure 3.1 on page 10.

3.2 Highway Sequence

Bellow is the code that implements the running average gaussian method for the highway sequence:

```
clearvars;
load('../Data/highway8.mat');
load('../Data/gtruth.mat');

close all

% Split into initial background estimation and work videos
skip_frames = 470;
first_frames = video(:,:,:,1:skip_frames-1);
video_frames = video(:,:,:,skip_frames:size(video,4));
gtruth_frames = gTruth(:,:,skip_frames:size(video,4));

% Storage for montage Video
[width,height,depth,frames]=size(video_frames);
video_montage = zeros([width*2,height*2,depth,frames],'uint8');

% Initial values
var_mode = 1;
cur_meds = median(double(first_frames),4);
```

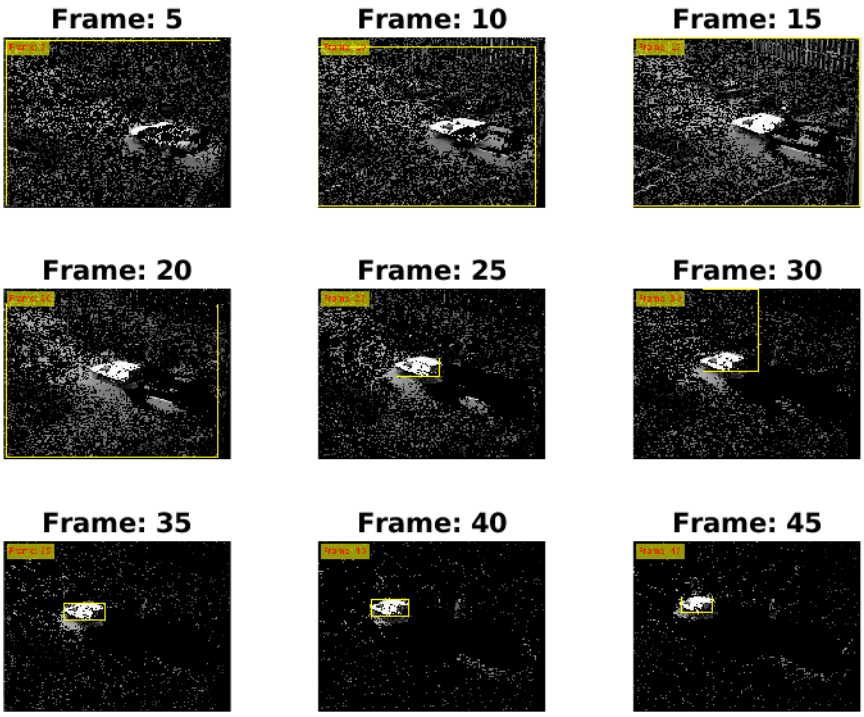

Figure 3: Running Average Gaussian for the cars sequence

```
cur_vars = var(double(first_frames),var_mode,4);

% Parameters
threshold = 2.5;
alpha = 0.01;
im2bw_level = 0.1;
track_objects=6;
maxArea = 40;

max_frames = 50000;
s_TP=0; s_FP=0; s_FN=0;
for k=1:min(size(video_frames,4),max_frames)
    cur_img = double(video_frames(:,:,:,k));

    cur_meds = alpha .* cur_img + (1-alpha) .* cur_meds;
    d_sq     = abs(cur_img - cur_meds).^2 ;
    cur_vars = alpha .* d_sq  + (1-alpha) .*cur_vars;

    crit = abs( (cur_img -cur_meds) ./ sqrt(cur_vars) );
    crit_norm = sqrt(sum(crit,3));
    is_onBG= (crit_norm>threshold);

    % Improve results a bit
    is_onBG=imfill(is_onBG,'holes');
```

```
    new_frame_noBG = cur_img .* is_onBG;


    % Calculate Confusion matrix
    curr_gtruth = imbinarize(gtruth_frames(:,:,k));
    m1=uint8(curr_gtruth(:));
    m2=uint8(is_onBG(:));

    %cm = confusionmat(m1,m2);
    TP = sum(bsxfun(@(a,b) a==1 & b==1, m1, m2));
    FP = sum(bsxfun(@(a,b) a==1 & b==0, m1, m2));
    FN = sum(bsxfun(@(a,b) a==0 & b==1, m1, m2));
    s_TP= s_TP+TP; s_FP= s_FP+FP; s_FN= s_FN+FN;
    PREC= TP/(TP+FP);
    REC = TP/(TP+FN);


    % get the bounding boxes
    im_bw = imbinarize(rgb2gray(new_frame_noBG), im2bw_level);
    st = regionprops(im_bw, 'BoundingBox', 'Area' );

    color_frame = uint8(cur_img);
    [maxAreas, indexOfMaxes] = maxk([st.Area],track_objects);
    for ob=1:min(track_objects, size(st,1))
        if  st(indexOfMaxes(ob)).Area > maxArea
            bb = st(indexOfMaxes(ob)).BoundingBox;
            % Add the bounding box
            color_frame=insertShape(color_frame,'rectangle',bb, '
               LineWidth', 2, 'Color', 'red');
            new_frame_noBG=insertShape(uint8(new_frame_noBG),'
               rectangle',bb, 'LineWidth', 2, 'Color', 'red');
        end
    end

    % Add a frame number
    text_str = ['Frame: ' num2str(k+skip_frames,'%4d') ];
    cur_img = insertText(uint8(cur_img), [5,25], text_str, '
       AnchorPoint','LeftBottom','TextColor','red');
    text_str_stat = ['Prec: ' num2str(PREC,3) ' Rec: ' num2str(REC,
       3) ];
    cur_img = insertText(cur_img, [5,230], text_str_stat, '
       AnchorPoint','LeftBottom','TextColor','red');

    % Montage
    video_montage(:,:,:,k) = [cur_img, new_frame_noBG; uint8(
       cur_vars), color_frame];


end
```

Figure 4: Running Average Gaussian for the highway sequence

```
h=implay(video_montage);
h.Parent.Position = [100 100 700 550];

T_PREC= s_TP/(s_TP+s_FP);
T_REC = s_TP/(s_TP+s_FN);
T_FSCORE = 2 * T_PREC * T_REC / (T_PREC + T_REC);

fileID = fopen('../Videos/highway_avg_gauss.txt','w');
fprintf(fileID,'Precision: %1.4f\n',T_PREC);
fprintf(fileID,'Recall:    %1.4f\n',T_REC);
fprintf(fileID,'F-Score:   %1.4f\n',T_FSCORE);
fclose(fileID);



videoSave('../Videos/highway_avg_gauss.avi',video_montage,10);
video_to_img_seq(video_montage,'../Videos/highway_avg_gauss.png');
```

Some frames is given in Figure 3.2 on page 12.

The precission, recall an F-score of the method is given bellow:

```
Precision: 0.2898
Recall:    0.5665
F-Score:   0.3834
```

4 Eigen Background

Code for the eigen background method.

```
clearvars;
close all

% Load image sequence
load('../Data/cars.mat');

% Trick to works with bw and color images
% https://stackoverflow.com/questions/19955653/matlab-last-dimension
   -access-on-ndimensions-matrix
otherdims = repmat({':'},1,ndims(ImSeq)-1);


% Resize Video
%   The SVD matrix is very large and bigger than
%    my computer memory
scale =0.5;
s = size(ImSeq);
video_small=zeros([floor(s(1:end-1)*scale) s(end)]);
for k=1:size(ImSeq,ndims(ImSeq))
    video_small(otherdims{:},k) = imresize(ImSeq(otherdims{:},k),
       scale);
end
frames=size(video_small,ndims(video_small));

% Calculate the mean vector
mean_vector = reshape(mean(video_small,ndims(video_small)) ,[] ,1);

% Calculate X matrix
videoAsCols = reshape(video_small,[],frames);
videoAsCols_norm = videoAsCols - repmat(mean_vector,1, frames);

% SVD
[U,~,~]=svd(videoAsCols_norm);

% Get first ratio columns as eigen background
keep_cols = 10;
Uk=U(:,1:keep_cols);
Uk_t=transpose(Uk);
clear 'U'

video_noBg = zeros(size(video_small));
threshold =  20;
```

```
for k=1:frames
    curr_frame = video_small(otherdims{:},k);
    curr_frame_vec = reshape(curr_frame,[],1);
    p = Uk_t * (curr_frame_vec-mean_vector);
    y = Uk*p + mean_vector;

    background=reshape(y,size(curr_frame));
    dif = abs(background-curr_frame);
    idx = dif > threshold;

    new_frame = zeros(size(curr_frame));
    new_frame(idx) = curr_frame(idx);

    video_noBg(otherdims{:},k)= new_frame;
    %video_noBg(otherdims{:},k)= background;
end



%implay8(ImSeq_noBg);
video_annot = anotate_video_box(uint8(video_noBg));

videoSave('../Videos/a5_eigen.avi',video_annot,10);
video_to_img_seq(video_annot,'../Videos/a5_eigen.png');
```

Some frames is given in Figure 4 on page 15.

Tools used: `Matlab`, LaTeX.

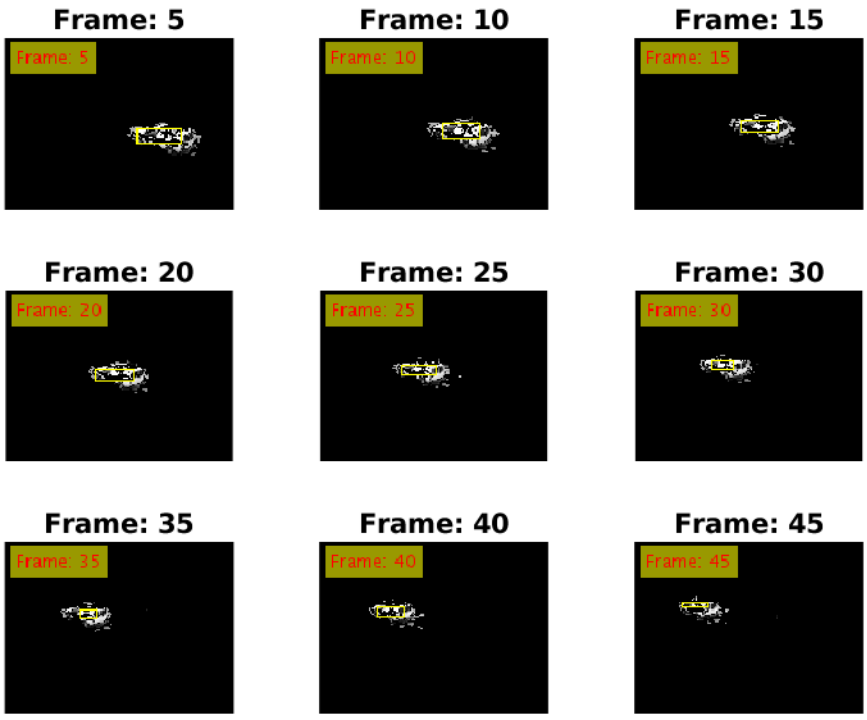

Figure 5: Eigen Background for the cars sequence