Histogram-based object tracking

Iuliia Alekseenko and Ricardo Luque

Abstract— **Object tracking is an open computer vision problem for which various methods have been proposed. The main objectives of this laboratory session are developing a framework for single-object tracking based on histograms. To model the object appearance representation, color features and gradient features represented by HOG descriptors were used as two widely-used approaches. Later, the fusion of these features is performed.**

The algorithms are implemented using C++ and Open CV and based on the two research papers [1] and [2]. This report represents the detailed evaluation of the algorithms by analyzing the strengths and weaknesses of the different features that compose the pipeline of the algorithm.

I. INTRODUCTION

Object tracking is an important area of image processing fields due to the ability of acquiring and providing relevant information of the real world. In fact, tracking impacts different several applications such as video surveillance, human-computer interaction, robot navigation and much more. In rough terms, object tracking refers to a task in which an initial object or set of objects detections are identified with a unique ID per object and carried through the whole video sequence while maintaining the same ID assignment.

To achieve this, tracking algorithms work through different frames in video sequences, with the aim of identifying an object or objects of relevance through *time*. In other words, tracking works in the way that it monitors an object throughout the video sequence.

Tracking can be performed through different methods, such as Kalman filtering, which was covered in the last laboratory practice. In this approach, a model for the objects' behavior through space is specified (such as constant velocity or constant acceleration), so that prior predictions can be performed based on this model. Therefore, it consisted on a process of propagating a corrected state prediction based on a prior corrected state and a prediction correction obtained through observations acquired in the scenes in context. Although this is quite a robust model, several other tracking approaches exist, based on different features to fetch and analyze from the Image Space.

With all the above mentioned, three approaches for object tracking are implemented, tested and analyzed in this lab practice. The first one refers to a color-feature based histograming, the second one to a gradient-based histograming and the last one to a mixture between the two previous methods.

The specifics will be described across this lab report, in which theoretical explanation, tests and analysis are provided to support the experimental results obtained.

II. METHOD

The following section explains the theoretical information behind the implementation, the experimental process and their respective analysis. These concepts are mentioned throughout the whole laboratory session.

A. *Histograms*

Histogram-based models refer to modeling a probability density function by extracting feature values from the image space and placing them into buckets or bins. Roughly speaking, the entire range of values for a given feature is divided into buckets, for which each bucket makes reference to a small interval as a part of an entire range of values. The histogram is then constructed by counting how many times a feature falls within each bin. Once the histogram is obtained, it is normalized in order to obtain a probability density function which, in our object tracking scenario, models the pixels information of the object to be tracked within a *BLOB*[1] given a specified feature. The following picture, Figure 1, depicts two images for which their corresponding histograms have been constructed.

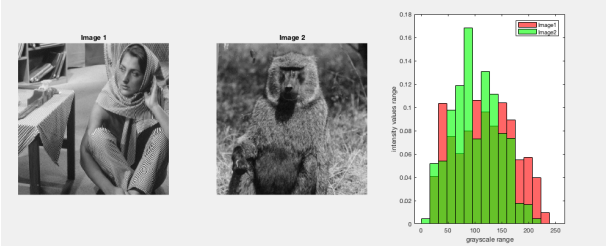

Fig. 1. **Histograms** of two images and showing their respective histograms and the overlap between them.

In this specific lab practice, low level features are used for simplicity, which include color and gradient information of images. Although the particular implementation per feature vary, the general pipeline for applying histograms-based model follows:

- **Feature extraction:** Features on the image space are extracted. Either gradient or color information.
- **Initialization step:** The initial position is annotated as a *BLOB* for the first frame, for which the features are extracted (within this blob area). Then, the histogram is computed and stored for modeling the target object of interest. This target object is later used in the matching process. As mentioned, the initial position is annotated,

[1]A **Binary Large Object** (BLOB) refers to a collection of binary data which contains an identified object in the scene

whereas the following frames use the previous best candidate prediction until the end of the video sequence is reached.

- **Candidates generation:** possible candidates for the new frame are generated around the last state.
- **Feature extraction on the given blobs:** is performed and histograms for each candidate are computed, they are normalized and the Probability Density function is obtained.
- **Similarity measure:** A measurement between each candidate's PDF and the target PDF is computed using the **Bhattacharyya distance**, so that each possible candidate's PDF is compared with the target's PDF.
- **Selecting the best candidate:** Choose the candidate with the most overlap between the Probability Density Functions, and assign the center location of the candidate as the new state on the space state.
- **Until reaching end:** Repeat process until all frames of the video sequence are covered.

B. *Candidates generation*

The candidates generation follows a grid search area approach. For this, a neighborhood area around the last frame's state is investigated, and the number of candidates is closely related to the step size or stride between each new candidate's center. Therefore, locations of the candidates are generated within this grid area. The proposed approach generates an area around the pixel in context. This means that a grid search area of $(2xneighborhood_{size} + 1)x(2xNeighborhood_{size} + 1)$ is computed with $(2xneighborhood_{size}/step_{size}) + 1)$ $x(2xneighborhood_{size}/step_{size} + 1))$ possible candidates. Once the locations are generated, a *Blob* of the target's *blob* size is generated for extracting features and putting them in bins. For instance, if the initial target has dimensions of *50x50* pixels, and a grid search area $neighborhood_{size} = 6$ with a $step_{size} = 2$ is generated, an area of 13x13 pixels is investigated in which $((2x6/2 + 1)x(2x6/2 + 1) = 49)$ possible candidate locations are provided. Then, a cropped part of size *50x50* pixels from the current frame is used around each candidate's center to compute the probability density function. This can be seen in Figure 2

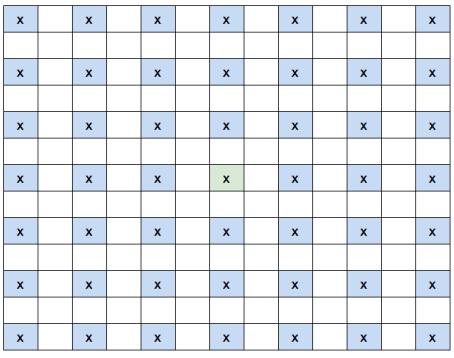

Fig. 2. **Grid Search Area** of a *neighborhood_size=6* and a *step_size=2*. 13x13 pixel area with 29 possible candidate locations

C. *Bhattacharyya distance*

In order to compare two Probability density functions several similarity functions such as *Canberra Distance* or *Histogram Intersection* can be applied, which provide information of the overlap between them. For this same purpose, the **Bhattacharyya distance** is used, so that we obtain the candidate whose probability density function (*PDF*) overlaps the most with the target's PDF. This candidate is then selected as the target's position, and used on the next frame. For instance, the **Bhattacharyya distance** could be computed between the two histograms shown if Figure 1, so that one scalar value is computed among different **PDFs**. The **Bhattacharyya distance** is defined as follows:

$$BC(p,q) = \sum_{i=1}^{n} \sqrt{(p_i q_i)} \tag{1}$$

For which, considering the samples *p* and *q*, *n* is the total number of bins, whereas q_i and p_i refer the i^{th} partion of the histogram or discrete probability density function in this case scenario.

D. *Color-based*

As mentioned earlier all three methods follow a similar approach. For color-based histograms the feature to be extracted from the image is color information. This information, however, can be extracted from different channels and in different color-spaces. The color information from the candidates is then placed in histograms. For instance, if *number_of_bins = 16* each bucket or bin has a range of 16 values for intensity values. The first bin collects information from 0-15, the second one from 16 to 31, and so on until 255 is reached. Therefore, this approach matches the candidates based on their overall color appearance. This is depicted in Figure 1, where two **PDFs** of two gray scale images have been computed.

It is important to mention that histogram information is invariant to translation and on the same plane rotation. However, it is not robust to scaling or in-plane rotations as they do not preserve space information.

E. *Gradient-based*

The second method, gradient-based histograms is based on the directional change in the intensity or color in an image. In fact image gradients are essential parts of image processing algorithms since they are used in different methods such as edge detection or other mid-level features. In this case scenario only the low-level feature is used, which refers to the intensity level difference between neighboring pixels. For this feature independent channels in RGB and HSV channel are used as well as the grey scale image itself. We can observe in Figure 3 the image gradient of the same two pictures for whose histograms were computed on the earlier subsection.

As a highlight, image gradient information is invariant to translation and on the same plane rotation. However, similarly to color information, it is not robust to scaling or in-plane rotations as they do not preserve space information.

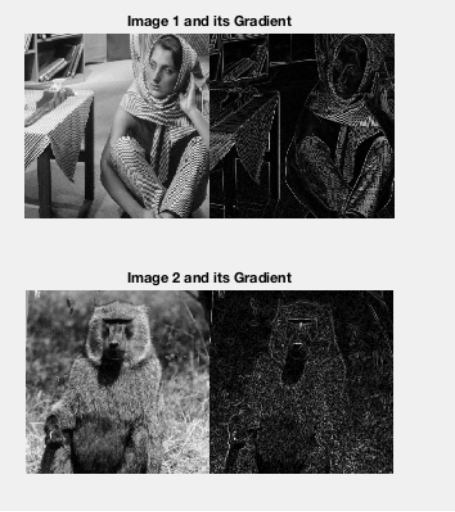

Fig. 3. **Color and Gradient information**. Two gray scale images and their respective gradient information

F. Fusion of features

The third method, **fusion of features**, refers to mixing both previously described approaches. This approach gives a score for each feature and a final score is obtained to select the best candidate of all candidates generated.

III. EXPERIMENTAL METHODOLOGY

The way the experiments were conducted for evaluation of the implementation is presented in this section. First, all the video sequences presented in Data (V), were converted to HSV and RGB color spaces, for which the Hue (**H**), and Saturation (**S**) were used for the first color space, while all three channels Red (**R**), Green (**G**) and Blue (**B**) for the RGB color space respectively. Additionally, a gray image of the frames was used in the gradient-based and fusion approaches.

Although several combinations were tried before deciding the parameters of evaluation, 3 main different configurations were chosen:

1) **Grid area**: 4; **Step size**: 2
2) **Grid area**: 15; **Step size**: 5
3) **Grid area**: 18; **Step size**: 3

Most evaluations are performed upon $n_{bins} = 16$, since those values are the ones that distribute the whole intensity range [0,255] in 16 bins of range size equal to 16. Also, this value was observed to be standard throughout different papers such as the ones in [1] and [2]. After being evaluated, it was opted, for some video sequences, different values of bin sizes, as well as **Grid area** and step size. This tuning varies on specific video sequences, aiming to obtain higher performances.

IV. IMPLEMENTATION

The specifics of the implementation are presented in this section.

A. Color-based tracking: code implementation

The main parameters to tune:

- **hist_bins** - the number of bins in histogram - coarse/fine quantization for the histogram may influence the performance;
- **size_nghbd** and **step** - two parameters which set the number of candidates;
- **selected_channel** - set the color features which will be used for tracking the target.

The first step of the algorithm is selecting the color features; the *select_channel* function was developed for this purpose. The supported conversions are H, S, R, G, B, or gray level (by default).

Later the *if condition* is implemented in order to choose whether the frame is first - then to the algorithms goes to the step of choosing the ground truth to get (x,y) coordinates of the model and initializing it. It also computes the histogram with the implemented function - *calculate_RegionHist* which builds the color histogram of the model using the selected number of bins (using the built-in function *calcHist* and also normalize it (buil-in function *normalize*). Otherwise, if it not the first frame, the previous result center is taken as the current state of the target.

The function *get_candidates* is implemented for applying later for computing the histograms (by calling the function *calculate_RegionHist*) and locations of possible candidates according to the parameters described above - **hist_bins**, **size_nghbd** and **step**. It returns two important vectors which will be proceeded in the next step of the algorithm - $std :: vector < Mat >$ **candidates** and $std :: vector < Rect >$ **candidate_rectangle** for storing a histogram of the candidates and candidate rectangles corresponding to the position of candidates for plotting the results.

The distance to evaluate the candidates is implemented in the *Battacharyya_distance* function which returns the index of the best candidate - the one with the minimum Battacharyya distance (**int index_best_distance**).

To plot the results - the histogram of the best candidate, the function *plot_histogram* was developed which not only draw the histograms of the model and candidate, but also normalize them before that.

B. Gradient-based tracking: code implementation

The logic of the gradient-based tracking method is exactly the same as in the color-based version, thus all the functions and variables were used in this implementation except the *calculate_RegionHist* function.

The gradient-based algorithm is using the *gradient_Histogram* function which uses the built-in implementation of HOG (Histogram of Oriented Gradients) descriptor and object detector - *HOGDescriptor hog*, and the *compute* function which computes HOG descriptors of a given image.

C. Color&Gradient-based tracking: code implementation

The name of the function developed before for building the color of histogram and gradient one were changed, although

the logic is the same. It was made for better code understanding. For instance, the *get_ColorHistogram* for calculating the color-based histogram, and *get_GradientHistogram* for the gradient one, respectively.

All the other functions are the same as in the previous versions. The main idea that this implementation computes both color-based and gradient-based histograms, then calculates the Battacharyya distance. Afterwards, the developed *bestCombinedCandidate* functions combine the two histograms with a simple sum operation.

V. FIRST DATA ANALYSIS: TRACKING PROBLEMS

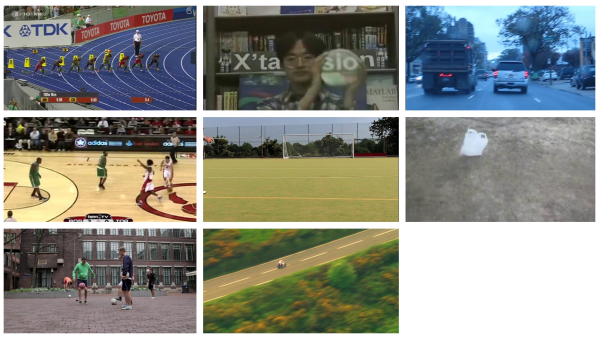

Fig. 4. Image examples from video sequences

In total, there are 8 videos provided for this laboratory session - 1 is for testing the proper work of the algorithms (*bolt 1*), while the rest is used for analysis over real data. Figure 4 represents an image frame from the dataset.

The test sequence **bolt1** which was used during the process of the implementation of the methods is representing the running competition of several sportsmen. The object to track is only one person among these people. This video sequence is challenging because of a set of reasons:

- the camera is not static, and it changes its position every frame by following the sportsmen;
- background is not homogeneous, therefore, quite complex (different structures, shape and colors) which together with camera motion makes tracking more complicated;
- the appearance of a target and other sportsmen - some of them look the same from this scale (for example, same clothes, skin color), as well as the similarity of the background color with respect to the target.

Thus, all these factors are a good start point to evaluate the performance and correct work of the developed histogram-based object tracking algorithms.

The **sphere** sequence contains the frames of a moving ball or sphere in the hands of the person. The problems are the quality video, for instance, comparing to the bolt sequence, this one is lack of illumines, and contrast. However, the main challenge is that the object surface "reflects" or correlates the background which is also has many small shapes and objects, which makes tracking even for people is a complex problem.

The **car** video captures a typical road situation when a single car tracking is required. This sequence represents a complex real one scenario because of a moving and shaking camera as well as detailed background full of objects which may be similar in appearance to the target. Also, cars moving along to the object of interest may affect the performance of a tracking algorithm due to their similarity to the target as well.

The **basketball** demonstrates a part of a basketball game where a object is interest is one of the players. This scene has such problems as moving cameras, complex background, but also similarities between players (for instance, the color of clothes and skin color), and occlusions between the players which may lead to false tracking results.

The **ball2** shows a part of soccer game when the ball is reaching the gates. The main challenges for tracking the scale of the ball is small regarding the whole scene as well as the background area around the location of the ball is quite complex (for instance, it has many similar in shape and color small objects).

The **bag** sequence displays frames of a flying bag on the floor with a movable camera. The white bag is different from the background; however, rapid movement throughout the frames is observed. The camera is shaking as well, which adds extra complexity to the scene.

The **ball** test sequence is frames of the person playing a ball. One of the things which should be taken into account is that the color of the ball is similar to the wall (background), thus it may affect the performance the color-based histogram tracking. At the same time, the possible problem in the video is that it may be quite inconsistent and fast, as well as there is occlusion with the leg in some frames.

The **road** captures the movement of a motorcycle on the road from a certain height. First, this scene may be considered challenging as the motion of camera is changing the position and rotation. Moreover, the scale of the target is small which may cause similarities with other objects in the scene, but also there are occlusions.

VI. RESULTS AND ANALYSIS

In order to proceed with the results and analysis the data had to be tested. The test was performed upon *bolt1*. For this we could observe a general behavior of each one of the approaches.

*1) **Bolt1**:* In the following Figure 5, we can observe what kind of information is provided by each of the channels, which is further used to deepen the analysis.

Here we can observe that some channels provide different information. At first glance, R and G channels convey similar information as well as S and B channels. The information they convey is not discriminant enough, since the runner seems to be misplaced and misclassified for another racer. This happens because low level features such as color are too similar among objects in the scene, therefore it is not about the information on each channel, but about the information on the overall scene. We can observe in the following image how two different candidates have very

Fig. 5. All channels information for the target object in *BOLT1* sequence. From top left to Bottom right, H, S, R, G, B channels respectively

similar color information, yet they are two different objects in the scene. This is depicted in Figure 6 It can be observed

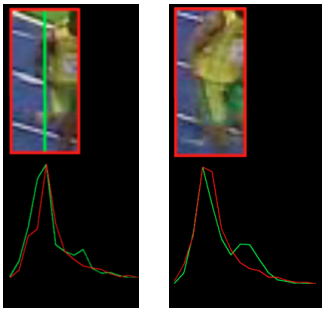

Fig. 6. Tracking performed for Bolt Sequence with a Grid Area of 18 and Stride of 3

that these images have a very similar histogram as observed on the lower part of the images, in fact they seem to match almost perfectly. Therefore, an optimal candidate is chosen according to the model. This happens when occlusions on the image occur, since the grid search area computes candidates around the area and maybe due to the position of the runner on a specific frame it conveys more similar information that the initial target object.

A similar analysis is explained for the next sequences, yet qualitative data is also provided.

A. *Color-based tracking: analysis over real data*

Each of the evaluated sequences will be shown in the following subsections.

*1) **SPHERE**:* For this sequence we can observe the target's channel information in the following Figure 7

We can observe that the H and S color channels does not provide as much relevant information. Therefore, it

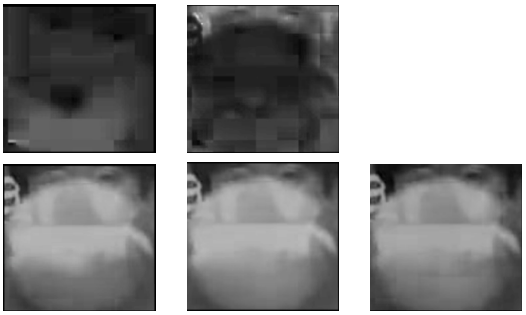

Fig. 7. All channels information for the target object in *SPHERE* sequence. From top left to Bottom right, H, S, R, G, B channels respectively

is expected that low values of accuracy are achieved for this sequence. For the R, G and B channels the ball is quite different from the background, which helps in the identification of the object in context. Also, the ball moves at an almost constant speed ratio across the 2D image plane, which is an ideal case for tracking.

As mentioned earlier, three configurations of grid-space were used, for which the specifics of the running time and tracking performance are shown on Table I and Table II.

TABLE I

COLOR-BASED TRACKING: FOR FIRST SEQUENCE: *SPHERE*. PERFORMANCE ACCURACY MEAN. μ

Config		Color Channels				
N	**Str**	**H**	**S**	**R**	**G**	**B**
4	2	0.13082	0.13158	0.33444	0.24284	0.19293
15	5	0.16380	0.20073	0.45962	0.50204	0.51025
18	3	0.15322	0.20323	0.46683	**0.53861**	0.53351

TABLE II

COLOR-BASED TRACKING: FOR FIRST SEQUENCE: *SPHERE*. AVERAGE PROCESSING TIME IN ***ms/frame***.

Config		Color Channels				
N	**Str**	**H**	**S**	**R**	**G**	**B**
4	2	2.13869	1.88403	1.65863	1.15453	1.31398
15	5	2.17957	2.14082	2.00954	1.73491	1.80536
18	3	5.36238	5.74359	4.56655	5.11811	5.64729

It is important to highlight the computational expense of increasing the number of candidates within the grid search area. The more candidates that are computed the more computational expensive it is. It is not necessarily linearly proportional, however, if too many candidates are chosen for a high resolution image, a long computation time is needed to compute each frame. For our case scenario we are not working with high resolution images, and also we are computing small parts of the image instead of computing the whole frame for a given image, which definitely reduces the computational cost.

As expected, the performance of tracking for H and S channels are much lower than R, G or B channels, regardless of the candidate configuration. This can be observed in Figure 8. H and S channels are not discriminative enough,

therefore a lot of candidates within the search area are chosen as the best match. This can also be observed on the PDFs matching on the lower parts of the Figure.

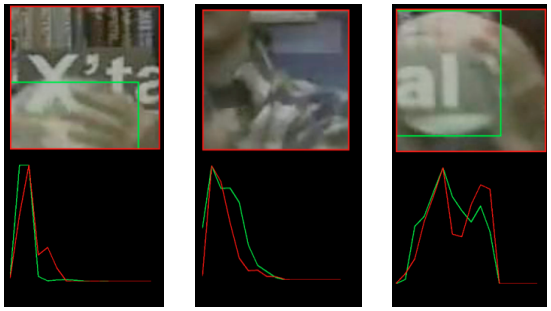

Fig. 8. HS vs RGB color space. Tracking performed for *SPHERE* Sequence with a Grid Area of 15 and Stride of 5. From left to right, H, S and R channel.

Increasing the grid search area improves drastically the result, by almost doubling the performance compared to a small search area. However, having more candidates within a bigger search area does not necessarily mean a better performance. In fact, candidates far from the last tracked target center may have similar color information, yet not be the object to track.

This in fact can be observed on 9 where increasing the search area leads to choosing a wrong object to be tracked.

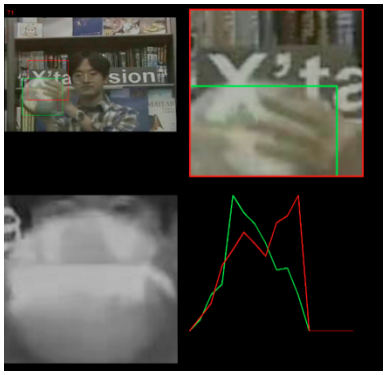

Fig. 9. Opening up the Grid Search, tracking performed for *SPHERE* Sequence with a Grid Area of 18 and Stride of 3 for the R channel.

*2) **White car**:* The target's channel information is shown in the following Figure 10

As observed the H channel does not provide relevant information for tracking the target. The car could easily be misinterpreted as any area with high values of intensity for the H channel. This channel only provides useful information about the back lights, therefore any other car with backlights will be spotted as a potential optimal candidate on this channel. The S, R, G and B channels; however, seem to provide more information. What is expected is that low performance occurs for H channel, whereas significantly better for the rest. The results are shown in the following Table III

We can also see the Average Processing time of each arrangement in the following Table IV

Fig. 10. All channels information for the target object in *SPHERE* sequence. From top left to Bottom right, H, S, R, G, B channels respectively

TABLE III

COLOR-BASED TRACKING: FOR SECOND SEQUENCE: *CAR1*. PERFORMANCE ACCURACY MEAN. μ

Config		Color Channels				
N	**Str**	**H**	**S**	**R**	**G**	**B**
4	2	0.02842	0.18636	0.03057	0.19806	0.03057
15	5	0.02030	0.27467	0.03811	**0.37006**	0.24771
18	3	0.04103	**0.40967**	0.02617	0.02030	0.14282

TABLE IV

COLOR-BASED TRACKING: FOR SECOND SEQUENCE: *CAR*. AVERAGE PROCESSING TIME IN ***ms/frame***.

Config		Color Channels				
N	**Str**	**H**	**S**	**R**	**G**	**B**
4	2	5.65941	2.0654	1.35172	1.27264	1.35172
15	5	3.2144	3.06362	1.47403	1.94682	1.85687
18	3	5.46073	5.3699	3.53913	3.2144	3.90922

Similarly to the first sequence, it can be seen that the more candidates that are generated the higher the computational cost of the operations. In fact for H and S are more computational expensive since a further conversion from a different color space is applied.

As expected very low performance was achieved on the H channel, which does not provide enough information of the object to track itself. Color channels on the RGB space, however, are not as discriminative as expected. This is due to the nature of the color information of the objects in the scene. For instance, it can be observed how the tracking is shifted towards the truck on the left, which contains very similar color information as observed in the respective PDFs matching on the lower part of the image. This can be seen in Figure 11

The S channel, however, seems to perform good even under small number of candidates (compared to the other channels). This can be observed in 12

It is important to mention, that it was expected that the Green Channel would result in a higher performance. However it can be seen that if the grid search area opened up too much then other candidates belonging to the Truck were detected as the optimal state instead. This can be seen in Figure 13. Then again, having more candidates does not necessarily mean better performance, instead it can lower the performance drastically as we can see on Table III

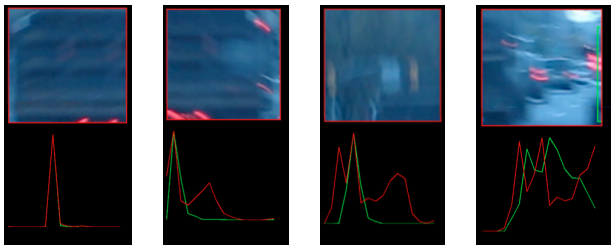

Fig. 11. HSV and RGB color space. Tracking performed for *CAR1* Sequence with a Grid Area of 15 and Stride of 5. From left to right H, R, G and B channel.

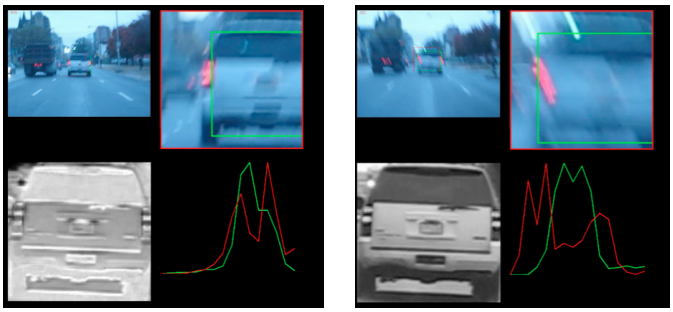

Fig. 12. Tracking performed for *CAR1* Sequence with a Grid Area of 18 and Stride of 3 for S channel, and of 15 and 5 for G channel respectively. From left to right S and G channel.

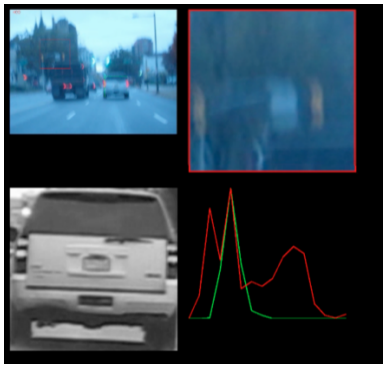

Fig. 13. Opening up the Grid Search, tracking performed for *CAR1* Sequence with a Grid Area of 18 and Stride of 3 for the G channel.

B. *Gradient-based tracking: analysis over real data*

Each of the evaluated sequences are shown in the following subsections.

1) ***BALL2****:* For this sequence we can observe the target's channel information in the following Figure 14

We can observe how information regarding the color channels is very similar for RGB channels, it seems to depict pretty good information that could be discriminant enough from the background. The H channel, however, shows very little important gradient information, which poses a problem for further tracking. The S channel seems to provide more discriminant information than the H channel. We can observe on the next table V showing the quantitative data.

For most of the arrangements the performance is very low. Moreover, qualitative information is also poor. We can see how in most evaluations the object is not being tracked. In

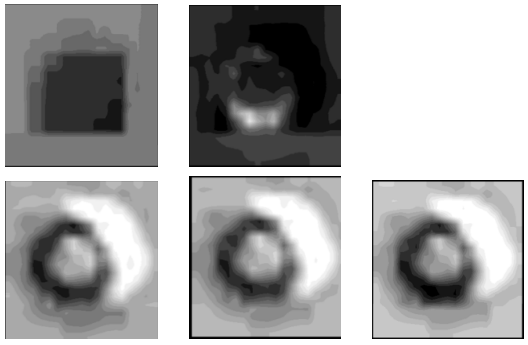

Fig. 14. All channels information for the target object in *BALL2* sequence. From top left to Bottom right, H, S, R, G, B channels respectively

TABLE V

GRADIENT-BASED TRACKING: FOR FIRST SEQUENCE: *BALL2*. PERFORMANCE ACCURACY MEAN. μ

Conf		**Color Channels**					
N	**S**	**H**	**S**	**R**	**G**	**B**	**Gray**
4	2	0.0288	0.0359	0.0402	0.0362	0.0365	——
15	5	0.0262	0.0916	0.0627	0.0654	0.0654	——
18	3	0.0261	0.1506	0.1549	0.1092	0.1699	——
30	3	——	**0.1819**	0.0244	0.0244	**0.0244**	0.0244

fact, it is only tracked during the beginning of the frames for the cases that have the best accuracy. However, as soon as the ball reaches the net all tracker configurations lose sight of the object in context. For configuration of **neighborhood_size=4** and **stride=2**, for which a maximum of 25 candidates are generated. We can see how performance is very low and the tracker never really has sight of the object more than for a period of 5 frames at the beginning of the sequence. Once the configuration changes to a **neighborhood_size=18** and **stride=3**, for which a maximum of 169 candidates are generated, the performance improved drastically, although still a low performance is obtained. We can observe the behavior of the tracker on the following Figure 15, for which the net is detected as an object and the ball itself, however the two of them show a similar gradient information distribution.

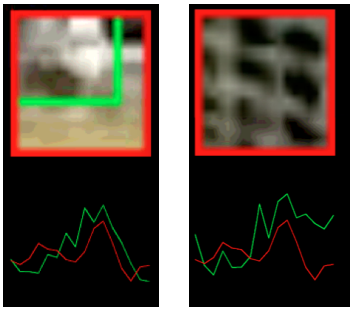

Fig. 15. Tracking performed for *BALL2* Sequence with a Grid Area of 30 and Stride of 3, and of 18 and 3 respectively for S channel. From left to right correct yet not optimal and incorrect target detection.

We can observe in Figure 15 that the net and ball have similar gradient information, reason why the target is not tracked correctly.

After applying all the already fixed parametrization values,

it was decided to add one more, with a grid search area of 30 and Stride of 3. This is because the ball changes its location in drastic amounts throughout subsequent frames, which implies having a greater area of search. With this arrangement the highest performance was achieved of **0.01819**. Having tried with several candidate numbers we observe that the problem is in the amount of discriminant information compared to the background. We tried different combinations, even by changing the **n_bins=9,12,18**. What happens is that the ball changes to fast and it goes out of the search area, then the net is located as an object of interest. However, if we choose a bigger search area to track the ball the candidates also go to parts of the net which again offer similar gradient information to the ball. If the search area is too small then the ball goes out of the search area. The number of bins increment should help, however there is a trade-off between size of beans and accuracy of tracking. This is because our probability density function is too fine, not leaving enough flexibility to changes in the target object.

2) ***BASKETBALL****:* For this sequence we can observe the target's channel information in the following Figure 16

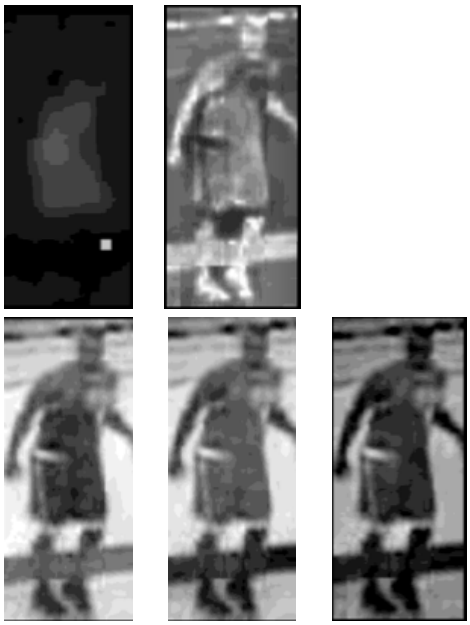

Fig. 16. All channels information for the target object in *BASKETBALL* sequence. From top left to Bottom right, H, S, R, G, B channels respectively

Here we observe that the least discriminative channel is definitely the H channel, which is very unlikely to result in good performance of the tracker. For the other 4 channels it seems to have good information; however, the rest of the scene has a lot of of similar information therefore they are still not optimally discriminative. On the next Table VI we can observe the results on the specified arrangements.

H channel performs impressively well instead of what was expected. Qualitatively speaking, it loses track somewhere in the middle until the candidates find the player back again. The detection loses the target due to the high speed change which may go outside of the grid search area. However, this is fixed once grid opens up. But again, opening up

TABLE VI

GRADIENT-BASED TRACKING: FOR SECOND SEQUENCE: *Basketball*

Conf.		Color Channels				
N	**Str**	**H**	**S**	**R**	**G**	**B**
4	2	0.11490	0.09516	0.06199	0.48973	0.09398
15	5	0.44085	0.46027	0.61451	0.60244	0.5897
18	3	0.36947	0.61433	0.08757	0.55872	0.55396

the grid also provokes a downfall in some scenarios such as the H and R for white the tracker chooses a different target due to their similitude in appearance. The R channel has some downfalls as the tracker follows the wrong player due the high values of intensity in the red channel on the shorts of other players. Moreover, the tracker loses the player and tracks the background which has red paint on the floor. As mentioned earlier opening up the grid search area generates a downfall in the tracking. This can be observed in Figure 17. The G channel has better performance than Red channel, since target person's shirt is green, therefore more discriminant on the same channel. Visually, there are some wrong tracking results in the beginning of the video as well as in the end. When opening the grid search area too much, the performance is worse compared to the **Neighborhood_size=15** and **Stride=5**. The shorts may be the problem, since they have similar gradient information on the waist and ends of the shorts, therefore if most of the image has another player with shorts equally discriminant from their skin tone the tracker is not performing optimally. Figure 17 depicts two correct tracking and one incorrect on the H,G and R channels respectively.

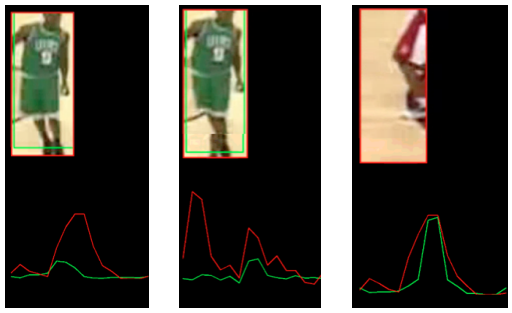

Fig. 17. Two correct and one incorrect classifications for the *BASKETBALL* sequence. From left to right matching of the candidate to the target for H and G channels, and wrong tracking for the R channel

It is important to notice how the incorrect target actually matches the gradient information almost perfectly when observing the two PDFs, this is as mentioned earlier depicting the problem of the R channel when performing the tracking. It is important to highlight that this scene faces several problems such as occlusion of the player with another player of a very similar color information. Therefore, the tracker confuses the target when two players of the same team are too close to each other. When the area around the player is clear the tracker performs optimally.

C. *Color&Gradient-based tracking: analysis over real data*

Each of the tested and analyzed video sequences will be described in details in the following sections.

*1) **BAG**:* The target's channel information is shown in the following Figure 18

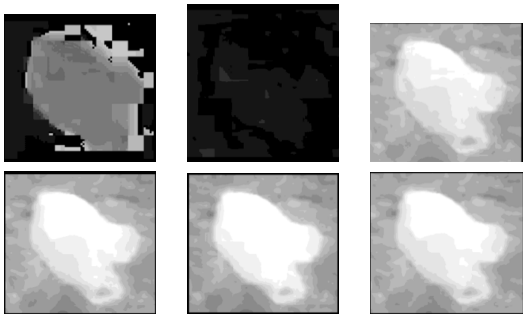

Fig. 18. All channels information for the target object in *BAG* sequence. From top left to Bottom right, H, S, R, G, B, Gray channels respectively

From this we can observe the most discriminative information is given on the H channel, for the rest of channels the information is pretty poor. Although for R,G,B and Gray channels the complete image of the bag is depicted, it does not necessarily mean that the bag is tracked accordingly, since information on color change through the bag itself and part of the background is included. For the H case scenario; however, optimal background differentiation is achieved.

In the first set of experiments, when the **grid neighborhood was 4** and **step equals to 2**, we can see quite low performance among all features. The main problem observed is that the object is moving faster than what the search area can work on. Thus, in all cases the tracker could not track the target in motion, but did it for a static one - therefore the candidates were not centered on the bag, but on parts of it.

Later, when the number of **grid neighborhood** and **step** was raised to **15** and **5** respectively, there is a significant increase in the accuracy. At this step, the algorithm can track the target both in dynamic and static, but sometimes loses the bag due to a complex camera motion. It happens equally for all features except S channel, which showed the worst performance among others. Although the H features seemed to be the best in this case, the difference in accuracy between H and B is not big.

The **grid of 18** and **step of 3** did not improve the performance dramatically, but in most cases even could reduce it (expect for S channel). However, since the tracking accuracy for S is low, we can conclude that both the increased number of grid neighborhood and step did not work and that the S channel is not discriminative for this sequence.

Also, we carried out the set of experiments regarding the number of bins in the histogram. We proceeded to decrease its number since the less bins, the less specific (coarser) PDF. Therefore there should be more increase in performance, which is known as the PDF-accuracy trade off. We could obtain an average tracking performance of **0.322** for the R channel when decreasing the value of number of bins, which is higher than **0.226** - for configuration of 18 (grid neighborhood) and 3 (step). An increasement of the histogram number to 18,20 and 24 were also performed. However, this did not change the results for the rest of channels. As the H channel was the best in terms of performance, we reduced the number of bins, but the accuracy did not improve. We can observe some results after the number of bins had been reduced in the following Figure 19

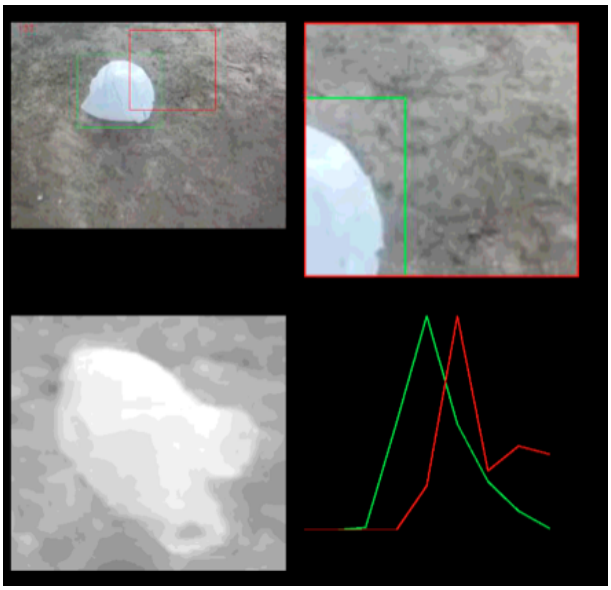

Fig. 19. **BAG** sequence with **number_of_bins=9** depicting its respective histogram and the overlap between them.

The detailed quantitative data representing the results of testing the **bag** sequence is shown in VII. The best performance was observed with parameters of 15 for the grid neighborhood and 5 for the step, and it was about **0.372146**.

TABLE VII

COLOR&GRADIENT-BASED TRACKING: FOR FIRST SEQUENCE: *Bag*

Config		Color Channels					
N	**Str**	**H**	**S**	**R**	**G**	**B**	**Gray**
4	2	0.055	0.022	0.065	0.121	0.127	0.144
15	5	**0.372**	0.072	0.345	0.330	0.360	0.303
18	3	**0.369**	0.077	0.226	0.317	0.316	0.327

As expected the highest performance is achieved on the H channel, for the discriminative power of the channel mentioned above. We can conclude that in general the overall performance for this video sequence is quite low (less than 0.5) that even the fusion of histograms did not succeed. The main reason may be that the scene is quite complex as it was discussed in the analysis of data, thus, the more complex approaches are required.

*2) **BALL**:* The target's channel information is shown in Figure 20

For initial settings which were the **grid neighborhood of 4** and **step of 2**, we can observe overall low performance. At this step we could not conclude which of the features are best for this video. The main problem which happened in

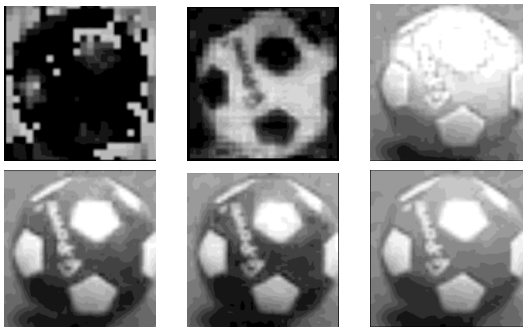

Fig. 20. All channels information for the target object in *BALL* sequence. From top left to Bottom right, H, S, R, G, B, Gray channels respectively

all channels was that the tracker lost the target at the same moment - when the ball starts its motion. This occurs due to the big displacement of the ball within subsequent frames, which translates in going out of the grid search area and therefore losing track of the object. Three incorrect track results can be observed in Figure 21

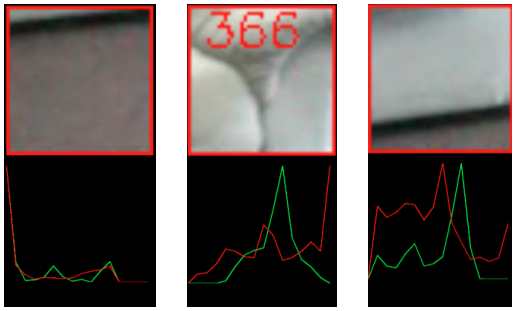

Fig. 21. Tracking performed for *BALL* Sequence with a Grid Area of 15 and Stride of 5. From left H, R and G channels respectively.

The next set of experiments with parameters of **15 (neighborhood)** and **5 (step)** increased the performance significantly - it was expected since the grid search area also increased. Comparing H and S, we can see that the better performance was for the H channel. Regarding the analysis of R, G and B separately, it is worth to notice that in the R channel the tracker loses the target when the ball reaching the couch. Since the pixel values for the red channel are high, they look similar to the white values in the red channel. Thus, it is discriminative when the ball is far from the couch, but not in the surroundings of it. Regarding the G channel, this is far more discriminative than R due to the pixel values in the RGB channels, while B showed the best accuracy in the RGB color space. Also, it is important to notice how the gray channel works - unlike the others, it rarely loses the ball.

Having increased the **grid neighborhood to 18** and **step to 3**, we could increase the performance for R, G, and gray significantly. Regarding the R features, the tracker still loses the object, but comes to the right state. The R color performs better than expected, and the search area definitely improves the performance. Three correct tracking results can be observed in Figure 22.

In case of G, the algorithm loses more track of the object

Fig. 22. Tracking performed for *BALL* Sequence with a Grid Area of 18 and Stride of 3. From left H, R and Gray channels respectively.

than expected, since blue is more discriminant to the white values in the red channel. However, low values of the green channel of the ball also interact with the low values of the chair and desk, which leads to the algorithm fails.

The detailed quantitative data representing the results of testing the **ball** sequence is shown in Table VIII.

TABLE VIII

COLOR&GRADIENT-BASED TRACKING: FOR FIRST SEQUENCE: *Ball*

Configurations		Color Channels					
N	**Str**	**H**	**S**	**R**	**G**	**B**	**Gray**
4	2	0.155	0.158	0.181	0.173	0.158	0.172
15	5	0.512	0.435	0.319	0.408	0.547	0.503
18	3	0.519	0.446	0.513	0.505	0.482	0.590

To conclude, the gray level and parameters of 18 and 3 showed the best accuracy for this video (**0.590247**). Its features seems to be more discriminant and the candidates are closer to each other since the stride decreases. At the same time, both the S and B channels showed the lowest accuracy, this happens as the channels do not present distinct information for tracking.

*3) **ROAD**:* The target's channel information is shown in the following Figure 23

Fig. 23. All channels information for the target object in *CAR1* sequence. From top left to Bottom right, H, S, R, G, and B channels respectively

In the first tests with the **neighborhood of 4** and **step of 2**, the overall performance of all features are low. This could happen either because the size may be too small (and the motion is too fast to be caught by the search area) or a large number of occlusions with trees since the tracker loses the target after these moments. Among others, the B channel seems to show the better accuracy, but not the optimal one.

In case of setting the **parameters to 15 and 5**, there are various reasons why the algorithm failed to track the target perfectly. The H channel has bad discriminative power, and, in fact it performs worse than the 4, 2 case due to the opening of the grid search area. The same low discriminative power is observed in the S features which failed for this video. In case of R, candidate generation is poor, and none of the distributions matches the target model. The green channel information is high, but the candidates tend to the parts of the background with a similar distribution of green color. Also, the gradient information is not as discriminant as expected, so as long as the green parts have similar gradients than those of the motorcycle they will be spotted. The B channel is robust to occlusions, therefore the object is tracked properly. The most important information to notice is that the values of the background are rich in the red and green channels, therefore the blue channel is the most optimal.

TABLE IX
COLOR&GRADIENT-BASED TRACKING: FOR FIRST SEQUENCE: *Road*

Configurations		**Color Channels**					
N	**Str**	**H**	**S**	**R**	**G**	**B**	**Gray**
4	2	0.101	0.038	0.075	0.077	0.227	0.076
15	5	0.053	0.036	0.071	0.071	0.431	0.072
18	3	0.035	0.116	0.266	0.290	0.435	0.275

When setting the **grid neighborhood** to 18 and **step** to 3, it supported the idea concluded before - the H channel has low discriminative power and does not work for this video. Also, the B channel proved the best performance (**0.435817**) due to informative features. It loses the track only when the green color is bright due to the higher value of the blue channel within this area. In general, the performance could increase due to the close location of the candidates and more options to choose from.

The detailed quantitative data representing the results of testing the **road** sequence is shown in Table IX.

VII. CONCLUSIONS

Having performed various combinations of grid areas, stride size and candidate numbers on different channels of different color spaces, as well as several coarseness of our Probability Density Functions it can be concluded that there is not one specific method or relationship to obtaining an improvement in the tracking performance. The three most common faced problems on video sequences are the similarity of the background from the object of interest, the presence of occlusions and the displacement of the object within two subsequent frames. Addressing the first problem, similarity between background and foreground, it is important to select a color space that provides as much possible discriminant information to the background included in the same area of interest within the *Blob*. This however, does not necessarily guarantee a good tracking since other parts of the frame may have exactly the same information as the one of our object of interest, which result in low tracking performance. Approximating a very specific Density Function may seem like a good idea; however, the trade off between a Fine PDF and performance results in a problem as well.

The presence of occlusions addressed by increasing the search area and stride as long as the occlusion is not too long. This is because as soon as the object is visible again, one of the candidates may be able to compute the optimal state and every subsequent state can be improved. The same solution is posed for large displacements of the object within subsequent frames.

Although increasing the grid search area may be beneficial for occlusions and rapid change of the object in time, it may pose another problem, wrong computation of an optimal candidate. This may occur since an object similar to the target is around the search area. If the optimal candidate is selected opposite the direction of the displacement of the object, and the target object leaves the grid search area, then it may be impossible to recover the tracking, until the object of interest returns to the search area.

It is important to mention that this is a very basic tracker, since it uses low level features, therefore very low performance is expected for sequences with occlusions, similarities among background and foreground and rapid movement of the object of interest.

VIII. TIME LOG

A. *Color-based tracking: code implementation*

The time spent on this part of the laboratory session is 6 hours in total - around 2 hours required to read the articles and learn the concepts out of it, 4 hours to implement the algorithm and to find bugs and.

B. *Gradient-based tracking: code implementation*

The time spent on this part is around 2 hours - as it required to implement only one function, the other parts of the code were reused.

C. *Color&Gradient-based tracking: code implementation*

The implementation of this version of the algorithms took around 3 hours - most parts of the code were reused, and only one new function was developed.

D. *Testing*

It took around 10 hours to test all video sequences and save the results.

E. *Report*

Around 6 hours: writing the draft, proof-reading and editing.

REFERENCES

[1] Fieguth, Paul, and Demetri Terzopoulos. "Color-based tracking of heads and other mobile objects at video frame rates." Proceedings of IEEE computer society conference on computer vision and pattern recognition. IEEE, 1997.

[2] Birchfield, Stan. "Elliptical head tracking using intensity gradients and color histograms." Proceedings. 1998 IEEE Computer Society conference on computer vision and pattern recognition (Cat. No. 98CB36231). IEEE, 1998.