Histogram based object tracking

María Fernanda Herrera and David Savary

I. INTRODUCTION

The goal of this programming task is to implement and evaluate multiple histogram based trackers. In total, three trackers will be implemented: a color based histogram tracker, a gradient based histogram tracker and a last one implemented by combining the two firstly mentioned. The trackers were implemented following the ideas described in [1], which makes use of a candidate evaluation system explained in [2]. The performance and accuracy of the different developed trackers were tested over different sequences obtained from the Vot challenge database [3] according to Intersection Over Union criterion and processing time. .

II. RUNNING THE CODE

To run the implemented code, it is necessary to compile it in a Linux machine with OpenCV installed by using the Makefile. The *make* command will generate a "*main*" executable that can be run by passing as argument the path of the desired videos separated by a space, for example: "*./main path/to/videosequence1/ path/to/videosequence2/ ...*". The rest of the different parameters associated to the tracking configuration can be modified by accessing at the main.cpp file.

III. COLOR BASED TRACKING

A. METHOD AND IMPLEMENTATION

The algorithm is implemented into five different files. The main file *main.cpp* reads video sequences, initializes and applies the color histogram tracking and finally displays the achieved results. The second and third file, *ColorTracker.hpp* and *ColorTracker.cpp*, contains the attributes and methods defined for the *ColorTracker* class used in the *main.cpp* , which performs color histogram tracking by using a single color channel (red, green, blue, hue, saturation or gray scale channel). Current implementation also allows to use multiple color channels, however, only the analysis of individual channels is considered. The process of color histogram tracking, defined in the *ColorTracker* class consists in four different different stages:

- Model initialization: Defines a color histogram of the target at the beginning of a video sequence.
- Candidate generation: Defines a search region containing many possible locations where the target may appear (candidates) and obtains a color histogram for each location.
- Candidate evaluation: Obtains a measure of the difference between target and candidate histograms.
- Candidate selection: Returns the candidate that best represents the target according to the color histogram comparison.

The two last files are *utils.hpp* and *utils.cpp*, which allow to read ground truth data and evaluate tracking results against the ground truth data.

1) INITIALIZATION: .
ColorTracker class requires to indicate 4 different parameters:

- Bins (*bins*): Defines the number of bins used to calculate the histogram of each candidate.
- Step (*candidate_step*): Defines the vertical and horizontal pixel distance between candidate centers.
- Levels (*candidate_levels*): Defines a grid that contains the bounding box position for each candidate in each frame. The number of candidates is obtained as $Candidates = (2candidate\_levels + 1)^2$, while the size of the search region containing those candidates is obtained as $SearchRegion = ((2candidate\_levels)(candidate\_step) + 1)^2$.
- Histogram type (*track_type*): Defines the color channel to be considered according to boolean flags.

Model initialization is performed once at the beginning of each sequence by reading the first ground truth bounding box and obtaining a normalized color histogram from this region, This process is defined within the *init_model* method. The histograms are calculated according to the *bins* and *track_type* parameters, where the last is used to obtain the required color channel(s) through the *get_color_space* method.

2) CANDIDATE GENERATION AND EVALUATION: Iteration over all considered candidates is performed within the *generate_candidate* method, which returns a bounding box and a distance measure associated with each candidate in each frame. To achieve this, it is necessary to use the *track_type* parameter to obtain the required color channel(s) through the *get_color_space* method. Then, *candidate_levels* and *candidate_step* parameters are used to generate candidate positions and the *get_distance* method is used to obtain the candidate histogram and *Battacharyya distance* between candidate and model histograms for each candidate.

3) CANDIDATE SELECTION: Finally, the candidate bounding box related to the smallest candidate distance is returned as the target location through the *track* method.

B. DATA DESCRIPTION: VOT DATASET

To evaluate color histogram based tracking, three different sequences were considered: bolt1, sphere and car1. Besides, all the frames for a video sequences in VOT Challenge

[3] are annotated for its challenging characteristics. The characteristics for each video sequence are summarized in table I.

C. EXPERIMENTAL METHODOLOGY

In order to analyze the effect of each parameter on the tracker performance, a test grid with different values for the tracker parameters was defined as follows:

- Bins (*bins*): 8, 16, 32 and 64
- Levels (*candidate_levels*): 3 levels (generates 49 candidates), 5 levels (generates 121 candidates) and 7 levels (generates 225 candidates)
- Step (*candidate_step*): 1 and 2, where the second number doubles the size of the search region, while keeping the same number of candidates.
- Histogram type (*track_type*): red, green, blue, hue, saturation and gray scale were used separately.

A total of 144 different parameter settings was used for each video sequence. The obtained results from these combinations are evaluated by comparing the Intersection over Union criterion and processing time for different number of bins and region widths. Accordingly, this value is determined as $SearchRegionWidth = ((2candidate\_levels)(candidate\_step) + 1)$ and the real region size is defined as $SearchRegionSize = (SearchRegionWidth)^2$. The comparison is shown in colored tables.

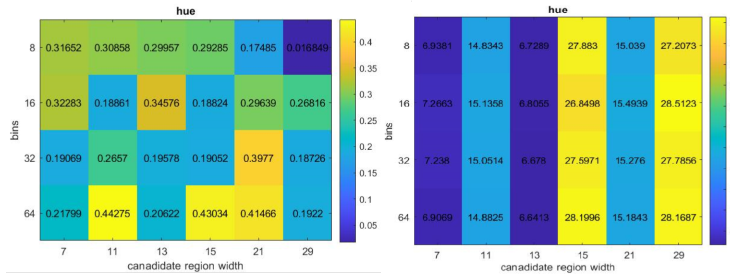

Fig. 1. Results obtained for bolt sequence using hue color channel for different combinations of number of histogram bins and width of search region for candidates. Left figure shows the result for Intersection Over Union criterion with ranges from 0 (worst) to 1(best). Right figure shows the result for time processing in ms/frame.

Fig. 2. Results obtained for bolt sequence using hue color channel, 64 bins histogram, 5 levels and stride of 1.

D. RESULTS AND ANALYSIS

1) Bolt: For this sequence, the best result was obtained from the hue channel with a configuration of 64 bins histogram and a search region of width 11, which is obtained by selecting 5 levels and stride equal to 1. The Intersection Over Union criterion achieves a value of 0.44 with a processing time of 14.8 ms/frame. The results obtained for other combinations of bins and search region size for the same channel are shown in Fig. 1. In this figure, it can be observed that different number of bins does not impact the processing time, on the contrary, this time is dependent on the number of selected candidates. Accordingly, a larger search region defines similar processing times when compared to a smaller region that contains the same number of candidates (i.e. search region of size 7 defined by 3 levels and step 1 has similar processing times to search region of size 13 defined by 3 levels and step 2). By observing at the results from other channels, it was also possible to conclude that combinations defined by more bins and a smaller search region achieved better results for this sequence. One explanation for this effect is that the object is surrounded by objects with similar characteristics. Thus, a large search region may cause the tracker to confuse the target with other objects. Besides, the object of interest is positioned far from the camera, which causes the object to appear small and also with small position variations. As a consequence, smaller search regions allow to better track the object. Finally, a histogram defined by more bins generates a more specific model that may allow to differentiate between similar candidates more easily. Fig. 2 shows a sample result when the configuration is set to 64 bins, 5 levels and stride of 1 and hue channel (search region width equal to 11), while 3 shows the worst result for the hue channel, which considers 8 bins and a search region defined by 7 levels and stride 2 search region width equal to 29).

2) Sphere: For this sequence, the best result was obtained from the green channel with a configuration of 64 bins histogram and a search region of width 29, which is obtained by selecting 7 levels and stride equal to 2. The Intersection Over Union criterion achieves a value of 0.476 with a processing time of 23.22 ms/frame. The results obtained for other combinations on bins and search region size for the same channel are shown in Fig. 4. It is important to highlight that he best combination selection should be based on

Fig. 3. Results obtained for bolt sequence using hue color channel, 8 bins histogram, 7 levels and stride of 2.

Video Sequence	Changes in object motion	Changes in object size	Occlusion	Camera motion	Illumination change
bolt1	*X*	-	-	*X*	-
car1	*X*	-	-	*X*	-
sphere	*X*	*X*	-	*X*	-
basketball	*X*	*X*	*X*	*X*	-
ball2	*X*	*X*	-	-	-
bag	*X*	*X*	-	*X*	*X*
ball	*X*	*X*	-	*X*	-
road	-	-	-	*X*	-

TABLE I

SUMMARY OF CHALLENGING CHARACTERISTICS THAT SHOWN TEST VIDEO SEQUENCES.

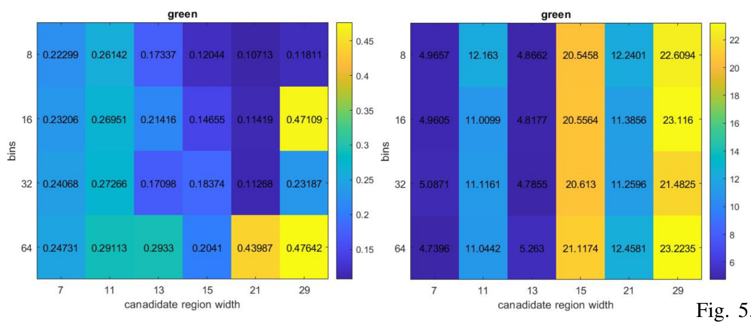

Fig. 4. Results obtained for sphere sequence using green color channel for different combinations of number of histogram bins and width of search region for candidates. Left figure shows the result for Intersection Over Union criterion with ranges from 0 (worst) to 1(best). Right figure shows the result for time processing in ms/frame.

Fig. 5. Results obtained for sphere sequence using green color channel, 64 bins histogram, 7 levels and stride of 2.

Fig. 6. Results obtained for sphere sequence using green color channel, 8 bins histogram, 7 levels and stride of 2.

application requirements. For this case, the best combination is mainly selected according to the best IoU result, however, for applications that may require faster responses, 64 bins and a search region defined with width equal to 21 provides similar results in half the processing time. Results from other channels allowed to observe that combinations defined by a bigger search region achieved better results for this sequence. This may be related to the fact that the object is described by a large bounding box and is close to the camera, which causes the position changes to look larger than when the object is far from the camera. Thus, larger search regions allow to better recover the object. Fig. 5 shows a sample result when the configuration is set to 64 bins, 7 levels, stride of 2 (region with equal to 29) and green channel is selected, while Fig 6 shows a bad result obtained from the same channel, which considers 8 bins and a search region defined by 7 levels and stride 2 (region width equal to 29).

3) Car1: For this sequence, the best result was obtained from the green channel with a configuration of 64 bins histogram and a search region of width 7, which is obtained by selecting 3 levels and stride equal to 1. The Intersection Over Union criterion achieves a value of 0.32 with a processing time of 8.52 ms/frame. The results obtained for other combinations of bins and search region size for the same channel are shown in Fig. 7. The two best results are obtained by defining a small search region. This is explained by the fact that the object of interest has the same trajectory as the camera and thus, it seems to stay in the middle of the image plane without showing large displacements from this position. Thus, a small search region may be better suited for this problem. However, one of the main difficulties for this sequence is that camera motion causes random and abrupt displacements of the object from its central position, which highly decreases the performance of the tracking. Results from other channels generated bad results for this problem, which confirms the complexity of the video sequence. Other channels that achieved an similar performance to the green channel were red and saturation channels with approximately a value of 0.25 for Intersection Over Union criterion. Fig. 8 shows a sample result when the configuration is set to 64 bins, 3 levels, stride of 1 (search region width equal to 7) and green channel is selected while 9 shows a bad result obtained

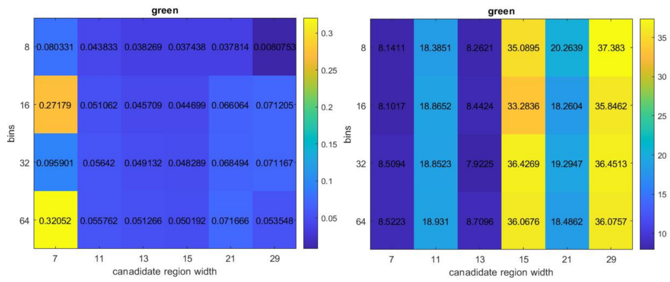

Fig. 7. Results obtained for car1 sequence using green color channel for different combinations of number of histogram bins and width of search region for candidates. Left figure shows the result for Intersection Over Union criterion with ranges from 0 (worst) to 1(best). Right figure shows the result for time processing in ms/frame.

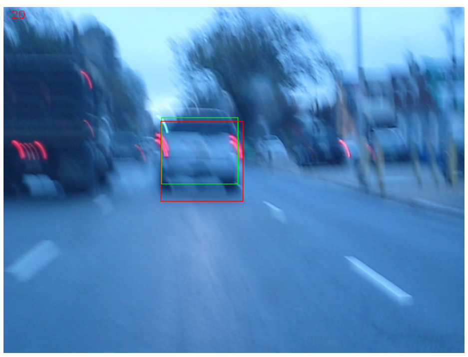

Fig. 8. Results obtained for car1 sequence using green color channel, 64 bins histogram, 3 levels and stride of 1.

from the same channel, which considers 8 bins and a search region defined by 7 levels and stride 2 (search region width equal to 29).

IV. GRADIENT BASED TRACKING

A. METHOD AND IMPLEMENTATION

The structure of the gradient based tracking algorithm is very similar to the one used for the color based tracker algorithm. The code is also separated in five files: *main.cpp*, *utils.hpp*, *utils.cpp*, *GradientTracker.hpp* and *GradientTracker.cpp*. The *main.cpp* file was mostly untouched, with the only differences being the initialization and use of a gradient tracker instead of a color tracker, while *utils.hpp* and *utils.cpp* are completely untouched. The *GradientTracker.hpp* and *GradientTracker.cpp* files implement the *GradientTracker* class that is used to perform the tracking. The process of gradient tracking implemented in the *track* function of the *GradientTracker* class has the same four stages as the *ColorTracker* class , each one having the same functionality:

- Model initialization: Computes an HOG descriptor of the target at the beginning of a video sequence.
- Candidate generation: Computes HOG descriptors for the regions where the object of interest may be located in each frame.
- Candidate evaluation: Obtains a measure of the difference between target and candidate descriptors.
- Candidate selection: Returns the candidate that best represents the target.

1) INITIALIZATION: .
The GradientTracker class requires three parameters:

- Bins (*bins*): Defines the length of the HOG descriptors computer for each candidate.
- Step (*candidate_step*): Defines the vertical and horizontal pixel distance between candidate centers.
- Levels (*candidate_levels*): Defines a grid that contains the bounding box position for each candidate in each frame. The number of candidates is obtained as $Candidates = (2candidate\_levels + 1)^2$, while the size of the search region containing those candidates is obtained as $SearchRegion = ((2candidate\_levels)(candidate\_step) + 1)^2$.

The model initialization is also performed once at the beginning of the video sequence by calculating the HOG descriptor of the first ground truth bounding box. The process is implemented in the *init_model* method. The HOG descriptors are extracted using the *HOGDescriptor* class from the OpenCV library.

2) CANDIDATE GENERATION AND EVALUATION: The candidate iteration process is identical to the ColorTracker approach. In this case, the HOG descriptor of each candidate is compared to the target HOG descriptor in the previous frame by getting the L2 distance using the *get_distance* function.

3) CANDIDATE SELECTION: In the last stage, the candidate with the smallest L2 distance to the target is selected and its bounding box is returned to the calling function.

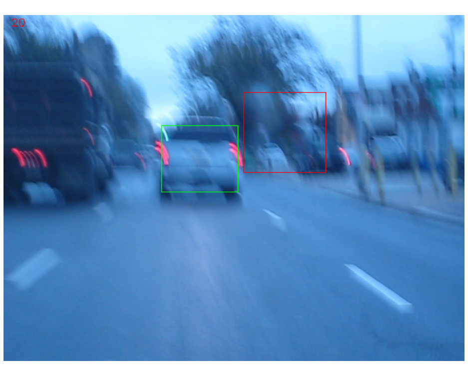

Fig. 9. Results obtained for car1 sequence using green color channel, 8 bins histogram, 7 levels and stride of 2.

B. DATA DESCRIPTION: VOT DATASET

To evaluate color histogram based tracking, three sequences were considered: bolt1, basketball and ball2. The characteristics for each video sequence are summarized in table I.

C. EXPERIMENTAL METHODOLOGY

In order to analyze the effect of each parameter on the tracker performance, a test grid with different values for the tracker parameters was defined as follows:

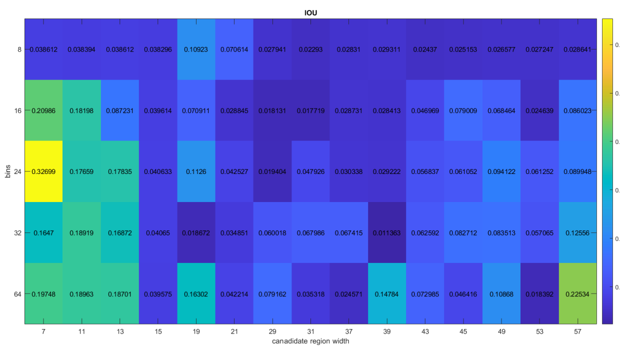

Fig. 10. Intersection Over Union results obtained for bolt sequence using HOG tracker for different combinations of number of histogram bins and width of search region for candidates.

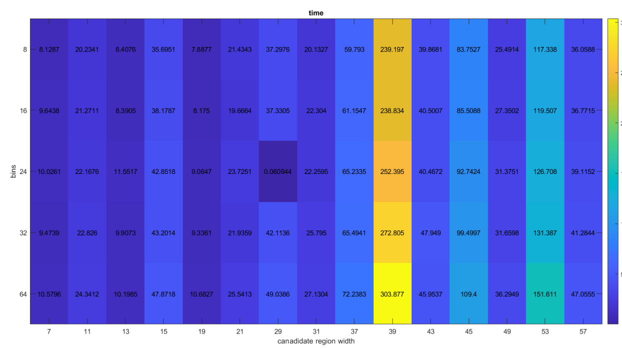

Fig. 11. Processing time obtained for bolt sequence using HOG tracker for different combinations of number of histogram bins and width of search region for candidates.

- Bins (*bins*): 8, 16, 24, 32 and 64
- Levels (*candidate_levels*): 3, 5, 7 and 19 levels with stride 1. 3, 5, 7, 9, 11 and 13 levels with stride 2. 3, 5 and 7 levels with stride 3, And 6 and 7 levels with stride 4,

A total of 75 different parameter settings was used for each video sequence.The obtained results from these combinations are compared according to the same strategy defined in color histogram experimental methodology.

D. RESULTS AND ANALYSIS

1) Bolt: The best result for the bolt sequence was achieved using 24 bins and a search area of 7 candidates wide (obtained from 3 levels and stride of 1). With these parameters, the *GradientTracker* achieves 0.32 Intersection Over Union at 10.02 ms/frame. Fig 10 shows different combinations of parameters tested, showing a clear maximum around small candidate numbers. Analyzing the ms/frame needed with every combination tested in Fig. 11 we can see that there are peaks at specific candidate region widths due to very inefficient parameter configurations, like high number of candidates and low stride. With the best parameter combination, the *GradientTracker* is not able to lock onto the target for the entire length of the sequence, as shown in Fig. 12, but is the combination that follows the target for the longest time. Fig 13 shows the same frame with a sub optimal configuration, defined by 16 bins and search region width equal to 7.

2) basketball: The best result achieved in the basketball sequence was 0.55 Intersection Over Union at 27.69 ms/frame. These scores were achieved using 6 levels of candidates, 16 bins and a stride of 4 (search region width equal to 49). Analyzing the score achieved with every combination in Fig, 14, it can be seen that next to the global maximum score, there is a very bad result achieved by setting the number of bins to 16, the levels of candidates to 13 and the stride to 2 (search region width equal to 53). Apart from being one of the worse scores, this configuration is also computationally expensive as Fig. 15 shows. The results of those two configurations are showed in Figs. 5 and 6

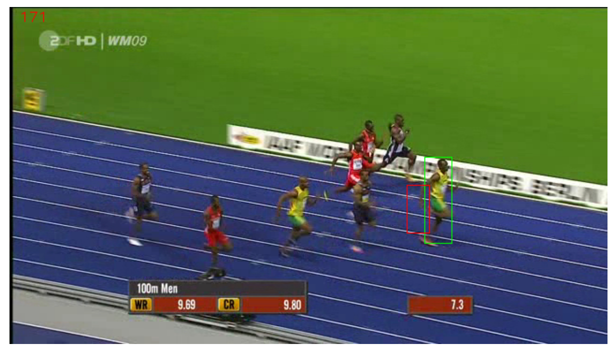

Fig. 12. Results obtained for bolt sequence using 24 bins, 3 levels and stride 1.

respectively. As the figures show, the optimal configuration is able to lock onto the target until the very end of the sequence, unlike the sub optimal configuration.

3) ball2: In this sequence, the best parameter configuration found was 16 bins, 19 candidate levels and stride of 1. These results achieve a Intersection Over Union score of 0.30 at 231 ms/frame. Although many other configurations were tested in terms of score and performance, as shown in Figs. 18 and 19, lowering the number of candidate levels and compensating the size of the search region raising the stride could not match the score of using a large number of levels with the minimum stride. This is due to the small size of the

Fig. 13. Results obtained for bolt sequence using 16 bins, 3 levels and stride 1.

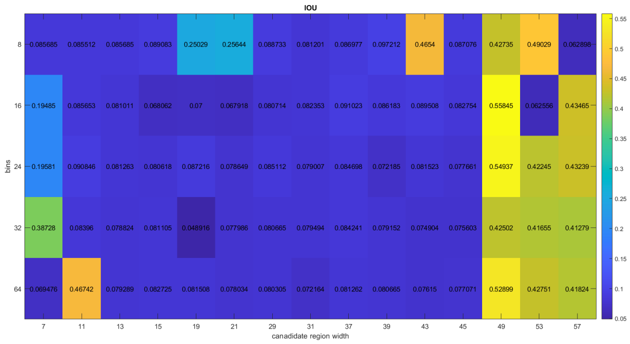

Fig. 14. Intersection Over Union results obtained for basketball sequence using HOG tracker for different combinations of number of histogram bins and width of search region for candidates.

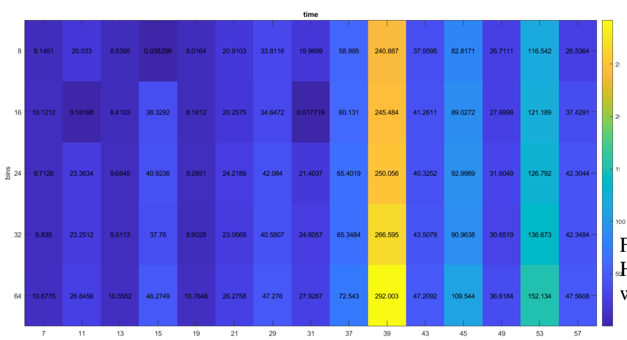

Fig. 15. Processing time obtained for basketball sequence using HOG tracker for different combinations of number of histogram bins and width of search region for candidates.

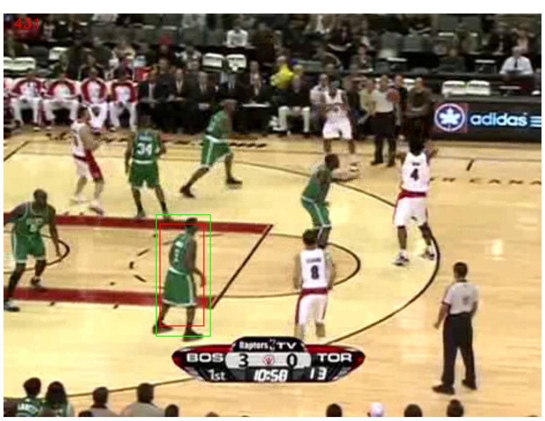

Fig. 16. Results obtained for basketball sequence using 16 bins, 6 levels and stride 4.

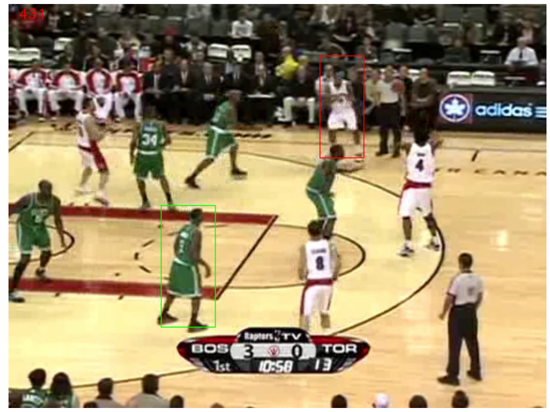

Fig. 17. Results obtained for basketball sequence using 16 bins, 13 levels and stride 2.

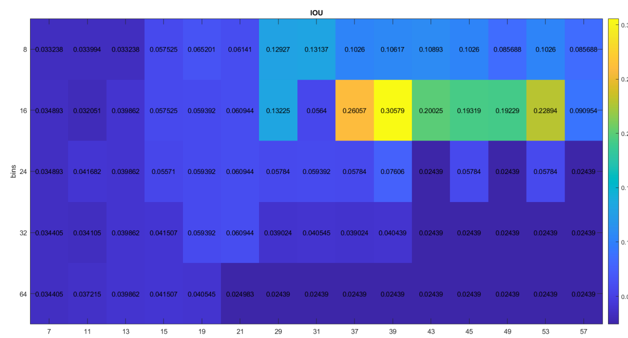

Fig. 18. Intersection Over Union results obtained for ball2 sequence using HOG tracker for different combinations of number of histogram bins and width of search region for candidates.

target, that causes a lot of background information in relation tho the target information if the candidate is shifted by only one or two pixels. When comparing the result of using the more computationally expensive expensive configuration in Fig. 20 with a less computationally expensive configuration that has less candidate levels and a higher stride value in Fig. 21, we can clearly appreciate that the first configuration is able to lock on the target for longer.

V. FUSION BASED TRACKING

A. METHOD AND IMPLEMENTATION

To implement the fusion based tracker, we combined the extracted information from the candidates of the ColorTracker and the GradientTracker to select the most similar candidate in every frame. In order to do so, we modified the *ColorTracker* class into a *FussionTracker* class, realizing the following changes in each stage:

1) INITIALIZATION: .
Apart from the parameters of the *ColorTracker* class, we added an extra one to specify the number of bins when extracting the HOG descriptors of the candidates and selecting which features to use for the tracking:

- Color bins (*cbins*): Defines the number of bins for model and candidate color histograms.If this value is

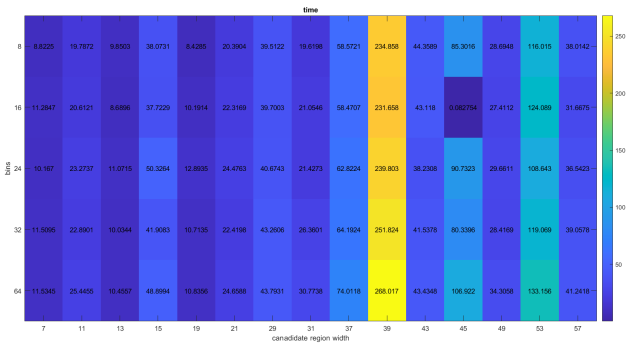

Fig. 19. Processing time obtained for ball2 sequence using HOG tracker for different combinations of number of histogram bins and width of search region for candidates.

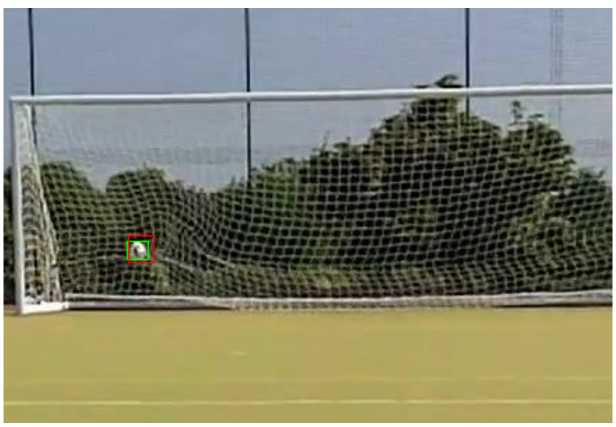

Fig. 20. Results obtained for ball2 sequence using 16 bins, 19 levels and stride 1.

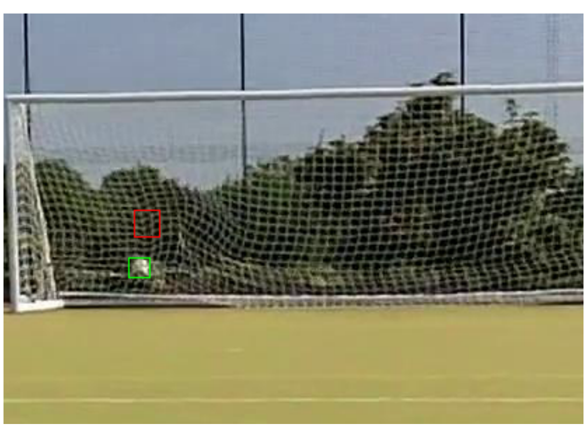

Fig. 21. Results obtained for ball2 sequence using 16 bins, 9 levels and stride 2.

set to 0, color histogram tracking is not activated.

- HOG bins (*gbins*): Defines the number of bins for model and candidate HOGs.If this value is set to 0, HOG tracking is not activated.
- Levels (*candidate_levels*): Defines a grid that contains the bounding box position for each candidate for each frame. The number of candidates is obtained as *Candidates = (2L+1)(2L+1)*.
- Step (*candidate_step*): Define the vertical and horizontal separation from one candidate to another.
- Histogram type (*track_type*): Defines the color channel to be considered according to boolean flags for color histogram based tracking.

Model initialization is still performed at the beginning of the video sequence using the first ground truth bounding box, and extracting the color histogram and/or HOG features as indicated by the *cbins* and *gbins* parameters.

2) CANDIDATE GENERATION AND EVALUATION: The candidate evaluation is done following the same strategy than in the two previous approaches. In this case, if both *cbins* and *gbins* are set to values greater than 0, the scores of the *ColorTracker* and *GradientTracker* are added together for each candidate. As the scores of bot features have different order of magnitude, after calculating the list of scores for every candidate, the scores are normalized before adding them together. https://www.overleaf.com/project/60a249106e229a65e2cdbe69

3) CANDIDATE SELECTION: The bounding box of the candidate with the lower score is returned.

B. DATA DESCRIPTION: VOT DATASET

To evaluate color histogram based tracking, four different sequences were considered: bolt1, bag, ball and road. The characteristics for each video sequence are summarized in table I.

C. EXPERIMENTAL METHODOLOGY

To run all of the tests for this third part, all of the sequences were firstly run with parameters known to have good performance in other videos for the color and gradient tracker. After finding the best configuration of each tracker, then the parameters of both trackers were manually tweaked when running together to improve the results to achieve better quantitative and qualitative results. In the specific case of the color tracker, all of the channels were tested to see which one had better result in each video. This tests were run in a different machine, causing some variations in the execution times. For reference, tracking the *bolt1* video using only color results in 42.2 ms/frame, whereas the previous machine only needs 14.8 ms/frame.

D. RESULTS AND ANALYSIS

1) Bolt1: For this sequence, the best Intersection Over Union score achieved was 0.46. The parameters used to get this result were the hue channel from the HSV colorspace, 62 color bins, 23 gradient bins, 5 levels of candidates and stride set to 1. The result of combining color and gradient

Fig. 22. Results obtained for the same frame using *ColorTracker* (top) and *FussionTracker* (bottom).

information results in smoother tracking than wen only using color information. The *FussionTracker* also performs better than the *GradientTracker* alone, as the last one is not capable of following the target during the whole sequence. The improvements can be appreciated in Fig. 22.

2) Bag: The best score achieved by the *FussionTracker* in this sequence was by using the red channel from the RGB colorspace, 6 color bins, 64 gradient bins, 8 levels and 1 pixel stride. This configuration gives an Intersection Over Union score of 0.34 at 154.67 ms/frame. In this case, the *FussionTracker* is barely able to outperform the *ColoTracker* at a big time cost, as the *ColorTracker* alone achieves a 0.339 Intersection Over Union Score with only 6 bins and needing 65.98 miliseconds per frame. The *GradientTracker* alone was unable to do any kind of useful tracking as it looses the target with every parameter combination tested. The only way of contributing to the *FussionTracker* was using a large abount of gradient bins. The main problem with this sequence is the variability of the target size, with combined with an erratic camera movement causes the tracker to include big portions of background in the bounding box, as the Fig 23.

3) Ball: The parameter combination selected for this sequence is also computationally expensive, but achieves much better results than the previous sequence. The used channel for the *ColorTracker* was hue from HSV and 32 color bins, 64 gradient bins for the *GradientTracker* , 7 candidate levels and a stride value of 2. With these parameters, the tracker is able to perfectly follow the target at all times and achieves 0.63 Intersection Over Union using 101 ms/frame. Without the help of the *GradientTracker*, the *ColorTRacker* is also able to track the ball during the whole sequence but much

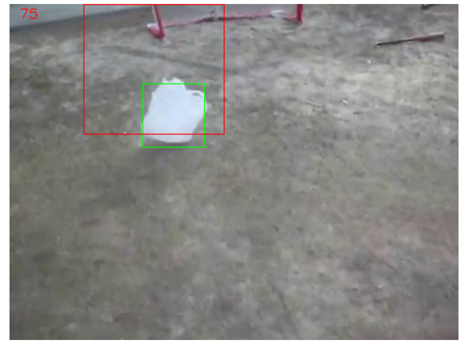

Fig. 23. Frame processed with the*FussionTracker*.

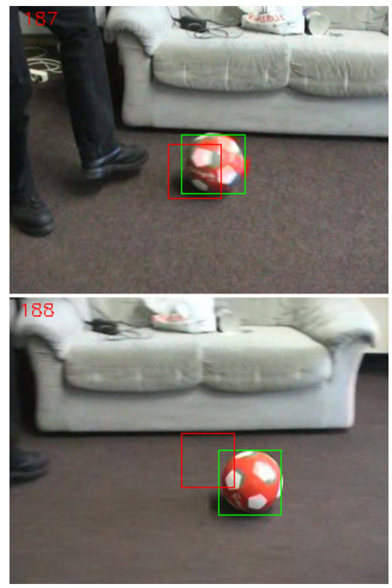

Fig. 24. The *FussionTracker* is able to follow the target even when big acceleration changes occur.

less smoothly, obtaining 0.51 Intersection Over Union and 27 ms/frame. The *GradientTracker* alone is unable to do the same. Therefore there is a tradeoff between performance and precission. By prioritizing precission with a big amount of close candidates, the *FuccionTrackes* is able to lock on the target even when there is big acceleration changes, as shown in Fig. 24.

4) Road: The last sequence also results in a very smooth tracking. Using a grayscale version of the frame and 8 bins for the *ColorTracker*, 16 bins for the *GradientTracker*, 4 levels of candidates and a stride of 4, the *FussionTracker* achieves 0.52 Itersection Over Union at 93.96 ms/frame. In this case, the *GradientTracker* is able to follow the target during the length of the sequence, except for the very final frames. When combining the scores of the gradient candidates and the color candidates, the tracker is able to follow the target even in those last frames, even when the target is occluded by other objects, as the Fig. 25 shows. With the parameters discussed above, the color tracker looses track of the target very early in the scene, meaning that the most important information is extracted using the HOG

Fig. 25. The *FussionTracker* to follow the target even when big occlusions occur.

descriptors.

VI. CONCLUSIONS

Color and gradient tracking are highly dependant on the data used when it comes to parameter optimization. Data characteristics that were observed to highly influence parameter configuration include camera movement, relative size and motion of the object of interest when the camera is considered as a reference point and degree of similarity between object of interest and background objects. This problem is even bigger when object of interest changes in size and this characteristic is not considered, meaning that the size of the tracking object is not updated at each frame. As a consequence, the algorithm adds background elements to the considered candidates or reduces the region of the tracking object that each candidate is able to capture. Tests suggest that, generally, the *ColorTracker* is more robust if the color channel is selected properly, although gradient information also helps when using the *FussionTracker*. Finally, search region size in dependant mainly on object size and range of displacement, where small objects with small range of displacements achieve better results by defining a small search region and vice-versa. However, in terms of performance, there is a trade-off between accuracy and performance when it comes to large objects with large ranges of displacements. For this cases, larger search regions may be required, which can be defined by either a high candidate level and small stride or a small candidate level and high stride. Better results are usually achieved when defining a region size with a small stride than when defining the same size with a large stride at the cost of a higher processing time. Accordingly and for the current work, the priority was accuracy when selecting the best parameter combinations during tests.

VII. TIME LOG

A. Maria Fernanda Herrera Perez

1) Color histogram based tracker development : : 6 hours

2) Gradient histogram based tracker development : : 2 hours.

3) Fusion tracker development : : 3 hours.

4) Color histogram based tracker evaluation: : 6 hours.

5) Gradient histogram based tracker evaluation: : 4 hours.

6) Fusion tracker evaluation: : 0 hours.

7) Results reporting for color histogram based tracker: : 3 hours.

8) Results reporting for gradient histogram based tracker: : 1 hours.

9) Results reporting for fusion based tracker: : 0 hours.

B. David Savary Martinez

1) Color histogram based tracker development : : 2 hours

2) Gradient histogram based tracker development : : 6 hours.

3) Fusion tracker development : : 3 hours.

4) Color histogram based tracker evaluation: : 0 hours.

5) Gradient histogram based tracker evaluation: : 1 hours.

6) Fusion tracker evaluation: : 5 hours.

7) Results reporting for color histogram based tracker: : 0 hours.

8) Results reporting for gradient histogram based tracker: : 3 hours.

9) Results reporting for fusion based tracker: : 4 hours.

REFERENCES

[1] S. Birchfield, ”Elliptical head tracking using intensity gradients and color histograms,” Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231), 1998, pp. 232-237, doi: 10.1109/CVPR.1998.698614.

[2] P. Fieguth and D. 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, 1997, pp. 21-27, doi: 10.1109/CVPR.1997.609292.

[3] https://www.votchallenge.net/