Histogram based object tracking

Abel Gebreslassie and Oluwatosin Alabi

I. Abstract

Histograms are widely used in computer vision due to their ability to represent data in a summarized way and providing a certain degree of robustness to what might be outliers or noise. A potential application area is object tracking in videos. Histograms of features(properties) that are less likely to vary for an object can be used for tracking with the use of a distance or similarity metric between extracted features. Moreover, one or more feature distances can be combined to give better results. In this work, color histogram feature based tracking and histogram of oriented gradients are tracking are implemented separately. The two models have approximately orthogonal failure modes hence their combination is expected to yield better results[1]. Candidate pixels for center of object are generated on a grid of certain size and padding[2]. Evaluation is done in a set of videos were different parameters, such as number of candidates and padding, are explored to find best configuration. OpenCV library is utilized for multiple image processing functionalities including creation and comparison of histograms. For performance evaluation the mean intersection over unions metric is used.

II. Color feature based tracking

A. Method and Implementation

A video frame can be represented using different color spaces that have one or multiple channels. In tracking, We can extract color histograms of object to be tracked from initial frame as features and compare them to estimate position of the objects in current frame. However, before we can extract features we first need to have candidate areas in current frame. To generate those candidates an nxn grid, where n is an odd integer, of candidate pixels for the center pixel of the object being tracked are generated with padding(δ). Objects are assumed not to change sizes as a result generated candidates will have the same size as the ground-truth in the first frame.

The model for comparison is set to the object's region of interest extracted from the first frame using the annotated bounding box. Color histogram features are then extracted for the model and each candidate where color channels and number of histogram bins can be varied to find best suited values for the specific video. The color histogram feature options implemented are red, green and blue channel form RGB colorspace, hue and saturation from HSV colorspace and grayscale.

Distances between histograms are then calculated using the battacharyya distance and the closest candidate to is predicted for the current frame. For gaining insight from visualization histogram plot and color-map are displayed along with the prediction and ground-truth bounding box.Battacharyya distance is for two color feature histograms H1 and H2 is given by

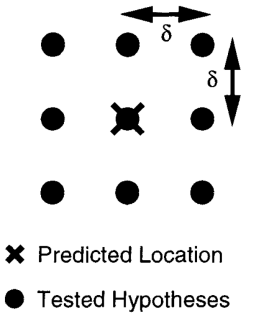


Fig. 1: Grid candidate generation[2]

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{H_1 H_2 N^2}} \sum_I \sqrt{H_1(I) H_2(I)}}$$

In the code implementation a parent tracker class is used because operations carried out by both color based tracker and gradient of histogram based tracker are similar with the only differences being on feature extraction and feature distance computation. Those methods are declared virtual in the parent method indicating every child class must implement it's own variant of them. In addition color based tracker has histogram plot and corresponding channel color-map visualization are declared and implemented in the color tracker class as they are specific to the class. The color track class implements color histogram extraction with its feature function implementation and implement the battacharyya distance on its distance function implementation.

Parameters required to initialize the tracker class are initial frame, bounding box annotation for object in first frame, the number of candidates to be generated for center pixel of object(should always be an odd perfects square integer), how far apart generated center position candidates need to be(δ), and the number of bins for histogram.

CandidateRegions is responsible for generating the number of candidates required with δ pixels between them whose corresponding bounding boxes are then extracted with extractRegion and passed to feature function for feature extraction. Finally, distance between all candidates and model features is computed to select the smallest distance center pixel as the prediction. Two helper functions are used to convert centers to bounding boxes and vise versa. The main function that is responsible for call and combining all those steps is

the track function. In addition to the current frame it takes the type of histogram to used for tracking. The available alternatives are provided as an enumeration with values RED, GREEN, BLUE, GRAY, HUE, SATURATION, and ALL. 'ALL' signifies the concatenation of histograms for all the other channels.

For color-map opencv's colormap jet was used were dark, and blue values represent lower channel values, blue green, green and yellow green represent medium values. Higher pixel channel values are shown with orange and red colors.

Fig. 2: OpenCv's jet colormap[3]

B. Experimental methodology

The code implementation was tested on different video sequences and several parameter configurations were evaluated to find best performing ones. First video sequence is 'Bolt1' which is a video recording of men's 100 meter and annotations in all frames for the winner of the race are provided. Different color features and parameter values, including the recommended number of candidates (81), were evaluated, only a representative set are show in the table below.

Color feature	candid/δ/bins			
	81/6/16	9/6/16	49/6/16	49/6/8
Red	9	31.3	13.3	3.5
Green	21.2	13.65	28.28	7.73
Blue	13.4	14.45	41.87	14.07
Hue	43.7	46.4	27.65	44.34
Saturation	16.7	17.6	16.82	16.89
Gray	11.28	12.01	11.69	7.5

TABLE I: mIoU of different color features and parameters [Bolt1]

Hue color feature has the best performance and works well with different configurations while grayscale feature has the lowest performance. Using very small (3 and below) values for δ makes the tracking fall behind when the athletes start running faster and very large (8 or above) values for δ make the tracking move from the athlete further away in the earlier frames where they are relatively running slower. Where are δ of 6 were able to keep a good balance both in earlier and later frames. This implies the δ parameter serves as how fast we expect the object to move from the current frame to the next frame.

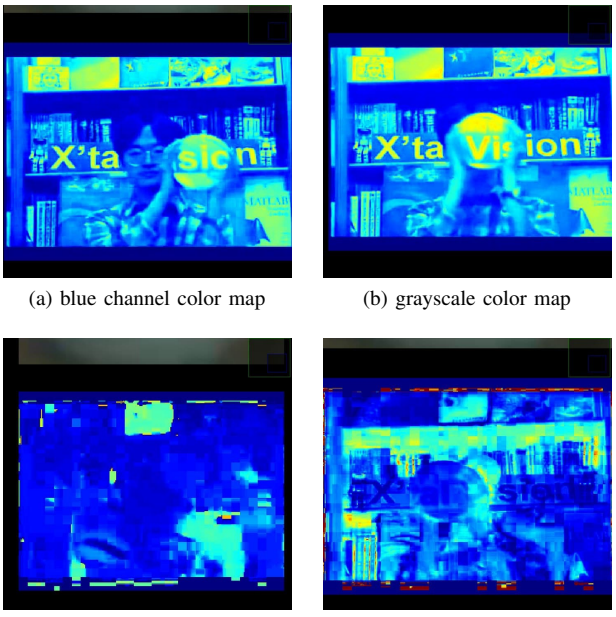
From the two configurations with 49 candidates and δ of 6 in the table it can be seen that the number of histogram bins affects color feature performance. The red, green and blue channel features significantly drop in performance when number of bins is lowered from 16 to 8 while hue channel improves. This is due to the fact that changing the number of bins means changing the dimension and values of the feature vector which directly affects the distance between different candidates.

The second video sequence experimented with is a transparent sphere ball moving around in different directions a camera zooming closer to the sphere. Generating more candidates results in better performance in this situations as the ball moves in different directions. Larger δ values work better because the video is taken from a close point of view and small movements result in larger pixel position change. Some of the parameter values evaluated for this sequence are presented in the table below.

Color feature	candid/δ/bins			
	9/5/16	25/8/16	81/8/16	81/8/8
Red	35.7	48.17	49.06	47.5
Green	32.15	57.3	58.09	57.57
Blue	38.15	55.73	56.65	58.18
Hue	20.6	13.8	11.98	25.36
Saturation	22.19	36.16	36.78	46.46
Gray	33.84	54.43	54.62	54.15

TABLE II: mIoU of different color features and parameters [Sphere]

The blue channels gives the best results in all cases, the RGB channel features in general perform better on this sequence. Grayscale features also give acceptable results while saturation feature is satisfactory. Hue feature has the worst performance in this sequence due to the low hue values in the video. From the colormaps in fig. 3 below it can be observed that the hue histogram features for most candidates will be similar as all regions in the frames have very low and similar hue values resulting in lower performance. For the remaining color features the sphere ball has a more distinct and higher values from other objects in the video frames. There are some regions that have similar saturation value as the ball and this lowered its performance slightly.



(a) blue channel color map (b) grayscale color map

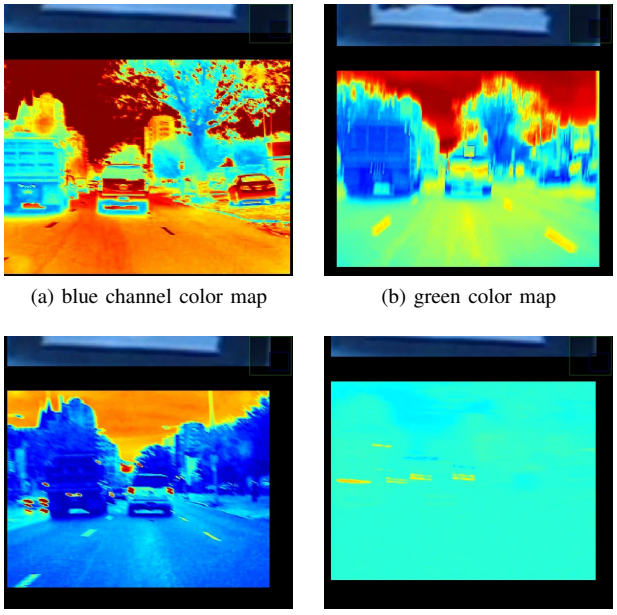
(c) hue channel color map (d) saturation channel color map

Fig. 3: Sphere colormap for different channels

The last video analyzed is 'car1' where an unstable and shaking camera inside a moving car is recording another car in front of it.

Color feature	candid/δ/bins			
	81/8/16	49/8/20	9/6/8	81/4/16
Red	63.57	57.94	44.17	55.54
Green	30.26	26.13	7.78	45.98
Blue	32.78	9.58	37.96	8.31
Hue	9.75	24.38	4.97	6
Saturation	21.62	2.64	17.92	17.18
Gray	57.65	27.33	37.47	51.86

TABLE III: mIoU of different color features and parameters [Car 1]



(a) blue channel color map (b) green color map

(c) red channel color map (d) hue channel color map

Fig. 4: Car1 colormap for different channels

The tracking behaviour for this sequence has similar behaviour as the sphere video. Hue color features have the worst performance, this is due to medium hue channel values all over the frame the colormap appears to be almost composed of one value expect around the car indicator lights. The blue channel performs low with 49/8/20 parameter values and this is due to the tracking jumping to another car that has similar blue values (see fig. 4). Similarly, for the green channel the lowest performance is recorded with the parameters 9/6/8 when tracking moves to the top of the tree that has relatively similar green histogram values as the car. From the red channel color map we can see the car looks different from other objects around it and this justifies the better performance with red channel histogram as color feature.

III. Gradient Feature Based Tracking

A. Method and Implementation

The gradient based feature tracker is similar to the color based tracker in all aspects with the only differences being use of Histogram of Oriented Gradients (HOG) and L2 distance instead of battacharyya. instead of color histogram features. As mentioned in the previous section, due to this similarity a parent tracker class was implemented from which both color and gradient feature classes inherit all functionalities and implement the class specific implementation of feature extraction and distance metric.

HOG is a popular descriptor used in a wide range of computer vision applications. It summarizes the gradient orientation of a certain image region using histogram. For every candidate region in the frame its HOG descriptor is generated and compared with the model using L2 distance to find the closes candidate. The L2 distance for two HOG discriptors H1 and H2 is given by

$$d(H_1, H_2) = \sum_I \frac{(H_1(I)H_2(I))^2}{H_1(I)}$$

B. Experimental methodology

Bolt1 video sequence was used as the first evaluation video for this method as well. Even though experiments and evaluation were carried out for multiple parameter values tracking results remained poor. While he is sprinting before reaching the finish line he keeps a structure that varies less and tracking is better. However, when he is about to reach the finish line his pose changes in addition his orientation from the camera changes (see fig. 5). This results in a larger difference between HOG descriptor we have for the model and his current orientation. As a result tracking starts to fail and ends up with a poor performance.

(a) model for tracking (b) beginning of tracking failure

Fig. 5: Gradient based tracking [Bolt1]

Gradient features	candid/δ/bins				
	9/5/20	9/5/16	9/3/20	49/5/8	81/5/16
HOG	16.4	2.15	13.83	1.8	1.3

TABLE IV: mIoU of HOG based tracking with different parameters [Bolt1]

The code was then tested on a basketball video where the aim is to track a basketball player with green jersey. A similar problem to the bolt sequence was encountered here. The player undergoes all sorts of pose changes, to defend his team of course, and this makes it hard for the HOG based tracking to make accurate predictions. All parameter configurations weren't able to yield a mIoU above 8%.

(a) Tracking failure due to orientation changes

(b) Tracking failure due to orientation changes

(c) model for tracking

Fig. 6: Gradient based tracking [basketball]

Gradient features	candid/δ/bins				
	9/5/20	9/5/16	9/3/20	49/5/8	81/5/16
HOG	7.73	3.3	7.66	3.39	3.04

TABLE V: mIoU of HOG based tracking with different parameters [Basketball]

The last video for evaluating is a ball in the air that was shot into a goal and is about to fall into the net. The size of the ball is very small and it's likely the HOG descriptor feature extracted for the model will not be good enough. Subsequently, the tracking results are of low performance.

Gradient features	candid/δ/bins				
	9/5/20	9/5/16	9/3/20	49/5/8	81/5/16
HOG	3	2.7	3.46	2.46	2.44

TABLE VI: mIoU of HOG based tracking with different parameters [Ball2]

(a) Poor ball tracking

(b) model for tracking

Fig. 7: Gradient based tracking [Ball2]

To observe how different distance metrics affect performance the metric of the gradient based detector was changed to battacharyya distance (only for the purpose of experimenting, see table below). From this we were able to see there was a performance gain in all video sequences, especially for basketball it was significant. Hence, the battacharyya distancee is more robust than L2 distance.

Video sequence	candid/δ/bins				
	9/5/20	9/5/16	9/3/20	49/5/8	81/5/16
HOG[Bolt1]	14.9	14.59	16.08	14.18	7.33
HOG[basketball]	59.97	56.15	42.91	55.72	55.5
HOG[Ball2]	4.9	4.9	3.46	15.99	18.88

TABLE VII: mIoU of HOG based tracking with different parameters using battacharyya distance

IV. Fused color and gradient feature tracker

A. Method and Implementation

The fusion tracker combines both color feature based tracking and gradient based tracking and it inherits most of it's functionality from the parent tracker class. It overrides the track method because it need to extract features and compute distances for both trackers. There are two boolean arguments that can be set to false to only use one of the trackers. If a boolean is set to false for one of the trackers then candidate distances for the tracker are set to zero.

B. Experimental methodology

First the bolt1 video was evaluated with both the recommended parameters, 9 bins and 100 candidates and δ of 6 (δ wasn't specified so the best delta from previous experiments was taken) those performed poorly as tracking moves in the wrong direction in early frames. Furthermore, parameters that had best performances in color feature and gradient feature based tracking. Color based tracking had a better performance on this video while the fused tracker has a lower performance than combining the two. It was observed the fused tracker starts to deviate from the athlete being tracked at about the same frame the gradient tracker alone starts to deviate. Therefore the lower performance is due to the gradient trackers lower performance. For all tables in this section $_c$ indicates color feature parameters while $_g$ indicates gradient tracking parameters.

Color feature	candid$_c$/δ_c/$bins_c$/$candid_g$/δ_g/$bins_g$	
	100/6/9/100/6/9	9/6/16/9/5/20
HUE	6.16	26.79

TABLE VIII: mIoU of color + gradient feature [Bolt1]

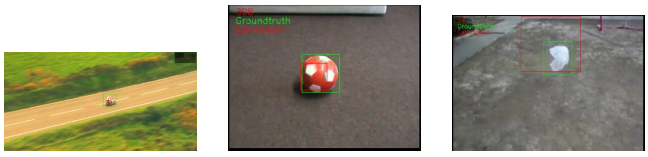
Three video sequences other than bolt1 were analysed with the fused tracker. Those are bag, ball and road. In the bag video, a plastic bag that is carried away by the wind is shown. Two people passing a red and white ball multiple times are shown in the ball video while the road video is aerial recording of motorcyclist riding in a forest road. Combinations of parameters that have good performance in previous sections were evaluated for those videos.

From the table we can see the first configuration has better performance for bag while the second configuration gives the best results for ball and road sequences. In the bag sequence it was observed tracking is reasonably good but performance metric seems to be lower than visual observation. This is mainly because the tracking bounding box size is kept constant while the annotation bounding box size might change

video sequence	candid$_c$/δ_c/bins$_c$/color feature/ candid$_g$/δ_g/bins$_g$		
	81/8/8/BLUE/ 9/5/20	81/8/16/RED/ 9/5/20	9/6/16/HUE/ 9/3/20
bag	42.92	28.19	28.12
ball	60.94	61.04	22.05
road	15.96	44.6	9.65

TABLE IX: mIoU of color + gradient feature [multiple videos]

(see fig. 8. This impacts performance for all videos analysed. In addition performance of the fusion tracker by turning one of the trackers was assessed (i.e. configurations in green in table IX).



(a) road (b) ball (c) bag

Fig. 8: Ground-truth and prediction bounding box sizes

For road video gradient only tracking has a mIoU of 36.20, color feature only tracking has a mIoU of 6.57 and combining then gives a better result (44.6). Similarly, for bag video gradient only, color only and fusion trackers have mIoU values of 10.5, 37.55 and 42.92 respectively. On the other hand, for ball sequence fusing both trackers has lower performance of 61.04 while color tracking alone achieves 63.48. From those results we can see that fusion leads often leads to better performance even though if one of the trackers has poor performance and gives larger distances for candidates then it will affect the fusion in a negatively.

Moreover, it can be observed that gradient based tracking performs well in the road sequence, this is due to the motorcyclist maintaining a less variant pose during the whole sequence.

V. Conclusions

Histogram based tracking was implemented using color feature histograms, histogram of oriented gradients(HOG) and fusing both. For all trackers implemented parameters such as number of candidates, number of bins in histogram and candidate generation padding (δ) play a major role in the performance of tracking. The feature distance metric also plays a major role as it was demonstrated that changing the metric from L2 distance to battacharyya distance for gradient tracker results in a significant performance improvement for some sequences. In color feature histogram tracking the color channel used plays a major role in tracking. Selecting channels that are more distinct for the object of interest than other objects in the frames results in better performance. On the other hand, HOG based tracker works better when the object being tracked has less pose variability and its relative orientation from the camera more or less stays the same. Fusing both trackers often results in better performance even though it might not be the case when one of the trackers performs bad and gives larger distances for candidates.

VI. TIME LOG

The time for the project is detailed below.

1) Color feature tracker implementation(4.1): 10 hours. 3 hours reading paper, 7 hours code implementation
2) Color feature tracker experiments(4.2): 9 hours. 6 hours experimenting with parameters and 3 hours writing report
3) Gradient feature tracker implementation(4.3): 4 hours.
4) Gradient feature tracker experiments and writing report(4.4): 7 hours. 4 hours experimenting and 2 hours writing report
5) Fusion tracker implementation(4.5): 3 hours.
6) Fusion tracker implementation(4.6): 6 hours. 4 hours experimenting 2 hours writing report

References

[1] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. 1998.

[2] Demetri Terzopoulos Paul Fieguth. Color-based tracking of heads and other mobile objects at video frame rates. 1997.

[3] Opencv documentation(https://docs.opencv.org/).