Robust tracker that uses detection, optical flow and colour segmentation

Julian Quiroga, Pablo Millán and Alfredo Restrepo

Pontificia Universidad Javeriana
Departamento de Ingeniería Electrónica
Bogotá, Colombia

Research Paper of ID Project 7422

Abstract. A multi-object tracker with a good performance level is presented. It uses a single shot detector, quantized colour histograms and optical flow. It tracks each vehicle that passes in front of a still street traffic camera, while keeping a record of the vehicles that have passed. The hue is quantized at 12 levels. For the optical flow output, both the circular median of the directions and the median of the speeds are computed.

Keywords: Tracking, Optical flow, Colour, Image detector, Hue, Circular histogram, Circular median

1 Introduction

The vehicles that pass in front of a camera that looks at a street from a high point view allows to keep a record of the vehicles that pass during daytime. The resolution of the camera is 720 x 480 pixels, and the output may be in AVI or MPEG-4 formats. The street (*carrera* 7, at the level of *calle* 41, in the sense north-south) is in the city of Bogota, Colombia. An image detector, Single Shot MultiBox Detector (SSD) [1], detects vehicles on the each of the frames of the video, and placing each of them within a box; see Figure 1. To each box, the vector machine assigns a label depending on the *type* of the detected vehicle; this is largely dependent on the size of the detection box. There are four types of vehicle: bus, car, motorcycle and bicycle. The boxes of each pair of consecutive frames are then placed into correspondence and an ID number is assigned to each pair of corresponding boxes. On occasions, a vehicle is not detected properly and this is alleviated with the help of colour and optical flow, using the corresponding detected boxes of previous and following frames. Also, at the times when a vehicle is entering or leaving the frame, the detector may run into problems, as is the case in Figure 1.

Tracking is faster than detection and, for long term tracking, both are needed [2]. The typical application is to follow an object in a hand held camera, here, the camera is fixed and the objects are entering and leaving the field of view.

One of the possible mistakes at the output of the SSD is to miss a vehicle in a frame, given another frame where the vehicle is detected, the use of colour and

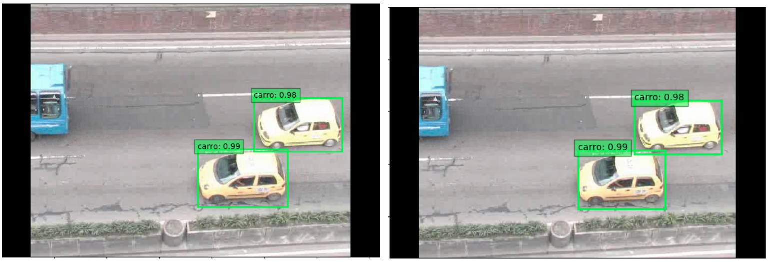

Fig. 1. Two consecutive frames of the video, after processed by the SSD. The blue bus, partially seen at the left was undetected. The two green detection boxes, in the left and right frames, are to be placed in correspondence.

optical flow based on colour helps to alleviate this problem. The colour of a box may be used as a characteristic of the vehicle, and the flow of the corresponding pixels may be used to estimate the position of the vehicle in other frames. Most of the time the velocity of the vehicles is constant but, on occasions, a vehicle changes lines, or it may stop when a traffic jam occurs.

The percentage of achromatic pixels of each box is computed and the hue of the remaining pixels is quantized and counted. That is, we partition the RGB colour cube into a central *achromatic solid* that includes the main diagonal,between the black and white colour points, and twelve surrounding *chromatic tetrahedra*, each tetrahedron being associated with a specific type of hue; see Section 4. Thus, each box has degree of achromaticity and a circular hue histogram with a corresponding support of a thresholded version of the histogram. If the degree of achromaticity is large, the average luminance is considered.

The majority of the pixels in a box are pixels of the vehicle, but the box includes as well other pixels such as asphalt pixels, which are achromatic. The colour of a pixel may be of two types: nearly achromatic or chromatic; for a chromatic pixel, 12 possible hues are possible. The colour of a box, i.e. its hue histogram together with the percentage of achromatic pixels (we call this percentage the *achromaticity* of the box) partition the pixels of the box into pixels that agree with the overall colour of the box, and pixels that do not. A pixel is said to agree with the colour of the box if its hue belongs to the support of the thresholded hue histogram, or, when the achromaticity of the box is large, if the pixel is nearly achromatic. For the pixels that agree with the colour of the box, the optical flow is computed. This allows to forecast the position of the box in consecutive frames by computing an median velocity (median speed and median direction). With (the negative of) the median velocity, it is also possible to assign a position to the corresponding box in previous frames of the video. Special cases for the optical flow algorithm are when the vehicle has stopped and when the vehicle is achromatic.

By using the colour category of achromatic pixels and, otherwise, quantizing the hue component of the colour of pixels, which we quantize into 12 hue subcategories, we alleviate up to a point some of the problems of noise and the need for some colour constancy against variations of the illumination of the street. In this way, for chromatic pixels, the information regarding luminance and saturation is left unused. To achieve a better colour constancy, more sophisticated techniques can be employed; for example, the ODOG filter of McCourt and Blakeslee [3], applied on each of the R, G and B components is a possible choice. See Figure 2. This is a topic that requires further exploration.

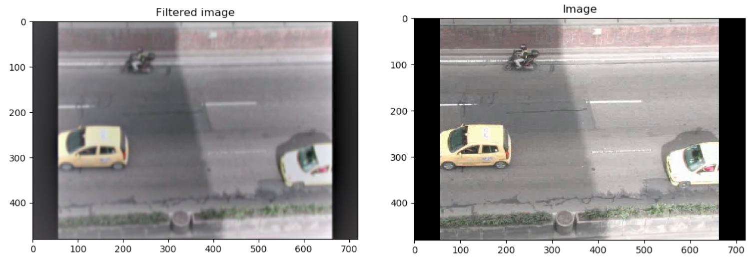

Fig. 2. The application of the ODOG filter to the image on the left produces the image on the right, where there is a bit more of colour constancy.

2 State of the art

Trackers are usually designed to track a single object in a video. We face the task of tracking several objects that pass in front of a camera, entering and leaving the field relatively often.

Trackers run in combination with detectors. The MOSSE tracker [4] for example, uses a detector based on correlation filters that, being implemented as FFT products, are very fast; combined with convolution tracking, a rate of 669 frames per second is achieved.

The median flow tracker [5], applied to our video gives impecable performance. The TLD tracker [2] follows the object from frame to frame while a detector corrects drift when necessary.

Current trackers are for example 'BOOSTING' [6], 'MIL' [7], 'KCF' [8](runs in real time, at 30 frames per second), 'TLD' [2], 'MEDIANFLOW' [5] (runs at 90 frames per second), 'GOTURN' [9] (100 fps), 'MOSSE' [4].

For our application needs, the best performance of the first six just mentioned trackers is MEDIANFLOW. Thus, it was our benchmark. Our algorithm MQR reached the same level of performance.

3 The frame object detector

The image detector is a method for detecting objects in images, SSD [1]. It improves over the YOLO and Overfeat methods is used. SSD is based on a convolutional neural network that uses a single deep neural network. It is relatively accurate and quite fast. Nevertheless, in some cases, it leaves a vehicle undetected. It adds two vertical black stripes at the left and right extremes of the frame, of a width of 54 pixels each. These two extremes of the frame are often problematic in the sense that vehicles entering and leaving the frame are often left undetected. The lack of performance of the detector is dealt with below, in Section 6.

4 On the use of colour

As in the case of a human tracker, the use of colour is advantageous. If you are told to look for a yellow flower in a prairie, we are likely to find it with relative easy. Yet, human colour constancy has not been successfully emulated in machines.

Even though each detected box contains pixels that surround the vehicle making the colour of the box noisy, the large majority of the pixels in the box do belong to the vehicle.

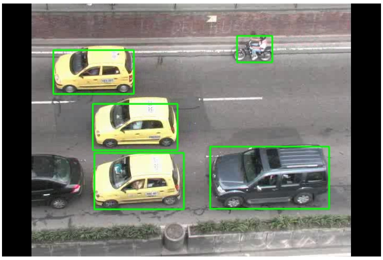

Fig. 3. A frame with achromatic vehicles

The presence of achromatic pixels in a box is ubiquitous since both the asphalt and the median lines are achromatic. The colour of the asphalt is gray of various shades, and the median lines are white. The tires are usually achromatic and the windows look partly so too. Nevertheless, achromatic pixels are not always noise; on the one hand, there are of course vehicles that are achromatic. Also, in a high dynamic range frame where many the pixels of a vehicle of a light colour, due to the saturation of the capture device of the camera, may turn white. We comment further on this last problem in Section 4.6.

4.1 Chromatic characterization

The colour of a pixel is characterized as follows. The RGB cube is partitioned into 12 *chromatic tetrahedra*. An *achromatic solid* is also used. If a pixel falls near the achromatic diagonal of the RGB cube, that is between the black and white vertexes of the cube, in the sense that the range (i.e. the max minus the min) of the RGB colour of the triple is small, the pixel is said to be of an achromatic colour. In this case the luminance (the distance to the black corner) is relevant. If a pixel is chromatic (i.e. it is not near the achromatic diagonal of the cube), the luminance is discarded and only the hue is used. The hue is quantized into 12 possible values, making the characterization of hue robust and less noisy.

The hue hexagon and the chromatic tetrahedra The chromatic hexahedron (a nonplanar hexahedron) is a topological circle embedded in the cube; we consider it to be the hue circle. It consists of the 6 edges (out of the 12 edges) of the cube that are disjoint from the black [0, 0, 0] and white [1, 1, 1] vertices.

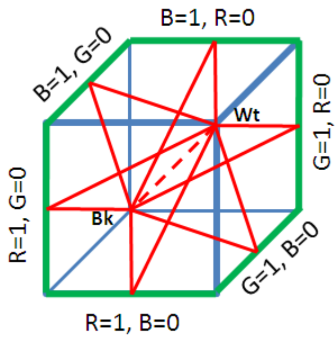

Fig. 4. RGB cube. The hue hexagon in green. The vertexes of the hexagon, together with the vertices of the red triangles determine the 12 chromatic tetrahedra.

The so-called *chromatic tetrahedra* provide the 12-value quantization of the hue; keep in mind that the hue is a cyclic variable. Each of the 12 tetrahedra has as two of its four vertices the points Black i.e. [R, G, B] = [0, 0, 0] and White i.e. [R, G, B] = [1, 1, 1]; the remaining two vertices of each tetrahedron are a pair of consecutive points in the cyclic list

- $[1, 0, 0]$ (*red*)
- $[1, 1/2, 0]$ (*orange*)
- $[1, 1, 0]$ (*yellow*),
- $[1/2, 1, 0]$ (*citrine*),

- $[0,1,0]$ (*green*),
- $[0,1,1/2]$ (*greenish cyan*),
- $[0,1,1]$ (*cyan*),
- $[0,1/2,1]$ (*bluish cyan*),
- $[0,0,1]$ (*blue*),
- $[1/2,0,1]$ (*bluish violet*),
- $[1,0,1]$ (*violet*), and,
- $[1,0,1/2]$(*purple*).

The colours indicated in parenthesis give the average chromaticity of the points in the corresponding tetrahedron.

4.2 The *achromatic solid*

The achromatic solid is given by the range (i.e. the max minus the min) ρ of the triple R, G, B being less that a small value, say $\rho < 0.04$ [reference]. The solid $\rho \leq 0.04$ is a hexagonal prism with axis the achromatic diagonal of the RGB cube. If a colour falls within this solid, it is classified as achromatic, regardless of the chromatic tetrahedron it may be in.

4.3 The chroma of a box

If a pixel is achromatic, it has an associated luminance; on the other hand, each chromatic pixel has one of twelve possible hues associated with it. Therefore, the pixels in a given box determine a histogram with two components, one component is a linear histogram of the luminances of the achromatic pixels and the second component is a twelve-bin circular histogram [10] of the chromatic pixels.

We define the *effective support* of the circular, hue histogram as the set of the bins for which the histogram has a value above a threshold, say $T = 1.5\%$. For many boxes, the effective support has a size between 3 and 5.

4.4 Comparing the colours of two boxes

In order to solve the problem of the correspondence of the boxes in two frames that contain approximately the same vehicles, the colour of the frames is advantageously used. Two measures of the similarity of the colours are used. One uses the effective support of the hue histograms, and the other uses a metric in the space of the boxed samples of the frames, taken as linear samples.

Initially, both the achromatic component and the support of the circular histograms are compared. Then, on the *subvectors* determined by the supports, the relative values are taken into account.

For example, consider two consecutive frames with three boxes each with thresholded hue histograms $[h_1, , , , , , , , , , h_{12}]$ given by

(frame 1:)
[0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0.3, 0.3, 0; 0.2] $\rightarrow$ support $= \{1, 10, 11\}$
[0, 0, 0.1, 0.2, 0.3, 0.2, 0, 0, 0, 0, 0, 0; 0.1] $\rightarrow$ support $= \{3, 4, 5, 6\}$
[0.2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.3, 02; 0.3] $\rightarrow$ support $= \{1, 10, 11, 12\}$

(frame 2:)
[0.15, 0, 0, 0, 0, 0, 0, 0, 0, 0.25, 0.3, 0; 0.2] $\rightarrow$ support $= \{1, 10, 11\}$
[0, 0, 0.15, 0.15, 0.3, 0.2, 0, 0, 0, 0, 0, 0; 0.1] $\rightarrow$ support $= \{3, 4, 5, 6\}$
[0.25, 0, 0, 0, 0, 0, 0, 0, 0, 0.25, 0.2, 0; 0.3] $\rightarrow$ support $= \{1, 10, 11, 12\}$

In this case, the correspondence between boxes is easily solved by making correspondent boxes those with the same effective support.

More sophisticated measures of the distance/similarity between two histograms are considered for example in [11]. However, as we point down below, this measure is not entirely appropriated to our case.

4.5 Distance between histograms

In [11], for vector data samples $A = [a_1, ...a_N]$ and $B = [b_1, ...b_N]$ of the same length N, where the data $a_i, b_i \in V = \{0, 1, 2, ...11\}$ range in the value set V, of size 12, the distance between the samples is given by the minimal cost of making the histogram H_A of one of the samples equal to the histogram H_B of the other, by changing the values one or several a_i's. In our case the samples have size N of the order of 10K (the size of a box) and take values are in the set $12 = \{0, 1, 2, ...11\}$. In order to use their algorithm, the size of the samples must be standardized.

The matching of the samples A and B results from changing the value one or several data a_i of sample A to make it correspond to values of data b_j of sample B. The cost of making each such change results in turn from a metric on V given by

$d(v, w) = |v - w|$ if $|v - w| \leq 6$ and
$d(v, w) = 12 - |v - w|$ if $6 < |v - w| \leq 11$

The distance between the samples A and B is then given by

$$D(A, B) = \min_{A,B} \sum_{i,j=1}^{N} d(a_i, b_j)$$

Problems with the histogram distance method of [11]. The method provides a distance between the points in the set of N-cyclic vectors with components in V, i.e. points of the space M^N. If we consider for example the sphere of radius 16 with center at $\theta := [0, ...0]$ which corresponds to the histogram $[10^4, 0, ...0]$. Then, the samples with histograms $[10^4 - 16, 16, 0, ...0, 0]$, $[10^4 - 16, 8, 0, ...0, 8]$ and $[10^4 - 16, 0, 4, 0, ...0, 4, 0]$ are all on the sphere of radius 16 centered at θ, yet only the first one is likely : the quantized hues do not spread that much.

4.6 Colours may loose saturation in high dynamic range frames

Given that we use an ordinary camera and the image frames may have a high dynamic range, particularly when the scene has both a sunny region and a shadowed region some pixels get a distorted RGB reading. When a vehicle is of a light colour and is in a sunny region, at a given vehicle pixel, one or more of the channels R, G and B of the camera may reach its maximal value, decreasing the saturation of the resulting colour and, at an extreme, making it white. For example, see Figure 2, a taxi cab of a light yellow, in a sunny region of a high dynamic frame will have a histogram with a high percentage of nearly achromatic (i. e. in the achromatic solid) pixels, but this will no longer be true when the same vehicle, in posterior frame, is in the shadow region, making harder to make the correspondence. To make a correspondence between the vehicle in the two frames, the values of the hue histogram are normalized by dividing by $1 - \alpha$, where α is the percentage of achromatic pixels in the boxes, whenever a vehicle is in a high dynamic range frame.

On the other extreme, in dark regions of a high dynamic range frame, it is possible that the camera detector give a zero R, G or B response to several pixels, changing the saturation of the colour and even making it black when the three channels are below threshold.

5 Optical flow: magnitude and direction

When a vehicle is undetected in a frame, and the vehicle has been detected in a previous frame (if the tracker is not running in real time, posterior frames can be considered as well), that is, when a box disappears, the optical flow of the pixels in the detected box can be used to estimate the position of a corresponding box in the problematic frame. This is done by applying a centrality estimator to the set of velocities given by the optical flow algorithm; estimators such as the median may offset the e.g. outlying flows of asphalt pixels in the box.

Given two consecutive frames of a video, modern optical flow algorithms [12] provide with a sensitive, fast detector of moving objects. The optical flow algorithm in [13] uses dense sampling and has a good contrast, while having a smoothing effect on descriptor matches; it is relatively slow, and an algorithm such as the one in [14] may be more appropriate for a real time implementation of the tracker.

The smoothness of the resulting optical flow map does not allow to estimate the boundary of the moving object with precision; actually, the boundary of the region of the nonzero velocity pixels is a smooth curve, of low curvature. This lack of precision at the boundaries impedes an accurate estimation of the distances between vehicles,by using optical flow alone, making the optical flow alone an unlikely tracker. See Figures 5 and 6.

The *median of flow* tracker algorithm in [5] derives its robustness both from a *forward-backward* validation method of the optical flow derived pixel trajectories, and from taking the median of the vertical and horizontal components of the best validated 50% of the trajectories. The tracker applies a Lucas-Kanade type optical flow algorithm to the boxes given by a frame detector. We use a similar approach, we give robustness to the optical flow estimation of the overall velocity of a box by computing both the *circular median* [15] of the directions of the flow and the *median* of the speeds of the flow.

Fig. 5. Above, the first of a pair of frames from which the direction component of the optic flow is computed with the algorithm in [13]. The direction colour code is as shown at the small circle below; cyan corresponds to the "West", or 180°, direction.

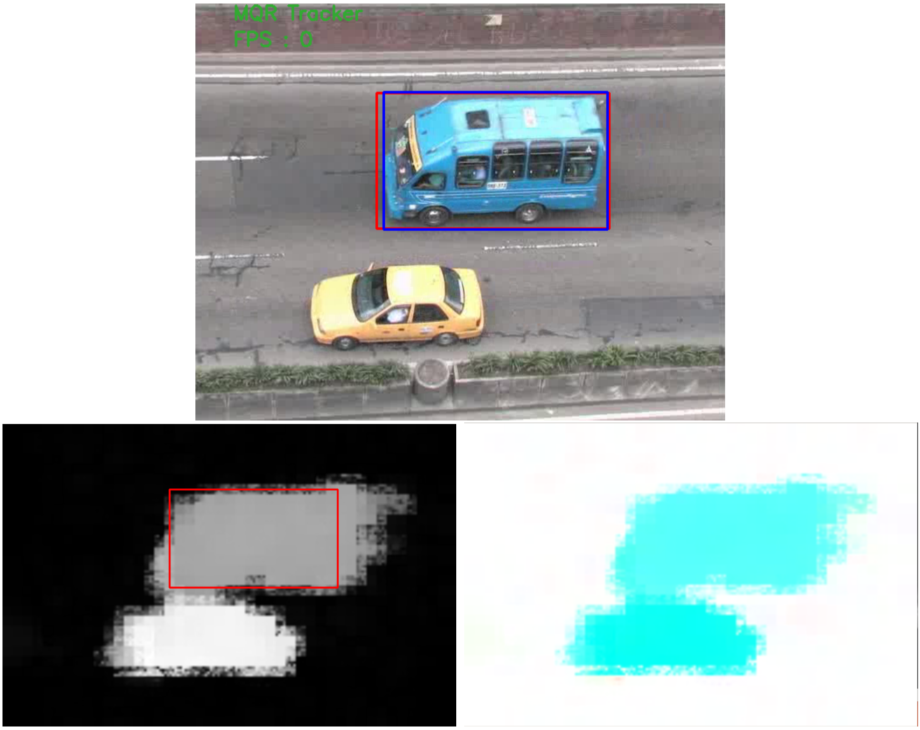

Fig. 6. Optical flow output (bottom). The speed map (left), gray coded, and the direction map, colour coded as indicated in Fig. 5, corresponding to the frame above. The taxi cab is moving at a faster speed.

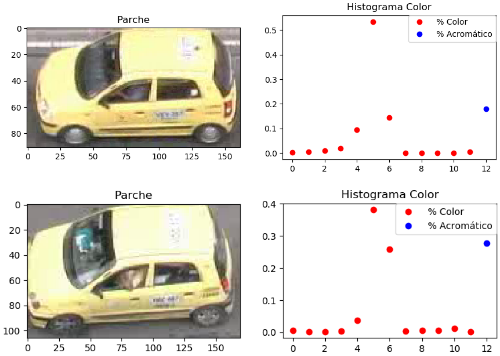

Fig. 7. The colour histograms of the two taxi cabs shown in Figure 1 . Even though they have the same support, $\{5, 6, 7\}$, the relative values are different, about 0.19 and 0.29. Also, the achromatic component is different.

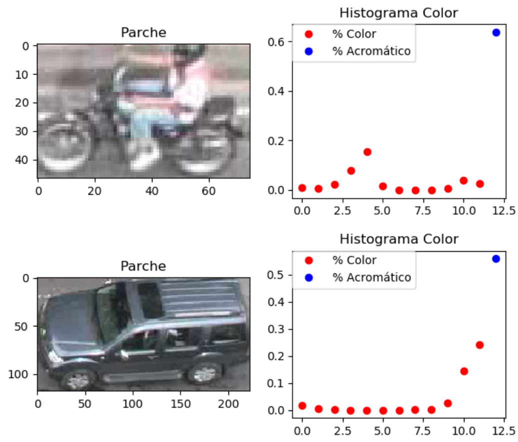

Fig. 8. The colour histograms of the achromatic USV and the motorcycle shown in Figure 3. The achromatic components are high in each case but the chromaticities are different.

6 Algorithm

Step 1: decimate video taking one out of every three consecutive frames, resequence the frames of the decimated video.

Step 2: get consecutive pair of (resequenced) frames F_i and F_{i+1}.

Step 3: obtain optic flow.

Step 4: apply SSD to each frame.

Step 5: get colour histograms and the effective supports of the histograms, for each of the bounding boxes in each frame.

Step 6: make correspondences between the boxes $B_{i,j}$ of F_i and the boxes $B_{i+1,j}$ of F_{i+1}.

Step 7: solve problem of undetected vehicles, if any.

Step 8: smooth box size changes, if any.

Step 9: update record.

Step 10: increment i.

Step 11: repeat until last frame is reached.

Step 12: output record and processed video.

For each box that is given by the vector machine, on the basis of the colour histogram of the pixels in the box, a *main colour* is computed. That is, even though the glasses, the windows and possibly the driver and the passengers, contribute to the colour histogram in ways different from the proper colour of the body of the car, assuming that the pixels corresponding to the car are the majority, by looking at similar histograms, taking into account the support, the main colour is considered. The algorithm proposed in [11] considers the possibility of matching similar values in two histograms.

6.1 Steps 7, 8: Solution of the problems detected in Step 6, of the main algorithm

Given the effective supports of the hue histograms of he boxes detected in Step 4 of the main algorithm, you can forecast the due position of a missing box. Also, a box that has changed size can have its size readjusted.

To find a missed box

– **Step 1:** In the frame where the vehicle was detected, compute optical flow of the pixels in the detection box.
– **Step 2:** Discard pixels for which the optical flow is not consistent (i.e. back and forward flow gives a different pixel).
– **Step 3:** Compute circular median of direction velocities and median of speeds.
– **Step 4:** With position and velocity of box, estimate its position in the frame(s) where it is missing.

Adjustment of box size

– **Step:** If a box changes size, the change is smoothed by making a gradual transition of the size with a moving average.

7 Results

The designed tracker will be referred to as MQR. When the tracker is run on a GPU GEFORFE GTX 1050 machine, using detection on every frame, there are no undetected vehicles, however it is not possible to achieve real time performance, resulting in a *delay rate* of about 1:3. Using detection every other other frame, real time is achieved; however, the rate of missed vehicles increases, at about two misses per minute, or two misses out of nearly 50 vehicles.

To test the algorithm, it was run on 5 videos of 30 seconds each.

We give as performance measures the center location error in a *precision plot* [16], see Figure 9; the plot is a cumulative distribution where the percentage of boxes having an offset less than a given (quantized) threshold level is indicated.

In Table 1, the bounding box overlap, or success is indicated. It is obtained from half the frames of Video 1 (every other frame) by averaging the resultibng overlap, with respect to the ground truth

Also, the rate of box matching success, or "found and followed" is indicated in Table 2.

Finally, regarding speed processing, in Table 3 we compare MQR with other six trackers, that have been, proposed since 2006, and are readily available on the internet.

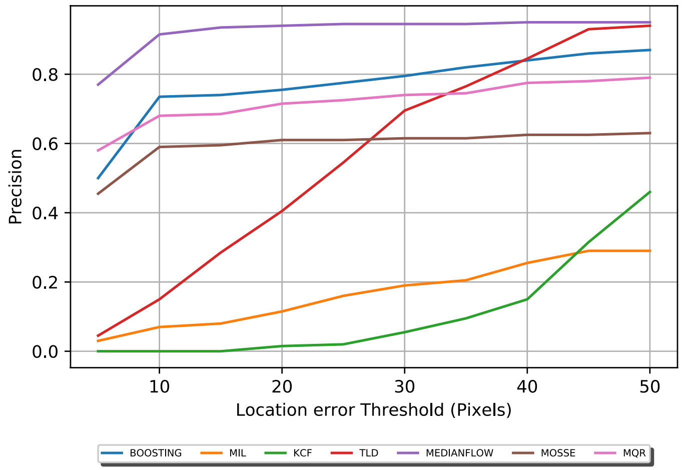

Fig. 9. Center location error. For a set of threshold levels (10, 20 ...), the percentage of boxes being off centered by an amount less than that of the threshold is indicated.

Table 1.

Tracker	mean b. box overlap
BOOSTING	0.83
MIL	0.45
KCF	0.69
TLD	0.82
MEDIANFLOW	0.95
MOSSE	0.61
MQR	0.81

Table 2.

Video	Performance Feature	cars	buses	motorcycles	bikes
1	Found & Followed	24	3	2	0
	Ground Truth	23	4	2	0
2	Found & Followed	16	3	2	0
	Ground Truth	16	3	2	0
3	Found & Followed	16	4	6	0
	Ground Truth	17	5	7	0
4	Found & Followed	12	3	1	0
	Ground Truth	12	3	1	0
5	Found & Followed	25	2	5	1
	Ground Truth	23	1	5	1

Table 3.

Tracker	speed
BOOSTING	16.13 fps
MIL	18.13 fps
KCF	67.89 fps
TLD	21.77 fps
MEDIANFLOW	273.8 fps
MOSSE	472.9 fps
MQR	48.86 fps

8 Conclusions

We have implemented a tracker, designed to track vehicles that move along a street of Bogota. It takes into account the possibility of changes in the speed of the vehicles.

The target templates used are patches of vehicles in the frames and colour histograms.

The performance is good. It can be improved regarding speed and rate of detection. It does not detect vehicles falsely.

To be probed further :

It is advisable to use a camera with a different aspect ratio, of a larger width, or perhaps, the combined image of two adjacent cameras.

Trackers are usually designed to track a single object. Perhaps an algorithm designed for tracking several objects could exploit different constraints.

The ODOG filter of Blakeslee and McCourt [3] has been used to predict the brightness perception, given on the luminance image, for, e.g., the White effect. It is hypothesized that if such a brightness is obtained for each of the R, G and B channels, colour constancy will be achieved [Land]. Although in our tests the ODOG filter performed below our expectations, its implementation in another colour space, such as Hering's, may give better results.

9 Attachments

,

1. Python code (860 lines), developed by Pablo Millán and Alfredo Restrepo.
2. Four Demos (30 seconds each)

References

1. W. Liu, D. Anguelov, D.E.C.S.S.R.C.Y.F., Berg1, A.: ssd: A single shot multibox detector (2016) https://arxiv.org/pdf/1512.02325.pdf.
2. Z. Kalal, K.M., Mata, J.: Tracking-learning-detection (2010) IEEE Trans. on Patt Anal. and Mach. Intell, VOL. 6, NO. 1.
3. Blakeslee, B., McCourt, M.E.: A multiscale spatial filtering account of the white effect, simultaneous brightness contrast and grating induction. Vision Research **39**(26) (1999) 4361–4377
4. D.S. Bolme, J.R. Beveridge, B.D., Lui, Y.M.: Visual object tracking using adaptive correlation filters (2010) www.cs.colostate.edu/ vision/publications/bolme_cvpr10.pdf.
5. Zdenek Kalal, K.M., Matas, J.: Forward-backward error: Automatic detection of tracking failure (2010) International Conference on Pattern Recognition.
6. H. Grabner, M.G., Bischof, H.: Real-time tracking via on-line boosting https://pdfs.semanticscholar.org/3a68/0eb285f31d6a72848530dec640d83c02a364.pdf.
7. Q. Dong, A.L.: Enhanced mil tracker with distribution field-based features and temporal fusion framework (2014) Electronics Letters, vol. 50, No. 24.
8. J.F. Henriques, R. Caseiro, P.M., Batista, J.: High-speed tracking with kernelized correlation filters (2014) arXiv:1404.7584v3.
9. D. Held, S.T., Savarese, S.: Generic object tracking using regression networks (2016) European Conference on Computer Vision (ECCV).
10. Mardia, K., Jupp, P.: Directional statistics (2000) John Wiley and Sons.
11. S-H Cha, S.S.: On measuring the distance between histograms. Pattern Recognition **35**
12. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation (2011) IEEE Trans on Pattern Anal. and Mach. Intell. , VOL. 33, NO. 3.
13. P. Weinzaepfel, J. Revaud, Z.H., Schmid, C.: Deepflow: Large displacement optical flow with deep matching (2013) ICCV - IEEE International Conference on Computer Vision, Sydney.
14. T. Kroeger, R. Timofte, D.D., Gool, L.V.: Fast optical flow using dense inverse search (2016) https://arxiv.org/pdf/1603.03590.pdf.
15. A. Restrepo, C.R., Vejarano, C.: Circular processing of the hue variable: a particular trait of colour image processing (2007) International Conference On Computer Vision Theory and Applications VISAPP, Barcelona.
16. Yi Wu, J.L., Yang, M.H.: Online object tracking: A benchmark (2013) CVPR.