

## Report #2

Ahmad Mostafa Mohamad

### Trackers comparison

---

- Method
- Results
- References

Method:

In order to create 10 scenarios to test the 6 trackers (which are):

- Boosting tracker [1]
- CSRT [2]
- KCF [3]
- Median flow [4]
- MIL [5]
- TLD [6]

All of those trackers are included in OpenCV library [7]

I followed the next steps:

- At first, I shot 3 videos:
  - o A Video with a blank white background and an artificial lighting with a big red object (ball)
  - o A Video with a blank white background and an artificial lighting with a small red object
  - o A video with a red light and a red big object

All the videos were captured using Realme XT rare camera with 1920x1080 resolution and 30 fps

- Then using open cv I created 7 more videos:
  - o A video with a black obstacle that covers the tracked object
  - o A video with a brightness that changes using HSV representation with changing the Value according to

$$f(v + 1) = f(v) + 55 * \sin\left(\frac{fps}{23.87324146378}\right)$$

The value of the denominator is the solution to achieve a full sine wave with a wavelength of 150 fps and an altitude of +-55 Value

- o A video with an RGB changing value that masks the object's color threshold and changes it with a random color every 15 frame
- o A video with an HSV representation and a changing value of the hue that masks the object according to

$$f(h + 1) = 180 + 179 * \sin\left(\frac{fps}{23.87324146378}\right)$$

- o A video with an exposure high increase for the whole video using by adding the filter

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 10 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

- o A video with random pixel values added to overlay the frame that changes every 30 frames
- o A video with a sharpening filter to express the edges more

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

All the videos have one source video which is the first video

- I wrote a script that processes the 6 trackers with the 10 videos using OpenCV and stores the videos, the fps average, the tracked time and the trajectory of the object on my drive

<https://drive.google.com/drive/folders/12-p9Dk3OIR1mbrORWgFOloheJ801CN0g?usp=sharing>

- I reviewed the 60 videos, the videos were all complete with no writing errors, then I put the numbers of frames that the object was detected in, number of true detection seconds and number of false detection seconds, and I added number the number of out box seconds for every tracker and every video

Tracker	Scenario	True tracking time	false tracking time	not detected time	out of bounding box time	average fps	rating
Tracker name	Blank						
	Small object						
	Brightness change						
	coloredRGB						
	coloredHSV						
	light						
	Noised						
	Red light						
	Sharpening						
	Virtual object						

Table 1: Sample table

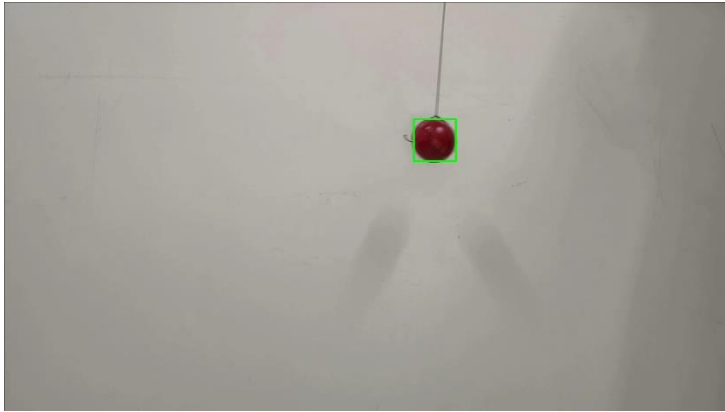


Figure 1: The original blank scnario video



Figure 2: small object scenario



Figure 3: RGB colored Scenario



Figure 4: HSV colored Scenario

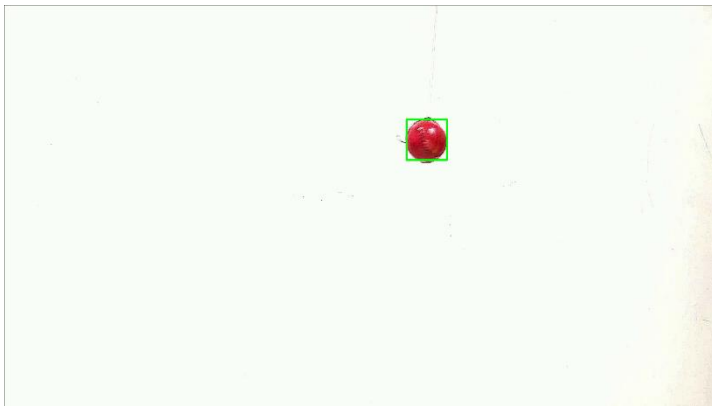


Figure 5: Exposure/light scenario

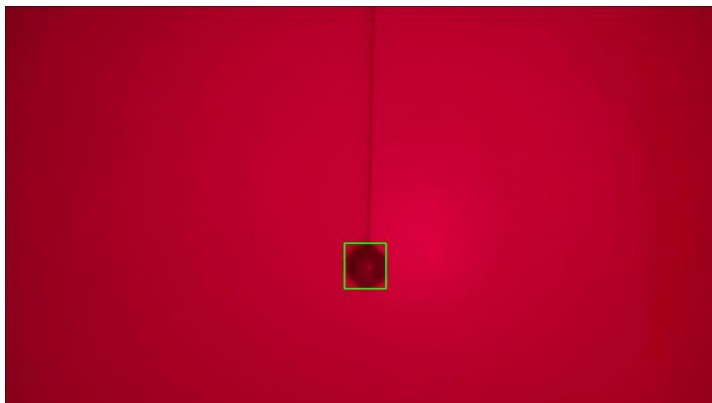


Figure 6: Red light Scenario

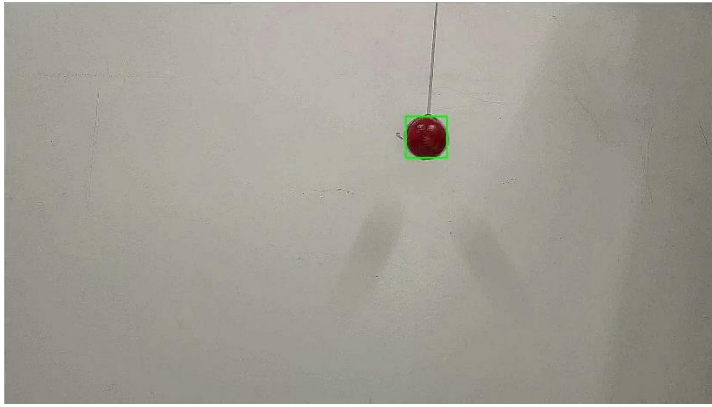


Figure 7: sharpened Scenario

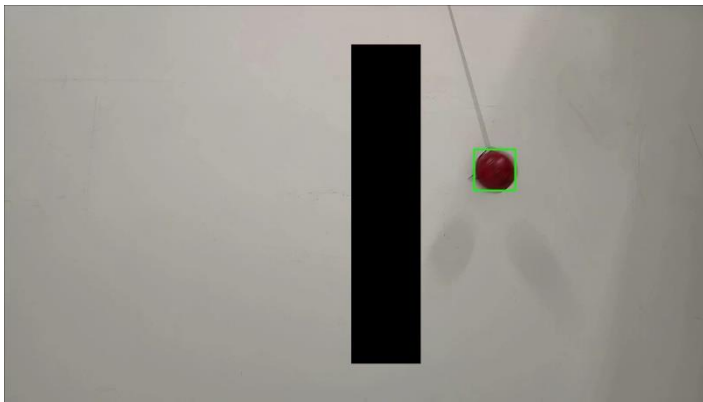


Figure 8: Virtual object scenario



Figure 9: the noised scenario

## Results:

The rating was calculated according to  $S = (T + 0.5 \times B)/15$

Where: S is the score, T is the true detection number of seconds, B is out of bounding box (at maximum half of the bounding box doesn't cover the item)

- Boosting tracker: [1]  
It is using linear classifier as weak classifier.  
It is not that good of a tracker, it got an average low score of 1.7, but it got 2.3 on light scenario and strangely 3.7 on red light scenario, with average frame of 11.5
- CSRT (Channel and Spatial Reliability Tracking): [2]  
It was created by integrating the Region based CNN (Faster RCNN) pre-trained object detection model that the OpenCV based CSRT.  
It got an average score of 9 which almost reached 10 but due to the low score it got on the Virtual object scenario 13.6.
- KCF (kernel correlation filter): [3]  
It got an average score of 1 and an average fps of 17.9
- Median flow: [4]  
It estimates reliability of a trajectory. A validation trajectory is constructed by backward tracking and compared to the trajectory in question.  
It got an average score of 4.3 and fps of 34.5, it did well on Blank, Small object, Brightness change and sharpening scenarios
- MIL (Multiple Instance Learning): [5]  
it is a type of weakly supervised learning algorithm where training data is arranged in bags. it got 0.3 score and 10 fps, didn't cross the barrier of 1 on any scenario
- TLD (Tracking-Learning-Detection): [6]  
It decomposes the long-term tracking task into tracking, learning and detection. The tracker follows the object from frame to frame. The detector localizes all appearances that have been observed so far and corrects the tracker if necessary. The learning estimates detector's errors and updates it to avoid these errors in the future. We study how to identify detector's errors and learn from them.  
It got a high score of 8.56 but a low fps of 3, Also the bounding box was so far from stable

#### References:

- [1] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," *BMVC 2006 - Proc. Br. Mach. Vis. Conf. 2006*, no. January, pp. 47–56, 2006, doi: 10.5244/c.20.6.
- [2] K. Farkhodov, S. Lee, K. Kwon, M. Jones, and B. Triggs, "Object Tracking using CSRT Tracker and RCNN Overview of the Object Detection," pp. 1–4.
- [3] Y. Liu, Y. He, Q. Tian, and J. Yang, "KCF Tracking Algorithm Based on Outlier Detection BT - Recent Developments in Intelligent Computing, Communication and Devices," 2019, pp. 129–135.
- [4] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," *Proc. - Int. Conf. Pattern Recognit.*, pp. 2756–2759, 2010, doi: 10.1109/ICPR.2010.675.
- [5] B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online Multiple Instance Learning," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 983–990, doi: 10.1109/CVPR.2009.5206737.
- [6] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, 2012, doi: 10.1109/TPAMI.2011.239.
- [7] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to OpenCV," in *2012 Proceedings of the 35th International Convention MIPRO*, 2012, pp. 1725–1730.