Comparing Various Tracking Algorithms in OpenCV

Akshat Mittal, Suryansh Singh and Manas Gupta

April 29, 2021

COMPARING VARIOUS TRACKING ALGORITHMS IN OPENCV

Akshat Mittal
Galgotias University
Greater Noida, India
akshat_mittal.scsebtech@galgotiasuniversity.edu.in

Suryansh Pratap Singh
Galgotias University
Greater Noida, India
suryansh_pratap.scsebtech@galgotiasuniversity.edu.in

Manas Gupta
Galgotias University
Greater Noida, India
manas_gupta.scsebtech@galgotiasuniversity.edu.in

Abstract—

Locating an item in consecutive frames of a video is known as object tracking. It is implemented by estimating the state of the concerned object present in the scene from previous information. Since the object has been tracked till the present frame, it's known how it has been moving. More simply, the parameters of the model are known. A motion model tells the speed and direction of motion of the object from previous frames.

Keywords—

Object Tracking, Motion History, OpenCV, Feature Classification, CSRT, KCF, Image Difference, Object Motion, Moving Camera, Comparison

Introduction—

This work builds an object identification system using two object detection algorithms simultaneously for higher accuracy and lower accuracy. The Channel and Spatial Reliability Tracker (CSRT) and Kernel Correlation Filter (KCF)

The Channel and Spatial Reliability Tracker (CSRT) used independently has greater object tracking precision but in lower FPS output and Kernel Correlation Filter (KCF) has a higher FPS output but with a slightly lower object tracking precision.

Used together these make the perfect combination for the objective of tracking the path of moving objects in live video. After implementing this project in Python 3.8 using OpenCV library 3.4, it was observed that the new system missed significantly fewer frames with lower latency pretty close to that of KCF.

Object tracking is a crucial area for computer vision. Tracking algorithms are used in numerous applications such as road management, and detection of face and the identity of the full human body.. Or learning how to track an item. Many algorithms have been developed over the years but the current state is yet not at a final solution for all use cases. Due to a lot of parameters and environment variables and locations (brightness, sequence features, background etc.), however it is almost impossible to create the universal tracking algorithm. [4, 7, 9]

Likewise, the decision of the proper calculation depends on its application and not just on its sort. Framework language, compiler and manual doing admirably can enormously influence execution just as a powerful calculation. Therefore, we chose to play out an examination of recently utilized tracking algorithms accessible in the OpenCV library.

OpenCV is a notable, intuitive library requiring designs and devices for PC vision algorithms; furthermore, it incorporates a huge arrangement of algorithms that are at first introduced prior to settling different pieces of an object tracking issue. Also, different streamlining strategies incorporate comparative projects, GPU registering and so on can be utilized to change the presentation of the chosen calculation. [6]

This paper endeavors to give tracking correlations and execution of algorithms, remembered for the OpenCV library. What's more, the fundamental standards introduced algorithms and effectiveness are examined.

Algorithm detecting features

While tracking is an exceptionally basic computer-vision issue and OpenCV is a generally utilized python computer-vision library, tragically, a couple of algorithms are accessible in the library. For our testing, we have utilized three element indicators, three unadulterated trackers and one complex tracking structure.

Feature indicators are not actually trackers, they simply attempt to discover the object set independently. It works this way: we have two pictures, one picture of something we need to follow and another first casing of a video or edge found in a live stream. We can get the principal picture from the subsequent picture by doing rectangular determination. Feature indicators and attempt to discover different features in picture of an object and attempt to track down the best guide of these features autonomous current.

This works best if the picture of the object is adequately huge and the actual object has accessible features like edges and surface. Something can turn unreservedly on a casing plane or marginally pivot on different planes while confronting practically a similar path. In OpenCV, we use findHomography () capabilities to discover the change between the comparing keys and the assignment viewTransform () to check focuses. We have utilized a marginally altered code from the OpenCV text model [8] to test feature features in the track.

The explanation we can't say about feature machines for genuine devotees is their inconsistency between outlines. Devotees follow the path, the investigators just track down the best match with two pictures that lead to outrageous instability,especially when you miss something that is trailed by something presently very much like the one portrayed before.

Another issue with just locators is conceivable where an object is like at least one objects in an image or a piece of a rehashing structure (for example windows, wall, and so forth) This can be an issue with tracking as a rule yet might benefit from outside input by zeroing in on something. This is the thing that fans come in for.

ORB [5], ISURF [2], SIFT [1] are three helpful features machines in OpenCV. The initial two utilize gliding point numbers yet you are protected. Third uses numbers and is subsequently not direct but rather quick too you have an amicable permit. Fundamental SIFT ideas and SURF algorithms and their applications through OpenCV can be found in [7] and [8].

Literature Review—

Nileshsingh V. Thakur (2017) et al, proposed an object detection framework discovers objects of this present reality present either in an advanced image or a video, where the object can have a place with any class of objects to be specific people, vehicles, and so on In request to distinguish an object in an image or a video the framework requirements to have a couple of parts to finish the undertaking of distinguishing an object, they are a model information base, a feature locator, a hypothesis and a hypothesis verifier. This paper presents a survey of the different methods that are utilized to identify an object, restrict an object, classify an object, separate features, appearance data, and some more, in images and recordings. The remarks are drawn dependent on the examined writing and major questions are likewise distinguished applicable to the object detection. Data about the source codes and on the web datasets is given to work with the new analyst in the object detection region. A thought regarding the conceivable arrangement for the multi class object detection is additionally introduced. This paper is appropriate for the scientists who are the novices in this space.

Dr. Rakesh Singhal (2017) et al. introduced object detection and tracking is one of the basic spaces of exploration due to
routine change moving of object and variety in scene size, impediments, appearance varieties, and sense of self movement and light changes. In particular, feature determination is the imperative part in object tracking. It is identified with numerous constant applications like vehicle insight, video reconnaissance and so on. To conquer the issue of detection, tracking identified with object development and appearance. The majority of the calculation centers around the tracking calculation to smoothen the video arrangement. Then again, hardly any strategies use
the earlier accessible data about object shape, shading, surface, etc. Tracking calculation which consolidates above expressed boundaries of objects is talked about and broke down in this exploration. The objective of this paper is to break down and survey the past approach towards object tracking and detection utilizing video groupings through various stages. Additionally, recognize the hole and propose another way to deal with improving the tracking of objects over video outlines.

Kuntal Dey (2018) et al. proposed Object detection is the distinguishing proof of an object in the image alongside its restriction and arrangement. It has widespread applications and is a basic segment for vision based programming frameworks. This paper tries to play out a thorough study of current object detection algorithms that utilize profound learning. As a component of the review, the subjects investigated incorporate different algorithms, quality measurements, speed/size compromises furthermore, preparing techniques. This paper centers around the two sorts of object detection algorithms-the SSD class of single step indicators and the Faster R-CNN class of two stage indicators. Procedures to develop identifiers that are convenient and quick on low fueled gadgets are moreover tended to by investigating new light weight convolutional base designs. At last, a thorough audit of the qualities and shortcomings of every indicator drives us to the current situation with the craftsmanship.

Proposed Model—

As already demonstrated in the Literature Review, object tracking algorithms have already been implemented in the past [1][2][3][5]. To solve the unique problem of maintaining high accuracy with low latency, we are going to use two object tracking algorithm implementations simultaneously.

CSRT due to its greater object tracking precision but lower FPS output and KCF has a higher FPS output but with a slightly lower object tracking precision.

Together both these will make the perfect combination for the objective of tracking the path of moving objects in live video.

This hybrid implementation will also include the capability of tracing the path of the object that is tracked.

The architecture design of the proposed model is shown in Fig 1. The implementation is based on it.

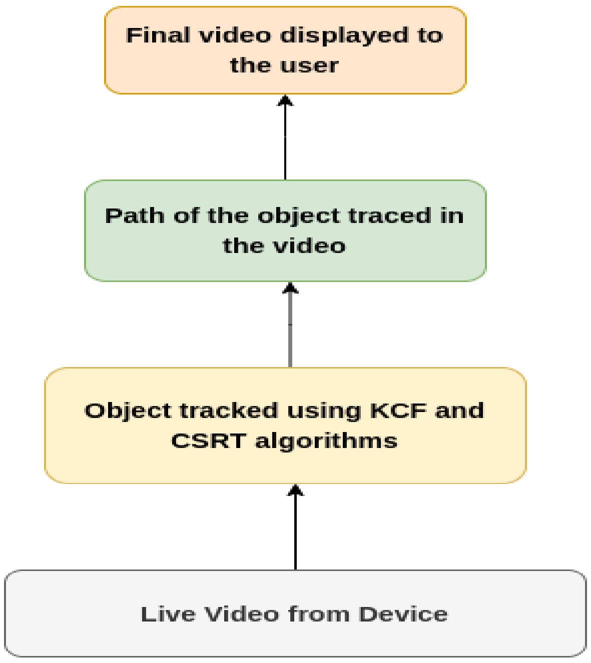

Final video displayed to the user
Path of the object traced in the video
Object tracked using KCF and CSRT algorithms
Live Video from Device

Fig 1 : Architecture Diagram

Implementation—

The code for this paper was written is Python 3.8 We used OpenCV library 3.4 We used a videos dataset, named PathTrack dataset, for multiple object tracking (MOT). PathTrack dataset features more than 3-4 person trajectories in 4-5 video sequences.

We tested the performance of the KCF and CSRT algorithms against the KCF and CSRT algorithms implemented together. The performance criteria we used was the tracking success rate and tracking consistency [11].

Pseudocode—

```
Input: video address and algorithm choice (KCF, CSRT or
    Hybrid)
Output: fps and frame drops per minute
    Initialisation :
 1: Object selected using mouse
 2: $fps$ = None, $frameskip = 0$
    LOOP Process
 3: for all frames of the video do
 4:     if object detected then
 5:         draw the coordinates on the video
 6:     else
 7:         $frameskip$++
 8:     end if
 9:     update $fps$
10: end for
11: return $fps$, $frameskip$
```

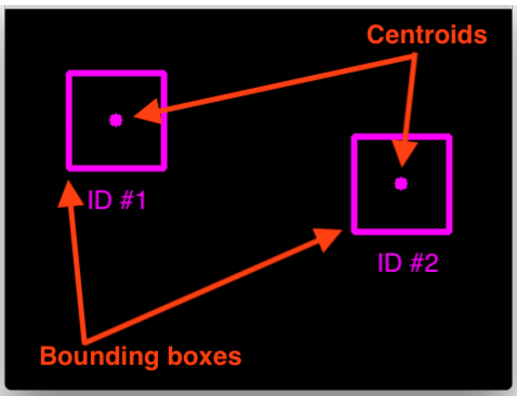
Assessing success—

For each video we used each of the following formulas: Scus (f) = | np∩rc | | rc∪kb | where Scuc (f) is a function of the effective condition of the framework f; rc binds rectangle back
from tracker and kb rectangular binding provided by
the truth of the earth. Basically we take the place of detention and separate it by the union area described above rectangular. This will provide as a scattering value of
is considered successful if it is greater than 0.5. [9]

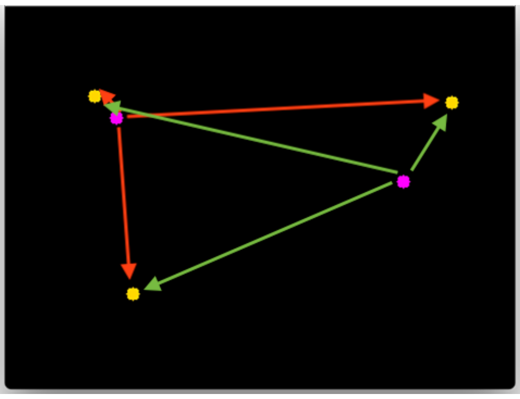
The centroid tracking algorithm --

This tracking algorithm is a multi-step process. We will review each of the steps in this section.

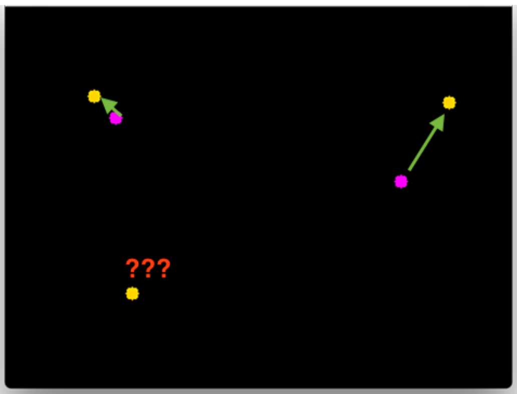
Step 1: Accept bounding box coordinates and compute centroid.



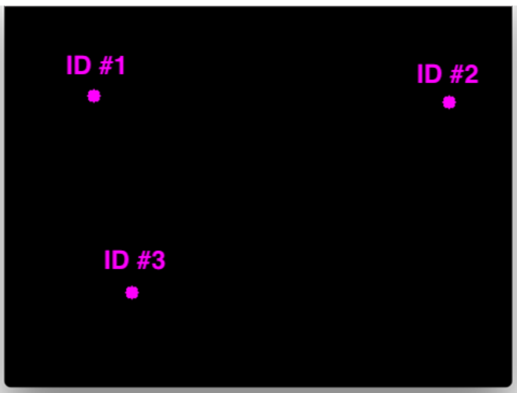
Step 2: Distance between new bounding boxes and existing objects.



Step 3: Update coordinates of existing objects.



Step 4: Register new objects.



Accuracy test—

As a proportion of the exactness of the calculation, we chose to utilize it. The rectangular scale found in the tracker measures ground exactness while the best precision is equivalent to 1 on the off chance that we utilize this equation:

Csec (f) = | rc | | kb |
where Csec (f) is found the errand of deciding exactness is right now getting looked at free f.

Time requires testing Clearly measure the time prerequisites of every calculation, we recommend that we basically gauge the hour of each casing: Cnit (f) = h where Cnit (f) is the time necessity of the calculation matter t is the time it took to deal with the current edge f.

Performance evaluation—

Step by step instructions to make algorithms while managing each expression of The issues portrayed in Table 1 were analyzed correspondingly in a way like Cnit (f) (see above) however just related recordings.

Algorithm enhancements—

Since this paper is about a particular execution of procedures remembered for OpenCV, ought to be referenced, that the last speed of the algorithms depends not just on plan, yet additionally with startup style and/or in great use.

The OpenCV library contains one significant strategy by working recreations on computer vision algorithms. It's a bunch of visual capacities and classes, about them work performed, generally altered for circle, characterized as a comparable activity.

Along these lines, at the point when an OpenCV library is incorporated with a particular examination structure, the checked segment of the calculation states it is consequently conveyed to a similar usefulness.

Result—

On evaluation of the performance of the three implementations, we found that the combined implementation of KCF and CSRT, provided a far superior result in tracking success rate measured by frame skips (Table 2) surpassing both KCF and CSRT. Tracking speed of the hybrid implementation was greater than CSRT but slightly lower than that of KCF (Table 1).

Table 1. FPS Throughput of all implementations

Videos	KCF		CSRT		KCF + CSRT	
	Highest FPS	Average FPS	Highest FPS	Average FPS	Highest FPS	Average FPS
Video 1	48	42	36	32	47	41
Video 2	44	39	33	29	46	36
Video 3	49	45	36	30	48	42
Video 4	41	35	31	28	40	34
Video 5	45	40	32	28	46	40

Table 2. Frame skips of all implementations

Videos	KCF	CSRT	KCF + CSRT
Video 1	39	26	19
Video 2	56	35	29
Video 3	110	75	63
Video 4	44	36	28
Video 5	96	81	62

Conclusion—

This paper has demonstrated that using multiple object tracking algorithms simultaneously can improve performance, that is, accuracy and FPS throughput of the implementation. More experiments can be done on various permutations of such algorithms to find effective solutions for different use cases.

References—

[1] Jiˇr ´ı Apeltauer, Adam Babinec, David Herman, and Toma´s Apeltauer. ˇ Automatic vehicle trajectory extraction for traffic analysis from aerial video data. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40(3):9, 2015

[2] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking,"in Proc. IEEE Int. Conf. Comput. Vis., 2009, pp. 261–268.

[3] K.-H. Jeong, P. P. Pokharel, J.-W. Xu, S. Han, and J. Principe, "Kernel based synthetic discriminant function for object recognition," in ICASSP, 2006.

[4] C. Xie, M. Savvides, and B. Vijaya-Kumar, "Kernel correlation filter based redundant class-dependence feature analysis (KCFA) on FRGC 2.0 data," in Analysis and Modelling of Faces and Gestures, 2005

[5] W.-L. Lu, J.-A. Ting, J. Little, and K. Murphy, "Learning to track and identify players from broadcast sports videos," IEEE Trans.
Pattern Anal. Mach. Intel., vol. 35, no. 7, pp. 1704–1716, Jul. 2013.

[6] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2411–2418, 2013.

[7] F. Wang, G. T. Jiao, and Y. Du, "Method of fabric defect detection in based on mathematical morphology," Journal of test and measurement technology, vol. 21, pp. 515-518, 2007

[8] W. Luo, T.-K. Kim, B. Stenger, X. Zhao, and R. Cipolla, "Bi-label propagation for generic multiple object tracking," in Proc. IEEE
Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2014, pp. 1290–

1297.

[9] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1396–1404, 2017.

[10] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle
filter for tracking multiple interacting targets," in Proc. Eur. Conf.
Comput. Vis., 2004, pp. 279–290.

[11] Patrick Sebastian, Yap Vooi Voon, Richard Comley. (2020) Parametric Tracking Across Multiple Cameras with Spatial Estimation. IETE Journal of Research 0:0, pages 1-15.

[12] Karanbir Chahal1 and Kuntal Dey, "A Survey of Modern Object Detection Literature using Deep Learning", International Research Journal of Engineering and Technology (IRJET), Volume 8, Issue 9, 2018

[13] Kartik Umesh Sharma and Nileshsingh V. Thakur, "A Review and an Approach for Object Detection in Images", International Journal of Computational Vision and Robotics, Volume 7, Number 1/2, 2017.

[14] Mukesh Tiwari, Dr. Rakesh Singhai, "A Review of Detection and Tracking of Object from Image and Video Sequences", International Journal of Computational Intelligence Research, Volume 13, Number 5 (2017).