

Comparing Various Tracking Algorithms In OpenCV

Suryansh Pratap Singh ^a, Akshat Mittal ^b, Manas Gupta ^c, Soumalya Ghosh ^d, Anupam Lakhanpal^e

^{a,b,c,d,e}School of Computer Science, Galgotias University, India.

^a suryansh_pratap.scsebtch@galgotiasuniversity.edu.in

^b akshat_mittal.scsebtch@galgotiasuniversity.edu.in

^c manas_gupta.scsebtch@galgotiasuniversity.edu.in

^d soumalya.ghosh@galgotiasuniversity.edu.in

^e anupam.lakhanpal@galgotiasuniversity.edu.in

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

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. Algorithms that track objects using this motion model are known as object tracking algorithms. There is a multitude of algorithms that can be used for the same purpose. The trouble is finding out which object tracking algorithms are best for a particular use case. In this work, we have compared two object tracking algorithms and their hybrid to find which performs better in the case of a live feed.

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

1. 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 the 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 a 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, for this work, it was chosen 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 work endeavours 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.

2. Literature Review

Nileshsingh V. Thakur (2017) et al, proposed an object detection framework that discovers objects of this present reality present either in advanced image or 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 work presents a study of the

various strategies that are used to distinguish an article, confine an item, arrange an item, separate highlights, appearance information, and some more, in pictures and accounts. The comments are drawn ward on the analyzed composition and significant inquiries are moreover recognized appropriate to the item location. Information about the source codes and on the web datasets is given to work with the new expert in the item discovery locale. An idea in regards to the possible plan for multi-class object discovery is also presented. This work is suitable for the researchers who are learners in this space.

Dr Rakesh Singhal (2017) et al. introduced object detection and tracking as one of the basic spaces of exploration due to routine change of object movement and variety in scene size, impediments, appearance varieties, and sense of self-movement and light changes. Specifically, include assurance is the basic piece of the article following. It is identified with different steady applications like vehicle information, video reconnaissance, and so forth To vanquish the issue of recognizable proof, coming up next is identified with fight headway and appearance. The greater part of the calculation bases on the accompanying assessment to smoothen the video plan. Scarcely any strategies use the earlier accessible data about object shape, covering, surface, etc Following assessment which joins the above-conveyed cutoff points of articles is examined and isolated in this examination. The goal of this work is to separate and study the previous methodology towards object following and location using video groupings through different stages. Furthermore, perceive the opening and propose another approach to manage to improve the following of articles over video diagrams.

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 work 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, and preparing techniques. This work centres 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 developing identifiers that are convenient and quick on low fueled gadgets are moreover tended to by investigating new lightweight 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.

3. 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, three-element indicators were utilized, three unadulterated trackers and one complex tracking structure.

Feature indicators are not 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 a 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 view transform () to check focus. 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 use 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].

4. 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.

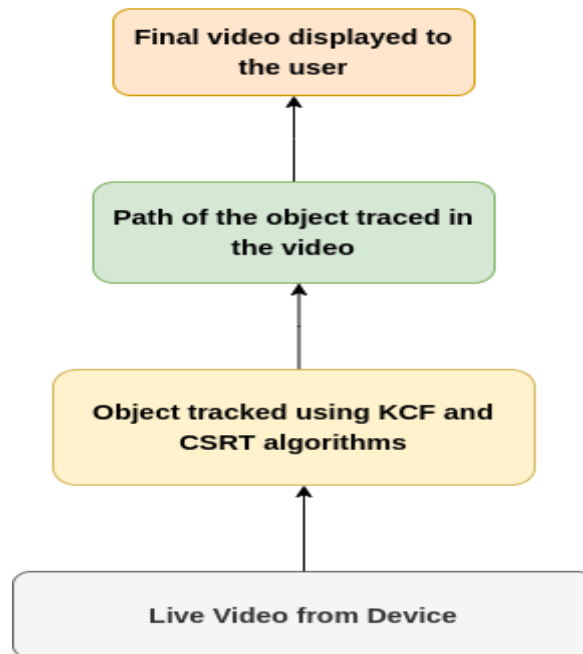


Fig 1: Architecture Diagram

5. Implementation

The code for this work was written in Python 3.8 We used OpenCV library 3.4 We used a video 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].

6. 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** $fps, frameskip$

7. Assessing success

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

8. Centroid Tracking Solution

The Centroid Tracking Solution is a multi-step process. The steps are listed down below in Figure 2.

Stage 1: Accept bouncing box arranges and process centroid.

Stage 2: Distance between new bouncing boxes and existing articles.

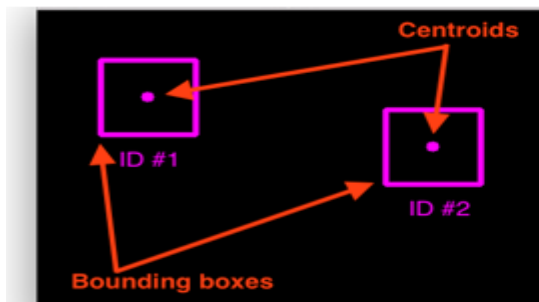


Figure 2.1

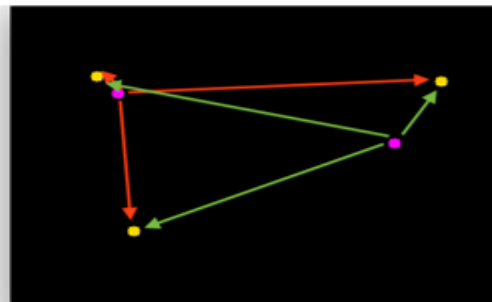


Figure 2.2

Stage 3: Update directions of existing items.

Stage 4: Register new items.

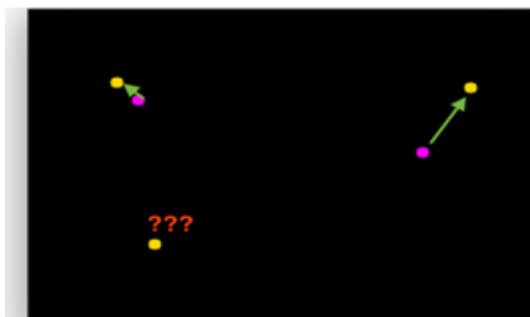


Figure 2.3

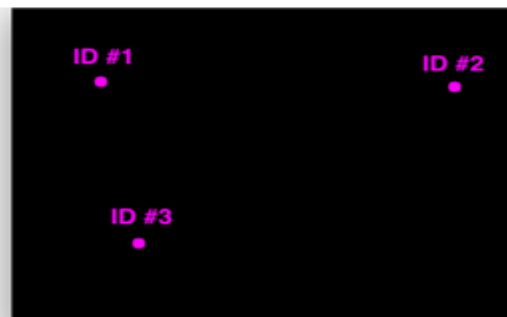


Figure 2.4

Figure 2: Centroid Tracking Algorithm Graphs

9. 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 the time prerequisites of every calculation, we recommend that we 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.

10. 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.

11. Algorithm enhancements

Since this work 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 the 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 which work is performed, generally altered for a 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 similar usefulness.

12. Result

On the 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 by 60%. The tracking speed of the hybrid implementation was greater than CSRT by 30% but 10% 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 per minute

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
---------	----	----	----

13. Conclusion

This work has demonstrated that the use of a combination of object tracking algorithms simultaneously can improve performance, that is, accuracy and FPS throughput of the implementation. In the case of this work, the KCF + CSRT combination outperformed the native algorithms in all metrics bar one. This work can be expanded by performing more experiments on various permutations of different object tracking algorithms to find effective solutions for different use cases.

References

1. Jiřr 'l Apeltauer, Adam Babinec, David Herman, and Toma's Apeltauer. 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 behaviour 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 is 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 Chahall 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)