Computer Vision II (EE 554/CSE 586) Project 1: Improved Siamese Network for tracking based on Lucas-Kanade **(SiamLK)**

Friday 10th April, 2020

Sweekar Sudhakara, Sai Raghav Reddy Keesara, Srikanth Banagere Manjunatha and Laxmaan Balaji

Abstract

The Siamese tracker approach compares the initial template to patches in the search image using a sliding-window approach. Traditional Lucas-Kanade's is a template tracker that was previously applied to salient features like corners to compute optic flow. Taking inspiration from this, we try to implement this on multi-channel features maps produced in the Siamese Tracker and analyze the performance. On completion of this project, we aim to build an improved tracker that employs a traditional Computer Vision method on the recent Siamese tracker that is based on features learned from deep convolutional networks.

Contents

1 Introduction and Motivation

The regression based strategies to object tracking have long held appeal from a computational standpoint. Specifically, they apply a computationally efficient regression function to the current source image to predict the geometric distortion between frames. The state of the art trackers in vision are based on a classification based approach, classifying many image patches to find the target object. The state-of-the-art methods employ deep learning based methods which adhere to the classification-based paradigm to tracking.

The SiamFC based tracking employs deep learning based regression networks framework for object tracking. Since, it is a regression based approach, it can be made to operate extremely efficiently. The SiamFC based tracking can process 100 Frames Per Second (FPS) on a high-end GPU, in contrast to classification-based methods to object tracking that do not rely on deep learning - such as correlation filters - can compete with SiamFC in terms of computational efficiency. However, they often suffer from poor performance as they do not take advantage of the large number of videos that are readily available to improve their performance. Finally, regression based strategies hold the promise of being able to efficiently track objects with more sophisticated geometric deformations than just translation (e.g. affine, homographies, thin-plate spline, etc.) - something that is not computationally feasible with traditional classification-based approaches.

MOTIVATION:

Although reliable, SiamFC has a fundamental drawback i.e. it performs well on objects that have similar appearance to those seen in training and fails when the object's viewpoints have not been seen in the training. However, this network, along with the employment of tracking datasets with large amounts of object variation, should in principle overcome this generalization issue but it fails. In this project we attempt to solve this problem by employing the classical algorithm in computer vision literature- namely the Lucas & Kanade (LK) algorithm. Specifically, we employ the Inverse-Compositional LK (IC-LK) along with the Siamese regression network that shares many similar properties to SiamFC. In the IC-LK algorithm, the regression function adapts to the appearance of the previously tracked frame unlike SiamFC where the the regression function is "frozen" - partially explaining why the approach performs so poorly on previously unseen object viewpoints.

2 Related Work

The idea of integrating Lucas-Kanade with deep convolutional features has been explored by Wang et. al. (2017) [1] referred as Deep-LK. While their baseline is GO-TURN for deep convolutional features, ours is SiamFC. The motivation behind why such an integration works best for object tracking remains the same, i.e., Unlike Siamese Networks, LK adapts its regressor to the appearance of the currently frame that is currently being tracked. Siamese Networks' performs poorly on samples and view points that are not seen during training. In our approach We try to bring the best of both worlds by implementing a bi-directional feedback kind of mechanism by sending output of SiamFC prediction to LK and vice-versa. This way, the probability of losing object being tracked is minimized.

While Deep-LK trains the siamese deep features such that the warp parameter prediction is close to the ground truth we do not re-train out network rather apply Lucas-Kanade only at the tracking stage.

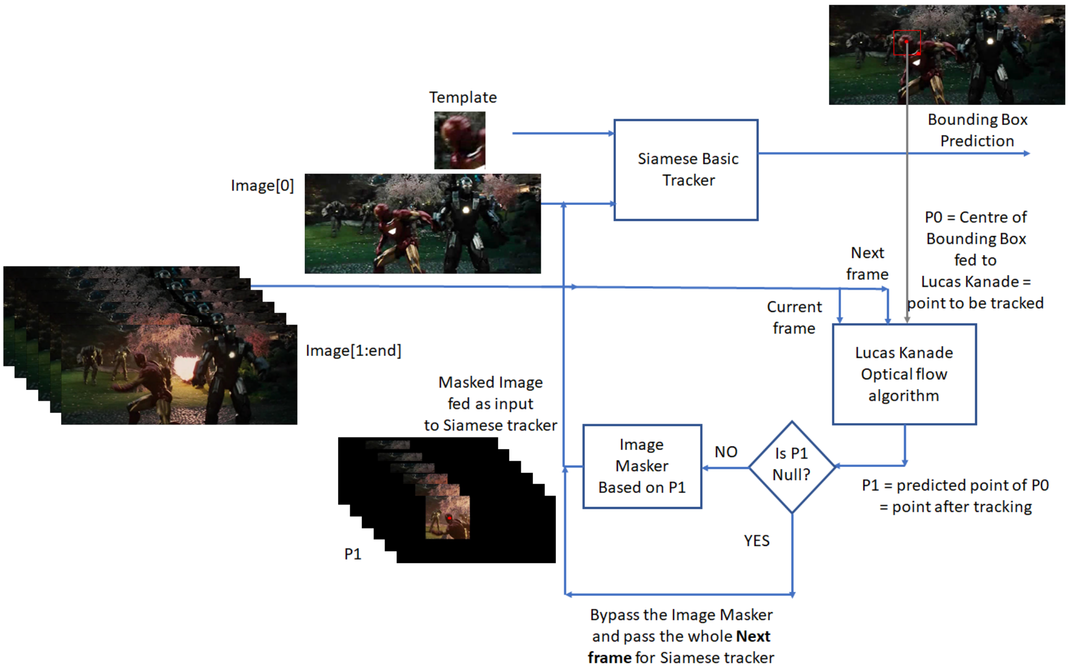
Zhaoyang et al. (2019) [2] addressed the 2D image tracking problem by introducing a modern synthesis of the classic inverse compositional (IC) algorithm for dense image alignment. That is, they proposed a robust version of the IC algorithm wherein they replaced the multiple components of the IC algorithm using more expressive models whose parameters are train in an end-to-end fashion from the data. The authors, conducted experiments on several challenging motion estimation tasks in order to demonstrate the advantages of the algorithm and also depicted how it outperforms the classic IC algorithm and the data-driven image-to-pose regression approaches.

3 Approach

The Siamese Tracker does excellent job with the feature comparison. However, it does not take care of where to search in the whole frame. During this process, it tends to adjust its bounding box across objects that are similar to the first template and misses out the fact that the object might have transformed. One idea to overcome this was to bring in a feedback, and update the template after each prediction. However, there are several problems associated with this approach. One small deviation from the actual object during the tracking process and the algorithm starts tracking the wrong object (due to incorrect template update, for the model which updates the template). However, the idea of bringing in a feedback to control the Siamese tracker is an excellent idea and needed more research. On thorough research of where the Siamese tracker lost its track was when the object underwent small transformation. Hence, there is a need for a tracker which tracks the object based on the difference in the Image frames, and Lucas Kanade's approach works on this idea. However, there are scenarios where the traditional Lucas Kanade's lost track of the object too. The traditional Lucas Kanade's approach does not operate incorporating the high dimensional features and correlation score of the image to track the object.

In each traditional Lucas Kanade's update, the algorithm works on tracking the point it predicted in the previous run. The point it tracks in the current run was not obtained based on the high dimensional feature matching. There is a need to improve the traditional Lucas Kanade's approach by bringing in an architecture which works on matching the high dimensional features of the template and the image. And as we indicated earlier, the Siamese tracker does excellent job in matching the high dimensional features of the template and the image frame.

The idea was to bring in a feedback to Siamese Tracker which could indicate where the probable position of the object could be, based on the Lucas Kanade's approach. With this feedback about the probable location of the object, the Siamese tracker could be forced to specifically look at certain space of the image frame to match the template. In the sub space there is no other object similar to the object itself, though the object has undergone transformation. Additionally, the Siamese tracker could be put in the loop of the Lucas Kanade's algorithm, controlling what point the Lucas-Kanade's algorithm needs to track in its next run. Hence, a closed system was designed incorporating both



Template
Image[0]
Siamese Basic Tracker
Bounding Box Prediction
P0 = Centre of Bounding Box fed to Lucas Kanade = point to be tracked
Next frame
Current frame
Image[1:end]
Lucas Kanade Optical flow algorithm
Masked Image fed as input to Siamese tracker
Image Masker Based on P1
NO
Is P1 Null?
P1 = predicted point of P0 = point after tracking
YES
P1
Bypass the Image Masker and pass the whole **Next frame** for Siamese tracker

Figure 1: Block diagram of the proposed closed system

the ideas, where the Lucas-Kanade's algorithm controls the Siamese tracker by forcing the tracker to search for the template in the subspace of the whole image, while the Siamese tracker controls the Lucas Kanade algorithm, what point it needs to track in the next run based on the correlation of the high dimensional features of the template and the image subspace. In the proposed closed system, the Lucas-Kanade's algorithm controls the Siamese tracker and vice versa. The closed system realized in the course of this project is depicted in the figure 1. To understand the approach and the block diagram better, an example of the video frames from the standard data set [3] can be considered. In the approach, the Siamese tracker matches the template with the first image and predicts the bounding box around the object. The center of the bounding box is given as input (indicated as P0 in the figure 1) to the Lucas-Kanade block to track. The actual current and next image frames are also produced for the Lucas-Kanade's algorithm to calculate the next point. The Lucas Kanade approach predicts the next point (indicated as P1 in the figure 1). Based on the obtained P1, the Siamese tracker needs to restrict its search region. The Image-Masker block in the figure 1 masks the actual next frame image and restricts the Siamese tracker to find a similar template in the non probable positions (that is the masked regions). The masked image samples are as shown in the figure 1.

These masked images are fed to the Siamese tracker which finds the high correlation of the template in the restricted search region. Based on the restricted search region, the next bounding box is predicted. The process is repeated except when the Lucas Kanade fails to track the point obtained from the Siamese tracker. The reason for this failure could be due to the very high contrast change from the current frame to the next frame. In those scenarios, the image masker block is bypassed and it is a fresh start for the Siamese tracker, where the Siamese tracker tracks the closest object to the template, throughout the image frame based on the correlation and high dimensional features.

Category	Description
Background Clutter (BC)	the background near the target has the similar color or texture as the target.
Deformation (DEF)	non-rigid object deformation.
Fast Motion (FM)	the motion of the ground truth is larger than t_m pixels ($t_m = 20$).
In-Plane Rotation (IPR)	the target rotates in the image plane.
Illumination Variation (IV)	the illumination in the target region is significantly changed.
Low Resolution (LR)	the number of pixels inside the ground-truth bounding box is less than t_r ($t_r = 400$).
Motion Blur (MB)	the target region is blurred due to the motion of target or camera.
Occlusion (OCC)	the target is partially or fully occluded.
Out-of-Plane Rotation (OPR)	the target rotates out of the image plane.
Out-of-View (OV)	some portion of the target leaves the view.
Scale Variation (SV)	the ratio of the bounding boxes of the first frame and the current frame is out of the range t_s, $t_s > 1$ ($t_s = 2$).

Table 1: OPE TB100 Categories

The results of the proposed closed system are compiled in section 4. The proposed approach is evaluated based on the AUC scores and plots, IoU plots, Mean IoU score, and a frame rate analysis is performed which can be found in section 4.

4 Results & Quantitative Performance Evaluation

The tracker is evaluated on the OPE TB100 benchmark [3], consisting of 100 videos belonging to categories such as Background Clutter (BC), Deformation (DEF), Fast Motion (FM), In-Plane Rotation (IPR), Illumination Variation (IV), Low Resolution (LR), Motion Blur (MB), Occlusion (OCC), Out-of-Plane Rotation (OPR), Out-of-View (OV), Scale Variation (SV) with descriptions as in Table 1.

We evaluated our tracker titled **SiamLK**, and the baseline **SiamFC** on the TB100 benchmark. We show that our tracker SiamLK operating at *5 scales* achieves comparable to the baseline SiamFC that is operating at *9 scales*.

4.1 AUC Plots

Fig 2 presents the success plots for both trackers across all categories. The AUC score is defined as the area under the success plot curve or as the average of all success rates at different thresholds when the thresholds are evenly distributed.

Category	SiamFC AUC	SiamLK AUC
ALL	**55.59**	46.74
BC	46.11	**46.80**
DEF	42.78	**47.42**
FM	**56.76**	41.15
IPR	48.05	**55.18**
IV	**50.47**	49.69
LR	**55.15**	52.76
MB	**56.97**	38.22
OCC	**49.52**	46.37
OPR	49.16	**53.28**
OV	**48.85**	40.55
SV	**54.99**	48.35

Table 2: AUC Scores (in %) across categories

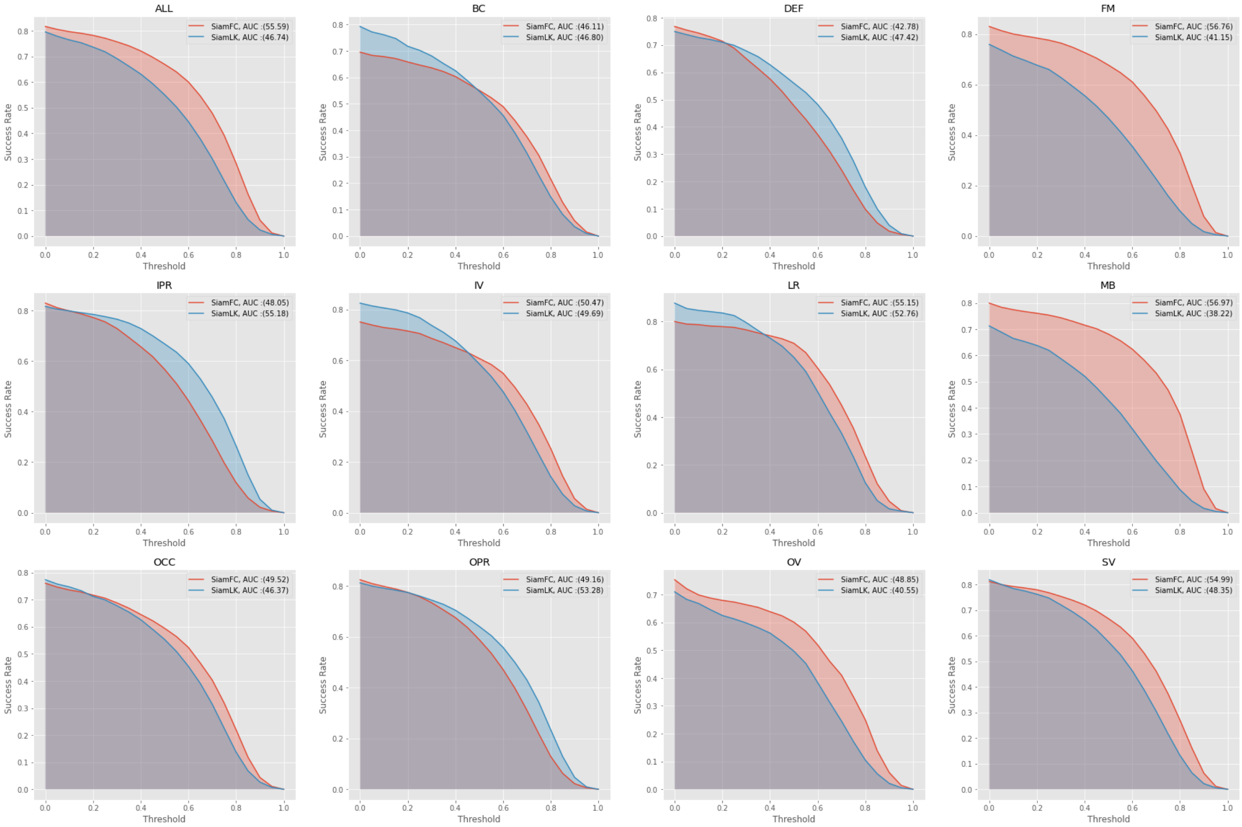


Figure 2: Success Rate Plots and AUC scores for each category

The results are summarized in Table 2 These particular scale values are chosen as they achieve comparable performance in more categories than other choices for scale values.

We see that overall, the SiamFC with 9 scales (SiamFC) edges out our tracker SiamLK with 5 scales (SiamLK) with an overall AUC of 55.59% as opposed to SiamLK's 46.74%.

However, when we analyze by category three main trends emerge:

4.1.1 Performance Improvement

SiamLK achieves better AUC scores than vanilla SiamFC in 4 categories:

- **Background Clutter**: The usage of optic flow uses a motion estimate as a guide to restrict the search region. This reduced search region helps the classifier tune out background clutter better than SiamFC

- **Deformation**: As noted in the principles of Lucas Kanade, the tracker is able to track object deformations better than vanilla SiamFC.

- **In Plane Rotation**: Leveraging optic flow and centering the search region around the object and masking out the rest of the image helps the classifier develop increased sensitivity to in plane rotations

- **Out of Plane Rotations**: Similar to In Plane Rotations, the restricted search region helps the tracker stay on the target even when the rotation is out of the image plane.

4.1.2 Shortcomings

In the following categories, SiamLK performs significantly worse than SiamFC. The following categories see SiamFC outperform SiamLK by at least 5% AUC.

- **Fast Moving Object**: When the object moves very fast, it can escape the search region determined by Lucas-Kanade algorithm for the next frame

- **Motion Blur**: The determination of optic flow for a blurry image is noisy and it may determine an incorrect search region for the next frame

- **Out of View**: When the point for which Lucas-Kanade is being applied leaves the frame, another incorrect point in the image may be incorrectly chosen as the point in the next frame resulting in incorrect Optic Flow calculation.

- **Scale Variation**: On one hand SiamFC operates on 4 more scales than SiamLK. On the other hand, rapid scale changes have similar optic flow to Fast Moving objects, in the perspective of a single point being tracked.

4.1.3 Performance compared to SiamFC

In the remaining categories, SiamLK proves to be competent with SiamFC with SiamFC achieving marginally better performance than SiamLK. However this trade-off is at the cost of lower framerate for SiamFC when compared to SiamLK as outlined in section 4.3

We see that given lower thresholds, SiamLK achieves comparable or higher success rate than SiamFC. SiamFC achieves more successes at higher thresholds.

4.2 IOU Plots

The intersection-over-union (IOU) is defined as the area of the intersection of the predicted bounding box P and the ground truth bounding box GT divided by the union of their areas.

$$IOU(P, GT) = \frac{Ar(P) \cap Ar(GT)}{Ar(P) \cup Ar(GT)}$$

A graphic representation of the same can be seen in Fig. 3

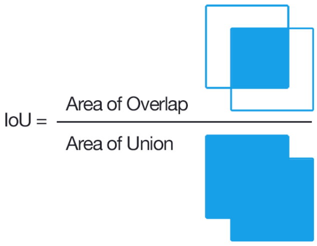


Figure 3: Calculation of IOU

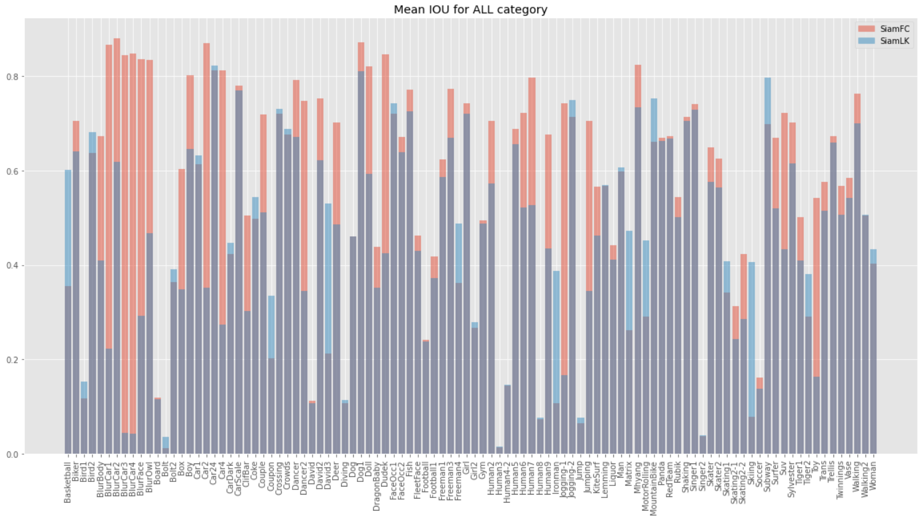
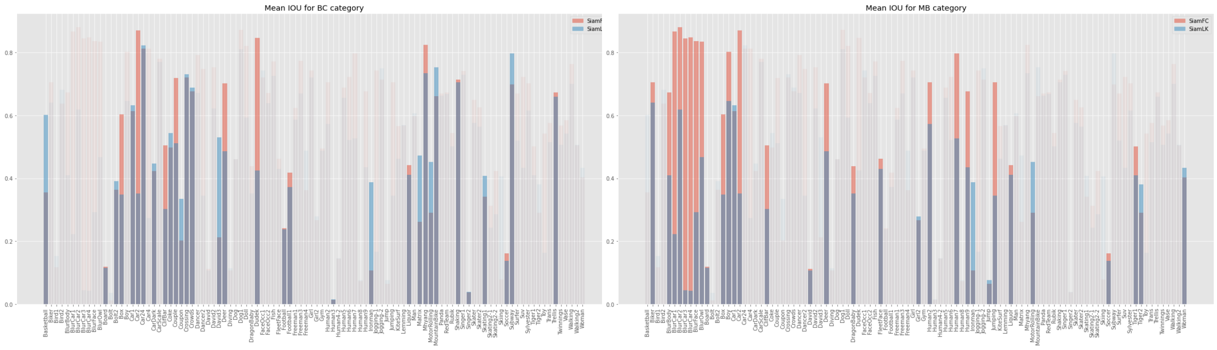


Figure 4: mean IOU for each example

The Mean IOU for both SiamFC and SiamLK are show in Fig. 4. The IOU plots are correlated with the AUC plots. In Fig. 4, we see that SiamFC achieves higher mean IOU than SiamLK across more samples, explaining the higher AUC.

Fig. 5a shows the IOU plots for examples belonging to the BC category, where there are more examples that SiamLK achieves higher IOU across when compared to SiamFC

Fig. 5b highlights those examples belonging to the MB category. As seen in the AUC curves, SiamLK suffers from poor performance whereas SiamFC manages to perform well even if there is motion blur in the video.



(a) Mean IOU for videos in BC Category

(b) Mean IOU for videos in MB Category

Figure 5: IOU plot highlighting categories

SiamFC	SiamLK
0.55	0.46

Table 3: Mean IOU for the tracker across all examples

The mean IOU across the entire OTB100 benchmark for the two trackers is summarised in Table 3. This correlates with the AUC results across all categories in Table 2.

4.3 Frame-rate Analysis

Defining frame-rate as the number of frames processed by the tracker each second, we present frame-reate comparisons for SiamLK-5 and SiamFC-9 in this section.
As seen in Fig 6, we see that SiamLK achieves greater framerate than SiamFC.

Although we compute the optic flow for each frame in our method, we do so in a lightweight fashion. Calculating optic flow for only the center point is faster than operating at 9 zoom levels.

5 Conclusion & Future Work

We proposed a novel regression-based object tracking framework by incorporating Lucas & Kanade algorithm into a Siamese deep learning framework. We conclude that the combination of offline trained feature representation and the efficient online adaptation of regression parameters respect to template images, are crucial advantages of our method to generalize well against real-world situations such as severe scaling, deformation and unseen objects/viewpoints. Compared to SiamFC that only exploits offline trained regression parameters, our SiamLK shows great robustness to unseen objects and viewpoints. Moreover, we can improve by model to performance to current state of the art trackers by considering following aspects

- Since we are considering a bounding box around a single point for the purpose of optic flow, we have a single point of failure. If the point currently being tracked

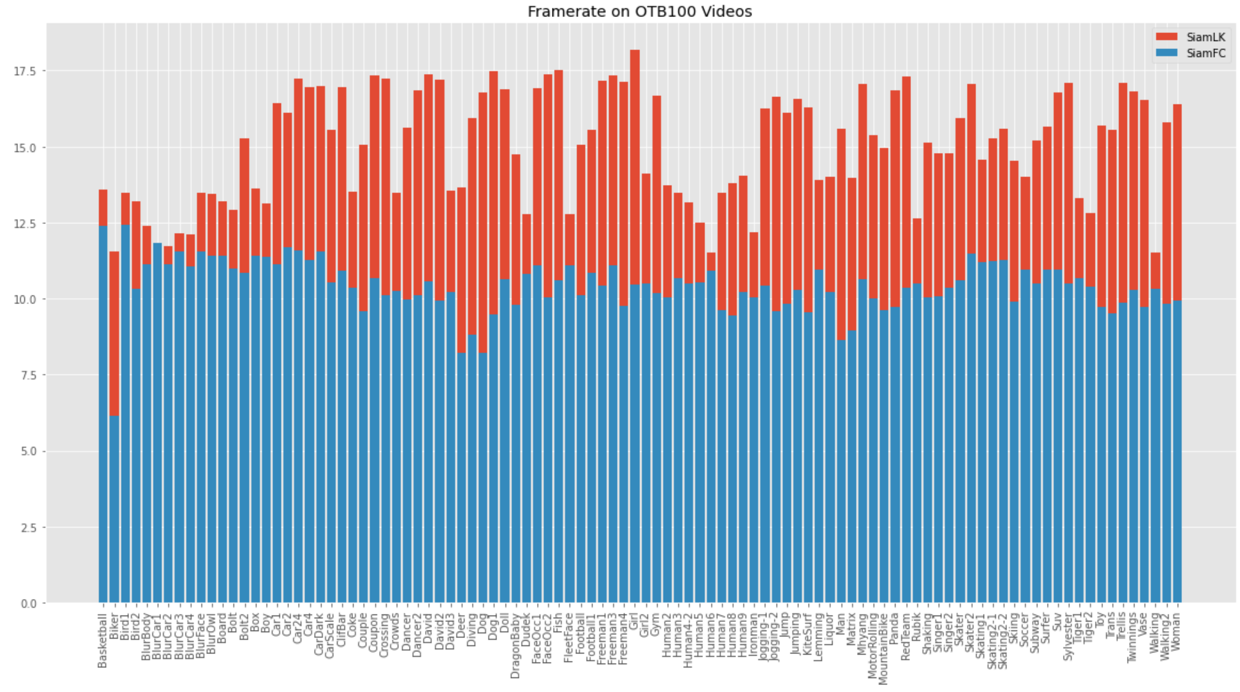


Figure 6: Framerate across examples in OTB 100 benchmark

goes out of the frame, the next point could be affected. We suppose that using Dense Lucas-Kanade may provide better results.

- Using Optical Flow Networks such as FlowNet 2[4] or LiteFlowNet [5], we could use GPU acceleration to compute optic flow faster than Lucas-Kanade Algorithm Integrated with SiamFC is something for us to explore
- Using Optical Flow Networks such as FlowNet 2[4] or LiteFlowNet [5] to tackle the problem of blur, where our method fails.
- Try including the template update with Lucas-Kanade in loop and evaluate if there is any improvement.
- Adaptive masking can be incorporated based on the bounding box information.
- Try including a combination of the next point information (from Lucas-Kanade's previous prediction) along with the centre of the bounding box (from the Siamese tracker) to predict the next point.

6 Code

Our implementation of the current paper can be found at:
https://github.com/Laxmaan/Siam-LK-Tracker

The SiamFC base code is available at:
https://github.com/bilylee/SiamFC-TensorFlow

References

[1] C. Wang, H. K. Galoogahi, C.-H. Lin, and S. Lucey, “Deep-lk for efficient adaptive object tracking,” *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 627–634, 2017. 2

[2] Z. Lv, F. Dellaert, J. M. Rehg, and A. Geiger, “Taking a deeper look at the inverse compositional algorithm,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4581–4590. 3

[3] Y. Wu, J. Lim, and M.-H. Yang, “Object tracking benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015. 4, 5

[4] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470. 10

[5] T.-W. Hui, X. Tang, and C. Change Loy, “Liteflownet: A lightweight convolutional neural network for optical flow estimation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8981–8989. 10