Polytechnique Montreal
Department of Computer Engineering

Lab 3 INF6804 - Winter 2020
Visual object tracking

Daniel Wang & Oumayma Messoussi

Supervisors:

David-Alexandre Beaupre
Soufiane Lamghari

April 2020

Table of Contents

1. Detailed presentation of the methods

In this lab of the computer vision course, we had the opportunity to study 2 methods for single object detection and tracking. In this first chapter of our report, we present the selected methods in detail describing their principles and specifications.

Single object tracking: This task refers to locating an object through the consecutive frames of a video. A bounding box is defined in the first frame to indicate the location of the object of interest. Then, the visual tracking method tracks the object in the next frames, usually outputting the predicted bounding box coordinates at each step.

1.1 MedianFlow

MedianFlow [1] is a visual tracking method proposed in 2010. It is an offline method that tracks points of the object inside its bounding box both forward and backward in time and computes discrepancies between both directions' resulting trajectories.

If the yielded trajectories are same only opposite in direction, the tracking is confirmed to be correct. Otherwise, if there is divergence between the 2 trajectories, the error is then detected. It takes the median of the the tracked points movements in the x-axis and y-axis. This gives MedianFlow the advantage of being very simple and comparable in performance to other trackers.

Pros:

- Smooth tracking when object is visible
- Accurate tracking failure detection
- Very fast and simple to understand and implement

Cons:

- Sensitive to occlusions and fast motion
- Usually not able to detect object again after failure

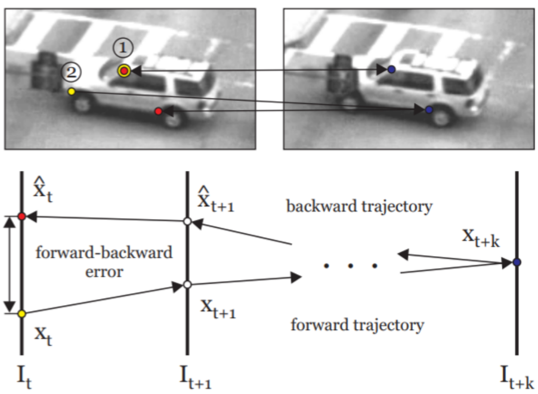

Figure 1.1: MedianFlow algorithm

1.2 SiamFC

SiamFC Networks employ a novel procedure whereby a convolutional neural network undergoes training to resolve a machine learning problem based on similarity. That is to say, the deep neural network architecture is first trained offline, and then evaluated online whilst it performs the task of object tracking. The objective of 'learning' to track arbitrary objects is achieved by engaging in the process of similarity learning. Siamese Fully Convolutional Neural Networks are designed with the aim of learning a multi-variable function f(z,x) that takes as arguments the image of the object being tracked, which we denote as z, and some candidate image x. The function f(z,x) assigns a score to the image pair based on the degree to which the objects in both images appear similar. It is essentially framed as an optimization problem, such that the objective is to maximize the similarity of the candidate image to the tracked image by testing every possible location in which the object may lie. It is a matter of convention to use the first frame as the exemplar image to which candidate images will be compared. The function f is trained in a supervised manner on a video dataset with labels given by object trajectories.

The Siamese architecture is fully convolutional in respect of the target image x. The definition of a fully-convolutional function is stated as a function that is commutative with respect to translation. For instance, a function is fully convolutional if:

$$h(L_{k\tau}x) = L_\tau h(x)$$

where k is the integer stride and L_τ is the translation operation given by $(L_\tau x)[u] = x[u - \tau]$.

Structuring a siamese network as fully convolutional permits the input of a search image of greater size. Upon input, the network computes the similarity for every sub-window translation effected on a dense grid during one evaluation round. To that end, the authors design a convolutional embedding function ψ whose output is then passed onto a cross-correlation layer:

$$f(z, x) = \psi(z) * \psi(x) + b1$$

where b1 represents a signal possessing a different value $b \in \mathbf{R}$ at every sub-window. The final output is a map of scores $D \subset \mathbf{Z}^2$.

Pros:

- rapid speed due to the dual nature of the network (designed to pair two images with high similarity)
- does not require re-training since the model is pre-trained offline

Cons:

- suffers from the generalization problem in deep learning, meaning that model performance is likely to decline when presented with data whose nature is different from that of the training dataset
- cannot adapt to changes in the scale of the object
- unable to handle the task of tracking a single object in clutter or where there are numerous other distracting objects

2. Description of experiments, datasets and evaluation criteria

2.1 VOT2013 dataset

The VOT2013 dataset [2] was proposed during the 2013 version of the Visual Object Tracking challenge [3], with the goal of motivating the research community to advance and compare new short-term single object tracking methods. The dataset presents 16 short video sequences with various challenging conditions such as illumination changes, cluttered backgrounds, fast motion and heavy occlusions.

For the purpose of evaluating our visual object tracking method, we selected the following 3 subsets:

2.1.1 Bicycle sequence

This sequence contains 271 images showing a person riding a bike on a sidewalk. The background varies from buildings to other humans, therefore it's slightly cluttered with many objects of different shapes and colors appear in the background (doors, windows, grass, marks on the road, etc). The person to be tracked on the bicycle is mostly visible in the entire sequence except towards end when it passes behind a pole which causes almost total occlusion. There is also very little paddling action in the last third of the sequence.

2.1.2 David sequence

This sequence of 770 images, in which we want to track the face of a person (we assume he's called David), presents very challenging illumination variations. In the first third of the sequence, the person is in an extremely dark part of a house, then moves gradually towards the light. He also turns sideways at some point and removes his glasses for some frames. However, the face is visible in the entire sequence.

2.1.3 Gymnastics sequence

In this sequence, we find 207 frames that present a gymnast giving a performance. The performance consists of a series of flips and movements involving many rotations and jumps, making the person to track change shape, orientation and speed. The person's motion is very fast.

2.1.4 Juice sequence

This sequence consists of 404 images taken from a scene with a square table on top of which appear a few objects. The goal is to track the rectangular juice box on the table when the camera moves around. The camera's movement is slightly fast in some parts of the video. The challenge is that some of the other objects on the table have similar geometrical shapes as the juice box so the tracker might mistakenly latch onto another object when the camera moves around.

Table 2.1: Summary of conditions per sequence

Sequence	Lighting changes	Fast motion	Occlusion	Rigid	Scale changes
Bicycle	No	∼No	Yes	∼Yes	Yes
David	Yes	No	No	∼Yes	∼No
Gymnastics	No	Yes	No	No	No
Juice	No	∼Yes	No	Yes	∼No

2.2 Evaluation criteria

The 2 main criteria we used to evaluate the performance of our visual trackers include both qualitative and quantitative evaluations.

The main quantitative evaluation metric we used in this lab to evaluate our trackers' performances is the Intersection over Union (IoU). It measures the overlap between the predictions and the ground truth bounding boxes. For each subset, we set a threshold for the IoU score which determines whether a prediction is considered as correct (true positive) or not. The thresholds we used are 30%, 50% and 70%.

We also evaluate our trackers qualitatively by looking at a few image samples to see whether the trackers are correctly predicting the next object locations and whether the bounding boxes are fitting the object well throughout the frames without losing it or latching onto other objects.

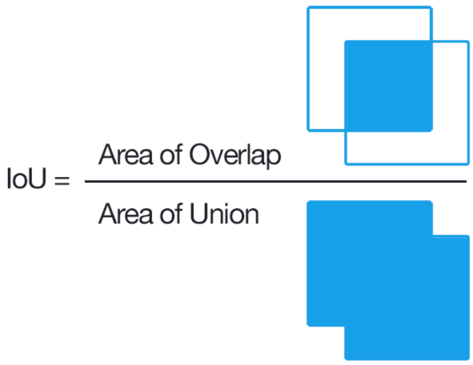

Figure 2.1: IoU evaluation metric

3. Description of the implementations used

3.1 MedianFlow

We start our implementation by creating a Jupyter notebook and importing the predefined functions provided by our TAs.

Then, we define the *track()* function which will run our tracker. It takes as arguments the path to the frames folder and the ground truths in the form a dictionary with keys being the frame indices and the values are tuples (x, y, width, height). The function then instantiates the tracker pre-built in OpenCV [4] by calling the *create* function and initializes it with the first frame and its corresponding ground truth box.

Next, we loop over the video frames, call the *update* function to get the tracker's prediction, store it in a dictionary with the same format as the ground truths described above and then display both boxes on the frame. Finally, the tracks dictionary is returned.

After obtaining all the tracks, we call the *evaluate()* to compute the IoU score between each of the ground truth bounding boxes and the predicted bounding boxes. We obtain an accuracy score which is the number of true positive predictions divided by the total number of ground truth bounding boxes, as well as a robustness value representing the average IoU by the number of true positives.

Algorithm 1: Single-object tracking algorithm

```
Input: Frames path, ground truth boxes
Output: Dictionary of tracks
Initialize tracker with first frame ground truth box
for frame i do
    Read frames
    Update tracker
    if update successful then
        Add predicted box to the tracks dictionary
        Display predicted box
    else
        Print error message
    end
    Display ground truth box
end
```

3.2 SiamFC Networks

We implement SiamFC networks by cloning the github repository found in [5]. Due to limitations in disk space, we have elected to use a pretrained model prepared by the inventor (downloaded from ftp://ftp.robots.ox.ac.uk/pub/outgoing/tvg/cfnet/cfnet-networks.zip). The network is trained based on a discriminative approach with the objective of minimizing a loss function.

$$l(y, v) = log(1 + exp(-yv))$$

Here, v denotes the score assigned to an image pair comprised of the exemplar image and the candidate search image, and $y \in \{1, -1\}$ is the ground truth value. Since the siamese network is fully convolutional in nature, the network will produce a map of scores $v : D \to R$. The loss of a score map is then best expressed as the mean of the set of losses:

$$L(y, v) = \frac{1}{|D|} \sum_{u \in D} l(y[u], v[u])$$

where $u \in D$ denotes the position of the sub-window in the image.
The parameters θ for the convolutional neural network are determined by way of resolving the optimization problem through stochastic gradient descent:

$$\arg\max_{\theta} \underset{z,x,y}{E} L(y, f(z, x; \theta))$$

Lastly, the value of the ground truth $y \in \{-1. + 1\}$ is determined by the sub-window's proximity to the centre of the image with radius R.

$$\text{y[u]=} \begin{cases} +1 & if k||u - c|| \leq R \\ -1 & otherwise \end{cases}$$

4. Experimentation results for validation tests

4.1 MedianFlow

Bicycle sequence results

Table 4.1: Experimental results for Bicycle sequence

IoU 30%		IoU 50%		IoU 70%	
Accuracy	**Robustness**	**Accuracy**	**Robustness**	**Accuracy**	**Robustness**
0.63838	0.72715	0.61254	0.74037	0.41328	0.80065

Figure 4.1: Tracking results example from Bicycle sequence

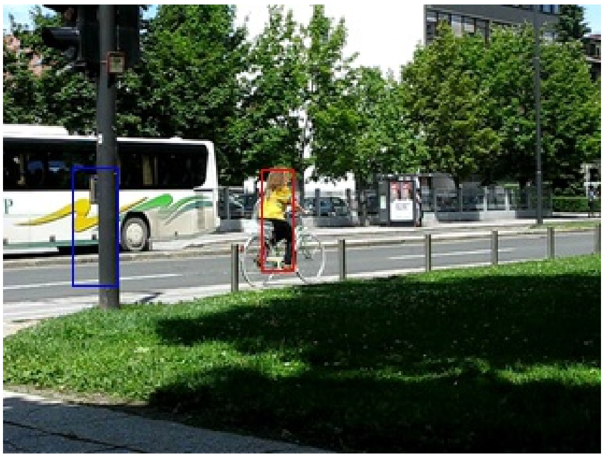

Figure 4.2: Tracking results example from Bicycle sequence

David sequence results

Table 4.2: Experimental results for David sequence

IoU 30%		IoU 50%		IoU 70%	
Accuracy	**Robustness**	**Accuracy**	**Robustness**	**Accuracy**	**Robustness**
1.0	0.64674	0.87532	0.67322	0.32468	0.76945

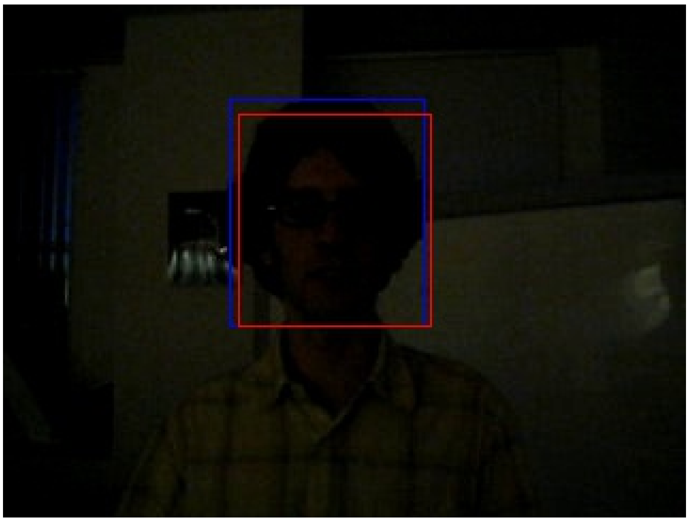

Figure 4.3: Tracking results example from David sequence

Figure 4.4: Tracking results example from David sequence

Gymnastics sequence results

Table 4.3: Experimental results for Gymnastics sequence

IoU 30%		IoU 50%		IoU 70%	
Accuracy	**Robustness**	**Accuracy**	**Robustness**	**Accuracy**	**Robustness**
0.42512	0.78600	0.39614	0.81545	0.33333	0.85194

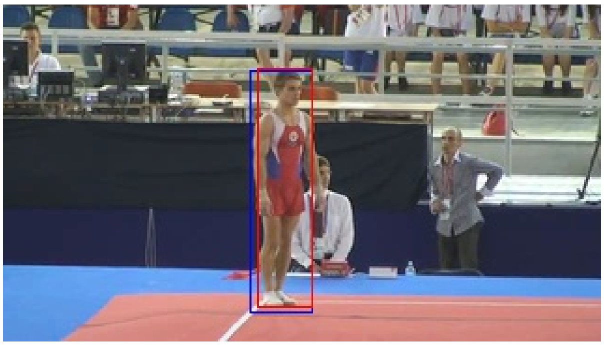

Figure 4.5: Tracking results example from Gymnastics sequence

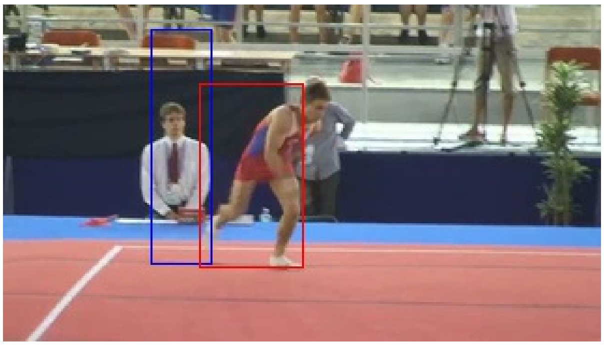

Figure 4.6: Tracking results example from Gymnastics sequence

Figure 4.7: Tracking results example from Gymnastics sequence

Juice sequence results

Table 4.4: Experimental results for Juice sequence

IoU 30%		IoU 50%		IoU 70%	
Accuracy	**Robustness**	**Accuracy**	**Robustness**	**Accuracy**	**Robustness**
1.0	0.89085	1.0	0.89085	1.0	0.89085

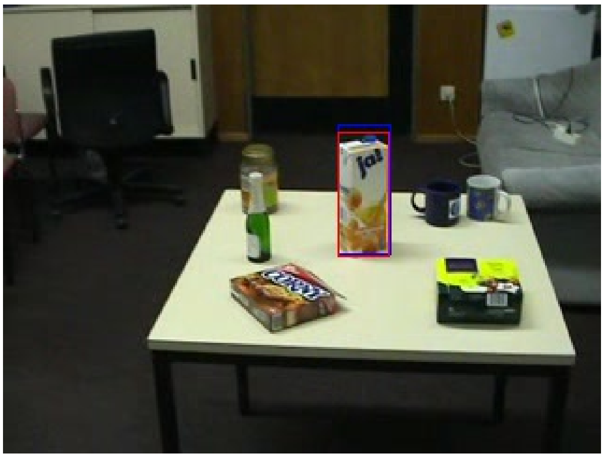

Figure 4.8: Tracking results example from Juice sequence

4.2 SiamFC Networks

Bicycle sequence results

Table 4.5: Experimental results for Bicycle Sequence

IoU 30%		IoU 50%		IoU 70%	
Accuracy	**Robustness**	**Accuracy**	**Robustness**	**Accuracy**	**Robustness**
1.0	40.52	1.0	43.01	1.0	46.09

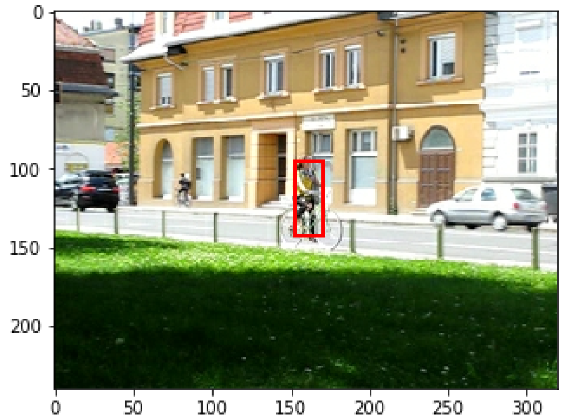

Figure 4.9: Tracking results example from Bicycle sequence

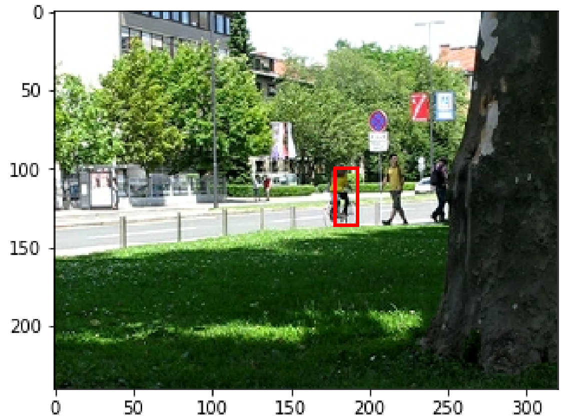

Figure 4.10: Tracking results example from Bicycle sequence

David sequence results

Table 4.6: Experimental results for David sequence

IoU 30%		IoU 50%		IoU 70%	
Accuracy	**Robustness**	**Accuracy**	**Robustness**	**Accuracy**	**Robustness**
99.87	34.65	1.0	69.53	1.0	35.66

Figure 4.11: Tracking results example from David sequence

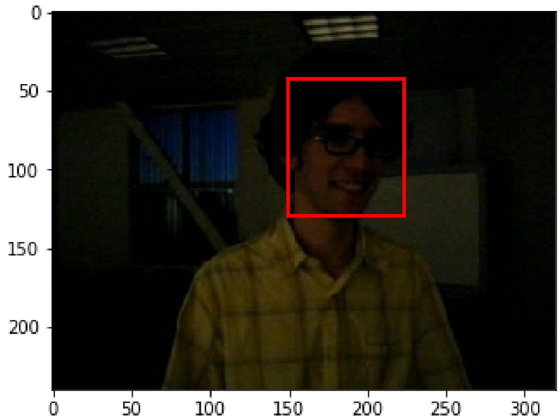

Figure 4.12: Tracking results example from David sequence

Gymnastics sequence results

Table 4.7: Gymnastics Sequence Results

IoU 30%		IoU 50%		IoU 70%	
Accuracy	**Robustness**	**Accuracy**	**Robustness**	**Accuracy**	**Robustness**
25.71	8.93	71.01	21.49	83.57	28.40

Juice sequence results

Table 4.8: Experimental results for Juice sequence

IoU 30%		IoU 50%		IoU 70%	
Accuracy	**Robustness**	**Accuracy**	**Robustness**	**Accuracy**	**Robustness**
1.0	42.55	1.0	43.61	1.0	46.47

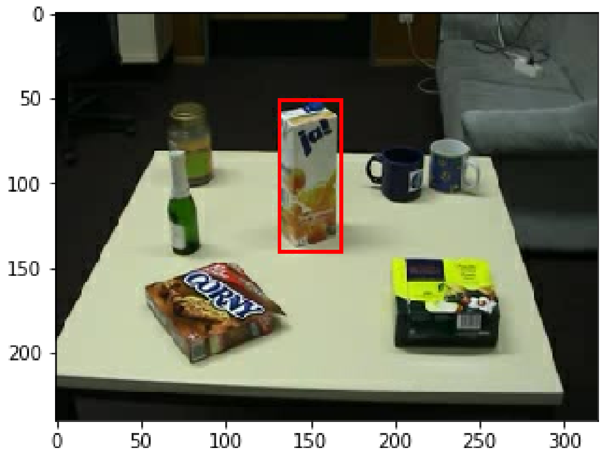

Figure 4.13: Tracking results example from Juice sequence

5. Discussion of results

5.1 MedianFlow

Bicycle sequence

MedianFlow performed fairly well on this sequence, with a 63.84% accuracy at IoU 30%. It managed to smoothly and consistently track the woman riding the bike in the first two thirds of the sequence since there was little camera jitter and the bike's motion is relatively constant. The bounding boxes fit the object of interest quite well.

However, in the last third of the sequence when the object's scale gets smaller and it passes behind a pole, the latter significantly occludes the bike and, as expected, the tracker fails to correctly follow the object and latches onto the pole and a region of the background bus that displays a yellow sign similar to the color of the woman's shirt. This was one of the mentioned cons of MedianFlow in chapter 1.

David sequence

The performance on the David sequence is very good, scoring and accuracies of 100% and 87.53% at IoU 30% and 50%. The tracker follows the person's face smoothly throughout the entire sequence, even in frames where the person turns sideways or removes his glasses or changes facial expressions, although the predicted bounding boxes are often not as fitted as the ground truth boxes.

The tracker was also very robust to lighting changes. It managed to correctly localize the face when the person moved from a darker to a lighter part of the house since the face was visible in all frames.

Gymnastics sequence

As expected, MedianFlow yielded poor results on the Gymnastics sequence mainly due to the extremely fast motion of the gymnast. The only correct bounding boxes are the ones resulting from nearly the first 70 frames where the person is still standing. But once he begins the performance, the various movements are extremely fast. Thus, the tracker fails, due to weaknesses with fast motion and tracking non-rigid shapes (the gymnast bends and flips a lot in the video).

Juice sequence

MedianFlow gave near perfect performance on this sequence. Even though the scene contained other objects with very similar geometrical shapes as the object of interest, and even when the camera moved slightly fast in some parts of the video, the tracker successfully tracked the juice box in all frames, with resulting bounding boxes fitting the object extremely well compared to the ground truth boxes.

It scored perfect accuracy with very strong robustness score at the 3 IoU reference levels. We even tested at IoU 80% in our notebook and the accuracy was 98.27% with a robustness of 89.28%. This method seems, thus, very robust for tracking rigid bodies.

5.2 SiamFC Networks

Bicycle sequence

SiamFC networks demonstrates impeccable results in light of the perfect accuracy scores achieved. The simple task of object tracking is an easy feat for SiamFC networks in regular conditions, but we have yet to see its performance in other environments.

David sequence

Performance on the David sequence is nearly perfect, signalling that SiamFC networks is impervious to variations in lighting.

Gymnastics sequence

In this case, performance of SiamFC networks begins to deteriorate as the target object undergoes changes in form, which in this scenario are the spontaneous transformations in human body pose. The empirical evidence show in this experiment affirms SiamFC net's weaknesses in tracking dynamic objects that transform with respect to shape and orientation. The test on the gymnastics sequence might be amenable to improvement by first training the model with data augmentation, whereby the training dataset is adjoined with samples of the original dataset modified by alterations in scale and orientation. This process was designed to ameliorate performance by familiarizing the model with variations in the data so that it may better perform its task on 'anomalous' samples of test data.

Juice sequence

SiamFC Networks performed with perfect accuracy on the juice dataset. The reason is most probably due to the idleness of the target object and its decent spacing from surrounding objects, which prevents the occasion of distractions that might confuse the tracker.

It scored perfect accuracy with very strong robustness score at the 3 IoU reference levels. We even tested at IoU 80% in our notebook and the accuracy was 98.27% with a robustness of 89.28%. This method seems, thus, very robust for tracking rigid bodies.

Bibliography

[1] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In *In Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR '10*, pages 2756–2759. IEEE Computer Society, 2010. 1

[2] VOT2013 benchmark. 3

[3] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, Nov 2016. 3

[4] OpenCV. 5

[5] SiamFC implementation. 6