# End-to-End Object Tracking Under Motion Blur

Li Yu[1], Shrey Nigam[2], Shivansh Rao[1], and Vikas Kumar[2]

[1]College of Information Science and Technology, Penn State University, PA, USA
{luy133, shivanshrao}@psu.edu
[2]Electrical Engineering and Computer Science Department, Penn State University, PA, USA
{sqn5300, vuk160}@psu.edu

## Abstract

*Given a bounding box across an arbitrary object in the first frame, the goal of visual object tracking is to learn how to track that object in the entire video. A common approach for tackling this problem is to find the similarity of a small candidate image (template) with a large image (search) at all the sub-windows of the search image in a single evaluation. Although there have been extensive work in the literature on visual object tracking, most of them deal with simple scenarios having no occlusion, illumination or appearance change and no motion blur. In this report we consider the motion blur problem in visual tracking. Specifically, a Generative Adversarial Network [4] is trained jointly with SiamRPN++ [18] to eliminate the blurred video frames, and perform efficient tracking. Experimental results on the Need For Speed dataset [12] shows that our method outperforms the original SiamRPN++ [18], and thus proves the efficiency of motion blur removal on the tracker's performance.*

## 1. Introduction

Visual Object Tracking has a very crucial role in computer vision research community, and has many real-time applications in human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging and video editing [23, 22, 11]. Despite having extensive work and decades of research, it is still a challenging task because of various challenging factors in the real-world videos such as occlusion, illumination, background clutter. Various efforts [8, 30, 6, 31] have been made to overcome the above mentioned issues, however motion blur still remains a common problem with most of the state-of-the-art trackers. Blur is usually caused either by the slow speed of the camera (see Figure 1) or the fast motion of the target. Ultimately, motion blur reduces the tracker's performance by degrading the visual features extracted by the tracker.


Figure 1. Samlpes of blurred-images from the Need For Speed dataset [12], along with their deblurred pairs generated using Deblur-GAN V2 [16].

Object tracking under motion blur is a challenging problem because: (a) blur leads to less visual information from search/template image, (b) intensity of blur can range from low-level to drastic blur, (c) motion blur also causes a few other issues like abrupt motion. A common solution to this is to first deblurr the input frames and then performing tracking. In [2] the authors deal with the motion blur by using deformation parameters and motion vectors to compute matching score and thus matching the blurred region. Whereas, in [10], a mean-shift tracker is used for tracking of motion-blurred target. Both the above meth-

All the authors have contributed equally. The code for reproducing the results can be found at *http://34.82.77.125:5000/tree/CV*

ods assume that the blurred target has been roughly segmented, whereas in our work we aim to automatically track the blurred region. In image processing, edge priors [17], image-restoration [25], removing camera-shake [3] have been used to deal with the motion-blur problem. However, these methods usually contain strong ringing artifacts, which generate "fake" features and further make the tracking process more complex.

In this report, we present a novel end-to-end object tracking under motion blur network. Our network is based on Deblur-GAN V2 [16] in the first stage for deblurring the input frames, which is jointly trained with SiamRPN++ [18] to perform tracking. Our method thus aims to first deblur the input frames and then use the deblurred frames to perform tracking. We evaluate our method on NFS dataset [12] which has both the fast-motion data subset and normal motion data subset. We find that performing deblurring in such a manner boosts the performance of the tracker.

The rest of the report is organized as follows. We discuss the related work in Section 2. Our approach is discussed in Section 3. We present our evaluation in Section 4. Conclusion is presented in Section 5.

## 2. Related Work

Motion blur is a known problem in object tracking. Previous work implicitly used motion blur as additional information for performance improvement [33, 21, 34, 32, 27]. However, these methods assumed that the performance of object tracker would be dependent on the deblurring component (which can introduce negative effects like noise or low resolution in the frames) that can worsen the performance of videos that have minimal motion blur. However, recent deblurring methods [16, 15] have proved to be better and faster with minimal adverse effects to the output frames compared to previously tested traditional methods. Additionally, these methods use traditional trackers and do not discuss how motion blur affect state-of-the-art trackers like SiamFC [1], SiamRPN [19] and SiamRPN++ [18]. The problem of motion-blur does exist in these trackers [29], but has not been addressed by the existing literature.

Several datasets were considered for our task. While VOT [13] and OTB [33] datasets contain noticeable motion blur, we required significant blurring in each video to track performance improvements using the proposed method. Additionally, fabricating motion blur in these datasets resulted in unrealistic results due to their low FPS (frames per second). On the contrary NFS [12] dataset contains videos that were captured with 240 FPS. Qing Gou et. al. [5] created the dataset for motion blur by taking average of the frames of videos captured at 240 fps to reduce them to videos of 30 fps. Replicating the same experiment, the output frames contained improbable motion blurs. To avoid this problem, we created motion blur using the method suggested in [16], where the videos were first interpolated to 3840 fps and an average pooling was performed over the same time window which resulted in smoother and continuous blurs.

DeblurGAN-v2 [16] is a deblurring conditional Generative Adversarial Network (cGAN) trained on the GoPro [24] and NFS [12] dataset. Building upon the success of DeblurGAN, the authors extend their work by modifying the architecture to include Feature Pyramid Network in the generator which makes it 10-100 times faster than its competitors. We use DeblurGAN-v2 to perform deblurring while maintaining real-time tracking speed. The architecture of DeblurGAN-v2 is shown in Figure 3. While DeblurGAN has already been used before for visual tracking [5], we test the results of better and faster version of the deblurring component for our work.

## 3. Our Approach

In this section, we first discuss our One-stage network, followed by the model learning and dataset explanation. At the end we discuss our setup and implementation details.

### 3.1. One-Stage Network

We propose to combine Deblur-GAN V2 with SiamRPN++ into a one-stage network and train them jointly, so that parameters of both models get refined to achieve more coherent performance. The architecture of our proposed network is shown in Figure 2. The one-stage network is built on SiamRPN++ with an additional module of deblurring. That is, similar to the network architecture of [18], our one-stage network has one Siamese network and one region proposal network (RPN), of which the Siamese network has two branches: template branch and search branch. The input of template branch is the object cropped from the first frame of a video which is usually clear. There is no deblurring planned for this branch. On the search branch, deblurring is performed before the search image is fed into the following feature extraction convolutional neural network (CNN) of Siamese. Since we only need the deblurring functionality of Deblur-GAN V2, the generator instead of the entire GAN structure is applied on the search image to yield a deblurred, clear image. The next steps are

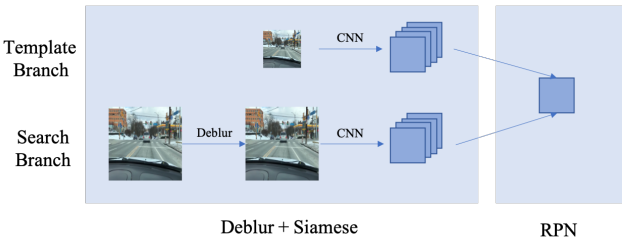


Figure 2. Architecture of our proposed joint network.

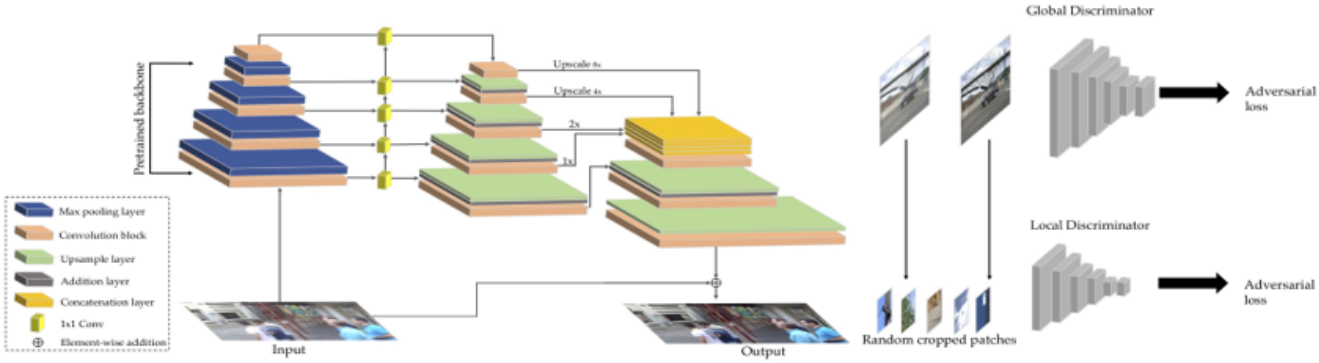


Figure 3. Architecture of DeblurGAN-v2 (excerpt from [16])

the same to SiamRPN++ where both template and search images are passed through a CNN for feature extraction. The RPN is then followed to compute cross-correlation and finally output the detection result.

### 3.2. Model Learning

We finetune the proposed network on the NFS dataset at a low learning rate, because both the Deblur-GAN V2 and SiamRPN++ have been well trained on their own datasets and have released their pretrained weights. We train the two networks jointly on the new dataset to yield a coherent one-stage network.

### 3.3. Dataset

We train and test our tracker on the benchmark dataset: 'Need for Speed' [12]. The dataset constitutes 100 videos with 380K frames. It was captured with the high frame rate of 240 FPS. 85 videos are used for training while 15 videos are used for testing. We create two versions of the dataset: a) the frames are averaged to reduce the frame rate to 30 fps (Version 1) as suggested in [5] and, b) the videos are interpolated to 3840 fps and average pooling is performed on the same time window (Version 2). Each version of dataset has two different subsets: a) sharp video subset (240 fps for both versions) and, b) blurred video subset (30 fps dataset for Version 1 and 240 fps for Version 2). To match the frame rate of the blurred video dataset, we only use every $8^{th}$ frame in the sharp video dataset of Version 1. The images of Version 2 has realistic motion blur (see Figure 4 for more details).

### 3.4. Setup and Implementation Details

We choose FPN-Inception as the backbone of Deblur-GAN V2, which is one of the backbones shipped with the model (the other is FPN-MobileNet). Specifically, Feature Pyramid Network (FPN) [20] is introduced in the generator of Deblur-GAN architecture, which is initially developed

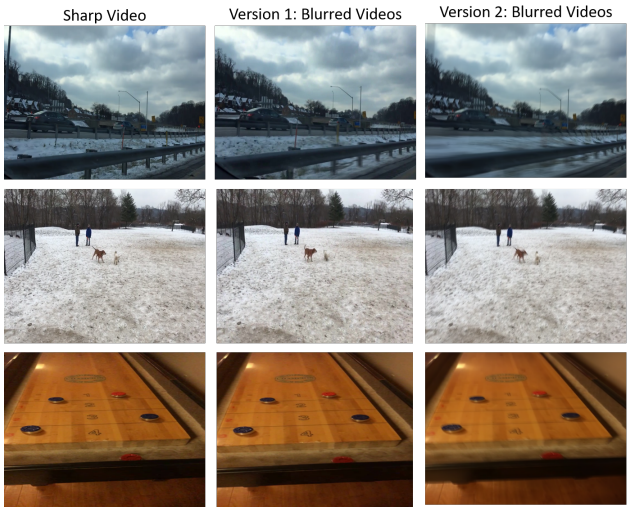


Figure 4. Sharp video and corresponding blurred videos from two versions of the dataset.

for object detection. Either Inception [28] or MobileNet [9] is available as backbone of the generator, with the later aimed at efficiency. For the backbone of SiamRPN++, we choose ResNet50 [7] out of a few choices provided by the authors of the paper. The other backbones include AlexNet [14] and MobileNetV2 [26].

We notice that the output of Deblur-GAN V2 has a dimension of 256 $\times$ 256, whereas the input dimension of the search branch of Siamese network of SiamRPN++ is 255 $\times$ 255. To accommodate the mismatch in dimension, we downsample the output image of deblur module to fit in the input of tracking module.

The learning rate of Deblur-GAN V2 module is set to be 0.001 and that of SiamRPN++ is set to be 0.01 (originally it was 0.1) during joint training. All the other hyperparameters are used as they are in provided source code. As mentioned in model learning subsection, we only finetune

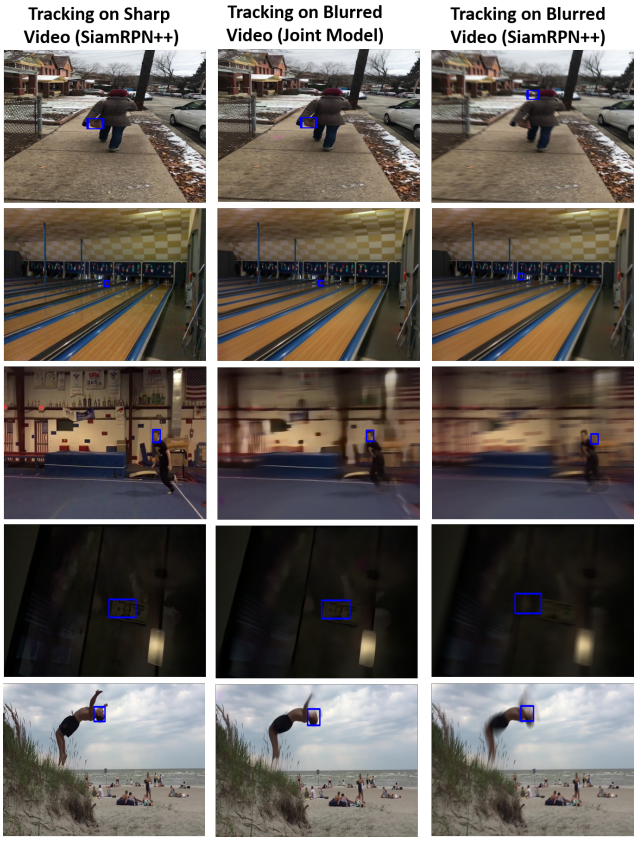


Figure 5. Qualitative results using Joint Model

the joint network so a total number of 2 epochs are trained on the NFS dataset.

## 4. Results

In this section, we first discuss the qualitative results, and then discuss the quantitative results between our proposed network and original SiamRPN++ tracker.

### 4.1. Qualitative Results

In Figure 5 we show the qualitative results of our proposed joint model over 5 different categories. The first column represents the tracking results of SiamRPN++ on sharp video, which can be viewed as the ground truth. The second column and the third column compare the performance of our proposed joint model and original SiamRPN++ respectively on the blurred videos. It is clear from the position of blue bounding boxes that our joint model correctly locates the objects (over all the 5 categories) whereas the original SiamRPN++ fails to track at cases of motion blur.

### 4.2. Quantitative Resutls

Table 2 and Table 3 shows that be deblurring the video the overall precision and success rate increases when compared to the blurred videos. This consistency is shown over both versions of the datatset. Additionally in Table 1 we show category wise results on the second version of the dataset. It is clear from Table 1 that for all the categories except Zebra Fish, Yoyo, and Walking3 the results after removing motion blur are better. We feel that for the above mentioned categories lack in improvement was because the videos did not contain much of motion blur. Additionally success plot is shown in Figure 6 which proves the fact that Sharp videos are the best for tracker's performance, followed by Deblurred video, and Blurred videos are the worst.

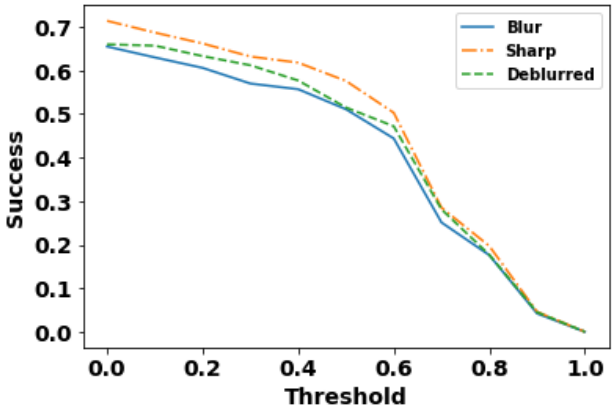


Figure 6. Success Plot

| | Blurred Video | | Deblurred Video | | Sharp Video | |
|---|---|---|---|---|---|---|
| **Method** | **Precision** | **Success** | **Precision** | **Success** | **Precision** | **Success** |
| Purse | 0.114 | 0.098 | 0.715 | 0.617 | 0.763 | 0.653 |
| Bunny | 0.483 | 0.626 | 0.532 | 0.652 | 0.576 | 0.661 |
| Skiing | 0.401 | 0.285 | 0.443 | 0.364 | 0.465 | 0.392 |
| Basketball2 | 0.546 | 0.379 | 0.568 | 0.466 | 0.934 | 0.739 |
| Bottle | 0.306 | 0.579 | 0.335 | 0.601 | 0.325 | 0.592 |
| Dog2 | 0.522 | 0.387 | 0.679 | 0.481 | 0.578 | 0.439 |
| Iceskating6 | 0.620 | 0.677 | 0.909 | 0.719 | 0.914 | 0.743 |
| Running100m2 | 0.523 | 0.538 | 0.595 | 0.646 | 0.570 | 0.673 |
| Zebra Fish | 0.441 | 0.630 | 0.431 | 0.610 | 0.410 | 0.598 |
| Yoyo | 0.523 | 0.296 | 0.513 | 0.922 | 0.466 | 0.240 |
| Walking3 | 0.154 | 0.622 | 0.158 | 0.620 | 0.147 | 0.614 |

Table 1. Quantitative Results using Joint Model on 2nd version of dataset. (Category-Wise)

| Blurred Video | | Deblurred Video | | Sharp Video | |
|---|---|---|---|---|---|
| **Precision** | **Success** | **Precision** | **Success** | **Precision** | **Success** |
| 0.593 | 0.504 | 0.608 | 0.514 | 0.623 | 0.523 |

Table 2. Quantitative Results using Joint Model on 1st version of dataset. (Overall)

| Blurred Video | | Deblurred Video | | Sharp Video | |
|---|---|---|---|---|---|
| **Precision** | **Success** | **Precision** | **Success** | **Precision** | **Success** |
| 0.434 | 0.402 | 0.476 | 0.426 | 0.513 | 0.461 |

Table 3. Quantitative Results using Joint Model on 2nd version of dataset. (Overall)

## 5. Conclusion and Future Work

In this report, we have proposed a method for tracking objects under motion blur. Our network performs motion deblurring in an end-to-end fashion: first stage being deblurring the blurred frames using state-of-the-art DeblurGAN V2 [16], second stage being the tracker [18] trained on the deblurred images. Our experiments show that this approach of motion deblurring can improve the overall tracker's performance. In the future, we would like to add occlusion removal, and multi-scale feature learning information to our network to make an attempt towards designing a robust tracker which not only overcomes motion blur but also overcomes the other challenges in visual object tracking.

## 6. Acknowledgment

The work was done while all the authors were taking the *"Topics in Computer Vision - CSE 586"* course at Pennsylvania State University. The authors acknowledge Dr. Robert Collins for his active suggestions during the project. The authors thank Google Cloud Platform for donating some of the GPUs used in this work.

## References

[1] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 2

[2] S. Dai, M. Yang, Y. Wu, and A. K. Katsaggelos. Tracking motion-blurred targets in video. In *2006 International Conference on Image Processing*, pages 2389–2392. IEEE, 2006. 1

[3] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH 2006 Papers*, pages 787–794. 2006. 2

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[5] Q. Guo, W. Feng, Z. Chen, R. Gao, L. Wan, and S. Wang. Effects of blur and deblurring to visual object tracking. *arXiv preprint arXiv:1908.07904*, 2019. 2, 3

[6] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2096–2109, 2015. 1

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[8] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2427–2434, 2013. 1

[9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[10] H. Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 18–25. IEEE, 2005. 1

[11] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pages 85–94. IEEE, 1999. 1

[12] H. Kiani Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1125–1134, 2017. 1, 2, 3

[13] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2137–2155, 2016. 2

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[15] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018. 2

[16] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8878–8887, 2019. 1, 2, 3, 5

[17] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70–es, 2007. 2

[18] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019. 1, 2, 5

[19] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018. 2

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3

[21] B. Ma, L. Huang, J. Shen, L. Shao, M.-H. Yang, and F. Porikli. Visual tracking under motion blur. *IEEE Transactions on Image Processing*, 25(12):5867–5876, 2016. 2

[22] L. Mihaylova, P. Brasnett, N. Canagarajah, and D. Bull. Object tracking by particle filtering techniques in video sequences. *Advances and challenges in multisensor data and information processing*, 8:260–268, 2007. 1

[23] P. Mountney, D. Stoyanov, and G.-Z. Yang. Three-dimensional tissue deformation recovery and tracking. *IEEE Signal Processing Magazine*, 27(4):14–24, 2010. 1

[24] H. Ovrén and P.-E. Forssén. Gyroscope-based video stabilisation with auto-calibration. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2090–2097. IEEE, 2015. 2

[25] W. H. Richardson. Bayesian-based iterative method of image restoration. *JoSA*, 62(1):55–59, 1972. 2

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 3

[27] C. Seibold, A. Hilsmann, and P. Eisert. Model-based motion blur estimation for the improvement of motion tracking. *Computer Vision and Image Understanding*, 160:45–56, 2017. 2

[28] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 3

[29] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1328–1338, 2019. 2

[30] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *2011 International Conference on Computer Vision*, pages 1323–1330. IEEE, 2011. 1

[31] W. Wang, J. Shen, X. Li, and F. Porikli. Robust video object cosegmentation. *IEEE Transactions on Image Processing*, 24(10):3137–3148, 2015. 1

[32] Y. Wu, J. Hu, F. Li, E. Cheng, J. Yu, and H. Ling. Kernel-based motion-blurred target tracking. In *International Symposium on Visual Computing*, pages 486–495. Springer, 2011. 2

[33] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013. 2

[34] L. Xu, H. Luo, B. Hui, and Z. Chang. Real-time robust tracking for motion blur and fast motion via correlation filters. *Sensors*, 16(9):1443, 2016. 2