Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Graphics and Algorithms
TU Wien
Favoritenstr. 9/186-3
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801 - 18661
Fax: +43 (1) 58801 - 18697
E-mail: anna_g@prip.tuwien.ac.at
URL: http://www.prip.tuwien.ac.at/

PRIP-TR-148 May 13, 2019

The effect of motion blur on the ManakinTracker [1]

Anna Gostler

Abstract

The ManakinTracker was developed to track a small tropical bird, called golden-collared manakin, which performs its courtship dance at a very high speed. The fast movement leads to strong motion blur. The ManakinTracker is based on a Convolutional Neural Network (CNN) as well as blob detection through background subtraction based on a Mixture of Gaussians model. The CNN was trained on images cropped from frames in a data set of videos depicting the golden-collared manakin displaying its courtship dance. In our experiments, we pre-processed (simulated motion blur, rotated, deblurred) the videos to assess how motion blur affects the performance of the ManakinTracker. We found that when we simulated motion blur, less motion blur lead to an increased robustness of the ManakinTracker.

[1]Supported by Walter Kropatch

1 Introduction

The ManakinTracker [6] is a visual tracker based on a Convolutional Neural Network (CNN) and blob detection through background subtraction with a Mixture of Gaussians model [15]. The ManakinTracker was developed to track a small tropical bird called golden-collared manakin (Manacus vitellinus) in high-speed videos recorded by a group of biologists. The tracker receives a bounding box enclosing the bird in the first frame of a video and outputs bounding boxes for the bird for all consecutive frames.

The videos show the birds displaying their elaborate courtship dance. During the courtship dance the male bird moves between saplings by jumping, producing loud wing snaps mid-flight [5]. Biologists are interested in how the male bird has to execute the dance to lead to mating success. They will use the output of the ManakinTracker as a basis for analyzing the videos. To gain a better understanding of how motion blur affects the performance of the ManakinTracker, we run the tracker on pre-processed frames (see section 4) as well as the original frames and compare the results.

2 What is motion blur?

Motion blur is an artifact where objects in an image appear streak-like because the camera integrates all positions of the object during exposure time. These streaks are characterized by their length and angle. Motion blur is caused by the relative movement of the camera and objects in the recorded scene. In our case, the camera is static and the motion blur results from the moving objects in the scene. Figure 2 shows cases where the ManakinTracker fails due to strong motion blur.

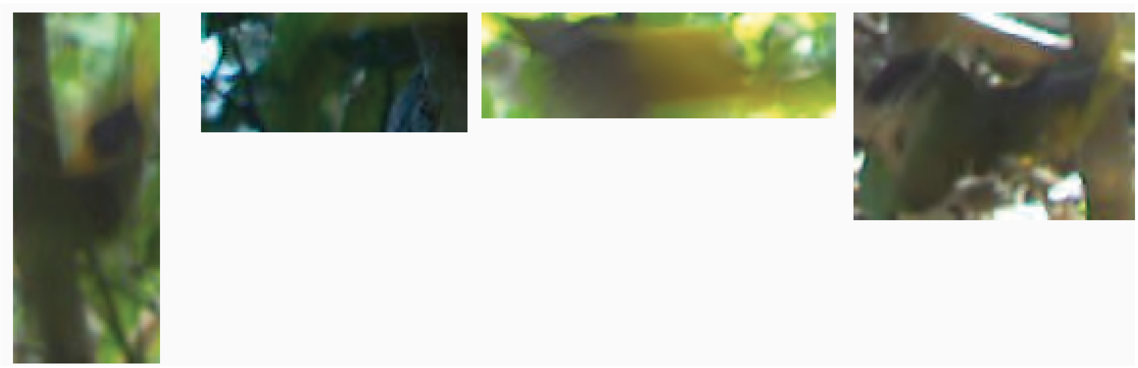


Figure 1: Images of birds affected by very strong motion blur. The ManakinTracker failed to recognize these as male birds.

3 Related Work

Non-blind image deblurring methods, such as Lucy-Richardson algorithm [3, 12] and Wiener Filter [1], require a known or estimated blurring function. The results achieved by these methods depends on the quality of the estimation [17]. Recently, a range of blind deblurring methods based on deep learning have been presented, which are trained with a set of blurred and corresponding non-blurred images. Gupta et al. [7] use coupled autoencoder for deblurring. Sun et al. [16] deblur images using a convolutional neural network and a Markov random field model to estimate non-uniform motion blur. Schuler et al. [13] predict the blur kernel for deblurring an image with a deep neural network, which was trained on a set of sharp images taken from ImageNet as well as these same images with artificial blur added. Chakrabarti [4] trained a neural network to perform blind motion deblurring by predicting the complex Fourier coefficients of a deconvolution filter. Nah et al. [10] put together a data set of blurred and non-blurred images that were not generated artificially but using regular and high-speed cameras. They trained a multi-scale convolutional neural network on that data set to deblur images.

Traditional local feature descriptors, which detect corners or gradients, are ineffective when dealing with blurred images [18]. To address this problem, Mustaniemi et al. [9] first deblur images to increase the robustness of such feature descriptors. However, their method relies on a Gyroscope to measure motion blur. Instead of deblurring images before applying a traditional local feature descriptor, Tong et al. [18] introduce a local feature descriptor that is invariant to blur to match blurred and non-blurred images. This method is designed to handle sudden starting and stopping of a tracked object. A strong change to the appearance of the object other than blur (e.g. opening the wings, turning) would cause image matching with feature descriptor to fail.

A different approach to deal with motion blur is to use motion blur as a clue to detect moving objects. Pang et al. [11] classify pixels as blurred or non-blurred in order to segment motion-blurred objects in videos. Shishido et al. [14] use the shape of a motion blurred region to predict the trajectory of a fast moving object. Xu et al. [8] developed a tracker that addresses the problem of tracking fast moving objects based on a Kernelized Correlation Filter. They estimate the motion state of the target with a point sharpness function, which might be able to handle the sudden starting and stopping of the bird. Wang et al. [19] present an algorithm to effectively remove motion blur. They estimate blur angle and the blur length by identifying the point spread function in the frequency domain.

4 Experiments

We performed experiments on 49 video sequences of golden collared manakins' courtship displays (with a total of 9430 frames that have a ground truth bounding box) with the following settings. Figure 4 shows a frame taken from our data set processed with all these settings.

- **none:** As a baseline, we ran the tracker on the original frames. During a jump, the bird moves 31.5 pixels per frame on average.

- **green circle:** A green circle distorted by varying degrees of motion blur superimposed on each video. The green circle was placed inside the bounding box of the bird, we used the *MATLAB* function *fspecial('motion', len, theta)* [2] (len is the length of the motion blur in pixels, theta is the angle of the motion blur in degrees) to get the filter kernel for simulating linear motion blur. Length was set to 0 (no distortion), 5, 10, 15 and 20. We calculated the angle based on the movement of the bird from the current frame to the next one.

- **mock bird:** The same as *green circle* but instead of one green circle, we used three circles (a black, a yellow and a green one) placed on top of each other to mimic the birds appearance more closely. For this setting we used motion blur lengths of 0 (no distortion), 5, 10, 15, 20, 25 and 30.

- **rotate90:** Each frame is rotated by 90 degrees to assess the effect on the performance of the tracker when the bird is being distorted by motion blur in a different direction than in the videos used for training.

- **lucy:** The region of the frame inside the ground truth bounding box is deblurred with the Lucy-Richardson algorithm [3, 12] using the angle and length based on the movement of the bird from the current frame to the next (vector between bounding box centers).

5 Results

This section presents the results of the experiments outlined in section 4.

We measure the **accuracy** of the ManakinTracker's predictions by calculating the Intersection over Union (IOU) of the predicted bounding boxes and the ground truth bounding boxes. Figure 3 shows that the highest IOU averaged over all video sequences is achieved with setting *mock bird*. The average IOU increases with increasing motion blur length when setting *mock bird* was used until length=20. However, it decreases again at length=25.

The number of restarts during tracking measures the ManakinTracker's **robustness**. If the bounding box predicted by the ManakinTracker does not overlap with the ground truth bounding box, the tracker is restarted using the ground truth bounding box. Figure 4 shows that the percentage of restarts is lowest (0.2%) for settings *mock bird* with length=0, 5, and 10, but gradually increases – reaching 3.34% at length=30. The highest percentage of restarts (19%) occur with setting *green circle* at length=10.

The ManakinTracker classifies blobs found trough background subtraction as well as regions cropped around the bird's current location with a Convolutional Neural Network (CNN). If the CNN assigns those blobs or regions a **confidence score** above a certain threshold, they are used to find the bounding box for the current frame. Since blobs and regions with confidence scores below the threshold are ignored, we averaged over all frames only the maximum confidence score assigned per frame. Figure 5 shows the average maximum confidence score (abbreviated as AMCS in the following) per setting. Settings *mock bird*, *none*, *rotate90* and *lucy* all reached an AMCS of over 0.95. The highest AMCS of 0.999 was reached with setting *mock bird* at length 0 and 5. Increased motion blur led to lower AMCS scores for setting *mock bird*: at length=30 a value of 0.97 was reached. Setting *green circle* achieved the lowest AMCS overall with a value of 0.71 for motion blur length=0 and even lower values for lengths=5, 10, 15, and 20.

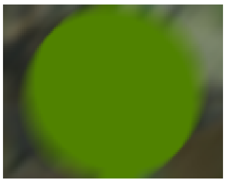
The ManakinTracker uses different **modules** to determine the bounding boxes during tracking: using blobs obtained through background subtraction [15], keeping

(a) original frame

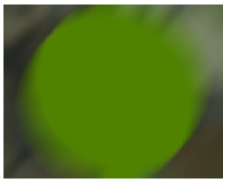
(b) green circle length = 0

(c) green circle length = 5

(d) green circle length = 10



(e) green circle length = 15



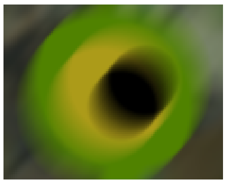
(f) green circle length = 20

(g) mock bird length = 0

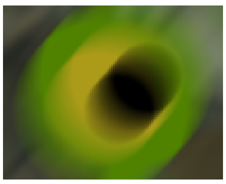
(h) mock bird length = 5

(i) mock bird length = 10

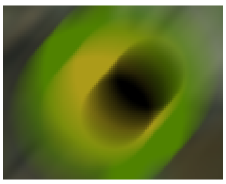
(j) mock bird length = 15



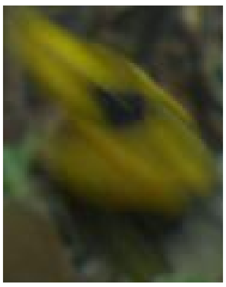
(k) mock bird length = 20



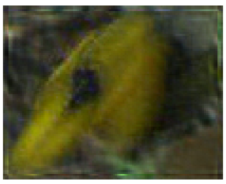
(l) mock bird length = 25



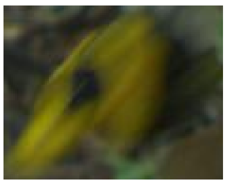
(m) mock bird length = 30



(n) rotate



(o) lucy



(p) none

Figure 2: Example images for all settings. In this frame the bird's speed is 63 pixels per frame. (a) shows the full original frame. (b)–(p) show the same frame pre-processed with different setting, cropped with the ground truth bounding box.

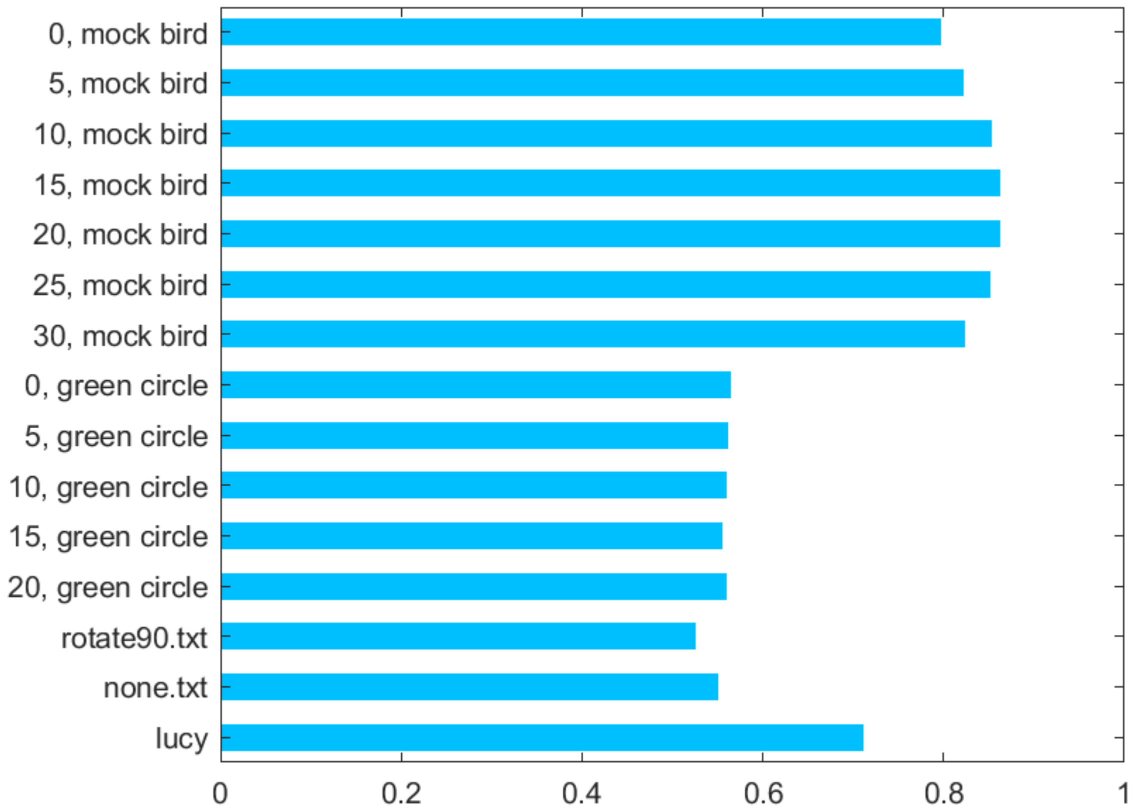


Figure 3: Average IOU per setting.

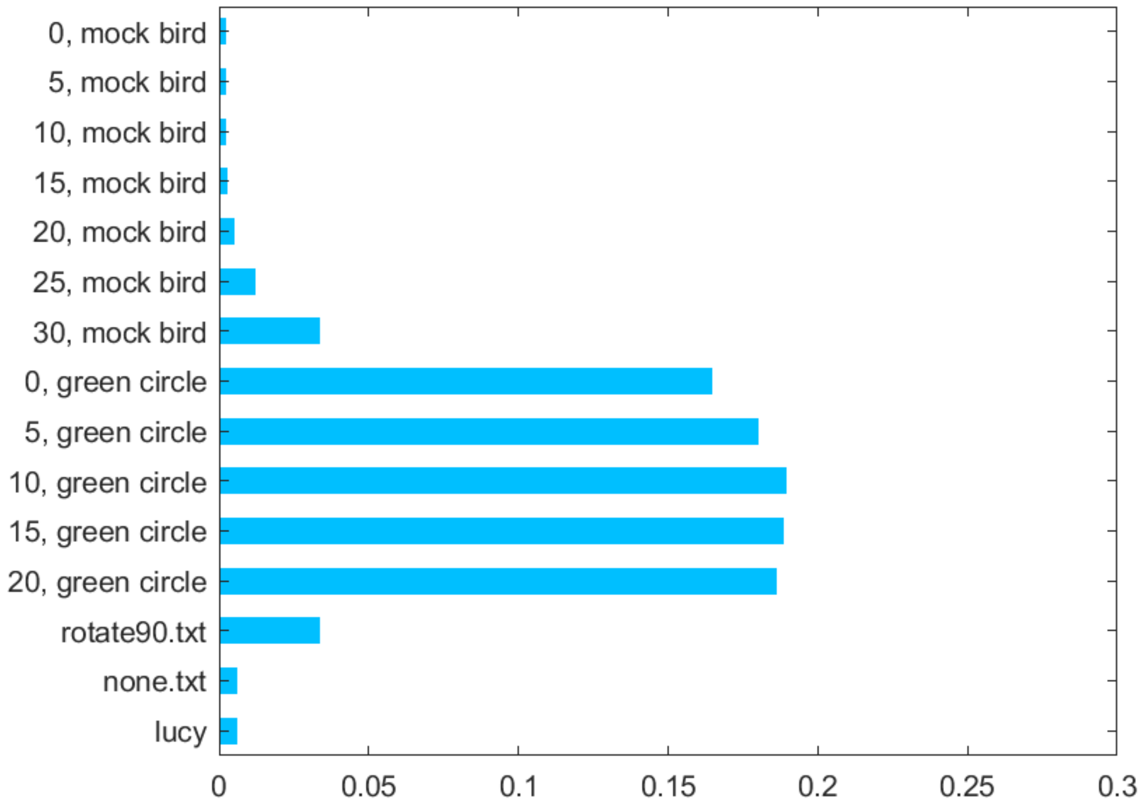


Figure 4: Percentage of restarts per frame per setting.

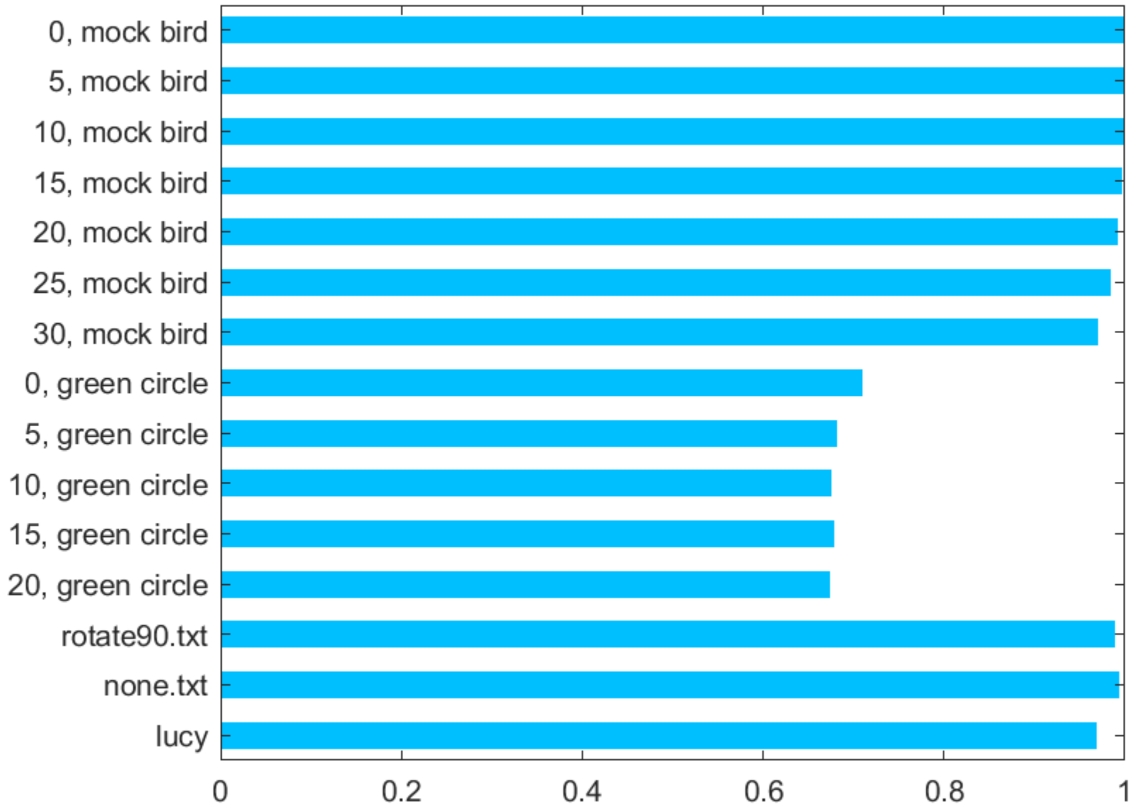


Figure 5: Average of maximum confidence score assigned by CNN to blobs or regions found in the frames per setting.

the bird's previous position, using the position predicted by a Kalman filter[20], searching the bird in regions cropped around the bird's previous position or using the ground truth (when the tracker is initialized or re-initialized after a restart).

Figure 6 shows the percentage of frames for which each module was used to determine the bounding boxes. Blobs are used as the current bounding box if they receive a CNN confidence score above a certain threshold. With setting *green circle* blobs where used in less than 0.33% of frames. For all other settings, blobs where the most commonly used module. Keeping the previous position (which is done if the correlation between the region inside the bounding box of the previous frame and the region inside that same bounding box in the current frame is above a certain threshold) is much more common for settings *lucy* (30%) and mock bird (23-26%) than for settings *rotate90*(1.6%) and *none* (0.8%).

6 Result Analysis

In this section we evaluate how the ManakinTracker's performance is influenced by motion blur.

The highest accuracy is achieved by setting *mock bird* and peaks at length=20. On the other hand, robustness with setting *mock bird* decreases with increasing motion blur.

We suspect that the increased accuracy when using a certain amount of motion blur might be caused by the shape being elongated by the motion blur which leads to a blob that more closely matches the ground truth bounding boxes that where

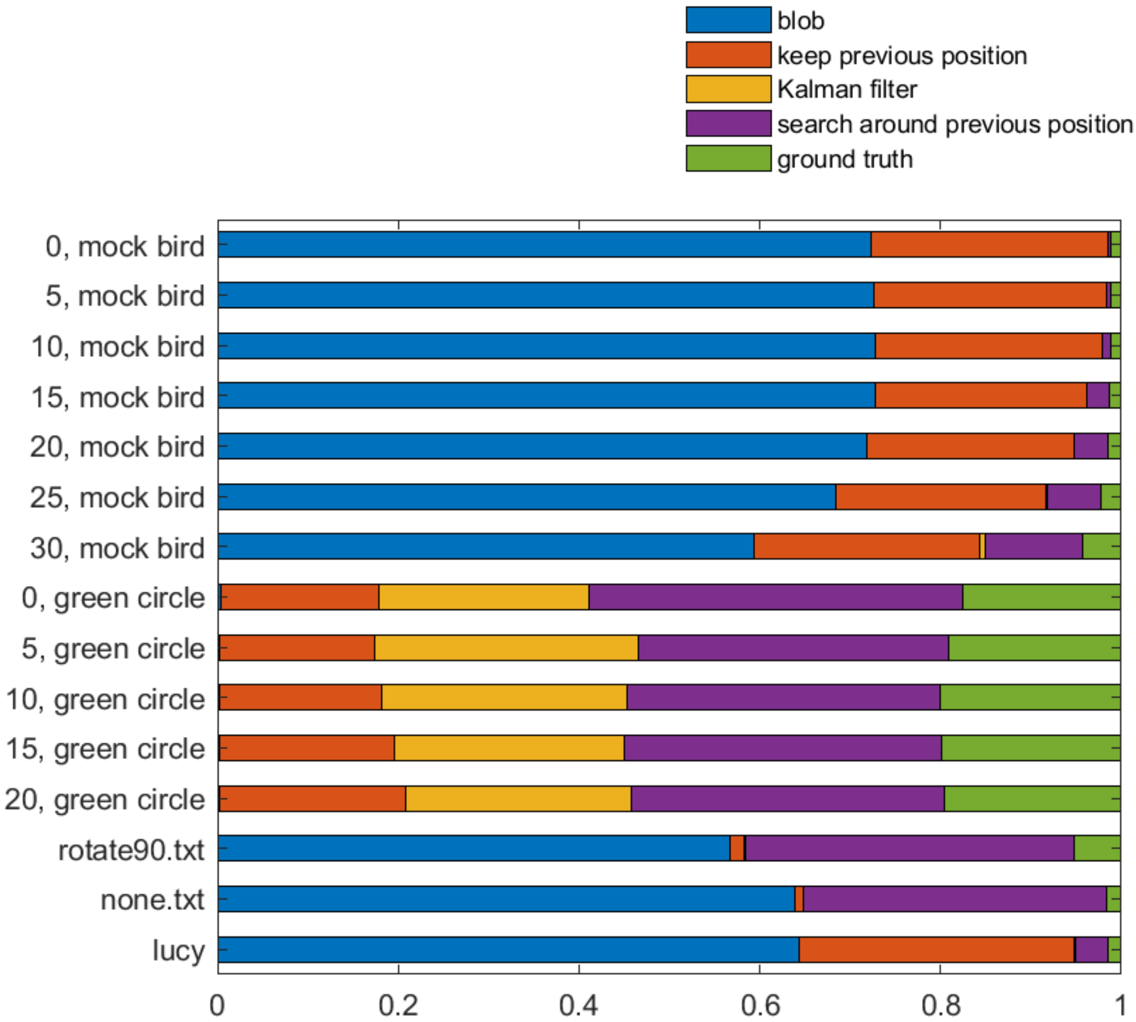


Figure 6: Percentage of each module of the ManakinTracker used to predict the bounding boxes per setting.

annotated in such a way as to enclose the motion blurred birds in the videos. Additionally, when using the settings *green circle* and *mock bird*, circles are superimposed onto the video. As a consequence (partial) occlusion is not an issue in contrast to the unprocessed, rotated and partially deblurred frames (settings *none*, *rotated90* and *lucy*). The lack of (partial) occlusion most likely contributes to higher accuracy and more robustness. Without occlusion, the bird can be detected as a blob and identified by the CNN; without partial occlusion blobs or regions do not have to be combined to form the final bounding box.

The CNN assigned the lowest confidence scores when setting *green circle* was used. The scores further decreased when motion blur was added. This indicates that the green circle does not fit well what the CNN learned as the bird's appearance.

With setting *mock bird* the CNN assigned the highest confidence scores (0.999) which suggests that the "male bird" re-created with three superimposed circles closely fits the CNN's learned model of the male bird's appearance. Additional motion blur decreases the assigned CNN confidence scores. This implies that the CNN does not rely on motion blur to identify the bird.

On the other hand, deblurring with setting *lucy* lead to lower confidence scores than on the unprocessed frames. This could be because the parameters estimated from the bird's movement did not accurately reflect the correct motion blur length and angle that caused the motion blur. Additionally, Gaussian blur was neglected for deblurring. Thus, the artifacts introduced by deblurring might change the appearance of the male bird so that it was less recognizable to the CNN.

Rotating the frames with setting *rotate90* slightly decreased the CNN's confidence scores compared to no pre-processing. We suspect that rotation changed the bird's appearance (jumping downwards instead of sideways) which made it less recognizable. However, that the scores are reduced only slightly shows the CNN's robustness to such changes.

The option of keeping the previous position as the current bounding box is used far less often with settings *none* and *rotate90* in comparison to the other settings. This option is supposed to be used when the bird is sitting and is achieved through calculating the correlation between the current and previous frame inside the previous frame's bounding box. We suspect that the simulated birds (settings *green circle* and *mock bird*) appear much more similar in consecutive frames which leads to higher correlation. With setting *lucy* this higher correlation might be caused by enhanced contrast achieved through deblurring.

7 Conclusion

We performed experiments to evaluate how motion blur influences the effectiveness of the ManakinTracker. We have hypothesized that the CNN might rely on motion blur to identify the bird. This could pose a problem when a camera with higher frame rate is used, which would decrease motion blur.

We simulated the bird by superimposing three colored circles and found instead that increasing motion blur lowered the confidence of the CNN and lead to more tracking failure.

On the other hand, deblurring the frames with the Lucy-Richardson algorithm lead to lower confidence scores from the CNN compared to the original frames. We suspect that this is caused by artifacts introduced by deblurring that make the bird less recognizable to the CNN.

References

[1] Deblurring images using a wiener filter. https://de.mathworks.com/help/images/deblurring-images-using-a-wiener-filter.html. Accessed: 2019-03-29.

[2] fspecial, create predefined 2-d filter. https://www.mathworks.com/help/images/ref/fspecial.html. Accessed: 2019-04-17.

[3] L. B. Lucy. An iterative technique for the rectification of observed distributions. *The Astronomical Journal*, 79:745, 05 1974.

[4] A. Chakrabarti. A neural approach to blind motion deblurring. *CoRR*, abs/1603.04771, 2016.

[5] M. J. Fuxjager, L. Fusani, F. Goller, L. Trost, A. T. Maat, M. Gahr, I. Chiver, R. M. Ligon, J. Chew, and B. A. Schlinger. Neuromuscular mechanisms of an elaborate wing display in the golden-collared manakin (manacus vitellinus). *Journal of Experimental Biology*, 2017.

[6] A. Gostler. Tracking golden-collared manakins in the wild. Technical Report PRIP-TR-141, PRIP, TU Wien, 2018.

[7] K. Gupta, B. Bhowmick, and A. Majumdar. Motion blur removal via coupled autoencoder. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 480–484, Sep. 2017.

[8] X. Lingyun, H. Luo, H. Bin, and Z. Chang. Real-time robust tracking for motion blur and fast motion via correlation filters. *Sensors*, 16:1443, 09 2016.

[9] J. Mustaniemi, J. Kannala, S. Särkkä, J. Matas, and J. Heikkilä. Fast motion deblurring for feature detection and matching using inertial measurements. *CoRR*, abs/1805.08542, 2018.

[10] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. *CoRR*, abs/1612.02177, 2016.

[11] Y. Pang, H. Zhu, X. Li, and J. Pan. Motion blur detection with an indicator function for surveillance machines. *IEEE Transactions on Industrial Electronics*, 63(9):5592–5601, Sep. 2016.

[12] H. W. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62:55–59, 01 1972.

[13] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Learning to deblur. *CoRR*, abs/1406.7444, 2014.

[14] H. Shishido, I. Kitahara, Y. Kameda, and Y. Ohta. A trajectory estimation method for badminton shuttlecock utilizing motion blur. pages 325–336, 10 2013.

[15] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking, 02 1999.

[16] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. *CoRR*, abs/1503.00593, 2015.

[17] S. Tiwari, V. P. Shukla, A. Singh, and S. Biradar. Review of motion blur estimation techniques. *Journal of Image and Graphics*, 1:176–184, 01 2014.

[18] Q. Tong and T. Aoki. A novel and blur-invariant local feature image matching. In *2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, pages 59–60, June 2017.

[19] Z. Wang, Z. Yao, and Q. Wang. Improved scheme of estimating motion blur parameters for image restoration. *Digital Signal Processing*, 65(C):11–18, 2017.

[20] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, Chapel Hill, NC, USA, 1995.