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-141 February 22, 2018

# Tracking Golden-Collared Manakins in the Wild[1]

*Anna Gostler*

**Abstract**

Male golden-collared manakins are tropical birds that perform an elaborate courtship display which determines their mating success. Biologists recorded the birds' displays in the jungle with high-speed cameras. To analyze what constitutes a good courtship performance the biologists use the bird's trajectory, which they currently obtain by manually annotating the videos frame by frame. Automatically tracking the bird can save a lot of time. The videos of the courtship displays are challenging for a tracker: the bird is susceptible to motion blur, quickly changes its appearance and often leaves the frame. The cluttered background contains elements that visually resemble or occlude the bird. We present an online visual tracking algorithm, which combines a Mixture of Gaussians model to detect moving objects, a Convolutional Neural Network trained to recognize the male golden-collared manakin, and a Kalman Filter as a motion model. Our tracker achieves better accuracy and robustness on a dataset of videos of courtship displays than state-of-the-art trackers.

[1]Supported by Nicole M. Artner

# 1 Introduction – Problem Statement

The golden-collared manakin (Manacus vitellinus) is a small tropical bird that lives in the Panama forest. The males perform elaborate, acrobatic displays to court mates [4]. During its courtship dance the male jumps between saplings, which make up his court, producing loud wing snaps mid-flight. Mating success seems to be related to superior motor skills [1] which allow a male bird to execute his dance faster and more precisely – however it is not fully clear yet how exactly the courtship dance has to be performed to impress a female.

To gain more knowledge about their dance, biologists recorded the birds in the wild with high-speed cameras. Manually annotating every frame in a video is a tedious process. So, our goal is to develop a tracker that generates bounding box annotations automatically.

The following properties of the videos make tracking the birds challenging:
**bird's speed:** While jumping, the bird changes its position very quickly.
**motion blur:** Strong motion blur can make the bird hard to recognize, as it loses most of its local features.
**bird's size and shape change:** The bird's size and shape changes when it opens or closes its wings, turns, or moves towards or away from the camera.
**occlusion:** The bird is often partly occluded by the sapling it sits on between jumps. The bird can also be occluded by leaves or trees when it jumps or sits.
**bird out of frame:** The bird often leaves the camera's field of view during the dance.
**bird's trajectory:** The bird's trajectory is characterized by abrupt stopping and starting, as well as direction changes (after landing the bird typically jumps in a different direction).
**background color:** The forest is colored mostly green, yellow and brown – similar to the golden-collared manakin male, which has a green body, black head and a yellow neck, and the female, which is well-camouflaged in the forest due to its green body.
**background motion:** There can be moving leaves and branches in the background. The saplings often move when the bird lands on them.

Tracking is made easier, however, by the **absence of camera motion**.

# 2 Related work

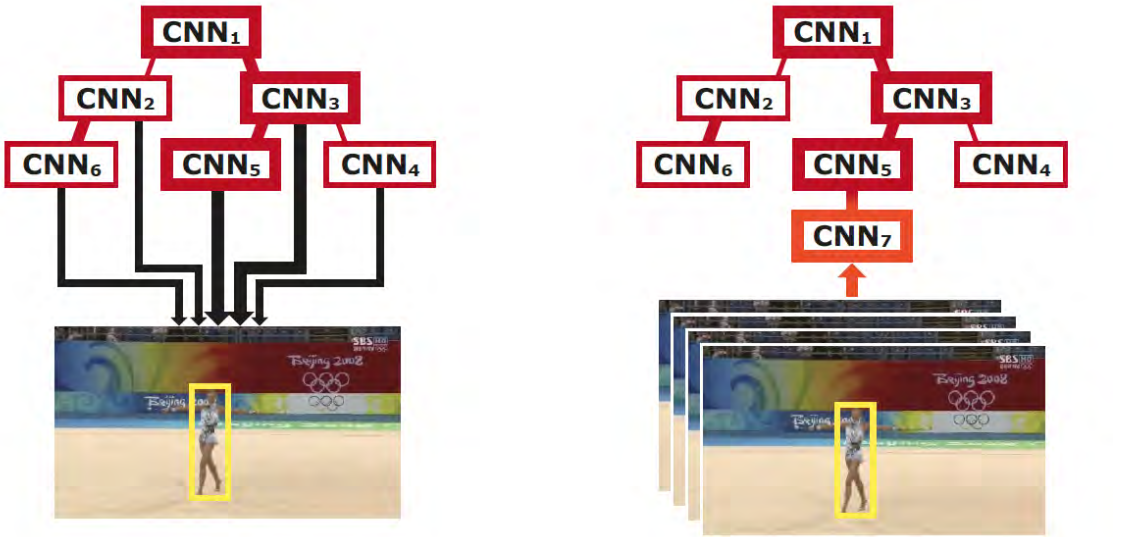
All trackers described below track one object (called a target) in videos. For the first frame of the video the trackers (with the exception of Oliva's tracker) receive the true location of the target (in the form of a bounding box) and then estimate the location of the target in all the following frames. At each point in time the tracker can only use the current frame and the previously seen frames, but no frames that come later in the video (online tracking). Two of the presented trackers (TCNN and C-COT) were among the top performing trackers in the VOT2016 challenge [5]. Both TCNN and C-COT use Convolutional Neural Networks (CNNs).

## 2.1 Tree of Convolutional Neural Networks (TCNN)

TCNN [7] uses up to 10 CNNs, which are stored in a tree structure, where a CNN in a child node is a fine-tuned version of the CNN in its parent node. This allows the tracker to use multiple models of a target object, which can change appearance both abruptly (CNNs in different branches) and smoothly (CNNs along the same branch).

All CNNs use the same trained layers as a CNN pre-trained on ImageNet, except for the fully connected layers and the output layer, which is reduced to size 2. The output of every CNN is a score for target and background. The input to the CNNs is a 75x75 pixels RGB image.

The tracker is initialized on the ground truth bounding box. For every following frame, the new bounding box is found by first extracting target candidates around the previous location of the target. Each of these target candidates is assigned a score by at most 10 CNNs, that were most recently added to the tree. The scores for each target candidate are combined into a weighted average, where CNNs that voted for a candidate with higher confidence and that are considered to be more reliable get a higher weight. The candidate which received the highest weighted average score is chosen as the new target location. This 75x75 pixels candidate bounding box is then tightened, using a regression function learned in the first frame, to get the final bounding box estimate for that frame. Every ten frames, the model is updated by adding a new CNN to the tree. The parent node of the new CNN is the CNN, which performed best at scoring the 10 most recent frames. The new CNN is a copy of its parent, which is additionally trained on the 10 most recent frames (see Fig. 1).




(a) State estimation (b) Model update

Figure 1: TCNN keeps multiple CNNs in a tree structure, which score candidates to determine the current target location (a). The model is updated by adding a new CNN (b). The width of the black arrow stands for the weight of a CNN's score. The thickness of the outline indicates a CNN's reliability. [7]

**Advantages of TCNN:**

- fine-tunes its CNNs during tracking to better fit the present data

- can handle both abrupt and smooth changes of the model's appearance

**Disadvantages of TCNN:**

- If a CNN was trained on incorrectly estimated bounding boxes that CNN is erroneously fine-tuned to recognize background as target. Although TCNN computes the reliability of a CNN to give such a CNN's vote less weight, an incorrect estimation decreases the overall performance of the tracker.
- searches the target around the previous location of the target – it might not find a fast moving target
- does not use methods that models object movement

## 2.2 Continuous Convolution Operator Tracker (C-COT)

C-COT [2] aims for high accuracy in localizing the target by computing a continuous confidence score function, which assigns a target score to every location in an image patch centered on the previous target location. A high target score means that the tracker is confident that a point inside the region is part of the target object.

The confidence score function is computed by convolving a set of learned continuous convolution filters with a feature map, which consists of the input image patch and convolutional layers, that are extracted from a CNN, that was pre-trained on ImageNet. The parameters of the continuous convolution filters are refined at each frame after tracking.

To train these filters, an image patch – centered on the target bounding box and 25 times its size – is extracted from the frame. This image patch is input to the CNN. In the process of classifying, the image patch is passed through a series of convolutional layers that extract different types of features: shallow layers extract low-level features, like edges, at a high spatial resolution, while deeper layers extract high-level features, which are more suitable for image classification, but at lower spatial resolution. To benefit from both properties, C-COT fuses the multiple resolution feature map with an interpolation operator. The filters are trained to assign a confidence score to the fused feature map that closely matches the desired confidence score output, which is a Gaussian function centered on the estimated location.

To localize the target, the trained filters are convolved with the interpolated feature maps from the current frame to produce a confidence score. The target is localized where the confidence score reaches its peak (Fig. 2).

**Advantages of C-COT:**

- computes a continuous score, which enables sub-pixel accuracy
- uses not only the final classification output of a CNN but multiple convolutional layers

**Disadvantages of C-COT:**

- uses the CNN as a feature extractor, but there are not many consistent features in our target (mainly due to motion blur)
- does not use methods that detect object movement

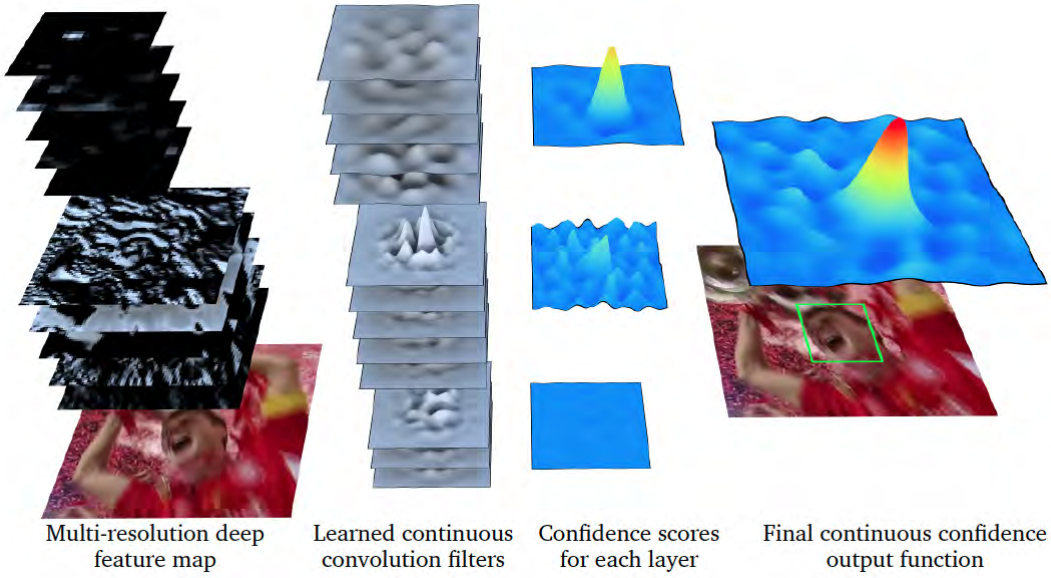



Figure 2: Continuous convolution filters (second, from left) are applied to a multi-resolution feature map (left) to produce a continuous confidence score of the target (right, top). The target's center is localized at the peak of the confidence score (right, bottom, green box) [2].

- does not fine-tune its CNN on new data during tracking. It uses a CNN that is trained only on ImageNet, which makes it vulnerable to blurry images [3].
- searches the target around the previous location of the target – it might not find a fast moving target

## 2.3 Oliva's Tracker

Oliva's tracker [8] was developed to track golden-collared manakin males in videos that were recorded earlier by the same team of biologists as the present data. In those videos the background was in general less bright and less saturated than in the present videos. Oliva's tracker, however, is based on the assumption that the neck of the golden-collared manakin male is yellow and highly saturated, which distinguishes it from the background.

For each frame, the tracker builds a foreground mask using Mixture of Gaussians (MOG) [9], and excludes from the foreground mask pixels whose hue and saturation do not lie between fixed lower and upper thresholds. The Morphological opening removes noise and enlarges the blobs in the foreground mask. The detected blobs are associated with the target in the previous frame, based on their distance from the current target location as predicted by a Kalman filter [10] and the similarity of their hue histogram. The most similar blob is chosen as the target in the current frame. Once the trajectory of the bird is determined for the entire video, its jumps between saplings are described by parabolas, which are fitted using non-linear regression.

Oliva determined the thresholds for hue and saturation based on the dataset he was using. For the evaluation of his tracker we used the same thresholds and did not adapt them to the present dataset.

**Advantages of Oliva's tracker:**

- takes advantage of the fact that our dataset was recorded with a stationary camera (blob detection, Kalman filter)
- delivers a compact description of the trajectory (parabolas)

**Disadvantages of Oliva's tracker:**

- tracks only the bird's yellow neck
  - ⇒ The estimated bounding boxes do not contain the entire bird.
  - ⇒ If the neck is not visible or the neck's hue and saturation do not lie within fixed thresholds, the bird cannot be reliably tracked.
  - ⇒ Only the male has a yellow neck, so this tracker cannot be extended to track the female bird.
- estimates the current location with a Kalman filter, which predicts an ongoing movement of the target. In cases where the bird abruptly stops or changes direction, the Kalman filter fails to correctly predict the birds location.

# 3 ManakinTracker – Proposed Approach

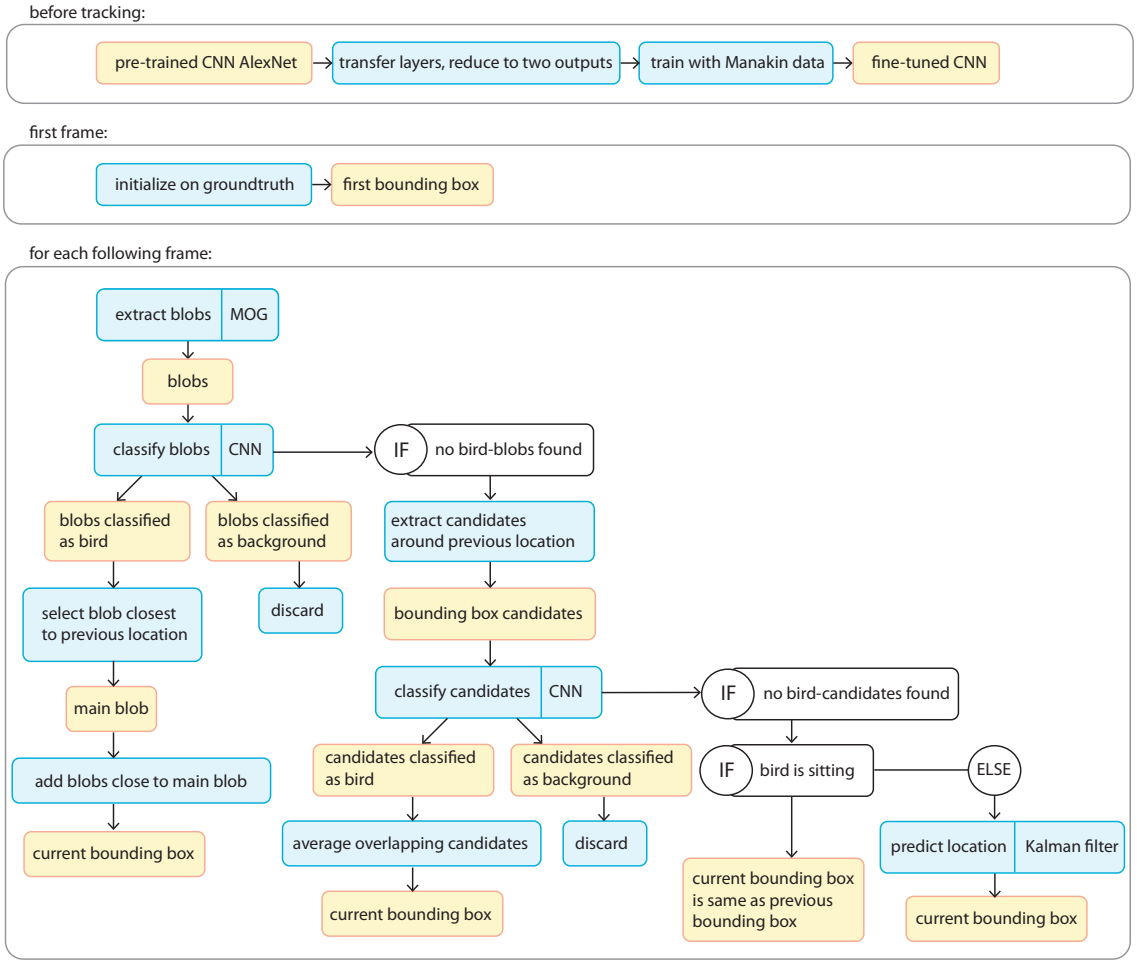



Figure 3: Flowchart of ManakinTracker.

We propose a tracker, that

- detects moving objects with a Mixture Of Gaussians model (MOG) [9],
- decides if a candidate location visually resembles a male golden-collared manakin with a fine-tuned Convolutional Neural Network (CNN),
- and estimates the location of the target using a Kalman filter [10] for frames without reliable visual cues (see Fig. 3).

## 3.1 Mixture Of Gaussians: Blob Detection

As the videos were recorded with a stationary camera, we can use a method based on background subtraction to segment the foreground. We chose Mixture Of Gaussians (MOG) because it can handle small movements in the background. For every frame, MOG generates a foreground mask from which we extract moving objects, called blobs (Fig. 4). Out of these candidate blobs, we aim to select the ones that contain the target – and discard those that contain other objects such as moving leaves, branches or the female bird.

Figure 4: Foreground mask (right) generated by Mixture Of Gaussians model of frame (left). The blue box marks the extracted blob.

## 3.2 CNN architecture

To decide which candidates contain the target, we use a CNN, as CNNs have shown top performance in object classification in images. Our CNN is based on AlexNet [6], which is a CNN pre-trained on ImageNet, that takes 227x227 pixels RGB images as input. We use the pre-trained layers of AlexNet for our CNN, except for the last 3 layers, which we replace with a new fully connected layer, a new softmax-layer and a new output layer to match our two classes: target (i.e. male golden-collared manakin) and background (i.e. forest). The output of our CNN is a background score and a target score in [0, 1]. We fine-tune this new CNN with image patches of male golden-collared manakins and background cropped from a set of sequences in our dataset, that does not contain the sequence in which we currently track the bird (see Section 4.1). During tracking the CNN is not updated further.

## 3.3 Kalman Filter

A Kalman Filter is used to predict the target location in absence of a reliable location estimation. A reliable location estimation is given by a blob, or candidate location(s) around the target's previous position, that receives a high target score by our CNN. We also use the Kalman Filter's location estimation if we find more than one blob. In this case, we select the blob that is closest to the Kalman Filter's location estimation. The linear Kalman Filter is initialized with the ground truth bounding box in the first frame and updated with the estimated target location at each frame after tracking. The linear Kalman Filter estimates an ongoing movement of the target at constant velocity, based on the target's previous locations.

## 3.4 Tracking

The male bird's location is initialized with the ground truth bounding box in the first annotated frame. For each frame, we detect blobs and classify them with our CNN. Out of the blobs that receive a target score greater than $t_1$, we select the one that is closest to the location predicted by the Kalman filter as our main blob. The bird can be partly occluded (e.g. by the sapling it sits on), so we add blobs

to the main blob, which were classified as target (target score greater than $t_1$) and which are close to the main blob. Since we do not know the width of the saplings, we consider two blobs as close if the distance between the two bounding box boundaries is less than the largest width or height of one of the two blobs. If we find a blob or combination of blobs, that fits these conditions, it becomes the target bounding box for the current frame (Fig. 5).


Figure 5: The two small blobs (small blue bounding boxes) are combined into a bigger blob (big blue bounding box). The white text indicates the blobs' target scores.

Figure 6: Bird is sitting and no blob was found (red box: candidate locations with target score >= $t_2$, blue box: blob, white box: final bounding box)

If we find no blobs in a frame or none that receive a high enough target score we search the bird in the region around its previous location. This usually happens, when the bird is not moving and thus not recognized as a blob. We shift the previous bounding box to the left, right, top, bottom and diagonally and crop image patches at these candidate locations. We resize these image patches to fit the CNN's input size and classify them with the CNN. A candidate location which receives a target score of $t_2$ or above, is assumed to contain the target. To avoid false positives, we exclude candidate positions, which do not overlap with the majority of overlapping candidate positions that were classified as target. The average of these candidate locations is the bounding box for the current frame (Fig. 6). Since the bird tends to sit still at the start of the video, we keep the initial bounding box if the CNN assigns a target score of $t_2$ or above to the image patch cropped at the bird's initial location and if we find no blob in that frame.

In our experiments, we achieved good results when we set the thresholds $t_1$ and $t_2$ to 0.9 and 0.99, respectively. $t_2$ is used for image patches that are not based on

blobs and thus typically contain sitting birds. $t_2$ is set to 0.99 instead of 1, because we noticed that when the bird is sitting, the ground truth location usually gets a score of 0.99 and above, but that candidate locations which receive a score of 1 are often not a better estimation than those with a slightly lower score. $t_1$ is set to a lower value than $t_2$ because $t_1$ is used for image patches that are based on blobs. In those cases the bird is typically moving which can make it harder to recognize for the CNN due to motion blur and shape distortion.

In case we find no candidate locations, which we can assume to contain the target – because the bird is largely or fully occluded or unrecognizable to the CNN – we will rely on the location predicted by the Kalman filter (Fig. 7). This prediction can be unreliable because the bird might either move or sit while it is not visible. The Kalman filter estimates the bird's movement as a continuation of its previous movement, which is only useful if the bird keeps moving while it is occluded. Because of this, we only use the Kalman filter's estimation if we assume that the bird is moving (i.e. not sitting). Otherwise, we use the bird's previous location as the current target bounding box.

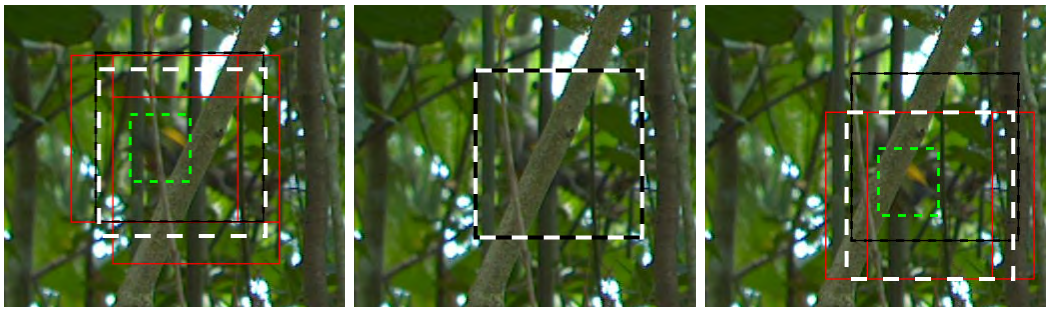


Figure 7: Middle: The Kalman filter's location estimation (black box) is used as the bounding box output when the bird becomes invisible to the camera during a jump. Left, right: The bird is visible and can thus be recognized by the CNN. (green boxes: ground truth; red boxes: candidate locations classified as target; white boxes: bounding box output for current frame)

We assume that the bird is sitting in the following cases:

- If both the current and the previous bounding box are based on blobs, and the current bounding box is contained within the previous bounding box. This case occurs either if the bird moved during sitting or if it has only been sitting for a short time and is thus still recognized as foreground by the MOG model.
- If both the current and the previous bounding box are not based on blobs, and the region of overlapping candidate locations in the current and previous frame overlap by at least $t_3$. $t_3$ should be low enough to allow for slight movement of the bird during sitting but high enough that slow jumping does not qualify as sitting. Setting $t_3$ to 80% gave good results in our experiments.

If the bird is recognized as sitting, the Kalman filter is re-initialized to the current location, because the bird tends to jump in a different direction than before it landed, and the Kalman filter would predict an ongoing movement in the same direction as before the landing. If the bird leaves the frame – i.e. we estimate its location to be

outside the frame – candidate locations are placed along the edges of the frame to detect the bird when it re-enters the frame. To avoid a false positive detection while the bird is outside the frame, the target bounding box estimate must be based on a blob in this case.

**Advantages of the ManakinTracker:**

- uses a Kalman Filter and MOG, and thus does not rely only on the visual information in a single frame but detects motion based on multiple frames
- uses a CNN, that is fine-tuned specifically to recognize male golden-collared manakins (including highly blurry and partly occluded) with high accuracy
- In most cases (particularly during jumps) blobs give very accurate bounding boxes and no further correction of the bounding boxes' dimensions is necessary.
- can track the bird efficiently in most frames by classifying only a limited number of image patches extracted from blobs (usually 1-4 per frame)

**Disadvantages of the ManakinTracker:**

- is fine-tuned to track golden-collared manakins. To track a different target, the CNN needs to be trained with ground truth data of that target.
- requires video sequences as input that were recorded with a stationary camera

## 3.5 Open Issues

Using blobs as bounding boxes works well as long as the entire bird moves – if the bird sits still and moves only partly, for example only its head, the bounding box enclosing the blob will only contain the moving part of the bird.

When the bird lands on a thin sapling, the sapling moves along with the bird, which leads to a blob that includes parts of the sapling along with the bird. This could be corrected in post-processing by adjusting bounding boxes, that strongly increased in size shortly before the bird started sitting.

When we shift the bounding box to different locations around the previous location, the bird is typically detected within multiple candidate bounding boxes. Taking the average of those bounding boxes is often not the best estimation of the bird's location. The bird's head alone seems to be easier to recognize for the CNN than the tail alone. As a result, the averaged bounding boxes are often not centered on the bird (see Fig. 8).

In some cases the tracker recognizes the female bird as the target, even though the CNN was only trained on male birds. This suggests that the CNN is not dependent on the male bird's yellow neck for a correct classification. The downside of this is, that if the male and female bird are both present in a frame the two have to be distinguished by the tracker. One solution would be to train a CNN on images of female birds also, which would require ground truth bounding boxes for the female birds. Currently, we handle this issue by choosing the blob that is closest to the location predicted by the Kalman filter if there are multiple blobs that get a high target score.

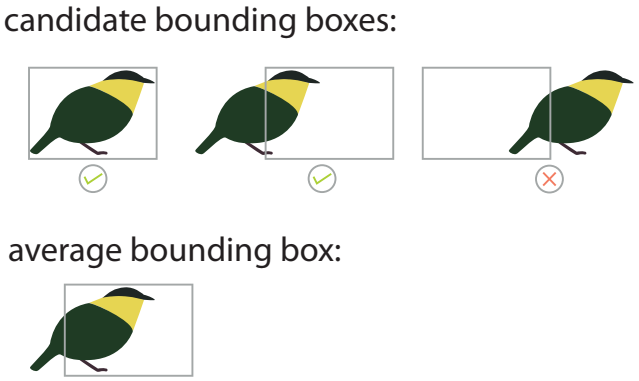



Figure 8: If the CNN classifies an image patch of the bird's head as target (top row, middle), but not an image patch of the bird's tail (top row, left), the average bounding box is biased towards the bird's head (bottom row).

# 4 Evaluation

To evaluate the performance of the ManakinTracker, we compared it to the other three trackers presented in Section 2: TCNN, C-COT and Oliva's tracker. A good tracker should be robust – avoid losing the target during tracking – and accurate – achieve a large overlap between its estimation of the bounding box and the ground truth. We evaluated the trackers' performance on our test dataset (see Section 4.1). When a tracker estimates a bounding box that has zero overlap with the ground truth bounding box, the tracker is re-started at the next frame that has a valid ground truth annotation. Only Olivas's tracker is not re-started, because it cannot be initialized with a ground truth bounding box.

## 4.1 Manakin Dataset

The dataset we use for evaluating the trackers consists of 78 video sequences. Every second frame in a sequence has a ground truth annotation, except for frames where the bird is out of frame. The ground truth annotations were provided by the biologists. Each sequence has 216 ground truth annotations on average (see Appendix A for more detail).

All videos were recorded with stationary high-speed cameras in the Panama forest. The video sequences show the male golden-collared manakin practicing its courtship dance alone, or performing it accompanied by a female.

## 4.2 Metrics

We assess tracking performance based on two measures: accuracy and robustness.

**Accuracy (bounding box overlap)** is the average overlap ratio between the estimated bounding boxes and the ground truth bounding boxes. To measure how accurate a tracker's bounding box estimate at frame t is, we calculate the overlap ratio between the estimated bounding box $b_t$ and the respective ground truth bounding box $g_t$.

$$accuracy(t) := \frac{area(g_t \cap b_t)}{area(g_t \cup b_t)}$$

**Robustness (number of re-starts)** is determined by the number of times the tracker has to be re-started during tracking. If the tracker loses the target, it is restarted by re-initializing the tracker at the next frame that has a ground truth annotation.

**Bird's speed**: If the target moves fast, it can become harder to recognize due to motion blur and harder to find than if it remained at its previous location. To measure how fast the target moves from one frame to the next, we calculate the overlap-ratio between consecutive ground truth bounding boxes $g_t$ and $g_{t+1}$ for each frame t and subtract the result from 1. If the bird's speed is high for a larger number of consecutive frames, that indicates that the bird is jumping between saplings; if speed is above zero only for a small number of frames the bird keeps sitting while moving (e.g. opening its wings or moving the head).

$$bird's\ speed(t) := 1 - \frac{area(g_t \cap g_{t+1})}{area(g_t \cup g_{t+1})}$$

| Tracker | Average overlap (accuracy) | Average number of re-starts (robustness) |
|---|---|---|
| ManakinTracker (Ours) | 58.3093% | 1.2051 |
| C-COT | 49.0720% | 4.6795 |
| TCNN | 53.3405% | 7.1154 |
| Oliva's | 1.7013% | – |

Table 1: Trackers' average performance on the test dataset.

## 4.3 Results

Table 1 shows that ManakinTracker performed the best out of the three trackers, both in terms of accuracy (58.3093% average overlap) and robustness (1.2051re-starts per sequence on average). TCNN achieves higher accuracy (53.34%) than C-COT (49.07%), but needs about 1.5 times more re-starts on average. Oliva's tracker achieved an average overlap of only 1.62%. In only 20 out of the 78 test sequences, Oliva's tracker could estimate at least one bounding box that overlaps with the ground truth (see Fig. 15). The reason for the bad performance is most likely that the fixed thresholds used to distinguish between background and target based on hue and saturation do not fit the present data, in which the background is brighter and more saturated than in the older videos. In cases where Oliva's tracker was able to track the target, overlap is still low (see Fig. 19) because the tracker's estimated bounding boxes contain only the bird's neck instead of the entire bird.

To analyze the trackers' results in more detail, we looked at the sequences where each tracker reached its highest and lowest accuracy (see Figures 16, 17, 18, 19). Figures 17, 18 show that all the trackers' performance is strongly related to the bird's motion. While the bird sits, the bounding box overlap changes only slightly, but when the bird's speed is high for a number of frames (i.e. the bird is jumping) accuracy tends to decrease strongly. In both the sequences where C-COT and TCNN achieve lowest accuracy respectively (Fig. 17 and 18, left), as well as the sequence where TCNN reaches maximum accuracy (Fig. 18, right) the tracker lost the target while it moved at a high speed. On the other hand, C-COT and ManakinTracker

both achieved maximum accuracy during sequence 90, in which the bird does not move for the majority of the frames, but as soon as the bird's speed increases, accuracy decreases sharply for both trackers (Fig. 17, 16, right).

The ManakinTracker's performance, however, can also increase when the bird is moving (Fig. 16, left) because the ManakinTracker uses blobs as target candidates. Those blobs are detected based on MOG and a fast-moving target can be an advantage for this technique because it shows up more clearly in the foreground mask than a stationary or slowly moving object.

The ManakinTracker was re-started most often in sequences 10 (10 re-starts), 6 (10 re-starts), and 41 (7 re-starts) (Fig. 9). In sequence 10, the tracker is re-started repeatedly when the bird sits, facing away from the camera, and the CNN detects an orange leaf as the target instead. In sequence 6, the tracker is re-started when both the male and the female bird receive a high target score, but the female is incorrectly chosen as target because it is a blob and blobs, if classified as target, are preferred as target objects by the ManakinTracker. In sequence 41 the tracker is re-started when a yellow leaf, resembling the bird's head, receives a higher target score than the sitting bird, which is heavily occluded.

TCNN and C-COT both use CNNs pre-trained on ImageNet. The performance of networks trained on the ImageNet dataset, which consists of still images, decreases strongly if images are blurry [3]. In contrast, the ManakinTracker's CNN was trained also on blurry images, extracted from videos similar to the ones it was tested on.

TCNN and C-COT look for objects in the current frame that have similar features to the target object in the previous frame. On our target, however, local features (like points and edges) often disappear due to motion blur.

### 4.4 Performance of the ManakinTracker's CNN

To train our CNN, we cropped image patches of size 227x227x3 (to fit the CNN's input layer) of the male golden-collared manakins centered on the ground truth bounding boxes, and image patches of the background at locations other than the ground truth bounding boxes. We then trained one CNN on the first half of the crops (8806 samples of male bird, 7927 samples of background) and a second CNN on the second half (8805 samples of male bird, 7926 samples of background). Both CNNs reached an accuracy of 98% on the test set, which was the set of crops each CNN was not trained on.

False negative classifications occurred if the bird was highly occluded, largely out of frame or extremely blurry; false positives seem to be caused mostly by yellow leaves which resemble the bird's neck (Fig. 11). True positives could even be achieved on image patches with different lighting conditions and also when the bird is blurry or occluded (Fig. 10).

## 5 Conclusion

In this report, we presented the ManakinTracker, a novel approach to track male golden-collared manakins in videos recorded in the wild. Our tracker can deal with a fast moving target, using a method based on Mixtures of Gaussians, thus taking advantage of the fact that the videos were recorded with a stationary camera. The ManakinTracker can classify image patches with high accuracy, using a CNN

that was fine-tuned with image patches extracted from videos of golden-collared manakins.

We evaluated the ManakinTracker's performance on 78 sequences of golden-collared manakins and it achieved higher accuracy and robustness than two of the winners of the VOT2016 challenge and another tracker that was specifically designed for tracking male golden-collared manakins in an older dataset.

# 6 Future Work

During the tracking process the CNN is not updated – we expect that fine-tuning the CNN with images patches extracted during tracking would decrease the number of tracking failures.

The ManakinTracker is also able to track the female bird, which does not have a prominent feature like the male bird's yellow neck, without any specific training. We would need ground truth of the female bird, however, to quantify how well the female bird can be tracked. We expect that we can easily extend the ManakinTracker to track the female bird as well. However, this also poses a challenge to the tracker, because we have to avoid confusing the two birds. The recommended solution is to train a CNN with three classes: male bird, female bird, and background. This would require ground truth bounding boxes of the female bird.

Another approach would be to train the CNN with more background samples. Currently we extract about one background sample for every frame in the training set. We could improve the CNN's ability to classify female birds as background by extracting many more background samples from the entire frame, so that we get more samples of the female bird. This could also further improve the CNN's accuracy.

A possible solution to ensure that the bird is not lost while it is sitting, would be to integrate a correlation filter in the tracker: if two image patches extracted from consecutive frames at the same location have a very high correlation, we can assume that the bird is still sitting at the same location and discard other candidates.


Figure 9: Examples of frames that caused ManakinTracker to fail. Sequence 10 (top, left), sequence 41 (top, right), sequence 6 (bottom) (ground truth: green box; estimated target bounding box: white box; target detected in candidate location: red boxes; blob: blue box; white text: target score of blob)


Figure 10: Examples of image patches correctly classified by our CNN


Figure 11: Examples of image patches incorrectly classified by our CNN

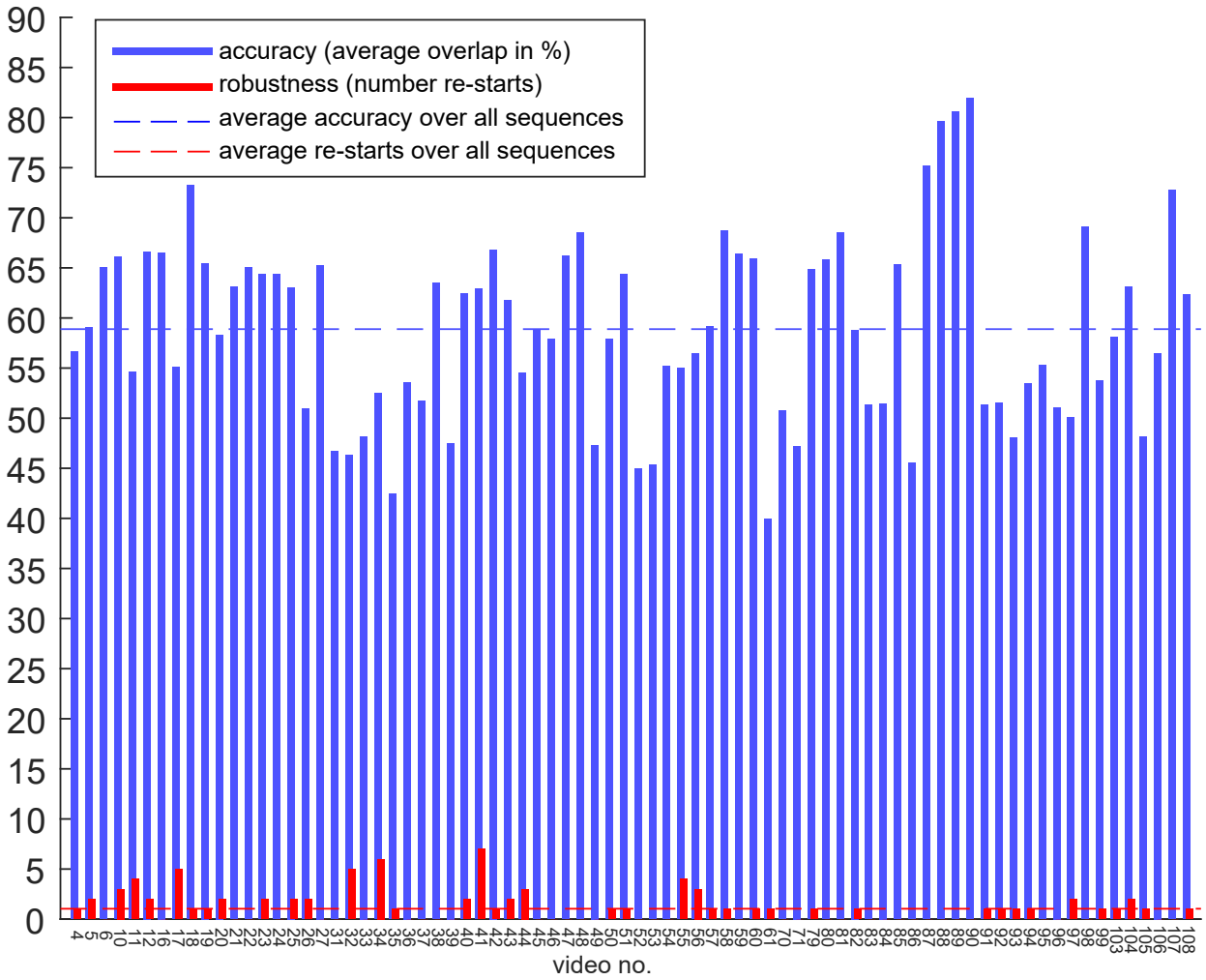



Figure 12: ManakinTracker: Overview

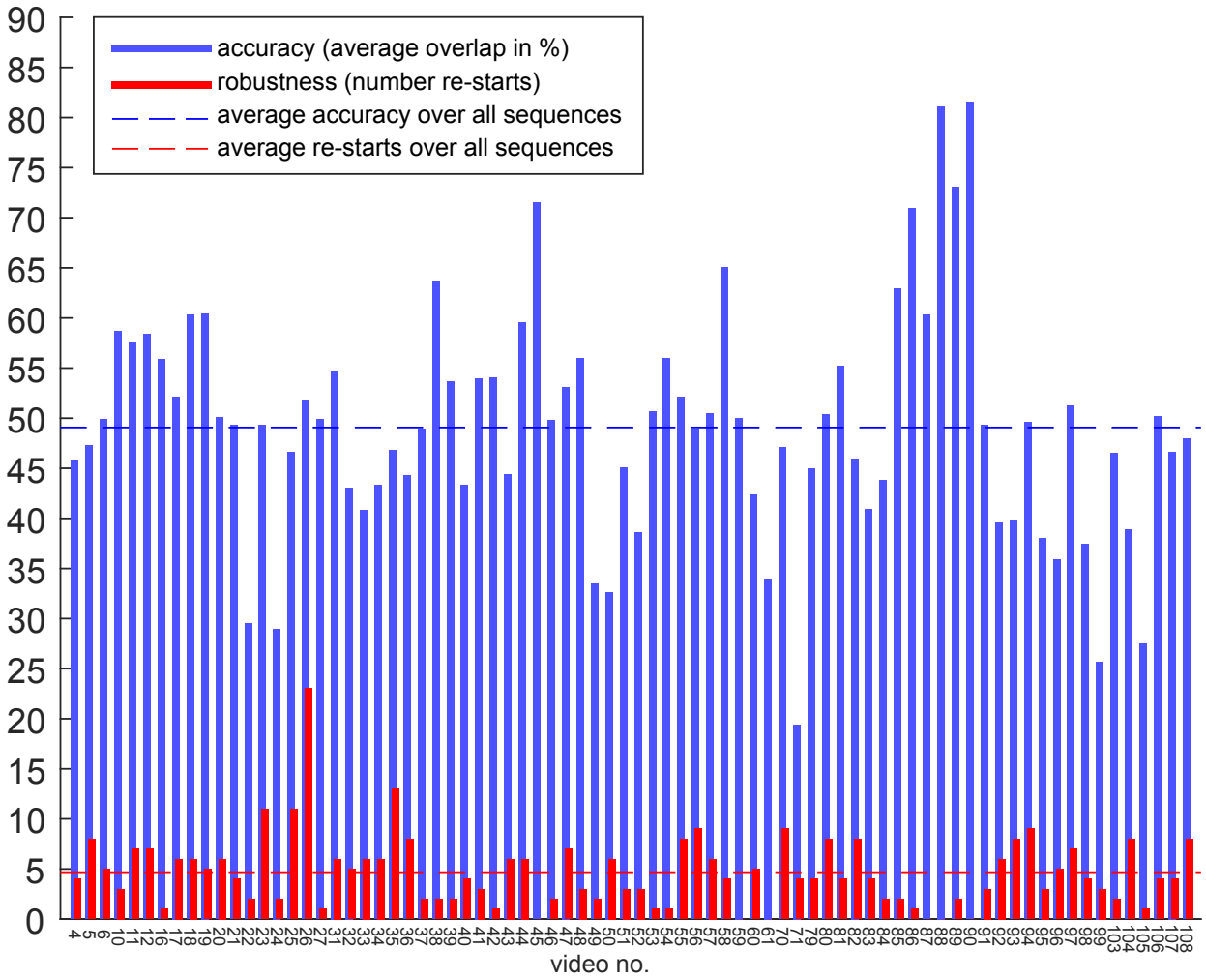



Figure 13: C-COT: Overview

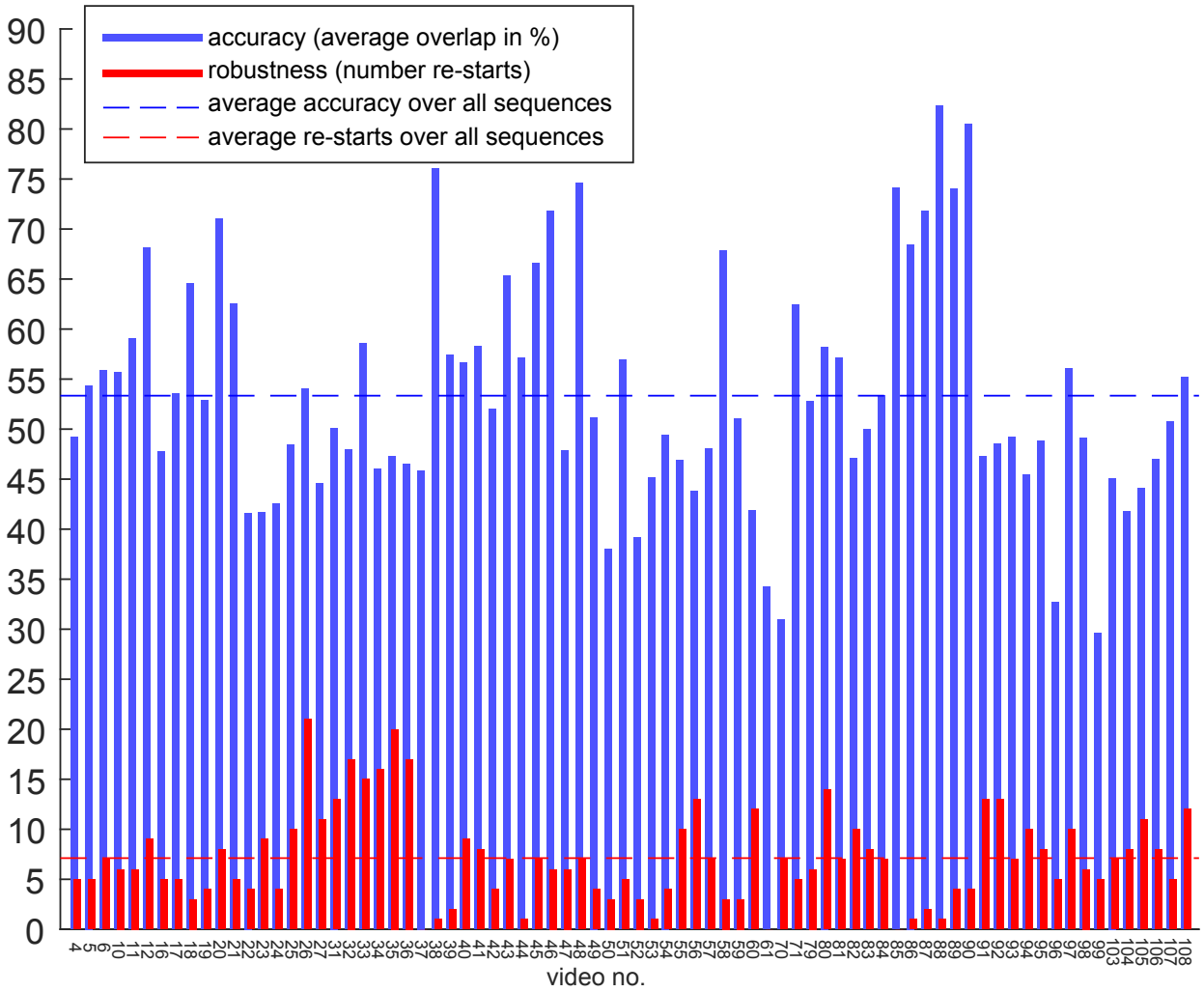



Figure 14: TCNN: Overview

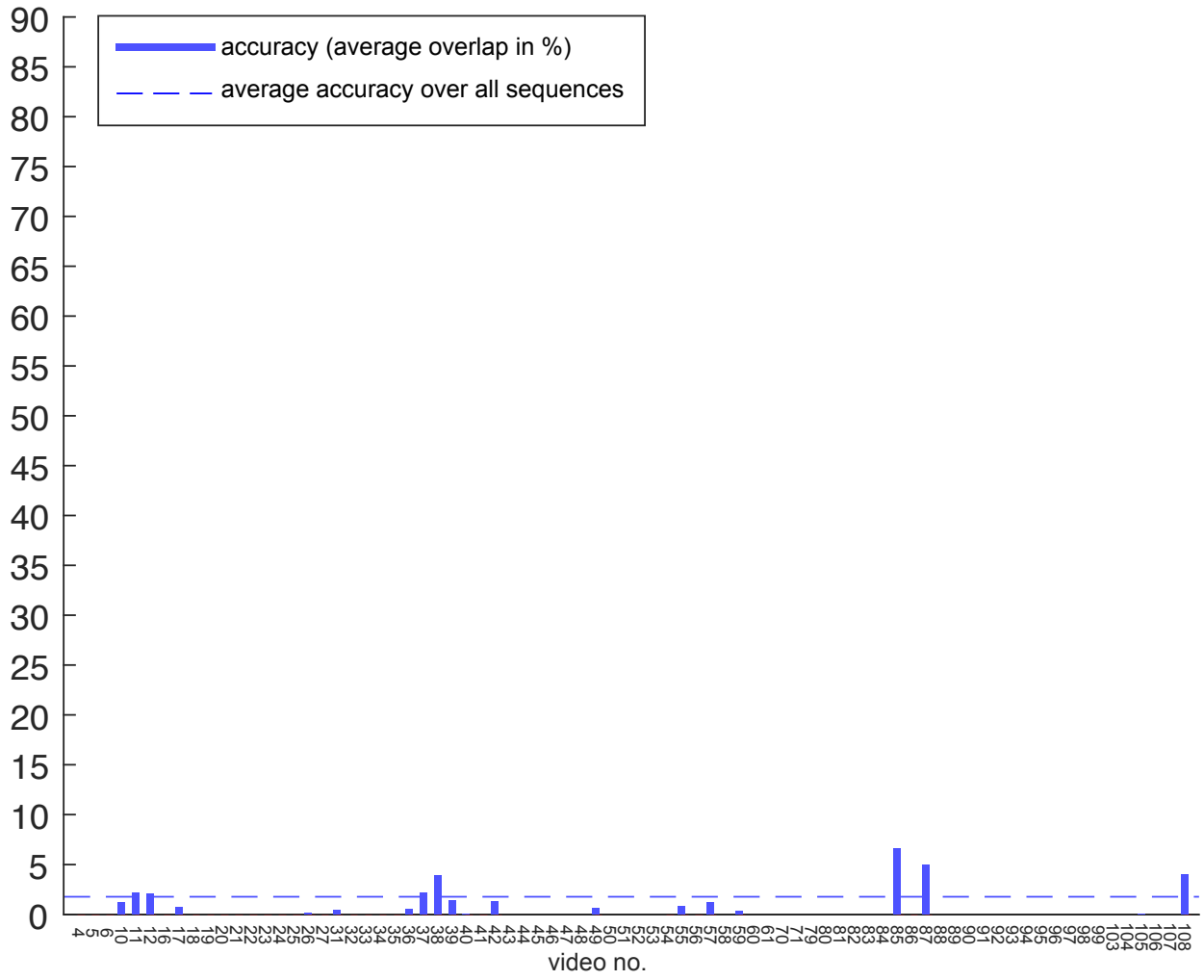



Figure 15: Olivas's tracker: Overview

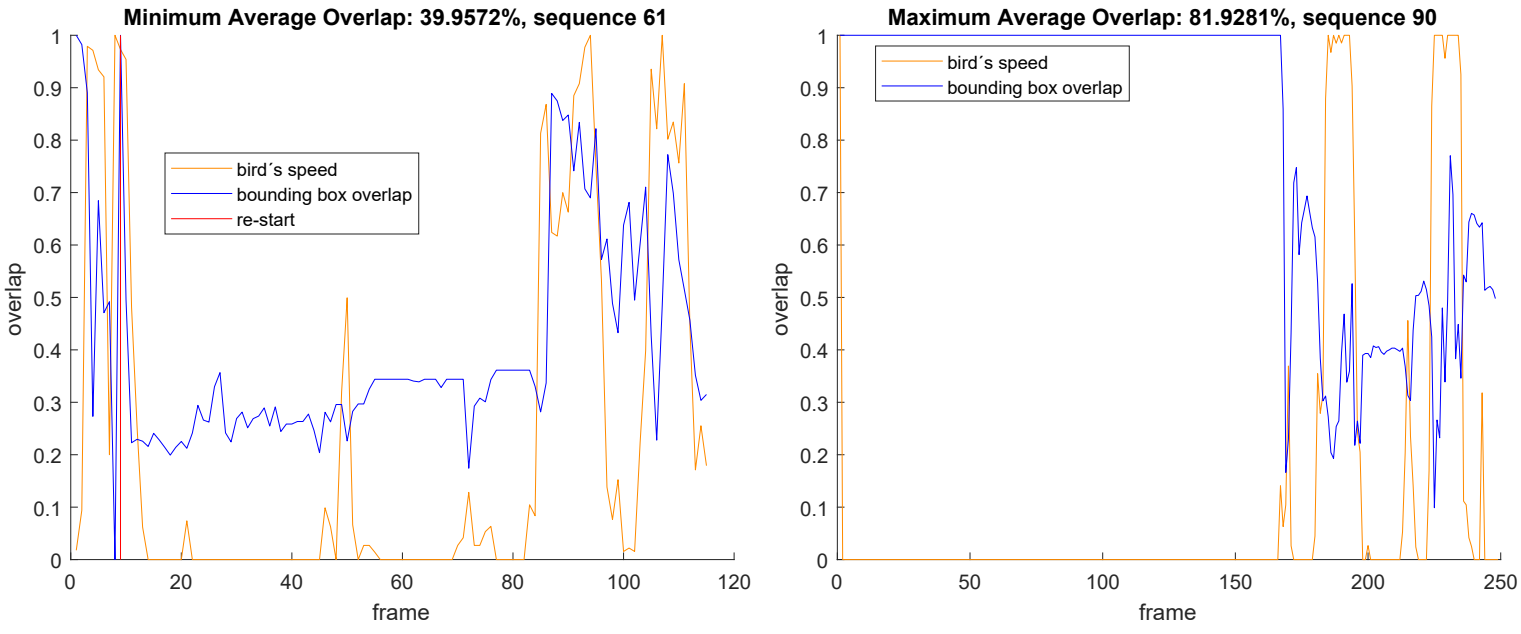
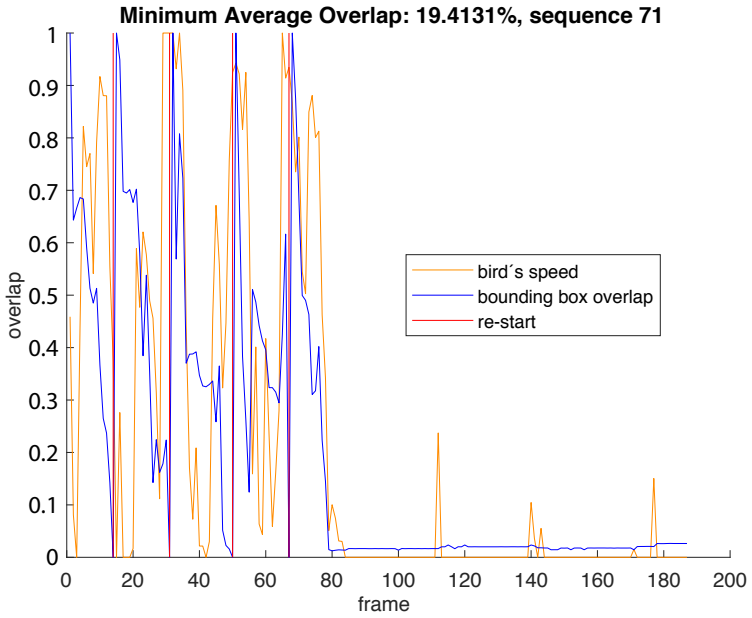



Figure 16: ManakinTracker



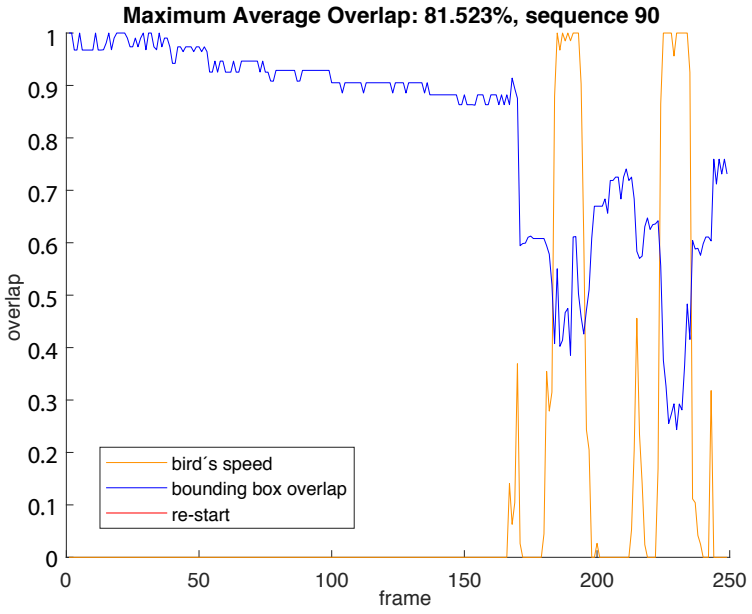
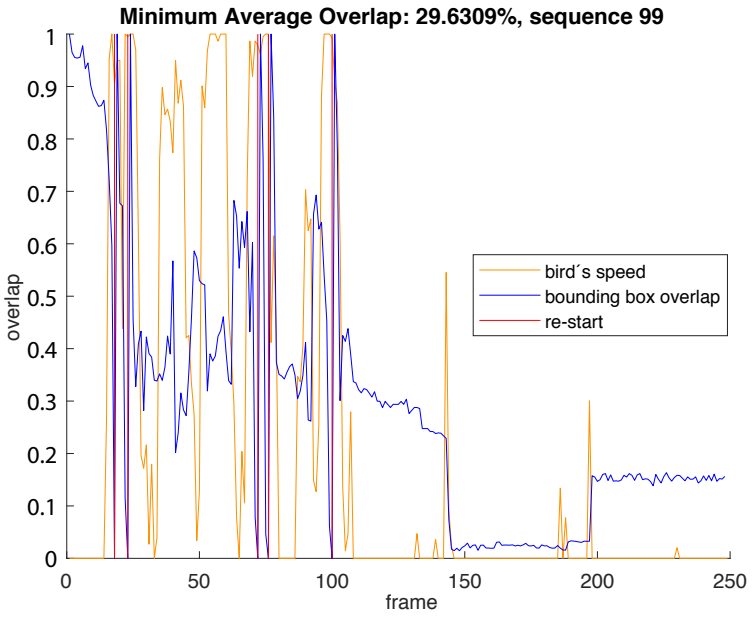




Figure 17: C-COT



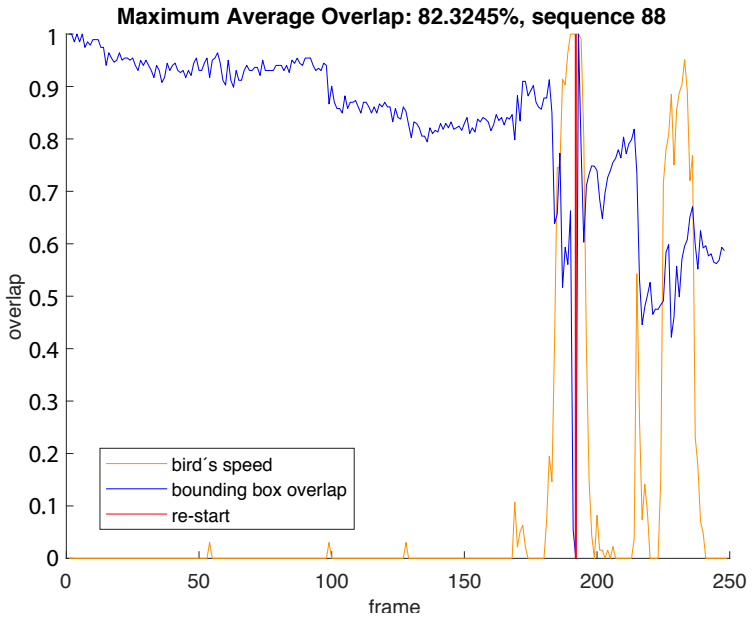
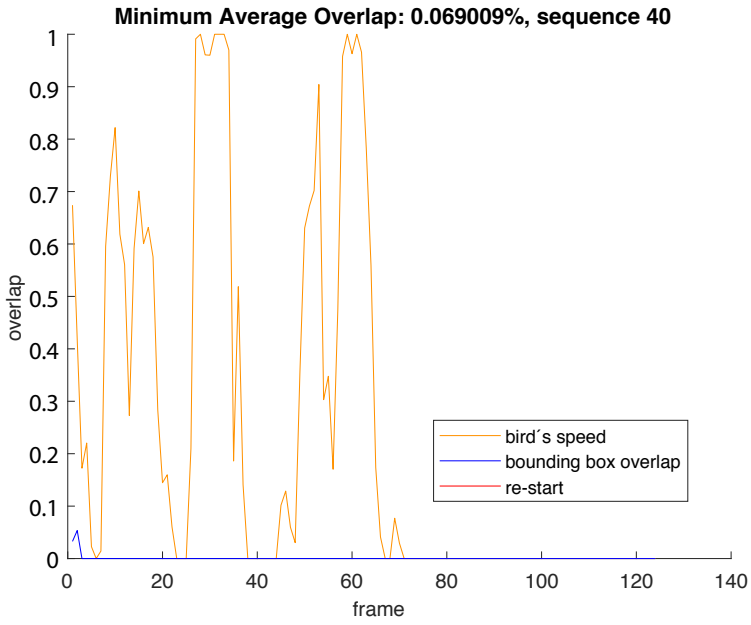




Figure 18: TCNN



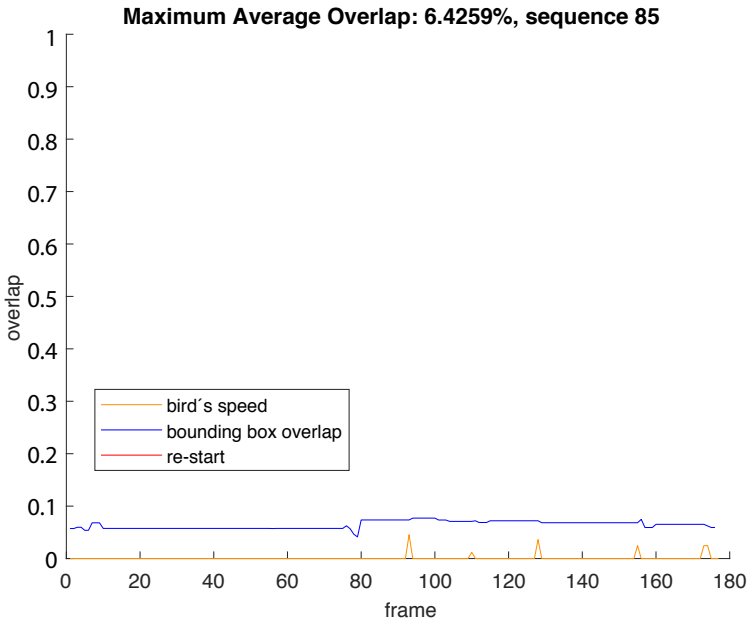




Figure 19: Oliva's tracker

# References


[1] J. Barske, B. A. Schlinger, M. Wikelski, and L. Fusani. Female choice for male motor skills. *Proceedings of the Royal Society of London B: Biological Sciences*, 2011.

[2] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016.

[3] S. F. Dodge and L. J. Karam. Understanding how image quality affects deep neural networks. *CoRR*, abs/1604.04004, 2016.

[4] 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.

[5] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, et al. *The Visual Object Tracking VOT2016 Challenge Results*, pages 777–823. Springer International Publishing, Cham, 2016.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[7] H. Nam, M. Baek, and B. Han. Modeling and propagating cnns in a tree structure for visual tracking. *CoRR*, abs/1608.07242, 2016.

[8] L. Oliva, A. Saggese, N. M. Artner, W. G. Kropatsch, and M. Vento. From trajectories to behaviors: an algorithm to track and describe dancing birds. Pattern Recongition and Image Processing Group, TU Wien and PRIP Club, Vienna, Austria, 2017.

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

[10] G. Welch and G. Bishop. An introduction to the kalman filter, 1995.

# A Appendix

This table provides the names of the sequences, that we refer to when using the respective sequence number in this report, along with the number of ground truth-annotated frames in that sequence:

| sequence number | sequence name | # ground truth-annotated frames |
|---|---|---|
| 4 | 20170315_145443.21994752 | 193 |
| 5 | 20170315_145443.21994769 | 183 |
| 6 | 20170315_145443.22023677 | 193 |
| 10 | 20170316_065121.21994752 | 238 |
| 11 | 20170316_065121.21994769 | 242 |
| 12 | 20170316_065121.22023677 | 250 |
| 16 | 20170316_142215.21994752 | 160 |
| 17 | 20170316_142215.21994769 | 160 |
| 18 | 20170316_142215.22023677 | 161 |
| 19 | 20170316_151125.21994752 | 143 |
| 20 | 20170316_151125.21994769 | 144 |
| 21 | 20170316_151125.22023677 | 143 |
| 22 | 20170316_152525.21994752 | 148 |
| 23 | 20170316_152525.21994769 | 150 |
| 24 | 20170316_152525.22023677 | 149 |
| 25 | 20170316_154525.21994752 | 250 |
| 26 | 20170316_154525.21994769 | 250 |
| 27 | 20170316_154525.22023677 | 250 |
| 31 | 20170402_082143.21994752 | 226 |
| 32 | 20170402_082143.21994769 | 250 |
| 33 | 20170402_082143.22023677 | 235 |
| 34 | 20170402_085109.21994752 | 250 |
| 35 | 20170402_085109.21994769 | 250 |
| 36 | 20170402_085109.22023677 | 250 |
| 37 | 20170402_141006.21994752 | 250 |
| 38 | 20170402_141006.21994769 | 250 |
| 39 | 20170402_141006.22023677 | 250 |
| 40 | 20170402_152038.21994752 | 249 |
| 41 | 20170402_152038.21994769 | 250 |
| 42 | 20170402_152038.22023677 | 250 |
| 43 | 20170403_070820.21994752 | 250 |
| 44 | 20170403_070820.21994769 | 250 |
| 45 | 20170403_070820.22023677 | 250 |
| 46 | 20170403_074232.21994752 | 250 |
| 47 | 20170403_074232.21994769 | 250 |
| 48 | 20170403_074232.22023677 | 250 |
| 49 | 20170403_074259.21994752 | 250 |
| 50 | 20170403_074259.21994769 | 250 |

| sequence number | sequence name | # ground truth-annotated frames |
|---|---|---|
| 51 | 20170403_074259.22023677 | 250 |
| 52 | 20170403_083222.21994752 | 250 |
| 53 | 20170403_083222.21994769 | 250 |
| 54 | 20170403_083222.22023677 | 250 |
| 55 | 20170403_083258.21994752 | 220 |
| 56 | 20170403_083258.21994769 | 220 |
| 57 | 20170403_083258.22023677 | 221 |
| 58 | 20170403_140649.21994752 | 182 |
| 59 | 20170403_140649.21994769 | 220 |
| 60 | 20170403_140649.22023677 | 219 |
| 61 | 20170403_140816.21994752 | 116 |
| 70 | 20170406_064801.21994752 | 235 |
| 71 | 20170406_064801.21994769 | 188 |
| 79 | 20170406_074047.21994752 | 220 |
| 80 | 20170406_074047.21994769 | 221 |
| 81 | 20170406_074047.22023677 | 221 |
| 82 | 20170414_142705.21994752 | 186 |
| 83 | 20170414_142705.21994769 | 187 |
| 84 | 20170414_142705.22023677 | 187 |
| 85 | 20170415_062646.21994752 | 250 |
| 86 | 20170415_062646.21994769 | 250 |
| 87 | 20170415_062646.22023677 | 250 |
| 88 | 20170415_075022.21994752 | 250 |
| 89 | 20170415_075022.21994769 | 250 |
| 90 | 20170415_075022.22023677 | 250 |
| 91 | 20170415_083144.21994752 | 250 |
| 92 | 20170415_083144.21994769 | 250 |
| 93 | 20170415_083144.22023677 | 250 |
| 94 | 20170416_082405.21994752 | 173 |
| 95 | 20170416_082405.21994769 | 166 |
| 96 | 20170416_082405.22023677 | 170 |
| 97 | 20170417_065607.21994752 | 250 |
| 98 | 20170417_065607.21994769 | 250 |
| 99 | 20170417_065607.22023677 | 250 |
| 103 | 20170425_074803.21994752 | 204 |
| 104 | 20170425_074803.21994769 | 201 |
| 105 | 20170425_074803.22023677 | 200 |
| 106 | 20170425_075212.21994752 | 100 |
| 107 | 20170425_075212.21994769 | 101 |
| 108 | 20170425_075212.22023677 | 101 |

# References

[1] J. Barske, B. A. Schlinger, M. Wikelski, and L. Fusani. Female choice for male motor skills. *Proceedings of the Royal Society of London B: Biological Sciences*, 2011.

[2] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016.

[3] S. F. Dodge and L. J. Karam. Understanding how image quality affects deep neural networks. *CoRR*, abs/1604.04004, 2016.

[4] 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.

[5] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, et al. *The Visual Object Tracking VOT2016 Challenge Results*, pages 777–823. Springer International Publishing, Cham, 2016.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[7] H. Nam, M. Baek, and B. Han. Modeling and propagating cnns in a tree structure for visual tracking. *CoRR*, abs/1608.07242, 2016.

[8] L. Oliva, A. Saggese, N. M. Artner, W. G. Kropatsch, and M. Vento. From trajectories to behaviors: an algorithm to track and describe dancing birds. Pattern Recongition and Image Processing Group, TU Wien and PRIP Club, Vienna, Austria, 2017.

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

[10] G. Welch and G. Bishop. An introduction to the kalman filter, 1995.