Multiple Ant Tracking with Global Foreground Maximization and Variable Target Proposal Distribution

Mary Fletcher
Dept of Computer Science
Colby College
Waterville, ME 04901
mvfletch@colby.edu

Anna Dornhaus
Dept of Ecology and Evolutionary Biology
University of Arizona
Tucson, AZ 85721
dornhaus@email.arizona.edu

Min C. Shin
Dept of Computer Science
Univ of North Carolina at Charlotte
Charlotte, NC 28223
mcshin@uncc.edu

Abstract

Motion and behavior analysis of social insects such as ants requires tracking many ants over time. This process is highly labor-intensive and tedious. Automatic tracking is challenging as ants often interact with one another, resulting in frequent occlusions that cause drifts in tracking. In addition, tracking many objects is computationally expensive. In this paper, we present a robust and efficient method for tracking multiple ants. We first prevent drifts by maximizing the coverage of foreground pixels at at global scale. Secondly, we improve speed by reducing markov chain length through dynamically changing the target proposal distribution for perturbed ant selection. Using a real dataset with ground truth, we demonstrate that our algorithm was able to improve the accuracy by 15% (resulting in 98% tracking accuracy) and the speed by 76%.

1. Introduction

The social network formed by an ant colony is of significant interest to biologists. It is relevant to understanding social organization in animal groups, and, in the case of social insects, deriving the self-organized algorithms by which collective behavior is optimized. Analysis of such social networks requires tracking the ants' motion and interactions. Manually following every ant in a colony is extremely time consuming, limiting research progress in this field. Therefore, in recent years there have been attempts to automate the process.

There are two main challenges that we address in this paper. First, due to the social nature of ants, they interact frequently. Interaction often involves them coming very close and touching or crawling over each other. Moreover, they can bend and turn away from the camera. Both cases lead to noisy measurements which cause drifting of tracking onto nearby ants. This makes the algorithm vulnerable to multiple tracks latching onto the same ant (Figure 2). The second challenge is speed. Ant colonies often contain more than 50 interacting ants and, to study their interaction networks, they may be filmed for more than 5 minutes (9K frames). The automated tracking is even more computationally expensive as the ants are tracked simultaneously. Markov Chain Monte Carlo [2] has been shown to reduce the computation requirement; however, with such a large dataset, further increasing the speed of the algorithm is important.

Figure 1. Tracking multiple ants. Results over 1000 frames are shown in black trails.

In this paper, we propose to improve the reliability and speed of multiple ant tracking. First, we reduce the drifting onto a nearby ant by adding a global measure to ensure proper tracking. We noted that during occlusions and/or non-rigid motion (such as bending), the score space becomes less reliable often resulting in multiple local maxima ultimately leading to tracking error (Figure 2). In many cases, the number of objects tracked stays constant (thus no object leaving or entering the scene) for a certain period of

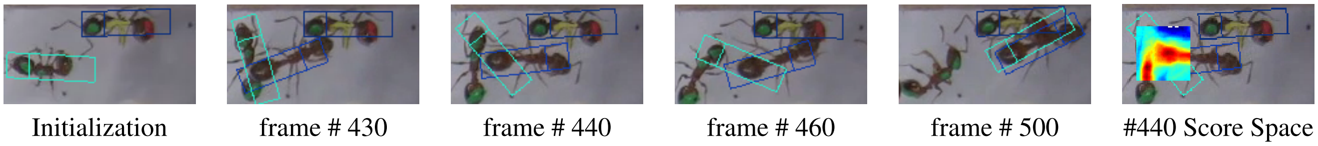

Figure 2. Occlusion and Non-rigid motion causing a drift in tracking. The ant (boxed in cyan) was initialized while it was bending (frame #1). It unbends and is occluded by another ant at frame #430 which causes a drifting onto that ant. The scoring space of frame #440 is shown on the far right. The color of each point reflects the score of the best orientation centered at that point. Red indicates high score (likeliness). Note the local maxima which lead to drifting. Eventually, tracking of two ants (cyan and blue) are following the same ant in frame #500.

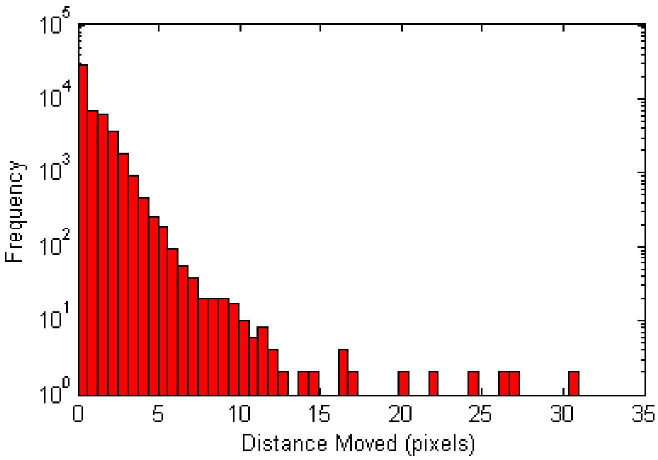

Figure 3. Movement distribution of ants. A histogram of distance traveled between two adjacent frames is computed from 49 manually tracked ants in 1000 frames. Note that y-axis is *log* scale, and most of the ants are not moving much.

time. In our dataset, the entire colony was included in the images, so the number of ants in each frame did not vary often. We leverage the assumption that all ants pixels are tracked to reduce drifting by encouraging the tracking of all ants to collectively (globally) cover as much foreground as possible. Second, we significantly increase the efficiency of the algorithm by using a variable target proposal distribution. We noted that not all ants move at all times (Figure 3), a fact that could apply not only to insect tracking, but to many other tracked objects as well. Rather than using a fixed target proposal distribution in Markov Chain Monte Carlo (MCMC) [2], we dynamically vary the distribution of targets to perturb based on the level of expected motion. This results in a higher level of sampling for ants that are likely to move more, without increasing the total number of samples needed to achieve the same performance.

This paper is organized in the following manner: Section 2 discusses related work, Section 3 provides an overview of MCMC-based tracking, Section 4 describes the appearance model we use, Section 5 explains how we use global foreground maximization to prevent drift, Section 6 describes how we vary a target proposal distribution based on predicted motion, and the remainder presents the performance evaluation.

2. Related Work

Object tracking has been studied extensively, and can be surveyed in [9]. Recently, there has been a growing interest in tracking animals and biological objects. A general framework for recognizing animal behavior using an affinity graph has been demonstrated on a synthetic animal video [8]. A large number of flying bats have been tracked using two cluster data association [1]. Honeybees in a hive have been tracked with adaptive appearance models [5] and their motion behavior was recognized using HMM [7]. Flying animals also have been tracked in 3D using multiple cameras using Extended Kalman Filter with data association [6]. Multiple ants have been tracked using MCMC [2, 3]. One of main challenges in tracking social insects is that they often interact. This interaction causes multiple tracks to overlap and occlude. Assuming that targets usually remained separate from one another, Khan *et al*. proposed an interaction factor to discourage tracks from overlapping [2]. In a later approach, they allowed merged measurements due to overlap and used a constant motion model to correctly track the targets after interaction [3]. In a densely populated ant colony like our dataset, however, ants often change their motion *direction* after interaction with others.

3. MCMC Multi-target Tracking

We build our tracking based on a multi-target tracking using Metropolis-Hastings Markov Chain Monte Carlo (MCMC) [2]. Compared to a joint particle filter, an MCMC-based method was shown to be much more efficient with a given number of "particles." We include a brief explanation for the portion of MCMC that is related to the proposed method.

Each ant is parameterized by its position (x, y) and orientation (θ). The joint state X contains the state of all ants. First, we initialize the markov chain sampler by taking a random sample from the markov chain of the previous frame. Let that sample be X^0. Each iteration of MCMC is listed below. Let n denote each iteration.

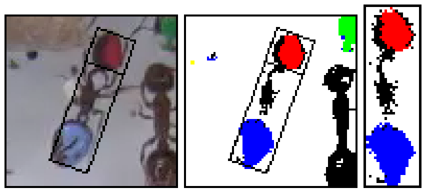

Figure 4. Appearance Model. Original image (left), color classification image (middle), and appearance model (right)

1. Let X^* be the previous sample, X^{n-1}.
2. Select one target m^* to perturb from a target proposal distribution, T. Note that this is uniform and fixed, and we propose to vary it over time (Section 6).
3. Perturb only the state of target m^* according to a single-target proposal distribution (state evolution).
4. Compute the acceptance ratio as $\alpha = \frac{P(Z_m|X_{m^*})}{P(Z_m|X_m)}$
5. Add X^* as X^n to the markov chain with probability of α. If the proposed configuration is rejected, then X^{n-1} is repeated as X^n.

The single-target proposal distribution is modeled with a zero-mean normal distribution. We use $\sigma_x = 2, \sigma_y = 2, \sigma_\theta = 4$. We draw N samples where N is the sum of N_{burn} burn-in samples and N_{mix} samples to form the markov chain. The locations of ants are estimated as the joint state (a sample in markov chain) in which $\prod_m P(Z_m|X_m*)$ is maximized. In our dataset, many ants move less than one pixel per frame, and their motion tends to be rather jittery. We found the single-target proposal distribution to spread the particles sufficiently for tracking, so we omit the motion model.

4. Appearance Modeling

Previously, unmarked ants have been tracked, and their appearance was modeled by t-distributions [2]. Such an approach is suitable in video with a lower density where occlusion is relatively infrequent. Our dataset contains over twice as many ants as in [2], and subsequently has a higher rate of occlusion. The *directionality* of motion often changes during occlusion, so motion prediction such as [3] does not suffice. To achieve higher reliability in long-term tracking of interacting ants, we proposed to track paint-marked ants. Note that even with the colored marks, the tracking can easily drift off due to occlusion and non-rigid motion (Figure 2).

To leverage the unique appearances of color marks, we have modeled the appearance of each ant as a template of pixels. Simply using the color of each pixel in the template, even after transforming to colorspaces such as LUV, was found to be unstable. Rather than using color *itself*, we model the appearance based on a template of *classified* color (Figure 4). There are 6 colors: 4 paint colors, natural ant color and background color. We model and classify each color class using a single Gaussian in LUV color space for each. The appearance model for each ant is initialized based on manual initialization for the first frame of the video (Figure 4). Then, the weight for each guess of an ant is computed as

$$P(Z_m|X_{m^*}) = \frac{1}{a\sqrt{\pi}} e^{-(x^2/a^2)^k} \tag{1}$$

where x is the number of pixels in the template for that position that do not match the model. We used $a = 45$ and $k = .7$.

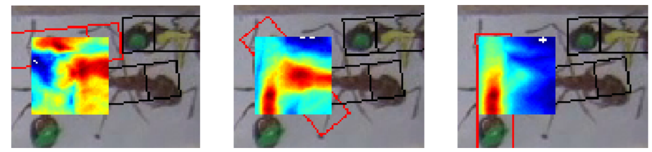

Figure 5. Improvement in Score Space due to Global Foreground Maximization. The color of each point reflects the score of the best orientation centered at that point. Red indicates high score (likeliness). Note that the extra local maxima are eliminated with global foreground maximization, resulting in accurate tracking.

5. Drift Prevention with Global Foreground Maximization

During occlusions, measurements become less reliable as some part of the appearance model could be hidden and occupied by other ants. A series of unreliable measurements could result in drifting onto a wrong ant nearby. Moreover, as ants are articulated objects, they frequently bend their head and abdomen. A rigid, fixed appearance model would not be able robustly track during such motion. However, nonrigid motion models would increase the dimensionality of state X and increase complexity of the tracking. Note that a general appearance model such as [4] is not suitable as we would like to leverage the unique colors of each ant.

We noted that, in many cases, the number of ants present in video stays constant for a certain finite duration. Our dataset is a video of the entire colony with a very low frequency in entering and exiting from the colony. Within such duration, the mis-tracking tend to increase the untracked (not enclosed by the rectangle appearance model) ant pixels by drifting onto the background or another ant. We propose to minimize the drifting by maximizing the tracked foreground pixels.

One method is individual maximization, which minimizes drift onto the background. In addition to the difference in appearance model (x) of scoring (Equation 1), we add a term for penalizing the inclusion of background pixels

$$P(Z_m|X_{m^*}) = \frac{1}{a\sqrt{\pi}} e^{-((x+b)^2/a^2)^k} \tag{2}$$

where b is the number of background pixels covered by the tracking. Here we use $a = 450$, and $k = 1.7$. Figure 5 shows how this eliminates maxima on the background, and creates a local maximum in the correct location, even when the appearance model is a poor match.

When those ants undergoing nonrigid motion are also occluded by other ants, the individual foreground maximization would not be able prevent multiple trackings from drifting onto the same ant. Thus, we propose to minimize such drifting by maximizing the foreground coverage *globally*, as multiple trackings latching onto same ant would reduce the coverage at the global level. The error space for global maximization in Figure 5 has eliminated the local maximum that would cause drift onto another ant. We discourage joint states with higher amounts of untracked ant pixels by multiplying the original appearance scoring (Equation 1) by a global factor

$$Q(Z_t|X_t) = f^\gamma \tag{3}$$

where f is the number of foreground pixels included in any template, and γ is a constant, which we found experimentally to work most effectively when $\gamma = 2750$ in our dataset. Combining this factor with the basic appearance modeling yeilds the following equation for computing the acceptance ratio.

$$\alpha = \frac{P(Z_{mt}|X'_{mt})}{P(Z_{mt}|X_{mt})} \frac{Q(Z_t|X'_t)}{Q(Z_t|X_t)} \tag{4}$$

6. Speed up with Variable Target Proposal Distribution

With a uniform target distribution, N iterations are evenly distributed among all targets. However, we noted that the vast majority of ants in our dataset do not move much between frames (Figure 3). Considerable variation among individuals in their behavior is frequently observed in biology. In ants it is particularly common, given that different ants have different roles in the colony. In fact, with all social insects a considerable proportion of individuals are usually observed as being inactive. For these very slow moving ants, it is likely that many fewer perturbations are required to adjust to the distribution in the next frame.

Thus, we speed up the algorithm by using a *variable* target distribution. By allowing the moving targets to be selected more often for perturbing, we reduce the overall length of markov chain needed (N) to achieve the same performance. We first estimate the expected level of motion by using the temporal difference (d_m). The difference in color classification between the previous and current frames at the previous location of target is computed as d_m. To prevent assigning ants zero probability of selection, a certain percentage of the total iterations (P_u) are split uniformly among all ants. The remainder are distributed in proportion to the temporal differences. Thus the target proposal distribution is

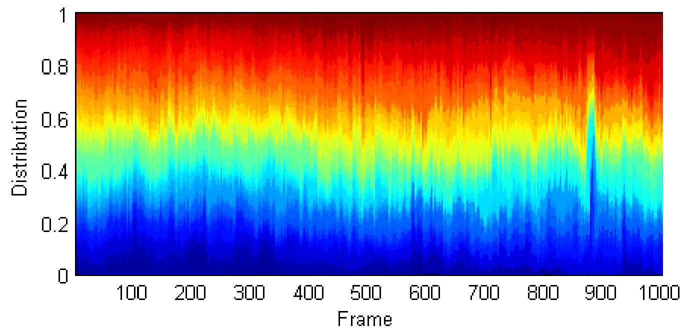

Figure 6. Variable target proposal distribution visualization. A specific color is assigned to each ant. The target proposal distribution is visualized for each frame (x-axis), with the height of each color indicating each ant's probability of selection.

$$T(m|d) = P_u \frac{1}{M} + (1 - P_u) \frac{d_m}{\sum_m d_m} \tag{5}$$

where M is the number of ants.

Variable distribution for our dataset is visualized in Figure 6. At each time point (x-axis), the cumulative distribution of all ants are shown on y-axis. Each ant is visualized with a specific color. The length of vertical line with same color indicates the fraction of T (Equation 5) assigned to that particular ant. The fluctuation of each color illustrates the change of the distribution for each associated ant. Note the sudden change near frame 900. That was due to a cockroach (not part of the actual experiment) crossing over a tracked ant causing a large increase in temporal difference. As a result, the ant was assigned a large number of perturbation. It was tracked successfully.

7. Results

7.1. Dataset

We evaluated the performance on a video of 49 *Temnothorax rugatulus* ants over 1000 frames, taken at thirty frames per second. A ground truth was obtained expediently by executing a single-target tracking and manually reinitializing whenever it drifted.

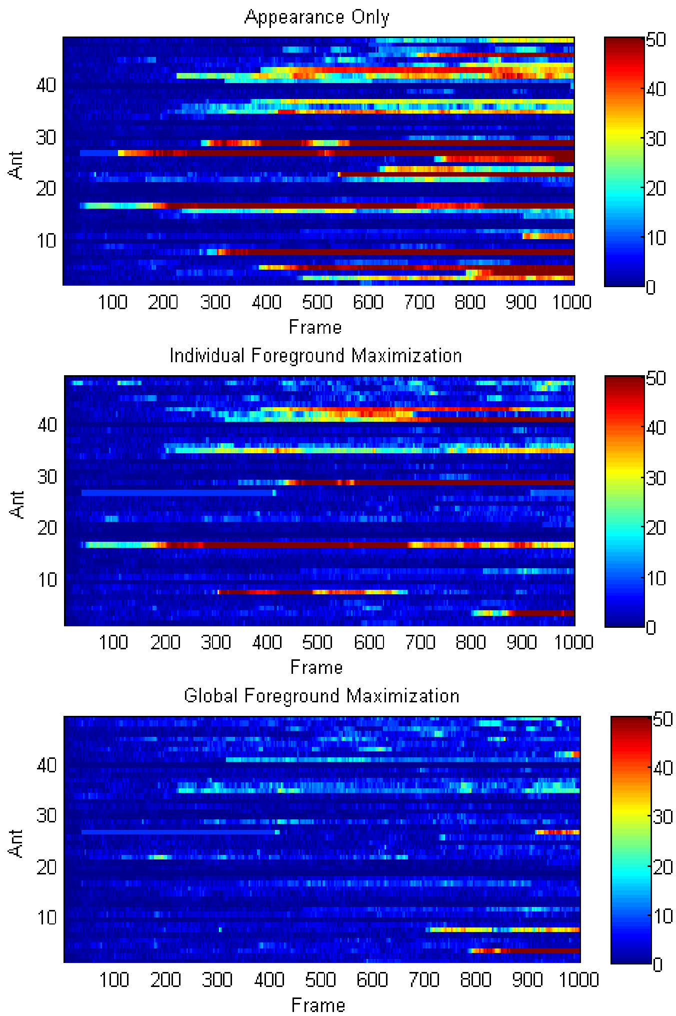

Figure 7. Comparison of Tracking Error over time. Each row corresponds to a different ant, and the x-axis is frame number. Y-axis indicates different ants. The error has been capped at 50 (red) to show the details in the lower error range. Locations in red ($\geq$ 50 pixel error) indicates drifting.

7.2. Tracking Accuracy

We measure tracking error as the distance between tracking positions and ground truth. The average of Euclidean distance between the centers, fronts, and abdomen is assessed. The error over time is visualized for three approaches of scoring: (1) appearance only, (2) with individual foreground maximization and (3) with global foreground maximization. Refer to Figure 7. Addition of individual foreground maximization reduces the drifts found without it. With global maximization, by the end of 1000 frames, only 4 out of 49 ants have drifted with error greater than 25 pixels. Examples of successful tracking during non-rigid motion, occlusion and occlusion during initialization

Method	Accuracy
Appearance only	82%
Appearance + Individual Foreground Max	92%
Appearance + Global Foreground Max	98%

Table 1. Improvement in tracked percentage using global foreground maximization.

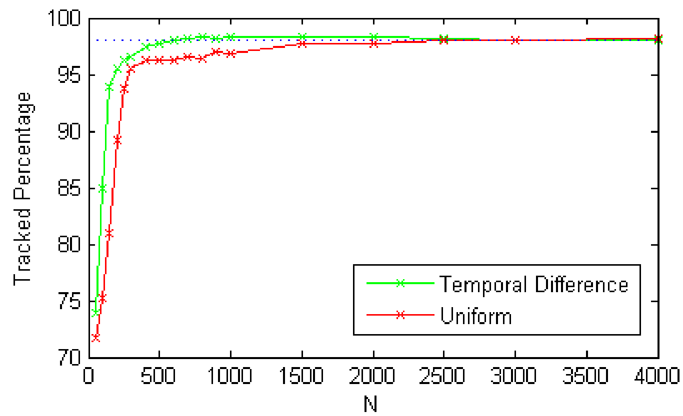

Figure 9. Variable distribution improves the tracked percentage for the same length of markov chain. Each point is the average performance of 100 trials. The dotted line indicates the maximum performance achieved when N=4000 was used.

are shown in Figure 8.

7.3. Tracked Percentage

Second, we measured the percentage of frames that were successfully tracked. For each ant in each frame, we consider it "tracked" if the average Euclidean distance is less than the typical width of head (20 pixels). Then we computed the percentage of the incidences of "tracked" out of 49,000 (= 49 ants/frame x 1000 frames). When tracked with $N = 2500$ (which resulted in the best performance), the tracked percentage was 82% with appearance only, 92% with individual foreground maximization and 98% with global maximization (Table 1). Note that the type of target proposal distribution used does not affect the maximum performance.

7.4. Level of Speed up

To evaluate the degree to which a variable distribution could reduce the necessary length of Markov chain (N), we executed tracking with N of 100 to 4000 using global foreground maximization. For a uniform distribution, this is equivalent to approximately 2 to 80 perturbations per ant. As MCMC involves randomization, we ran 100 times for each N, which allowed us to conduct statistical testing for equivalence. Averages of 100 executions at each N are

(a) Nonrigid Motion. The ant (boxed in green) is successfully tracked even when it curls up significantly.

(b) Occlusion. The ant (boxed in blue) is occluded, but correctly tracked.

(c) Occlusion during initialization. The ant (boxed in green) in the first frame was occluded by another ant (boxed in pink) .

Figure 8. Sample Tracking Results.

shown in Figure 9. To keep the computation load to a reasonable level, we conducted finer sampling of N near where the performance was changing more dramatically (lower N).

We found that the tracked percentage quickly increased with N. The performance increase dramatically slowed down beyond $N = 1500$, so we used the performance at $N = 4000$ as its maximum performance. Tracked percentage was 98%. As expected, the performance at $N = 4000$ was statistically same between the two distributions (uniform and variable).

To compute the speed up, we estimated N needed to reach the maximum tracked percentage for both distributions by comparing the performance at each N to the performance at $N = 4000$, using t-tests. Since the speed is directly proportional to N and the time required for computing temporal difference is minimal, we use the difference in N to estimate the speed up. We found that $N = 2500$ was needed for the uniform distribution while only 600 was needed for the variable distribution. Thus, the usage of variable distribution reduced the required N to achieve the maximum performance by 76%.

With variable distribution, the performance improvement slows down near $N = 150$. The performance was a very reasonable 94%. In comparison, the performance with the uniform distribution at $N = 150$ was 80%.

8. Conclusions and Future Work

We have demonstrated that globally maximizing tracked foreground pixels minimizes the drifting due to non-rigid motion and occlusions with other targets. Furthermore, we have shown that dynamically varying the target proposal distribution to concentrate more on the targets with greater movement enables a speed up by reducing the length of markov chain by 76%. We have noted that the level of activity (motion) varies over time, especially with a longer video. Thus, our future work includes varying the overall *length* of markov chain in addition to the distribution of markov chain to further increase the speed of tracking.

9. Acknowledgements

This study was supported by grants from the National Science Foundation (NSF-DBI-0754748 and 0851745). We would like to thank Jefferson Brady for the data ground truth.

References

[1] M. Betke, D. Hirsh, A. Bagchi, N. Hristov, N. Makris, and T. Kunz. Tracking large variable numbers of objects in clutter. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007. 2

[2] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets.

IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1805–1918, 2005. 1, 2, 3

[3] Z. Khan, T. Balch, and F. Dellaert. Mcmc data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1960–1972, 2006. 2, 3

[4] Z. Khan and F. Dellaert. Robust generative subspace modeling: The subspace t distribution. Technical report, GVU Center, College of Computing, Georgia, 2004. 3

[5] P. Maitra, S. Schneider, and M. Shin. Robust bee tracking with adaptive appearance template and geometry-constrained resampling. *Applications of Computer Vision (WACV), 2009 Workshop on*, pages 1 –6, dec. 2009. 2

[6] A. D. Straw, K. Branson, T. R. Neumann, , and M. H. Dickinson. Multi-camera real-time three-dimensional tracking of multiple flying animals. *J R Soc Interface*, 2010. 2

[7] A. Veeraraghavan, R. Chellappa, and M. Srinivasan. Shape-and-behavior encoded tracking of bee dances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:463–476, 2008. 2

[8] X. Xue and T. C. Henderson. Video-based animal behavior analysis from multiple cameras. *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, pages 335 –340, sep. 2006. 2

[9] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006. 2