Ant Tracking with Occlusion Tunnels

Thomas Fasciano*, Anna Dornhaus† and Min C. Shin*
* Univ. of North Carolina at Charlotte †University of Arizona
Charlotte, NC 28223 Tucson, AZ 85721
twfascia@uncc.edu dornhaus@mail.arizona.edu mcshin@uncc.edu

Abstract

The automated tracking of social insects, such as ants, can efficiently provide unparalleled amounts of data for the of study complex group behaviors. However, a high level of occlusion along with similarity in appearance and motion can cause the tracking to drift to an incorrect ant. In this paper, we reduce drifting by using occlusion to identify incorrect ants and prevent the tracking from drifting to them. The key idea is that a set of ants enter occlusion, move through occlusion then exit occlusion. We do not attempt to track through occlusions but simply find a set of objects that enters and exits them. Knowing that tracking must stay within a set of ants exiting a given occlusion, we reduce drifting by preventing tracking to ants outside the occlusion. Using four 5000 frame video sequences of an ant colony, we demonstrate that the usage of occlusion tunnel reduces the tracking error of (1) drifting to another ant by 30% and (2) early termination of tracking by 7%.

1. Introduction

Social insects, such as ants, have become model systems for understanding division-of-labor, task specialization, and distributed problem-solving [12] as well as studying adaptive networks [4]. Traditionally, biologists have modeled the interactions by manually watching the videos. To maximize the throughput of the analysis, automated tracking methods have been proposed [7]. However, the tracking is challenging as social insects interact frequently leading to a high level of occlusion. Moreover, they are very similar in appearance. Thus, the tracking tends to drift onto a wrong ant or terminate early. Additional markings have been used to assist automated tracking such as QR codes [9] or color paints/tags [5]. However, the QR codes are too big and heavy for general applicability to a wide range of insects Also, the color paints/tags have a much lower number of available permutations and ant shape undergoes non-rigid deformation which add challenges to the robust modeling of appearance even with paint markings. Moreover, these

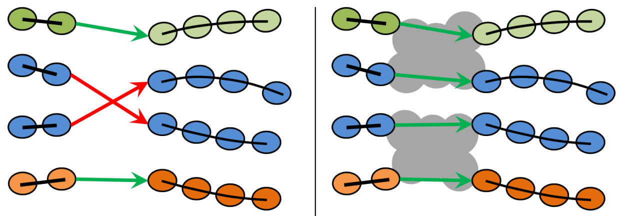

Figure 1. Due to difficulty of detection during occlusions, tracking trajectories are often broken into smaller tracklets, represented as connected colored circles, which must be matched together to form continuous tracking. Note that two tracklets with similar appearance and motion feature could cause incorrect matching as highlighted by red arrows while green arrows indicate correct matches in (a). Occlusion tunnel which is highlighted in gray in (b), groups ants that travel through same occlusion and filters the incorrect tracklet. Tracking is constrained to be within occlusion tunnel thus reducing drifting.

paint marks could fall off or be obscured reducing their reliability over time.

Particle filtering has been a popular method for tracking objects and have been applied for tracking insects as well [7, 14, 5]. However, occlusions have caused drifting to another insect that are often not recovered. Recently, data association based (DAT) methods have been used to recover the tracking. Rather than tracking sequentially, the detections in all frames are connected to form *tracklets* (short trackings) which are then matched to form longer tracking. The tracklets have been matched independently through a greedy approach [2, 11]. However, this can lead to ID switches (matching tracklets of two different objects) in social insect videos where objects with similar appearances are densely populated. Methods for simultaneously solving all trajectories have been proposed. These methods are often solved within predefined temporal windows to reduce problem complexity. Andriyenko *et al.* framed tracking as an energy minimization problem [1]. Huang *et al.* originally proposed a MAP formulation for the tracking problem which generates an initial set of short confident tracklets by linking detections in adjacent frames, which are then pro-

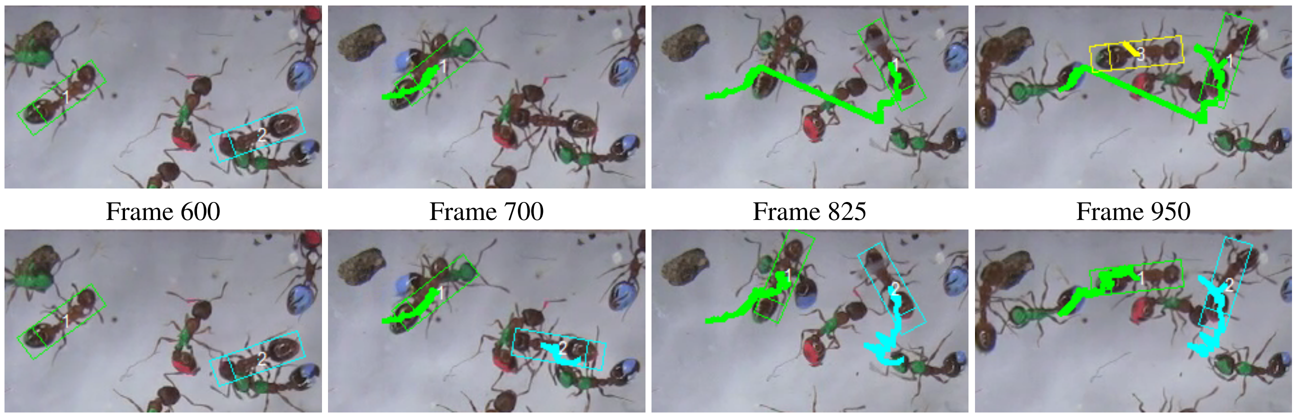

Figure 2. The top row shows an instance of an ID switch between two nearby, similar ants. Each ant is involved in separate, severe occlusions at the same time. The bottom row shows the results after filtering the respective confusing matches using occlusion tunnels.

gressively linked together to form full trajectories of tracking. The MAP formulation can be efficiently solved using the Hungarian algorithm [6] or minimum-cost network flows [16]. Much work has been done on the affinity measure (or linking cost) of the MAP framework. Many works have focused on improving the appearance modeling between tracklets using both offline and online learning frameworks [13, 15], as well as works to improve the motion prediction model [3].

Although these data association based methods can improve retention of tracking, due to the similarity of appearance and motion, they are still prone to two types of errors: matching to the incorrect ant (*ID switch*) or not matching to any thus terminating tracking early (*fragment*). The reliance on appearance and motion features allows for a number of incorrect matches to appear similar and thus confusing.

In this paper, we propose to improve the accuracy of tracklet matching by filtering the false candidates with similar features. Typically, tracking is broken into multiple tracklets due to missing detections. In most cases, missing detections occur because of occlusions where detection become unreliable. These periods of occlusion provide useful information for data association techniques. Tracklets entering and exiting the same occlusion must be matched to each other. Thus, we segment the tracklets into smaller groups based on the occlusions. Thus, rather than running tracklet matching for all possible matches, we run tracklet matching for each occlusion tunnel independently. This approach reduces the size of the matching problem and, more importantly, filters out incorrect, potentially confusing matches, thus improving tracking accuracy as shown in Figures 1 and 2. In the next section, we give a brief review of the data association based tracking framework [3]. In Section 3, we describe how we incorporate occlusion tunnels into the data association framework. Next, we evaluate our method on four 5,000 frame videos in Section 4. We conclude and discuss future work in Section 5.

2. Data Association based Tracking

Here we briefly describe the MAP approximated data association based tracking method of [3] which our method extends. This method is broken into four steps 1) tracklet building where an initial set of short confident tracklets are created, 2) tracklet matching where tracklets are iteratively linked together to form longer trajectories, 3) tracking convergence based matching to match tracklets over long temporal gaps, and 4) a final gap filling step to estimate missing detections within the completed trackings.

Tracklet Building. Given a set of detections in all frames, $\mathcal{D}$, an initial set of confident tracklets is built by linking detections of adjacent frames. For each pair of adjacent frames, an association cost matrix measuring the similarity between two detections, $\mathbf{S}$, is computed where rows correspond to detections in frame f and columns to detections in frame $f+1$. $\mathbf{S}(i,j)$ is computed as the product of three similarity measures based on size, location, and appearance. Pairs who's similarity exceeds a minimum threshold, $\mathbf{S}(i,j) > \theta_1$, and is also θ_2 greater than any other value in row i and column j in $\mathbf{S}$ are linked together to form *tracklets*, T. This heuristic ensures that a set of relatively short, but highly confident tracklets is created. Any isolated detections not associated to any other are considered as single detection tracklets as the removal of false alarms is more efficiently handled during the tracklet matching stage.

Tracklet Matching. Given an initial set of tracklets, $\mathcal{T}^0$, from the tracklet building step, tracklets are iteratively linked together. At each stage k, the output from the previous stage $\mathcal{T}^{k-1}$ becomes the input for the next stage of matching k where links of up to τ_k frames apart are considered. Each tracklet, $T_i^{k-1} \in \mathcal{T}^{k-1}$, is (1) matched to another tracklet, (2) labeled as the start or end of an object trajectory, or (3) classified as a false alarm and ignored from further matching. This matching process is formulated as a MAP problem by [6, 8] and solved using the Hungarian Algorithm. For the affinity score (or link cost), we use the HybridBoost method of [8] and use the set of features described in [3].

Tracking Convergence based Matching and Gap Filling The third stephandles solving fragments over large temporal gaps and filling in missing detections within trajectories. Matching over large temporal distances ($>$ 500 frames) becomes unreliable as predicting motion becomes increasingly difficult and appearance models are less likely to be unique with more ants falling within the predicted motion path. The convergence of particle-filter based tracking [5] was used to match tracklets with large gap lengths as well as fill in the positions during the gaps. Two tracklets, T_i and T_j, are matched if the forward tracking from the tail (or end) of T_i converges on the head (or start) of T_j and the backward tracking from the head of T_j converges on the tail of T_i.

Forward and backward tracking is also performed within completed trackings' gaps to estimate missing detections. The positions during gaps are obtained by first looking for the frame, f_c, where the forward and backward trackings are closest spatially. The positions up to frame f_c of the forward tracking and positions after frame f_c of the backward tracking are kept as detections within the gap. During the forward and backward trackings, foreground pixels associated with other objects, found during the tracklet matching step, are removed to reduce the chance of drifting.

Ant Detection We first detect foreground blobs by using same method described in (Section 3). Blobs which meet an area (460-1000 pixels) and eccentricity (2.5-6.5) constraint are considered as ant detections, $\mathcal{D}$. Secondly, we exploit the fact that ants tend to remain still for long periods of time by searching for still ants. For each ant, a duration of no motion is found by examining the difference in a SIFT feature taken at the first frame with subsequent frames at the same location. Once the Euclidean distance between the original feature and that of a subsequent frame exceeds a threshold, the ant is designated as moving. The region in the first frame is manually initialized and duplicated for the duration of no motion to create a tracklet. This process is repeated from the last frame, searching backwards in time. To ensure only one detection per ant, we remove any detections in D which overlap with the no motion ant detections. The still ant tracklets are added to the initial set $\mathcal{T}^0$ after the tracklet building step.

3. Occlusion Tunnel

Overview. Tracking ants is challenging due to the high level of similarity in appearance and motion characteristics. Furthermore, when the occlusion duration is increased, the reliability of motion feature is further reduced resulting in either (1) matching to an incorrect object or (2) not matching to the correct object and terminating tracking early. We use occlusion to filter incorrect tracklets which may have similar features to improve tracklet matching accuracy. Our key assumption is that a set of tracklets (1) enter the occlusion, (2) travel through occlusion and (3) exits the occlusion. In other words, a tracklet entering the occlusion must match to a tracklet exiting the same occlusion. Our goal is not to track through occlusion, but simply group the tracklets based on the occlusion that they enter and exit and prevent matching to a tracklet outside the occlusion. We define this sequence of occlusion areas to be "occlusion tunnel." This is similar to the activity tube idea of [10] where an object is viewed as being able to travel through a series of foreground blobs through time. Note that we make the distinction between "detection", which refers to the detection of ant region, from "blobs" which are simply a group of connected foreground pixels. We assume that during occlusion, detection is unreliable and often missing yet blob extraction remains reliable.

Occlusion Tunnel Construction. Given a set of blobs from a sequence of images, we first construct a tunnel of all blobs as a directed graph, $F = (V, E)$, where $V = \{v_1, ..., v_p\}$ denotes the set of nodes and $E = \{e_1, ..., e_q\}$ denotes the set of edges. Each node, $v_i = \{t_i, p_i\}$ represents a blob containing a set of pixels p_i in frame t_i corresponding to one (thus no occlusion) or more (occluding) objects. Next, edges are added between two nodes that contain at least one same object. We estimate it based on proximity in space and time. Two nodes, v_i, v_j are considered temporally close if $0 < t_j - t_i \leq n$ where n is a temporal threshold (we set $n = 2$ in our experiments). Because the movement between consecutive frames is relatively small, two nodes are considered spatially close if $p_i \cap p_j \neq \emptyset$ where p_i, p_j are the set of pixels for v_i and v_j. For objects with larger motion, our simple spatial measure can be replaced by distance between more sophisticated features.

To find occlusion tunnels, we split F into a set of subgraphs by removing nodes that correspond to tracked objects. Any node $v_i \in F$ which satisfies $d_j^p \cap p_i \neq \emptyset$ and does not exceed an area threshold α is removed, where p_i is the set of connected component pixels associated with v_i, d_j is a detection in $\mathcal{T}$, and d_j^p is location of d_j. The area of a foreground node is an easy identifier of whether the foreground blob contains multiple objects. We set α to 1,000 pixels which is slightly larger than a typical object

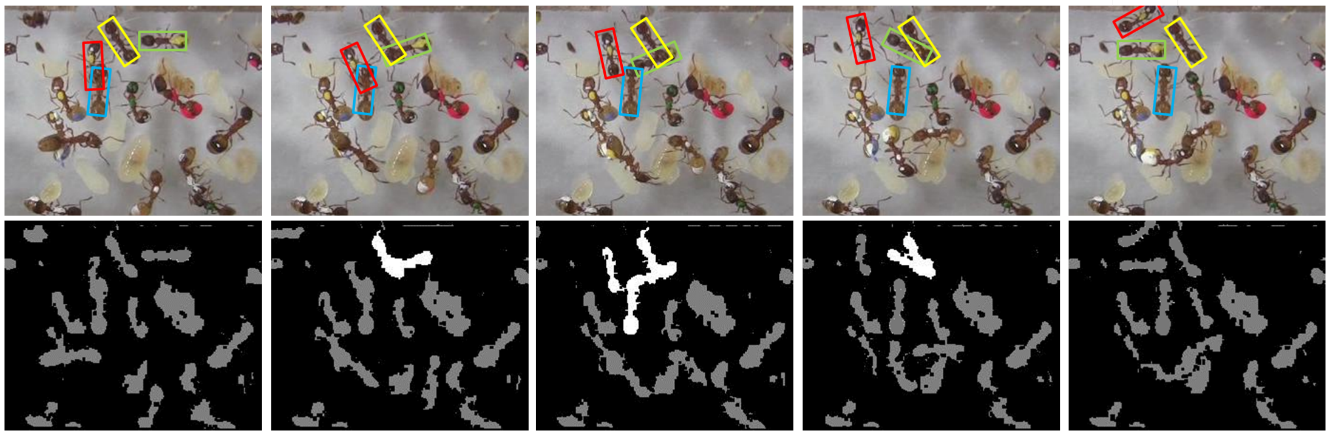
Figure 3. Occlusion Tunnel. Top row: A series of original images are shown with the ants involved with the occlusion tunnel highlighted by bounding boxes. Bottom row: The foreground blobs corresponding to the occlusion tunnel of those four ants are highlighted in white.

(ant) within our videos. After removing nodes associated with tracklets, we are left with the modified graph F'. Each disjoint subgraph within F' is an occlusion tunnel.

Using occlusion tunnel. Given a set of occlusion tunnels, we first map the ends of each tracklet to nodes in the unmodified graph F. The node associated to the end of each tracklet will have an edge to a node within F' and is seen as *entering* an occlusion tunnel. The node associated to the start of each tracklet will have an edge from a node in F' and is seen as *exiting* an occlusion tunnel. Tracklets entering an occlusion tunnel (a disjoint subgraph in F') can only match to tracklets which exit the same occlusion tunnel.

We can now extend our assumption on the state of each tracklet as stated in the tracklet matching portion of Section 2. The start (end) of each tracklet can now be viewed as 1) entering (exiting) the scene, 2) starting (ending) a longer trajectory, or 3) exiting (entering) an occlusion tunnel. Instead of satisfying the MAP equation for the entire set $\mathcal{T}$, we can now solve the MAP estimate independently for each set of tracklets $\{T_O\} \in \mathcal{T}$ *s.t.* each T_O enters or exists a specified occlusion tunnel.

Foreground blob detection. There are a number of methods for detecting blobs in images *e.g.*, background subtraction, color classification, min-cut segmentation, *etc*. For our dataset of ants, we use the color classification method of [5] which we also use to create our detections (described in section 2). We apply a morphological closing operator to clean the foreground responses and remove any connected component region less than an area threshold to remove noise responses. We set the area threshold to 50 pixels in our experiments as this is smaller than an individual ant section (*e.g.* the head) which is prone to being separated from the rest of the ant region, but large enough to remove pieces of debris. Note that V will contain blobs that corresponds to occlusions of multiple objects as well as isolated objects.

4. Results

4.1. Data

We evaluated our method on four 5,000 frame videos of *Temnothorax rugatulus* ants taken at thirty frames per second. Each video contains 30 to 50 ants within each frame. Some ants are painted to assist in identification, and most stay within the colony for the entirety of the video with only a few ants entering and leaving. Each video is fully ground truthed, and we use four-fold cross validation for training and testing.

4.2. Implementation Details

We use seven stages of association for our tracklet matching method. The maximum allowed gap lengths at each stage of tracklet matching, τ_k, are set to 8, 16, 32, 64, 128, 256, and 512. For each stage, we train a strong rank classifier H, which contains 100 weak rank classifiers, to compute the link cost following [8]. We set the combination coefficient for HybridBoost to $\eta = 0.75$ and the threshold for the strong ranker output to $\zeta = 0$ [8, 3]. We follow [6] in setting the true positive and false positive costs based on the performance of our detector, and the termination and initialization costs based on the maximum allowed frame gap and miss detection rate of our detector. Any remaining variables are learned empirically from the data or learned by the algorithm.

4.3. Evaluation Metrics

We use the same evaluation metrics as [6, 8, 3] which consist of the following metrics:

Fragments (Frag) The total number of times a ground truth trajectory is interrupted by tracking results.

Matching Stage	1	2	3	4	5	6	7	All
Confusing Matches	185	15	16	8	22	18	26	290
Filtered by Occlusion Tunnel	6	2	0	1	6	4	9	28
Percent Filtered	3.2%	13.3%	0.0%	12.5%	27.3%	22.2%	34.6%	9.7%

Table 1. A confusing match is an incorrect match which has a better affinity score than the tracklet's correct match. These are likely to cause ID switches within the tracklet matching step. Note, the highest filtering was observed in the stage which corresponds to the largest allowed gap size.

	Frag	IDS	MT	PT	ML
Tracklet matching	50.5	11.0	18.5	15.3	7.8
Tracklet matching with OT	46.8	7.8	18.5	15.3	8.0

Table 2. The average tracklet matching accuracy on four 5,000 frame videos.

	Frag	IDS	MT	PT	ML
[3]	47.3	32.0	30.8	8.0	2.8
[3] with OT	26.0	14.0	32.3	6.0	3.3

Table 3. The average tracking performance on four 5,000 frame videos after tracking convergence based matching and gap filling.

ID Switches (IDS) The total number of times a tracked trajectory chants its matched ground truth identity.

Mostly Tracked (MT) The number of ground truth trajectories which are successfully tracked for more than 80%.

Partially Tracked (PT) The number of ground truth trajectories which are successfully tracked between 20% and 80%.

Mostly Lost (ML) The number of ground truth trajectories which are tracked for less than 20%.

4.4. Tracking Accuracy

First, we assess the improvement in accuracy of tracklet matching (Section 2) when occlusion tunnel is used. Refer to Table 2. The addition of occlusion tunnels to filter incorrect potential matches reduced the number of ID switches at the tracklet matching step by 30% and reduced fragments by 7%. As expected the number of mostly tracked, partially tracked and mostly lost did not change much as these measures are assessed at detection level.

Second, we assess the improvement in the tracking performance after tracking convergence based matching and gap filling when occlusion tunnel is used (refer to Table 3.) Note that tracklet matching with tracking convergence based matching and gap filling is [3]. Addition of occlusion tunnel to [3] reduced ID switches by 56% and fragments by 44%. Note that the final two steps of the tracking framework (tracking convergence based matching and gap filling) rely on the tracklets for initialization and termination points. ID switches created during the tracklet matching step will often cause additional errors within these two later steps (Table 3). During tracking convergence based matching, a tracklet's correct match may already be taken causing the tracker to drift onto another nearby ant, while during the gap filling stage, the foreground pixels of tracked ants are removed causing the tracker to drift. This is shown in the final results of [3], which resulted in double the ID switches created (and therefore an increased number of fragments) almost entirely from drifting during gap filling.

Third, we assess the accuracy of filtering matches to tracklets of incorrect ants by using occlusion tunnel (refer to Table 1). For each matching stage with a varying τ_k, we counted the number of incorrect tracklet matches with affinity scores higher than the correct match in all four testing videos. Note that this is most likely to result in ID switches and possibly fragments. This number is listed as "confusing matches" in Table 1. Then, we counted the number of the confusing matches that were filtered by the occlusion tunnel. We noted that the occlusion tunnel was able to filter total of 28 confusing matches. The highest number of filtering occurred in the largest gap size ($\tau_k = 512$) in stage 7. This is especially interesting as the tracklet matching becomes more challenging as gap size grows. Finally, an example of ID switch reduction using occlusion tunnel is visualized in Figure 4.

5. Conclusion & Future Work

We have proposed the use of occlusion tunnels to filter incorrect matching during tracklet matching and improve the performance of data association based multiple ant tracking. The addition of occlusion tunnels reduced ID switches by 30% and fragments by 7% during tracklet matching. Most importantly, it was able to correctly filter 9% of the confusing matches, or matches which had a better affinity score than the correct match for a given tracklet and were likely to cause an ID switch. Because the final steps of the tracking framework (tracking convergence based matching and gap filling) rely on the results of the tracklet matching round, the improvement in tracklet matching with occlusion tunnel resulted in the reduction of ID switches by 56% and fragments by 44%. For future work, we plan

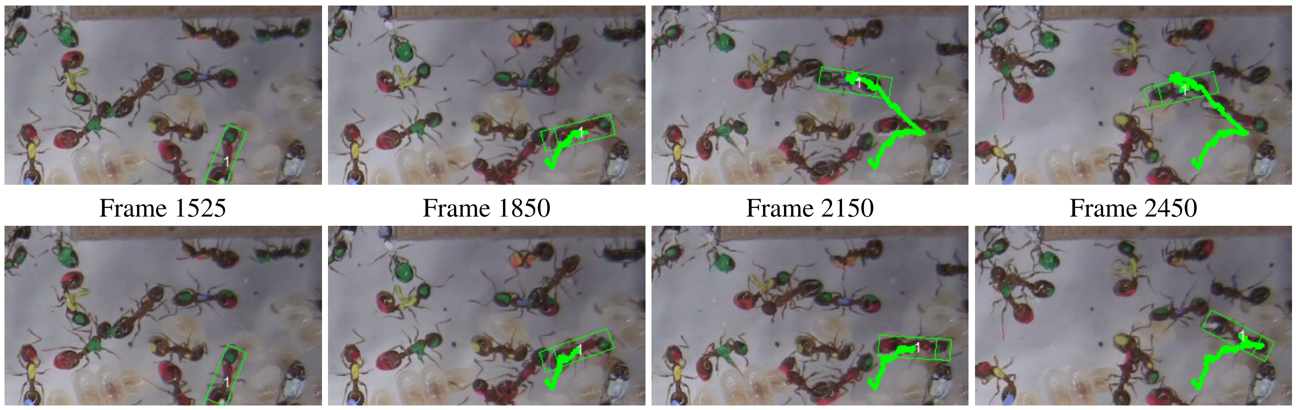

Figure 4. Reduction of ID switch using the occlusion tunnel. The top row shows an instance of an ID switch to a nearby, similar ants (highlighted in frame 2450) without occlusion tunnel. The bottom row shows reliable tracking with occlusion tunnel.

to improve the robustness of creating occlusion tunnels to work with more occlusion-robust detection methods. We also plan to extend the role of occlusion tunnels within the tracklet matching framework beyond the filtering step.

References

[1] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *CVPR 2011*. IEEE, 2011. 1

[2] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1806–1819, 2011. 1

[3] T. Fasciano, H. Nguyen, A. Dornhaus, and M. Shin. Tracking multiple ants in a colony. In *WACV, 2013*, pages 534–540, 2013. 2, 3, 4, 5

[4] J. H. Fewell. Social Insect Networks. *Science*, 301(5641):1867–1870, Sept. 2003. 1

[5] M. Fletcher, A. Dornhaus, and M. C. Shin. Multiple Ant Tracking with Global Foreground Maximization and Variable Target Proposal Distribution. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 570–576. IEEE, 2011. 1, 3, 4

[6] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, pages 788–801. Springer-Verlag, 2008. 2, 3, 4

[7] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, 27(11):1805–1819, 2005. 1

[8] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR 2009*, pages 2953–2960. IEEE, 2009. 3, 4

[9] D. P. Mersch, A. Crespi, and L. Keller. Tracking individuals shows spatial fidelity is a key regulator of ant social organization. *Science*, 2013. 1

[10] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg. Webcam synopsis: Peeking around the world. In *Computer Vision, 2007 IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 3

[11] D. Salvi, J. W. Waggoner, A. Temlyakov, and S. Wang. A graph-based algorithm for multi-target tracking with occlusion. In *WACV*, pages 489–496, 2013. 1

[12] T. D. Seeley. *Honeybee democracy*. Princeton Univ. Press, 2010. 1

[13] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *CVPR 2012*. IEEE, 2012. 2

[14] A. Veeraraghavan, R. Chellappa, and M. Srinivasan. Shape-and-Behavior Encoded Tracking of Bee Dances. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(3):463–476, 2008. 1

[15] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *CVPR 2012*, pages 1918–1925. IEEE, 2012. 2

[16] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008, IEEE Conference on*, pages 1–8. IEEE, 2008. 2