The bird gets caught by the WORM: tracking multiple deformable objects in noisy environments using Weight ORdered logic Maps

Debajyoti Karmaker[1] [0000−0003−0020−5391], Ingo Schiffner[2] [0000−0003−0044−1677], Michael Wilson[1] [0000−0002−3115−9319], and Mandyam V. Srinivasan[1,3] [0000−0002−2875−4985]

[1] The Queensland Brain Institute, University of Queensland QLD, Australia
[2] Bangor University, Gwynedd, Wales, UK
i.schiffner@bangor.ac.uk
[3] The School of Information Technology and Electrical Engineering, University of Queensland, Australia
{d.karmaker,m.wilson,m.srinivasan}@uq.edu.au

Abstract. Object detection and tracking are active and important research areas in computer vision as well as neuroscience. Of particular interest is the detection and tracking of small, poorly lit, deformable objects in the presence of sensor noise, and large changes in background and foreground illumination. Such conditions are frequently encountered when an animal moves in its natural environment, or in an experimental arena. The problems are exacerbated with the use of high-speed video cameras as the exposure time for high-speed cameras is limited by the frame rate, which limits the SNR. In this paper we present a set of simple algorithms for detecting and tracking multiple, small, poorly lit, deformable objects in environments that feature drastic changes in background and foreground illumination, and poor signal-to-noise ratios. These novel algorithms are shown to exhibit better performance than currently available state-of-the art algorithms.

Keywords: object detection · Data association · object tracking · shortest paths

1 Introduction

The use of Unmanned Aerial Vehicles (UAVs) is becoming more and more pervasive every year, rendering current ground-based control systems inadequate in avoiding mid-air collisions. Most airborne creatures, such as birds or flying insects, are extremely adept at avoiding collisions with their conspecifics or other moving objects in their environments. How they achieve this is largely unknown. In the past decade, we have started to gain insights into how birds and insects control their flight speed [22], avoid obstacles [27] and perform smooth landings [25]. This has drawn considerable attention from roboticists, who are challenged with similar problems in the design of guidance systems for unmanned aerial vehicles. However, studying bird flight is not simple: firstly one has to be able to collect accurate data from flying animals, with sufficient temporal and spatial resolution. This is best achieved using high-speed cameras and stereo or multi-camera setups. While high-speed cameras are now becoming more and more affordable, algorithms for accurately tracking moving animals in video sequences are scarce; and tracking flying animals, in particular birds, is by no means a trivial task. Flying birds are one of the more challenging "objects" to detect, partly because of their constantly changing shape [8, 9]. Tracking multiple birds using high-speed cameras, and dealing with frequent occlusions, is an even more challenging task. However, it is also the most feasible way of measuring the birds' motion states and undertaking a quantitative study of their flight behavior. Conversely, the development of tracking systems that can handle such a challenging task may provide robust techniques for tracking other, less challenging objects.

Robust object tracking is one of the most ubiquitous problems in computer vision science. It has numerous applications, including automatic visual surveillance, traffic monitoring, vehicle navigation, and motion-based object recognition [32]. Despite the continued growth of interest in automated object tracking, the problem continues to be extremely challenging due to factors such as variation of illumination, deformation of non-rigid objects, occlusions, complex motion, and the presence of background clutter. Tracking techniques can be broadly categorized into two approaches [28]: (1) tracking by detection and (2) temporal tracking. In the first approach, objects are detected by spatially processing the image to detect specific points or key features of the object [16, 33, 34], and then subtracting the background [5, 13, 15, 18, 37, 38], using various techniques to segment the object from the background [11, 23]. In recent years, this approach has been bolstered by using supervised learning [1, 14, 35], implemented in support vector machines and neural networks [31, 14]. The second approach makes use of the temporal information that is available in a video sequence to enhance the accuracy and speed of detection.

In both approaches, the object states from the previous frame and the current frame are usually passed on to either a Kalman filter (KF) [4, 10, 36] or a particle filter (PF) [3, 21] to obtain the best estimate of the object's location in the current frame. However, the KF technique is a recursive process which performs poorly when dealing with complex trajectories [28], and is only suitable for linear state models with Gaussian noise. On the other hand, the PF technique models the process stochastically and can successfully tackle nonlinear and non-Gaussian tracking problems, but is computationally expensive.

Tracking a single object involves locating and tracking the position of the object from frame to frame in the camera's field of view. Tracking multiple birds is more challenging, as it requires the simultaneous prediction of the movement trajectory of each object, as well as correctly identifying each object in each frame. Multiple object tracking (MOT) can be classified into two categories: Offline MOT and Online MOT. Offline MOT takes the entire video as an input, detects the object in each frame, and then constructs the most likely trajectory of each object by associating the data collected over the entire video sequence.

Online MOT, on the other hand, can only use information from the video frames that have been acquired up to the current point in time, to predict the location of each object in the current frame.

Most multi-object tracking algorithms are based on the use of an 'appearance' model, specified either in the spatial domain or the frequency domain, to detect the objects in each frame. When dealing with multiple, similar looking objects (e.g. the birds in our case) an appearance model alone is not sufficient to determine the identities of the individual objects accurately and reliably, because the objects (e.g. birds) can have similar sizes and movement patterns, and can frequently occlude one another in the image. Another major issue with tracking live objects is that most tracking algorithms rely on the assumption that the temporal changes are smooth from frame to frame i.e. objects do not change their position significantly between frames. A further problem with tracking multiple objects is that the number of objects can vary from one frame to the next, for example, when one object leaves the camera's field of view, or another object enters it. Thus, a robust MOT system needs to tackle all of the above challenges.

In this paper, we present a technique for detecting and tracking multiple moving objects in environments that feature drastic changes in background and foreground illumination. During long term tracking, moving objects can display abrupt changes in motion, and also be subject to partial or complete occlusion. When birds fly, they produce certain patterns of motion. These patterns often display special spatial properties – for example, their flight path may be restricted to certain regions of the experimental environment, or two birds may strive to achieve a minimum separation when they fly past each other. These structural and motion constraints can be exploited to construct data associations that help preserve and disambiguate the identities of individual objects.

2 Related works

Most contemporary MOT systems rely on a detection framework for tracking [24], instead of using background subtraction to overcome problems such as cluttered and dynamic backgrounds. In fact, when it comes to tracking one kind of object (e.g. human, bird or car), tracking by detection is more suitable, because it avoids object fragmentation. However, detection frameworks have difficulty with long-term tracking of small flying creatures such as birds or insects from a stationary camera when the object moves well away from the focal plane of the camera. The object then becomes blurry and detection fails due to the inability to track individual features of the object. Moreover, birds are inherently difficult to detect because they are highly articulated, and constitute an extreme example of a nonrigid, deformable object. To tackle this problem, we take a step back and investigate how the background subtraction method can be applied in complex scenarios with dynamic backgrounds and drastic illumination changes.

In motion analysis, background modeling and subtraction play a key role. The concept behind this approach is to construct a probabilistic representation of the scene based on the dynamically changing background over time, which is used to perform the subtraction from the current input image to extract the object. One way to model the static background is to perform image averaging over a certain number of frames, which gives a reasonable portrayal of the mean background image. Ridder [19] proposed a Kalman-filter based approach to model the background, whereas Wren [30] made use of a Gaussian distribution to model the background – which turns out to be too simplistic for modeling real-world scenarios. Stauffer and Grimson [26] proposed a Mixture of Gaussian (MoG) approach to better model multiple background intensity distributions that are generated, for example, by ripples on water surface surfaces, and flickering scene illuminations. However, this requires a decision on how many Gaussians to use. To avoid this difficulty inherent in parametric models Elgamal et al. [7] proposed a non-parametric model that describes background density by extraction of a histogram of pixel intensities at each pixel location. An intensity value that falls outside the range of the histogram at a particular location is taken to belong to the object. Both parametric and non-parametric methods [7, 12, 13, 17] work efficiently only for environments with gradually evolving changes and small variations. They decline in performance when dealing with dynamic background scenes such ocean waves, rain, waving trees, moving clouds, or illumination changes [18].

To overcome the problems with traditional background subtraction methods for describing dynamic scenes, Weng [29] proposed another variation of background subtraction – known as the local dependency histogram (LDH). In this technique, each pixel is modeled as a group of weighted LDHs. Next, the process of labeling the pixel as the foreground or background is performed by comparing the new LDH computed in the current frame against its LDH model. Finally, a fixed threshold value is used to define whether or not a pixel matches its model. Another prominent technique for background subtraction is kernel density estimation [6] to handle scenarios where the background is not completely static but consists of small motion and illumination changes. For each pixel, this technique builds a histogram of background values by accumulating a set of real values sampled from the pixel's recent history. Then, they estimate the probability density function with this histogram to determine whether or not a pixel value of the current frame belongs to the background.

In contrast to all of the above background modeling techniques, which can only deal with slow changes and require constant updating, our method is capable of successfully dealing with a variety of rapid background changes without the need for model updating. Moreover, as will be shown below, this detection method – based on the decomposition of a logical map – has reduced complexity and computational expense. It outperforms all of the background modeling techniques described above and allows more time to be allocated to the data association problem – namely, the problem of connecting the detected locations in a sequence of frames to create an unambiguous track. Approaches that use Kalman filtering make assumptions of constant velocity or acceleration, which are not viable when dealing with objects that are constantly changing their shapes, speeds, and flight directions. As a consequence, Kalman filtering approaches suffer from a greater likelihood of ID swaps when tracking multiple objects in comparison with our data association approach, which exhibits significantly better performance.

3 Proposed method

In this section, we describe our multiple object tracking method that is based on object detection. Our object detection method is based on computing the inter-frame difference, extraction of a series of intensity levels from the inter-frame difference image, and determining the centroid for each intensity level. The determination of the centroids is highly affected by the level of noise present in the original images that are used to generate the inter-frame difference.

In Section 3.1, we describe our object detection method. In Section 3.2, we demonstrate how the object locations determined in the sequence of video frames are connected to produce an accurate and unambiguous track(s).

3.1 Object Detection

The method presented here for object detection uses frame-to-frame image differences to detect objects in environments that feature drastic changes in background and foreground illumination, as well as noise (an example video is available at https://www.youtube.com/watch?v=9IE8-82agYc). Unlike traditional background subtraction methods, initial results are not stored in the commonly used format of a logic map. Instead, image frame differences are decomposed into multiple logic maps according to their level of intensity difference. After this step, separate centers of gravity are formed for each individual logic map. Due to this separation into individual maps, regular changes in the background illumination and static noise will automatically be distinguished from the actual target. Using these centers of gravity, one can simply look for a set of n nearest neighbors (pixels around these centers of gravity) and compute their shared center of gravity to define the actual position of the target. As changes in background illumination, unless they are extremely regular, will be represented by centers of gravity that can theoretically lie anywhere on the image, the number of nearest neighbours necessary to successfully track an object is solely dependent on the level of noise present in the video footage. Due to the random nature of noise, most of the centers of gravity will most likely lie in locations very different from that of the image of the object, and noise will, therefore, be effectively suppressed.

The traditional approach of detecting a moving object is to compute the inter-frame difference – that is, to subtract the intensity values of pixels in the current frame (CF) from those in the previous frame (PF). A pixel in the difference image (FD) is considered to belong to the moving object if its value is higher than a prescribed threshold (θ) – assuming that intensity variations in the pixels that constitute the background are small, and below this threshold. If the value of the pixel is lower than the threshold, it is considered to belong to the stationary background. This operation can be expressed as

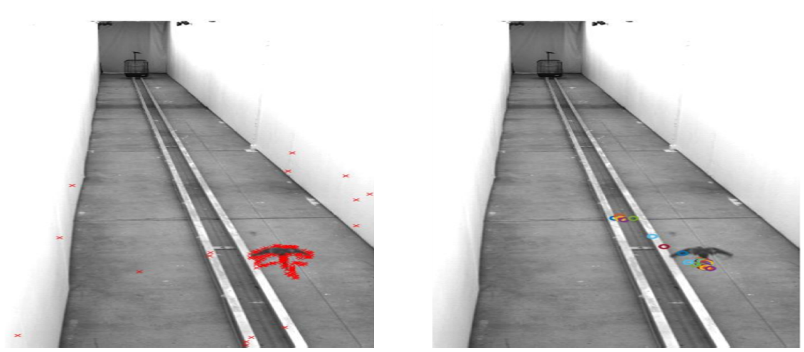
$$FD\,(x, y) = |CF\,(x, y) - PF(x, y)|$$

From this we derive a logic map which defines the motion-associated pixels and stationary background pixels according to

$$logic\ map(x, y) \begin{cases} 1, & \text{FD(x,y)} \geq \theta \text{ Motion region} \\ 0, & \text{FD(x,y)} < \theta \end{cases} \tag{1}$$

In our approach, we analyze the logic map at multiple intensity levels in the difference image according to the intensity difference k by using k intensity difference maps:

$$logic\ map_k(x, y) \begin{cases} 1, & \text{FD(x,y) == k} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$



(a) (b)

Fig. 1: (a) Logic map for k=20 on raw frames 79 & 78 (b) Centroids of different logic maps on raw frames 79 & 78.

Next, the centroid is computed for each of these logic maps. The motion of complex objects (in our case birds) will result in a large number of logic maps. Using the centroids of these maps, we look for a set of n nearest neighbors and select their shared center of gravity to best estimate the true position of the moving object. Figure 1,a shows an example of the logic map for k=20 and Figure 1,b shows the centroids for all logic maps.

For detecting multiple objects, after decomposing the logic map at multiple levels, we apply K-means clustering to find the center of gravity of each object. Here in K-means clustering, the value of K is set to the maximum number of objects expected to be present in the video footage. Even if K objects are not present in a particular frame of the video, we can distinguish the false detections by applying a constraint on the maximum number of logic maps (max value of k).

3.2 Data Association

The next challenge is to track the trajectory of the objects reliably, without interchanging their identities. We pose the data association problem as one of finding the shortest path in a graph. Thus, we represent the detection hypotheses from all the frames as a directed acyclic graph (DAG). This graph $G = (V, E)$ consist of a set of nodes (V), which are hypothesized detections, and a set of edges (E), which connect pairs of detections (nodes) and associate each connection with a cost value. The cost of an edge between two nodes is represented as a sum of three sub-costs: (a) appearance dissimilarity, (b) internode separation and (c) change in direction of movement in relation to the previous node pair. Let v and w be nodes in successive frames connected by an edge e. The appearance dissimilarity between v and w is computed by $(1 - v_b.w_b)$ where $v_b.w_b$ denotes the dot product of the detection bounding boxes of v and w respectively. (The greater the overlap between the bounding boxes, the lower the dissimilarity). The cost associated with the change of direction is computed using $\left(\tan^{-1}\frac{v_{b_y}}{v_{b_x}} - \tan^{-1}\frac{w_{b_y}}{w_{b_x}}\right)$ by finding the centroids $(v_b(x,y)\ and\ w_b(x,y))$ of the bounding boxes of $v_b and\ w_b$. Assigning edge weights in this manner will produce low weights when two successive nodes represent the same object, and large weights when they represent different objects. We also ensure that every edge E points forward in time (as shown in Figure 2), i.e., the frame number of the detection v is strictly smaller than that of the detection w.

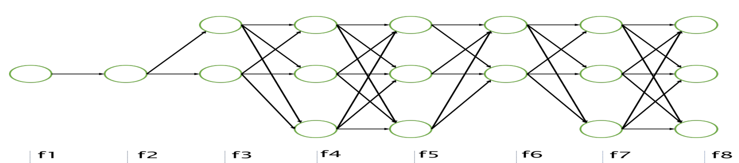


Fig. 2: Initial graph representation of all possible trajectories

Figure 2 shows an example of an initial representation of a graph that represents all of the possible trajectories that can be associated with the detected nodes. The nodes are organized in an ascending order with respect to the frame number. Each node in a frame is connected to every node in the following frame with an edge cost as described above. To tackle situations where nodes are missing, we apply a constraint on the maximum number of frames over which an object can be absent. If the object does not reappear within that range, it is considered to have gone out of the camera's visual field. If it returns within this critical period, it is assumed that the object was temporarily occluded, and a dummy node is added in the location where it was last detected, as shown in Figure 3. To maintain these criteria, we need to scan the graph from left to right and ensure that at any level of the graph, the number of incoming edges is equal to or lower than the number of outgoing edges. If there are more outgoing edges, this implies that a new object has arrived.

The maximum edge cost for a bird from one frame to next frame is estimated from the ground truth of our dataset. Based on this maximum edge value, we remove all the edges between two nodes - where the cost is greater than the estimated max edge value. This helps to reduce search complexity.

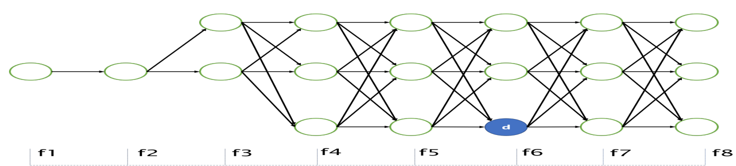


Fig. 3: Dummy node insertion

We then apply Dijkstra's algorithm to find the shortest path for the graph of Figure 3. Classically, Dijkstra's algorithm works on a single source and a single destination. In our case, no source or destination is defined. Furthermore, if there are multiple objects, we need to determine a different trajectory for each of the objects. For this reason, we represent the graph as a time sequence and consider any node from f1 as the source node. See Figure 4, orange path. For the destination we do not set any node, rather we continue until the algorithm reaches a node at the last level, or finds a dead end.

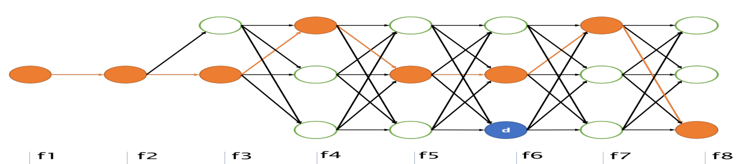


Fig. 4: Selecting the 1st trajectory

Next, we delete all the nodes and every incoming and outgoing edge from these nodes. The remaining graph is shown in Figure 5. We then select any node which does not have any incoming edge and repeat the process to successively determine the trajectories of each of the other moving objects in the video sequence.

The time complexity of heap implementation of Dijkstra's algorithm is $O(nlogn)$, where n is the number of nodes. As it is guaranteed that our graph is not dense i.e. not all pairs of nodes are connected, the worst-case time complexity of our

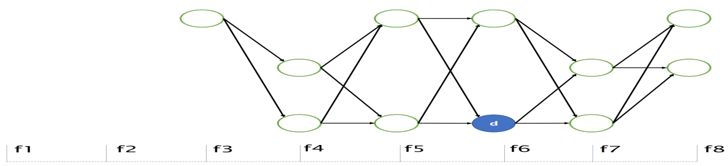


Fig. 5: Removal of nodes and edges

implementation is no more than $O(nlogn)$ for detection of a single trajectory. However, detection of multiple trajectories requires less time in reality. The reason is evident from the fact that we keep removing the nodes from the original graph every time a trajectory is detected. This technique progressively reduces the graph size. Our experiments with a real-world dataset illustrate that our technique is able to provide results in query time (<1 s) on a standard machine configuration (8 GB RAM, Ubuntu 16, Intel Core-i7 CPU with 3.40 GHz clock speed).

4 Experimental Results

We have compared the method of detection and tracking with four other popular methods available. For this comparison, the ground truth was manually generated for bird locations (bounding boxes) in 10 videos. The same video dataset was used to test all 5 methods. Detection was considered to be correct if successive bounding-boxes overlapped by more than 50%, while tracking multiple objects was deemed incorrect due to misidentification if one object ID was assigned to any other object in any other frame during the entire video. The Kalman-filter based blob detection and tracking routine provided in the MATLAB computer vision toolbox performs well for detection but suffers from ID swap problems. The data association technique using a network flow algorithm [20] performs really well in minimizing ID swaps, but both [20] and [2] perform poorly in relation to object detection. This is due to the nature of our dataset, where the illumination is poor, and more importantly, whenever the bird flies beyond the camera focus area, it becomes blurry and any feature extraction algorithms such as HoG or Deep learning will be compromised. Table 1 reveals that the proposed method exhibits the best performance with respect to detection accuracy, and the second-best performance with respect to ID swaps.

5 Conclusion

The contributions of this work focus on two distinct aspects of the problem of detection and tracking multiple, deformable moving objects. These are (i) detection using interframe differencing and (ii) trajectory generation using a shortest path algorithm. The main advantage with weight ordered logic map decomposition of our detection method is that it does not need any background model to be

Table 1: Comparison with other methods

	Detection Accuracy	ID Swaps
Our method	93%	8%
Blob detection & Kalman Filter-based tracking (Matlab)	86%	37%
Background subtraction & FIR filter [5]	79%	19%
Deep Learning & Network Flow [20]	67%	3%
HoG & KLT [2]	65%	24%

updated over time. Thus, it is likely to be computationally much faster than the standard background modeling techniques. Approaches that use Kalman filtering make assumptions of constant velocity or acceleration, which are not viable when dealing with flying birds that are avoiding collisions – they are constantly changing their shapes, speeds, and flight directions. As a consequence, Kalman filtering approaches suffer from a greater likelihood of ID swaps when tracking multiple objects in comparison with our approach, which shows significantly better performance.

Acknowledgment

This work was co-funded by the Australian Research Council and by Boeing Research & Technology Australia through a Linkage Project Grant (LP 130100483), and an ARC Distinguished Outstanding Researcher Award (DP140100914).

References

1. Avidan, S.: Support Vector Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(8), 1064–1072 (2004)
2. Benfold, B., Reid, I.: Stable multi-target tracking in real-time surveillance video. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 3457–3464 (2011)
3. Cao, X., Gao, C., Lan, J., Yuan, Y., Yan, P.: Ego motion guided particle filter for vehicle tracking in airborne videos. Neurocomputing **124**, 168–177 (2014)
4. Chan, Y.T., Hu, A.G.C., Plant, J.B.: A Kalman Filter Based Tracking Scheme with Input Estimation. IEEE Transactions on Aerospace and Electronic Systems **AES-15**(2), 237–244 (1979)
5. Choi, I.H., Pak, J.M., Ahn, C.K., Lee, S.H., Lim, M.T., Song, M.K.: Arbitration algorithm of FIR filter and optical flow based on ANFIS for visual object tracking. Measurement: Journal of the International Measurement Confederation **75**(July), 338–353 (2015)
6. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(5), 1–37 (2002)
7. Elgammal, Ahmed and Harwood, David and Davis, L.: Non-parametric model for background subtraction. In: Computer VisionECCV 2000. pp. 751—-767. Springer (2000)

8. Everingham, M., Eslami, S.M.A., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: Assessing the Significance of Performance Differences on the PASCAL VOC Challenges via Bootstrapping **1** (2013)
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence **32**(9), 1627–1645 (2010)
10. Fu, Z., Han, Y.: Centroid weighted Kalman filter for visual object tracking. Measurement: Journal of the International Measurement Confederation **45**(4), 650–655 (2012)
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014)
12. Harville, M.: A Framework for High-Level Feedback to Adaptive, Per-Pixel, Mixture-of-Gaussian Background Models. Framework **3**, 543–560 (2002)
13. Javed, O., Shafique, K., Shah, M.: A hierarchical approach to robust background subtraction using color and gradient information. Proceedings - Workshop on Motion and Video Computing, MOTION 2002 pp. 22–27 (2002)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. Advances In Neural Information Processing Systems pp. 1–9 (2012)
15. Lee, D.S.: Effective Gaussian mixture learning for video background subtraction. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(5), 827–832 (2005)
16. Lowe, D.G.: Distinctive Image Features from. International Journal of Computer Vision **60**(2), 91–110 (2004)
17. Mittal, A., Huttenlocher, D.: Scene modeling for wide area surveillance and image synthesis. Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662) **2**, 160–167
18. Monnet, A., Mittal, A., Paragios, N., Ramesh, V.R.V.: Background modeling and subtraction of dynamic scenes. Proceedings Ninth IEEE International Conference on Computer Vision (Iccv), 1305–1312 vol.2 (2003)
19. Ridder, C., Munkelt, O., Kirchner, H.: Adaptive background estimation and foreground detection using kalman-filtering. Proceedings of International Conference on recent Advances in Mechatronics pp. 193—-199 (1995)
20. Roshan Zamir, A., Dehghan, A., Shah, M.: GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **7573 LNCS**(PART 2), 343–356 (2012)
21. Sardari, F., Ebrahimi Moghaddam, M.: A hybrid occlusion free object tracking method using particle filter and modified galaxy based search meta-heuristic algorithm. Applied Soft Computing Journal **50**, 280–299 (2017)
22. Schiffner, I., Srinivasan, M.V.: Direct Evidence for Vision-based Control of Flight Speed in Budgerigars. Scientific Reports **5**, 10992 (2015). https://doi.org/10.1038/srep10992
23. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **22**(March), 888–905 (2005)
24. Solera, F., Calderara, S., Cucchiara, R.: Learning to divide and conquer for online multi-target tracking. Proceedings of the IEEE International Conference on Computer Vision **11-18-Dece**, 4373–4381 (2016)

25. Srinivasan, M.V.: Visual control of navigation in insects and its relevance for robotics. Current Opinion in Neurobiology **21**(4), 535–543 (2011)
26. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Cat No PR00149 **2**(c), 246–252 (1999)
27. Vo, H.D., Schiffner, I., Srinivasan, M.V.: Anticipatory manoeuvres in bird flight. Sci Rep **6**, 27591 (Jun 2016), 27270506[pmid]
28. Watada, J., Musa, Z., Jain, L.C., Fulcher, J.: Human tracking: A state-of-art survey. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **6277 LNAI**(PART 2), 454–463 (2010)
29. Weng, J., Zhang, Y., Hwang, W.S.: Candid covariance-free incremental principal component analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(8), 1034–1040 (2003)
30. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: real-time tracking of the human body. IEEE Trans. on Pattern Analysis and Machine Intelligence **19**(7), 780–785 (1997)
31. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(9), 1834–1848 (2015)
32. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A Survey. ACM Computing Surveys **38**(4), 13–es (2006)
33. Yokoi, K.: Probabilistic BPRRC: Robust change detection against illumination changes and background movements. IEICE Transactions on Information and Systems **E93-D**(7), 1700–1707 (2010)
34. Yokoyama, M., Poggio, T.: A Contour-Based Moving Object Detection and Tracking. 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (1), 271–276 (2005)
35. Zhang, S., Sui, Y., Yu, X., Zhao, S., Zhang, L.: Hybrid support vector machines for robust object tracking. Pattern Recognition **48**(8), 2474–2488 (2015)
36. Zhong, X., Zhong, X., Peng, X.: Robots visual servo control with features constraint employing Kalman-neural-network filtering scheme. Neurocomputing **151**(P1), 268–277 (2015)
37. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. **2**(2), 28–31 (2004)
38. Zivkovic, Z., Heijden, F.V.D.: Efficient adaptive density estimation per image pixel for the task of background subtraction (2006)