JOURNAL OF THE ROYAL SOCIETY Interface

O fly, where art thou?

Dhruv Grover, John Tower and Simon Tavaré

J. R. Soc. Interface 2008 **5**, 1181-1191
doi: 10.1098/rsif.2007.1333

JOURNAL OF THE ROYAL SOCIETY Interface

J. R. Soc. Interface (2008) **5**, 1181–1191
doi:10.1098/rsif.2007.1333
Published online 26 March 2008

O fly, where art thou?

Dhruv Grover[1], John Tower[1] and Simon Tavaré[1,2,*]

[1] *Molecular and Computational Biology Program, Department of Biological Sciences, University of Southern California, Los Angeles, CA 90089-2910, USA*
[2] *Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge CB3 0WA, UK*

In this paper, the design of a real-time image acquisition system for tracking the movement of *Drosophila* in three-dimensional space is presented. The system uses three calibrated and synchronized cameras to detect multiple flies and integrates the detected fly silhouettes to construct the three-dimensional visual hull models of each fly. We used an extended Kalman filter to estimate the state of each fly, given past positions from the reconstructed fly visual hulls. The results show that our approach constructs the three-dimensional visual hull of each fly from the detected image silhouettes and robustly tracks them at real-time rates. The system is suitable for a more detailed analysis of fly behaviour.

Keywords: real-time three-dimensional tracking; *Drosophila* activity monitoring; visual hull construction; extended Kalman filtering

1. INTRODUCTION

Trajectory modelling is an effective way to understand the behaviour of many dynamical systems (Turchin 1998; Gruen & Akca 2005; James 2007). In animals, trajectories can be used to explain complex behaviour patterns such as migration, foraging, territorial aggression and mating. An intuitive way to generate trajectories is to use sensors to detect and track animals over a period of time. Over years, many approaches have been used to track animals of different shapes and scales. Large animals have been fitted with sensors giving us telemetry data about their locations (Preisler *et al.* 2004). Stereo photography has been used on static images to reconstruct positions of birds in a flock (Ballerini *et al.* 2008). Algorithms were also developed to study the movement and schooling behaviour of fishes (Parrish & Turchin 1997). At the insect level, ants and bees have been tracked with cameras to provide valuable data for developing multi-agent modelling tools (Balch *et al.* 2001; Feldman & Balch 2004). Tracking algorithms have even been applied for monitoring the location of drug molecules inside cells, since understanding this is important for the development of new drugs (Murphy 2004).

As a model organism, the fruit fly *Drosophila melanogaster* plays a vital role in our understanding of biological processes. Flies are interacting insects, exhibiting a multitude of behaviours such as grooming, flight, foraging, fighting, mating and egg laying. Therefore, understanding the behaviour of a group of insects is an important but challenging problem: they are fast moving and are virtually indistinguishable. Many attempts to solve this problem have been made in recent years by either watching for wing movements of tethered flies (Graetzel *et al.* 2006) or analysing trajectories of free flying flies. Most approaches are limited to filming a single fly (Tammero & Dickinson 2002), which is not suitable for explaining complex behavioural traits such as courtship. Among the approaches that track multiple insects, a large number of them project their three-dimensional tracks onto one plane. Some approaches use a single camera to film the insects, prohibiting the three-dimensional reconstruction due to lack of depth information.

In order to overcome these limitations, we have developed a method for detecting and tracking fly movement using three calibrated and synchronized cameras. Our method produces the real-time three-dimensional tracks of a group of flies suitable for further analysis of flight behaviour. We synchronize the multiple camera views to minimize the effects of occlusions and improve the estimation of three-dimensional reconstruction. As a test, we used the fly tracking system proposed in this paper to study the behaviour of flies in a real biological experiment. In that study, it was found that hydrogen peroxide feeding and conditional expression of superoxide dismutase transgenes dramatically altered specific fly behaviours that were not possible to detect without using our system (Brown *et al.* submitted).

Our paper is broken down into the following sections. Section 1.1 presents a brief survey of related literature on vision-based tracking of insects and visual hull reconstruction. Section 1.2 provides an overview of the proposed system. Section 2 introduces the algorithm to construct a visual hull of the fly body. Section 3 reviews the extended Kalman filter (EKF) approach to state estimation and its application to tracking flies.

*Author and address for correspondence: Molecular and Computational Biology Program, Department of Biological Sciences, University of Southern California, Los Angeles, CA 90089-2910, USA (stavare@usc.edu).

Received 11 December 2007
Accepted 22 February 2008

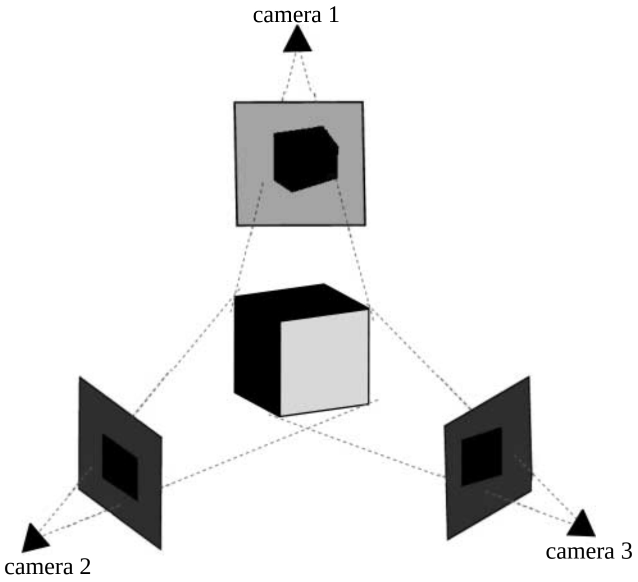

Figure 1. Three-dimensional visual hull from two-dimensional silhouettes.

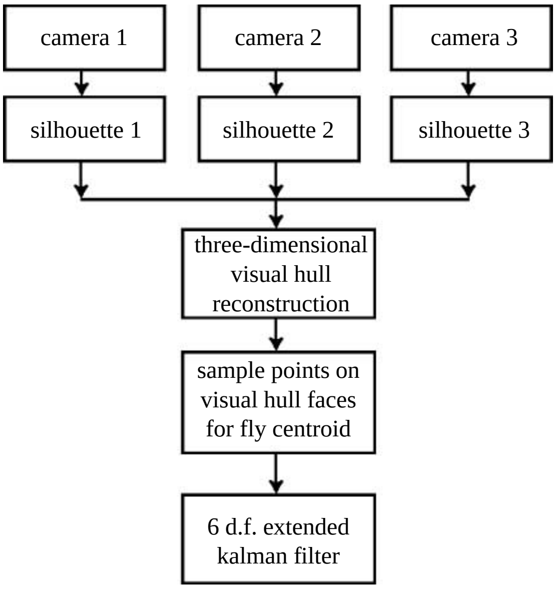
camera 1
camera 2
camera 3
silhouette 1
silhouette 2
silhouette 3
three-dimensional visual hull reconstruction
sample points on visual hull faces for fly centroid
6 d.f. extended kalman filter

Figure 2. Outline of the fly tracking system.

Section 4 describes the experimental set-up in detail. Section 5 demonstrates our visual hull construction and rendering algorithms in a real-time fly tracking system. Section 6 provides a conclusion and describes some future directions of this work.

1.1. Previous work

Analysing and tracking insect motion and behaviour have been active areas of research for the last few years. Many algorithms originally intended for tracking people have been extended to insects. These methods are often not reliable since insects are fast moving and are virtually indistinguishable. Also, these methods are based on non-interacting objects, which make them unsuitable for tracking and analysing the behaviour of interacting insects such as flies. The classic algorithms in this class of non-interacting target tracking include the nearest neighbour approaches (Deriche & Faugeras 1990; Parrish & Turchin 1997), Bayesian multiple hypothesis tracker (Cox & Leonard 1994) and data association methods such as the joint probabilistic data association filter (Rasmussen & Hager 2001). Among the approaches suitable for tracking multiple insects, the analysis has been carried out by either tracking or limiting the insect movement to a two-dimensional space. A combination of colour- and motion-based methods was used to track ants in the two-dimensional space (Balch *et al.* 2001). Their system was susceptible to errors owing to occlusion, clumping and motionless ants. A system to track and analyse the behaviour of honeybees using human trainable models was also proposed (Feldman & Balch 2004). These methods are video based, and are therefore not capable of distinguishing between models of behaviour, which are too similar. The accuracy of this behaviour recognition method is highly dependent on the size of the training set. Some tracking methods limit filming to a single fly, which is not suitable for explaining complex behavioural traits such as courtship (Graetzel *et al.* 2006).

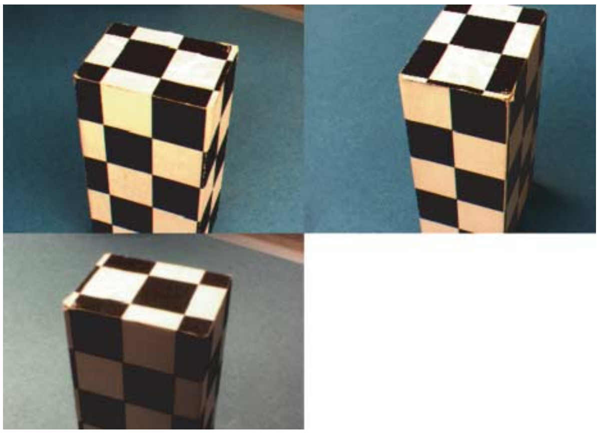

Figure 3. Three-dimensional chessboard used for camera calibration.

Simulated models of fish schooling were used to study aggregate behaviour, since it was difficult to generate tracking information of groups of fishes for more than a few seconds (Parrish *et al.* 2002). When tracking information was available, schooling behaviours could only be identified based on their paths (Parrish & Turchin 1997). In order to overcome these limitations, we used a visual hull approach to construct the three-dimensional models of each fly. This model provides a more accurate description of the fly and is better suited for identifying the patterns of behaviour than the video-based methods mentioned above.

The visual hull concept introduced by Laurentini (1994) falls under the classification of shape from silhouette methods (Baumgart 1974). It describes the maximal three-dimensional geometric model constructed from all possible object silhouettes (figure 1). Since it is not possible to extract all possible silhouettes of an object, the visual hull is computed using a finite number of them. As the number of silhouettes increases, so does the rendered quality of the reconstructed object model. Optimal viewpoints to take silhouette images for the three-dimensional shape reconstruction were discussed by Shanmukh & Pujari

(1991). Visual hulls are most commonly reconstructed by projection of silhouettes in a three-dimensional grid of volume elements or voxels (Potmesil 1987; Szeliski 1993). An octree data structure was proposed to speed up the construction of the visual hull model (Potmesil 1987). A method using splines instead of polygonal meshes was proposed to improve the shape of the model (Sullivan & Ponce 1998). Another method for constructing the visual hull models using voxels was proposed, but the processing was done off-line (Moezzi *et al.* 1996). An algorithm constructing an exact polyhedral representation of the visual hull was presented by Matusik *et al.* (2001), but exhibited performance unsuitable for real-time tracking of *Drosophila*.

There is a wealth of literature on the use of visual hulls for recognizing different behaviours in humans and its advantages over traditional video-based methods (Mikic *et al.* 2001; Werghi & Yijun 2002; Cohen & Li 2003; Chu & Cohen 2005). These methods construct the visual hulls and infer human postures and gestures using different underlying statistical models. They do however have some shortcomings that make them unsuitable for use in a real-time fly tracking system like ours. These methods construct the visual hull of only a single object and infer behaviours from that. They also offer no tracking algorithms since the single object is present in the view of all cameras at all times. Also, they do not run at real-time rates and cannot infer behaviours due to interactions (e.g. courtship and mating in flies). Our system overcomes all of the above-mentioned shortcomings by constructing the visual hulls of multiple flies and tracking them even during occlusions over a period of time at real-time rates of 60 frames s^{-1}. Our approach therefore not only provides three-dimensional path information for multiple flies but also is better suited for identifying complex behaviours comprising physical motions of the flies, for example wing extension or arcing of the body, or egg laying.

1.2. Proposed system

The first step in vision-based tracking is to detect and separate moving objects in images using background subtraction techniques (Stauffer & Grimson 1999; Khan & Shah 2000; Xu & Ellis 2001). A three-dimensional model of the object can then be reconstructed from the set of two-dimensional image silhouettes. This classical approach to three-dimensional reconstruction is known as the shape from silhouette method. Popular approaches to three-dimensional model construction from image silhouettes use volumetric techniques that often produce visual artefacts in the three-dimensional model (Potmesil 1987; Szeliski 1993; Dyer 2001). This is a significant drawback when low-resolution approximations of the object are needed for real-time applications such as fly tracking. There are other approaches to constructing the three-dimensional models that are view dependent. One of the popular methods in this class, view ray sampling, constructs a three-dimensional model from a discrete set of viewing rays (Matusik *et al.* 2000). In this paper, we have developed algorithms for rendering polyhedral visual hulls of the flies in real time. This representation has significant advantages over other three-dimensional reconstruction methods. It is view independent and needs to be computed only once for a given set of input silhouettes. This enables us to freeze a particular frame and change the viewing angle to better study the three-dimensional fly model for any clues of its behaviour. It can be constructed even when flies are occluded in an individual camera view, which is a common occurrence as they cross paths. (Note that a minimum of two views are required to construct the visual hull.) It can be computed and rendered quickly on current graphics hardware, and is ideal for real-time applications such as fly tracking.

Once the three-dimensional object model is generated, it is then possible to track it in explicit three-dimensional coordinates. Vision-based object tracking typically involves an iterative method to estimate the state of the moving object from the past state measurements. We use a 6 d.f. EKF for state estimation to track the flies in the three-dimensional space. An outline of our system is illustrated in figure 2.

2. GENERATING THE THREE-DIMENSIONAL FLY MODEL FROM VIDEO

The major purpose of this work is to develop an algorithm for detecting flies from camera images and tracking them over time, enabling us to understand their movement behaviour. Tracking the flies in the three-dimensional space required reconstructing the three-dimensional models of each fly in the vial. Three calibrated and synchronized cameras were used to capture fly image silhouettes to reconstruct the three-dimensional model. The cameras were calibrated using a set of 20 feature points (see §2.1). This method of reconstructing the three-dimensional models from the two-dimensional silhouettes is known as the visual hull reconstruction. The two-dimensional fly image silhouettes were detected using a Gaussian background subtraction technique (Wren *et al.* 1997). These silhouettes were used to compute the visual hull of the fly. The following is a detailed description of the steps for generating the three-dimensional fly model.

2.1. Camera calibration

To estimate the epipolar geometry between the three cameras, we calibrated them using a set of 20 feature points on a multi-planar chessboard (figure 3). Each square grid in the chessboard has a length of 1 cm. The method proposed by Tsai (1986) was used for the calibration process. Projection matrices mapping image coordinates to three-dimensional coordinates and vice versa were generated for each camera. In order to test the accuracy of our mapping, we reprojected three-dimensional positions of test points on the chessboard to image (silhouette) coordinates using the projection matrices and compared them with the true two-dimensional image coordinates. The difference in pixels between the two points in the x- and y-directions is shown in the reprojection error plot (figure 4). Standard deviations in the x- and y-directions were found to be 0.12403 and 0.11649 pixels, respectively.

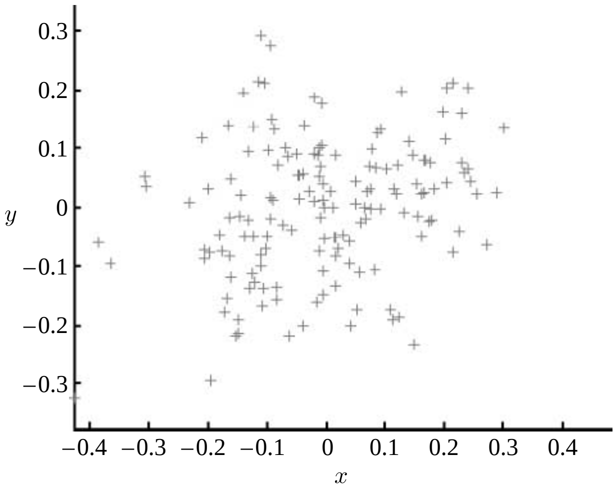

Figure 4. Reprojection error plot (units in pixels) for test points used in calibrating the three cameras.

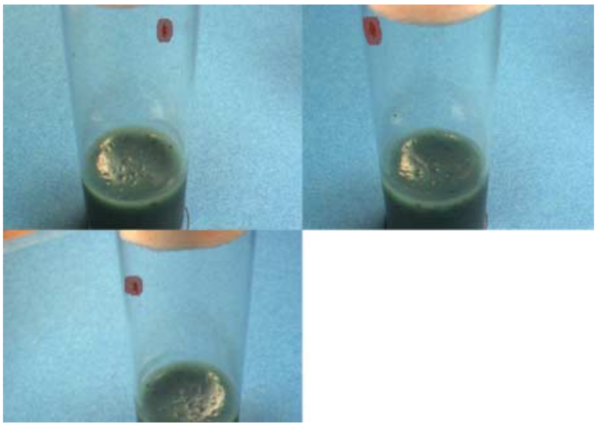

Figure 5. Silhouette detection of a single fly in the vial. The green boundary is the detected edge of the fly, light red region outside the green boundary is the shadow region eliminated from the final dark red silhouette.

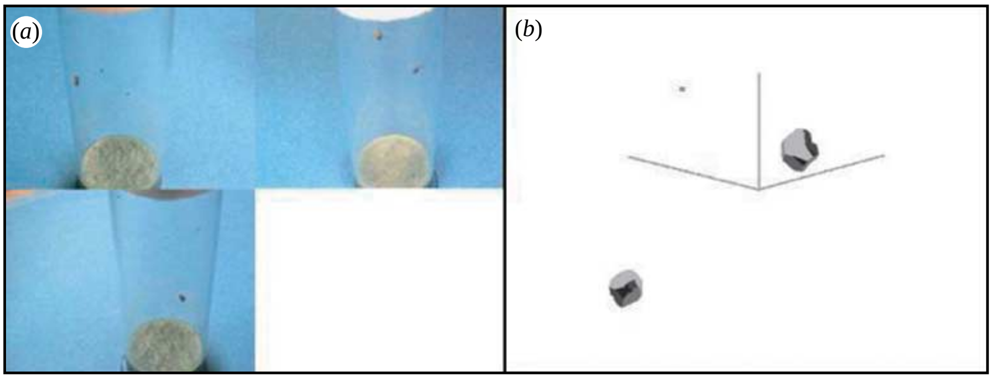

Figure 6. (*a*) Flies as seen by the three cameras and (*b*) the three-dimensional visual hull reconstruction of the two flies in the vial. The visual hulls are constructed even when the flies are not visible in each camera view. *Note*. A minimum of two views are necessary for visual hull construction.

2.2. Silhouette detection

The first step to building the visual hull is to find the flies in the images. This was done using a background subtraction technique where intensity values and variance of each pixel in the background image was calculated and fit to a Gaussian model. Flies could then be identified since their pixel intensities exceeded a threshold level based on the model fit from the background (Wren *et al.* 1997). The drawback of this method is that it detected flies along with their shadows, since they were cast on the background and segmented out. For a shadow pixel, the colour difference between the foreground and background pixels is small, since the difference lies mainly in the intensity values. Therefore, by comparing the intensity difference of the foreground and background pixels, we can eliminate shadows. Also, since our experiments were performed indoors, shadows had blurred edges. These facts were used to eliminate shadow regions, giving us accurate fly image silhouettes (Chu & Cohen 2005). The error probabilities of pixels in the image being classified as silhouette or non-silhouette pixels are mentioned in §6. These silhouettes computed by the real-time background subtraction algorithm were used for reconstructing the visual hull models. The algorithm is easy to implement and fast enough for extracting silhouettes at 60 frames s^{-1}, which is the frame rate at which our cameras capture images. Figure 5 shows a demonstration of our silhouette detection algorithm on a single fly.

Algorithm 1

```
Three-dimensional fly visual hull reconstruction
(i) Compute the silhouette cones for each input two-
      dimensional polygonal fly silhouette s
(ii) Perform a pairwise intersection of each pair of
      silhouette cones and save the set of polygons
(iii) Intersect the polygon sets computed in step (ii) to find
      the faces of each fly visual hull
(iv) Merge these faces to form the three-dimensional
      geometric model known as the visual hull for each fly
```

2.3. Visual hull reconstruction

The three-dimensional shape of the fly can be approximated by reconstructing its visual hull (Laurentini 1994). In our system, the visual hull is generated using the set of two-dimensional silhouette images of the fly from different calibrated camera views. A minimum of two views is required to construct the fly visual hull. Our algorithm uses polygonal representations of the image silhouettes to compute the visual hull similar to Matusik *et al.* (2001). The set of these two-dimensional polygons representing each silhouette *s* consists of a

Table 1. Three-dimensional tracking accuracy of EKF on a randomly chosen sequence of 60 s from test data of 30 min.

actual	flies detected	correct tracks	per cent correct	fram-e rate
1	1	1	100	60
2	2	2	100	60
5	5	5	100	60
7	7	6	86	45
10	10	8	80	35
13	11	8	62	15
15	12	6	40	5

Algorithm 2

```
Tracking flies using the EKF
Initialization
(i) Initialize with state $\hat{X}_{i,0} = E[X_{i,0}]$
    $P_{i,0} = E(X_{i,0} - E[X_{i,0}])(X_{i,0} - E[X_{i,0}])^{\mathrm{T}}$
Prediction
(ii) Predict the current state $\tilde{X}_{i,k} = m(\hat{X}_{i,k-1})$
(iii) Compute the conditional error covariance
    $\tilde{P}_{i,k} = M_{i,k} P_{i,k-1} M_{i,k}^{\mathrm{T}} + Q_{i,k-1}$
Update
(iv) Compute the Kalman gain matrix
    $D_{i,k} = H_{i,k} \tilde{P}_{i,k} H_{i,k}^{\mathrm{T}} + R_{i,k}$
    $G_{i,k} = \tilde{P}_{i,k} H_{i,k}^{T} D_{i,k}^{-1}$
(v) Compute the measurement $\tilde{Y}_{i,k} = h(\tilde{X}_{i,k})$
(vi) Apply measurement correction to the state estimate
    $\hat{X}_{i,k} = \tilde{X}_{i,k} + G_{i,k}(Y_{i,k} - \tilde{Y}_{i,k})$
(vii) Apply measurement correction to the error
    covariance $P_{i,k} = (I - G_{i,k} H_{i,k}) \tilde{P}_{i,k}$
Recursion
(viii) Repeat steps (ii)–(vii) for next time points for each
    fly $i$
```

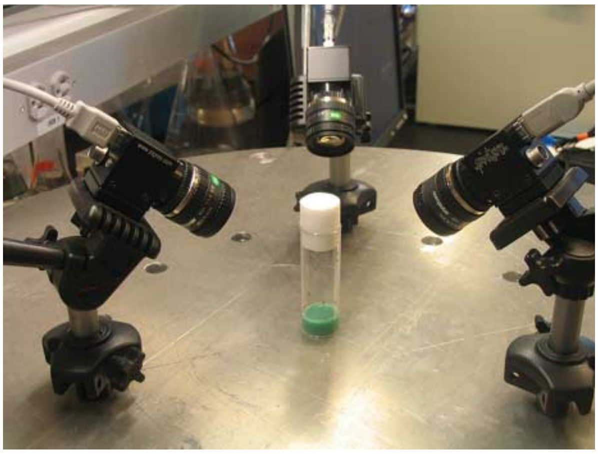

Figure 7. Experimental set-up of three cameras on the circular rig. Flies are placed in the vial in the centre.

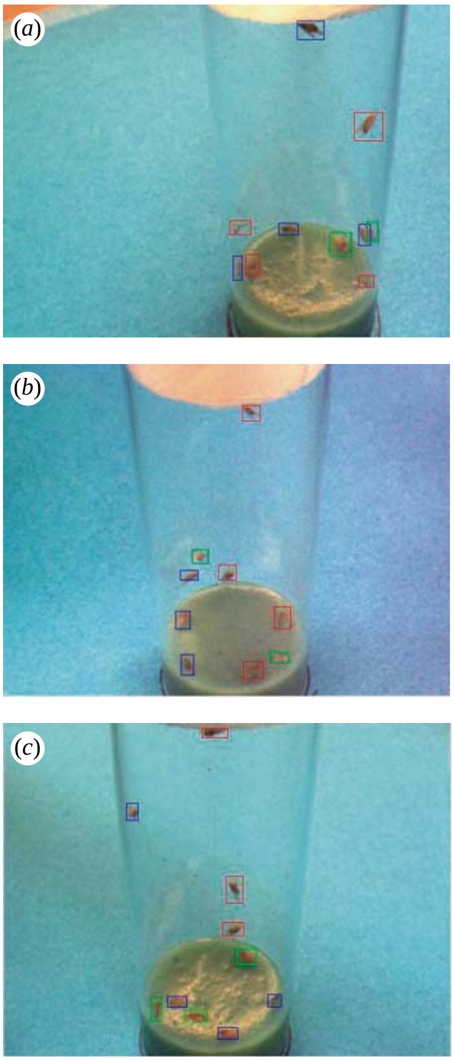

Figure 8. (*a*–*c*) Detection of fly silhouettes in two-dimensional camera images. This bounding box representation around the detected fly silhouettes shows the accuracy of the silhouette detection algorithm in distinguishing between flies when they are clumped together, as shown in (*a*).

set of edges joining consecutive vertices. Projection matrices mapping the coordinates of the two-dimensional image silhouette space to those of the three-dimensional is known for each camera. Given the polygonal representation of the image silhouettes and the associated camera projection matrices, the visual hull is computed by taking the intersection of the silhouette cones. This produces a three-dimensional polyhedral structure whose faces lie on those of the original silhouette cones. Algorithm 1 summarizes the steps involved in computing the visual hull.

This method produces polygonal meshes of the visual hull for each frame. Graphics hardware acceleration allows us to render these meshes at speeds fast enough to reconstruct all the flies in the vial in real time. Images of the flies from the three cameras and the reconstructed visual hull of the two flies in the vial are shown in figure 6.

3. TRACKING FLIES

The next step is to use an EKF (Julier *et al.* 1995) to track the flies, giving us spatial information for further analysis of fly movement.

3.1. Extended Kalman filtering

Tracking an object involves optimizing the state estimate from input measurements. Flies are small fast moving insects that invariably change direction while moving. They often cross paths and interact with each other. Predicting their motion is therefore a challenging problem and nonlinear methods are required to accomplish the task. For nonlinear dynamical systems, a variety of Bayesian techniques can be used to optimize the state estimate. Commonly used approaches include the EKF, unscented Kalman filter (Stenger *et al.* 2001*a*) and the particle filter (Stenger *et al.* 2001*b*). Since our detection algorithm detects flies at 60 frames s^{-1}, we can assume Gaussian error distributions; this enables us to use the EKF. However, increasing the number of flies beyond 10 (table 1) causes significant error since the measurement function is highly nonlinear. A discussion on increasing the number of flies while maintaining the tracking accuracy is provided in §6. We now give a brief overview of the EKF state estimation equations. For a complete derivation of the Kalman filter equations, refer to Maybeck (1979) and Welch & Bishop (1995). The EKF extends the linear assumption of the following basic Kalman filter equations to systems with a nonlinear measurement process:

$$X_{i,k} = m(X_{i,k-1}) + w_{i,k-1}, \tag{3.1}$$

$$Y_{i,k} = h(X_{i,k}) + v_{i,k}. \tag{3.2}$$

We describe the state-space model for estimating the state of each object i with the process equation (3.1) and the measurement equation (3.2). Here, X is the current state of the system and Y is the measurement of the system at time point k, m is the process model of the system, h is a nonlinear measurement model, and w and v are the process and measurement noise, respectively. The basic assumption of the EKF is that the process and measurement noise of the system should be independent, white and Gaussian with mean zero and covariance matrices Q and R, respectively. Since the state of each object $X_{i,k-1}$ is unknown, we use $\hat{X}_{i,k-1}$, the *a posteriori* estimate of the state at the previous time point $k-1$, to solve the process and measurement equations (3.1) and (3.2).

In order to use the EKF for state prediction of a nonlinear system, partial derivatives of the process model m and measurement model h need to be computed. This linearizes the system, and the current state can be estimated. The basic Kalman filter equations (3.1) and (3.2) therefore result in the following EKF process and measurement equations (3.3) and (3.4):

$$X_{i,k} = \tilde{X}_{i,k} + M_{i,k}(X_{i,k-1} - \hat{X}_{i,k-1}) + w_{i,k-1}, \tag{3.3}$$

$$Y_{i,k} = \tilde{Y}_{i,k} + H_{i,k}(X_{i,k} - \tilde{X}_{i,k}) + v_{i,k}. \tag{3.4}$$

Here, $\tilde{X}_{i,k}$ and $\tilde{Y}_{i,k}$ are the approximate state and measurement values $m(\hat{X}_{i,k-1})$ and $h(\tilde{X}_{i,k})$ for each object i, M and H are the partial derivative matrices of the process model m and measurement model h. Algorithm 2 summarizes the process of performing state estimation of multiple flies using the EKF.

3.2. *Tracking flies using EKF*

Using algorithm 2, we apply the EKF for the real-time three-dimensional tracking of multiple flies. The visual hull computed from multiple two-dimensional image silhouettes gives us the position of each fly in the three-dimensional space $[x, y, z]$ and orientation $[\alpha, \beta, \gamma]$, which forms the measurement Y of the system. In order to find the three-dimensional spatial position of the fly, we sample points on the surface of the visual hull. Each of these sampled points belongs to the same cluster or the visual hull and a simplified K-means clustering algorithm can be used to find the centroid (MacQueen 1967). The objective then is to estimate the state of each fly in every frame. In our system, the fly state includes the object's spatial three-dimensional position $[x, y, z]$, orientation $[\alpha, \beta, \gamma]$, translation $[x', y', z']$ and rotation $[\alpha', \beta', \gamma']$. Since we use 60 frames s^{-1} high-speed cameras, we can assume that the flies have constant velocity between consecutive frames. Thus, the state X of a fly is defined by a 6 d.f. representation denoted by $[x, y, z, \alpha, \beta, \gamma, x', y', z', \alpha', \beta', \gamma']$. The partial derivative matrices M and H of the process model m and measurement model h can be written as

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & t & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where t is the time between consecutive frames.

4. METHODS

4.1. *Experimental set-up*

All tracking experiments were performed with flies housed in standard 25×75 mm vials. The bottom end of the vial contained food stained with blue colour (Kroger brand), and the open end was closed with a cotton ball. The vial was placed in the centre of the circular camera rig 70 cm in diameter.The three cameras are positioned facing downwards to eliminate the possibility of background movement giving us false fly silhouettes (figure 7). The imaging of the tracking set-up consisted of three calibrated and synchronized Point Grey Flea digital cameras mounted on the camera rig at a distance of 15 cm from the vial. Each camera was fitted with an Edmund Optics 8 mm megapixel fixed focal lens. The cameras were connected with off-the-shelf FW800PCI-E cards to a computer running two Intel Dual-Core Xeon Processors (2.8 GHz per core) with 4 GB RAM and an NVIDIA Quadro 3450 PCI-E video card. The cameras were calibrated as described earlier (see §2.1). The tracking algorithm was implemented in MICROSOFT VISUAL C++ using OpenGL, OpenCV and VXL libraries, optimized for a multi-threaded environment. Details of the tracking set-up and source code for the visual hull reconstruction and tracking algorithms will be made available on request.

4.2. Video acquisition parameters

The 8 mm fixed focal lens had an aperture range of F1.4-16C and was set at F8 for all three cameras. The camera resolution was set at 640×480 to achieve 60 frames s^{-1}. Since flies are housed in vials, masking images are used to specify regions of interest for detection and tracking in each frame. The Point Grey MULTISYNC software is used to synchronize the image acquisition of multiple cameras across different 1394a and 1394b buses. This ensures a timing correlation between cameras on separate buses and preserves the frame rate.

5. EXPERIMENTAL RESULTS

Our system reconstructs the three-dimensional visual hulls of each fly at a maximum of 60 frames s^{-1}, which is the frame rate of our cameras. The actual frame rate of rendering the visual hull varies with the complexity of the object and the polygonal input silhouette profiles. To increase the frame rate of the visual hull reconstruction, the polygonal complexity can be lowered using a less refined input silhouette.

In figure 6, we demonstrate a rendering of a three-dimensional polyhedral visual hull fly that was captured in real time from our system. We can increase the accuracy of the reconstruction by sacrificing the frame rate. Since the purpose of generating the visual hull in our paper is to obtain three-dimensional trajectories of the flies, we can sacrifice on the quality of the rendering. A discussion on improving the quality of the rendered visual hull for behaviour recognition can be found in §6.

In figure 5, we show the silhouette detection of a single fly in the vial. The fly is visible from all three camera views. The green boundary indicates the detected edge of the fly. The light red region outside of the green border is the shadow region eliminated from the final dark red silhouette. To analyse more rigorously the silhouette detection method, we calculated error probabilities of pixels in the image being classified as silhouette or non-silhouette pixels. Since shadow pixels are removed from the final silhouette using methods described in §2.2, they fall under the category of non-silhouette pixels. The probability that a silhouette pixel was incorrectly marked as a non-silhouette pixel was estimated to be 0.028 and the probability of a non-silhouette pixel being marked as a silhouette pixel was estimated to be 0.011.

In figure 8, we present more results of silhouette detection of flies in the images captured by the three synchronized cameras. The bounding boxes around the detected fly silhouettes in the images show that the silhouette detection algorithm is able to distinguish between flies even when they are clumped together (figures 8*a* and 9).

In order to measure the accuracy of our fly detection algorithm, we conducted an experiment where we varied the number of flies in the vial at intervals of 30 s for 1 hour (Feldman & Balch 2004). We also recorded the fly detection algorithm's count (this is the number of visual hulls constructed) at each of these 30 s intervals. The comparison is shown in figure 10. The correlation between the number of flies detected by the algorithm and the number of flies actually present was 0.98 (figure 10*b*). For this experiment, we made sure that the flies were visible in at least two of the three camera views, necessary for visual hull construction. For situations where flies are occluded from two or more camera views, the EKF is used to estimate their centroid positions (results of which are discussed below). These results suggest that our detection algorithm is robust and efficient in distinguishing between multiple flies with partial occlusions and clumping.

In figure 11, a three-dimensional trajectory of a single fly tracked using the EKF is shown. Our tracking system generates similar real-time trajectories for multiple flies in the vial. We also show the two-dimensional trajectories as viewed from each camera in figure 9. The flies move at a fast pace, flying and hopping from one end of the vial to the other, frequently crossing paths. Thus, predicting the motion of these insects is a challenging problem. The EKF is used to make sure that each fly is tracked even during periods of occlusion, and that there are no gaps in fly trajectories at any time. Our tracking system generates trajectories of the moving flies in real time at a maximum speed of 60 frames s^{-1}, but performance is dependent on the number and grouping of flies in the vial (table 1). Extensions to the hardware and tracking algorithms to create better visual hulls and track more flies are discussed in §6.

Accuracy of our EKF tracking approach is demonstrated by comparing a simulated ground-truth path of an object with the EKF estimated path (figure 12). In order to validate our EKF tracking approach, we collected test datasets with varying numbers of flies (Khan *et al.* 2005). Each test dataset recorded a video of fly activity for 30 min at 60 frames s^{-1}. Then, 60 s intervals were chosen from each 30 min dataset. The starting point of each 60 s interval was chosen using a random number generator in R (R Development Core Team 2007) to eliminate any bias in our tracking accuracy results. We analysed the total number of flies detected and tracked using the EKF from the start to the end of the sequence without errors in each of the 60 s intervals. The tracking correctness was evaluated by comparing the EKF estimated trajectories against the actual paths of the flies similar to figure 12. Since we construct the visual hulls for each fly in every frame, we

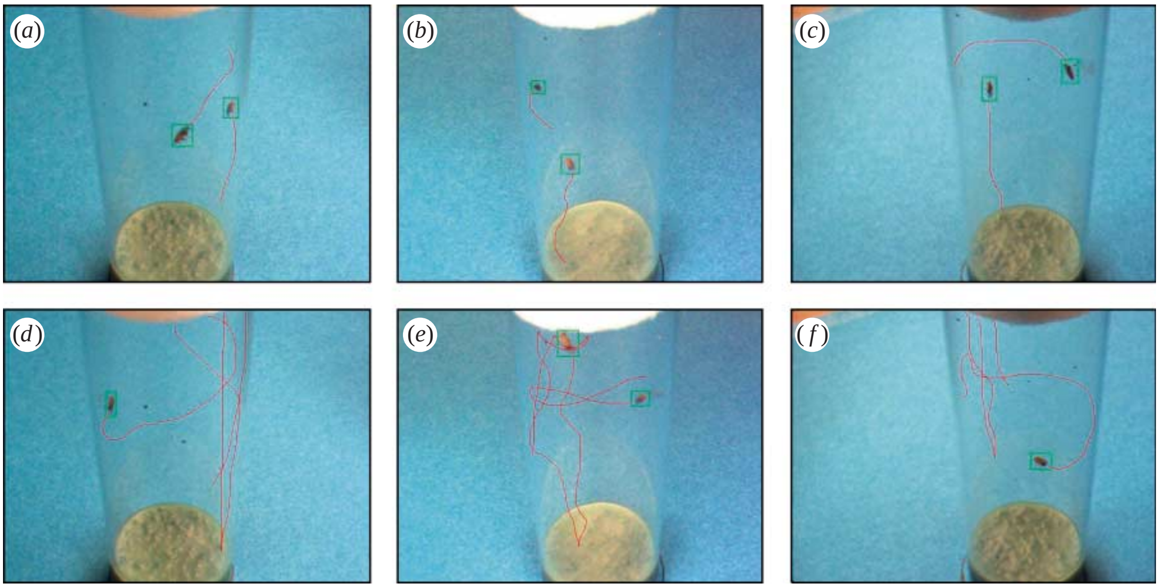

Figure 9. (*a*–*f*) Application of EKF to flies in the two-dimensional space as seen by the three cameras. (*d*–*f*) Our tracking algorithm keeps track of the flies even when they disappear from the view of the cameras.

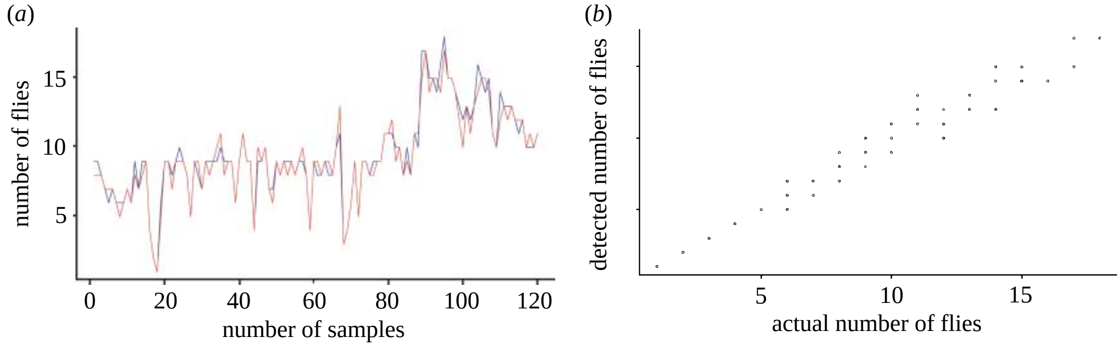

Figure 10. Accuracy of our fly detection algorithm. (*a*) The number of flies recognized by the detection algorithm (red) compared with the number of flies actually present (blue). This refers to the number of fly visual hulls constructed. (*b*) Correlation of 0.98 between the number of flies detected by the algorithm and the number of flies actually present. The evaluation was conducted over 1 hour, with samples taken every 30 s.

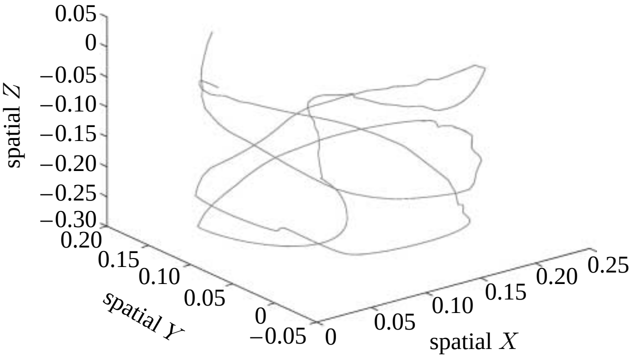

Figure 11. Trajectory of a single fly in the three-dimensional space. Our tracking software generates similar real-time trajectories for multiple flies in the vial.

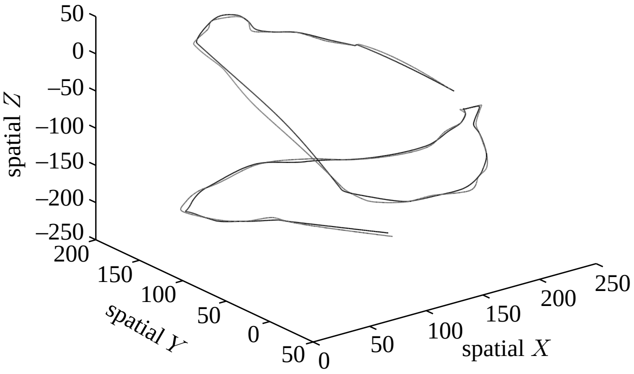

Figure 12. Accuracy of EKF tracking algorithm used in this study. Ground-truth object state (black line) versus EKF estimated state (grey line).

can reliably use the centroid of that three-dimensional fly rendering as the actual position of a fly in that frame. The actual path was therefore constructed by joining the centroid positions of a fly for each frame in the 60 s interval. For the cases where flies crossed paths or interacted with each other, to remove any uncertainty about where the fly moved in the next frame, we visually inspected the video sequence to identify the correct movement. The results of this experiment are summarized in table 1. We also present the tracking errors of the dataset consisting of two flies (figure 13). The flies in this dataset were tracked at 60 frames s^{-1}

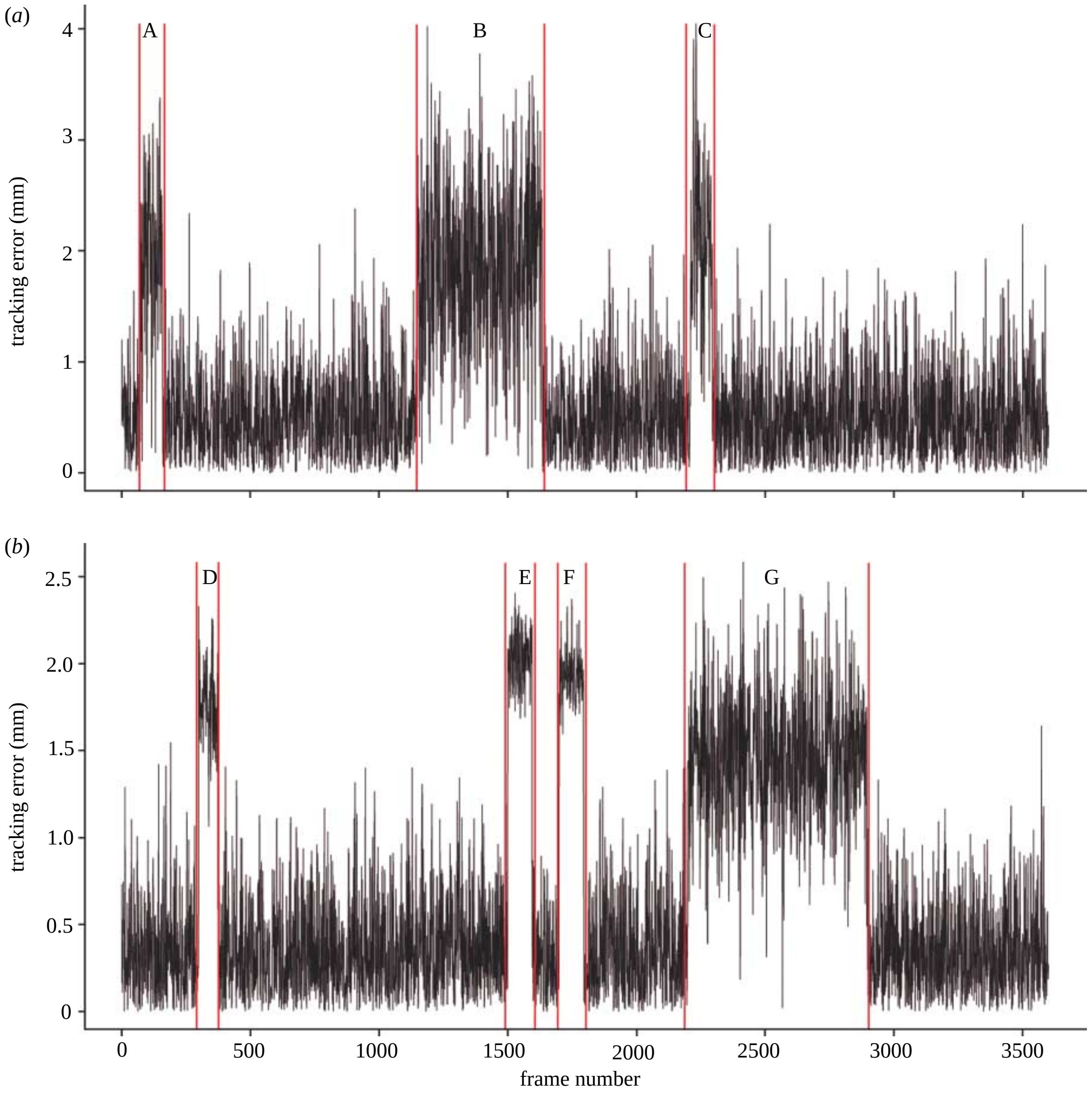

Figure 13. Tracking error of the dataset of two flies from table 1. Periods of occlusion in one or more cameras are marked with red bars. (*a*) Fly 1: occlusion A was in camera 3, B in camera 1, C in cameras 1 and 3. (*b*) Fly 2: occlusion D was in camera 1, E in cameras 1 and 2, F in cameras 1 and 2 and G in camera 1.

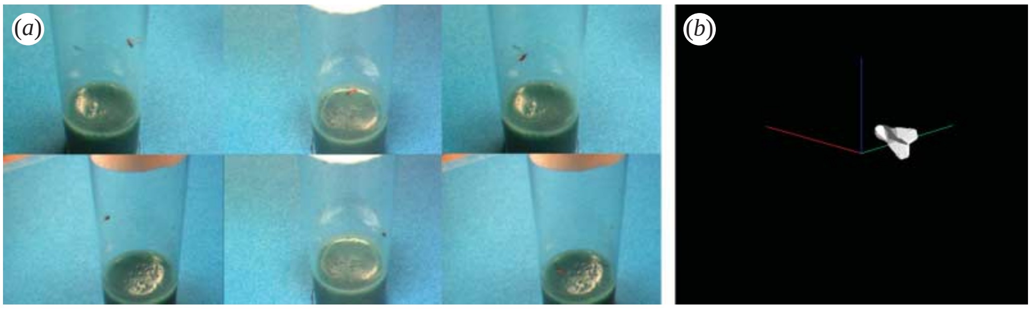

Figure 14. (*a*) Single fly as seen by six cameras and (*b*) the three-dimensional visual hull reconstruction of the fly.

for 60 s, generating 3600 frames (table 1). The tracking error refers to the difference between the actual centroid position and the EKF estimated position of the fly in each frame. The frames with higher tracking errors (between the red bars in figure 13) are those where the fly was occluded in one or more camera views. In figure 13*a* (fly 1) occlusion A was in camera 3, B in camera 1, C in cameras 1 and 3, and in figure 13*b* (fly 2) occlusion D in camera 1, E in cameras 1 and 2, F in cameras 1 and 2 and G in camera 1. Our approach is robust and efficient at tracking up to 10 flies simultaneously. The results show that as we increase the number of flies beyond 10, the tracking efficiency reduces. Also, the frame rate of our system falls below real-time rates, which is critical if we are to build on this system for behaviour analysis. A discussion of the steps that are taken to improve the tracking efficiency while increasing the fly numbers is provided in §6.

6. CONCLUSION

We have presented an approach for real-time tracking of multiple flies using three calibrated and synchronized cameras. The system constructs the three-dimensional visual hulls of each fly from the detected fly silhouettes. The reconstructed fly visual hulls are used by an EKF to generate *a posteriori* nonlinear state estimates of all the flies in the vial. The novelty of the system lies in the fact that we can not only track multiple flies in the three-dimensional space, but also use the reconstructed three-dimensional visual hulls to identify physical motions of flies, which in turn can be used to identify specific behaviours.

6.1. Future work

Several extensions of this work are currently being implemented. We are working on mixture Kalman filters (Chen & Liu 2000) to build on the EKF approach used currently. It is a sequential Monte Carlo method and is more effective in dealing with computational difficulties of nonlinear systems. It will enable us to maintain the tracking accuracy of the EKF while increasing the number of flies in the vial. We are also working on algorithms to further analyse the behaviour of flies. The underlying assumption here is that there is a direct correlation between fly movement and behaviour. We are therefore working on algorithms to match trajectories formed by multiple flies to better understand whether behaviour of flies varies under different conditions. Another extension to our work is to use the three-dimensional visual hulls to identify specific patterns of behaviour in flies. We do this by grouping together the sequence of motions (wing extension, body arcing, etc.) identified from the three-dimensional fly visual hull. For this purpose, we are expanding our hardware set-up to six cameras, giving us more accurate fly silhouette information that will undoubtedly improve the visual hull reconstruction and aid towards behaviour recognition. Figure 14 shows a more detailed fly visual hull with higher polygonal complexity taken using six cameras. Finally, we are also working on algorithms to identify the fluorescence of green fluorescent protein and its analogue red fluorescent protein in flies. This will enable us to correlate specific gene expression with fly behaviour, a key step in understanding the inner workings of the fruit fly.

D.G. and S.T. were supported in part by NIH grant R01 GM67243. J.T. was supported by NIH grant AG11833. S.T. is a Royal Society Wolfson Research Merit Award holder.

REFERENCES

Balch, T., Khan, Z. & Veloso, M. 2001 Automatically tracking and analyzing the behavior of live insect colonies. In *Proc. Fifth Int. Conf. on Autonomous Agents, Montreal, Quebec, Canada. AGENTS '01*, pp. 521–528. New York, NY: ACM.

Ballerini, M. *et al.* 2008 Interaction ruling animal collective behavior depends on topological rather than metric distance: evidence from a field study. *Proc. Natl Acad. Sci.* **105**, 1232–1237. (doi:10.1073/pnas.0711437105)

Baumgart, B. G. 1974 Geometric modeling for computer vision. PhD thesis, Stanford University.

Brown, C., Grover, D., Ford, D., Hoe, N., Tavaré, S. & Tower, J. Submitted. Hydrogen peroxide stimulates activity and alters behavior in *Drosophila melanogaster.*

Chen, R. & Liu, J. S. 2000 Mixture Kalman filters. *J. R. Stat. Soc. B* **62**, 493–508. (doi:10.1111/1467-9868.00246)

Chu, C.-W. & Cohen, I. 2005 Posture and gesture recognition using 3D body shapes decomposition. *IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn.* **3**, 69. (doi:10.1109/CVPR.2005.510)

Cohen, I. & Li, H. 2003 Inference of human postures by classification of 3D human body shape. In *Proc. IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures, 17 October, 2003. AMFG*, p. 74. Washington, DC: IEEE Computer Society.

Cox, I. & Leonard, J. 1994 Modeling a dynamic environment using a Bayesian multiple hypothesis approach. *Artif. Intell.* **66**, 311–344. (doi:10.1016/0004-3702(94)90029-9)

Deriche, R. & Faugeras, O. 1990 Tracking line segments. *Image Vision Comput.* **8**, 261–270. (doi:10.1016/0262-8856(90)80002-B)

Dyer, C. R. 2001 Volumetric scene reconstructions from multiple views. In *Foundations of image understanding* (ed. L. S. Davis), pp. 469–489. Boston, MA: Kluwer.

Feldman, A. & Balch, T. 2004 Representing honey bee behavior for recognition using human trainable models. *Adapt. Behav.* **12**, 241–250. (doi:10.1177/105971230401200309)

Graetzel, C. F., Fry, S. N. & Nelson, B. J. 2006 A 6000 Hz computer vision system for real-time wing beat analysis of *Drosophila*. In *The First IEEE/RAS-EMBS Int. Conf. on Biomedical Robotics and Biomechatronics 2006. BioRob 2006*, pp. 278–283. Piscataway, NJ: IEEE Press.

Gruen, A. & Akca, D. 2005 Least squares 3D surface and curve matching. *Int. J. Photogramm. Remote Sens.* **1**, 151–174.

James, G. M. 2007 Curve alignment by moments. *Ann. Appl. Stat.* **1**, 480–501. (doi:10.1214/07-AOAS127)

Julier, S. J., Uhlmann, J. K. & Durrant-Whyte, H. F. 1995 A new approach for filtering nonlinear systems. *Proc. Am. Control Conf.* **3**, 1628–1632. (doi:10.1109/ACC.1995.529783)

Khan, S. & Shah, M. 2000 Tracking people in presence of occlusion. In *Proc. Asian Conf. on Computer Vision, Taipei, Taiwan, January 2000.*

Khan, Z., Balch, T. & Dellaert, F. 2005 MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 1805–1918. (doi:10.1109/TPAMI.2005.223)

Laurentini, A. 1994 The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 150–162. (doi:10.1109/34.273735)

MacQueen, J. B. 1967 Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability*, vol. 1, pp. 281–297. Berkeley, CA: University of California Press.

Matusik, W., Buehler, C., Raskar, R., Gortler, S. & McMillan, L. 2000 Image-based visual hulls. In *Proc. 27th Annual Conf. on Computer Graphics and Interactive Techniques. Int. Conf. on Computer Graphics and Interactive Techniques*, pp. 369-374. New York, NY: ACM Press/Addison-Wesley Publishing.

Matusik, W., Buehler, C. & McMillan, L. 2001 Polyhedral visual hulls for real-time rendering. In *Proc. 12th Eurographics Workshop on Rendering Techniques, 25–27 June 2001*, pp. 115–126. London, UK: Springer-Verlag.

Maybeck, P. S. 1979 *Stochastic models, estimation, and control mathematics in science and engineering*, vol. 141. New York, NY: Academic Press.

Mikic, I., Trivedi, M., Hunter, E. & Cosman, P. 2001 Articulated body posture estimation from multi-camera voxel data. *IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn.* **1**, 455.

Moezzi, S., Kuramura, D. Y. & Jain, R. 1996 Reality modeling and visualization from multiple video sequences. *IEEE Comput. Graph. Appl.* **16**, 58–63. (doi:10.1109/38.544073)

Murphy, R. F. 2004 Automated interpretation of subcellular location patterns. In *Proc. IEEE Int. Symp. on Biomedical Imaging: Nano to Macro, April 2004*, pp. 53–56. (doi:10.1109/ISBI.2004.1398472)

Parrish, J. K. & Turchin, P. 1997 Individual decisions, traffic rules and emergent patterns in schooling fish. In *Animal groups in three dimensions* (eds J. K. Parrish & W. H. Hamner), pp. 126–142. New York, NY: Cambridge University Press.

Parrish, J. K., Viscido, S. V. & Grunbaum, D. 2002 Self-organized fish schools: an examination of emergent properties. *Biol. Bull.* **3**, 296–305. (doi:10.2307/1543482)

Potmesil, M. 1987 Generating octree models of 3D objects from their silhouettes in a sequence of images. *Comput. Vision Graph. Image Process.* **40**, 1–29. (doi:10.1016/0734-189X(87)90053-3)

Preisler, H. K., Ager, A. A., Johnson, B. K. & Kie, J. G. 2004 Modeling animal movements using stochastic differential equations. *Environmetrics* **15**, 643–657. (doi:10.1002/env.636)

Rasmussen, C. & Hager, G. 2001 Probabilistic data association methods for tracking complex visual objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**, 560–576. (doi:10.1109/34.927458)

R Development Core Team 2007 R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. (http://www.r-project.org)

Shanmukh, K. & Pujari, P. 1991 Volume intersection with optimal set of directions. *Pattern Recogn. Lett.* **12**, 165–170. (doi:10.1016/0167-8655(91)90045-N)

Stauffer, C. & Grimson, W. E. L. 1999 Adaptive background mixture models for real-time tracking. *Proc. Comput. Vision Pattern Recogn.* **2**, 246–252.

Stenger, B., Mendonca, P. R. S. & Cipolla, R. 2001*a* Model-based hand tracking using an unscented Kalman filter. *Proc. Br. Mach. Vision Conf.* **1**, 63–72.

Stenger, B., Mendonca, P. R. S. & Cipolla, R. 2001*b* Model-based 3D tracking of an articulated hand. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition 2001 (CVPR'01)*, vol. 2, pp. 310–315. (doi:10.1109/CVPR.2001.990976)

Sullivan, S. & Ponce, J. 1998 Automatic model construction, pose estimation, and object recognition from photographs using triangular splines. In *Proc. Sixth Int. Conf. on Computer Vision, 4–7 January 1998*, pp. 510–516. Washington, DC: IEEE Computer Society.

Szeliski, R. 1993 Rapid octree construction from image sequences. *CVGIP: Image Understand.* **58**, 23–32. (doi:10.1006/ciun.1993.1029)

Tammero, L. F. & Dickinson, M. H. 2002 The influence of visual landscape on the free flight behavior of the fruit fly *Drosophila melanogaster*. *J. Exp. Biol.* **205**, 327–343.

Tsai, R.Y. 1986 An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, Miami Beach, FL*, pp. 364–374.

Turchin, P. 1998 *Quantitative analysis of movement: measuring and modeling population redistribution in plants and animals.* Sunderland, MA: Sinauer Associates.

Welch, G. & Bishop, G. 1995 An introduction to the Kalman filter. Technical report no. TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill.

Werghi, N. & Yijun X. 2002 Posture recognition and segmentation from 3D human body scans. In *Proc. First Int. Symp. on 3D Data Processing Visualization and Transmission*, pp. 636–639.

Wren, C., Azarbayejani, A., Darrell, T. & Pentland, A. 1997 PFINDER: real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 780–785. (doi:10.1109/34.598236)

Xu, M. & Ellis, T. 2001 Illumination-invariant motion detection using colour mixture models. In *Proceedings of the British Machine Vision Conference 2001, BMVC 2001, Manchester, UK, 10–13 September 2001.* Malvern, UK: British Machine Vision Association.