

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/321416437>

# Improvement of Honey Bee Tracking on 2D Video with Hough Transform and Kalman Filter

Article in Journal of Signal Processing Systems · November 2017

DOI: 10.1007/s11265-017-1307-x

CITATION

1

READS

63

2 authors:



Cheng Yang

Auckland University of Technology

6 PUBLICATIONS 8 CITATIONS

SEE PROFILE



John Collins

Auckland University of Technology

13 PUBLICATIONS 38 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Honey bees tracking and pollen sacs measurement on monitoring video [View project](#)

# Improvement of Honey Bee Tracking on 2D Video with Hough Transform and Kalman Filter

Cheng Yang<sup>1</sup> · John Collins<sup>1</sup>

Received: 31 August 2016 / Revised: 28 July 2017 / Accepted: 8 November 2017  
© Springer Science+Business Media, LLC, part of Springer Nature 2017

**Abstract** This paper introduces a novel approach to improve bee tracking. For this research, 50 Hz 2D videos have been recorded. The first part of the model tracks bees using a method combining colour thresholding and foreground detection, and then produces the bees' shapes as blobs on a binary image. These blobs are analysed to estimate the positions and directions of motion of the bees. However, when bees cross over one another in the image, they are hard to track. This paper tackles this problem by using the standard Hough transform applied to bee research. Then a Kalman filter is used to track the bees using their estimated position information. Because of the 50 Hz frame rate, the trajectories of the bee movements are too variable to track reliably. The Kalman filter is modified to fit this situation. Multiple bees are being tracked, so the Hungarian assignment algorithm is used to assign predictions and measurements to individual bees. The experiment shows the bees are reliably tracked in the close view 50 Hz 2D video.

**Keywords** Bee detection · Bee tracking · Hough transformation · Kalman filter · Object tracking

✉ Cheng Yang  
royang@aut.co.nz

John Collins  
jcollins@aut.ac.nz

<sup>1</sup> School of Engineering, Computer & Mathematical Sciences, Auckland University of Technology, 34 St Paul Street, City Campus, Auckland, New Zealand

## 1 Introduction

Honey bees are traditional pollination insects, and they also produce honey. In particular, Manuka honey is an export product exclusive to New Zealand. Many scientists and bee keepers are concerned about the health of bees, and they observe and analyse bees' behaviour. The use of an observation platform inside the beehive is popular, but there is little monitoring of bees outside the beehive, especially using a close up view. Campbell et al. [1] tracked bees outside the hive, but the shadows of the bees affected the outcome. Kimura et al. [2] tracked honey bees on a flat surface, but it did not mention whether their method could be used in the natural environment. This paper tracks the flight of bees outside the beehive. As the camera is close to the beehive entrance, the video can display the bees' bodies clearly.

In multiple object tracking, the common method is filtered tracking. Dearden et al. [3] introduced the particle filter to track soccer players on the field. Weng and Shantaiya et al. [4, 5] used the Kalman filter to track human beings moving on a video. All of the above [3–5] used filtered tracking. The filters predict the position assuming the velocity is constant and then combine the prediction and detection to track the object.

In this paper, the frame rate of the camera is 50 Hz, but the bees are flying fast. Therefore the situation is different from the people tracking in [3–5].

The approach of bee detection is implemented by the combination of foreground detection [6, 7] and colour base segmentation [8]. Detection produces bee blobs from the video image and displays them as a binary image. These blobs can be analysed to obtain the bee positions, and then bounding boxes can be drawn around the bees. Lu et al. [9] and Thouho et al. [10] used blob analysis to get the same information.

Position information is used for bee tracking using the Kalman filter. Normally, the Kalman filter is used to track

human beings and traffic vehicles, because their velocity is almost constant. The filter relies on the steady value of the velocity. However, in this research, there is no such steady value available for tracking. The Kalman filter predicts the bee positions for the next frame, and then combines the predictions and measurements (detections) to calculate the Kalman result for the next prediction.

The research in this paper is also concerned with multiple bee tracking. The Hungarian assignment method [11, 12] is used to assign predictions and detections.

If bees appear individually in the video frame, they can easily be tracked. However, because the video is only 2D, when two bees cross over each other, the detection only produces one blob of these bees. Therefore, there are two predicted positions corresponding to one merged blob in the frame. In this case, the Hough transform is used to calculate the positions of these two bees. Ballard [13] introduced the Hough transform to detect arbitrary shapes. Prasad and Leung [14] detected ellipse from edge information. Maji and Malik [15] used the Hough transform to detect objects. Their voting space could find the possible location of objects. In this paper, the Hough transform is applied to the detection of merged bees using information about the edges and orientations of their blobs in the immediately previous frames. This is a new application of Hough transform. Then the Kalman filter uses those positions as measurements (detections) to track the bees. This technique solves the problem of bees merging. The experiment is performed using MATLAB.

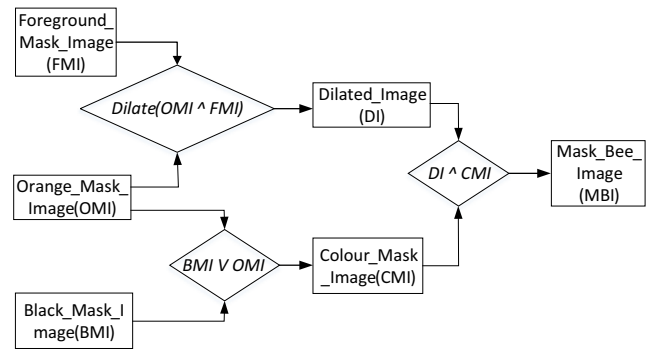
## 2 Bee Detection

### 2.1 Bee Detection with Image Segmentation

The image segmentation model uses a combination of foreground detection and colour base segmentation to detect bees.

Foreground detection uses the Gaussian Mixture Model (GMM) [7] to detect moving bees and ignores the unchanging background. This produces the binary image that is called the foreground mask image (FMI) in this paper. However, this also detects bee shadows and vegetation movement, which affects the result. The colour global thresholding using the Hue-Saturation-Value (HSV) colour space [16], detects the orange and black colour of bees. This removes the shadow and vegetation movement. This colour segmentation method produces two binary images we call the orange mask image (OMI) and black mask image (BMI). These three binary mask images are combined to detect the bee blobs. The logic is displayed in Fig. 1. More details are provided in [17].

Each blob can be analysed to estimate the position of the blob centre and to draw rectangular bounding boxes (bbox) around each blob.



**Figure 1** The combination logic.

### 2.2 Hough Transform for Merged Bee Detection

A difficulty occurs when two bees pass across each other in the image. Because we use 2D video, when the crossing happens, bee detection produces one big blob rather than two separate bee blobs.

After the merging has been identified (a procedure discussed in section 3), the next step is to detect bees in the merged blob. This is a new application of the Hough transform method, because the shapes of the merged bees are arbitrary.

It is a challenge to detect and separate two bees when they are merged together, because the bees have the same colour and patterns. Figure 2 displays the situation. In Fig. 2a the segmentation detects two bee blobs. In Fig. 2b the result of segmentation is only one merged blob in the following frame. The solution is to use the two individual bee shapes in Fig. 2b, to detect each bee in the merged blob in Fig. 2d.

Although the bee shapes appear different in the merged detection, the single bee shapes in the Fig. 2b frame are partly observable in the merged shape of the Fig. 2d frame. Therefore, the shape parameters from the Fig. 2b frame can be used to detect the individual bees in the merged frame of Fig. 2d. However, this method can only find an approximate area of pixels for each bee's position, because of the changes in the shapes.

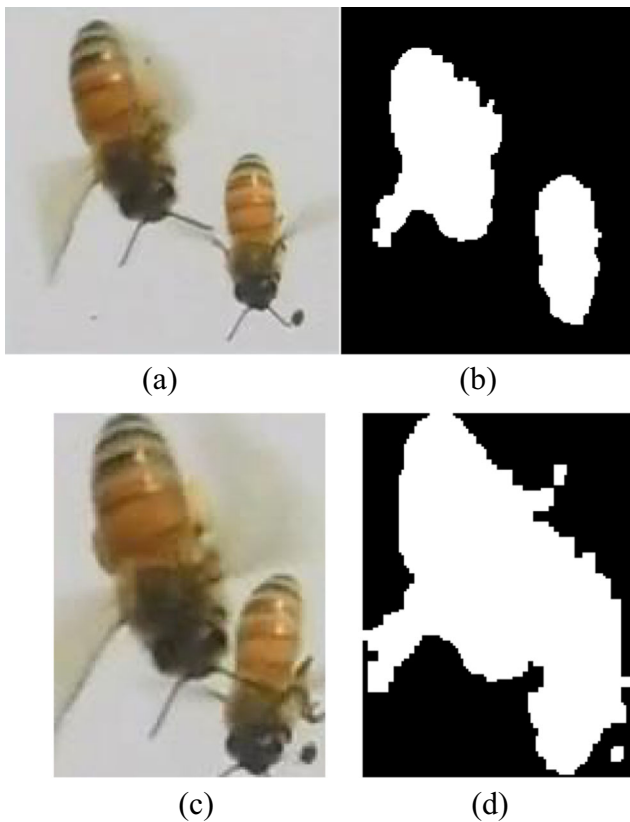
In Fig. 2b, the blob of each single bee can be cropped out. This blob image can be used to produce the edge image. Using the bee blob near the bottom right corner in Fig. 2b as an example, Fig. 3 displays the edge of this blob.

It is assumed that the edge image ( $I_s$ ) has width  $I$  and height  $J$ . The image pixel at  $i, j$  has the value:

$$I_s(i, j) = \begin{cases} 1, & \text{on the edge} \\ 0, & \text{not on the edge} \end{cases} \quad (1)$$

where  $1 \leq i \leq I$  and  $1 \leq j \leq J$ .

The centre of the image is at  $(i_c, j_c)$ . The pixels with value "1" define the edge curve on the image  $I_s$ . The set of pixels (S) on the edge of the single bee blob is:



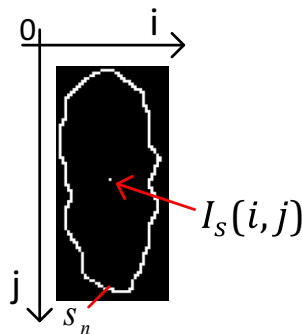
**Figure 2** The segmentation of bees blob. **a** The first frame of two bees **(b)** The blob segmentation of **(a)**. **c** The next frame of the two bees. **d** The merged blob segmentation of **(c)**.

$$S = \{(i, j) | I_s(i, j) = 1\}. \quad (2)$$

It is assumed that the set of edge pixels has  $N$  members. Each member of the set  $S$  is denoted by  $s_n = (i_n, j_n)$  representing the coordinates of edge pixel number ( $n \in [1, N]$ ).

In the edge image, the origin is at the top left corner. To detect a single bee in the merged blob, it is convenient to shift the origin to the centre of the edge image. The transformation is:

$$c_n = [u_n, v_n] = [i_n - i_c, j_n - j_c], \quad (3)$$



**Figure 3** The edge image of a single bee blob.

then the transformed edge pixel set ( $C$ ) is:

$$C = \{c_n | n \in [1, N]\}. \quad (4)$$

Figure 4 shows the merged blob edge of the two bees in Fig. 2c. This blob can also be cropped. The edge image of the merged blob ( $I_m$ ) has width  $X$  and height  $Y$ , and its pixels are:

$$I_m(x, y) = \begin{cases} 1, & \text{on the edge} \\ 0, & \text{not on the edge} \end{cases}, \quad (5)$$

where  $1 \leq x \leq X$  and  $1 \leq y \leq Y$ .

The pixels with value “1” are on the edge of the blob on the image  $I_m$ . The set of pixels ( $M$ ) on the edge of the merged blob is:

$$M = \{(x, y) | I_m(x, y) = 1\}. \quad (6)$$

This set of pixels has  $K$  members. Each member is denoted by  $m_k = (x_k, y_k)$  representing the coordinates of pixel number ( $k \in [1, K]$ ).

In the Hough transform technique, the edge pixels of the single bee blob are drawn on the merged blob edge image, with the centre of the single bee blob positioned on each pixel on the edge of the merged blob. Therefore, if the merged blob edge has  $K$  pixels, the single blob edge is drawn  $K$  times on the merged blob edge image. The coordinates of the resulting pixels drawn on the image are

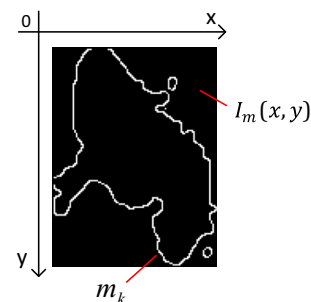
$$t_{nk} = [u_n + x_k, v_n + y_k]. \quad (7)$$

This transform set is known as  $T$ , where:

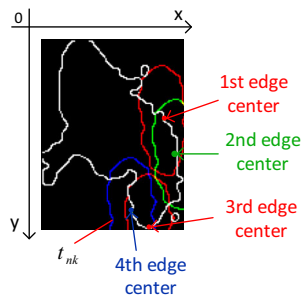
$$T = \{t_{nk} | n \in [1, N], k \in [1, K]\}. \quad (8)$$

These are the Hough Transform pixels. Some of the pixels  $t_{nk}$  are in the same position as each other. Figure 5 is an example of the single bee blob edge drawn four times on the merged blob edge image, with each drawing shown as a different colour. The centre of each drawing is at a different position on the edge of the merged blob. All of the pixels on these single bee edge drawings belong to  $T$ .

Each pixel in the merged blob edge image ( $I_m$ ) may match none, one or several of the Hough Transform pixels ( $T$ ). This



**Figure 4** The edge image of merged bee blob.



**Figure 5** An example of drawing the single bee blob edge on the merged blob edge image.

creates a voting map ( $V(x, y)$ ), whose size is the same as the size of merged blob edge image. Each element of the voting map array corresponds to a pixel of merged edge image. These elements are bins that record the number of Hough Transform pixels located at position  $(x, y)$ .

For  $(x, y)$  inside the merged blob,  $V(x, y)$  is the number of  $t_{nk}$  values equal to  $(x, y)$ . These points are  $x \in [1, X], y \in [1, Y]$ .  $V(x, y)$  is also called the voting value.

The voting map indicates the possible positions of the single bee on the merged image. If the single bee blob edge fits part of the merged blob edge, there is expected to be a corresponding element of voting map that has a peak value. This peak point is a candidate for the single bee's position in the merged blob.

However, it was observed that the motion of the bee can change its orientation by up to  $\pm 10^\circ$  between successive video frames. If the bee changes orientation, the voting map may produce a lower valued peak point. Conversely, the fitting orientation leads the voting map to get a sharp high peak point. The rotation of the bee's blob is described as:

$$\alpha = \{\alpha_l | (l \in 1, 2, \dots, L)\}, \quad (9)$$

where the  $L = 21$  and the

$$[\alpha_1, \dots, \alpha_{21}] = [-10^\circ, -9^\circ, \dots, 0^\circ, \dots, 9^\circ, \dots, 10^\circ]$$

are the angles which are used to rotate the single bee blob edge from its original orientation. The plus sign indicates an anti-clockwise rotation and the minus sign indicates a clockwise rotation. If the single bee blob edge is rotated to  $\alpha_l$  the edge pixel coordinates relative to the centre of the single bee blob are:

$$c_{n\alpha_l} = \begin{bmatrix} u_n \times \cos\alpha_l + v_n \times \sin\alpha_l, \\ v_n \times \cos\alpha_l - u_n \times \sin\alpha_l \end{bmatrix}. \quad (10)$$

Each rotated single bee blob edge curve can be drawn on the merged edge image to generate the voting map for an  $\alpha_l$  rotation. In this case, there are 21 different voting maps corresponding to the 21 different rotations. The different voting maps display different valued peak points. If a rotation makes

the single bee edge fit the merged edge, a peak point in the voting map is high and sharp, with other peak points being much lower. If the rotation does not fit the merged edge, there may be two or three peak points with a similar, but lower, value. Figure 6 shows an example. In this figure, the  $X$  and  $Y$  values are the coordinates of the peak point, and the  $Z$  value is the voting value of the point. The rotation of 0 degree means no change in the orientation of the single bee blob edge (Fig. 6a–c). In this case, there are three peak points having the same level. The rotation of 10 degree in Fig. 6d–f does not fit the merged edge. The three peak points have a similar level 35, 36 and 37. Figure 6g–i is the fitting rotation in which the peak point in the voting map is sharply 53, while the second high peak level is only 39. It can be seen that the fitting rotation produces the correct detection and usually has the highest peak value.

In summary, the 21 voting maps produce 21 candidate positions since each voting map produces exactly one candidate position, corresponding to its peak value. However, the fitting rotation may not be the highest peak value. This is because the single bee shape sometimes fits another part of the merged shape rather than the correct part, so that the wrong position also includes the high peak point. If the peak point is near the correct position, the region around this point includes many higher voting value points. This is described as the **correct region**. Conversely, the region around the wrong position has some low value points, and this is known as the **incorrect region**, so that this peak point is sharp and isolated.

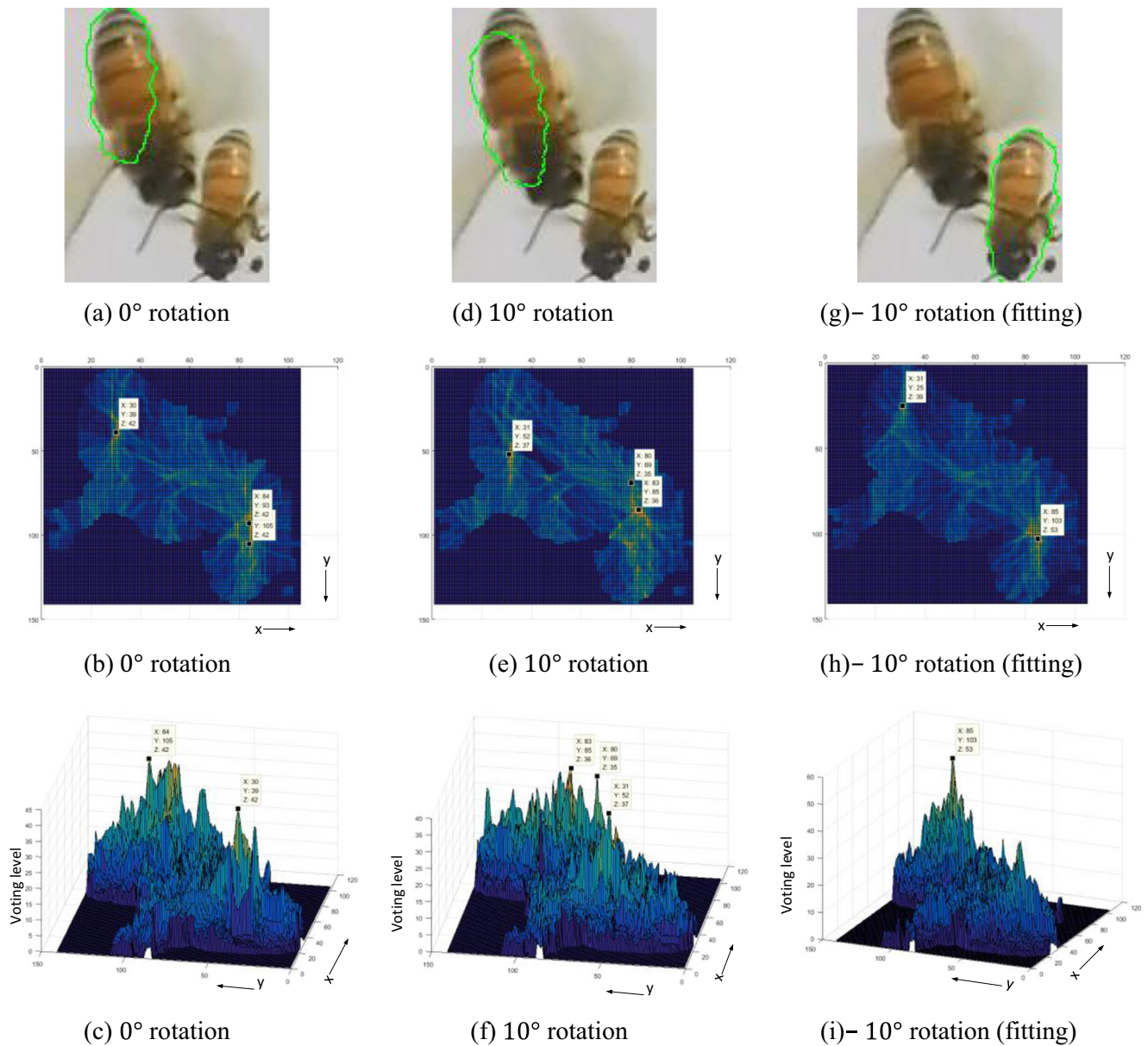
Figure 7 indicates the situation. It can be seen that the top point of wrong rotation (Fig. 7a–b) has a similar level (94) to the top point of the fitting rotation (level 95 in Fig. 7c–d). However, the top point in Fig. 7a–b is more isolated than the top point in Fig. 7c–d. The problem is solved by using the sum of the region values around the top points. In each of the 21 voting maps, all the values at voting points within  $\pm 5$  pixels of the top point are added together. It is assumed the top point in the voting map for orientation index  $l$  is  $V_l(x_p, y_p)$ , and the total value of the region around this top point is:

$$r_l = \sum_{x=x_p-5}^{x_p+5} \sum_{y=y_p-5}^{y_p+5} V_l(x, y), \quad (11)$$

where  $l \in [1, 21]$ . The top value of the 21 region total values produces the correct region which includes the actual position of the bee. In addition,  $l$  gives the orientation of the bee. Take Fig. 7 as an example. After the calculation, the region around top point of Fig. 7b has a value of 3950; but the value of the region around the top point of Fig. 7d is 5501. Therefore, the correct region is from Fig. 7c–d in which the fitting rotation is  $10^\circ$ .

The top point in the correct region might be expected to be the candidate position. However, this point is only near the





**Figure 6** The final detection of results with different rotation of the single bee edge. The -10° rotation fit the detection, and voting map have top peak point.

actual position, because of the shape noise. The shape noise comes from the segmentation. It is difficult to get accurate shape detail when performing segmentation, but the detection can be improved by a weighted average calculation near the correct region. The correct region itself is too small to be used to modify the position calculation. The correct region is extended to an area of  $\pm 20$  pixels from the peak point, and a weighted average of the voting map is calculated as below:

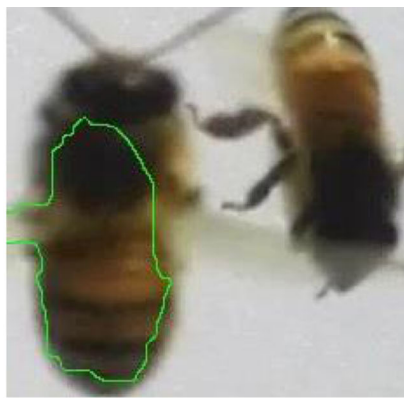
$$w_x = \frac{\sum_{x=x_p-20}^{x_p+20} \sum_{y=y_p-20}^{y_p+20} x V(x, y)}{\sum_{x=x_p-20}^{x_p+20} \sum_{y=y_p-20}^{y_p+20} V(x, y)}, \quad (12)$$

$$w_y = \frac{\sum_{x=x_p-20}^{x_p+20} \sum_{y=y_p-20}^{y_p+20} y V(x, y)}{\sum_{x=x_p-20}^{x_p+20} \sum_{y=y_p-20}^{y_p+20} V(x, y)}. \quad (13)$$

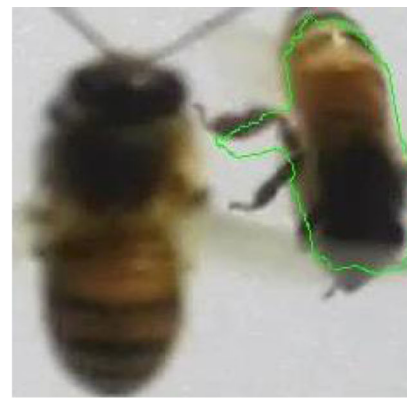
In formula (12) and (13), the point at  $(w_x, w_y)$  replaces the peak point as the candidate position of the bee.

This is the final detection of the bee on the merged image. The other merged bee is detected in the same way.

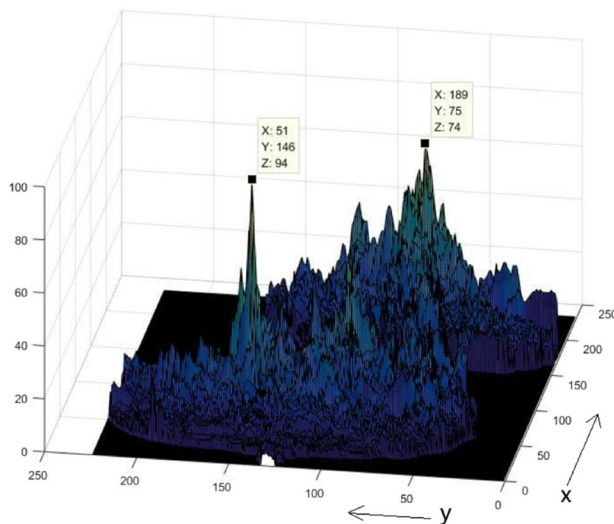
In the following merged frame, the fitting rotation still uses the original orientation, and the single bee detection which is just before the first merged frame is still used to detect the merged bee. This method works until the two bees separate.



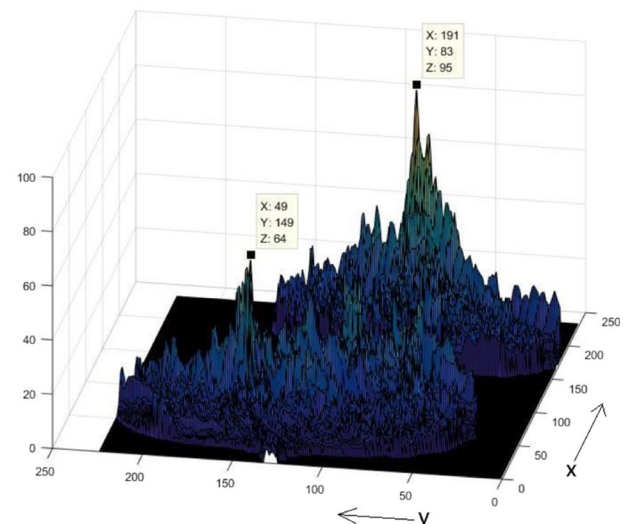
(a) 3° rotation



(c) 10° rotation



(b) 3° rotation



(d) 10° rotation

**Figure 7** The peak point problem with different rotation. The fitting rotation is 10°.

### 3 Tracked Bees with Kalman Filter

The video has a 50 Hz frame rate and its view is very close to the bees. Therefore, a difficulty is that the bees can quickly fly anywhere from one frame to the next. In this case, the normal Kalman filter for object tracking [5] may not work well, because the bees are changing positions, velocities and accelerations. There is no steady variable for the normal Kalman filter. However, taking another view, the Kalman filter can combine the prediction and measurement to produce a correction for the prediction of next frame. Therefore, if the position of a bee is the state of Kalman filter, the reasonable correction is close to the actual position of the bee. In this research, the measurement of bee detection is more reliable than the prediction when tracking a single bee, so the correction is close to the detection. Conversely, in the merged bee situation, the detection is not as reliable as single bee tracking, so the correction depends on both the prediction and detection. This

method requires the estimation of the covariance matrices of the noises.

#### 3.1 The Kalman Filter Principle for Tracking

In object tracking, the Kalman filter is discrete at frame number  $k$ . The Kalman filter estimation is:

$$\text{Prediction :} \quad p_k = A s_{k-1} \quad (14)$$

$$P_k = A Q_{k-1} A^T + Q_0 \quad (15)$$

$$\text{Measurement :} \quad m_k = H S_k \quad (16)$$

$$\text{Correction :} \quad K_k = P_k (P_k + R_k)^{-1} \quad (17)$$

$$s_k = p_k + K_k(m_k - p_k), \quad (18)$$

$$Q_k = (I - K_k)P_k. \quad (19)$$

From Eq. (14) to (19), the matrix  $P_k$  is the covariance matrix of the prediction noise. The matrix  $R_k$  is the covariance matrix of the measurement noise. The matrix  $S_k$  is the actual state from the detection measurement. The matrix  $Q_k$  is the covariance matrix of the estimated state  $s_k$ . The matrix  $H$  is the measurement model matrix, which is identity matrix in this case. The correction is controlled by  $P_k$  and  $R_k$ . If the prediction ( $p_k$ ) is unreliable, but the measurement from the bee detection is accurate, the noise of  $P_k$  is greater than  $R_k$ , so that the Kalman gain  $K_k$  is nearer to one. Then the estimate state  $s_k$  is closer to the measurement than to the prediction. If the situation is opposite, the state  $s_k$  is closer to the prediction than the measurement.

In this research,  $Q_0$  is the variance due to the fact that the bee velocity is not constant. It adds to the noise of the prediction.  $Q_0$  is not only added once at the beginning of tracking, it is added in each Kalman filter calculation.

To apply the Kalman filter formulas, the state of a bee is its position  $(x_k, y_k)$ , and the tracking model in frame  $k$  is:

$$x_k = 2x_{k-1} - x_{k-2}, \quad (20)$$

$$y_k = 2y_{k-1} - y_{k-2}. \quad (21)$$

If the prediction of the Kalman filter is:

$$p_k = A_{k-1}s_{k-1}, \quad (22)$$

then the predicted state vector is:

$$p_k = [x_k, y_k]^T. \quad (23)$$

The state vector in the frame  $k$  is:

$$s_k = [x_k, y_k, x_{k-1}, y_{k-1}]^T. \quad (24)$$

The transition matrix is defined as:

$$A_{k-1} = \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 \end{bmatrix} = A. \quad (25)$$

The measurement model of the Kalman filter is:

$$m_k = H_{k-1}S_k. \quad (26)$$

The measurement state vector is:

$$S_k = [x_k, y_k]^T. \quad (27)$$

The measurement vector is:

$$m_k = [x_k, y_k]^T. \quad (28)$$

The measurement transition matrix is:

$$H_{k-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = H. \quad (29)$$

The covariance matrix of corrected noise is:

$$Q_k = \begin{bmatrix} q_{11k} & 0 & 0 & 0 \\ 0 & q_{22k} & 0 & 0 \\ 0 & 0 & q_{11k-1} & 0 \\ 0 & 0 & 0 & q_{22k-1} \end{bmatrix}. \quad (30)$$

### 3.2 Estimation of the Covariance Matrices

The covariance values are estimated from the differences between the system calculation of the bee positions in the model and the actual bee positions as calculated manually. The error is calculated by:

$$e_m = [x_d - x_a, y_d - y_a], \quad (31)$$

where  $e_m$  is the measurement error,  $[x_d, y_d]$  is the detected position and  $[x_a, y_a]$  is the actual position. In the similar way, the prediction error is calculated from the difference between the system prediction and the actual position.

$$e_p = [x_p - x_a, y_p - y_a], \quad (32)$$

where  $e_p$  is the prediction error and  $[x_p, y_p]$  is the predicted position.

It is assumed the  $x$  and  $y$  values are independent to each other, so both the covariance matrices are diagonal. For single bee tracking, 172 single bee images were used to estimate errors and calculate the variances. The measurement covariance matrix was:

$$R_k = \begin{bmatrix} 13 & 0 \\ 0 & 10 \end{bmatrix}. \quad (33)$$

The  $x$  and  $y$  variances could be different because  $x$  is parallel to the front of the beehive and  $y$  is perpendicular to the front of the beehive. Both  $x$  and  $y$  are parallel to the ground surface below the beehive.

In addition, the covariance matrix  $Q_0$  was calculated from bee positions on images.

$$Q_0 = \begin{bmatrix} 124 & 0 \\ 0 & 78 \end{bmatrix} \quad (34)$$

These two covariance matrices are assumed to be constant during single bee tracking. However, because bees can merge with each other in the image, the prediction and measurement covariance matrices in the merged tracking are different from



the single tracking. 91 merged bee images were utilised to get the following matrices for the merged situation:

$$R_k = \begin{bmatrix} 419 & 0 \\ 0 & 717 \end{bmatrix}, \quad (35)$$

$$Q_0 = \begin{bmatrix} 435 & 0 \\ 0 & 825 \end{bmatrix}. \quad (36)$$

These two covariance matrices above are also assumed to be constant during merged bee tracking.

### 3.3 The Tracking Model

In the frame where a bee first appears, there is no previous position, so the predicted position for the next frame is taken to be the same as the current measured position. (This applies to all bees in the first frame of a new video.) Therefore, the first prediction will not be accurate, but the next prediction should agree with the detection. In the following frame, bees are tracked using the Hungarian assignment method [12]. This method assigns predictions to detections in an optimal way, to minimize the sum of the distances between the assigned predictions and detections. As a constraint, a prediction will not be assigned to a detection when

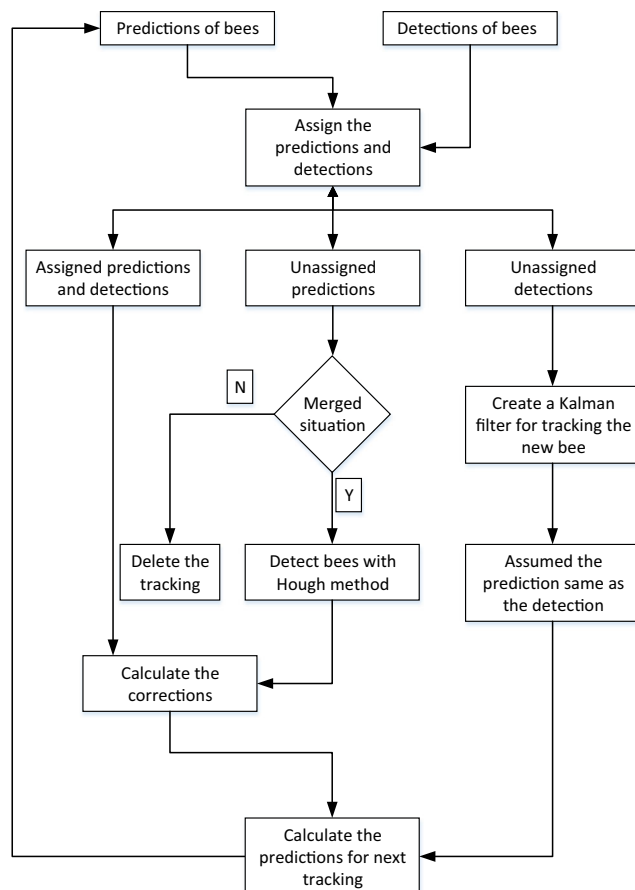
the distance is more than 80 pixels. This will usually result in some unassigned predictions and detections. When a detection is unassigned, it is probably a bee appearing for the first time. In addition, unassigned predictions can be caused by a bee disappearing or bees moving across each other in the video.

Figure 8 shows the whole tracking model process. When a bee just appears for the first time, a Kalman filter model is created for tracking it. After the creation of the tracking Kalman filter, predictions and detections are assigned to each other for the correction of the Kalman filter results. The correction produces the next prediction. If a prediction is unassigned, it has to be estimated whether it is in the merged situation or not.

In the merged situation, two predicted bounding boxes (bboxes) overlap one detected bounding box (bbox). Following the Hungarian assignment method, there is an unassigned prediction. The bbox information from the blob analysis is used to identify the merge occurrence.

For example, Fig. 9 displays a merged situation, where the two predicted bboxes (red) overlap the merged detected bbox (blue). In this situation, one prediction is assigned with this merged detection. The other prediction is unassigned, even though the unassigned prediction bbox overlaps the merged detected bbox. This is how the merge situation is recognised.

In Fig. 8, if a merged situation is recognised, the Hough transform is used to locate each bee (as outlined in section 2). If it is not a merged situation, the tracking model will be deleted, because it is assumed the bee has disappeared, so it does not need to be tracked any more.

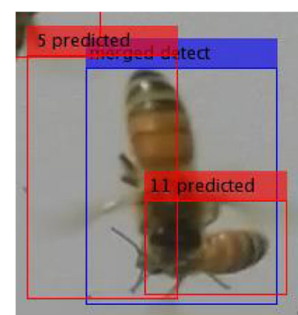


**Figure 8** The tracking model process.

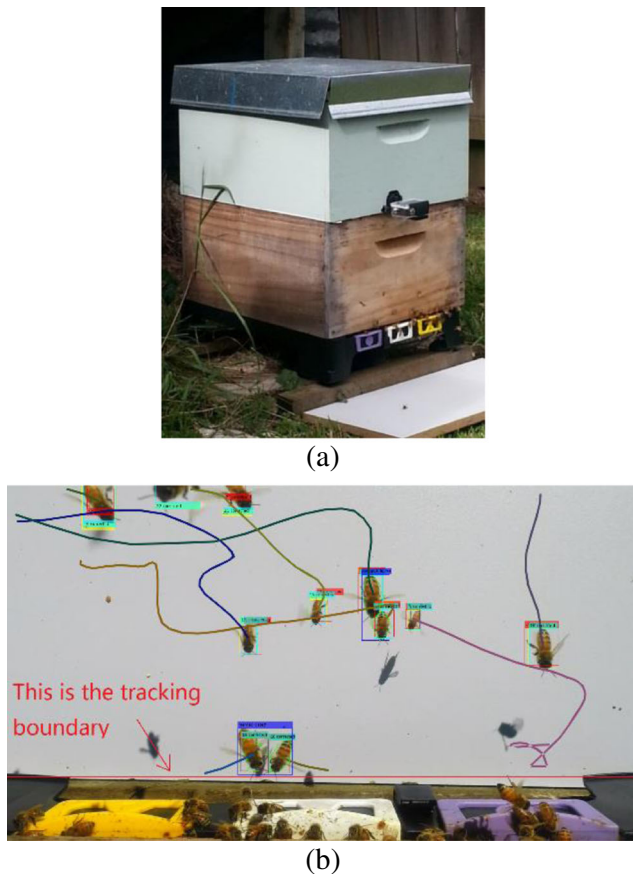
## 4 Implementation and Results

### 4.1 Implementation Background

The camera is attached to the front wall of the beehive, facing down about 30 cm above the entrance. A single white colour board is placed on the ground in front of the entrance of beehive, to simplify the image background. This is necessary, because the natural background includes withered grass with a similar colour to the bees and this interferes with the bee detection. Figure 10a shows the situation. The video is



**Figure 9** The merged bees situation.

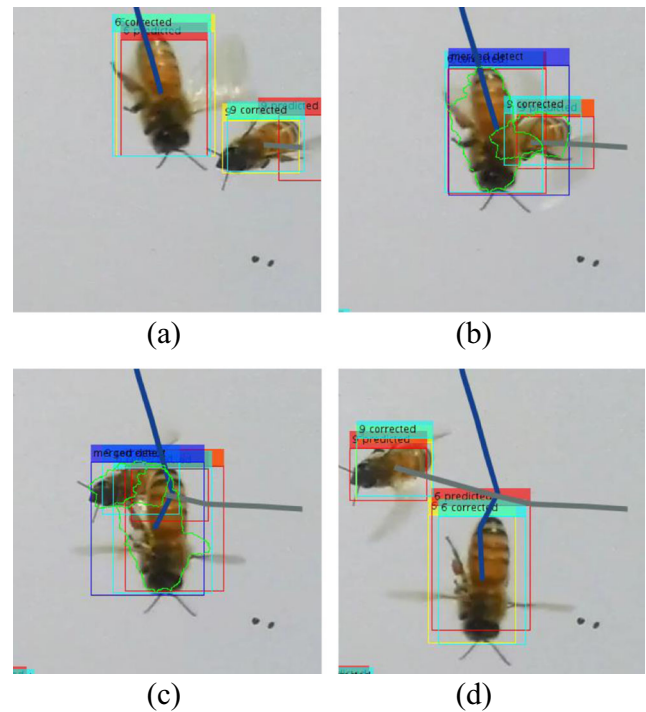


**Figure 10** The camera position (a) and the camera view (b).

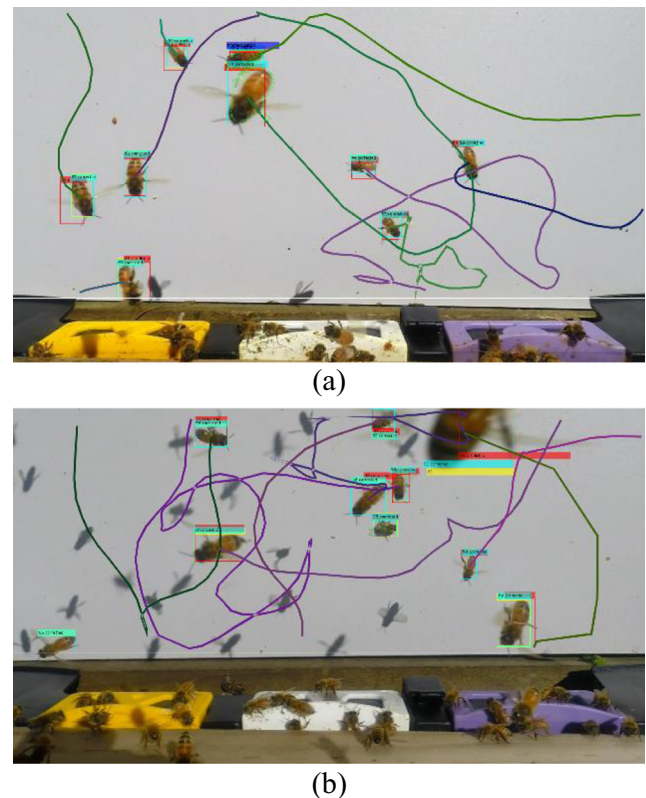
1920\*1080 resolution at a rate of 50 frames per second. The videos are recorded and copied to a computer for MATLAB processing. The computer is a 64bit Intel Core i7 3.40GHz with 32GB RAM. MATLAB with the computer vision tool-box is used to run a program following Fig. 8. The program can track bees automatically in the videos.

A Matlab function performs the blob analysis. It not only calculates the blob information, but also outputs bounding boxes around each bee blob. The tracking is indicated by adding indexes to each bounding box and curves marking each bee's trajectory with different colours in the video. A tracking boundary is set up just in front of the hive entrance, which is shown by the red line in Fig. 10b, so that the system only tracks bees in front of the white background area. This is because the area below the boundary line has complex colour information which affects the bee detection. Figure 10b shows an example of tracking output over 40 frames. The blue and red bounding boxes mark the detected and predicted bee positions respectively.

The model implementing tracking in the merged bee situation is shown in Fig. 11. The green outline shapes show the detections produced by the Hough transform when the bees are merged. The blue bounding boxes show the detection of merged blobs. The images (a) to (d) are successive frames. These examples demonstrate successful tracking.



**Figure 11** The result of merged situation tracking. Each image is the neighbour frame among the frame flow.



**Figure 12** Example frames from two videos. **a** The active video. **b** The shadow video.

**Table 1** The evaluation of single bee tracking.

Videos	Frames	Bees' images	Single bee tracking		
			Correct	Incorrect	Accuracy %
Active	300	1744	1740	4	99.77
Shadow	300	1630	1625	5	99.69

## 4.2 Evaluation of the Merged Detection and Tracking Results

There are two main situations that need to be evaluated: single bee tracking and merged situation tracking. There are several factors that affect the tracking accuracy.

First, merged detection failures. In the frames, bees continually overlap each other when they fly actively. It is common to see more than two bees merge, sometimes up to five bees merging in a single blob. Sometimes, bees fly close to the camera, so that the image has a large bee overlapping many smaller bees. The merged detection method may not be able to track bees in this situation. In addition, if bees keep merging over a long period of time, they may change direction and shape. Therefore, the detection may fail, because it is only based on the shape information before the merging. This factor is the main problem of this tracking method.

Second, boundary errors. Most bees come from beyond the image boundary. Some of them are only partly visible on their first appearance in a frame. In the single bee situation, tracking starts with these partly visible bodies. For merged bees on the boundary, tracking will not start if a bee is merged with another bee when it first becomes visible. In addition, bees may stay merged as they disappear beyond the frame boundary, so they are not seen separately. Then it is hard to know whether the merged tracking is correct or not. Therefore, as mistakes are unpredictable on the boundary, they will not be included in the evaluation of merged tracking.

Third, bees flying close to the camera. As the video is recorded in the natural environment, the video may show some bees as being very big, because they may fly close to the camera. These big blobs of bees may lead to unpredictable mistakes, which are ignored in the evaluation.

Finally, fourth, bee shadows. Although the single colour (white) board simplifies the background, shadows can still appear on the video. The shadows do not have an orange colour, but they may be detected as moving objects. If the bee detection method (section 2.1) cannot successfully remove shadows, the system may recognise them as bees. Therefore, shadows may affect bee tracking accuracy. One type of video which includes many shadows has been chosen to see test whether shadows affect the tracking or not.

Considering the four factors, two videos have been chosen to demonstrate the results. All of them were taken in a sunny environment, when detection is better than on a cloudy day. The first video is an active video (Fig. 12a). There are about eight bees in the frame view. This video was taken in the middle of the day, when the sunlight was clear and bees were flying actively. This video captures not only single bees, but also many merged situations, so it is used to estimate results for the single and merged situation tracking. The second video is also an active one, but there are many shadows on the board (Fig. 12b). This is the shadow video. This video also captures bees flying in single and merged situations. In addition, it tests the method when there are many shadows in the background.

The evaluation results for single bee tracking are listed in Table 1. The active and shadow videos were evaluated over 300 frames to capture single tracking situations. In the table, the single bee tracking columns record the total number of bees tracked in these frames. If a bee index was assigned to the same bee between two consecutive frames in the video, this was counted as one correct tracking; otherwise, it was incorrect. After processing, the videos were all examined manually to determine whether each bee had been tracked correctly. In the active video, single tracking accuracy is over 99%, and in the shadow video accuracy is also over 99%. It can be seen that the shadows have little effect on tracking.

The merged tracking with the Hough transform is compared to tracking without it. Both of the tracking models use the Kalman filter described in chapter 3. In the tracking model without the Hough transform, the merged tracking uses the centre of the merged blob as the measurement for the Kalman filter.

Table 2 reports the accuracy of merged tracking. The Correct column records that a bee after merging was identified correctly as the same bee before merging. For example, if three bees merged and then separated, and one of them was

**Table 2** The tracking evaluation and comparison.

Videos	Frames	Tracked bees	Kalman tracking			Kalman + Hough tracking		
			Correct	Incorrect	Accuracy%	Correct	Incorrect	Accuracy%
Active	1000	190	128	62	67.4	152	38	80.0
Shadow	1000	178	136	42	76.4	156	22	87.6
Total	2000	368	264	104	71.7	308	60	83.7



correctly identified with itself before merging, but the other two were incorrectly identified, then one result is Correct and other two are Incorrect. The Incorrect results do not include boundary errors.

The table shows that during 1000 frames of active video, there were a total of 190 bees in merged situations. When using only Kalman filter tracking, 128 are correctly tracked and 62 are incorrectly tracked. The accuracy is 67.4%. In the model of Kalman filter with the Hough transform, 152 bees were tracked correctly, and 38 incorrectly. The accuracy increased to 80.0%. The merged tracking on the shadow video was sometimes more accurate because other factors were causing the errors. The Kalman filter tracked bees with 136 correct and 42 incorrect. The tracking accuracy was 76.4%. The result of using Hough transform is 156 correct tracking and 22 incorrect tracking. The accuracy level is increased by 11.2% to 87.6%. Visual inspection of the frames also showed that shadows have had little effect on the tracking. The variation between the results for the active and shadow videos appear to be not related to the presence of shadows.

## 5 Conclusion

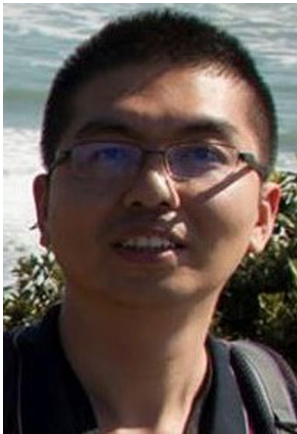
This paper used a combination method for image segmentation to detect individual bees. In addition, it introduced a new way to apply the Hough transform to detect merged bees. The Kalman filter in this paper is utilized in the situation of non-steady variable tracking. The tracking model with the Kalman filter correctly tracked over 99% of single flying bees, and correctly tracked bees in the merged situation 72% of the time. However, with the Hough transform, merged bee tracking success increased to 84%. This method helps to solve the problem of the merged situation on 2D video.

However, there is still some future work required. Firstly, the Hough transform is too slow to find the bees' positions in the merged situation. At present, this takes 8 to 33 s per frame, depending on size of the blob and number of bees. It is necessary to improve the speed of this procedure to achieve real time tracking. Secondly, as bee image segmentation relies on the orange and black colour of bees, there is a problem in differentiation arising from the natural background containing black (soil) and orange (withered grass) colours.

**Acknowledgements** The paper is supported by Darren Bainbridge, who provided access to beehives.

## References

- Campbell, J., Mummert, L., & Sukthankar, R. (2008). Video monitoring of honey bee colonies at the hive entrance. *Visual Observation & Analysis of Animal & Insect Behavior*, 8, 1–4.
- Kimura, T., Ohashi, M., Crailsheim, K., Schmickl, T., Odaka, R., & Ikeno, H. (2012). Tracking of multiple honey bees on a flat surface. *Emerging Trends in Engineering and Technology. Fifth International Conference* (pp. 36–39). IEEE. <https://doi.org/10.1109/ICETET.2012.25>.
- Dearden, A., Demiris, Y., & Grau, O. (2006). Tracking football player movement from a single moving camera using particle filters. *Proceedings of the 3rd European Conference on Visual Media Production* (pp. 29–37). <https://doi.org/10.1049/cp:20061968>.
- Weng, S.-K., Kuo, C.-M., & Tu, S.-K. (2006). Video object tracking using adaptive Kalman filter. *Journal of Visual Communication and Image Representation*, 17(6), 1190–1208. <https://doi.org/10.1016/j.jvcir.2006.03.004>.
- Shantaiya, S., Verma, K., & Mehta, K. (2015). Multiple object tracking using Kalman filter and optical flow. *European Journal of Advances in Engineering and Technology*, 2(2), 34–39.
- Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition. IEEE Computer Society Conference*, 2, 264–252. <https://doi.org/10.1109/cvpr.1999.784637>.
- Wang, K., Liang, Y., Xing, X., & Zhang, R. (2015). Target detection algorithm based on Gaussian mixture background subtraction model. In Z. Deng & H. Li (Eds.), *Proceedings of the Chinese intelligent automation conference* (Vol. 336, pp. 439–447). Berlin: Springer. [https://doi.org/10.1007/978-3-662-46469-4\\_47](https://doi.org/10.1007/978-3-662-46469-4_47) Lecture Notes in Electrical Engineering.
- Cheng, H. D., Jiang, X. H., Sun, Y., & Wang, J. (2001). Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12), 2259–2281. [https://doi.org/10.1016/S0031-3203\(00\)00149-7](https://doi.org/10.1016/S0031-3203(00)00149-7).
- Lu, S., Tsechpenakis, G., Metaxas, D. N., Jensen, M. L., & Kruse, J. (2005). Blob analysis of the head and hands: a method for deception detection. *System Sciences. Proceeding of the 38th Annual Hawaii International Conference* (pp. 20–23). <https://doi.org/10.1109/HICSS.2005.122>.
- Thou-Ho, C., Yu-Feng, L., & Tsong-Yi, C. (2007). Intelligent vehicle counting method based on blob analysis in traffic surveillance. *Innovative Computing. Information and Control. Second International Conference* (pp. 238–238). <https://doi.org/10.1109/icic.2007.362>.
- Kuhn, H. (2005). The Hungarian method for the assignment problem. *Naval Research Logistics*, 52(1), 7–21. <https://doi.org/10.1002/nav.20053>.
- Nandashri, D., & Smitha, P. (2015). An efficient tracking of multi object visual motion using hungarian method. *International Journal of Engineering Research and Technology*, 4(4). [10.17577/IJERTV4IS041410](https://doi.org/10.17577/IJERTV4IS041410).
- Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2), 111–122. [https://doi.org/10.1016/0031-3203\(81\)90009-1](https://doi.org/10.1016/0031-3203(81)90009-1).
- Prasad, D. K., & Leung, M. K. (2012). Methods for ellipse detection from edge maps of real images. *Machine Vision-Applications and Systems*. InTech. <https://doi.org/10.5772/35150>.
- Maji, S., & Malik, J. Object detection using a max-margin Hough transform. *Computer Vision and Pattern Recognition. IEEE Conference* (pp. 1038–1045). <https://doi.org/10.1109/cvpr.2009.5206693>.
- Cardani, D. (2001). Adventures in hsv space. Laboratorio de Robótica, Instituto Tecnológico Autónomo de México.
- Cheng, Y., & Collins, J. (2015). A model for honey bee tracking on 2D video. *International Conference on Image and Vision Computing New Zealand* (pp. 1–6). <https://doi.org/10.1109/IVCNZ.2015.7761542>.



**Cheng Yang** received M.E.S degree in Electrical and Electronics Engineering from the Auckland University of Technology in the year 2012. He is currently pursuing his PhD degree in Electrical and Electronics Engineering and is attached to the Centre for Robotics and Vision Team (CeRV), School of Engineering, Auckland University of Technology. His research interests include image processing and machine vision.



**John Collins** received the BSc(Hons) degree in physics and mathematics and the PhD degree in electronic engineering from the University of Auckland, New Zealand. He is currently a senior lecturer in the School of Engineering at the Auckland University of Technology. He has extensive industrial experience in the development of embedded systems and has worked as a consultant on a number of embedded systems projects. His

current research interests include robotics, embedded systems design and verification, and image processing for embedded systems.