Computers and Electronics in Agriculture 122 (2016) 10–18

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Automatic behaviour analysis system for honeybees using computer vision

Gang Jun Tu [a,*], Mikkel Kragh Hansen [a], Per Kryger [b], Peter Ahrendt [a,*]

[a] Department of Engineering, Aarhus University, Finlandsgade 22, 8200 Aarhus N, Denmark
[b] Department of Agroecology, Aarhus University, Forsøgsvej 1, 4200 Slagelse, Denmark

ARTICLE INFO

Article history:
Received 4 June 2015
Received in revised form 7 January 2016
Accepted 7 January 2016
Available online 22 January 2016

Keywords:
Video monitoring system
Raspberry Pi
Computer vision
Honeybees
Counting
In-and-out activity

ABSTRACT

We present a fully automatic online video system, which is able to detect the behaviour of honeybees at the beehive entrance. Our monitoring system focuses on observing the honeybees as naturally as possible (i.e. without disturbing the honeybees). It is based on the Raspberry Pi that is a low-cost embedded computer with very limited computational resources as compared to an ordinary PC. The system succeeds in counting honeybees, identifying their position and measuring their in-and-out activity. Our algorithm uses background subtraction method to segment the images. After the segmentation stage, the methods are primarily based on statistical analysis and inference. The regression statistics (i.e. R^2) of the comparisons of system predictions and manual counts are 0.987 for counting honeybees, and 0.953 and 0.888 for measuring in-activity and out-activity, respectively. The experimental results demonstrate that this system can be used as a tool to detect the behaviour of honeybees and assess their state in the beehive entrance. Besides, the result of the computation time show that the Raspberry Pi is a viable solution in such real-time video processing system.

1. Introduction

Honeybees play an important role in agriculture when pollination is required, and in such practices pollination units are regarded as an important factor in agricultural economy. They are clearly the hard working pollinators. For example, only in the United States, the worth of honeybee pollination is estimated at $15 billion (Sagili and Burgett, 2011). However, in recent years, the rates of honeybee colony failure have increased significantly across much of the world because of Colony Collapse Disorder (CCD) (Russell et al., 2013). CDD is characterized by a sudden disappearance of honeybees from beehives, and in the worst case the entire population disappears. The relative contribution of those stressors in CCD events remains unknown (Henry et al., 2012). For this reason, the research of honeybee behaviour in colony has drawn great attentions from both bee-keeper and biologist, and recently also from computer scientist.

The study of behaviour of honeybees under video recordings is still an open challenge in computer vision. Developing an automatic monitoring system to detect the behaviour of honeybees is an important technical issue in the particular field of study (Chen et al., 2012). An automatic monitoring system can efficiently obtain the behaviour of honeybees, which can be used to analyse the activities of the honeybees such as foraging activity (Adeva, 2012). In the past ten years, two technologies have dominated for monitoring honeybees: (1) RFID sensors such as (Henry et al., 2012); (2) video-based surveillance such as (Campbell et al., 2008; Kimura et al., 2011; Chen et al., 2012; Chica, 2012; Chica et al., 2013). There are also other technologies such as capacitance-based sensors (Campbell et al., 2005) and infrared light sensors (Struye et al., 1994; BeeSCAN, 2005).

The video-based surveillance is relatively recent and has aimed at the behavioural analysis from both the inside and outside of beehives. One approach was based on video from the inside of the beehive for identification and tracking of individual honeybees without any markers and employing vector-quantization for image segmentation (Kimura et al., 2011). Another study aimed at counting the number of honeybees outside of the entrance of the beehive (Campbell et al., 2008). In this study, an elliptical shape was fitted to detect a single honeybee and used a motion model to describe various activities such as loitering and crawling as well as flying in/out. Unfortunately, the authors did not show their experimental results in details. In Chen et al. (2012), the honeybees moved in/out through a passageway one at a time. In this system, the in-and-out

* Corresponding authors. Tel.: +45 28825522 (G.J. Tu), +45 41893275 (P. Ahrendt).

E-mail addresses: tujun@eng.au.dk, tuganjun@hotmail.com (G.J. Tu), pah@ase.au.dk (P. Ahrendt).

http://dx.doi.org/10.1016/j.compag.2016.01.011
0168-1699/

activity of the honeybees was restricted because of the passageway.

In order to monitor the behaviour of honeybees as naturally as possible we designed and implemented a video monitoring extension box for the beehive. The honeybees have to pass through the box when they enter or leave the beehive. This allows for markerless unobtrusive monitoring of the bees in their natural habitat. The system is a full real-time video monitoring system with both internal camera, power supply, illumination and computer system. It is based on the low-cost Raspberry Pi (Raspberry, 2012) and succeeds in detecting behavioural parameters of the honeybees such as counting bees, identifying their positions and measuring their in-and-out activity at the beehive entrance.

The rest of the paper is organised as follows: Section 2 describes the materials and methods. The experimental results are shown in Section 3. The discussion is given in Section 4.

2. Materials and methods

2.1. Video monitoring system

In this section, we describe the architecture of our video monitoring system, including the hardware equipment and the software architecture. The Raspberry Pi is a credit-card sized computer that costs much less than an ordinary PC (Raspberry, 2012).

2.1.1. Hardware equipment

The extension system box (length: 45 cm, width: 45 cm and height: 22 cm) is named video monitoring unit (VMU) and it serves primarily two purposes: (1) it can control the lighting facilities for image capture; (2) the honeybees enter and leave the beehive through the special passage, which does not allow overlapping of the honeybees.

The inside of the VMU is shown in Fig. 1a. In the inside there is a camera, a Raspberry Pi B+ (i.e. processing unit) that is connected to the Internet, some stable LED light sources with diffusers, and a mirror. The camera is a standard low-cost camera module which is sold with the Raspberry Pi (Raspberry, 2012). The camera is directed towards the mirror that is positioned at a 45-degree angle to view the bees from below. Along the sides of the VMU there are four sets of LED strips, which provide a constant illumination with intensity comparable to lightly clouded daylight. The LED array is white and thus a possible annoyance to the honeybees. However, based on our own experience as well as others (Hendriks et al., 2012), it is not the light itself, but rather changes in the light (switching on or off) that cause changes in the activity of the honeybees. The specific LED array can be found in (LED, 2012). The architecture of the inside is shown in Fig. 2.

The sketch of this system is illustrated in Fig. 1b. The VMU connects the inside of the beehive with the outside, thus the honeybees have to pass the VMU in opposite direction when exiting and entering the beehive. The bottom of the honeybee pool is transparent and covers the mirror. Through the mirror, the camera records the entire passage from the bottom.

Fig. 1c shows the VMU with the unmodified entrance of a standard beehive, and the entrance of the VMU covers the whole entrance of the beehive.

2.1.2. Software architecture

Our system runs Linux on a Raspberry Pi, and the primary programming languages for the system are C++ and MySQL. The primary program library is OpenCV, which is open source (OpenCV, 1999) and used for computer vision. Fig. 3 shows the overall framework of the software, which includes the following four phases:

- Recording data: it captures raw data into h.264 video, which is then converted to mp4 format.
- Video processing: it fetches frames from the video (i.e. mp4) and then segments the frames; counts the honeybees and measures their in-and-out activity based on the segmented images.
- Send and Delete: it sends the experimental results to a MySQL database via Internet and then deletes the h.264 and mp4 videos.
- Result database: it saves the experimental results into the database.

2.2. Material and activity

Two real-time video monitoring systems (see, Fig. 1c) were set up at the research centre Flakkebjerg, Aarhus University, Denmark. Each system recorded 30 s for every thirty minutes (i.e. a video) from Jun. 16, 2014 to Nov. 24, 2014. The type of the honeybees and beehives were *Buckfast* and *Segeberger* with ten frames made of polystyrene, respectively.

Each camera acquired colour images of size 1920×1080 at 25 fps. Using the command *raspivid* of Raspberry Pi, the raw data was recorded to video format h.264, which was then directly converted to mp4 format.

2.2.1. Data used in our experiment

In our experiment, two distinct data types were used: training and test data. 25 and 100 videos were selected for the training and test data, respectively. The training data was used to estimate the parameters of our methods and the test data was used to evaluate our methods. The videos were selected to represent five groups with different numbers of the honeybees in a frame. Otherwise, they were selected randomly from the full set of recordings.

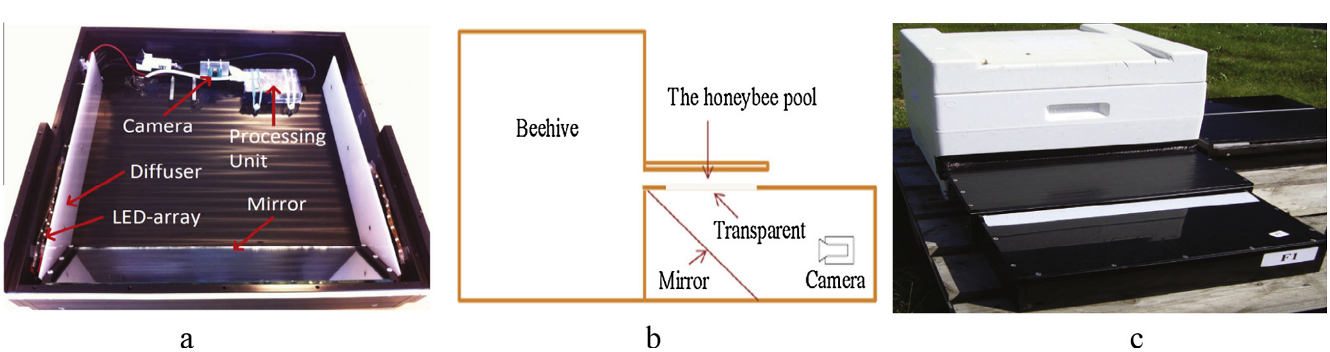

Fig. 1. The automatic detection system: (a) the inside of the VMU, the distance from the diffuser to the bounding box is 4 cm; (b) the sketch of this system, the distance from the entrance of the beehive to the honeybee pool is 3 cm; (c) the VMU connects with a standard beehive.

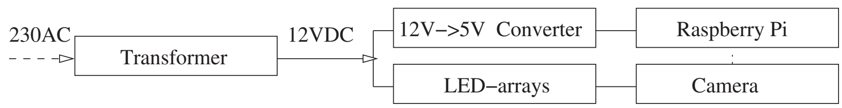

Fig. 2. The inside architecture of the VMU.

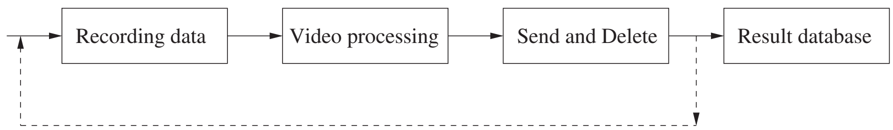

Fig. 3. The four phases of the software architecture in our system: Recording data; Video processing; Send and Delete; Result database.

An overview of the five data groups in our experiment is shown in Table 1. For example, the second column in this table means that there are 25 videos, which were selected, and each scene in these videos had between 6 and 25 honeybees in the frames. Fig. 4 shows an example of frames that was used in our experiment.

As mentioned before, the videos for the algorithm development were recorded for 30 s with a frame rate of 25 fps with an interval of 30 min. However, after consulting honeybee experts, we decided to implement the system with 30 s of recording every 10 min. We therefore had to reduce the frame rate for the videos from 25 to 5 fps for the Raspberry Pi to finish processing of the four phases (see Fig. 3) within the 10 min available.

Fig. 4. An example of bottom-view frames in our data: a honeybee has to pass through the pool to enter or leave the beehive.

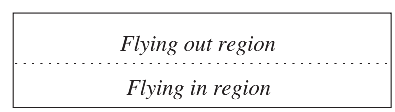

Fig. 5. The honeybee pool is split into two regions: *Flying in* and *Flying out*.

2.2.2. Definitions of the activity

After analysing the movement direction and speed of the honeybees in our data sets, we have observed that a honeybee needs at least 0.5 s. to pass through the pool (see, Fig. 4). This pool is split into two regions for measuring in-and-out activity: namely, *Flying in* and *Flying out*, which are shown in Fig. 5. This division can guarantee that the camera is able to capture a honeybee at least one time (5 fps), if it moves from the outside into the *Flying in* region or from the beehive into the *Flying out* region. The in-and-out activity of the honeybees is defined as follows:

- In-activity: the in-activity increases by one when a honeybee moves into the *Flying in* region from the outside.
- Out-activity: the out-activity increases by one when a honeybee moves into the *Flying out* region from the beehive.

2.3. Methodology

There are two characteristic properties of the proposed system: (1) Illumination is stable; (2) Properties of the foreground objects are fixed (e.g., the colour and size of the honeybees do not change much).

Additionally, from the object detection and tracking perspective, there are major issues regarding shadows and occluding honeybees. However, since Raspberry Pi has limited computational resources and the computation time in our system is restricted, we can only use a simple background subtraction method to segment the images (i.e., the two issues are not dealt with in our segmentation stage). Instead, after the segmentation stage, the methods for counting honeybees and measuring in-and-out activity are primarily based on statistical analysis and inference, respectively.

Table 1
An overview of the five data groups used in our experiment. NH: the number of honeybees; G: group.

	G1	G2	G3	G4	G5
NH	0–5	6–25	26–45	46–65	>65
Training	5	5	5	5	5
Test	20	20	20	20	20

2.3.1. Counting honeybees

Our approach for counting honeybees has two stages: segmentation and counting. Since we do not attempt to remove shadows or split connected honeybees, the counting of honeybees is based on linear regression on the foreground areas. The flowchart of the approach is shown in Fig. 6.

The segmentation stage is based on the background subtraction technique. First, the current colour image is converted to a grayscale image and the mean over all pixel intensities in the grayscale image is calculated (Note: the mean decreases, when the number of honeybees increases, since the illumination is fixed). Second, according to the calculated mean, a threshold is chosen and used to segment the image.

In the counting stage, the number of honeybees is counted by using the segmented image. If the total number of foreground pixels in the binary image is greater than a manually determined threshold (corresponding to 45 honeybees in the image), the number of honeybees is calculated by using a linear regression equation (i.e. Eq. (1)), which is manually estimated:

$$y_1 = a_1 \cdot x_1 + b_1, \tag{1}$$

where y_1 is the number of the honeybees in the image, x_1 is the total area of foreground objects (i.e. the total number of foreground pixels) in the segmented image, and a_1 and b_1 are parameters.

We use the area of the individual segmented foreground objects to count the honeybees, when the total number of foreground

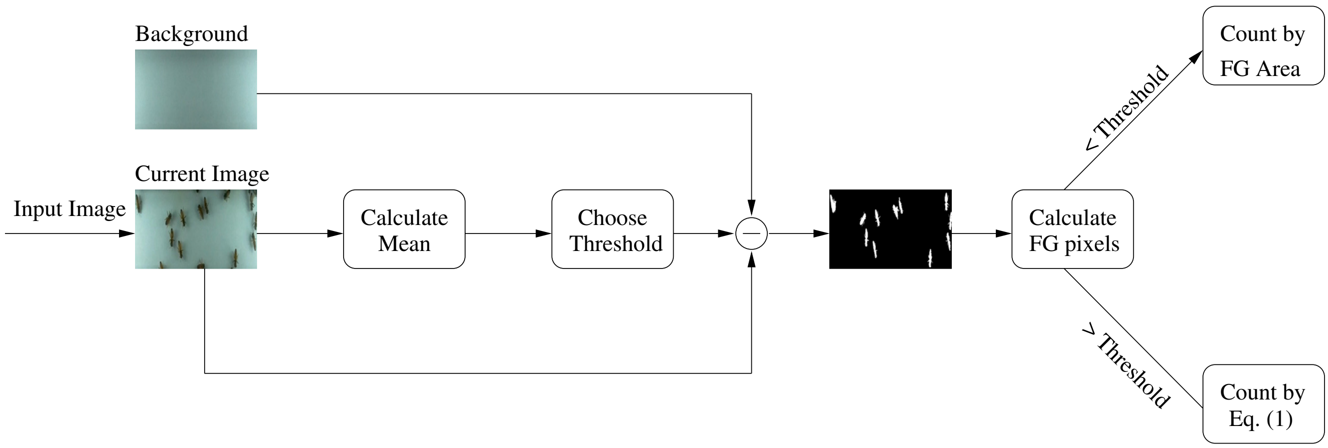

Fig. 6. Two stages in our counting approach. (1) Segmentation stage: we segment the honeybees and the binary image shows the segmented foreground objects (FG). The areas of FG are used to count the honeybees. (2) Counting stage: after calculating the FG pixels, we count the honeybees using either Eq. (1) or the algorithm that is described in Table 2.

pixels in the binary image is less than the manually determined threshold. The pseudo-code is shown in Table 2. Since the shadows are not dealt with in the segmentation stage and can be included within the foreground pixels, we have to manually find a linear regression equation, which can express the relation between the number of the honeybees in the individual foreground object and its corresponding area. The equation is given as follow:

$$y_2 = a_2 \cdot x_2 + b_2, \tag{2}$$

where y_2 is the number of the honeybees in the individual foreground object, x_2 is the corresponding area, and a_2 and b_2 are parameters.

In Section 2.3.3, we describe how to estimate the four parameters (a_1, b_1, a_2, b_2) in Eq. (1) and (2). The thresholds in Table 2 are estimated based on the analysis of the segmented images.

2.3.2. Measuring activity

It is difficult to measure the in-and-out activity that is defined in Section 2.2.2. In order to limit the computation time, the segmented image is resized to 375×96 pixels (i.e., the width W and height H of the image are 375 and 96, respectively).

We first estimate the position and direction of honeybees in the resized image. The flowchart is shown in Fig. 7A. For the individual foreground objects, a new linear regression equation Eq. (3) is manually estimated:

$$y_3 = a_3 \cdot x_3 + b_3, \tag{3}$$

Table 2
The pseudo-code for counting honeybees by using the area of the foreground objects: this corresponds "Count by FG Area" in Fig. 6.

```
   // input the segmented image
1. Input(Segment)
   // the number of honeybees in the image
2. int totalBees = 0;
3. for i = 1: the number of objects
     // remove noise and small objects
   4. if (the area of objects[i] > minArea)
     // the number of honeybees in the object
     int countBees = 0
     if (the area of objects[i] < maxBeeSize)
       countBees = 1
     else// using Eq. (2)
       countBees=a2·(the area of objects[i])+b2
     totalBees = totalBees + countBees
   end if
  end for
5. Return totalBees
```

where y_3 is the number of the honeybees in the individual foreground object in the resized image, x_3 is the corresponding area, and a_3 and b_3 are the parameters that are estimated in Section 2.3.3.

To measure the in-and-out activity, the temporal contextual information (e.g. the position and direction of honeybees) is used to track the foreground objects between the previous and current frames. An illustration for this construction is shown in Fig. 7C and D. Each circle in the sub-figures represents a honeybee.

In Fig. 7C, the red circle represents a honeybee in current frame that will be evaluated as a new honeybee or not. To evaluate the red circle we make two criterion to find correspondences of the red circle: Backward and Forward. In "Backward criteria" (i.e. line 2), the green circles represent the honeybees in the previous frame which correspond to the red circle (i.e. these are potential candidates for the red circle in the previous frame). The blue circle in line 2 is a simulation honeybee which does not appear in the previous frame. It is possibly located at the outside or beehive and also a potential candidate for the red circle. In "Forward criteria" (i.e. line 3), the circles represent the honeybees in the current frame that correspond to the circles in the previous frame (i.e. each circle in line 2 possibly moves to the corresponding circles in line 3).

In Fig. 7D, a honeybee B1 is located in the *Flying in* region in the current frame and it will be evaluated as a new honeybee or not. First, we find the correspondences of B1 in the previous frame using "Backward criteria". Within a radius of R around B1 in the previous frame A1 and A2 are located. Also, we create a simulation honeybee S (i.e. the blue circle). Next, we find the correspondences of A1, A2 and S in the current frame using "Forward criteria", and the result is shown in the lower left corner in the sub-figure.

Now we describe the circles and the two criterion in more detail. Each coloured circle in the sub-figures is modelled using an ellipse which includes the centroid and direction. The direction is given by the angle between the positive direction of the x-axis and the major axis of the ellipse (i.e. angle $\in [0, \pi]$). The x- and y-axes are horizontal and vertical, respectively. The following describes these circles, where (x_0, y_0) is the centroid of the red circle, L is the simulated length of one step of the honeybee, and H and W are the height and width of the resized image, respectively.

- The red circle: it is a possible new honeybee in the current frame, which has just moved in the *Flying in* region or *Flying out* region from the outside or beehive, respectively.
- Backward criteria: the criteria is used to select the green circles in the previous frame, which correspond to the red circle in the current frame. The green circles are selected as follows:

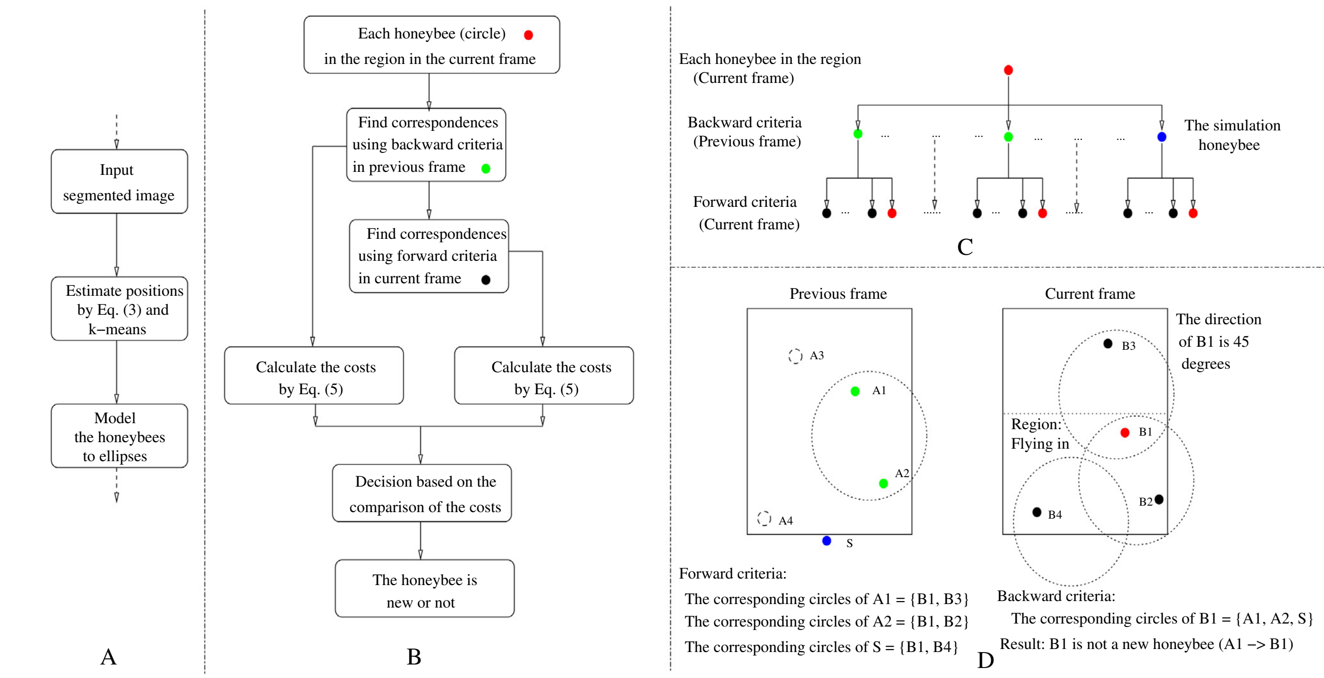

Fig. 7. A: the flowchart for Model ellipses – the position and direction of honeybees; B: the flowchart for Evaluation – decide the honeybee in the current frame is a new honeybee or not; C: an illustration for the temporal contextual information about the honeybees between the previous and current frames, the four red circles represent the same honeybee which will be evaluated; D: an example for how to process in the evaluated approach, i.e., how to evaluate whether B1 is a new honeybee or not: B1 → find correspondences using "Backward criteria" → find correspondences using "Forward criteria" → Calculate the costs → Decision → Result. The circles A3 and A4 are not within this evaluation. B1 is not identified as a new honeybee. The movements between the previous and current frames: A1→B1; A2→B2; A3→B3; A4→B4.

  - First, we find the green circles in the previous frame which are within a circular region of the red circle with the radius R.
  - Second, we remove the green circles that satisfy all the following conditions:
    - the directions of the red and green circles are similar (i.e. the two ellipses are parallel),
    - and the two corresponding major axes (of the two ellipses) don't fall on the same straight line,
    - and the distance between the two centroids is less than L.

- The blue circle: it is a simulation honeybee which does not appear in the previous frame.
  - The direction θ of the blue circle is equal to the direction of the red circle.
  - The x-value of centroid of the blue circle is simulated as follows:

$$x = \begin{cases} 0, & \text{if } x_0 + \mathrm{L} \cdot cos(\theta) < 0; \\ \mathrm{W}, & \text{if } x_0 + \mathrm{L} \cdot cos(\theta) > \mathrm{W}; \\ x_0 + \mathrm{L} \cdot \cos(\theta), & \text{otherwise}. \end{cases}$$

  - The y-value of centroid of the blue circle is simulated as follows. For the *Flying in* region: if $y_0 + \mathrm{L} \cdot \sin(\theta) < \mathrm{H}, y_{in} = \mathrm{H}$; otherwise, $y_{in} = y_0 + \mathrm{L} \cdot \sin(\theta)$. For the *Flying out* region: if $y_0 - \mathrm{L} \cdot \sin(\theta) > 0, y_{out} = 0$; otherwise, $y_{out} = y_0 - \mathrm{L} \cdot \sin(\theta)$. Note that there are 3 cm from the entrance of the beehive to the honeybee pool (see Fig. 1).
- Forward criteria: the criteria is used to select the black circles in the current frame, which correspond to the green circles in the previous frame. The black circles are selected in the same manner as green circles.

From our data analysis, the highest length of one step of a honeybee is approximately 30 pixels. Thus, the simulated length of one step of a honeybee L is set to 20 pixels and the radius R is set to 40 pixels.

In order to evaluate the red circle, based on the motion of honeybees a cost function is defined. We first define the following variables that are issues of central importance in the cost function:

- The honeybees in the passageway ($\mathtt{N}$): we simulate a passageway between the centroid of two honeybees (i.e. the two black circles), which is shown in Fig. 8. The width of the passageway is set to 12 pixels. This $\mathtt{N}$ represents how many honeybees are in the passageway (i.e., the other circles in the passageway in the figure).
- Time ($\mathtt{T}$): the time represents how long a honeybee takes to pass through the empty passageway (i.e., there are no honeybees in the passageway). The time $\mathtt{T}$ is defined by the following equation:

$$\mathtt{T} := \frac{\text{Distance between the two black circles}}{\mathrm{R}}, \tag{4}$$

where R is the radius. The time $\mathtt{T}$ is always less than or equal to 1 and increases when the distance increases in the walking.

Let A and B represent the honeybees in the previous and current frames, respectively. The cost function for the motion ($A \to B$) is defined by

$$C(A \to B|(\theta_a, \theta_b, \mathtt{N}, \mathtt{T})) := \frac{1 - \mathtt{T}}{\mathtt{N} + 1} \cdot \begin{cases} 0, & \text{if } \cos(\theta_a - \theta_b) < 0, \\ \cos(\theta_a - \theta_b), & \text{otherwise}, \end{cases} \tag{5}$$

where θ_a and θ_b are the directions of A and B, and $\mathtt{N}$ and $\mathtt{T}$ are defined as before. This definition indicates that the cost increases, when the motion duration decreases.

We track frame-to-frame changes in the centroid and direction of the honeybees to measure their in-and-out activity. For each frame we have two steps: Model ellipses and Evaluation, which are described as follows.

Model ellipses: The flowchart for this step is shown in Fig. 7A. First, noise and small objects are removed by a threshold in the current resized image. Second, in the binary image we calculate the area of each foreground object. Based on the area we determine whether the foreground object contains a single honeybee or not by using a threshold:

(1) If the foreground object only contains a single honeybee, it is modelled to an ellipse. The centroid of the ellipse is calculated by the centre of mass of the foreground object and the major and minor axes of the ellipse are modelled by using the moments of the foreground object.
(2) Otherwise, we estimate the centroids and model the honeybees to ellipses.
  - Centroids: we use the k-means algorithm to estimate the centroid of the honeybees by using all coordinates in the foreground object, where k is the number of honeybees which is calculated by using Eq. (3).
  - Model ellipses: each honeybee (i.e. each centroid) that is nearly located in the boundaries of the foreground object is modelled to an ellipse. We find an appropriate convex point in the foreground object which is close to the centroid. The centroid of the ellipse is the same as the centroid, and the direction of the ellipse is estimated by using the two points (i.e. the centroid and convex point). Note that we do not model an ellipse for the centroids located in the middle of the foreground object, since these are most likely not new honeybees.

Evaluation: The flowchart for this step is shown in Fig. 7B. The approach for the evaluation of the red circle in *Flying in* region is described as the below and the approach for *Flying out* region is the same.

Each honeybee that was modelled to an ellipse in the *Flying in* region in the current frame is seen as a red circle. To determine whether the red circle is a new honeybee or not, we do the following steps:

(1) The corresponding blue circle is simulated. The corresponding green and black circles are found in the previous and current frames, respectively. See Fig. 7C.
(2) If there are no green circles in the previous frame, the red circle is identified as a new honeybee.
(3) Otherwise, using Eq. (5) we calculate the cost for each pair of circles that is shown in Fig. 7C:
  - The costs in "Backward criteria" = {C(each circle in line 2 $\rightarrow$ the red circle in line 1)}.
  - The costs in "Forward criteria" = {C(each circle in line 2 $\rightarrow$ the corresponding circle in line 3)}.

If the cost C(the blue circle $\rightarrow$ the red circle) is simultaneously highest in "Backward criteria" and "Forward criteria", then the red circle is identified as a new honeybee in the current frame. This means that the simulation honeybee in the previous frame corresponds to a new honeybee in the current frame.

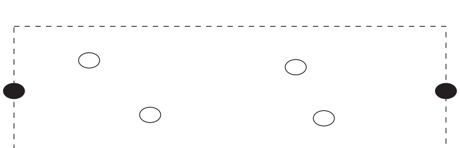

Fig. 8. A passageway between the centroid of two honeybees (i.e. the two black circles). The white circles represent the honeybees in the passageway.

Since we cannot distinguish between the head and tail of the honeybee, if $\theta_a = \theta_b$ and the two corresponding major axes fall on the same straight line and the value of $\mathtt{N}$ is fixed, then the cost in Eq. (5) increases when $\mathtt{T}$ decreases. In this case, a failed detection has possibly occurred.

An example for how to process the evaluated approach is shown in Fig. 7D.

2.3.3. Parameter estimation

The parameters of the three linear regression equations as described before are estimated based on the segmented images in the training data.

As mentioned before, Eq. (1) is a relation between the number of honeybees and the total area of foreground objects in the segmented image. To estimate the two parameters, we selected the original images in the training data and manually counted the honeybees in each image. The range for the number of honeybees was from 40 to 90 and for each one (almost) in the range at least 3 images were counted. The total area of the foreground objects was calculated by using the segmented image.

Eqs. (2) and (3) are the relations between the number of honeybees and the area of the corresponding object in the original and resized segmented images, respectively. For both the original and resized segmented images, we selected the objects which had the honeybees ranging from 2 to 8. We manually counted the honeybees and measured the area in each object.

This estimation is plotted in Fig. 9, the x-axis is the area and the y-axis is the number of honeybees that is manually counted. The estimated parameters are given in Table 3. The three equations increase the computing speed in our system.

3. Experimental results

In this section, we will show our experimental results, which are obtained by using our test data.

3.1. Qualitative evaluation

To test the effectiveness of our simple segmentation approach, we evaluate qualitatively the segmented images. Since there are five data groups in the study, we have selected five results which represent the general results of each group. The results for segmentation and the position of the honeybee are shown in Fig. 10, which shows that the shadows do not significantly affect the counting and positioning of the honeybees. This qualitative evaluation demonstrates that the segmented images can be used to count the honeybees and measure the in-and-out activity.

An example for the elliptical modelling is shown Fig. 11. Not all ellipses are perfectly aligned. This is because the corresponding convex point in the foreground object is not correctly selected due to the influence of the shadows.

In video processing, it is difficult to deal with the connection of the honeybees. Therefore, the elliptical modelling by using the segmented images seems like a good and practical idea.

3.2. The number of honeybees

To evaluate the predicted result for counting honeybees, over 400 original images were randomly selected to manually count the honeybees in the images. The range of the number of honeybees in the individual images was from 0 to 90, and at least 3 images were selected for each one (almost) in the range. The comparison between the predicted (x-axis) and manual (y-axis) results is shown in Fig. 12. The linear regression equation for this comparison is given by

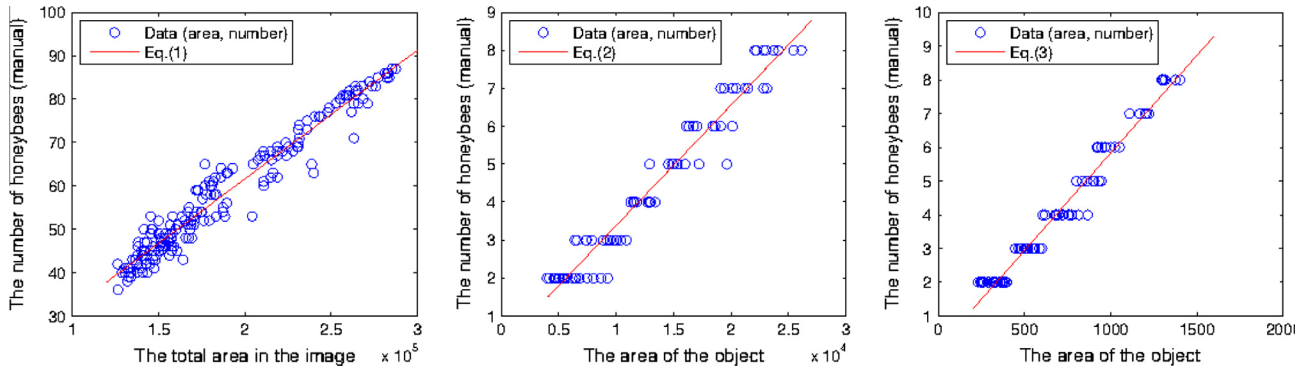

Fig. 9. The three linear regression equations: left – Eq. (1); middle – Eq. (2); right – Eq. (3). *x*-axis: area; *y*-axis: the number of honeybees.

Table 3
The estimated parameters for the three linear regression equations. The R_i^2 is calculated by using the residual variance.

Eq. (1)			Eq. (2)			Eq. (3)		
a_1	b_1	R_1^2	a_2	b_2	R_2^2	a_3	b_3	R_3^2
0.0003	2.2696	0.9502	0.0003	0.2038	0.9503	0.0058	0.0501	0.9705

$$y = 1.0073 \cdot x + 0.2589, \tag{6}$$

and the R^2 for this equation is 0.9868. This result demonstrates that our system succeeds in counting honeybees at the entrance of the beehive.

3.3. In-and-out activity

As mentioned, it is difficult to measure the in-and-out activity. To evaluate our results for measuring in-and-out activity, we randomly selected 15 videos in the first 3 data groups (i.e. 5 videos for each group) to manually label the in-and-out activity.

Fig. 13 shows an example of a typical result for an individual video. The *y*-axis is the accumulated number of the in-activity or out-activity by the frame-to-frame changes and the *x*-axis is the frame number. The red line is for manual results and the blue line is for the predicted results.

The comparison between the two accumulated numbers (i.e. predicted and manual) for the 15 videos is shown in Fig. 14. The two linear regression equations are given by

$$y = 1.0124 \cdot x - 2.3555, \quad \text{and} \quad R^2 = 0.9525, \tag{7a}$$

$$y = 1.0300 \cdot x - 3.6029, \quad \text{and} \quad R^2 = 0.8882. \tag{7b}$$

The first sub-equation is for the in-activity and the second sub-equation is for the out-activity. The result for the in-activity is better than the result for the out-activity. This is because there is more noise and shadow in the entrance of the beehive.

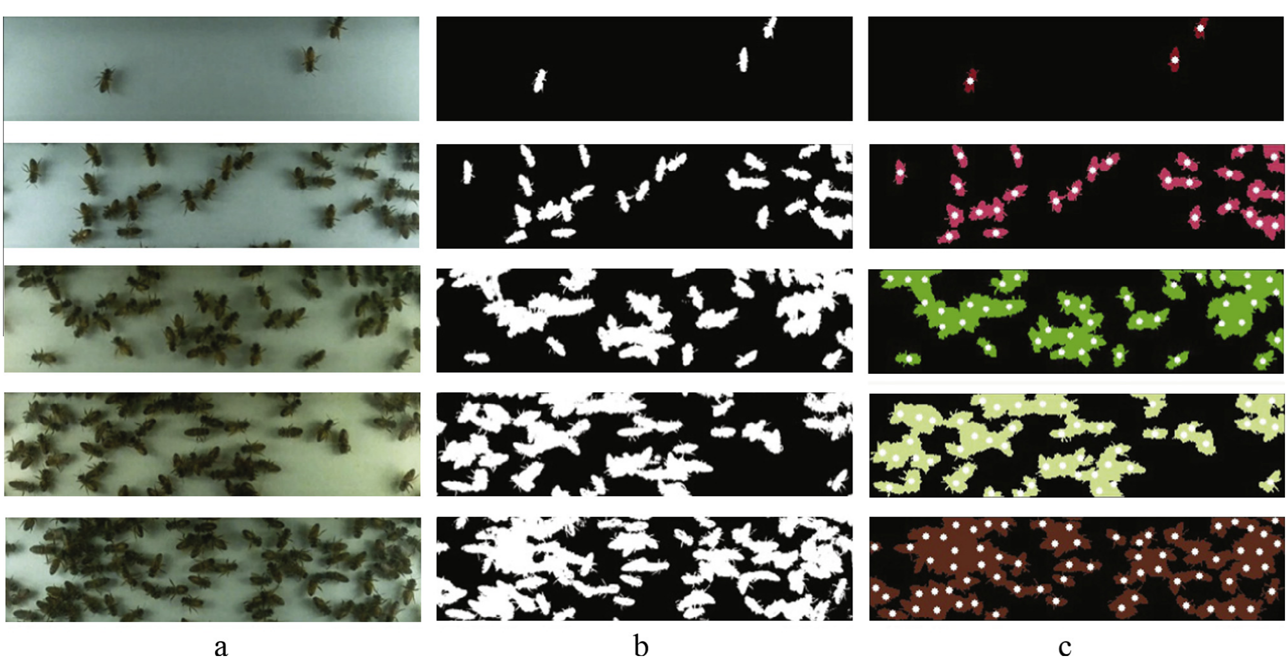

Fig. 10. The general results for segmentation and the honeybee position in the different data groups: (a) the original image, (b) the segmented image; (c) the predicted position of the honeybees found from our algorithm.

Fig. 11. The general results for the elliptical modelling: left – the original image; right – the simulated ellipses.

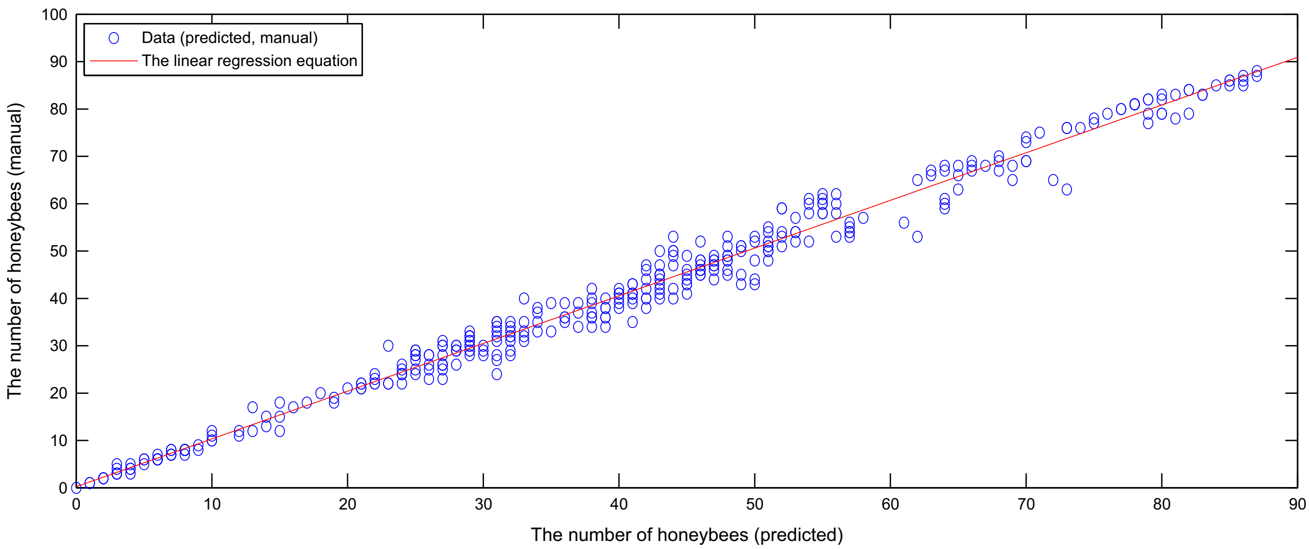

Fig. 12. The comparison between the predicted and manual results for the number of honeybees; *x*-axis: predicted result; *y*-axis: manual result.

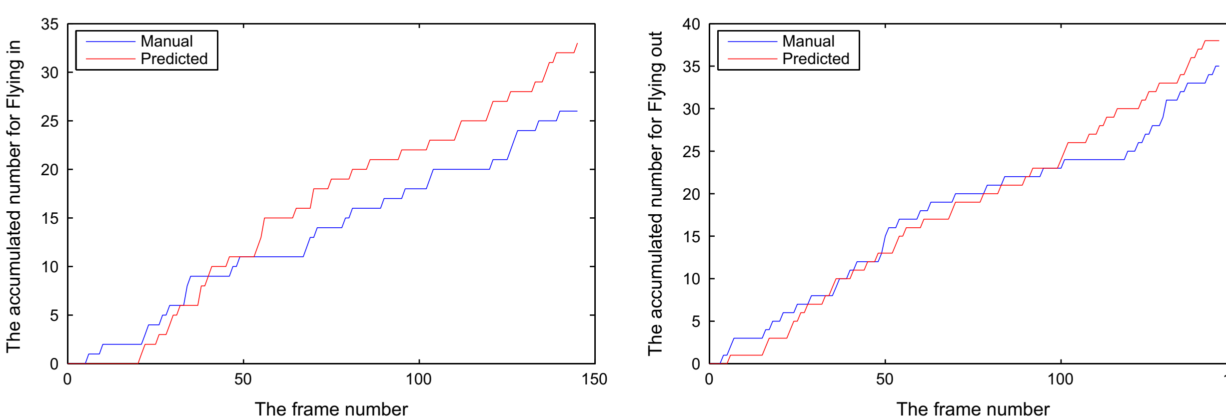

Fig. 13. Example of a typical result from a single video when measuring in-and-out activity: the left is for *Flying in* and the right is for *Flying out*. *x*-axis: the frame number, *y*-axis: the accumulated number of the in-activity or out-activity, the red line – predicted and the blue line – manual. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Note that group 4 and 5 are not included in the activity study, because our method for measuring the in-and-out activity does not work in the most videos due to the extreme densities of honeybees. Some of the videos are almost impossible to manually label the activities.

3.4. Time consumption

Our goal is to make a real-time system by using Raspberry Pi, which executes as fast as possible. As described in Section 2.1.2, our software architecture consists of four phases. The first two phases (i.e. Recording data and Video processing) are the most time consuming. A video in our data included 150 frames (i.e. recorded 30 s with 5 fps) and the maximum time consumption for the two phases for a video (among 125 videos) is shown in Table 4. As one can see in the table, the total maximum time consumption is less than ten minutes.

4. Discussion

We have focused on monitoring without much interference to the beefamily. Most previous systems may impede the natural motion of the honeybees such as tunnel systems that the honeybees have to follow (such as predominant photodiode-based

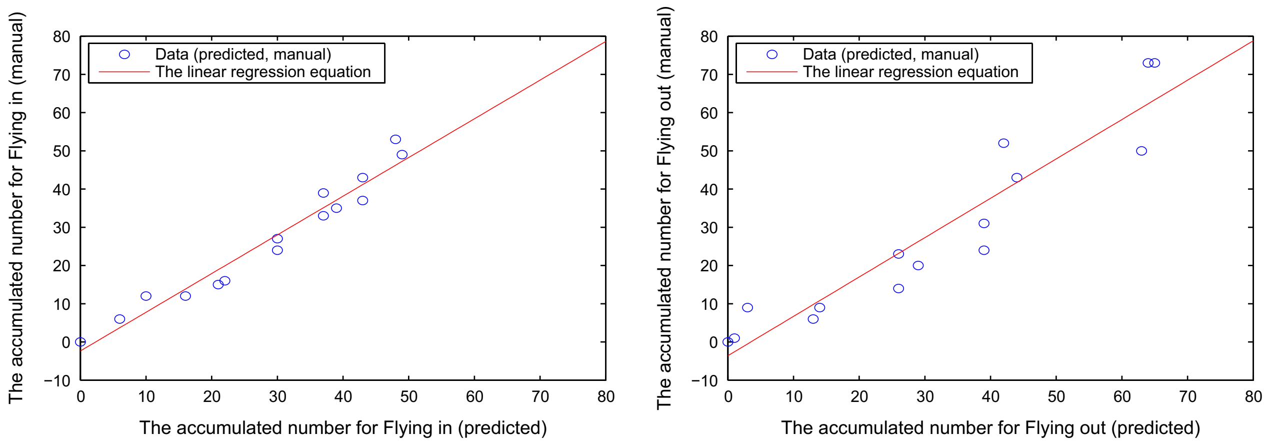

Fig. 14. The comparison between the two accumulated numbers (*x*-axis: predicted and *y*-axis: manual); left: the in-activity; right: out-activity.

Table 4
The maximum time consumption in the different phases for a video in our data. The unit of time is second.

	Recoding data	Video processing			Total times
	Capturing & converting	Fetching	Segmenting & counting	In-and-out activity	
Time	96	131	58	282	567

systems (Struye et al., 1994; BeeSCAN, 2005) and where the tunnels may get blocked by e.g. dead honeybees). We also observed various phenomena in our recordings such as drone eviction, wasp attacks and that occurred unhindered by our system. Other successful previous vision systems are marker-based systems that directly track individual honeybees, but aimed at researchers rather than commercial beekeepers. We demonstrate a viable solution for a complete online real-time vision system that could be used by e.g. commercial beekeepers due to low cost.

We have developed and verified a system that is both capable of counting honeybees in individual images and monitoring in-out activity. It is still unclear how these measures can be fully comprehended in relation to honeybee behaviour, but we have experienced situations where the two measures supplement each other well. For instance, the number of dead honeybees will occur as a static count offset without any change in in-out activity. We also observe daily variations in relation to honeybee count that seem to relate to climatic variables and pre-swarming and drone eviction stages also appear to involve a high ratio between honeybee count and activity although we do not have firm evidence for this hypothesis. One of the main problems with the current system is for very high densities of honeybees where we cannot measure in-out activity and also have problems with counting. Certainly, stronger algorithms and better image quality might solve some of the problems. However, we focused on a realistic low-cost system and therefore we had strong restrictions on camera quality and computer performance.

We show that it is possible to carry out all different phases of the data processing within the 10 min of available time between recordings. Still, better designed illumination may remove some of the shadows with only small changes to the current design and we could also have programmed the Raspberry Pi more efficiently by using e.g. the built-in graphics engine.

Acknowledgement

The study is part of the project "The Application of Information Technologies in Precision Agriculture (ITApic)", financed by the "ICT-AGRI 2012".

References

Adeva, J., 2012. Simulation modelling of nectar and pollen foraging by honey bees. Biosyst. Eng. 112, 304–318.
BeeSCAN, 2005. Lowland electronics bvba. <http://users.pandora.be/lowland> (accessed on October 2014).
Campbell, J., Dahn, D., Ryan, D., 2005. Capacitance-based sensor for monitoring bees passing through a tunnel. Measure. Sci. Technol. 16, 2503.
Campbell, J., Mummert, L., Sukthankar, R., 2008. Video monitoring of honey bee colonies at the hive entrance. Visual Observ. Anal. Animal Insect Behav., ICPR 8, 1–4.
Chen, C., Yang, E., Jiang, J., Lin, T., 2012. An imaging system for monitoring the in-and-out activity of honey bees. Comput. Electron. Agr. 89, 100–109.
Chica, M., 2012. Authentication of bee pollen grains in bright-field microscopy by combining one-class classification techniques and image processing. Microsc. Res. Tech. 75 (11), 1475–1485.
Chica, M., Gomez-Kramer, P., Menard, M., Requier, F., 2013. 3d tracking of honeybees enhanced by environmental context. In: Image Analysis and Processing, ICIAP 2013, pp. 702–711.
Hendriks, C., Yu, Z., Lecocq, A., Bakker, T., Locke, B., Terenius, O., 2012. Identifying all individuals in a honeybee hive – progress towards mapping all social interactions. Workshop Vis. Observation Anal. Anim. Insect Behav. ICPR 2012.
Henry, M., Beguin, M., Requier, F., Rollin, O., Odoux, J., Aupinel, P., Aptel, J., Tchamitchian, S., Decourtye, A., 2012. A common pesticide decreases foraging success and survival in honey bees. Science 336 (6079), 348–350.
Kimura, T., Ohashi, M., Okada, R., Ikeno, H., 2011. A new approach for the simultaneous tracking of multiple honeybees for analysis of hive behavior. Apidologie 42, 607–617.
LED, 2012. <http://www.farnell.com/datasheets/1915205.pdf> (retrieved October 2014).
OpenCV, 1999. <http://opencv.org/> (retrieved October 2014).
Raspberry, 2012. <http://raspberrypi.org/> (retrieved October 2014).
Russell, S., Barron, A., Harris, D., 2013. Dynamic modelling of honey bee (apis mellifera) colony growth and failure. Ecol. Modell. 265, 158–169.
Sagili, R., Burgett, D., 2011. Evaluating Honey Bee Colonies for Pollination. Pacific Northwest Extension Publications.
Struye, M., Mortier, H., Arnold, G., Miniggio, C., Bornek, R., 1994. Microprocessor-controlled monitoring of honeybee flight activity at the hive entrance. Apidologie 25, 384–395.