

# APPLICATION OF AN ADAPTIVE BACKGROUND MODEL FOR MONITORING HONEYBEES

Uwe Knauer<sup>1</sup>, Michael Himmelsbach<sup>1</sup>, Frank Winkler<sup>1</sup>, Fred Zautke<sup>2</sup>, Kaspar Bienefeld<sup>2</sup>, Beate Meffert<sup>1</sup>

<sup>1</sup> Humboldt-Universität zu Berlin, Institut für Informatik, 10099 Berlin

<sup>2</sup> Länderinstitut für Bienenkunde, Hohen-Neuendorf

{knauer, mhimmels, meffert}@informatik.hu-berlin.de

## ABSTRACT

Results for the application of an adaptive background model to the problem of detecting changes on a combs surface are reported. It is demonstrated that the combined search for uncapped brood cells in the current image and the background image increases the overall detection rate of the system.

## KEY WORDS

Varroa, background, modeling, application, infrared, honeybee

## 1. Introduction

One of the biggest threats for the native honeybee *Apis mellifera* is the mite *Varroa destructor* [1]. The varroa mites are external honeybee parasites. As mites develop on the bee brood, which often results in a deformation of the bees and a drop in honey production, the infestation of a colony is a serious problem. Until now, it is not possible to cure an afflicted colony.

There are different strategies for blocking the mites. For instance, the apiarists can use acaricide agents like formic acid. However, since the first acaricide resistant mites have been reported, there is an urge to develop alternative methods, in particular because acaricides may lead to unwanted residues in wax and honey.

One of the most promising approaches is the rearing of resistant bees. Therefore, current research in the field of apiculture focuses on the genetic selection of hygienic bees [2]. Hygienic behaviour is characterized by three components: finding dead or damaged brood quickly, uncapping dead or damaged brood, and removing dead or damaged brood from the cells. The selection of hygienic bees has so far required a time consuming observation of the combs. Processing all the material that is typically recorded for a period of one week (24 hours a day) requires at least twice the time for analysis by a human expert. Therefore, it would be very helpful to develop algorithms for an automated observation of the combs as well as for the detection of the hygienic bees. In this paper, an image processing approach is presented which supports the human observer.

## 2. Experimental setup

To develop such image processing algorithms a test beehive was installed at the Länderinstitut für Bienenkunde (Institute of Apiculture) in Hohen-Neuendorf. In order to keep the ambient conditions near to the normal living conditions of the bees, the test comb, with 25 artificially varroa infested brood cells, was equipped with near-infrared illumination consisting of LEDs which are mounted on a wooden frame. The behaviour of the bees will not be influenced by this illumination, because bees are blind in this spectral range. The observation is done by infrared cameras. Fig. 1 shows a top view of the experimental setting.



Fig. 1. Experimental setup from left to right: camera, illumination frame, comb

To be able to identify an individual bee, they are manually labeled with numbered badges. The acquired image sequences are stored on a digital long-time videotape recorder using a resolution of 480x348 pixels. Image processing is done with the Open Computer Vision Library [3] and the C++ programming language.

A first step towards an automatic recognition of hygienic bees is the detection of uncapped cells. If such a cell is found, a (backward) search for the initial opening can be performed to identify the bee(s) first active in the uncapping process. The main difficulty in the detection of

open cells is the crowd of bees. The comb has to be populated with 2000 bees to keep the conditions quite normal. Hence, the open cells are often covered by bees and are only rarely visible. However, this can be handled if it is possible to find a background model appropriate for this application.

### 3. Modeling image background

The surface of the comb can be regarded as a static image background with bees moving or resting in the foreground. The event of uncapping a cell results in a change of the background, and thus a background model is required which allows for updates. Modeling an image background is a typical problem that emerges in many applications like visual surveillance and traffic monitoring [4, 5].

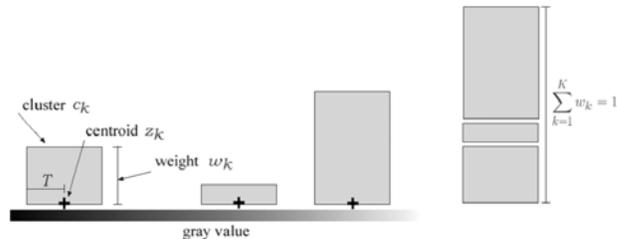
The simplest approach is to initialize the background model with an image that does not contain any bees. However, a set of suitable rules is required in order to update all regions of the background currently visible.

One adaptive method for background estimation is to calculate an average image by using the moving average of the gray values of each pixel in a limited sequence of images. This corresponds to a low-pass filter operation applied to the changing gray values. By choosing the values of the filter coefficients, we can determine which period of a temporal change will become part of the background. Another method of image background estimation is the principal component analysis (PCA) [6]. The update of such a model is computationally expensive, because a high dimensional PCA has to be performed. This method is not well-suited for monitoring bees on combs, because most of the time only small parts of the background are visible. Therefore, the principal components will reflect the foreground rather than the background.

Statistical models seem to be more adequate for this comb model. Modeling the background statistics as a weighted sum of Gaussian functions [7] allows the system to adapt to different states of the background. In comparison to the moving average approach, the model recovers faster to previous states of the background. This situation frequently occurs when a bee has rested for a long time and starts moving again. Modeling the background with kernel functions can be seen as an extension of this approach [8]. Computational power and memory requirements still limit the application of statistical models. Several approaches are published to overcome these limitations and allow real-time processing. Here the method of Butler et al. [9] is used.

The basic idea of this method is to model the distribution of the changing gray values of each pixel with  $K$  clusters. A cluster  $c_k$  is defined by its centroid  $z_k$  and its weight  $\omega_k$ . The weights  $\omega_k$  are initialized with a default value  $\omega_{new}$  and are updated for every new frame as it will be described later. The sum of the weights is always equal to one. For faster computation, all clusters have the same

width, which is defined by a fixed threshold  $T$ . Fig. 2 shows a background model for  $K=3$  clusters. The weight of each cluster is illustrated by its height. If a new gray value  $g_t$  of a pixel is available at time  $t$ , the pairwise distances between  $g_t$  and the centroids of all clusters are compared.



**Fig. 2. Approximation of the gray value distribution of a single pixel with three clusters**

If the condition  $|z_k - g_t| < T$  is fulfilled,  $g_t$  lies inside the cluster  $c_k$ . The gray value  $g_t$  is allowed to match only one cluster. If the clusters overlap then the condition may be true for more than one cluster, and in this case the cluster with the highest weight is chosen. If  $g_t$  does not lie inside any cluster, it replaces the centroid of the cluster with the smallest weight.

Then, the weights  $\omega_k$  of all clusters have to be updated. To indicate different points in time the weights will get a second index  $t$  and  $t-1$ . If a cluster  $c_k$  was matched, its new weight  $\omega_{k,t}$  is calculated by Eq. (1).

$$\omega_{k,t} = \omega_{k,t-1} + \frac{1}{L}(1 - \omega_{k,t-1}) \quad (1)$$

The integer  $L$  controls the adaptation rate. The weights of the clusters that were not matched are decreased using Eq. (2).

$$\omega_{k,t} = \omega_{k,t-1} - \frac{1}{L}\omega_{k,t-1} \quad (2)$$

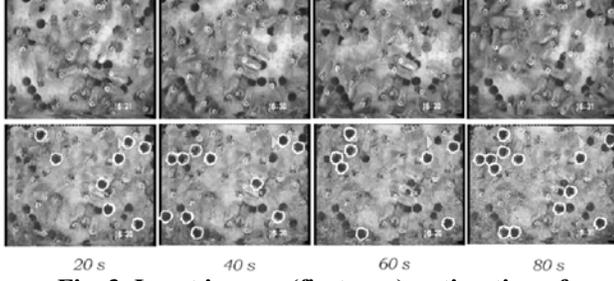
Butler et al. suggested not to update the centroids for every new frame. The differences between the current gray value  $g_t$  and the centroid  $z_{match}$  of the matched cluster  $c_{match}$  should be accumulated, i.e. the centroids should not be adjusted before this value exceeds  $L-1$  or falls below  $-L$ . Finally, the weights have to be normalized such that their sum is 1 again. The (new) normalized weight is:

$$\hat{\omega}_k = \frac{\omega_k}{\sum_{k=1}^K \omega_k} \quad (3)$$

To decide whether a pixel should be labeled as foreground or as background a value  $P$  is calculated using Eq.(4). In other words,  $P$  sums only the weights of unmatched clusters that have higher weights than the matched cluster.

$$P = \sum_{k=1}^K \alpha_k \cdot \omega_k, \text{ where } \alpha_k = \begin{cases} 1, & \text{if } \omega_k > \omega_{match} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

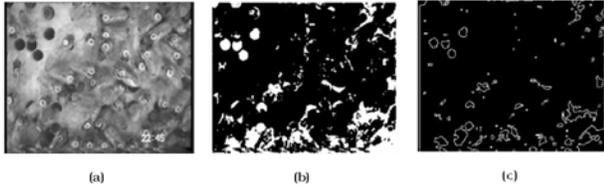
The pixel belongs to the foreground if the value of  $P$  is greater than a given threshold. Fig. 3 illustrates the application of this background estimation method for different times. The white circles in the second row mark the uncapped cells that are not visible in the current image. The details of the detection process are described in the next section of this paper.



**Fig. 3. Input images (first row), estimation of background (second row), and uncapped cells that are not visible in the input images (white circles)**

#### 4. Detection of uncapped cells

An obvious feature of the uncapped cells is their low gray value and the hexagonal structure. Fig. 4 illustrates a binarization which allows the extraction of cells.



**Fig. 4. Current frame (a), binary image (b), and edge image (c)**

From the resulting binary image, the contours of the objects can be extracted. For this the Canny edge detection algorithm [10] is used in combination with the Teh-Chin chain approximation algorithm [11]. Fig. 4c shows the result. After edge detection, five features are derived which describe each contour. The corresponding measures are calculated as follows. Eq.(5) defines a measure for the ratio of the main axes of the contour. This measure  $ra$  is calculated from the elements of the covariance matrix COV.

$$ra = \frac{COV_{2,2} + COV_{1,1} - \sqrt{(COV_{2,2} + COV_{1,1})^2 - 4 \cdot (COV_{1,1} \cdot COV_{2,2} - COV_{1,2}^2)}}{COV_{2,2} + COV_{1,1} + \sqrt{(COV_{2,2} + COV_{1,1})^2 - 4 \cdot (COV_{1,1} \cdot COV_{2,2} - COV_{1,2}^2)}} \quad (5)$$

Eq.(6) is used to calculate the coordinates of a center point  $\mu$  of the contour. The coordinates of  $\mu$  are defined as the average coordinates of all contour points  $P_i$ .

$$\mu = \frac{1}{N} \sum P_i \quad (6)$$

The feature  $cdev$  describes the deviation from a circle. For that Eq.(7) calculates the mean square error of all points compared to a circle with center point  $\mu$  and radius  $\mu_r$ . The value of  $\mu_r$  is the mean Euclidian distance between all contour points and the center point  $\mu$ .

$$cdev = \frac{1}{N\mu_r^2} \sum_i (\|P_i - \mu\| - \mu_r)^2 \quad (7)$$

Eq.(8) defines the compactness of the contour  $cmp$  as the ratio between the circumference of a circle  $U_{circle}$  and the circumference of the contour  $U_{contour}$ .

$$cmp = \frac{U_{circle}}{U_{contour}} \quad (8)$$

Eq.(9) gives a measure for the convexity  $cnv$  of the contour. It is defined as the ratio between the circumference of the convex hull  $U_{hull}$  and that of the contour  $U_{contour}$ .

$$cnv = \frac{U_{hull}}{U_{contour}} \quad (9)$$

The convex hull is calculated using Sklansky's 3-coin algorithm [12].

#### 5. Results

Table 1 shows contour features for some typical cell contours. The hexagonal structure of an uncapped cell is reflected in high values for the convexity  $cnv$  and the compactness  $cmp$ , as well as a small deviation from a circle  $cdev$ .

|                | cnv   | cmp   | ra    | cdev  | $\mu$ |  |
|----------------|-------|-------|-------|-------|-------|--|
| open cells     | 0.994 | 0.920 | 0.623 | 0.023 | 7.065 |  |
|                | 0.993 | 0.974 | 0.840 | 0.007 | 8.531 |  |
|                | 0.992 | 0.940 | 0.686 | 0.018 | 7.477 |  |
| other contours | 0.648 | 0.213 | 0.176 | 0.271 | 5.839 |  |
|                | 0.587 | 0.146 | 0.453 | 0.054 | 8.376 |  |
|                | 0.981 | 0.771 | 0.266 | 0.159 | 7.660 |  |

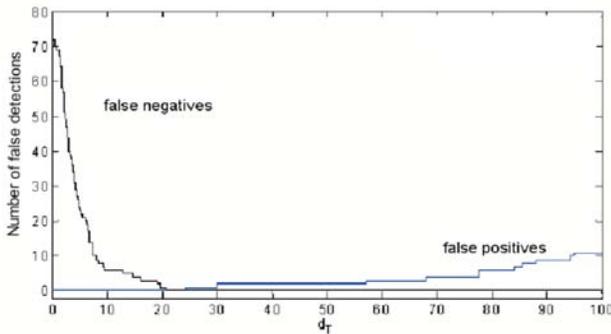
**Table 1. Properties of some typical contours**

To investigate the performance of the contour features, a training set of 66 uncapped cells was used. The mean values of all features were arranged in a column vector  $m_{cell}$ . Then the covariance matrix COV between all features was calculated. To decide whether a contour is an uncapped cell or not, the Mahalanobis distance was applied. If  $c_u$  is a vector containing the contour features of

an unknown contour, then the Mahalanobis distance  $d_{cell}(c_u)$  between the unknown contour and a hexagonal one is defined as

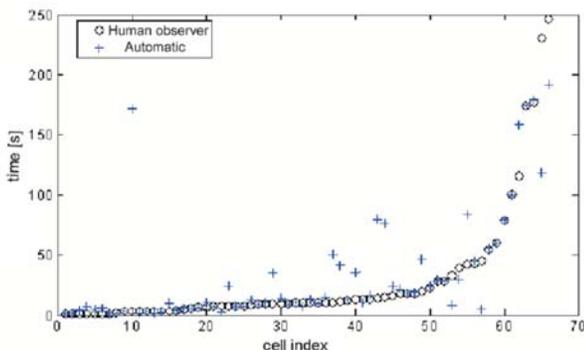
$$d_{cell}(c_u) = \sqrt{(c_u - m_{cell})^T \cdot COV^{-1} \cdot (c_u - m_{cell})} \quad (10)$$

A threshold  $d_T$  for  $d_{cell}$  is defined empirically. Other contour classes are defined using the same approach. Fig. 5 shows the detection results on a test set (71 contours of uncapped cells, ~200 contours of other objects). For a threshold distance  $d_T > 20.75$ , all open cells from the training set are classified correctly. Choosing  $d_T = 21$ , all negative examples are rejected.



**Fig. 5. Number of false positive and false negative detections as a function of the threshold distance  $d_T$  given the Mahalanobis distances  $d_{cell}$  for a training set of 71 uncapped cells and ~200 other contours**

In a second experiment a four minute movie was investigated. For comparison the uncapped cells are labeled by a human observer as soon as they become visible. The detection rate in this experiment was 90.4% with only 2 false positives. 24.3% of the automatically detected cells were found in the background image only. The remaining cells were detected from the current image. In most cases, the human observer was faster than the software detector. The mean delay between human and automatic detection was 4.9s. Fig. 6 shows the detection times for all cells. The circles mark the detection time for a human observer and the crosses the corresponding values for the detector.



**Fig. 6. Comparison between manual detection and automatic detection times since start of the movie**

## 6. Conclusions

An adaptive background model has been described which can be used in the detection of uncapped cells on a comb crowded with bees. Background modeling is typically done in applications where most of the background is visible, whereas in this problem, the foreground objects dominate the image most of the time. The proposed background model can recover faster to previously seen states of the background, compared to conventional approaches as e.g. averaging.

The typical hexagonal structure of complete open cells can be easily identified. While compactness, convexity, and ratio of the principal axes are invariant for scale, the parameters ‘deviation from circle’ and ‘mean distance from center’ are not. For a different experimental setup, these two parameters would have to be set according to the characteristics of the camera and the distance to the object.

Detection of uncapped cells in a comb is a first step towards identifying hygienic bees. Once the position and the contour of an open cell are known, the movie can be searched backwards to find the initial opening and the bees involved. Tracking those bees might then be another application.

The results of an experiment (illustrated in Fig.6) has shown that automatic detection could replace the human observer. Although the observer can detect open cells more accurately even under partial occlusion, this capability is connected with high attention which will be lost with time. The need for high attention favors automation.

## References:

- [1] D. De Jong, Mites : Varroa and other parasites of brood, In: *Honey Bee Pests, Predators and Diseases* (R.A. Morse and R. Nowogrodski eds., Cornell University Press, Ithaca), 1990, 200-218.
- [2] K. Bienefeld, G. Arnold, Studies on the genetic determination of Uncapping of Varroa-infested brood cells, *Proc. First European Conference of Apidology, EurBee*, Udine, Italy, 2004, 103-104.
- [3] G. R. Bradski, *Open Source Computer Vision Library*, Intel Corporation, 2001.
- [4] F. Gassmann, S. Mueller, Hintergrundschätzung für Aufgaben der Verkehrsflussanalyse mittels digitaler Bildverarbeitung, 48. *Internationales Wissenschaftliches Kolloquium*, Ilmenau, Germany, 2003.

- [5] I. Haritaoglu, D. Harwood, L. S. Davis, Multiple People Detection and Tracking Using Silhouettes, *Proc. 10<sup>th</sup> International Conference on Image Analysis and Processing*, Venice, Italy, 1999, 280-285.
- [6] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, Tracking Groups of People. *Computer Vision and Image Understanding*, 80(1), 2000, 42-56.
- [7] C. Stauffer, W. E. L. Grimson, Adaptive background mixture models for real-time tracking, *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Vol 2*, Fort Collins, CO, USA, 1999, 200.
- [8] A. Elgammal, D. Harwood, L. Davis, Non-parametric Model for Background Subtraction, *Proc. 6<sup>th</sup> European Conference on Computer Vision*, Dublin, Ireland, 2000, 751-767.
- [9] D. Butler, S. Sridharan, V. M. Bove Jr., Real-time adaptive background segmentation, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, China, 2003, 349-352.
- [10] J. Canny, A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Vision*, 8(6), 1986, 679-698.
- [11] C. H. Teh, R. T. Chin, On the Detection of Dominant Points on Digital Curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8), 1989, 859-872.
- [12] J. Sklansky, Measuring concavity on a rectangular mosaic. *IEEE Transactions on Computing*, 21(12), 1972, 1355-1364.