Background subtraction for honey bee detection in hive entrance video

Ratko Pilipović, Vladimir Risojević, Zdenka Babić
Faculty of Electrical Engineering
University of Banja Luka
Republic of Srpska, Bosnia and Herzegovina
Email:{ratko.pilipovic, vlado, zdenka}@etfbl.net

Goran Mirjanić
Faculty of Agriculture
University of Banja Luka
Republic of Srpska, Bosnia and Herzegovina
Email: goran.mirjanic@agrofabl.org

Abstract—**Honey bees play crucial role in pollination across the world, therefore it's of great significance to observe status of honey bee colony honey inside bee hive. Use of automated honey bee analysis will minimize human impact on bee colony and protect her from outer influences. In hive entrance video honey bees represent fast moving objects, so the task of identifying honey bees represent widely known moving object detection problem. Paper gives researchers guidelines for finding suitable background subtraction technique that would be sufficiently accurate and computationally efficient to execute on embedded systems. This would enable researchers to develop more complex algorithms for honey bee analysis behavior. To find the most suitable technique we underwent trough comparative analysis of techniques that are often use for detecting moving objects. In the end we selected Mixture of Gaussians for honey bee detection.**

Keywords: Background subtraction, foreground detection, frame differencing, median, Kalman filter, MOG, honey bee

I. INTRODUCTION

Nowadays, much attention is dedicated towards preservation of environment. Most crucial process for environment preservation is pollination, which is necessary for plant reproduction. Honey bees are probably the most important pollinators across the world and their crucial role ensures stability of ecosystem [1]. Beside the part in pollinating process, honey bees produce honey, which is irreplaceable part of healthy human nutrition. For beekeepers, the most important information are strength and health of beehive colony. Main indicator of bee colony strength is number of bees that are entering and leaving a bee hive. Therefore finding this number would enable beekeepers to act on time and prevent diseases and parasites that would harm bee colony health and reproduction.

Identifying honey bees from a hive entrance video sequence is a very complex task that leads to building algorithms for honey bee behavior analysis. In video sequences honey bees represent fast moving objects, so the task of identifying honey bees represents widely known moving object detection problem. The most common approach for solving this problem is background subtraction. Nowadays background subtraction techniques are widely used in video surveillance, traffic monitoring, human detection and tracking applications. In background subtraction techniques, each video frame is compared to a background model. Pixels that significantly deviate from background model represent moving objects or foreground, while pixels which belong to background, are very similar with background model. Foreground pixels are later used for further analysis and processing, so it is very important that they accurately correspond to moving object.

The main goal of this paper is finding suitable background subtraction technique that would be incorporated into a method for honey bee behavior analysis. At the same time our desired background subtraction technique would be sufficiently accurate and computationally light to execute on embedded systems.

Background subtraction techniques differ from each other in a way that they compute background model. They can be divided into two groups: non-recursive and recursive techniques [2]. Non-recursive techniques are based on use of buffer to store frames on which background model is build. Background model is based on temporal variation of each pixel within the buffer. The most simple non-recursive technique is frame differencing, which is based on comparing two consecutive frames. More advanced techniques observe more than two consecutive frames, e.g. temporal median filter [3] that calculates background model as median value of frames stored in buffer. Instead of median value, linear predictive filter [4] calculates a background estimate as a prediction based on frames in buffer. Non-recursive techniques must use long buffer in order to cope with slow moving objects and long term changes in background. While increasing buffer length, complexity of calculating background model is increased. For these reasons recursive techniques are popular in motion detection challenges.

Recursive techniques do not require buffer because they recursively update background model on every frame. Recursive version of temporal median filter is called approximate median filter [5]. In this recursive technique, background estimate of each pixel is incremented by one if current value of pixel is larger than estimate or decreased by one if current value of pixel is smaller than estimate. More complex technique is based on applying Kalman filter on each pixel index frame to track pixel intensity. Unlike Kalman filter that tracks pixels intensity, Mixture of Gaussians models value of single pixel using a mixture of Gaussian distributions which differ in mean values and standard deviations.

To the best of our knowledge, background subtraction techniques are primarily used in traffic analysis application e.g. detecting cars and pedestrians, but it has never been applied them in honey bee detection. In this paper we discuss

This research was supported by the Ministry of Science and Technology of the Republic of Srpska under contract 19/6-020/961-187/14.

about the use of background subtraction in detecting honey bees in hive entrance video. We concentrate our research on examination of two non-recursive techniques: Frame differencing and Temporal median filter and two recursive techniques: Kalman filter and Mixture of Gaussians. These techniques are chosen, because they are appropriate for use in embedded system. To find the most suitable technique for bee detection, we compared background subtraction techniques using F-measure.

Rest of paper is organized as follows: the survey of background subtraction techniques and are presented in Section 2. In Section 3 we present experimental results. Finally, we conclude our paper and discuss future work in Section 4.

II. Background subtraction methods

Background subtraction is a widely used approach for detecting moving objects in videos obtained from a static camera. The essence of this approach is comparing current frame with reference frame called "Background model" or just "Background". Background model represents observed scene without moving objects. Although different, most background subtraction methods share a common framework: they make the hypothesis that observed video sequence I is made of background model B on which moving objects are placed. In this section the chosen techniques will be explained in detail.

A. Frame differencing:

This technique is the simplest background subtraction technique. Frame differencing uses the frame in time instant *t-1* and frame in time instant *t*. The foreground mask FM is calculated as follows:

$$FM_{m,n} = \begin{cases} 1, & \left| I_{m,n}(t) - I_{m,n}(t) \right| \geq T_s \\ 0, & otherwise \end{cases} \quad (1)$$

where T_s represent threshold value, I represents frame and *m,n* represents spatial coordinates of appropriate frame and mask.

The main advantage of this technique lies in its simplicity since it is based on differencing two neighboring frames but main disadvantage is inability to detect large uniform objects because, frame differencing technique detects only edges of moving object[6].

B. Temporal median filter:

In this approach background model is computed as a median value of previous frames inside a buffer [7]. In the basis of this technique lies assumption that background pixels stay in background more than half of frames in buffer. The foreground mask is calculated as follows:

$$FM_{m,n} = \begin{cases} 1, & \left| B_{m,n} - I_{m,n}(t) \right| \geq T_s \\ 0, & otherwise \end{cases} \quad (2)$$

where T_s represent threshold value, B represents background model, I represents current frame and *m,n* represents spatial coordinates of appropriate frame and mask

The main advantage of this technique lies in its simplicity since background estimation is based on calculating median value of pixels trough frames, but main disadvantage is use of a buffer. The length of buffer increases complexity of calculating background model.

C. Kalman filter

Kalman filter represents recursive system state estimator whose estimation is based on prior system state and available measurements. Kalman filter is widely used in automatic control, tracking and detecting moving objects.

When moving object detection is concerned, Kalman filter is used for background estimation. Procedure of using Kalman filter for background estimation is described in [8]. The background estimate is obtained by applying a matrix of one dimensional estimation filters. This method is based on assumption that pixels in background have intensities that don't evolve quickly in time, while foreground pixels do.

The algorithm is a two-step process of mean intensity update and standard deviation update. The standard deviation and prior intensity values are updated using Kalman filter prediction/correction equation manipulated to mean and standard deviation update:

$$\mu_{m,n}(t) = \frac{(g_{m,n}(t) \cdot \mu_{m,n}(t-1)) + (\dfrac{z_{m,n}(t)}{d_{m,n}(t)} \cdot \sigma_{m,n}(t-1))}{g_{m,n}(t) + \sigma_{m,n}(t-1)} \quad (3)$$

$$\sigma_{m,n}(t-1) = \frac{g_{m,n}(t) \cdot \sigma_{m,n}(t-1)}{g_{m,n}(t) + \sigma_{m,n}(t-1)} \quad (4)$$

where $\mu_{m,n}$, $\sigma_{m,n}$ represent values of mean value and standard deviation of pixel on position *m,n* and time *t*. The d variable represents process noise gain, while s variable represents measurement noise gain, g represents ratio between these two variables and $g_{m,n}(t)$ represents current value of pixel on position *m,n*. The background estimate and foreground mask is calculated FM is calculated as follows:

$$B_{m,n}(t) = \mu_{m,n}(t) \quad (5)$$

$$FM_{m,n} = \begin{cases} 1, & \left| I_{m,n}(t) - I_{m,n}(t) \right| \geq T_s \\ 0, & otherwise \end{cases} \quad (6)$$

Main benefits of this approach are independence of scene content, long and short term history behavior and deterministic execution.

D. Mixture of Gaussians:

Mixture of Gaussians (MOG) models the value of a particular pixel using a mixture of Gaussian distributions which differ in mean values and standard deviations [9]. The probability of a pixel value X_t at time instance t is:

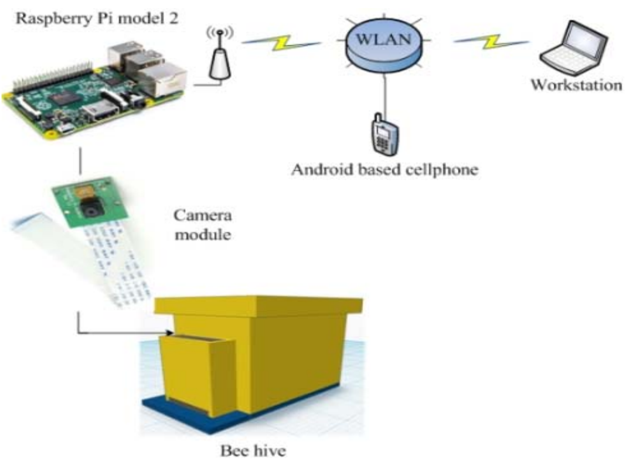

Figure 1. Block scheme of remote sensing platform

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (7)$$

here η represents the *i*-th Gaussian distribution with mean value $\mu_{i,t}$ and covariance matrix $\Sigma_{i,t}$:

$$\eta(X_t, \mu_{i,t}, \sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\Sigma_{i,t}\right|^{\frac{1}{2}}} e^{(X_t - \mu_{i,t})\left|\Sigma_{i,t}\right|^{-1}(X_t - \mu_{i,t})^T} \quad (8)$$

Distribution weight $\omega_{i,t}$ represents portion of the data corresponding to the *i*-th Gaussian distribution. For computational reasons, the covariance matrix is assumed to be a form of the form:

$$\Sigma_{i,t} = \sigma_{i,t} I \quad (9)$$

represents standard deviation of the *i*-th Gaussian distribution. Number of Gaussian distribution K depends on available memory and computational power, but in practical application e.g. traffic analysis vary from 3 to 5.

The algorithm proceeds as follows. Every new value X_t is checked against all Gaussian distributions. If there is a Gaussian such that pixel value is within 2.5 standard deviations of distribution it is considered a match. If none of the K distributions match the pixel value, then distribution with smallest weight is replaced with a new Gaussian distribution that has low prior value, high variance and mean value that equals the current pixel value.

On the other hand, if a match is found, then prior weights of Gaussian distributions are adjusted in accordance to:

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} + \alpha M_{i,t} \quad (10)$$

where $M_{i,t}$ is equal to 1 if *i*-th distribution matches the pixel value, and 0 otherwise. The parameter α represents the learning rate. After the parameters update, the weights are renormalized.

Mean values and variances of unmatched Gaussian distributions remain the same. For the matched Gaussian distribution they are updated as follows:

$$\mu_{i,t} = (1-\rho_{i,t})\mu_{i,t-1} + \rho_{i,t} X_t \quad (11)$$

$$\sigma_{i,t}^2 = (1-\rho_{i,t})\sigma_{i,t-1}^2 + \rho_{i,t}(X_t - \mu_{i,t})^T (X_t - \mu_{i,t}) \quad (12)$$

where $\rho_{i,t} = \alpha / \omega_{i,t}$

After the parameters update, the Gaussian distributions are sorted in ascending order according to the value of. The most probable background distributions remain on top, while foreground distributions are at the bottom of list. The first B Gaussian distributions model background, where:

$$B = \arg\min\left(\sum_{i=1}^{B} \omega_{i,t} > T\right) \quad (13)$$

where T represents the minimum portion of the data that should be regarded as background. The higher the value of this parameter, the more pixels are incorporated in background. The pixel will be part of background, if its value is within 2.5 standard deviations of distribution that models the background. Otherwise, pixel will be part of foreground.

MOG successfully copes with long term luminance changes, repetitive changes and sleeping foreground objects. Sleeping foreground objects represent foreground objects that stopped moving and became part of background, e.g. cars on parking ground. Unlike non-recursive techniques MOG algorithm has slower adaption rate that depends on learning constant.

III. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

In this section we compare the performances of the methods discussed above. Tested background subtraction techniques are given in Table 1. We vary the foreground threshold T_s or in the case of MOG weight threshold T to show relation between applied threshold and F-measure. The different set of threshold values is applied on each technique. This is done in order to find optimal value of threshold for each technique

TABLE I. CHOSEN BACKGROUND SUBTRACTION TEHNIQUES

	Background subtraction techniques		
	Technique	***Fixed Parameters***	***Test Parameters***
1.	Frame differencing	None	Foreground threshold T_s
2.	Temporal median filter	Buffer size L = 50	Foreground threshold T_s
3.	Mixture of Gaussians	Number of Gaussians K = 3, Learning rate α = 0.005	Weight threshold T
4.	Kalman filter	Process gain d = 1 Measurement noise gain s = 0.6	Foreground threshold T_s

For temporal median filter we have limited the length of frame window to 50 frames. This is done, because further increasing the buffer length would not significantly improve,

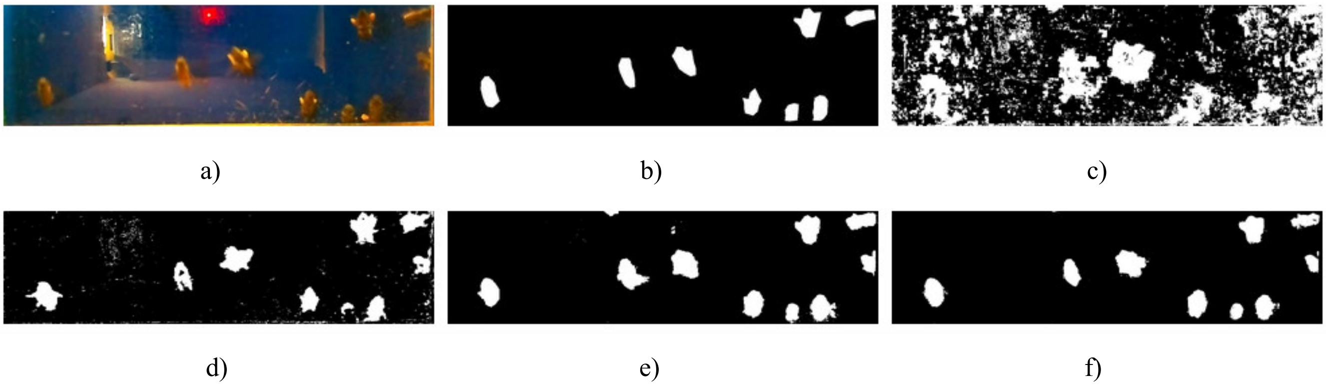

Figure 2. Resulting forerground masks for different background subtraction tehniques: a) Original frame b) Ground thruth c) Frame differencing d) Temporal median filter e) Kalman filter f) Mixture of Gaussians

while memory requirements would increase. Furthermore, we converted frames into grayscale images before calculating median value of frames inside window. This is done in order to lower complexity which increases with the increase of a length of buffer.

In the case of Kalman filter we have fixed process gain and measurement noise gain. Values for process gain and measurement noise gain presented in Table 1 are heuristically

A. *Test sequences*

For testing we use frames from a surveillance video of a bee hive entrance. The monitoring hardware consists of a sensing platform, computational hardware and a communication module, as it is shown in Fig. 1. The sensing platform consists of a specially designed wooden box (sensing box) with a Raspberry Pi camera module inside, mounted on the front side of a standard hive, above the hive entrance. Camera is placed behind protective glass, so in the frame appear reflections of camera. As computational hardware, we used a Raspberry Pi model 2 board, which is also mounted on the sensing box. The task of the Android based cellphone is to establish a WLAN connection between the Raspberry Pi board and a remote workstation.

Tested video sequence consist of 562 frames with resolution of 1280x720 pixels and at 30 frames per second. A sample frame is shown in Fig. 1. This sequence is showing movement of honey bees on bee hive entrance. Some honeybees walk on bee hive entrance, while others fly into beehive without touching flight board and there are also bees that walk on protective glass. In order to avoid non-uniform illumination in the camera field of view under the sensing box, we analyze only the lower part of each video frame (1280x360 pixels) close to the hive entrance

B. *Quantive measure for perfomance analysis*

For comparative performance analysis of selected background subtraction techniques, we have chosen 10 non-consecutive frames, and manually labeled the moving objects or in our case honey bees. In this way we obtained ground

obtained. To implement Kalman filter we have used MATLAB code given in [8].

For testing purposes, we used the implementation of MOG algorithm from Computer Vision toolbox. In order to lower memory demands and computational power we use 3 Gaussian distribution per pixel. The value of learning rate is heuristically obtained. All experiments are performed on a computer with Intel i5 processor and 8 GB of RA

truth frames, which will be later used to calculate precision and recall. An example frame and corresponding ground truth frame are shown on Fig. 2. (a) and (b), respectively.

To quantitatively evaluate chosen techniques for each technique we will evaluate F-score or F-measure. F measure is defined as harmonic mean of precision and recall:

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{14}$$

Recall is defined as ratio of number of foreground pixels correctly identified by the algorithm and number of foreground pixels in ground truth. Precision is defined as ratio of number of foreground pixels correctly identified by the algorithm and number of foreground pixels in detected by algorithm

C. *Results*

In Fig. 2. (c) – (f) are shown the results of each evaluated background subtraction technique applied to frame in Fig. 2. (a). Resulting frames are compared to ground truth, which is shown in Fig. 2 (b) in order to measure precision and recall of each algorithm. Comparing ground truth and results of each algorithm we can conclude that temporal median filter and frame differencing detect large amount of background pixels as foreground while Kalman filter and MOG algorithm will tend to detect foreground pixel as background. This behavior is result of lighting changes in the observed scene (Frame differencing and temporal median filter) and finite adaption time on changes in observed scene (MOG and Kalman filter).

As we stated in Section III.B for each of chosen techniques we will calculate F-measure, in order to

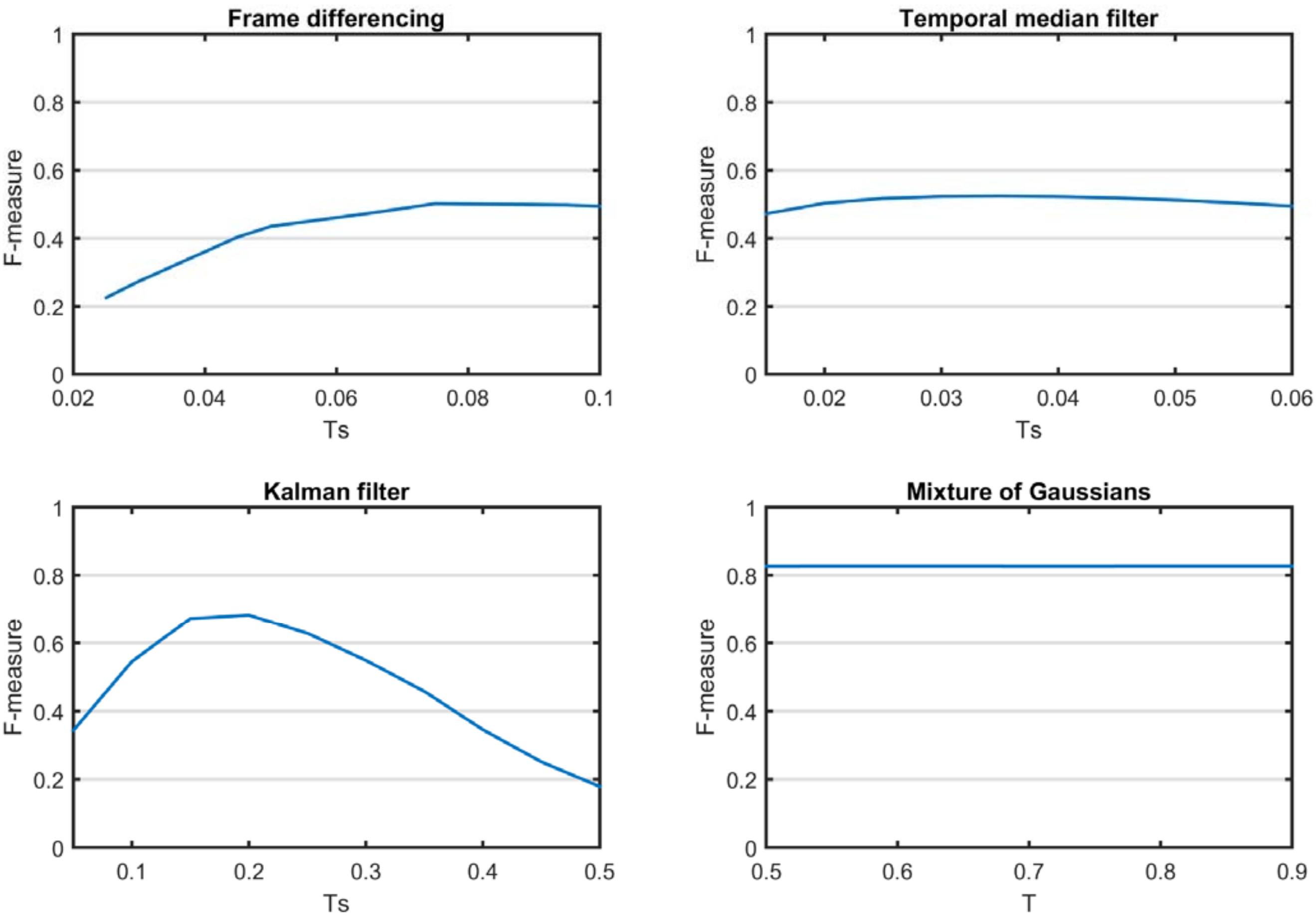

Figure 3. Relation beetwen F-measure and applied treshold for chosen techniques

quantitatively evaluate them. Relation between F-measure and applied foreground threshold for chosen background subtraction technique is depicted in Fig. 3. The first two plots correspond to non-recursive techniques: Frame differencing and Temporal median filter. The curves are generated varying foreground threshold . The remaining plots correspond to recursive techniques: Kalman filter and Mixture of Gaussians. Plot for Mixture of Gaussians is generated varying background ratio T while plot for Kalman filter is generated varying foreground threshold T_s

Based on Fig. 2 and Fig. 3 we came with these conclusions:

- The Mixture of Gaussias is best suited for detecting bees as foreground objects based on achieved F-measure. The second comes Kalman filter, while Temporal median filter and Frame differencing have very close F-measure value.
- Non-recursive techniques perform very poorly in bee detection while recursive techniques have satisfying results. Poor performances of non-recursive techniques are consequence of inability to cope with lighting changes while recursive techniques are adaptive in term of lighting changes. While non-recursive techniques have problem with lightning changes that is result of recording condition (i.e. presence of sensing box) recursive technique have problem with slow moving objects. These objects will be incorporated in background, so slow moving bees will be partially detected or not detected at all.
- Frame differencing produces the worst results, because it is very sensitive to lightning changes. On the other hand its main advantage is simplicity, therefore targeted embedded system doesn't need to be computationally strong

- Temporal median filter produces satisfying results. This results comes with price of increased memory demands for embedded system. Also increased window length increases complexity of computation, so we must carefully choose length of the window in accordance to the target systemKalman filter, which is primarily used to track moving objects, performs very well in detecting honey bees. Without need for frame buffer, Kalman filter can be used in embedded system with limited memory resources. But on the other hand, Kalman filter, as a recursive technique, have finite adaption time, that would sometime cause false background detections.

- MOG algorithm has best results in honey bee detection. Problems with slow adaptation rate can be corrected with more Gaussian distribution and larger learning rate. If we add more Gaussians, the complexity of algorithm will increase. If we increase learning rate algorithm, adaptation rate will increase, but algorithm will be less immune to lightning changes.

Beside F-measure we have measured the time needed to process frame on desktop computer described in Section III. For Frame differencing it takes 86 ms to process one frame. Temporal median needs 1.9 seconds to complete the task of processing one frame. The reason of this long processing time is the use of relatively long buffer. For the same job Kalman filter needs 173 ms, while MOG needs 80 ms.

IV. Conclusion

This paper discuss discusses about possibilities of using background subtraction for honey bee detection in hive entrance surveillance videos. Because honey bees are fast moving objects, we treat problem of detection honey bees as problem of moving object detection, and test different BS techniques on this task. The four specific background subtraction techniques are used: Frame differencing, Temporal median filter, Kalman filter and Mixture of Gaussians. To find the most suitable technique, we compared them using F-measure. Testing were made on video which is result of remote video recording of bee hive entrance. After detailed comparative analysis we concluded that Mixture of Gaussians is best suited for detection honey bees in hive entrance video. With this conclusion we have made the first step in creating method for honey bee behavior analysis.

In the future work we plan to implement the bee detection algorithms in an embedded system mounted on a bee hive. It will be used for assessment of real-time detection algorithms.

Acknowledgment

We are grateful to J. Scott author of [8], for making the code for Kalman filter publicly available.

References

[1] G Potts, S. G., Biesmeijer, J. C., Kremen, C., Neumann, P.,Schweiger, O., & Kunin, W. E., “Global pollinator declines: trends, impacts and drivers.” Trends in ecology & evolution, 25(6), pp. 345-353, 2010.

[2] S. Cheung, C. Kamath, “Robust techniques for background subtraction in urban traffic video,” Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 5308, pp. 881-892, Jan., 2004

[3] Piccardi, M. "Background subtraction techniques: a review." In proc. IEEE international conference on Systems, man and cybernetics, vol. 4, 2004, pp. 3099-3104.

[4] Cucchiara, R., Grana, C., Piccardi, M. and Prati, A., Detecting moving objects, ghosts, and shadows in video streams. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(10), pp.1337-1342, Oct 2003.

[5] McFarlane, N.J. and Schofield, C.P.,. “Segmentation and tracking of piglets in images” Machine vision and applications, 8(3), pp.187-193, 1995.

[6] Cheung, S.C.S. and Kamath, C., “Robust background subtraction with foreground validation for urban traffic video” EURASIP Journal on Advances in Signal Processing, pp.1-11, 2005.

[7] R. Cutler and L. Davis. View-based detection and analysis of periodic motion. In International Conference on Pattern Recognition, page SA14, Brisbane, Australia, August 1998.

[8] Scott J, Pusateri MA, Cornish D. “Kalman filter based video background estimation” In proc. Applied Imagery Pattern Recognition Workshop (AIPRW), Oct 2009, pp. 1-7.

[9] Stauffer, C., & Grimson, W. E. L. “Adaptive background mixture models for real-time tracking.” In proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999. (Vol. 2).