2012 International Conference on System Science and Engineering
June 30-July 2, 2012, Dalian, China

Image-based Intelligent Attendance Logging System

Hary Oktavianto
Electrical Engineering Department
Electronic Engineering Polytechnic Institute of Surabaya
Surabaya, Indonesia
hary@eepis-its.edu

Hsu Gee-Sern[1], Chung Sheng-Luen[2]
[1)] Mechanical Engineering Department
[2)] Electrical Engineering Department
National Taiwan University of Science and Technology
Taipei, Taiwan
slchung@mail.ntust.edu.tw

***Abstract*—This paper proposes an extension of the surveillance camera's function as an intelligent attendance logging system. The system works as a time recorder within two phases; learning phase and monitoring phase. By placing a camera inside a working room, the sytem enters the learning phase by locating the sitting area automatically based on the camera's images. The learning phase has a defined time duration to complete a working map. Next, the system switches to the monitoring phase to report each occupant's working hour corresponding to their sitting area in the working map. The system detects the entering occupant, the leaving occupant, the sitting occupant, and the standing occupant from their seat. The system also tracks the occupants. The working hours of the occupant will be counted as how long they sit in their desk. When the occupant sits at his/her seat, a start time is given. Later, after he/she leaves from their seat, a stop time is generated. The experimental setup has been done in our laboratory. The result shows that the system can achieve a good result.**

Keywords-Activity map, Attendance, Logging System, Learning pahse, Monitoring phase, Surveillance camera

I. INTRODUCTION

Many working environments need supervision. For instance, in the line assembly of the factory or in the office. One kind of supervision is the action to keep monitoring the worker for being present or not. Usually, the company provides a time clock or time recorder. A time recorder is a mechanical or electronics timepiece that is used to assist in tracking the hours an employee of a company worked [1]. The time recorder could be mechanic (such as punched-card) or electronic (such as magnetic stripe card, RFID tag, hand-punch, or fingerprint). This paper wants to seek a new way to design a time recorder. Considering that intelligent buildings have increased as a research topic recently [2], [3], [4] and many buildings are installed with surveillance cameras for security reasons or as the infrastructure for context-aware applications, this paper proposes a technique to implement a system that works like a time recorder. By extending the function of the existing surveillance cameras as an intelligent attendance logging system, the proposed system has purposes to monitor and to report the occupant's attendance. This system is suitable to be implemented in the sedentary working environment. For the occupants, they do not need to bring any tag or badge. The system will monitor the occupants' attendance by using a camera that is placed inside the working room and sends the events/messages in real-time. The system is also designed to support the context-aware applications such as smart home or intelligent buildings.

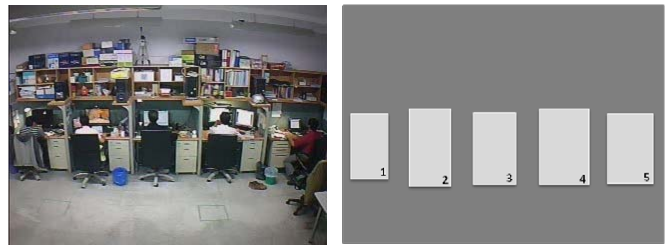

Figure 1. Occupant's working room (left) and a map consists of occupants' sitting areas (right).

To monitor the occupants' attendance, the locations of the occupant's working area should be defined previously. The occupant's working area is also called as the sitting area. There are two options to define the sitting area. The first option is the user specifies the locations manually and informs to the system. The second option is the system learns to find the locations automatically. After the locations are defined clearly, the system starts to monitor the occupants' attendance. The second option is chosen in this paper. Hence, the system is so called intelligent since it has ability to learn from a given environment to locate the sitting areas. The collection of the sitting area is called as a map.

Fig. 1 shows an example scenario. Ideally, the occupant enters into the room and sits to start working. After that, the occupant stands up from his/her seat and leaves the room. The sitting and the standing up events will be used to decide where the sitting area of the occupant is and when the occupant works. In the system implementation, this paper also intends to exploit the advantages of the existing open source library for the computer vision, OpenCV [5] and cvBlob [6].

The goal of this paper is to design an image-based intelligent attendance logging system with at least has the following functions:

1) Learning phase: detection and identification of seat locations in an unknown environment.
2) Monitoring phase: detection of entering and leaving events for each occupant into and from respective seat.
3) Real-time system: implementation of real-time intelligent attendance logging system.

978-1-4673-0945-5/12/$31.00 ©2012 IEEE

During the past decades, intelligent building has been developed. By using the cameras, the researchers extracted the information from the environment into a map. In [4], the map is labeled already. In [7], [8], [9], [10], and [11], the map is learned automatically to indicate interesting image regions and the way objects move between these places. In the case of the camera that was used, a static camera is used by the system as in [3], [4], [9], [10]. In [11], they applied omnidirectional camera to their system. The others, [2], [7], [8], used stereo camera to reduce the effect of lighting intensity and occlusion.

By integrating the existing open source library with additional algorithms, we have designed an intelligent attendance logging system which works in two phases. Corresponding to the objectives, there are three contributions in this paper:

1) Learning mechanism that locates seats in an unknown environment.
2) Monitoring mechanism that detects entering and leaving events of occupants.
3) Integrating system with real-time performance up to 16 fps, ready for context-aware applications.

This paper is organized with the following sections. Section II describes the problem definition and preliminary. Section III explains about the proposed approach. Section IV discusses about the experiment results. Finally, the concluding remark and future work are given in Section V.

II. Problem Definition and Preliminary

We made an observation on the sample image taken from the camera in the working environment. The working environment that we used in this paper is our laboratory. There are some desks in a line. To have a better view and considering the limitation of our camera's view, we set the camera perpendicular to the desks. We recorded some image sequences as a video clips. We would like to know what we can learn from the one sample image taken by a camera in the real condition. Fig. 2 shows the snapshot of the image samples. The observations on these samples are listed as the following:

1) The path of the walking occupants in the scene is horizontal.
2) A person may walk behind the sitting person.
3) When a person sits, he/she is occluded by the chair.
4) The part of the person has similar color with the surrounding furniture.
5) An occlusion of two people happened sometimes.
6) There are seats in the both edge. This could be a problem if the occupant suddenly moves outside of the scene.

The system will be designed by considering the result from the observation. For instance, the horizontal path of the walking occupants can make the grouping algorithm of broken blobs easier. Also, we used four attribute features for tracking which are centroid, size, ground position, and color. We defined centroid as a center of the occupant's bounding box. Size feature is pixel density of one detected occupant. Ground position is the position of the occupant's foot which is calculated at the bottom-centered of the occupant's bounding box. Color is the RGB histogram from the occupant.

Figure 2. Snapshot of some image samples.

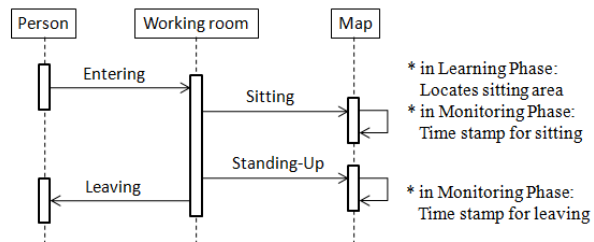

Figure 3.The diagram of the system.

A. Tracking states

During the appearance of the object in the scene, we would like to keep tracking the object which is an occupant. Since the system can track a person, we just follow the person using the four attribute features. The system has four tracking states that have responsibility to track each person. The tracking states are entering state (ES), sitting state (SS), standing-up state (US), and leaving state (LS). The tracking states will be explained in section 3. Fig. 3 shows the diagram of the system. A person enters to the working room. The person sits into his/her seat. He/her stands up from the seat. Finally, the person leaves the working room. There will be two phases in this system, learning phase and monitoring phase. Both phases are triggered by sitting event and standing-up event. During the learning phase, sitting event is used to locate the seating area. In the monitoring phase, sitting event is used to give a time stamp of the sitting time and standing-up event is used to give time stamp of the leaving time.

B. Approaches and Performance Criteria

This intelligent attendance logging system is designed based on two assumptions. The first assumption is the environment is unknown, in that, the number of seats and the locations of these seats are not known before the system monitors. The second assumption is each occupant has his/her own seat, as such, detecting the presence/absence of a particular seat amounts to answering the presence/absence of that corresponding occupant. In order to achieve the research

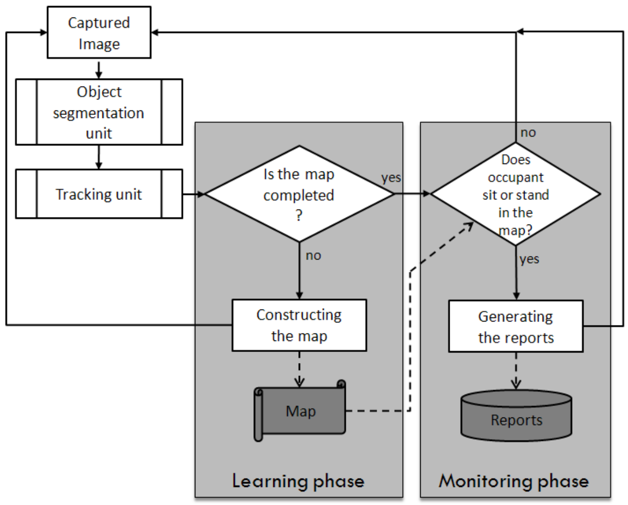

Figure 4. The algorithm of the system.

objectives, the algorithm of the proposed system is shown in Fig. 4. The system consists of an object segmentation unit, a tracking unit, learning phase, and monitoring phase. The report on presence or absence of the occupants is the final output of the system for further analysis.

There are two performance criteria to evaluate the system regarding to the main functions of the system. The main functions of the system are to detect the occupants' sitting area and to report the monitoring result. The first criterion is the system should detect the sitting areas given by the ground truth. The second criterion is the system should be able to monitor the occupants during their appearance in the scene to generate the accurate report. The system will be evaluated by testing it with some video clips using different scenario. The occupants enter into the scene. They will sit, stand up, leave the scene and sometimes make a group then separate. The first scenario is the occupants enter to the scene and leave one by one without making any occlusion. The objective is to see how good the system can detect the location of the occupants' sitting area. The second scenario is the same as the first scenario but the occupants will make a group to see how good the system can handle the occlusion problem. It is assumed that the scene in the video clips has horizontal path only, meaning that the occupants walk in the horizontal direction.

C. Broken blobs

We would like to have a person corresponding to a single blob. But sometimes, the part of the person has the similar color with the background or the person sits at his/her chair. Then, the whole body is segmented into different parts. After we got the blobs, we need to group them if the unconnected blob belongs to a person. There are three conditions to examine the broken blobs. The first is the intersection distance of blobs (B_I). The second is the nearest vertical distance of blobs (B_{dy}). The third is the angle of blobs (B_A) from their centroids. The unit of B_I and B_{dy} are in pixels and the unit of B_A is in degree. If those three conditions are satisfied (1) then the broken blobs are grouped into one blob. T_I, T_{dy}, and T_A are the threshold values for the intersection distance, the nearest vertical distance of blobs, and the angle of blobs, respectively. Each broken blob

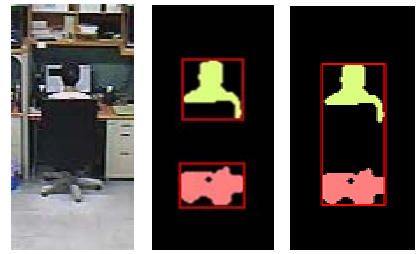
Figure 5. Blob grouping result of the occupant which is occluded by chair.

will be marked by a rectangular box. The B_I may have zero value or negative value, meaning that the rectangular box of two broken blobs is overlaping each other. The B_{dy} has absolute value (positive only). The B_A has absolute value relative to each other. In the experiments, T_I is 0 pixel, T_{dy} is 50 pixels, and T_A is 30°. Fig. 5 shows the result of blob grouping algorithm when an occupant has the same color with the background image. The left image is the current image in the scene. The middle image is shown the broken blobs of the occupant. In this case, according to (1) the logic of G will be ((false) OR (true AND true)), that is 1. Hence, the two broken blobs are grouped. The right image is the result of blob grouping (it is shown by one rectangular box). Here, the blob grouping algorithm is expected to reconstruct the occupant with broken blobs. After the broken blobs are grouped, the centroid of the grouped blobs will be located at the center of the rectangular box.

$$G = \begin{cases} 1 & (B_I \le T_I) \vee \left((B_{dy} < T_{dy}) \wedge (B_A < T_A) \right) \\ 0 & otherwise \end{cases} \quad (1)$$

III. Proposed Approaches

A. Tracking an individual

The system needs some mechanism to keep monitoring the occupants' status in the scene whether they seated or unseated. The status of the occupant is represented by the tracking states. As shown in Fig. 3, there are four states:

(1) Entering state (ES), an incoming blob that appears in the scene for the first time will be marked as entering state. This state also receives information from the motion detector to detect whether the incoming blob is an occupant or a noise. If the incoming blob is considered as noise and it remains there for more than 100 frames then the system will delete it. For instance, the size is too small because of shadows. To erase the noise from the scene, the system re-initializes the Gaussian model to the noise region so that the noise will be absorbed as a background image. An incoming blob is classified as an occupant if the incoming blob has motion at least for 20 frames continuously and the height of the blob is more than 60 pixels. Entering state will also group the broken blobs if any. A unique identification (ID) number and a bounding box are attached to the incoming occupant. The blob or the grouped blobs detected as an incoming occupant is called as a track. The system adds this track in the tracking list.

(2) Sitting state (SS), detects if the occupant is sitting. In the experiments, sitting occupant can be assumed with

three cases in which there is no movement from the occupant for a defined time. First, the x-axis displacement is zero more than 20 frames. Second, the optical flow result is zero more than 100 frames. Third, the track appears in the scene more than 100 frames. As a note, the camera's image is chosen to be 340x240 pixels. Using this image size, the experiment shows that the occupant's movements such as breathing or typing on the keyboard do not change the x-axis of the bounding box.

(3) Standing-up state (US), detects when the sitting occupant starts to move to leave his/her desk. In the experiments, a standing up occupant is detected when the sitting occupant produces movements, the height increases above 75%, and the size changes to 80%-140% compared to the size of the current bounding box.

(4) Leaving state (LS), deletes the occupant from the list. A leaving occupant is detected when occupant moves to the edge of the scene and occupant's track loses its blob for 5 frames.

Among these four states, the tracking features are used in different way, meaning that not all of the features are used for each state. There are two groups of states that used the tracking features in the same way to match the track from frame to frame. The first group is the sitting state (SS) and standing-up state (US). They use all of the tracking features. The track's position will be updated with the blob's position in the next frame when meets these requirements: the distance of their centroids is below the threshold T_I, their ground positions are lying in the range of the predefined ellipse boundary, their color matching score is above 0.8, and their size ration is in the range of $\pm 20\%$. The second group is entering state (ES) and leaving state (LS). They only use centroid feature with the same handling as the first group.

B. *Learning phase*

It is easier when we had the location of the occupants' sitting areas previously. This system intends to build the map which consists of a set of occupant's sitting areas automatically. It is called a learning phase which has responsibility to locate the seats. The sitting state (SS) is used to build the sitting area. Once the sitting event is triggered, the system starts to count. If the occupant stays more than 200 frames then the current location of the occupant's bounding box is declared as a sitting area. The system will remember that sitting area and put it into the map as shown in Fig. 1.

The learning phase is running continuously for predefined duration such that the map is expected to be finished completely. After the learning phase is terminated, the system switches to the monitoring phase.

C. *Monitoring phase*

The monitoring phase takes place after the learning phase is finished to construct the map. This phase has responsibility to monitor the occupant's presence. The monitoring phase utilizes the sitting state (SS) and the standing-up state (US). Sitting state (SS) sends an event to tell that the occupant is detected as sitting. After sitting state is triggered and if rule (2) is satisfied then the system generates a sitting time report.

Sitting area number	Event	Time stamp
1	Sitting	09:02:09 Wed 2 June 2010
2	Sitting	09:07:54 Wed 2 June 2010
3	Sitting	09:12:16 Wed 2 June 2010
2	Leaving	10:46:38 Wed 2 June 2010
2	Sitting	10:49:54 Wed 2 June 2010
3	Leaving	12:46:38 Wed 2 June 2010

Figure 6. A report example.

Figure 7. Occluded occupant problem.

$$if \begin{cases} \text{the centroids are overlaping} \\ \text{the overlaping area ratio is more than } 70\% \\ \text{ground positions are overlaping} \end{cases} \quad (2)$$

When standing up state (US) is triggered to inform that the occupant is leaving his/her seat, the system generates a leaving time report. Fig. 6 shows an example of generated report. The monitoring phase runs forever or until the user terminates the program to report the occupant's attendance. Next time when the user restarts the program, the system tries to find the map. If the current map exists then the system runs the monitoring phase directly. On the other hand, if the current map does not exist then the system runs the learning phase. It is also allowable for the user to restart the system in the learning phase, for instance, when the scene is changed.

D. *Differentiation of occluded individuals*

A challenging situation may happen when two occupants or more are in the scene. Their objects can make collision each other. We called this situation as merge and split. Merge and split can be detected by using proximity matrix as in [22]. In the merge condition, only centroid feature is used to match the track position to the next possible position since the other three features are not useful when the objects merge. After a group of occupants split, the color feature will be used to match who is who. This can be done since during tracking, the previous color information of the occupants is saved continuously. For this purpose, during the objects merge, their color information is not saved to maintain the color information before they merge.

In the experiments, when more than two occupants merge after that they split, sometimes an occupant remains occluded. Later, the occluded occupant splits. Fig. 7 shows the occluded problem. In the left image, three occupants merged then split become two, the remaining occupant is still occluded. The system detects this event then marks the splited object. The system keeps watching on them until one of the marked objects splits, shown in the right image. When the hiding occupant splits, the system will re-identify each occupant and correct their previous ID number just before they have merged.

Figure 8. Scenario type-1. It shows how the system builds a map. The current images (left) and a map is shown as filled rectangles (right images).

Figure 9. Scenario type-2. It shows how the system monitors the occupants based on the map that has been found with occlusion handling. The current images (left) and the processing result (right images).

TABLE I . TEST RESULTS OF SCENE TYPE 1. THE NUMBER OF DETECTED SEAT BY THE SYSTEM FOR 10 TIMES EXPERIMENTS.

Occupant	**Desk number**				
	#1	***#2***	***#3***	***#4***	***#5***
Sitting	9	10	10	10	9
Leaving	**0**	9	10	10	9

TABLE II. TEST RESULTS OF SCENE TYPE 2. THE NUMBER SHOWS THE SUCCESS RATE OF MONITORING WITHOUT OCCLUSION FOR 10 TIMES EXPERIMENTS.

Sitting area	**Desk number**				
	#1	***#2***	***#3***	***#4***	***#5***
Detected	7	10	10	10	8
Missed	3	0	0	0	2

IV. EXPERIMENT RESULT

This paper demonstrates the usage of the surveillance camera as an intelligent attendance logging system. It mentioned earlier that the system works like a time recorder. The system assists for tracking the hours of occupant attendance. Using this system, the occupants no need to bring special tag or badge. In this section, the environment setup, result, and discussion are described.

A. Environment Setup

A static network camera is used to capture the images from the scene. It is a HLC-83M, a network camera produced by Hunt Electronic. The image size taken from the camera is 320 x 240 pixels. The test room is our laboratory. The camera is placed about 6.24 meters far, 2.63 meters high, and 21° facing down from the horizontal line. The occupant desks and the camera view are orthogonal to get the best view. There are 5 desks as the ground truths.

The room has inner lighting from fluorescent lamps and the windows are covered so the room is free from the sunlight during the test.

B. Analysis of Experiment Result

Visual C++ and OpenCV platform on Intel® Core™2 Quad CPU at 2.33GHz with 4 GB RAMs is used to implement the system. Both offline and online methods are allowed. In the scene without any detected objects, the system ran at 16 frames per second (fps). When the number of incoming objects is increasing, the lowest speed can be achieved is 8 fps.

The algorithm was tested with 2 types of scenarios. The first scenario is sitting occupants with no occlusion (Fig. 8). This scenario demonstrated the working of learning phase. The second scenario is the same as the first scenario but the occupants are allowed crossing each other to make an occlusion (Fig. 9). This scenario demonstrated the merge-split handling.

Table 1 shows the test result of scenario type 1. There are 5 desks as ground truth (Fig. 1). Five occupants entered, sit, and left is set in sequence so that there was no occlusion between occupants. This scenario was repeated 10 times. In the case of the desk number 1, sometimes the occupant's blob merges with his/her neighbor occupant. So, the system cannot detect or track the occupant that sits into desk number 1. In the case of the desk number 5, the occupant's color was similar to the

TABLE III. TEST RESULTS OF SCENE TYPE 2. THE NUMBER OF OCCUPANT MISTAKENLY ASSIGNED IN MERGE-SPLIT CASE FOR 10 TIMES MERGED.

Total occupant	Sitting occupant	Walking occupant	Merge	Split	
				Succeeded	Failed
2	0	2	10	9	1
2	1	1	10	8	2
3	0	3	10	9	1
3	1	2	10	9	1
3	2	1	10	9	1

color of the background image. This caused the occupant produced small blob. The system cannot track the occupant because his/her blob' size becomes too small.

Table 2 shows the test result of scene type 2. The system monitored the occupants based on the map that has been found. The experiments were done 10 times without occlusion. There are some errors that the system failed to recognize the sitting occupant. The system failed to detect the occupant because of the same problems in the previous discussion; the system lost to track the occupant because the occupant has the similar color to the background image so that the occupant suddenly has small blob. The system also failed to recognize the leaving event from desk number 1. The system detects a leaving occupant when the occupant split with his/her seat. Since the desk number 1 is near the edge, the system can not detect the standing-up event. The system still detected that the desk number 1 is always occupied even the corresponding occupant has left that location.

Table 3 shows the test result of scene type 2. The experiments were done 10 times with occlusion. The system should be able to keep tracking the occupants. To test the system, three occupants enter to the scene to make the scenario as shown in Table 3. Some occupants walk through behind the sitting occupant or the occupants just walk and cross each other. Most of the cases, the system can detect which is which after they split. The error happened because of the occupant's color and the sitting occupant. If the occupants have a similar color then the system may get confuse to differentiate them. Another time, when the sitting occupant makes a movement, it creates a blob. However, the system still does not have enough evidence to determine that this blob will change the status of sitting occupant become standing-up occupant. Another occupant walked closer and merged with this blob. After they split, the system confused since the blob had no previous information. As the result, the system missed count the previous track being merged. The ID number of occupant is restored incorrectly

V. CONCLUSION

We have already designed an intelligent attendance logging system by integrating the open source with additional algorithm. The system works in two phases; learning phase and monitoring phase. The system can achieve real-time performance up to 16 fps. We also demonstrate that the system can handle the occlusion up to three occupants considering that the scene seems become too crowded for more than three occupants. While the regular time recorder only reports the time stamp of the beginning and the ending of the occupant's working hour, this system provides more detail about the timing information. Some unexpected behavior may cause an error. For instance, the occupant has the color similar to the background, the desk position, or the occupant moves while sitting.

In the future, the events generated by this system can be used to deliver a message to another system. It is possible to control the environment automatically such as adjust the lighting, playing a relaxation music, setting the air conditioner when an occupant enters or leaves the room. The summary report of the occupant's attendance also can be used for activity analysis. The current system does not include the recognition capability since it only detect whether the working desk is occupied or not. However, if occupant recognition is needed then there are two ways. After the map of sitting areas are found, user may label each sitting area manually or a recognition system can be added. Having a larger view or changing the camera orientation also become our consideration for the future. Fisheye camera or omnidirectional camera are able to capture wider angle in a small room like in the office or laboratory.

VI. ACKNOWLEDGEMENT

This research was supported in part by NSC 99-2221-E-011 -134 and NSC 100-2221-E-011 -051.

REFERENCES

[1] Wikipedia, "Time clock," http://en.wikipedia.org/ wiki/ Time_clock (June 24, 2010).

[2] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafers, "EasyLiving technologies for intelligent environments," Lecture Notes in Computer Science, Volume 1927/2000, pp. 97-119, 2000.

[3] S. -L. Chung and W. –Y. Chen, "MyHome: A residential server for smart homes," Lecture Notes in Computer Science, Volume 4693/2010, pp. 664-670, 2007.

[4] Z. Zhou, X. Chen, Y. –C. Chung, Z. He, T. X. Han, and J. M. Keller, "Activity analysis, summarization, and visualization for indoor human activity monitoring," IEEE Transactions on Circuits and Systems for Video Technology, Volume 18, Issue 11, pp. 1489-1498, 2008.

[5] OpenCV. Available: http://sourceforge.net/projects/ opencvlibrary/

[6] cvBlob. Available : http://code.google.com/p/cvblob/

[7] D. Demirdjian, K. Tollmar, K. Koile, N. Checka, and T. Darrell, "Activity maps for location-aware computing," Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision, pp. 70-75, 2002.

[8] K. Koile, K. Tollmar, D. Demirdjian, H. Shrobe, and T. Darrell, "Activity zones for context-aware computing," Lecture Notes in Computer Science, Volume 2864/2003, pp. 90-106, 2003.

[9] N. Brandie, D. Bauer, and S. Seer, "Track-based finding of stopping pedestrians - A practical approach for analyzing a public infrastructure," Proceeding of IEEE Conference on Intelligent Transportation Systems, pp. 115-120, 2006.

[10] A. Girgensohn, F. Shipman, and L. Wilcox, "Determining activity patterns in retail spaces through video analysis," Proceeding of the 16th ACM International Conference on Multimedia, pp. 889-892, 2008.

[11] B. Morris and M. Trivedi, "An adaptive scene description for activity analysis in surveillance video," Proceeding of IEEE International Conference on Pattern Recognition, 2008.